



Pós-Graduação em Ciência da Computação

“Geração de Carga Sintética para o
Planejamento de Capacidade em
Ambientes de Servidores Virtuais”

Por

Hugo Estanislau Soares Galindo

Dissertação de Mestrado



Universidade Federal de Pernambuco
posgraduacao@cin.ufpe.br
www.cin.ufpe.br/~posgraduacao

RECIFE, MARÇO/2010



Universidade Federal de Pernambuco

CENTRO DE INFORMÁTICA

PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Hugo Estanislau Soares Galindo

“Geração de Carga Sintética para o Planejamento de Capacidade em Ambientes de Servidores Virtuais”

Este trabalho foi apresentado à Pós-Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Mestre em Ciência da Computação.

ORIENTADOR: Prof. Dr. Paulo Romero Martins Maciel

RECIFE, MARÇO/2010

Galindo, Hugo Estanislau Soares

Geração de carga sintética para o planejamento de capacidade em ambientes de servidores virtuais / Hugo Estanislau Soares Galindo. - Recife: O autor, 2010.

xvii, 117 folhas: il., fig., tab.

Dissertação (mestrado) - Universidade Federal de Pernambuco. Cln. Ciência da Computação, 2010.

Inclui bibliografia.

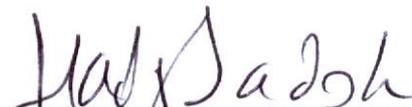
1. Redes de computadores. 2. Ambientes virtuais compartilhados. 3. Métodos de simulação.I. Título.

004.6

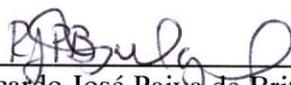
CDD (22.ed.)

MEI-2010-024

Dissertação de Mestrado apresentada por **Hugo Estanislau Soares Galindo** à Pós-Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, sob o título “**Geração de Carga Sintética para o Planejamento de Capacidade em Ambiente de Servidores Virtual**”, orientada pelo **Prof. Paulo Romero Martins Maciel** e aprovada pela Banca Examinadora formada pelos professores:



Prof. Djamel Fawzi Hadj Sadok
Centro de Informática / UFPE



Prof. Ricardo José Paiva de Britto Salgueiro
Departamento de Ciência da Computação e Estatística / UFS



Prof. Paulo Romero Martins Maciel
Centro de Informática / UFPE

Visto e permitida a impressão.
Recife, 04 de março de 2010.



Prof. Nelson Souto Rosa

Coordenador da Pós-Graduação em Ciência da Computação do
Centro de Informática da Universidade Federal de Pernambuco.

Dedico este trabalho a minha mãe Lilian Estanislau, que esteve ao meu lado nas etapas mais importantes da minha vida, sempre me apoiando, aconselhando e incentivando. Por todos os princípios morais e éticos que me foram passados. Pela garra e pela humildade. Pelo exemplo de força e determinação ao transformar sua dor em coragem e esperança no momento mais difícil de sua vida. Em quem me espelho todos os dias em busca de meus ideais. Obrigado.

AGRADECIMENTOS

Gostaria de registrar meus sinceros agradecimentos a todas as pessoas que, direta ou indiretamente, contribuíram para a realização deste trabalho.

Ao Prof. Dr. Paulo Maciel, amigo e orientador, por todo o apoio, disponibilidade, sabedoria, conselhos e enorme paciência, elementos imprescindíveis para o desenvolvimento desta pesquisa; por todas as oportunidades e pela confiança depositada em mim para a elaboração deste trabalho.

Aos professores Djamel Sadok e Ricardo Salgueiro, por terem aceitado o convite para compor a banca examinadora, contribuindo para a melhoria da qualidade deste trabalho.

A todos os integrantes do projeto GCAP-HP, Bruno Silva, Sérgio Galdino e Wagner, por toda contribuição e apoio durante a execução deste trabalho.

A todos os meus companheiros do CIn e participantes do grupo MoDCS. Dentre eles: Alysson Barros, Alexandre Jansen, Almir Guimarães, Bruno Nogueira, Ermeson Andrade, Erica Sousa, Eduardo Tavares, Fábio Chicout, Gustavo Callou, Julian Menezes e Marcelo Marinho. Por todo apoio e ajuda durante toda a minha jornada no mestrado.

A todos os amigos que dividiram e os que ainda dividem apartamento comigo em Recife: Thiago, Lek, Robertinho, Fabiane, Suzanna, Braga, Ícaro e Lucas. Meu agradecimento especial a Fabiane, Thiago, Lek e Robertinho pelos momentos de descontração, conselhos e por todo o apoio.

A toda minha família, em especial a minha mãe, Lilian, minhas irmãs, Carla e Isabella, meus cunhados Dilson e Marçal, e meus sobrinhos Pedro e Bárbara, que direta ou indiretamente sempre me apoiaram e contribuíram para esta vitória.

É melhor atirar-se à luta em busca de dias melhores, mesmo correndo o risco de perder tudo, do que permanecer estático, como os pobres de espírito, que não lutam, mas também não vencem, que não conhecem a dor da derrota, nem a glória de ressurgir dos escombros. Esses pobres de espírito, ao final de sua jornada na Terra não agradecem a Deus por terem vivido, mas desculpam-se perante Ele, por terem apenas passado pela vida.

—BOB MARLEY

RESUMO

Atualmente os sistemas de informação estão presentes em todas as atividades do nosso cotidiano, sendo aplicados em diversas áreas, desde saúde, comunicação e educação até finanças, segurança e entretenimento. Dado o uso disseminado destes sistemas, a indisponibilidade ou desempenho insuficiente dos serviços providos têm sido objetos de atenção particular dos provedores de serviços, infraestruturas, projetistas e da comunidade científica. O planejamento de capacidade destas infraestruturas, permite que os serviços providos atendam a níveis aceitáveis de qualidade (como tempo de resposta, disponibilidade e confiabilidade), fazendo com que os recursos sejam dimensionados de maneira que os custos associados sejam minimizados e os níveis de serviços estabelecidos sejam atendidos. Atualmente, grande parte das infraestruturas computacionais existentes pelo mundo vêm utilizando cada vez mais a tecnologia de virtualização em seus sistemas. A virtualização muda a forma sobre como são utilizados os recursos computacionais. É possível consolidar múltiplas plataformas rodando diferentes sistemas operacionais e aplicações heterogêneas em uma única poderosa e confiável plataforma. Grandes empresas, tais como IBM, *Microsoft* e HP, estão investindo cada vez mais em diversas soluções de virtualização. Neste contexto, este trabalho visa o desenvolvimento de um gerador de carga para auxiliar o planejamento de capacidade de sistemas em um ambiente de servidores virtuais. Durante esta pesquisa foram desenvolvidos mecanismos para geração de cargas de trabalho aplicadas a uma ferramenta de planejamento de capacidade em um ambiente de servidores virtuais (HP VSE), o HP *Capacity Advisor*. A carga sintética gerada realiza a simulação do consumo dos principais recursos computacionais das infraestruturas (processador, memória, disco e rede) e é gerada no formato de um trace suportado pelo *Capacity Advisor*. Desta forma o trace gerado pode ser importado pela ferramenta, possibilitando o estudo e a influência destes recursos no desempenho do sistema para o planejamento de capacidade da infraestrutura computacional.

Palavras-chave: Avaliação de desempenho de sistemas, geração de carga sintética, planejamento de capacidade, ambiente de servidores virtuais.

ABSTRACT

Information systems are increasingly present in all activities of our daily lives and in the most diverse areas of knowledge, from health, education and communication to finance, security and entertainment. Given the widespread use of information systems, performance of services provided have been object of attention of service providers, infrastructure managers, application designers and the scientific community. The capacity planning of these infrastructures, allows that the services provided by the systems meet the appropriate quality levels (such as response time, availability and reliability), minimizing the associated costs and meeting the established service level agreement. Nowadays, many computational infrastructure around the world are increasingly using virtualization technology in their systems. Virtualization changes the way that computing resources are used. You can easily consolidate multiple platforms running different operating systems and heterogeneous applications into a single powerful and reliable platform. Companies such as IBM, Microsoft and HP are increasingly investing in virtualization solutions. This work proposes the development of a workload generator to assist the capacity planning of systems in an environment of virtual servers. During this research were developed mechanisms for generating workloads applied to a tool for capacity planning in an environment of virtual servers (HP VSE - Virtual Server Environment), the HP Capacity Advisor. The synthetic workload simulates the consumption of main computational resources (CPU demand, memory size, disk data rate and network data rate) and it is generated in a trace format compatible with the Capacity Advisor. Thus the trace generated can be imported by the tool, enabling the study of these resources and influence on system performance for capacity planning of computational infrastructure.

Keywords: Performance evaluation, synthetic workload generation, capacity planning, virtual server environments.

SUMÁRIO

Lista de Figuras	xii
Lista de Tabelas	xiv
Lista de Abreviaturas	xv
Capítulo 1—Introdução	1
1.1 Contexto	1
1.2 Motivação	2
1.3 Trabalhos Relacionados	5
1.4 Objetivos	7
1.5 Estrutura da Dissertação	8
Capítulo 2—Fundamentos	10
2.1 Variáveis aleatórias e distribuições de probabilidade	10
2.1.1 Distribuição Normal	12
2.1.2 Distribuição Exponencial	13
2.1.3 Distribuição Erlang	14
2.1.4 Distribuição Geométrica	15
2.1.5 Distribuição de Poisson	16
2.1.6 Distribuição Log-normal	17
2.1.7 Distribuição de Pareto	18
2.1.8 Distribuição de Weibull	20
2.1.9 Distribuição Triangular	21
2.1.10 Distribuição Uniforme	21

2.2	Geração de números aleatórios	22
2.3	Geração de variáveis aleatórias	24
2.3.1	Transformada inversa	25
2.3.2	Composição	26
2.3.3	Convolução	26
2.3.4	Aceitação-Rejeição	27
2.4	<i>Distribution fitting</i>	28
2.4.1	Teste <i>Kolmogorov-Smirnov</i>	28
2.4.2	Teste <i>Anderson-Darling</i>	30
2.4.3	Teste <i>Chi-Squared</i>	31
2.5	Sumário	32
Capítulo 3—Ambiente para a geração de carga		34
3.1	HP VSE - <i>Virtual Server Environment</i>	34
3.1.1	Principais conceitos e terminologias	35
3.1.2	Arquitetura	36
3.1.3	VSE Management Software	39
3.1.3.1	HP <i>Virtualization Manager</i> :	40
3.1.3.2	HP <i>Global Workload Manager</i> (gWLM):	40
3.1.3.3	HP <i>Capacity Advisor</i> :	42
3.1.3.4	HP <i>Integrity Virtual Machines Manager</i> (VM Manager):	44
3.1.3.5	HP <i>Application Discovery</i> :	45
3.2	HP <i>Capacity Advisor</i>	47
3.2.1	Visão Geral	48
3.2.2	Arquitetura	53
3.2.3	Formato dos <i>traces</i>	56
3.3	Sumário	58
Capítulo 4—Metodologia para a concepção do gerador de carga		59
4.1	Visão Geral	59
4.2	Metodologia adotada	61

4.2.1	Compreensão do problema e entendimento do sistema	62
4.2.2	Estudo das metodologias de geração de carga	63
4.2.3	Definição dos cenários de geração de carga	63
4.2.4	Desenvolvimento do <i>kernel</i>	64
4.2.5	Validação do <i>kernel</i>	65
4.2.6	Definição do processo de geração de carga	65
4.2.7	Desenvolvimento do gerador de carga	66
4.2.8	Estudos de caso e avaliação de cenários	66
4.3	Desenvolvimento do ferramental	67
4.4	Validação	69
4.5	Sumário	73
Capítulo 5—Ferramental desenvolvido		75
5.1	Cenários de geração de carga	75
5.1.1	Cenário 01 - <i>ReTrace</i> : Geração de traces sintéticos baseados em traces reais	75
5.1.2	Cenário 02 - <i>Statistic Trace</i> : Geração de traces sintéticos baseados em resumos estatísticos e distribuições de probabilidade	76
5.1.3	Cenário 03 - <i>Statistic ReTrace</i> : Geração de resumos estatísticos e traces sintéticos baseados em traces reais	77
5.2	Processo de geração de carga	77
5.3	<i>Kernel</i>	79
5.4	WGCap - <i>Workload Generator for Capacity Advisor</i>	84
5.4.1	Módulo de geração de carga	84
5.4.2	Módulo de geração de resumos estatísticos	90
5.5	Sumário	91
Capítulo 6—Estudos de Caso		93
6.1	Estudo de caso - <i>ReTrace</i>	93
6.1.1	Geração de um <i>trace</i> sintético baseado em um <i>trace</i> real utilizando o mesmo número de amostras	93
6.1.2	Geração de um <i>trace</i> sintético baseado em um <i>trace</i> real utilizando um número maior de amostras	96

SUMÁRIO	xi
6.2 Estudo de caso - <i>Statistic Trace & Statistic ReTrace</i>	99
6.2.1 Geração de um <i>trace</i> sintético baseado em distribuições de probabilidade	99
6.2.2 Geração de um <i>trace</i> sintético baseado em resumos estatísticos . .	102
6.3 Sumário	106
Capítulo 7—Conclusões	107
7.1 Contribuições, limitações e dificuldades	108
7.2 Trabalhos Futuros	109
Referências	117

LISTA DE FIGURAS

1.1	Virtualização (Exemplo).	3
2.1	Variável aleatória.	11
2.2	Transformada Inversa - Representação Gráfica.	25
2.3	Transformada Inversa - Variáveis aleatórias discretas.	26
2.4	ECDF x Normal CDF.	29
3.1	Domínio de gerenciamento do HP SIM e VSE [Her06].	37
3.2	Arquitetura de funcionamento do VSE [Her06].	39
3.3	HP <i>Virtualization Manager</i>	41
3.4	HP <i>Global Workload Manager</i> (gWLM).	42
3.5	HP <i>Capacity Advisor</i>	45
3.6	HP <i>Integrity Virtual Machines Manager</i>	46
3.7	HP <i>Application Discovery</i>	47
3.8	<i>Historic profile editor</i>	49
3.9	<i>Workload's forecast editor</i>	51
3.10	<i>Capacity Advisor baseline editor</i>	52
3.11	<i>Capacity Advisor scenario</i>	54
3.12	Arquitetura do <i>Capacity Advisor</i>	55
3.13	Arquivo CSV gerado pelo <i>Capacity Advisor</i>	58
4.1	Arquitetura de funcionamento do gerador de <i>traces</i> para o <i>Capacity Advisor</i>	60
4.2	Metodologia adotada para a concepção do gerador de carga.	62
4.3	Fluxograma das atividades adotadas durante a implementação do ferramental.	68
4.4	Função de distribuição acumulada - Amostra real x Amostra empírica.	72
4.5	<i>Kernel</i> - Número de amostras x Tempo médio de geração.	72

4.6	<i>Kernel</i> - Número de amostras x Tempo médio de geração (Cont. distribuição Poisson).	73
5.1	Cenário 01 - <i>ReTrace</i>	76
5.2	Cenário 02 - <i>Statistic Trace</i>	77
5.3	Cenário 03 - <i>Statistic ReTrace</i>	78
5.4	Fluxograma das atividades do processo geração de carga.	78
5.5	<i>NetBeans</i> IDE - Classes implementadas pelo <i>kernel</i>	81
5.6	<i>Screenshot</i> do <i>kernel</i> - Geração de <i>random variates</i>	82
5.7	<i>Screenshot</i> do <i>kernel</i> - Resumo estatístico.	83
5.8	<i>Screenshot</i> do WGCap - Tela inicial.	85
5.9	<i>Screenshot</i> do WGCap - Propriedades do <i>trace</i> sintético.	85
5.10	<i>Screenshot</i> do WGCap - Período das amostras.	87
5.11	<i>Screenshot</i> do WGCap - Seleção das métricas.	88
5.12	<i>Screenshot</i> do WGCap - Seleção da metodologia de geração de carga.	88
5.13	<i>Screenshot</i> do WGCap - Seleção da distribuição utilizada.	89
5.14	<i>Screenshot</i> do WGCap - Definição do resumo estatístico.	90
5.15	<i>Screenshot</i> do WGCap - Geração do <i>trace</i> sintético.	90
5.16	<i>Screenshot</i> do WGCap - Geração de resumos estatísticos.	91
6.1	WGCap - Distribuição empírica (DISK_UTIL).	95
6.2	Função de distribuição acumulada - DISK_UTIL (Real) x DISK_UTIL (Sintético).	95
6.3	<i>Capacity Advisor</i> - Utilização da largura de banda do disco rígido em MB/s.	96
6.4	Função de distribuição acumulada - CPU_UTIL (Real) x CPU_UTIL (Sintético).	98
6.5	<i>Capacity Advisor</i> - Utilização de CPU (<i>trace</i> real).	98
6.6	<i>Capacity Advisor</i> - Utilização de CPU (<i>trace</i> sintético).	99
6.7	WGCap - Distribuição normal (MEM_UTIL).	101
6.8	MEM_UTIL - Histograma dos valores gerados.	101
6.9	<i>Capacity Advisor</i> - Utilização de Memória RAM (Distribuição normal).	102
6.10	WGCap - Definição do resumo estatístico para a métrica DISK_UTIL.	104
6.11	WGCap - Distribuição triangular (DISK_UTIL).	105
6.12	<i>Capacity Advisor</i> - Utilização da largura de banda do disco rígido em MB/s.	106

LISTA DE TABELAS

3.1	VSE - Sistemas operacionais e plataformas de <i>hardware</i> suportadas. . . .	38
3.2	Métricas utilizadas pelo <i>Capacity Advisor</i>	57
4.1	Resultados do teste <i>Kolmogorov-Smirnov</i>	70
4.2	Resultados do teste <i>Anderson-Darling</i>	71
4.3	Resultados do teste <i>Chi-Squared</i>	71
5.1	Distribuições teóricas implementadas pelo <i>kernel</i>	80
6.1	Resumo estatístico - <i>profile legacy01</i> (DISK_UTIL).	103
6.2	Resumo estatístico - Distribuição triangular (DISK_UTIL).	105

LISTA DE ABREVIATURAS

API - *Application Program Interface.*

CDF - *Cumulative Distribution Function.*

CMS - *Central Management Server.*

CP - *Capacity Planning.*

CPU - *Central Processing Unit.*

CSV - *Comma Separated Values.*

DES - *Discret Event System.*

ECDF - *Empirical Cumulative Distribution Function.*

GB - *Gigabyte.*

GHz - *Gigahertz.*

GOF - *Goodness of Fit test.*

GUI - *Graphical User Interface.*

HP - *Hewlett Packard.*

HP gWLM - *HP Global Workload Manager.*

HP SIM - *HP Systems Insight Manager.*

HP-UX - *(Hewlett Packard UniX.*

HP VSE - *HP Virtual Server Environment.*

IBM - *International Business Machines.*

IDE - *Integrated Development Environment.*

IP - *Internet Protocol.*

I/O - *Input/Output.*

JVM - *Java Virtual Machine.*

Mb - *Megabit.*

MB - *Megabyte.*

OVPA - *HP OpenView Performance Agent.*

PMP - *HP Performance Management Pack.*

RAM - *Random Access Memory.*

SO - *Sistema Operacional.*

SPL - *Sun Public License.*

TCP - *Transmission Control Protocol.*

UTIS - *UNIX Time Stamp.*

VA - *Variável Aleatória.*

VM - *Virtual Machine.*

VMM - *Virtual Machine Monitor.*

WGCap - *Workload Generator for Capacity Advisor.*

WWW - *World Wide Web.*

CAPÍTULO 1

INTRODUÇÃO

Este capítulo apresenta uma breve introdução aos conceitos de planejamento de capacidade e virtualização, destacando a importância de se realizar o planejamento de capacidade em ambientes virtuais. Em seguida é apresentado o conceito de geração de cargas sintéticas, as principais motivações desta pesquisa, os trabalhos relacionados, os principais objetivos e contribuições da dissertação, além de sua estrutura.

1.1 CONTEXTO

Os Sistemas de Informação estão presentes em todas as atividades do nosso cotidiano. A internet acelerou de forma significativa a inserção destes sistemas nas mais diversas áreas de conhecimento, desde saúde, educação e comunicação até finanças, segurança e entretenimento, para citar apenas algumas.

Dado o uso disseminado dos sistemas de informação, a indisponibilidade ou desempenho insuficiente dos serviços providos têm sido objetos de atenção particular dos provedores de serviços, infra-estruturas, projetistas de aplicações e da comunidade científica.

O Planejamento de Capacidade (*Capacity Planning - CP* [All08]) consiste na definição dos recursos necessários para que os serviços providos (realizados) pelos sistemas atendam os níveis de qualidade (tempo de resposta, nível de disponibilidade, etc) adequados aos processos que utilizam estes serviços ou recursos, considerando demandas futuras. De fato, para que esta atividade seja realizada com sucesso é necessário que os recursos sejam dimensionados de maneira que os custos associados sejam minimizados e os níveis de serviços estabelecidos sejam atendidos.

Atualmente, grande parte das infra-estruturas computacionais existentes pelo mundo vêm utilizando cada vez mais a tecnologia de *virtualização* [SN05, Sin04, BDF⁺03] em seus sistemas. A virtualização aplicada a sistemas operacionais muda a forma sobre como são utilizados os recursos computacionais. É possível consolidar múltiplas platafor-

mas executando diferentes sistemas operacionais e aplicações heterogêneas em uma única plataforma eficiente e confiável. Isso reduz o número de sistemas a serem gerenciados para a mesma tarefa e operação, o que ajuda a economizar energia e minimizar custos.

A virtualização de sistemas é uma solução de *software* que isola os sistemas operacionais e seus aplicativos de recursos da plataforma de *hardware*. Cada instância de sistema operacional é chamada de *Virtual Machine* (VM) e roda em sua própria partição. O *software* de virtualização, chamado *Virtual Machine Monitor* (VMM), gerencia requisições e atividades do sistema operacional (SO), mudando o controle do *hardware* para cada SO requisitado. Partições separadas isolam falhas ou ataques de *software* para uma única VM, mantendo um alto nível de segurança e disponibilidade do meio virtual. Essa tecnologia dá o controle de cada partição, deixando que se pare ou reinicie qualquer VM sem afetar outras atividades na plataforma [Sin04]. VMs podem ser duplicadas em um único servidor para recuperação em caso de alguma instância do aplicativo falhar, possibilitando maior continuidade de negócios em menos plataformas e por menor custo. Grandes empresas, tais como IBM, *Microsoft* e HP, estão investindo cada vez mais em diversas soluções de virtualização [ibm10, mic10, dyn09a].

A Figura 1.1 ilustra um diagrama com um exemplo de aplicação da virtualização de sistemas. Neste exemplo, vários serviços rodam de maneira independente em diferentes servidores virtuais (Servidor *Proxy*, servidor de arquivos, servidor *Web* e servidor de banco de dados) utilizando apenas um servidor físico.

Neste contexto, este trabalho visa o desenvolvimento de metodologias para geração de carga voltada para o planejamento de capacidade em ambientes de servidores virtuais.

1.2 MOTIVAÇÃO

Ferramentas de Planejamento de Capacidade possibilitam monitorar e avaliar infraestruturas computacionais em função de cargas de trabalho, objetivando a análise da utilização dos recursos, *tuning* e planejamento. Os sistemas a serem avaliados podem estar configurados em uma estrutura de rede ou formarem *clusters* de processamento.

Uma das características importantes deste ambiente de avaliação é o uso de cargas obtidas pelo monitoramento automático de sistemas reais. O resultado da análise dos *traces* associados a essas cargas são avaliados e utilizados como insumos.

Embora as ferramentas de Planejamento de Capacidade utilizem cargas, obtidas a par-

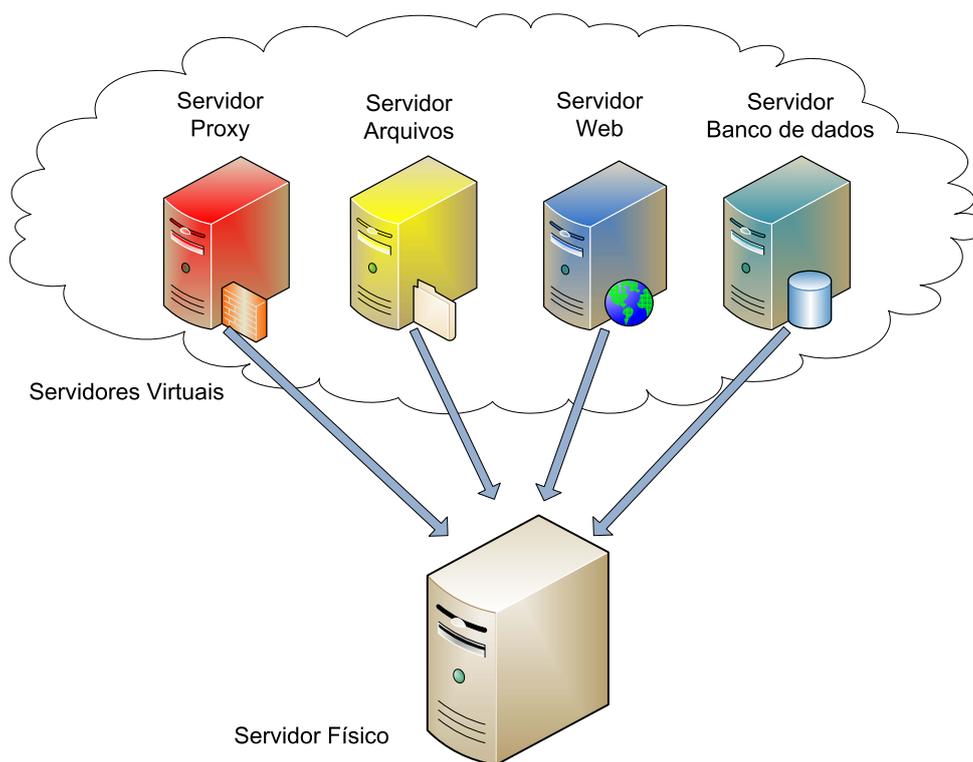


Figura 1.1: Virtualização (Exemplo).

tir de *traces*¹ relativos ao monitoramento de sistemas reais, o que é uma característica de grande interesse prático, diversas linhas de investigação podem ser conduzidas de maneira a tornar o processo de geração de cargas mais flexível e completo [Gan95, GS00, BA94, EBDN00, WST92, ABB⁺06, KRLM08]. É importante salientar a criação de mecanismos que permitam a avaliação de situações em que os cenários sejam de complexidade não trivial ou ainda quando *traces*, obtidos de sistemas reais, não estiverem disponíveis.

A geração de cargas sintéticas vem se destacando e crescendo cada vez mais nas mais diversas áreas da informática. Sua utilização é de extrema importância em ambientes de simulação, na avaliação de desempenho de sistemas e no planejamento de capacidade de infra-estruturas e sistemas computacionais. Atualmente a geração de cargas sintéticas vem sendo aplicada em várias áreas de concentração, como banco de dados, redes de computadores, simulação da utilização de recursos computacionais (como CPU, memória, rede e disco rígido), entre outras.

¹*Trace: Conjunto de informações que identificam as características de uma determinada máquina e descrevem a utilização de seus recursos (Processamento, rede, disco e memória) num determinado período de tempo. Podem ser reais (coletados a partir de uma determinada infra-estrutura computacional) ou sintéticos (gerados por uma ferramenta).*

O objetivo de um gerador de cargas sintéticas é realizar a geração de cargas virtuais (cargas geradas artificialmente) de maneira que o seu comportamento consiga simular o padrão observado em cargas de trabalho retiradas dos sistemas reais em produção a serem estudados. A geração dessas cargas é realizada de acordo com o objetivo específico do problema, onde podem ser utilizadas diversas estratégias para que os valores sejam gerados de maneira satisfatória.

Uma das estratégias de geração de carga bastante utilizada consiste na geração de valores aleatórios baseados em uma distribuição de probabilidade que determine o comportamento da carga real a ser simulada pelo gerador, esses valores são chamados de *Random Variates* [Dev86, Cas08, Gen98]. Os parâmetros de entrada para que a carga sintética seja desenvolvida é gerado a partir de um conjunto de resumos estatísticos retirados de *traces* dos sistemas reais em produção.

Alguns pesquisadores utilizam *traces* retirados de sistemas reais para a realização de experimentos, porém essa abordagem trás alguns problemas [Gan95, GS00, KRLM08]:

- Dificuldade para conseguir autorização dos administradores de sistema para realizar a coleta de *traces*;
- Cada *trace* coletado representa uma medida única de comportamento, o que torna difícil o estabelecimento de um grau de confiança estatístico nos resultados. Às vezes torna-se difícil realizar a distinção entre características reais de desempenho do sistema em teste e comportamentos anômalos do *trace*;
- É muito difícil isolar e/ou modificar características específicas de uma carga de trabalho de um *trace* coletado do sistema;
- *Traces* reais não permitem estudos de cargas de trabalho futuras, uma vez que não é possível coletar informações que ainda não existem.

A geração de cargas sintéticas pode solucionar a maioria dos problemas que surgem com a utilização de *traces* retirados de sistemas reais, trazendo mais flexibilidade para a realização de pesquisas:

- Ao aplicar diferentes sementes (*seeds*) no gerador de números aleatórios, podem ser geradas diversas amostras de uma carga com as mesmas características sendo possível conseguir uma certa confiança estatística;

- Alterações à carga de trabalho podem ser feitas com facilidade, tornando-se possível a geração de cargas de trabalho representando comportamentos futuros.

A principal desvantagem da utilização de cargas sintéticas é o perigo de ocorrer a geração de uma carga que, de alguma maneira, não seja fiel a carga real, comprometendo assim a validade dos resultados.

1.3 TRABALHOS RELACIONADOS

Muitos trabalhos envolvendo a geração de cargas e *traces* sintéticos tem sido desenvolvidos ao longo dos anos nas mais diversas áreas de concentração, dentro de diversos contextos.

Ganger [Gan95] mostra o desenvolvimento de uma abordagem para a validação de geradores sintéticos de requisições de disco rígido, nesta abordagem uma série de dados estatísticos são retirados de requisições de disco rígido reais para que possam ser geradas as cargas sintéticas. Posteriormente, Gómez and Santonja [GS00] apresentam uma nova abordagem para a geração de cargas sintéticas representando requisições de disco rígido que se mostrou muito mais eficiente do que a abordagem mostrada em [Gan95]. O modelo de geração de cargas de disco proposto utiliza o processo estocástico auto-similar nos padrões das requisições de disco, e consegue capturar o comportamento *heavy-tailed* observado em requisições de disco rígido retiradas de sistemas reais, característica essencial para reproduzir o congestionamento do subsistema de disco rígido (*disk subsystem congestion*) [GS00]. Desta forma o gerador consegue fornecer cargas sintéticas muito mais precisas do que nos modelos comumente utilizados.

Barford and Crovella [BC98] apresentam uma metodologia que aplica uma série de observações do uso de um servidor *Web* para criar uma ferramenta para a geração de cargas que representem o tráfego *Web* de maneira realista simulando um conjunto real de usuários acessando um servidor. A ferramenta, chamada *Surge (Scalable URL Reference Generator)* gera referências correspondendo a medições empíricas da 1) Distribuição do tamanho do servidor de arquivo; 2) Distribuição do tamanho das requisições; 3) popularidade relativa dos arquivos; 4) referências de arquivos embutidos; 5) localidade temporal de referência, e 6) períodos ociosos de usuários individuais.

Busari and Williamson [BW02b] mostram o desenvolvimento e a utilização de um gerador de cargas sintéticas para servidores *Web Proxy* chamado *ProWGen* com o intuito de investigar a sensibilidade das políticas de substituição da *cache* dos servidores *Web Proxy*

para cinco características de cargas *web* selecionadas. Três políticas de substituição de *cache* representativas são consideradas no estudo da simulação: *least-recently-used*, *least-frequently-used-with-aging* e *greedydual-size*. Já Markatchev and Williamson [MW02] abordam outra ferramenta chamada *WebTraff*. Essa ferramenta é uma evolução do *ProW-Gen*, fornecendo uma GUI (*Graphical User Interface*) interativa que pode ser utilizada para a modelagem e análise de cargas de servidores *Web Proxy*.

Anderson et al. [ABB⁺06] apresenta a ferramenta SWORD (*Scalable WORKload generator*), um gerador de cargas para sistemas de processamento de dados distribuídos. A ferramenta foi desenvolvida com o intuito de realizar testes de performance em sistemas de processamento de dados, permitindo a geração de carga de uma variedade de aplicações e conteúdos.

Denneulin et al. [DRT04] mostra uma abordagem para a geração de cargas de trabalho voltadas para *clusters* de processamento. O modelo proposto baseia-se na análise de um *log* referente à utilização do *I-cluster* (cluster com 225 processadores), durante o período de um ano. A partir deste *log* foi extraído um padrão representando uma carga de trabalho típica para este tipo de sistema, e posteriormente foi desenvolvida uma estratégia para a geração de carga representando este comportamento de maneira automática, permitindo o planejamento de capacidade dessas infraestruturas. A principal limitação desta abordagem é o fato da geração de carga estar restrita às características observadas no *log* específico utilizado.

Outra abordagem interessante dentro da geração de carga, é a geração de *traces* sintéticos. Barriga and Ayani [BA94] mostram os problemas da utilização de *traces* reais em simulações (*Trace-driven simulation*) e introduzem a abordagem da utilização de *traces* sintéticos, realizando uma avaliação de desempenho entre as duas abordagens. Eeckhout et al. [EBDN00] define outra proposta para a geração de *traces* sintéticos baseados em estatísticas refinadas retiradas de *traces* reais e realiza uma avaliação comparativa entre os *traces* reais e os *traces* gerados.

MacGregor and Gburzynski [SMG01] mostram um gerador de *traces* sintéticos que simula o tráfego IP (*Internet Protocol*) considerando a distribuição de endereços de destino nos pacotes IP que chegam ao roteador. Os *traces* gerados visam possibilitar estudos experimentais em protocolos de roteamento e *cache*.

Khayari et al. [KRLM08] apresenta um gerador de *traces* sintéticos parametrizado, chamado *ParaSynTG - A Parameterized Synthetic Trace Generator for Representation of WWW Traffic*, cujo objetivo é gerar *traces* que representem o tráfego *Web* de maneira

realista, considerando a maioria das propriedades observadas em cargas de trabalho reais do tráfego *WWW*.

Além dos trabalhos mostrados, muitas outras pesquisas interessantes envolvendo a geração de carga e *traces* sintéticos são mostradas em [GS03, Mah97, AW97, BW02a, BW00, SF02, CXZ07, WST92].

A maioria destes trabalhos foca em situações específicas, onde são aplicadas metodologias para a geração de carga para simular o comportamento de alguma situação particular, como a requisição de disco rígido, o tráfego em servidores *web* e *proxy* e a geração de *traces* sintéticos simulando o tráfego de pacotes IP pela *internet*. Esta dissertação mostra o desenvolvimento de um gerador de *traces* sintéticos com o intuito de simular o consumo de vários recursos computacionais, tais como processador, memória, disco e rede, para auxiliar o planejamento de capacidade de sistemas em uma infraestrutura de servidores virtuais. A principal vantagem da abordagem desenvolvida é a possibilidade de geração de cargas adaptáveis a diversas situações, onde poderão ser gerados *traces* a partir de informações observadas em sistemas reais existentes ou baseados em distribuições de probabilidade, permitindo a geração de valores referentes a novas situações e hipóteses, sem que haja a necessidade de uma infraestrutura física.

Parte do trabalho descrito nesta dissertação pode ser encontrado na seguinte publicação: [GSM⁺09].

1.4 OBJETIVOS

Este trabalho concerne o desenvolvimento de conceitos e técnicas para geração de cargas em um ambiente de planejamento de capacidade, as validações destes conceitos, o desenvolvimento de protótipos e o emprego deste ferramental em estudos de caso. Desta forma, a pesquisa tem por objetivo prover meios confiáveis para geração de cenários de carga significativos de forma a possibilitar o planejamento de recursos computacionais, permitindo a realização de estudos sobre os efeitos do número de processadores, capacidade de processamento e mecanismos de armazenamento.

Mais especificamente, este trabalho propõe o desenvolvimento de mecanismos para geração de cargas de trabalho aplicadas a uma ferramenta de planejamento de capacidade em um ambiente de servidores virtuais (*HP VSE - Virtual Server Environment*), o *HP Capacity Advisor* [cap09]. A carga sintética gerada irá simular o consumo dos principais recursos computacionais (processador, memória, disco e rede) e será gerada

no formato de um *trace* suportado pelo *Capacity Advisor*. Desta forma o *trace* gerado poderá ser importado pela ferramenta, possibilitando o estudo e a influência destes recursos no desempenho do sistema para o planejamento de capacidade da infra-estrutura computacional.

Neste sentido, este trabalho apresenta duas linhas de investigação visando o desenvolvimento de metodologias e ferramental para suporte à geração de carga:

- Geração de *traces* sintéticos baseados em *traces* reais;
- Geração de *traces* sintéticos considerando um conjunto significativo de distribuições de probabilidade e resumos estatísticos.

Um dos pontos principais desta pesquisa é a geração de *traces* sintéticos representativos, através da aplicação de diversas metodologias para a geração dos valores a serem armazenados nos *traces*. Poderão ser gerados valores baseados em uma série de distribuições de probabilidades importantes (contínuas e discretas), além de valores baseados em *traces* reais, onde podem ser gerados números aleatórios pertencentes a um determinado comportamento observado em um sistema real em produção. Desta forma o gerador de carga permite uma grande flexibilidade durante a geração dos valores responsáveis por simular o consumo dos recursos computacionais (CPU, memória, disco e rede).

É importante salientar que a estrutura do *Capacity Advisor* permite apenas o planejamento de capacidade através da utilização de *traces* pertencentes a sistemas reais em produção e que estejam configurados em plataformas compatíveis com a metodologia atual de coleta de *traces* para a ferramenta. Esta pesquisa possibilita o planejamento de capacidade com o *Capacity Advisor* em infraestruturas computacionais independentemente da plataforma utilizada e sem a necessidade da utilização de *traces* pertencentes a sistemas reais, já que o gerador irá simular a coleta dos *traces*, que por sua vez, serão gravados em arquivos suportados pelo *Capacity Advisor*.

1.5 ESTRUTURA DA DISSERTAÇÃO

O Capítulo 2 introduz os conceitos fundamentais a serem utilizados na dissertação, tais como: distribuições de probabilidade, geração de números aleatórios, geração de variáveis aleatórias (*random variates*) e técnicas de *distribution fitting*. O Capítulo 3 apresenta o ambiente virtual da HP, destacando os principais componentes e funcionamento do VSE

(*Virtual Server Environment*) e *Capacity Advisor*. A metodologia adotada para o desenvolvimento do gerador dos traces sintéticos para o *Capacity Advisor* é descrita no Capítulo 4. O Capítulo 5 apresenta o gerador de cargas desenvolvido e suas principais funcionalidades. Os estudos de caso envolvendo a avaliação da ferramenta e a sua utilização em diversos cenários são apresentados no Capítulo 6. Por fim, no Capítulo 7 apresentam-se as considerações finais deste trabalho e trabalhos futuros que poderão dar continuidade à pesquisa desenvolvida.

CAPÍTULO 2

FUNDAMENTOS

Este capítulo apresenta os principais conceitos da dissertação. Primeiramente, são introduzidos os conceitos de variáveis aleatórias e distribuições de probabilidade, destacando as principais distribuições contínuas e discretas. Em seguida, é mostrado o processo de geração de números aleatórios, enfatizando a importância do método congruencial linear. Posteriormente é descrito o processo de geração de variáveis aleatórias, onde são demonstrados os principais métodos utilizados para a geração de valores aleatórios baseados em distribuições de probabilidade. Por fim, são apresentados os principais testes de *distribution fitting*.

2.1 VARIÁVEIS ALEATÓRIAS E DISTRIBUIÇÕES DE PROBABILIDADE

Uma *Variável Aleatória (VA)* associa um valor numérico a cada resultado de um experimento aleatório. A palavra aleatória indica que, em geral, este valor só é conhecido após o acontecimento do experimento. Desta forma temos que, dado um experimento aleatório com espaço amostral S , uma VA é uma função que associa um número real a cada elemento do espaço amostral [Tri05]. A função VA é geralmente representada por um X ou qualquer letra maiúscula e possui valor único para cada experimento, determinado aleatoriamente. O valor que a VA pode assumir geralmente é representado por um x ou outra letra minúscula.

A figura 2.1 ilustra um diagrama representando a definição de uma VA, onde “ IR ” representa o conjunto dos números reais, “ S ” representa o espaço amostral, “ s ” representa um elemento do espaço amostral e “ $X(s)$ ” representa a função VA, responsável por associar cada elemento pertencente a “ S ” a um número real.

Uma variável aleatória pode ser classificada como **discreta** ou **contínua** [Tri05, MdOB06]:

Variável aleatória discreta - Numa amplitude determinada, admite um número finito

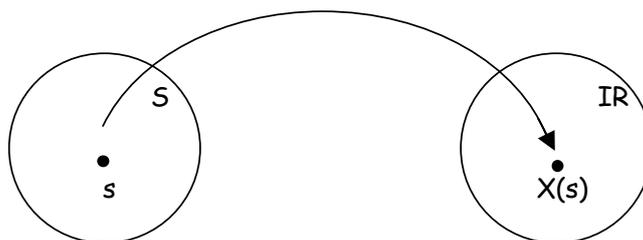


Figura 2.1: Variável aleatória.

de valores ou possui uma quantidade enumerável de valores.

Ex.: Número de valores determinados em um lançamento de um dado; número de pessoas em uma fila de banco.

Variável aleatória contínua - Admite um número infinito de valores, formados por qualquer intervalo dos números reais, caracterizando assim um conjunto não enumerável de valores. Pode ser associada a uma mensuração em uma escala contínua.

Ex.: Altura de um homem adulto selecionado aleatoriamente.

Além de identificar os valores de uma variável aleatória, podem ser atribuídas probabilidades para representar a frequência de ocorrência de cada um desses valores. Quando conhecemos todos os possíveis valores de uma variável aleatória com suas respectivas probabilidades de ocorrência, temos uma *Distribuição de probabilidade* [Mon03, Tri05, MdOB06].

Uma distribuição de probabilidade é uma correspondência que associa probabilidades aos valores de uma variável aleatória, ou seja, é uma função $P(X = x) = f(x)$, que fornece a probabilidade de ocorrência de cada valor que uma variável aleatória pode assumir. Para qualquer conjunto de eventos ou proposições existem muitas maneiras de determinar probabilidades, de forma que a escolha de uma ou outra distribuição é equivalente a criar diferentes hipóteses sobre os eventos ou proposições em questão.

Há várias formas equivalentes de se especificar uma distribuição de probabilidade. A mais comum constitui-se em especificar uma função densidade da probabilidade. Daí, a probabilidade de um evento ou proposição é obtida pela integração da função densidade. A função distribuição pode ser também especificada diretamente. Em uma dimensão, a função distribuição é chamada de função distribuição acumulada. [Mon03].

Uma distribuição de probabilidade também pode ser representada através de um histograma de probabilidade, onde o eixo horizontal (X) representa os possíveis valores

da variável aleatória e o eixo vertical (Y) representa as probabilidades referentes a cada valor. O histograma de probabilidade permite a visualização da forma da distribuição de probabilidade.

Uma distribuição é chamada de **distribuição discreta** se for definida em um conjunto contável e discreto, tal como o subconjunto dos números inteiros; ou é chamada de **distribuição contínua** se tiver uma função distribuição contínua que pode tomar qualquer valor dentro de um intervalo especificado dos números reais, tal como uma função polinomial ou exponencial.

Dentre as distribuições discretas importantes, pode-se citar a distribuição de Poisson, a distribuição Binomial, a distribuição Binomial negativa, a distribuição Geométrica e a distribuição de *Maxwell-Boltzmann*. Dentre as distribuições contínuas, pode-se citar a distribuição Normal, a distribuição Erlang, a distribuição Weibull, a distribuição Log-Normal, a distribuição de Pareto e a distribuição Exponencial [Mon03].

Algumas das principais distribuições e probabilidades, contínuas e discretas, serão brevemente detalhadas nas próximas subseções.

2.1.1 Distribuição Normal

A distribuição normal, também chamada de distribuição *gaussiana*, é uma família de distribuições contínuas de probabilidade [Mon03]. Cada membro da família pode ser definido por dois parâmetros, localização e escala: a média (μ) e a variância (σ^2 - desvio padrão ao quadrado), respectivamente. A distribuição normal padrão é a uma distribuição normal com a média zero e a variância um. *Carl Friedrich Gauss* associou-se a este conjunto de distribuições quando ele analisou dados astronômicos utilizando-as, e definiu a equação de sua função densidade de probabilidade. A distribuição normal também é conhecida como curva do sino (*bell curve*) pois o gráfico da sua densidade de probabilidade lembra um sino.

A importância da distribuição normal como um modelo de quantitativos fenômenos é devido em parte ao *Teorema do Limite Central* [Mon03]. Muitas medições, que vão desde fenômenos psicológico a fenômenos físicos (em particular, o ruído térmico - *thermal noise*) podem ser aproximadas, em diferentes graus, pela distribuição normal. A distribuição normal também é importante pela sua relação com a estimativa “menos-quadrados” (*least-squares*), um dos mais simples e mais antigo método de estimativa estatística. Na teoria das probabilidades, a distribuição normal aparece como a distribuição

limite de várias famílias de distribuições discretas e contínuas.

Função densidade de probabilidade - A função densidade de probabilidade (*probability density function* - pdf) da distribuição normal possui a seguinte forma:

$$f(x; \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2}, x \in \mathbb{R}$$

onde $\sigma > 0$ é o desvio padrão e o parâmetro real μ é média.

Função distribuição acumulada - A função distribuição acumulada (*cumulative distribution function* - cdf) de uma distribuição de probabilidade, avaliada em um número x , é a probabilidade do evento em que uma variável aleatória X com essa distribuição seja inferior ou igual a x . A função distribuição acumulada da distribuição normal é expressa em termos da função densidade de probabilidade do seguinte modo:

$$F(x; \mu, \sigma) = \int_{-\infty}^x f(u; \mu, \sigma) du = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^x e^{-(u-\mu)^2/2\sigma^2} du, x \in \mathbb{R}.$$

Média e variância - Média = μ ; Variância = σ^2 .

2.1.2 Distribuição Exponencial

A distribuição exponencial [Mon03] é uma distribuição contínua aplicada em muitos problemas em engenharia e em empresas nas áreas de serviços e manufaturas, em geral denominados problemas de fila de espera (teoria das filas). Essa distribuição calcula probabilidades para um certo tempo e espaço entre eventos sucessivos, ocorrendo em um processo de Poisson homogêneo.

É comumente usada para tempos entre chegadas a, por exemplo, cabines de pedágios. Pode-se usar também esta distribuição para calcular probabilidades de falhas, quanto ao estudo de confiabilidade; ou seja, o intervalo de tempo decorrido entre o instante em que uma peça é sujeita a um esforço mecânico e o instante em que ocorre uma falha (a quebra da peça, por exemplo). Quando os serviços prestados por uma empresa para clientes externos ou internos são de duração variável, a distribuição exponencial é indicada também para analisar esses experimentos; por exemplo, a duração do atendimento do caixa de um banco ou de postos de saúde, o tempo de operação sem interrupção de um equipamento etc [Tri06].

A distribuição exponencial é definida pelo único parâmetro λ referente a taxa média do processo, que reflete em média, quantos eventos ocorrem em uma unidade de tempo.

Função densidade de probabilidade - A função densidade de probabilidade (*probability density function* - pdf) da distribuição exponencial possui a seguinte forma:

$$f(x; \lambda) = \begin{cases} \lambda e^{-\lambda x}, & \text{para } x \geq 0 \\ 0, & \text{para } x < 0. \end{cases}$$

Função distribuição acumulada - A função distribuição acumulada é definida por:

$$F(x; \lambda) = \begin{cases} 1 - e^{-\lambda x}, & \text{para } x \geq 0 \\ 0, & \text{para } x < 0. \end{cases}$$

Média e variância - Média = $\frac{1}{\lambda}$; Variância = $\frac{1}{\lambda^2}$.

2.1.3 Distribuição Erlang

A distribuição Erlang foi desenvolvida por *A.K. Erlang* para examinar o número de chamadas telefônicas que poderiam ser feitas ao mesmo tempo pelos operadores das estações de comutação telefônica. Este trabalho em engenharia de tráfego telefônico foi expandido para considerar tempos de espera em sistemas de filas em geral. A distribuição atualmente é utilizada no domínio dos processos estocásticos.

A distribuição Erlang é uma distribuição contínua, que tem um valor positivos para todos os números reais maiores que zero, e é composta por dois parâmetros: a forma (*shape*) k , que é um número inteiro, e a taxa (rate) λ , que é um número real [Mon03]. Quando o parâmetro forma k for igual a 1, a distribuição é simplificada para a distribuição exponencial. A distribuição também é definida às vezes utilizando o inverso do parâmetro taxa, a escala θ (tempo) = $\frac{1}{\lambda}$. É importante salientar que a distribuição Erlang é um caso especial da distribuição Gama onde o parâmetro forma k é um número inteiro. Na distribuição Gama, este parâmetro é um número real.

Função densidade de probabilidade - A função densidade de probabilidade (*probability density function* - pdf) da distribuição Erlang possui a seguinte forma:

$$f(x; k, \lambda) = \frac{\lambda^k x^{k-1} e^{-\lambda x}}{(k-1)!}.$$

Função distribuição acumulada - A função distribuição acumulada da distribuição Erlang é definida por:

$$f(x; k, \lambda) = 1 - \sum_{n=0}^{k-1} \frac{e^{-\lambda x} (\lambda x)^n}{n!}.$$

Média e variância - Média = $\frac{k}{\lambda}$; Variância = $\frac{k}{\lambda^2}$.

2.1.4 Distribuição Geométrica

A distribuição geométrica [Mon03] é uma distribuição de probabilidade que pode ser constituída por duas funções de probabilidade discretas:

- A distribuição de probabilidade referente ao número X de experimentações de *Bernoulli* necessárias para alcançar um sucesso, suportadas pelo conjunto $\{1, 2, 3, \dots\}$ (uma experimentação de Bernoulli é uma experiência cujo resultado seja aleatório e possa ser qualquer um de dois resultados possíveis, “sucesso” e “falha”);
- A distribuição de probabilidade do número $Y = X - 1$ de insucessos antes do primeiro sucesso, suportadas pelo conjunto c .

A distribuição geométrica é uma distribuição “*memoryless*”. Isso significa que ao repetir uma experiência até que ocorra o primeiro sucesso, dado que o primeiro sucesso não ocorreu ainda, a distribuição de probabilidade condicional do número de experimentações adicionais não depende de quantas falhas foram observadas. O jogo de dados ou os lançamentos de uma moeda, por exemplo, não têm uma “memória” destas falhas. A distribuição geométrica é de fato a única distribuição discreta “*memoryless*”.

Função de probabilidade - Se a probabilidade de sucesso de cada tentativa é p , então a probabilidade de n tentativas serem necessárias para ocorrer um sucesso é definida por:

$$P(X = n) = (1 - p)^{n-1} p$$

para $n = 1, 2, 3, \dots$. De forma equivalente, a probabilidade de serem necessários n insucessos antes do primeiro sucesso é:

$$P(X = n) = (1 - p)^n p$$

para $n = 1, 2, 3, \dots$

Função distribuição acumulada - A função de distribuição cumulativa da distribuição geométrica é definida por:

$$1 - (1 - p)^k.$$

Média e variância - Média = $\frac{1}{p}$; Variância = $\frac{1-p}{p^2}$.

2.1.5 Distribuição de Poisson

Na teoria da probabilidade e na estatística, a distribuição de Poisson [Mon03] é uma distribuição de probabilidade discreta. Ela expressa, por exemplo, a probabilidade de um certo número de eventos ocorrerem num dado período tempo, caso estes ocorram com uma taxa média conhecida e caso cada evento seja independente do tempo decorrido desde o último evento.

A distribuição foi descoberta por *Siméon-Denis Poisson* (1781-1840) e publicada, conjuntamente com a sua teoria da probabilidade, em 1838 no seu trabalho *Recherches sur la probabilité des jugements en matières criminelles et matière civile* (“Inquérito sobre a probabilidade em julgamentos sobre matérias criminais e civis”) [Tri05]. O trabalho focava-se em certas variáveis aleatórias N que contavam, entre outras coisas, o número de ocorrências discretas (por vezes chamadas de “chegadas”) que tinham lugar durante um intervalo de tempo de determinado comprimento.

Função de probabilidade - A função de probabilidade (*Probability mass function* - pmf) da distribuição de Poisson é definida por:

$$f(k; \lambda) = \frac{\lambda^k e^{-\lambda}}{k!}, k \in \{1, 2, 3, \dots\}$$

onde k é o número de ocorrências de um evento e λ é um número real positivo, igual ao número esperado de ocorrências que ocorrem durante o intervalo dado.

A distribuição de Poisson pode ser aplicada a sistemas com um grande número de possíveis eventos, sendo cada um deles um evento raro. Um exemplo clássico é o decaimento nuclear dos átomos. A distribuição de Poisson também pode ser utilizada

para o número de eventos em outros intervalos especificados, como a distância, área ou volume [Tri06, MdOB06].

Função distribuição acumulada - A função distribuição acumulada da distribuição de Poisson é definida por:

$$\frac{\Gamma(k+1, \lambda)}{k!}, k \geq 0$$

onde $\Gamma(x, y)$ é a função Gamma Incompleta (*Incomplete gamma function*).

Média e variância - Média = λ ; Variância = λ .

2.1.6 Distribuição Log-normal

A distribuição Log-normal, também conhecida por log normal ou lognormal [Mon03], é baseada na distribuição normal. Ela descreve uma variável X , onde $\log(X)$ é normalmente distribuído. Ela é válida para valores de X que são maiores do que zero. Se X é uma variável aleatória com uma distribuição normal, então $Y = \exp(X)$ tem uma distribuição log-normal; de igual modo, se Y é log-normalmente distribuído, então $\log(Y)$ é normalmente distribuído.

A distribuição lognormal descreve muitas populações que ocorrem naturalmente. Uma variável pode ser modelada como log-normal se ela pode ser pensada como o produto multiplicativo de muitos pequenos fatores independentes. Por exemplo, a taxa de retorno a longo prazo em um estoque investimento pode ser considerada como sendo o produto das taxas de retorno diárias. Na comunicação sem fio, a atenuação causada por sombreamento ou desvanecimento lento a partir de objetos aleatórios são muitas vezes assumidos log-normalmente distribuídos. Nas indústrias de mineração e extração, observou-se que quando o valor de um item é proporcional à dimensão, a população está provavelmente distribuída log-normalmente, com poucos elementos valiosos e lotes de itens não-comerciais, as biociências podem ter uma percepção diferente [Tri06, MdOB06].

Função densidade de probabilidade - A função densidade de probabilidade da distribuição log-normal tem a seguinte forma:

$$f(x; \mu, \sigma) = \frac{1}{x\sigma\sqrt{2\pi}} e^{-\frac{(\ln(x)-\mu)^2}{2\sigma^2}}$$

Para $x > 0$, onde μ e σ são a média e o desvio padrão da variável do logaritmo natural (por definição, a variável do logaritmo é normalmente distribuída). Esses parâmetros são, neste contexto medidos em *neper*, desde que logaritmos naturais sejam utilizados. O *neper* é muitas vezes utilizado para expressar relações de tensão e amplitudes correntes em circuitos elétricos (ou pressão em acústica), enquanto que o decibel é utilizado para expressar relações de potência (No contexto da comunicação sem fio normalmente medido em decibéis) [SC89].

Função distribuição acumulada - A função distribuição acumulada da distribuição log-normal é definida por:

$$\frac{1}{2} + \frac{1}{2} \operatorname{erf}\left[\frac{\ln(x)-\mu}{\sigma\sqrt{2}}\right].$$

Média e variância - Média = $e^{\mu+\frac{\sigma^2}{2}}$; Variância = $(e^{\sigma^2} - 1)e^{2\mu+\sigma^2}$.

2.1.7 Distribuição de Pareto

A distribuição de Pareto é uma distribuição inclinada para a esquerda definida em termos da moda e do fator forma (*shape*) [Mon03]. Trata-se de uma distribuição de cauda pesada (*heavy-tailed*), o que significa que uma variável aleatória seguindo uma distribuição Pareto pode ter valores extremos.

Esta distribuição foi originalmente desenvolvida para descrever a distribuição de renda, seguindo o fundamento de que uma grande percentagem da população possui baixa renda, enquanto apenas algumas pessoas possuem renda elevada [Tri05]. Esta ideia é, por vezes expressa de forma mais simples como o princípio de Pareto ou a “regra 80 – 20”, onde se descreve uma série de situações, no suporte ao cliente significa que 80% dos problemas vêm de 20% dos clientes ou em economia onde 80% da riqueza estão nas mãos de 20% da população.

A distribuição de Pareto, que contém o nome do economista italiano *Vilfredo Pareto*, é uma distribuição de probabilidade que coincide com os fenômenos sociais, científicos, geofísicos, atuariais, e muitos outros tipos de fenômenos observáveis. Os seguintes exemplos são por vezes visto como Pareto-distribuídos aproximadamente [Tri06]:

- Os tamanhos dos assentamentos humanos (poucas cidades, muitas aldeias / vilas);

- A distribuição do tamanho dos arquivos no tráfego da *Internet* que utilizam o protocolo TCP (muitos arquivos pequenos, poucos arquivos grandes);
- Núcleos de condensação de *Bose-Einstein* condensado próximo do zero absoluto;
- Os valores das reservas de petróleo nos campos de petróleo (um pequeno número de grandes campos, muitos pequenos campos);
- A distribuição do comprimento de *jobs* atribuído a supercomputadores (alguns grandes, muitos pequenos);
- O retorno do preço padronizado em estoques individuais;
- Tamanho das partículas de areia ;
- Tamanho dos meteoritos ;
- O número de espécies por gênero (Existe subjetividade envolvida: A tendência para dividir um gênero em dois ou mais aumenta com o número de espécies no mesmo);
- As áreas queimadas nos incêndios florestais;
- Gravidade de acidentes de grandes prejuízos para determinadas linhas de negócio, como a responsabilidade civil geral, comercial automotiva, e os trabalhadores compensação.

Função densidade de probabilidade - A função densidade de probabilidade de uma distribuição de Pareto possui a seguinte forma:

$$f(x; a, b) = \frac{ab^a}{x^{a+1}}, \text{ para } x \geq b$$

onde $a > 0$ é um parâmetro de distribuição, chamado de parâmetro forma (*shape parameter*) e $b > 0$ é o parâmetro escala (*scale parameter*). A distribuição é suportada no intervalo $[b, \infty)$.

Função distribuição acumulada - A função distribuição acumulada da distribuição de Pareto é definida por:

$$F(x; a, b) = 1 - \left(\frac{b}{x}\right)^a.$$

Média e variância - Média = $\frac{ab}{a-1}$, para $a > 1$ (se $a \leq 1$, o valor esperado é infinito);
 Variância = $\frac{ab^2}{(a-2)(a-1)^2}$, para $a > 2$ (se $b \leq 2$, a variância é infinita).

2.1.8 Distribuição de Weibull

A distribuição de Weibull [Mon03], nomeada pelo seu criador Waloddi Weibull, é uma distribuição de probabilidade contínua introduzida em 1933. É bastante utilizada em estudos de tempo de vida de equipamentos, estimativa de falhas e no campo da análise dos dados. Devido à sua flexibilidade, essa distribuição consegue simular o comportamento de outras distribuições, tais como as distribuições normal e a exponencial.

Esta distribuição de probabilidade é composta por dois parâmetros: forma - β (*shape*), e escala - α (*scale*). Se a taxa de insucesso diminui ao longo do tempo, então $\beta < 1$. Se a taxa de insucesso é constante ao longo do tempo, então $\beta = 1$. Se a taxa de insucesso aumenta ao longo do tempo, então $\beta > 1$.

Um entendimento da taxa de insucesso pode fornecer discernimento sobre o que está causando as falhas:

- Uma taxa de falhas decrescente poderia sugerir “mortalidade infantil”. Isto é, elementos defeituosos falham precocemente e a taxa de insucesso diminui com o tempo que se saiam da população;
- A taxa de falhas constante sugere que os itens fracassam a partir de eventos aleatórios;
- Um aumento da taxa de insucesso sugere “desgaste” - as partes são mais propensas a falhar com o passar do tempo.

Quando $\beta = 1$, a distribuição de Weibull reduz à distribuição exponencial. Quando $\beta = 3.4$, a distribuição de Weibull parece semelhante à distribuição normal.

Função densidade de probabilidade - A função densidade de probabilidade de uma distribuição de Weibull possui a seguinte forma:

$$f(x; \alpha, \beta) = \begin{cases} \frac{\beta}{\alpha^\beta} x^{\beta-1} e^{-(\frac{x}{\alpha})^\beta}, & \text{para } x > 0 \\ 0, & \text{para } x \leq 0. \end{cases}$$

Função distribuição acumulada - A função distribuição acumulada da distribuição de Weibull é definida por:

$$F(x; \alpha, \beta) = \begin{cases} 1 - e^{-\left(\frac{x}{\alpha}\right)^\beta}, & \text{para } x > 0 \\ 0, & \text{para } x \leq 0. \end{cases}$$

Média e variância - Média = $\alpha\Gamma\left(1 + \frac{1}{\beta}\right)$; Variância = $\alpha^2\Gamma\left(1 + \frac{2}{\beta}\right) - \mu^2$, onde Γ é a função *Gamma*.

2.1.9 Distribuição Triangular

Em probabilidade e estatística, a distribuição triangular é a distribuição de probabilidade contínua que possui um valor mínimo a , um valor máximo b e uma moda c , de modo que a função densidade de probabilidade seja zero para os extremos (a e b), e afim entre cada extremo e a moda, de forma que o seu gráfico forme um triângulo [Tri05, MdOB06].

Função densidade de probabilidade - A função densidade de probabilidade de uma distribuição triangular possui a seguinte forma:

$$f(x; a, b, c) = \begin{cases} \frac{2(x-a)}{(b-a)(c-a)}, & \text{para } a \leq x \leq c \\ \frac{2(b-x)}{(b-a)(b-c)}, & \text{para } c \leq x \leq b \\ 0, & \text{qualquer outro caso.} \end{cases}$$

Função distribuição acumulada - A função distribuição acumulada da distribuição triangular é definida por:

$$F(x; a, b, c) = \begin{cases} \frac{(x-a)^2}{(b-a)(c-a)}, & \text{para } a \leq x \leq c \\ 1 - \frac{(b-x)^2}{(b-a)(b-c)}, & \text{para } c \leq x \leq b \end{cases}.$$

Média e variância - Média = $\frac{a+b+c}{3}$; Variância = $\frac{a^2+b^2+c^2-ab-ac-bc}{18}$.

2.1.10 Distribuição Uniforme

Em estatística e probabilidade, a distribuição uniforme é a distribuição de probabilidades contínua mais simples de conceituar: a probabilidade de se gerar qualquer ponto em um intervalo contido no espaço amostral é proporcional ao tamanho do intervalo [Tri05, MdOB06].

Seja $[a, b]$ o espaço amostral, então temos que a função densidade de probabilidade, para $a \leq x \leq b$, é:

$$f(x) = \frac{1}{b-a}$$

Se $x > b$ ou $x < a$ então a função densidade de probabilidade é igual a 0.

A aplicação da distribuição uniforme é extremamente importante na informática, onde a maioria das linguagens de programação, pacotes estatísticos ou planilhas de cálculo possuem um gerador de números aleatórios, que geram os valores a partir de uma distribuição uniforme, com valores entre 0 e 1. Sua utilização também pode se aplicar na simulação de outras distribuições de probabilidade. Qualquer outra distribuição contínua, na qual a função distribuição acumulada seja invertível, pode ser simulada a partir da distribuição uniforme [Tri06].

Função densidade de probabilidade - A função densidade de probabilidade de uma distribuição uniforme possui a seguinte forma:

$$f(x) = \begin{cases} f(x) = \frac{1}{b-a}, & \text{para } x \in [a, b] \\ 0, & \text{qualquer outro caso.} \end{cases}$$

Função distribuição acumulada - A função distribuição acumulada da distribuição uniforme é definida por:

$$F(x) = \begin{cases} 0, & \text{para } x < a \\ \frac{x-a}{b-a}, & \text{para } a \leq x < b \\ 1, & \text{para } x \geq b. \end{cases}$$

Média e variância - Média = $\frac{a+b}{2}$; Variância = $\frac{(b-a)^2}{12s}$.

2.2 GERAÇÃO DE NÚMEROS ALEATÓRIOS

A geração de números aleatórios é um processo bastante utilizado em diversos campos da ciência, tais como simulação, otimização, probabilidade, estatística, entre outras áreas. Por exemplo, em simulação, utiliza-se a geração de números aleatórios para simular a chegada de pessoas em uma fila, a fim de avaliar o tempo de espera; para simular a chegada de automóveis em um semáforo, com o propósito de avaliar a melhor forma de calibrá-lo, etc. Em otimização, pode-se utilizar tal processo em algoritmos genéticos, a fim de produzir indivíduos de uma população; e no processo *Ant Colony Optimization*, a

fim de gerar indivíduos na região de busca. A geração de números aleatórios também é extremamente importante em simulações de sistemas de eventos discretos (*Discret Event System* - DES), onde é essencial a geração de números aleatórios que a representem, por exemplo, tempo de serviço ou intervalo de tempo que pertençam a alguma distribuição de probabilidade [Cas08].

Variáveis aleatórias de qualquer distribuição podem ser obtidas transformando variáveis aleatórias de uma distribuição uniforme ao longo do intervalo $[0, 1]$. Variáveis aleatórias geradas a partir da distribuição $U[0, 1]$ são chamadas de **números aleatórios**. Assim, o termo “**gerador de números aleatórios**” refere-se a um mecanismo responsável por gerar amostras aleatórias com base na distribuição uniforme - $U[0, 1]$ [Cas08].

Para gerar números aleatórios, há diversos geradores, geralmente contidos em pacotes de softwares, em calculadoras, em aplicativos como Excel, Minitab e similares. Esses geradores, na verdade, implementam funções que geram números pseudo-aleatórios, já que a partir de um valor inicial (semente - *seed*), geram uma sequência fixa de números, como pode ser visto em [L'E98], [Rip90] e [Hel98].

As principais técnicas utilizadas para a geração de números aleatórios baseiam-se no *Método Congruencial Linear* [Cas08]:

- Método congruencial misto;
- Método congruencial multiplicativo;
- Método congruencial aditivo.

O *método congruencial misto* gera uma sequência de números inteiros pseudo-aleatórios no intervalo 0 a $m - 1$, com base na seguinte relação recursiva (onde cada número é calculado a partir do anterior):

$$X_{n+1} = (aX_n + c) \text{ mod } m, \text{ com } n = 0, 1, 2, \dots$$

Onde temos,

X_0 : Semente (*seed*) - referente ao primeiro valor das amostras;

a : Constante multiplicativa - número inteiro ($a < m$);

c : Incremento - número inteiro ($c < m$);

m : Módulo - número inteiro que representa o número de valores diferentes a serem gerados.

A quantidade de números consecutivos numa sequência antes de se voltar a repeti-los é designada por “**período da sequência**”. Durante a geração de números aleatórios, o valor máximo que pode ser alcançado pelo período é m .

O **método congruencial multiplicativo** é um caso particular do método congruencial misto em que $c = 0$, ou seja:

$$X_{n+1} = (aX_n) \bmod m, \text{ com } n = 0, 1, 2, \dots \text{ e } X_0 = \text{semente.}$$

O **método congruencial aditivo** é outro caso particular do método congruencial misto em que $a = 1$ e c é um dos números aleatórios que precedem X_n na sequência.

2.3 GERAÇÃO DE VARIÁVEIS ALEATÓRIAS

Um dos principais recursos oferecidos por linguagens e ambientes de simulação é a habilidade de poder gerar sequências de valores (*streams*) que estão distribuídos segundo alguma distribuição de probabilidade particular. Os valores que compõe uma sequência deste tipo são chamados de variáveis aleatórias (*Random Variates*) [Dev86, Gen98, Cas08]. A geração desses valores é fundamental para a geração de cargas sintéticas e, desta forma, será de extrema importância para a execução deste trabalho.

Existem vários métodos utilizados para a geração de valores aleatórios associados a uma enorme quantidade de distribuições de probabilidade. Os métodos para a geração de variáveis aleatórias mais utilizados são [Cas08]:

- Transformada inversa;
- Composição;
- Convolução;
- Aceitação-Rejeição.

A utilização de cada método irá depender da distribuição que for utilizada. Os métodos serão descritos brevemente à seguir.

2.3.1 Transformada inversa

Suponhamos que queremos gerar uma variável aleatória X com uma função de distribuição de probabilidade $F(x)$ contínua e estritamente crescente. Nessas condições poderemos gerar uma variável aleatória com distribuição F utilizando o seguinte algoritmo:

1. Gerar um número aleatório U com $U(0, 1)$ - Distribuição Uniforme.
2. Retornar $X = F^{-1}(U)$, onde onde F^{-1} é a inversa da função acumulada.

A Figura 2.2 ilustra o funcionamento deste processo graficamente.

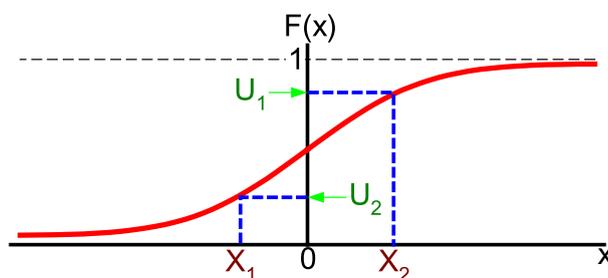


Figura 2.2: Transformada Inversa - Representação Gráfica.

Ao longo do eixo vertical $F(x)$ os valores são sorteados randômicamente sobre a distribuição uniforme. Se traçarmos uma reta partindo deste ponto sorteado no eixo vertical até encontrar a curva da transformada, e deste ponto para baixo em direção ao eixo horizontal, teremos então o valor resultante (a random variate), que segue uma segunda distribuição.

Também podemos aplicar o método da transformada inversa a variáveis discretas. Para gerar variáveis aleatórias discretas, com uma distribuição $F(x)$ podem ser utilizados os seguintes passos:

1. Gerar um número aleatório U com $U(0, 1)$ - Distribuição Uniforme.
2. Determinar o menor inteiro positivo I tal que: $U \leq F(x_I)$, e retornar $X = x_I$.

A Figura 2.3 ilustra o funcionamento deste processo graficamente.

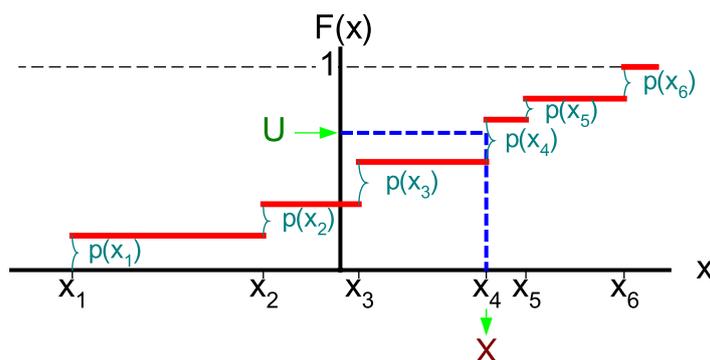


Figura 2.3: Transformada Inversa - Variáveis aleatórias discretas.

2.3.2 Composição

Este método se aplica quando uma função de distribuição de probabilidade é uma combinação de outras funções de distribuição, ou seja:

$$F(x) = \sum_{j=1}^{\infty} p_j F_j(x)$$

em que p_j é positivo e $F(x) = \sum_{j=1}^{\infty} p_j F_j(x)$. Nessas condições podemos aplicar o seguinte algoritmo:

1. Gerar um inteiro positivo aleatório, tal que $P[J = j] = p_j$ para $j = 1, 2, \dots$
2. Retornar X com distribuição F_j .

2.3.3 Convolução

O método da convolução é muito comum em aplicações em que a variável aleatória desejada X é a soma de outras variáveis.

$$X = Y_1 + Y_2 + \dots + Y_n$$

em que cada Y tem distribuição G . O método mostra-se bastante simples, onde é utilizado o seguinte algoritmo:

1. Gerar Y_1, Y_2, \dots, Y_n com distribuição G .
2. Retornar $X = Y_1 + Y_2 + \dots + Y_n$.

2.3.4 Aceitação-Rejeição

O método da Aceitação-Rejeição é um método muito útil e de aplicação geral para a geração de variáveis aleatórias. Este método é geralmente utilizado quando há uma certa complexidade para a execução das outras metodologias de geração de variáveis aleatórias descritas anteriormente.

Vamos assumir que queremos gerar variáveis aleatórias com distribuição $F(x)$ e densidade $f(x)$. Para aplicar o método, necessitamos especificar alguma função $g(x)$ que "majoriza" a função densidade para todo o x , ou seja: $g(x) \geq f(x)$ para todo x . Esta função é chamada de *função majorante* e não é única. De fato, a escolha de uma função majorante mais conveniente pode melhorar a eficiência desta técnica.

Como $g(x)$ em geral não é uma função densidade, determinamos uma constante de normalização, c , o que nos permite transformá-la em uma função densidade. Em particular definimos:

$$c = \int_{-\infty}^{\infty} g(x) dx$$

supondo que $g(x)$ é escolhido de forma que $c < \infty$, definimos uma função densidade de probabilidade $h(x)$ dada por:

$$h(x) = \frac{g(x)}{c}$$

A ideia é gerar uma variável aleatória Y com a densidade $h(x)$ o mais eficiente possível, e então utilizar uma regra simples para decidir se Y deve ser aceita como uma variável aleatória válida a partir de $f(x)$ ou não. Depois de deduzidas as funções anteriores, é o seguinte algoritmo:

1. Gerar Y com densidade $h(x)$.
2. Gerar U com $U(0, 1)$, independente de Y .
3. Se $U \leq \frac{f(Y)}{g(Y)}$ retornar $X = Y$;
caso contrário, voltamos ao ponto 1 e tentamos novamente.

2.4 DISTRIBUTION FITTING

Os testes de *Distribution fitting*, também conhecidos como *Goodness of Fit tests* - (GOF), são responsáveis por medir a compatibilidade de uma amostra aleatória com uma função de distribuição de probabilidade teórica. Em outras palavras, esses testes mostram o grau de compatibilidade (ajuste) em que a distribuição selecionada se ajusta aos dados selecionados [Gen98, NIoSTN03].

O procedimento geral consiste na definição de um teste estatístico (*statistic*) formado por uma função responsável por calcular a discrepância entre os valores dos dados observados e os valores esperados sob o modelo em questão (hipótese). Posteriormente, é definida a probabilidade de obtenção dos dados que possuem um valor maior que os valores esperados durante a realização do teste estatístico, assumindo que a hipótese é verdadeira. Esta probabilidade é chamada de nível de confiança e o valor máximo esperado é denominado valor crítico [NioSTN03].

Dentre os principais testes de *distribution fitting* os mais utilizados são: o teste *Kolmogorov-Smirnov*, o teste *Anderson-Darling* e o teste *Chi-Squared*. Estes testes serão brevemente detalhados a seguir.

2.4.1 Teste *Kolmogorov-Smirnov*

O teste *Kolmogorov-Smirnov*, também conhecido como teste K-S [CLR67], é usado para determinar se uma amostra proveniente de uma população pertence a uma distribuição específica. O princípio deste teste baseia-se na comparação da curva da função de distribuição acumulada empírica (*empirical cumulative distribution function* - ECDF) dos dados, com a função de distribuição teórica em hipótese.

Dado uma quantidade N de dados ordenados Y_1, Y_2, \dots, Y_N , a função de distribuição acumulada empírica (ECDF) é definida como:

$$E_N = \frac{n(i)}{N}$$

onde $n(i)$ é o número de valores menores que Y_i e o conjunto Y_1, Y_2, \dots, Y_N está ordenado do menor ao maior valor. Esta função aumenta em $1/N$ o valor de cada ponto dos dados ordenados.

A figura 2.4 ilustra um gráfico de uma função de distribuição empírica para 100

amostras aleatórias normais, com uma função de distribuição acumulada teórica normal. O teste K-S baseia-se na distância máxima entre estas duas curvas.

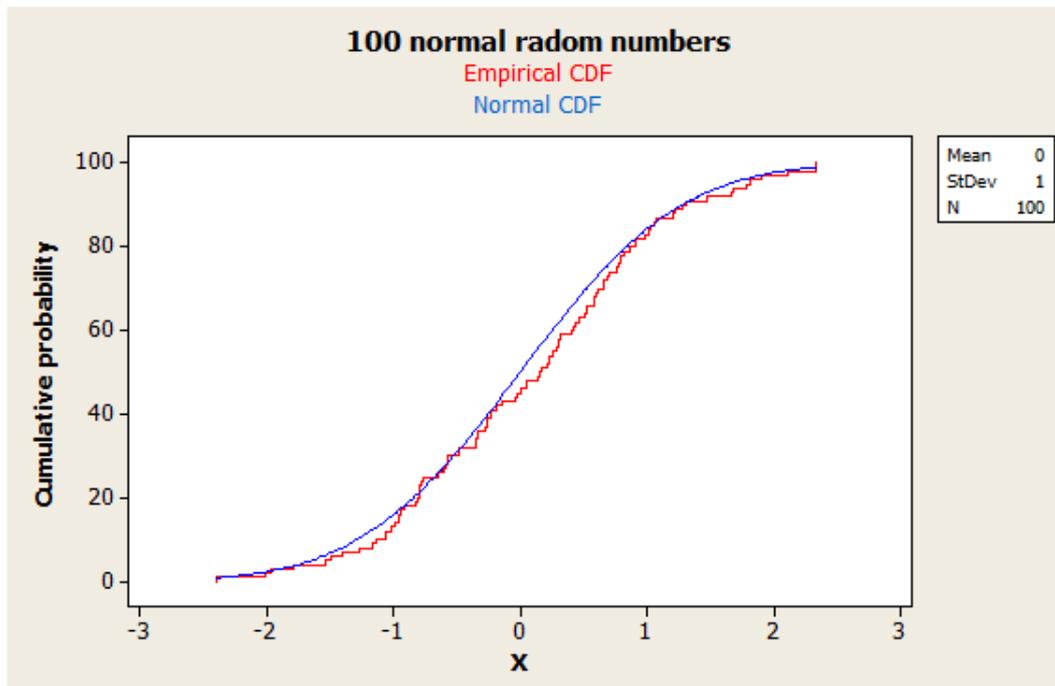


Figura 2.4: ECDF x Normal CDF.

Uma característica interessante deste teste é que a distribuição da estatística do teste K-S em si não depende da distribuição acumulada da função que está sendo testada. Apesar desta vantagem, o teste K-S possui algumas limitações importantes:

- Só se aplica a distribuições contínuas, entre as quais podemos destacar as distribuições: normal, log-normal, Weibull e exponencial;
- Tende a ser mais sensível no centro das distribuições do que com em suas extremidades;
- A distribuição teórica deve ser completamente especificada. Ou seja, se os parâmetros de localização (*location*), escala (*scale*) e forma (*shape*) forem estimados a partir dos dados, a região crítica do teste K-S não é mais válido.

A estatística do teste *Kolmogorov-Smirnov* é definida por:

$$D = \max \left(F(Y_i) - \frac{i-1}{N}, \frac{i}{N} - F(Y_i) \right), 1 \leq i \leq N$$

onde F é a distribuição acumulada teórica da distribuição que está sendo testada, devendo ser uma distribuição contínua completamente especificada. O resultado desta estatística fornece a distância máxima entre distribuição acumulada teórica e a distribuição acumulada empírica dos dados. Os hipóteses resultantes possíveis são:

H_0 - **Hipótese nula (Null hypothesis)** : Os dados seguem a distribuição teórica especificada;

H_a - **Hipótese alternada (Alternate hypothesis)** : Os dados não seguem a distribuição teórica especificada.

A hipótese sobre distribuição é rejeitada de acordo com o nível de significância escolhido (α - alfa) se a estatística do teste, D , for maior do que o valor crítico, obtido a partir de uma tabela pré-definida de acordo com o valor de α .

2.4.2 Teste *Anderson-Darling*

O teste *Anderson-Darling* [Ste74] também é utilizado para determinar se uma amostra proveniente de uma população pertence a uma distribuição específica. Este teste é uma modificação do teste *Kolmogorov-Smirnov*, tendo a vantagem de ser mais sensível que o teste K-S, pelo fato de dar mais peso aos pontos das caudas da distribuição. Assim, valores pequenos da estatística de *Anderson-Darling* indicam que a distribuição estima melhor os dados [Ste74].

O teste de *Anderson-Darling* faz uso de distribuições específicas para o cálculo dos valores críticos. Isto trás a vantagem de permitir um teste mais sensível e a desvantagem de que os valores críticos devem ser calculados para cada distribuição utilizada. Atualmente, tabelas de valores críticos estão disponíveis para as distribuições normal, lognormal, Weibull, exponencial, e distribuições de logística [Ste74, Ste76, Ste77].

Para estabelecer um critério de rejeição ou não rejeição do modelo (distribuição de probabilidade), é formulado o seguinte teste de hipótese:

H_0 - **Hipótese nula (Null hypothesis)** : Os dados seguem a distribuição teórica especificada;

H_a - **Hipótese alternada (Alternate hypothesis)** : Os dados não seguem a distribuição teórica especificada.

A estatística do teste *Anderson-Darling* é definida por:

$$A^2 = -N - \sum_{i=1}^N \frac{(2i-1)}{n} [\ln F(Y_i) + \ln(1 - F(Y_{n+1-i}))]$$

onde F é a função de distribuição acumulada da distribuição específica. Observe que Y_i são os dados ordenados [NioSTN03].

Os valores críticos ou de rejeição para o teste de *Anderson-Darling* dependem da distribuição específica que está sendo testada. O teste é unicaudal e a hipótese nula (H_0) é rejeitada de acordo com o nível de significância escolhido (α - alfa), se o teste estatístico fornecer valor superior ao crítico. É importante salientar que este teste pode ser ajustado (pode ser multiplicado por uma constante, a qual usualmente depende do tamanho da amostra, N). Estas constantes podem ser encontradas nos trabalhos de Stephens [Ste74, Ste76, Ste77] ou em alguns livros como [NioSTN03].

2.4.3 Teste *Chi-Squared*

O teste *Chi-Squared* [SC89], também conhecido como teste “Qui-Quadrado”, é mais uma alternativa utilizada para determinar se uma amostra proveniente de uma população pertence a uma distribuição específica. Ao contrário dos testes *Kolmogorov-Smirnov* e *Anderson-Darling*, que são restritos a distribuições de probabilidade contínuas, o teste *Chi-Squared* pode ser aplicado a distribuições discretas, tais como as distribuições geométrica, poisson e binomial.

Uma característica importante deste teste é a possibilidade de sua aplicação a qualquer distribuição univariada em que possa ser calculada sua respectiva função de distribuição acumulada. O teste *chi-squared* é aplicado a dados agrupados em *bins* (ou seja, os dados são divididos e agrupados em classes), desta forma, o valor da estatística utilizada no teste depende da forma como os dados são agrupados em *bins*. Os dados podem ser agrupados em intervalos de probabilidades ou largura iguais. A primeira abordagem é geralmente mais aceitável, pois lida muito melhor com os picos dos dados [NioSTN03].

É importante ressaltar que uma deficiência deste teste é a necessidade de um tamanho mínimo suficiente de amostras para que a aproximação do teste *chi-squared* seja válida.

Novamente, para estabelecer um critério de rejeição ou não rejeição do modelo (distribuição de probabilidade), é formulado o seguinte teste de hipótese:

H_0 - **Hipótese nula (Null hypothesis)** : Os dados seguem a distribuição teórica especificada;

H_a - **Hipótese alternada (Alternate hypothesis)** : Os dados não seguem a distribuição teórica especificada.

Para o teste *Chi-Squared*, os dados são divididos em k bins e sua estatística é definida por:

$$\chi^2 = \sum_{i=1}^k \frac{(O_i - E_i)^2}{E_i}$$

onde O_i é a frequência observada pelo bin i , e E_i é a frequência esperada pelo bin i . Esta frequência é calculada por:

$$E_i = N (F(Y_u) - F(Y_l))$$

onde F é a função de distribuição acumulada para a distribuição sendo testada, Y_u é o limite superior para a classe i , Y_l é o limite inferior para a classe i e N é o tamanho da amostra.

A estatística do teste *Chi-Squared* segue, aproximadamente, uma distribuição *chi-squared* com $(k - c)$ graus de liberdade, onde k é o número de bins e $c =$ número de parâmetros estimados para a distribuição + 1 (incluindo os parâmetros localização, escala e forma). Por exemplo, para uma distribuição de Weibull 3-parâmetros, temos $c = 4$. Desta forma, a hipótese de que os dados são de uma população com uma distribuição especificada é rejeitada se

$$\chi^2 > \chi_{(\alpha, k-c)}^2$$

onde $\chi_{(\alpha, k-c)}^2$ é a inversa da função distribuição acumulada da distribuição *chi-squared* (*percent point function* - ppf) com $k - c$ graus de liberdade e um nível de significância α .

2.5 SUMÁRIO

Este capítulo apresentou os principais conceitos fundamentais utilizados nesta dissertação. Primeiramente, foram introduzidos os conceitos de variáveis aleatórias e distribuições de

probabilidade, onde foram mostradas suas propriedades, além de algumas das principais distribuições de probabilidade contínuas e discretas. Subsequentemente, foi mostrado o processo de geração de números aleatórios, destacando o método congruencial linear misto, aditivo e multiplicativo. Posteriormente, foram descritas as principais metodologias utilizadas para a geração de variáveis aleatórias (*random variates*), onde foram demonstrados os métodos da transformada inversa, composição, convolução e aceitação-rejeição. Por fim, foram apresentados os principais testes de *distribution fitting*, responsáveis por medir a compatibilidade de uma amostra de dados com uma função de distribuição de probabilidade teórica.

CAPÍTULO 3

AMBIENTE PARA A GERAÇÃO DE CARGA

Este capítulo apresenta os principais conceitos referentes ao ambiente onde o gerador de cargas deve atuar. Inicialmente é introduzido o ambiente de servidores virtuais HP VSE (*Virtual Server Environment*), onde são descritas suas principais características, composição e funcionamento. Posteriormente é apresentado o *HP Capacity Advisor*, ferramenta de planejamento de capacidade pertencente ao VSE, destacando seu funcionamento, estrutura e arquitetura.

3.1 HP VSE - VIRTUAL SERVER ENVIRONMENT

O VSE (*Virtual Server Environment* - “Ambiente de Servidores Virtuais”) [vse09, int09, Her06, dyn09a], é uma solução de virtualização desenvolvida pela HP (*Hewlett Packard*) que fornece um ambiente de gerenciamento e monitoramento de servidores virtuais e suas aplicações. Este ambiente é constituído por um conjunto integrado de produtos e tecnologias multi-plataforma, composto pelo *HP VSE Management Software* [dyn09a], que atua ajudando a maximizar a utilização dos recursos do servidor, provendo a visualização, configuração, a política de *workloads* (cargas de trabalho), o gerenciamento de aplicações, e ferramentas de planejamento da capacidade para a otimização dos recursos no ambiente.

O *HP VSE Management Software* oferece um grupo de aplicações, acessíveis através de um ambiente gráfico integrado, para o gerenciamento de servidores físicos, servidores lógicos, máquinas virtuais, servidores *blades*, partições virtuais, aplicações e *workloads*. Sua instalação deve ser realizada em servidores *HP Integrity*, *HP 9000* ou *HP ProLiant*, podendo ser configurado nos sistemas operacionais *HP-UX* (versões 11i v1, 11i v2, ou 11i v3) ou *Microsoft Windows 32-bits* [dyn09a].

As próximas subseções detalham brevemente os principais conceitos e terminologias referentes ao VSE, além de seus principais componentes, sua arquitetura e funcionamento.

3.1.1 Principais conceitos e terminologias

HP-UX : O HP-UX (*Hewlett Packard UniX*) [hpu10] é uma implementação proprietária do sistema operacional *Unix* desenvolvida pela HP, lançada em 1986. Sua primeira versão, o HP-UX 1.0, baseou-se no sistema *Unix System III* e posteriormente passou a se basear no sistema *Unix System V*. Este sistema operacional é compatível com os sistemas *HP 9000 PA-RISC* e *HP Integrity* com processadores *Itanium* da *Intel*. Sua versão atual é o HP-UX 11i v3 (*Update 2*), cujas principais características envolvem alta performance, alta disponibilidade e facilidade de gerenciamento. Esta versão do HP-UX permite as seguintes configurações de *hardware*:

- Capacidade para 128 processadores;
- Disponibilidade para até 2 TB (*terabytes*) de memória principal;
- Sistema de arquivo com no máximo 32 TB;
- Tamanho de arquivo máximo de 16 TB;
- Capacidade de armazenamento para 100 milhões de ZB (*zettabytes*).

HP Systems Insight Manager (HP SIM) : O *HP Systems Insight Manager* (HP SIM) [Her06, dyn09a] é uma ferramenta de gerenciamento centralizado que permite a gestão e o monitoramento de diversos sistemas operacionais e suas configurações. Os *softwares* de gerenciamento do VSE (*HP VSE Management Software*) estão estreitamente integrados com o HP SIM, que fornece uma interface, acessível através de qualquer navegador *web* (*browser*), contendo todos os recursos necessários para a realização do gerenciamento das diversas funcionalidades do VSE.

Central Management Server (CMS) : O CMS, ou Servidor Central de Gerenciamento (*Central Management Server*) [Her06, dyn09a], é o servidor principal no domínio de gerenciamento do VSE. Nele são instalados os principais componentes necessários para se monitorar o VSE, tais como o HP SIM e os *softwares* de gerenciamento do VSE (*HP VSE Management Software*). Todas as operações centrais do HP SIM são iniciadas a partir do CMS.

Management Domain (Domínio de gerenciamento) : O *Management Domain* (domínio de gerenciamento) [Her06, dyn09a] representa o conjunto de sistemas monitorados e controlados pelo HP SIM e pelos *softwares* de gerenciamento do VSE.

Managed System (Sistema Gerenciado) : O *Managed System* (Sistema Gerenciado) [Her06, dyn09a] representa cada sistema que esteja sob o controle do HP SIM.

System List (Lista de Sistemas) : A Lista de sistemas (*System List*) [Her06, dyn09a] é um conjunto de sistemas gerenciados que são agrupados para fins de administração e gerenciamento das infraestruturas. Listas de sistemas personalizadas podem ser criados para facilitar o gerenciamento dos sistemas, podendo ser formadas através da especificação do conjunto de atributos de uma consulta, tais como “todos os servidores HP-UX”, ou selecionando uma lista específica de servidores, tais como “servidor A”, “servidor B”, etc.

Workloads (Cargas de trabalho) : Os *workloads* [Her06, dyn09a], ou cargas de trabalho, representam uma coleção de processos pertencentes a uma imagem de SO executando em um sistema gerenciado. Podemos citar como exemplos de *workloads* os processos pertencentes a uma aplicação ou todos os processos utilizados por um usuário específico.

System Page (Página do Sistema) : Uma página do sistema (system page) [Her06, dyn09a] é uma página de relatórios provida pelo HP SIM que fornece informações e *links* para os sistemas pertencentes ao domínio de gerenciamento.

3.1.2 Arquitetura

A Figura 3.1 ilustra um exemplo com a representação dos domínios de gerenciamento pertencentes ao HP SIM e ao conjunto de *softwares* para a gerência do VSE (*VSE Management Software*) [dyn09a]. É importante salientar que o domínio de gerenciamento do HP SIM é mais abrangente do que o domínio do VSE, permitindo a utilização de uma grande variedade de sistemas, entre os quais podemos destacar os servidores *HP NonStop*, *HP ProLiant*, HP 9000 PA-RISC, servidores HP *Integrity*, servidores IA-32, estações de trabalho IA-32, impressoras e *switches* de rede. Além disso o HP SIM também suporta uma grande variedade de sistemas operacionais, tais como o HP-UX, *Linux*, *Microsoft Windows*, *OpenVMS* e *HP NonStop*. Pode-se observar na Figura 3.1 que o domínio de gerenciamento referente ao HP SIM envolve todos os sistemas e dispositivos encontrados neste exemplo.

O domínio de gerenciamento referente ao *VSE Management Software* funciona como um subconjunto do domínio de gerenciamento do HP SIM, sendo restrito apenas às

plataformas de *hardware* dos servidores HP *Integrity*, HP 9000 PA-RISC e HP *ProLiant*, além de suportar somente os sistemas operacionais HP-UX, *Linux* e *Windows*. Conforme pode ser observado, o CMS também pode fazer parte do domínio de gerenciamento do *VSE Management Software* (ver Figura 3.1), no entanto devem ser configurados em servidores HP *Integrity* ou HP *ProLiant*, com os sistemas operacionais HP-UX ou *Microsoft Windows*. A Tabela 3.1 detalha as plataformas de *hardware* e os respectivos sistemas operacionais suportados pelo VSE, tanto nos CMS's quanto nos sistemas gerenciados (*managed systems*) [dyn09a].

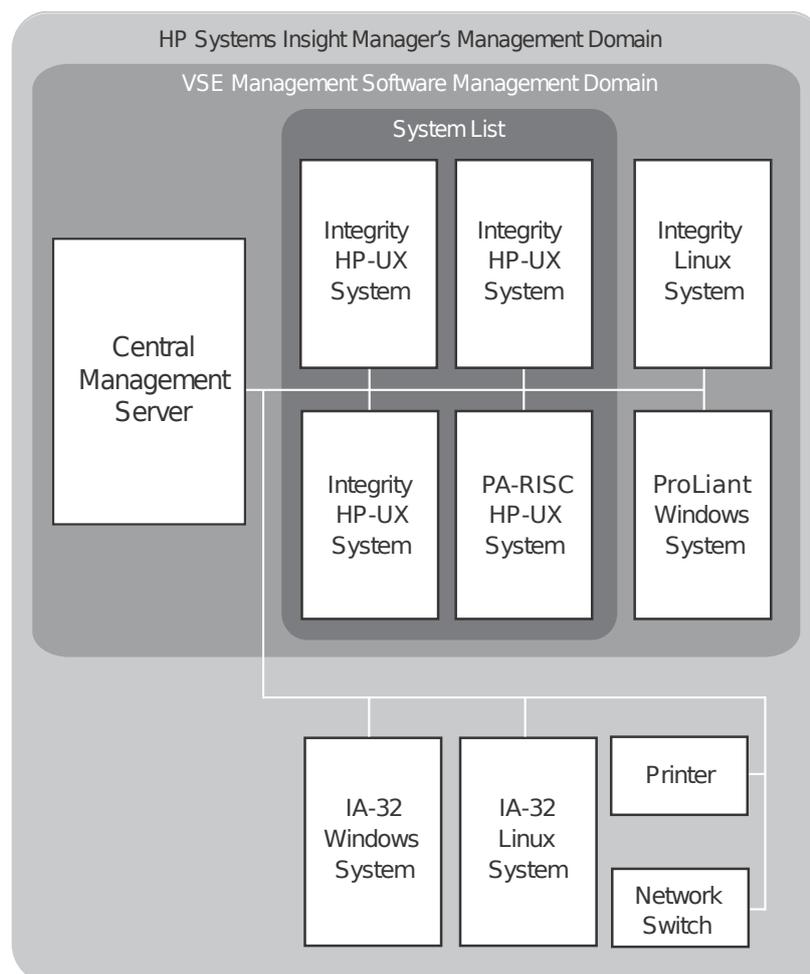


Figura 3.1: Domínio de gerenciamento do HP SIM e VSE [Her06].

Outra característica importante detalhada na Figura 3.1, é a representação de uma lista de sistemas (*System List*) dentro do domínio de gerenciamento do *VSE Management Software*. Neste exemplo é ilustrada uma lista composta por quatro servidores que possuem o sistema operacional HP-UX instalado. Estas listas podem ser criadas com base em

Tabela 3.1: VSE - Sistemas operacionais e plataformas de *hardware* suportadas.

Infraestrutura	Plataforma de <i>Hardware</i>	Sistema Operacional
CMS	HP <i>Integrity</i>	HP-UX 11i v1, 11i v2, or 11i v3
	HP <i>ProLiant</i> ML/DL	32-bit <i>Microsoft Windows</i>
	HP <i>BladeSystem c-Class ProLiant</i>	32-bit <i>Microsoft Windows</i>
<i>Managed systems</i>	HP <i>Integrity</i> e HP 9000	HP-UX 11i v1, 11i v2, or 11i v3
	HP <i>Integrity</i>	64-bit <i>Microsoft Windows</i>
	HP <i>Integrity</i>	<i>Linux</i>
	HP <i>ProLiant</i> ML/DL	32-bit <i>Microsoft Windows</i>
	HP <i>BladeSystem c-Class ProLiant</i>	32-bit <i>Microsoft Windows</i>

dois métodos: Através da especificação individual de cada sistema que deverá fazer parte da lista de sistemas, como por exemplo, uma lista contendo todos os servidores de banco de dados existentes em uma determinada infraestrutura (independentemente do sistema operacional configurado em cada servidor); ou através de uma consulta, onde deverão ser selecionados os servidores com características em comum, como por exemplo, uma lista determinada pelos servidores que possuem o mesmo sistema operacional instalado (exemplo ilustrado na Figura 3.1). Com a utilização do primeiro método, caso seja introduzido um novo servidor no domínio de gerenciamento com as características definidas pela lista gerada, sua adição à lista deverá ser realizada manualmente, caso contrário a lista será mantida com os mesmos sistemas inicialmente agrupados. Já através da utilização do segundo método, caso seja introduzido um novo servidor no domínio de gerenciamento com as características definidas pela lista gerada, sua adição à lista será realizada automaticamente, atualizando o grupo de sistemas definidos.

A arquitetura de funcionamento do VSE é ilustrada na Figura 3.2 [Her06, dyn09a, dyn09b]. Todo o controle do VSE é realizado por uma infraestrutura unificada de gerenciamento através do *HP Systems Insight Manager* (HP SIM), onde podem ser acessados todos os componentes pertencentes ao conjunto composto pelo *VSE Management Software*. Através desta infraestrutura, os servidores virtuais podem ser gerenciados de maneira centralizada e organizada, possibilitando a otimização de seu funcionamento e uma melhor utilização de seus recursos. Na Figura 3.2 também podem ser observados três dos principais componentes pertencentes ao conjunto de aplicações do VSE: O *HP Capacity Advisor*, responsável pelo planejamento de capacidade dos sistemas pertencentes ao domínio de gerenciamento do VSE; o *HP Virtualization Manager*, responsável pela visualização dos sistemas pertencentes ao domínio de gerenciamento em diversos níveis de

detalhes, servindo como ponto central para realização das configurações no ambiente; e o HP *Workload Manager*, responsável pelo gerenciamento de políticas para as cargas de trabalho (*workloads*) pertencentes a cada sistema. Estes e outros componentes pertencentes a família de *softwares* do VSE são detalhados na próxima subseção.

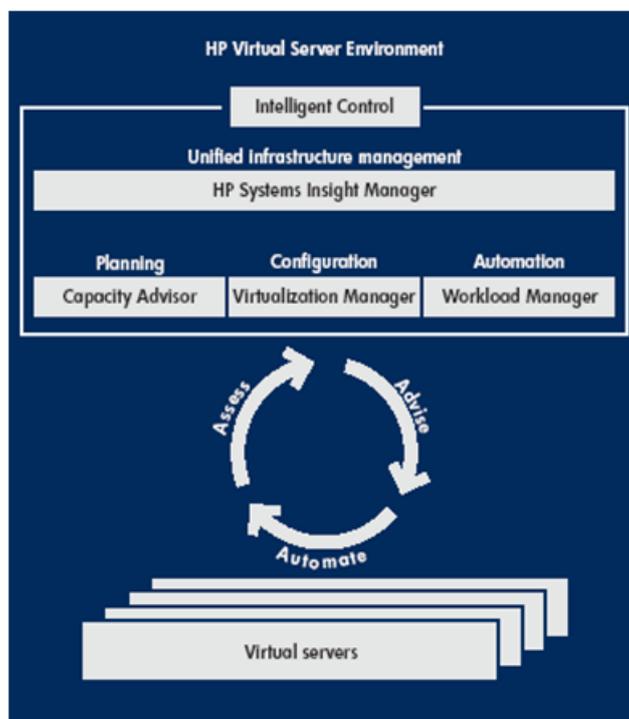


Figura 3.2: Arquitetura de funcionamento do VSE [Her06].

3.1.3 VSE Management Software

Como pôde ser observado, o HP VSE fornece um controle inteligente e simplificado de seus servidores virtuais através de uma família de *softwares* totalmente integrados para o planejamento, gestão e automação dos sistemas pertencentes ao ambiente. Este grupo de *softwares* compõe o VSE *Management Software* [dyn09a], conjunto responsável pelo gerenciamento de todas as funcionalidades fornecidas pelo VSE, através dos seguintes componentes: HP *Virtualization Manager*, HP *Global Workload Manager* (gWLM), HP *Capacity Advisor*, HP *Integrity Virtual Machines Manager* (VM Manager) e HP *Application Discovery*. Estes componentes são brevemente detalhados a seguir.

3.1.3.1 HP *Virtualization Manager*: O HP *Virtualization Manager* é um *software* de visualização que permite ao usuário observar todos os sistemas pertencentes ao VSE em diversos níveis de detalhes. Esta aplicação permite a criação de um inventário do VSE, funcionando como um ponto central a partir do qual é possível configurar os recursos do servidor para a sua utilização ótima. Todos os sistemas e *workloads* são exibidos a partir de um *framework* cuja interface deve ser acessada via *browser* através do HP SIM.

As principais funcionalidades fornecidas pelo *Virtualization Manager* são listadas abaixo:

- Exibição das relações hierárquicas entre os sistemas e as suas métricas de utilização atuais;
- Customização do nível de detalhamento mostrado sobre o ambiente de servidores virtuais (VSE);
- Acesso facilitado as ferramentas adicionais de gerenciamento do VSE (*VSE Management Software*) para a realização do gerenciamento e configuração dos sistemas e *workloads*;
- Criação de resumos com informações sobre o ambiente VSE configurado;
- Configuração dos recursos do servidor para a sua utilização ótima.

A Figura 3.3 ilustra um *screenshot* do HP *Virtualization Manager* sendo acessado através do HP SIM. Neste exemplo, são exibidos todos os sistemas pertencentes a este ambiente de servidores virtuais específico e seus respectivos percentuais de utilização de CPU.

3.1.3.2 HP *Global Workload Manager* (gWLM): O gWLM (Global Workload Manager) é um gerenciador de *workloads* (cargas de trabalho) multi-plataforma, que funciona como uma máquina inteligente de controle e implantação de políticas para o gerenciamento automatizado das cargas de trabalho pertencentes ao VSE, com o intuito de aperfeiçoar a utilização dos recursos de cada servidor para atingir os níveis de serviço esperado.

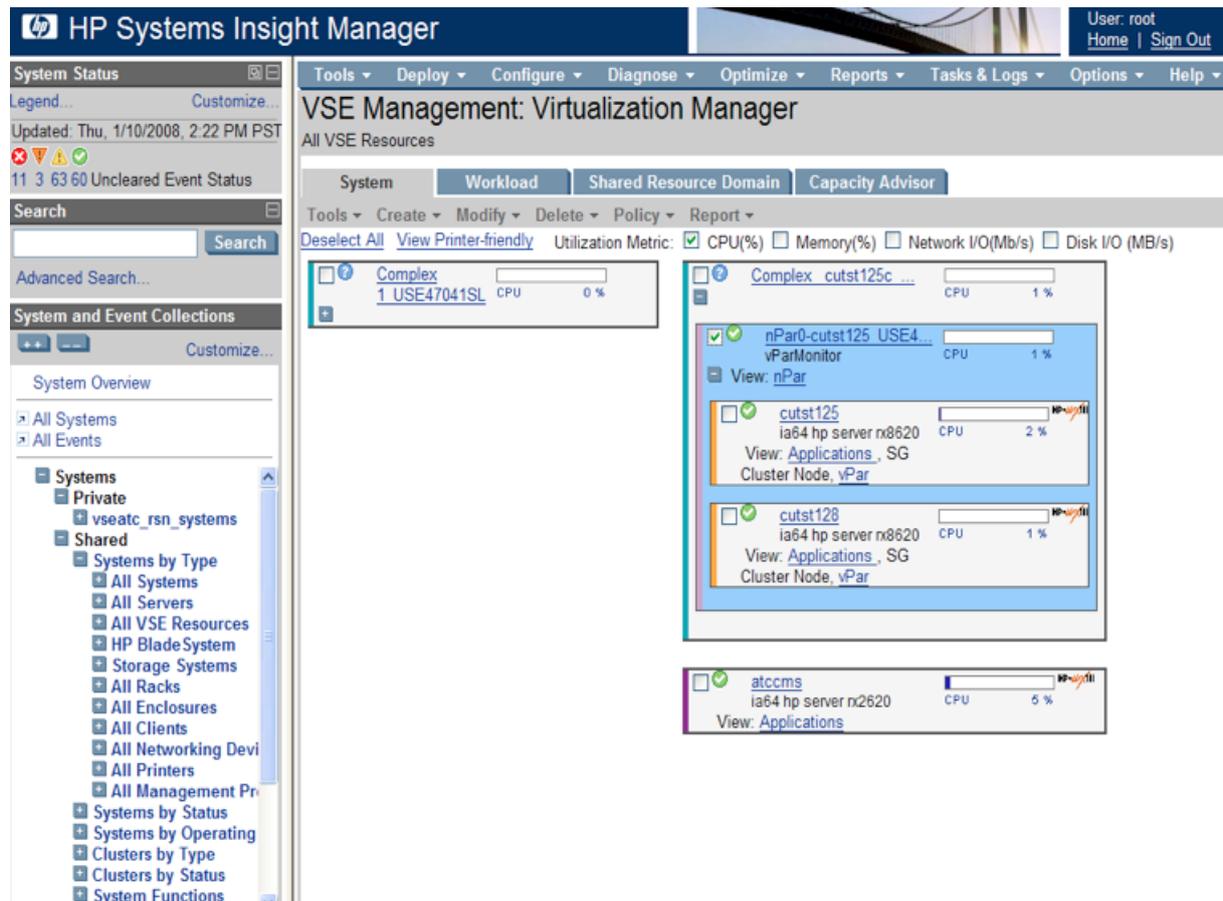


Figura 3.3: HP Virtualization Manager.

O gWLM permite a definição das políticas de compartilhamento de recursos, podendo ser aplicadas em múltiplos servidores. Sua aplicação pode otimizar a utilização do sistema e facilitar o compartilhamento controlado de seus recursos. Além disso, o gWLM proporciona o monitoramento em tempo real dos sistemas e os respectivos históricos de alocação de seus recursos computacionais. Os principais benefícios da utilização do gWLM são mostrados a seguir:

- **Melhor utilização da capacidade dos servidores existentes** - O gWLM permite a combinação de vários *workloads* com diferentes padrões de demanda em apenas um servidor, realizando o uso da capacidade ociosa do servidor.
- **Garantia de que *workloads* críticos terão sempre os recursos necessários** - Mesmo com vários *workloads* em um servidor, é possível garantir que cargas de trabalho de missão crítica obtenham os recursos necessários. O gWLM é capaz de ajustar automaticamente a alocação de recursos de acordo com as demandas

necessárias.

- **Redução de custos com a administração do sistema** - Com o gWLM é possível combinar mais *workloads* em menos servidores, reduzindo assim os custos com a administração.

A Figura 3.4 ilustra um *screenshot* do HP *Global Workload Manager* (gWLM) sendo acessado através do HP SIM.

The screenshot displays the HP Global Workload Manager interface. The main window title is "VSE Management: Global Workload Manager". The interface includes a navigation bar with tabs for "System", "Workload", "Shared Resource Domain", and "Capacity Advisor". Below the navigation bar, there are options to "Show" "Deployed SRDs" (selected) or "Undeployed SRDs", and a "Refresh Data" link. A table lists SRDs with columns for SRD Name, Status, Mode, State, Size, CPU Utilization, and Last Update. One SRD, "cutst12X.vpar.srd", is shown with a status of "Deployed" and 51.2% CPU utilization. Below the table, there are "Details for SRD: cutst12X.vpar.srd" including Shared Resource Domain, Mode, State, and CPU Utilization. A sub-table shows "Workload" details with columns for Workload, CPU Utilization, Type, Policy (Active), Status, Request, Allocation, Size, and Hostname. Two workloads are listed: "cutst125.cup.hp.com" and "cutst128.cup.hp.com". At the bottom, there are status messages: "Unknown", "Critical", "Major", and "Minor".

Figura 3.4: HP *Global Workload Manager* (gWLM).

3.1.3.3 HP Capacity Advisor: O *Capacity Advisor* [cap09, Her06] é um componente do *VSE Management Software* que atua como um software de análise e planejamento de capacidade, permitindo a otimização de *workloads* no VSE para obter uma utilização máxima dos recursos do servidor. Esta ferramenta fornece uma análise de

cenários que representam o comportamento das cargas de trabalho pertencentes aos sistemas gerenciados pelo VSE, e proporciona o planejamento da infraestrutura. Desta forma, o *Capacity Advisor* permite a otimização dos recursos atuais dos servidores, o planejamento de futuras expansões nos *workloads* e a unificação de servidores, onde podem ser realizados testes que simulam alterações nas configurações do ambiente, sem que seja necessário fazê-las fisicamente. Desta maneira, é possível observar o comportamento do sistema em várias situações, e assim definir a configuração mais adequada para o melhor aproveitamento dos recursos computacionais da infraestrutura [Her06].

Os dados trabalhados pelo *Capacity Advisor* são compostos pela avaliação dos seguintes recursos:

- ***CPU cores (number) and speed*** - Responsável pela avaliação da capacidade de processamento de cada sistema pertencente a infraestrutura, focando no número de processadores e suas respectivas velocidades de processamento;
- ***Memory*** - Responsável pela avaliação da utilização de memória RAM em cada sistema, medida em GB - *gigabytes*;
- ***Network I/O bandwidth*** - Responsável pela avaliação da largura de banda utilizada pela rede de cada sistema, medida em Mb/s (*megabits* por segundo);
- ***Disk I/O bandwidth*** - Responsável pela avaliação da largura de banda utilizada pelo disco rígido em cada sistema, medida em MB/s (*megabytes* por segundo).

Durante o processo de planejamento de capacidade, o *Capacity Advisor* realiza periodicamente a coleta de dados referentes a utilização dos recursos computacionais de cada sistema, e permite a visualização dos valores através de gráficos representando os seguinte comportamentos: histórico de utilização, picos dos valores (*data peaks*), duração dos picos e previsão de comportamentos futuros dos dados (*forecast trends*). Adicionalmente, a ferramenta permite a criação de cenários para o planejamento e previsões através de situações hipotéticas (*what if scenarios*), executando as seguintes tarefas:

- **Para os sistemas pertencentes aos cenários:**
 - Criação de novos sistemas;
 - Remoção de sistemas;
 - Edição de atributos dos sistemas;
 - Adição de novas máquinas virtuais.

- Para *workloads* pertencentes aos cenários:
 - Criação de novos *workloads*;
 - Coleta dos dados de utilização dos *workloads*;
 - Importação de perfis de *workloads* existentes;
 - Edição das demandas dos *workloads*;
 - Migração de *workloads* entre os sistemas;
 - Exclusão de *workloads*.
- Para os próprios cenários:
 - Visualizar e desfazer/refazer mudanças nos cenários;
 - Criação de relatórios de comparação entre os cenários.

A Figura 3.5 ilustra um *screenshot* do *Capacity Advisor* em execução, onde pode ser visualizado o gráfico de utilização de CPU pertencente a um dos sistemas dentro do VSE durante o período de aproximadamente um mês.

Mais detalhes sobre o *Capacity Advisor* são mostrados na próxima seção, onde são destacados seus principais componentes, seu funcionamento e a estrutura de sua arquitetura.

3.1.3.4 HP Integrity Virtual Machines Manager (VM Manager): O *VM Manager* é uma aplicação pertencente ao *VSE Management Software* responsável por gerenciar as máquinas virtuais (*HP Integrity Virtual Machines*) instaladas no ambiente. Esta ferramenta é acessível via *browser* através do HP SIM, e funciona como um *software* de particionamento e virtualização utilizado para a criação, configuração, monitoramento e avaliação de máquinas virtuais dentro de servidores físicos (*HP Integrity Servers*) ou *nPartitions*, chamados de *VM Host*.

As máquinas virtuais instaladas podem compartilhar um único conjunto físico de recursos de *hardware*, no entanto cada máquina virtual simula um ambiente completo, funcionando como um sistema real e executando a sua própria instância de sistema operacional (*guest OS*). Como uma máquina real, uma máquina virtual pode possuir um ou mais processadores (*virtual CPUs* ou *vCPUs*), memória, disco, dispositivos de rede, etc. Todos esses elementos são virtuais, o que significa que eles são parcialmente emulados em

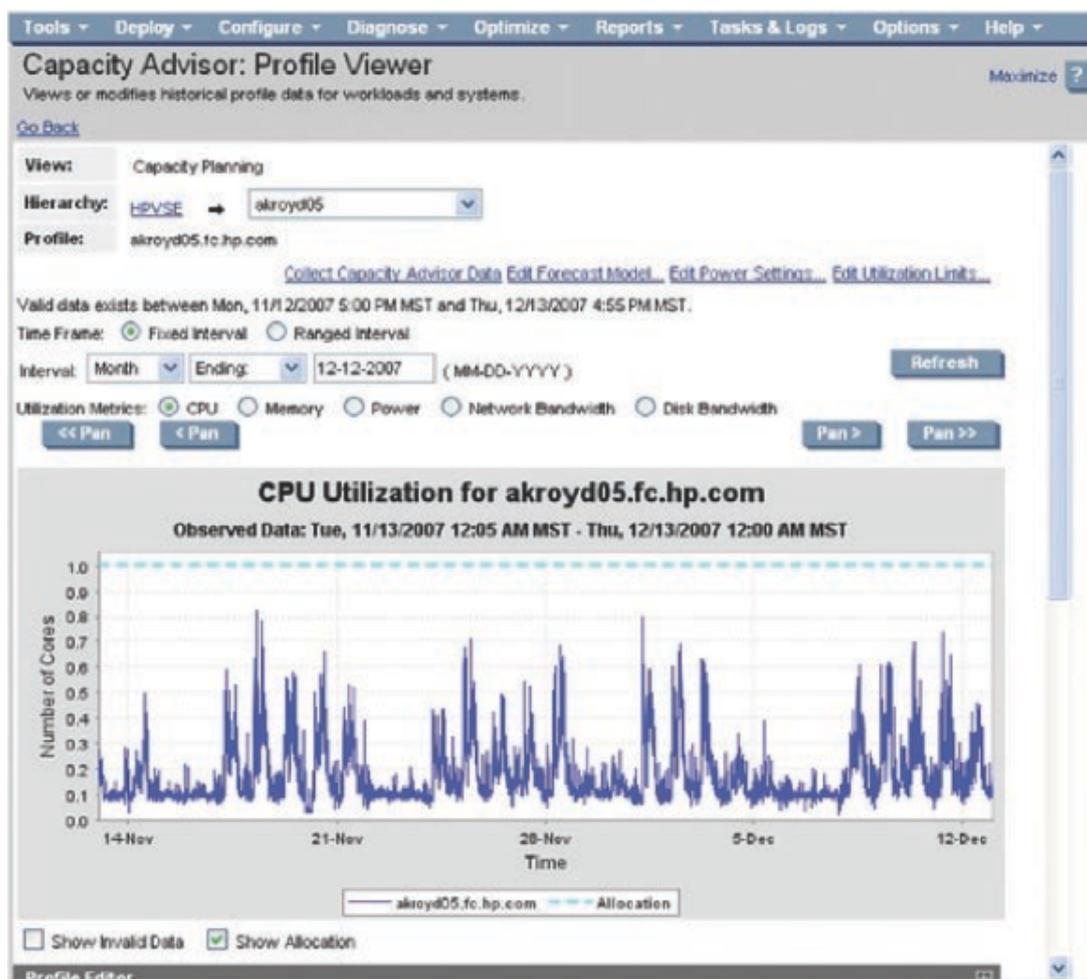


Figura 3.5: HP *Capacity Advisor*.

softwares e não implementados totalmente através de *hardwares*, porém, para o sistema operacional instalado eles aparecem como componentes físicos reais.

A Figura 3.6 ilustra um *screenshot* do HP *Integrity Virtual Machines Manager* (VM *Manager*) em execução, onde são listadas todas as máquinas virtuais instaladas na infraestrutura, seus sistemas operacionais e recursos computacionais.

3.1.3.5 HP *Application Discovery*: O HP *Application Discovery* é uma aplicação pertencente ao VSE *Management Software* responsável por localizar e exibir as aplicações instaladas e processos em execução pertencentes a rede onde o VSE estiver instalado. Com ele é possível consultar os dados das aplicações a partir da localização do servidor, do tipo de aplicação e do seu estado. Os dados coletados pelo *Application Discovery* podem

VM Name ↑ (Hostname)	HW	OS	Operating System	VM vCPU Utilization	Memory Utilization	Disk I/O
bitsv00 (10.1.55.7)	✓	✓	HP-UX 11i HP-UX 8.11.23	0 %	47 % of 4GB	0 MB/s
bitsv01 (?)	✓	✓	HP-UX 11i HP-UX	No Data	No Data	No Data
vm001 (?)	✓	✓	HP-UX 11i HP-UX	No Data	No Data	No Data
vman01 (?)	✓	✓	HP-UX 11i HP-UX	No Data	No Data	No Data
vman01 (?)	✓	✓	HP-UX 11i HP-UX	No Data	No Data	No Data

Figura 3.6: HP Integrity Virtual Machines Manager.

ser vistos e utilizados pelo *Virtualization Manager*, com o intuito de se acrescentar novas aplicações aos *workloads*.

Com o *Application Discovery* é possível realizar as seguintes tarefas:

- Descobrir a localização e o número de aplicações reconhecidas, instaladas e em execução pertencentes a rede ou ao *data center*.
- Monitorar as atividades e utilização de recursos dos processos e aplicações em execução;
- Selecionar os tipos de eventos que deverão enviar notificações e definir o nível de importância de cada evento;

A Figura 3.7 ilustra um *screenshot* do HP *Application Discovery* em execução, onde é mostrado um resumo com o número de aplicações instaladas, aplicações em execução, total de processos, processos do sistema, e outras informações pertencentes a três *hosts* instalados no VSE.

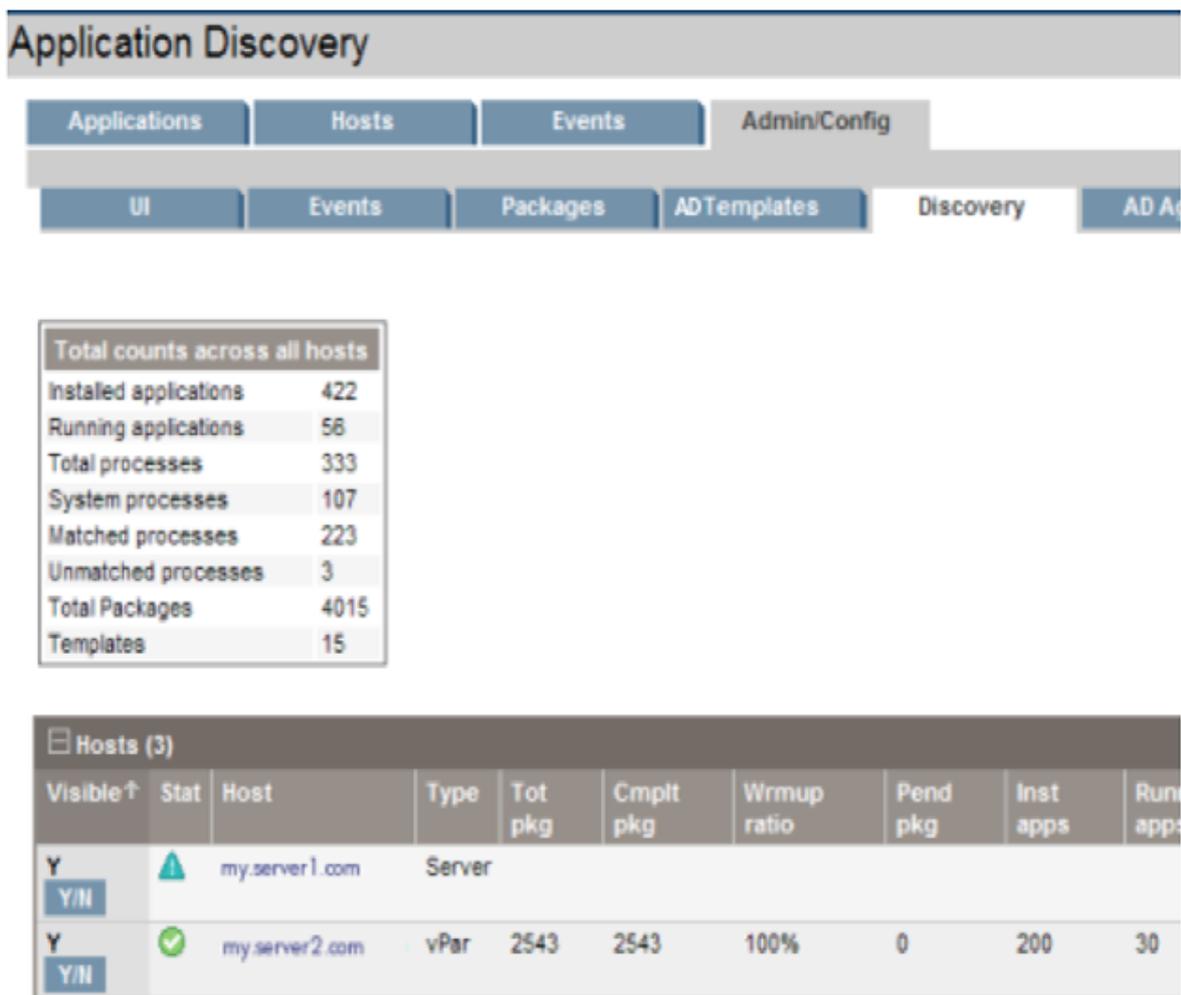


Figura 3.7: HP *Application Discovery*.

3.2 HP CAPACITY ADVISOR

Como foi visto na seção anterior, o *Capacity Advisor* é um componente do VSE *Management Software* responsável pela análise e planejamento de capacidade dos sistemas pertencentes ao VSE, visando a otimização de seus *workloads* para um maior aproveitamento dos recursos computacionais pertencentes a infraestrutura (CPU, memória, disco e rede) [Her06, cap09]. Esta seção tem como objetivo detalhar um pouco mais o *Capacity*

Advisor, destacando seus principais componentes, seu funcionamento, sua arquitetura e sua infraestrutura de coleta de dados, para que desta forma sejam definidos os principais requisitos que deverão ser utilizados durante a criação de um gerador de cargas para o *Capacity Advisor*, objetivo principal deste trabalho.

3.2.1 Visão Geral

O HP *Capacity Advisor* permite que os *capacity planners* (analistas responsáveis por realizar o planejamento de capacidade) possam analisar os dados do histórico de utilização de *workloads* (cargas de trabalho) com o intuito de realizar o planejamento de migrações ou introduções de novos *workloads* nos sistemas gerenciados. Desta forma, os passos que são tradicionalmente realizados manualmente pelos analistas são automatizados, liberando assim os profissionais para se focarem apenas no planejamento de migrações, introduções e consolidações de novos *workloads*.

O *Capacity Advisor* realiza a coleta dos dados de utilização de *workloads* para cada *workload* existente na infraestrutura de servidores virtuais do VSE. Estes dados são coletados diariamente em cada sistema e são armazenados no HP *Systems Insight Manager CMS* (*Central Management Server*). O conjunto de dados referentes a utilização dos *workloads* coletados são chamados de “*workload’s historic profile*”.

Workload’s historic profile : O *Workload’s historic profile* nada mais é do que o histórico de consumo dos recursos de cada *workload* presente no VSE . Estas informações consistem de históricos de utilização de CPU, memória, disco I/O e rede I/O, onde são disponibilizados relatórios e gráficos referentes a cada uma dessas métricas. Além de gerar relatórios e gráficos, o *historic profile* pode ser editado através da utilização do “*Historic Profile Editor*”.

O *historic profile editor* pode ser utilizado para invalidar certos pontos de dados no *workload’s historic profile*. Alguns eventos atípicos podem ocorrer em ambientes de computação, alterando as exigências de recurso reais para um determinado *workload*. Por exemplo, uma falha intermitente de *hardware* pode fazer com que uma aplicação requisite novas tentativas de utilização desnecessárias, gravando métricas de utilização anormalmente elevadas. Para eventos como estes, o *historic profile editor* do *Capacity Advisor* pode ser utilizado para invalidar os raros pontos de dados elevados. Os pontos de dados não desejados podem ser invalidados de acordo com um determinado intervalo de datas (dias e meses) ou pode ocorrer um descarte

de valores acima de certo limite.

A Figura 3.8 ilustra um *screenshot* do *historic profile editor* em execução, onde é mostrado o gráfico de utilização de CPU referente ao *workload* *Web* durante o período de uma semana. Observe que na opção *Time Intervals* é disponibilizada a opção de invalidar um determinado período pertencente ao gráfico, caso seja necessário.

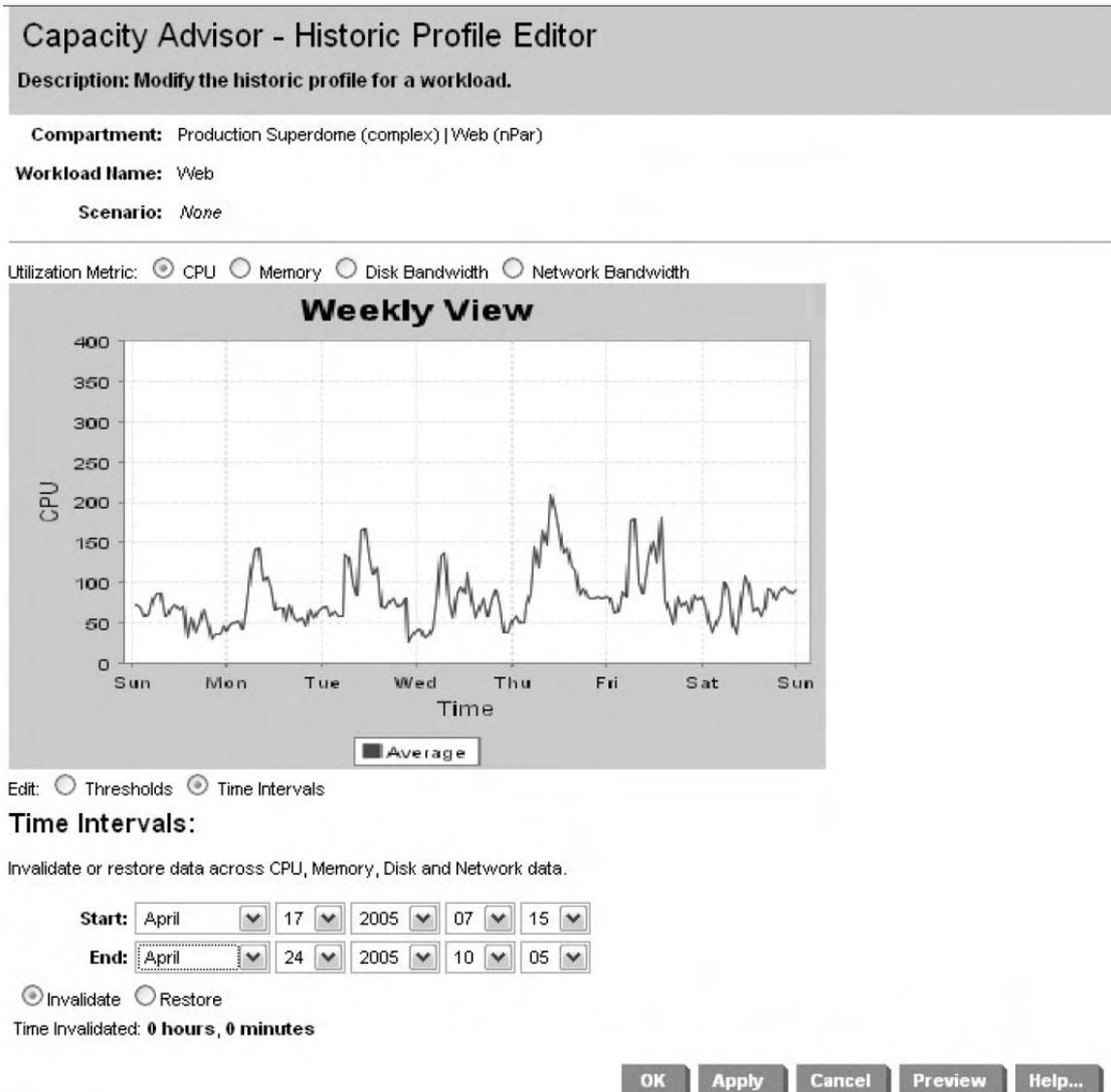


Figura 3.8: *Historic profile editor*.

Workload's forecast (Previsão de Workloads) : Depois de assegurar que o *historic profile* reflete com precisão a natureza da carga de trabalho, o próximo passo

é a edição do *workload's forecast*, onde devem ser definidas as características para possibilitar previsões em comportamentos futuros das cargas de trabalho. A previsão funciona como uma extrapolação da base de dados no futuro, permitindo que o crescimento das tendências seja introduzido, ajustando assim os aumentos esperados dos recursos necessários em cada *workload*. Em situações em que a alteração dos recursos necessários para um *workload* é esperada, o *forecast editor* permite que a previsão de um *workload* seja modificada. Por exemplo, se um *workload* possui uma expectativa de crescimento de 10% ao longo dos próximos doze meses, o *forecast editor* pode ser utilizado para alterar o comportamento do *workload* de acordo com essa expectativa.

A Figura 3.9 ilustra um *screenshot* do *workload's forecast editor* em execução, onde também é mostrado o gráfico de utilização de CPU referente ao *workload Web* durante o período de uma semana. Além do gráfico, pode ser visualizado o percentual referente a taxa de crescimento anual observado para cada uma das métricas (*Observed Annual Growth Rate*). Observe que este percentual pode ser redefinido de acordo com as expectativas de crescimento escolhidas.

Workload's baseline : Como parte da edição do *workload's forecast*, o *workload's baseline* também pode ser alterado. Uma *baseline* (linha de base) representa um modelo da utilização dos recursos de um *workload* baseado em seu *workload's historic profile*. O *workload's baseline* nada mais é do que uma representação de performances anteriores de *workloads* de um sistema para definir um comportamento futuro. Uma maneira de criar uma *baseline* é através da seleção de um dia específico no histórico de *workloads* como uma amostra representativa para ser utilizada com o objetivo de planejamento de capacidade. *Workloads* que têm padrões de consumo diário repetido, como sistemas de *backup*, são candidatos a utilizar este tipo de *baseline*. Outra forma é a criação de uma *baseline* a partir de um período arbitrário dos dados do histórico de *workloads*. *Workloads* que trabalham numa base semanal são melhores modelados por este tipo de *baseline*. Desta forma, são disponibilizadas várias opções para a criação de *baselines*. É importante não esquecer de que as *baselines* devem ser criadas utilizando o conjunto de dados adequado para cada *workload*. Somente *baselines* que refletem com exatidão a utilização do modelo do *workload* são úteis para o objetivo de planejamento de capacidade.

A edição de uma *baseline* é realizada através da utilização do *Capacity Advisor baseline editor*. Caso nenhuma alteração tenha sido feita, o *Capacity Advisor* deriva

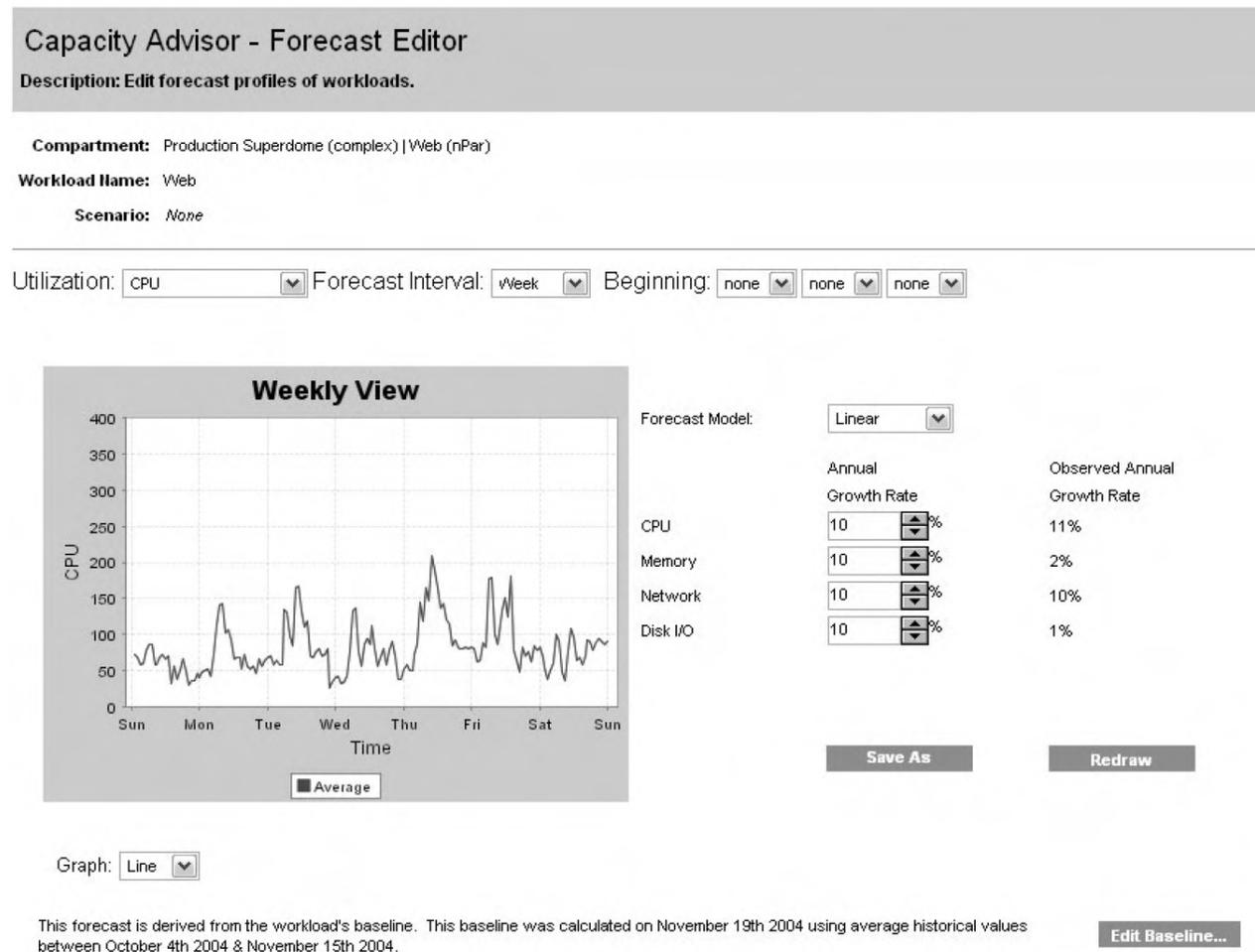
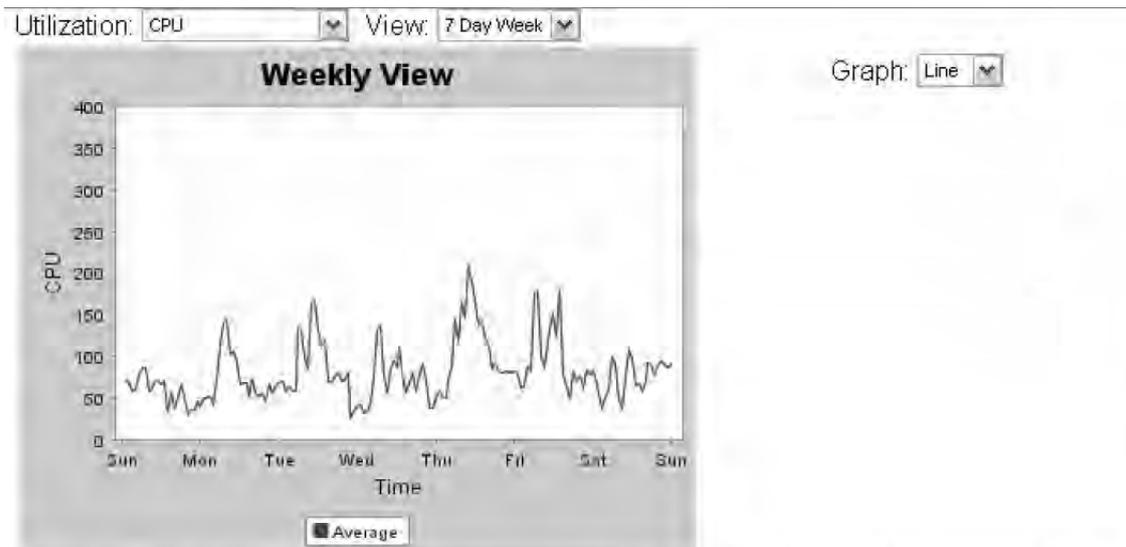


Figura 3.9: *Workload's forecast editor.*

a *baseline* diretamente da base de dados de informações históricas do *workload's historic profile*. Em alguns casos, a *baseline* não exige nenhuma modificação, no entanto, na maior parte do tempo os *workloads* são dirigidos a eventos ou são de natureza cíclica e o ajuste das *baselines* resulta em um modelo muito mais exato e confiável para o objetivo de planejamento da capacidade.

A Figura 3.10 ilustra um *screenshot* do *Capacity Advisor baseline editor* em execução, onde é mostrado o gráfico de utilização de CPU durante o período de uma dia. As opções para seleção das *baselines* são ilustradas na parte inferior da Figura 3.10, onde podem ser selecionados os diversos períodos que servirão como base para o comportamento futuro dos *workloads*.

Capacity Advisor scenario : Uma vez personalizados o *workload's historic profile* e os *forecasts* (previsões), o próximo passo é criação de um cenário através do *Capacity*



This baseline was calculated on November 19th 2004 using average historical values between October 4th 2004 & November 15th 2004.

Change Baseline <Historical Data Range e.g., 1/1/2002 - 11/10/2004>

Baseline is a "typical" representation of past system performance used for predicting future behavior. Choose one of the following *Baseline* types that best meet your future performance needs.

Historic Averages Edit
 Past Days Ending (mm/dd/yyyy or yesterday)
 Single Day: (mm/dd/yyyy)
 Estimated Day - Simple
 CPU: GHz CPUs Memory: MB Network: GB Disk: MB
 Estimated Day - Advanced Edit Customize values over a 24 hour period.

Figura 3.10: Capacity Advisor baseline editor.

Advisor scenario, permitindo que vários *workloads* sejam avaliados como um grupo. Um cenário é um conjunto de sistemas e *workloads* combinados entre si com o objetivo de realizar planejamento de capacidade e previsões. Através da utilização de previsões de *workloads* (*workload forecasts*), cenários hipotéticos (*what-if scenarios*) podem ser avaliados para determinar se migrações, consolidações, ou introduções de *workloads* serão bem sucedidas. Sistemas hipotéticos e *workloads* também podem ser definidos dentro de um cenário para proporcionar o planejamento de novos recursos. Os sistemas hipotéticos podem ser utilizados para ajudar a responder

perguntas como: “Será que estes *workloads* irão ter um desempenho aceitável neste novo sistema que irei comprar?”. Além disso, cenários podem ser utilizados para avaliar os efeitos da movimentação de um *workload* a partir de um sistema para outro. Essencialmente, os cenários de planejamento de capacidade fornecem banco de testes para a realização de análises utilizando os sistemas reais e hipotéticos ao mesmo tempo e vários *workloads*. Estes cenários são confiáveis pois se baseiam em dados do históricos de utilização dos *workloads*. Os dados históricos são então utilizados para criar previsões e *baselines* para os *workloads*. Desta forma o *Capacity Advisor* permite que *capacity planners* planejem mudanças em um *datacenter* de forma eficaz e precisa.

A Figura 3.11 ilustra um *screenshot* da criação de um novo cenário para o *Capacity Advisor* representando um sistema hipotético (“*What-if*” *system*), onde são definidos seus principais componentes, tais como: número e velocidade dos CPUs, quantidade de memória, limite da largura de banda da rede em Mb/s, limite da largura de banda do disco rígido em MB/s e o Sistema Operacional utilizado. Após a criação do sistema, poderão ser introduzidos *workloads* para a simulação e avaliação de seu comportamento com base no cenário gerado.

3.2.2 Arquitetura

A Figura 3.12 ilustra os principais componentes pertencentes ao *Capacity Advisor* e a arquitetura do seu funcionamento [cap09].

O acesso ao *Capacity Advisor* é realizado através de um navegador *web* (*browser*), podendo ser executado por qualquer computador que esteja configurado para ter acesso ao CMS (*Central Management Server*), sistema no domínio de gerenciamento onde é executado o HP SIM (*Systems Insight Manager*) e o VSE *Management Software*. Como já foi observado, todas as operações centrais no HP SIM são iniciadas a partir deste sistema. A interface *web* é fornecida pelo HP SIM, que também permite o acesso ao conjunto de aplicações pertencentes ao VSE *Management Software*. Como o *Capacity Advisor* é um componente do VSE *Management Software*, seu funcionamento dependerá da correta instalação e configuração do HP SIM e do VSE no CMS.

Conforme pode ser observado na Figura 3.12, o *Capacity Advisor* possui duas bases de dados onde são armazenados os valores referentes ao consumo dos recursos computacionais dos sistemas monitorados: uma base de dados referente ao histórico de utilização dos

Create "What-If" System

Create a new system that only exists in the scope of this scenario

Compartment: None

Specify the system name*:
Note: system name should be the fully qualified host name of the system..

Specify a description of the system:
Note: this can be the same as the System Name or it can be any other descriptive text used to provide more information about the system.

Specify the number of CPUs*: Specify CPU speed in GHz*: IA

Specify system memory in GBs*:

Specify network I/O limit in Mbs*:

Specify disk I/O limit in Mbs*:

Specify an OS type: Include typical OS workloads estimates

* required field

Figura 3.11: Capacity Advisor scenario.

recursos computacionais dos sistemas em um determinado período - *Historical Data*; e uma base de dados onde são armazenados os valores gerados durante a criação e execução dos cenários no *Capacity Advisor - Scenario Data*.

Para o armazenamento dos valores pertencentes ao histórico de utilização dos recursos na base de dados do *Capacity Advisor*, podem ser utilizados diversos mecanismos para a coleta de dados. Estes mecanismos estão divididos em: *mecanismos recomendados para a coleta de dados* e *mecanismos alternativos para a coleta de dados* [cap09] (ver Figura 3.12).

Mecanismos recomendados para a coleta de dados: os mecanismos recomendados são realizados por agentes instalados nos sistemas, responsáveis por realizar a coleta dos dados referente ao histórico de utilização dos recursos computacionais. Nos sistemas gerenciados pelo VSE (*Windows* e *HP-UX*) é utilizado o agente "*Utilization Provider*" [uti09], já em sistemas configurados com o *VMware ESX* [vmw09] é utilizado o agente *VMM (Virtual Machine Manager)*. Esses agentes permitem que *traces* referentes aos dados dos históricos de utilização de CPU, memória, disco e rede sejam coletados

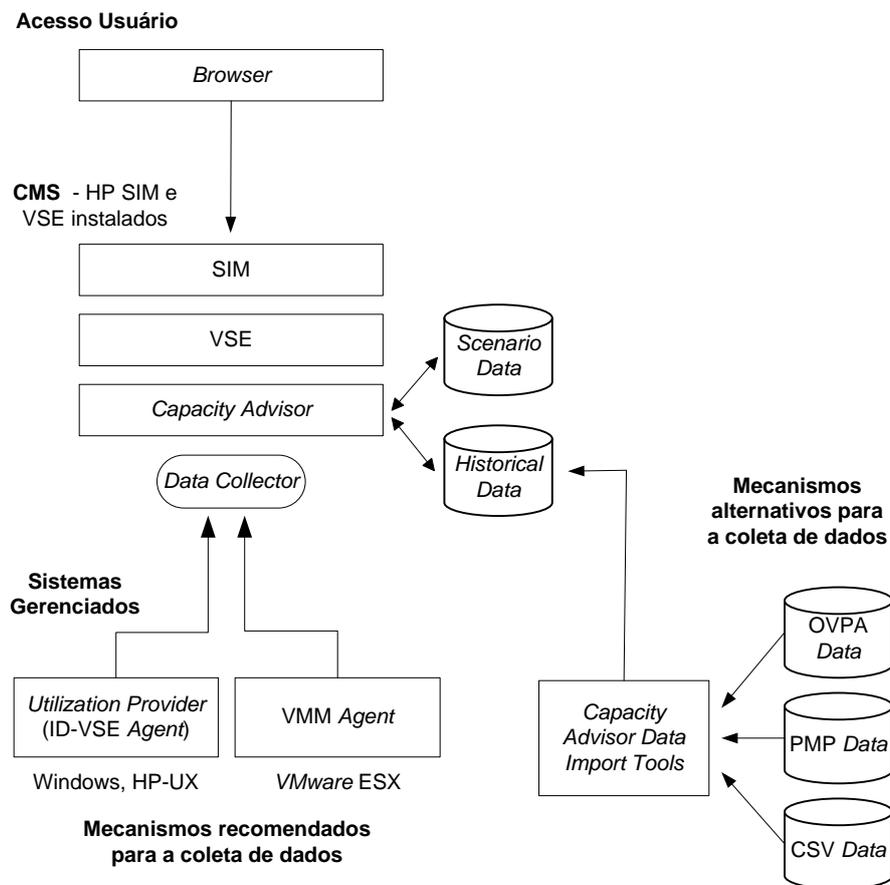


Figura 3.12: Arquitetura do *Capacity Advisor*.

em intervalos de 5 minutos, para posteriormente serem acessados pelo “*Capacity Advisor Data Collector*”, funcionalidade responsável por armazenar os valores na base de dados *Historical Data*.

Mecanismos alternativos para a coleta de dados: além dos mecanismos recomendados, são disponibilizados alguns mecanismos alternativos de coleta de dados para o *Capacity Advisor*. Os dados referentes ao histórico de utilização dos recursos computacionais podem ser introduzidos diretamente na base de dados do *Capacity Advisor* (*Historical Data*) através da utilização dos dados OVPA (OVPA data), coletados a partir do HP *OpenView Performance Agent* - OVPA [ovp09], dados PMP, coletados a partir do HP *Performance Management Pack* - PMP [pmp09] e arquivos CSV - *Comma Separated Values* [Sha05], arquivos em formato CSV exportados pelo *Capacity Advisor*. Estes valores são armazenados através do *Capacity Advisor Data import Tools*, responsável por extrair e armazenar os valores na base de dados *Historical Data*.

3.2.3 Formato dos *traces*

Conforme observado na seção anterior, o *Capacity Advisor* permite que os dados coletados nos sistemas monitorados, pertencentes a sua base de dados, sejam exportados em arquivos estruturados com o formato CSV (*Comma Separated Values*) para que possam ser utilizado posteriormente em qualquer ambiente que possua o *Capacity Advisor* instalado. Os arquivos gerados são compostos por uma série de linhas compostas pelas seguintes informações:

Cabeçalho (*profile header*) - Contendo as datas em que as amostras foram coletadas e os identificadores das métricas representando o consumo dos recursos computacionais utilizados;

Valores (*values*) - Após o cabeçalho, cada linha do arquivo representa uma amostra contendo os valores referente ao consumo dos recursos computacionais (processador, memória, disco e rede). Estes valores são incrementados em intervalos de tempo fixos de cinco minutos.

As sete primeiras linhas do arquivo correspondem ao cabeçalho (*header*) do *trace*, onde são definidas as principais propriedades do sistema. Estas informações são utilizadas para identificar as características de cada sistema monitorado quando o arquivo é inserido na base de dados do *Capacity Advisor*. A descrição de cada informação definida no cabeçalho é mostrada a seguir [cap09]:

#Profile: *name* - Nome definido para identificar o *Profile* (Identificação) do *trace* gerado;

#Host: *hostname* - Nome definido para identificar o sistema (*hostname*) simulado;

#CPU: *CPU_count@CPU_speedGHz* - Número de processadores (*CPU_count*) do sistema e as suas respectivas frequências de operação em *gigahertz* (*CPU_speedGHz*);

#Memory: *MEM_sizeMB* - Quantidade de memória RAM pertencente ao sistema em *Megabytes* (*MEM_sizeMB*);

#OS: *platform* - Sistema Operacional (OS) da plataforma a ser simulada;

#Model: *model* - Modelo (*model*) do *hardware* em que o sistema está configurado;

[YYYYMMDDhhmm, UTIS, métrica [, métrica, ...]].

A última linha do cabeçalho contém a lista com os identificadores referentes às datas e métricas coletadas, separados por vírgula. Os campos desta linha do cabeçalho são descritos a seguir:

YYYYMMDDhhmm - Identificador das datas (*Timestamp*), opcional, que representa o tempo de coleta referente a cada amostra, onde YYYY representa o ano, MM representa o mês (de 01 a 12), DD representa o dia (de 01 a 31), hh representa as horas (de 00 a 23) e mm representa os minutos (de 00 a 59);

UTIS - Universal Time (GMT) In Seconds - Padrão *UNIX timestamp* medido em segundos (desde 01 de Janeiro de 1970).

métrica - Seleção de pelo menos uma das métricas descritas na Tabela 3.2.

Tabela 3.2: Métricas utilizadas pelo *Capacity Advisor*.

Métricas	Descrição
CPU_ALLOC	Quantidade de processadores (CPUs) alocados
CPU_UTIL	Utilização de CPU representada pelo número de CPUs utilizados
DISK_UTIL	Utilização da largura de banda de disco em MB/s
MEM_ALLOC	Quantidade de memória alocada em GB
MEM_UTIL	Utilização de memória em GB
NET_UTIL	Utilização da largura de banda de rede em Mb/s
PHYS_CPUS	Número de CPUs físicos
PHYS_MEM	Quantidade de memória física disponível em GB

Observe que a última linha do cabeçalho do arquivo deve incluir pelo menos uma das métricas disponíveis, onde pode ocorrer qualquer combinação entre as oito métricas existentes, sem importar a ordem. O padrão *UNIX timestamp* também deverá estar presente no *trace* obrigatoriamente, para representar a data em que cada amostra foi coletada. No entanto, devido à sua complexidade, é disponibilizado um campo opcional para fornecer uma melhor compreensão e legibilidade durante a identificação das datas, seguindo o seguinte formato **YYYYMMDDhhmm**.

Após o cabeçalho, cada linha subsequente contém os valores referentes ao consumo dos respectivos recursos computacionais coletados em um determinado período de tempo,

sendo incrementados em intervalos de cinco minutos (300 segundos). A Figura 3.13 ilustra um trecho de um arquivo CSV gerado pelo *Capacity Advisor* contendo alguns valores pertencentes a um *traces* coletado no sistema “*legacy13*”.

```
#Profile: legacy13
#Host: legacy13
#CPU: 2 @ 2.7GHz
#Memory: 2 GB
#OS: WINNT
#Model: ProLiant DL140 G3
YYYYMMDDhhmm, UTIS, CPU_UTIL, MEM_UTIL, NET_UTIL, DISK_UTIL, CPU_ALLOC, MEM_ALLOC, PHYS_CPUS, PHYS_MEM
200708311800,1188604800, 0.62380, 1.53180, 1.24000, 31.28125,2, 2.000,2, 2.000
200708311805,1188605100, 0.63320, 1.53180, 1.79000, 34.75000,2, 2.000,2, 2.000
200708311810,1188605400, 0.69780, 1.53240, 1.13000, 39.03125,2, 2.000,2, 2.000
200708311815,1188605700, 0.71680, 1.53260, 1.80000, 34.75000,2, 2.000,2, 2.000
200708311820,1188606000, 0.76500, 1.53280, 1.08000, 42.68750,2, 2.000,2, 2.000
200708311825,1188606300, 0.77640, 1.53320, 1.31000, 42.50000,2, 2.000,2, 2.000
200708311830,1188606600, 0.77820, 1.53320, 1.07000, 39.09375,2, 2.000,2, 2.000
200708311835,1188606900, 0.71820, 1.53320, 1.78000, 20.87500,2, 2.000,2, 2.000
200708311840,1188607200, 0.65120, 1.53400, 1.25000, 29.26562,2, 2.000,2, 2.000
200708311845,1188607500, 0.68600, 1.53720, 1.47000, 37.12500,2, 2.000,2, 2.000
200708311850,1188607800, 0.77900, 1.53860, 1.10000, 45.82812,2, 2.000,2, 2.000
200708311855,1188608100, 0.80640, 1.53220, 1.35000, 50.45312,2, 2.000,2, 2.000
200708311900,1188608400, 0.84800, 1.53180, 1.26000, 51.56250,2, 2.000,2, 2.000
200708311905,1188608700, 0.81720, 1.53360, 1.73000, 39.35938,2, 2.000,2, 2.000
200708311910,1188609000, 0.80240, 1.53500, 1.37000, 41.87500,2, 2.000,2, 2.000
200708311915,1188609300, 0.83000, 1.53680, 2.07000, 47.25000,2, 2.000,2, 2.000
200708311920,1188609600, 0.67580, 1.53900, 1.36000, 54.51562,2, 2.000,2, 2.000
200708311925,1188609900, 0.62360, 1.49720, 1.12000, 54.54688,2, 2.000,2, 2.000
200708311930,1188610200, 0.60680, 1.51400, 1.61000, 37.56250,2, 2.000,2, 2.000
```

Figura 3.13: Arquivo CSV gerado pelo *Capacity Advisor*.

3.3 SUMÁRIO

Este capítulo apresentou os principais conceitos referentes ao ambiente onde o gerador de cargas desenvolvido deve atuar. Primeiramente foi introduzido o ambiente de servidores virtuais HP VSE (*Virtual Server Environment*), onde foram destacadas suas principais características e funcionamento, além de sua composição, formada pelo *VSE Management Software*, conjunto de aplicações responsáveis pelo gerenciamento de todas as funcionalidades fornecidas pelo VSE. Posteriormente, foi apresentado o funcionamento, estrutura e arquitetura de um dos principais componentes do *VSE Management Software*, o *HP Capacity Advisor*, ferramenta responsável pela realização do planejamento de capacidade do ambiente.

CAPÍTULO 4

METODOLOGIA PARA A CONCEPÇÃO DO GERADOR DE CARGA

Este capítulo apresenta a metodologia utilizada para a concepção do gerador de carga para o *HP Capacity Advisor* [cap09]. Inicialmente é mostrada uma visão geral sobre o objetivo do trabalho, ilustrando brevemente as características e funcionamento do gerador de carga desenvolvido. Em seguida é apresentada a metodologia adotada para o seu desenvolvimento, composta por diversas etapas importantes, que envolvem o entendimento do ambiente, o estudo de metodologias de geração de carga existentes, a definição dos cenários de geração de carga, o desenvolvimento do gerador e a avaliação da carga gerada. Por fim, são descritos os principais mecanismos utilizados durante o desenvolvimento do ferramental e sua validação.

4.1 VISÃO GERAL

Como foi visto no capítulo introdutório, o principal objetivo desta pesquisa é o desenvolvimento de um gerador de carga, cuja função é gerar *traces* sintéticos que simulem o consumo dos principais recursos computacionais, tais como processador, memória, disco e rede, e que posteriormente sejam introduzidos no *HP Capacity Advisor*. Os *traces* sintéticos deverão ser gerados baseados em *traces* reais ou em um conjunto significativo de distribuições de probabilidade e/ou resumos estatísticos, e posteriormente armazenados em arquivos com o formato CSV - *Comma Separated Values* [Sha05], seguindo a mesma estrutura dos arquivos que são exportados atualmente pelo *Capacity Advisor* (conforme descrito no Capítulo 3), desta forma os *traces* sintéticos poderão ser importados pela ferramenta.

A Figura 4.1 ilustra a arquitetura do *Capacity Advisor*, descrita no Capítulo 3 (Seção 3.2.2), e a proposta de funcionamento do gerador de carga desenvolvido, o *WGCap - Workload Generator for Capacity Advisor*.

Como pode ser observado, o gerador de carga trabalha com dois tipos de entrada

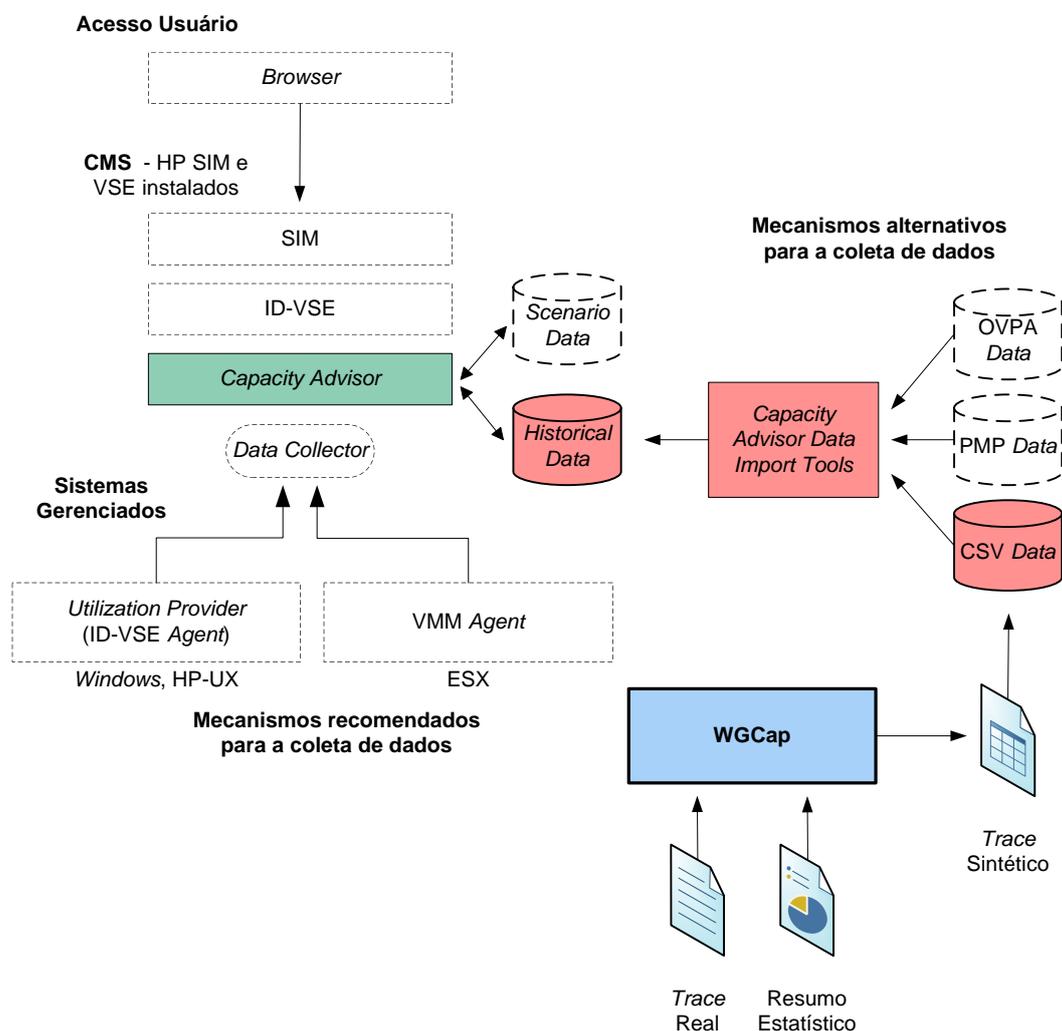


Figura 4.1: Arquitetura de funcionamento do gerador de *traces* para o *Capacity Advisor*.

para a geração dos *traces* sintéticos (ver Figura 4.1): *Traces* reais - *Real trace*, valores referentes ao consumo de cada recurso computacional coletado a partir de sistemas reais, ou Resumos estatísticos - *Summary statistic*, valores referentes a estatísticas e parâmetros de distribuições de probabilidade para a representação sintética do consumo dos recursos computacionais selecionados. Após serem gerados, os *traces* são armazenados em arquivos CSV com a representação da utilização de pelo menos um dos recursos computacionais disponíveis: Processador, memória, disco e rede. Posteriormente os arquivos gerados devem ser inseridos no *Capacity Advisor* através da funcionalidade “*Capacity Advisor Data Import Tools*” que extrai os valores contidos no arquivo CSV e os armazena na base de dados históricos da ferramenta (*Capacity Advisor historical data*).

A geração de *traces* sintéticos para o *Capacity Advisor* é extremamente importante, uma vez que a sua estrutura atual permite apenas o planejamento de capacidade através da utilização de *traces* pertencentes a sistemas reais em produção e que estejam configurados em plataformas compatíveis com a metodologia atual de coleta de *traces* para o *Capacity Advisor*. O desenvolvimento do gerador de carga possibilita a realização do planejamento de capacidade com a ferramenta independentemente da plataforma utilizada e sem a necessidade da utilização de *traces* pertencentes a sistemas reais, já que o gerador simula a coleta dos *traces*, que por sua vez, são gravados em arquivos compatíveis com o *Capacity Advisor*.

As próximas seções detalham a metodologia utilizada para a concepção do gerador de carga desenvolvido, as principais atividades executadas durante o seu desenvolvimento e o seu processo de validação.

4.2 METODOLOGIA ADOTADA

A metodologia adotada para a concepção do gerador de *traces* sintéticos para o *Capacity Advisor* é composta por oito etapas:

1. Compreensão do problema e entendimento do sistema;
2. Estudo das metodologias de geração de carga;
3. Definição dos cenários de geração de carga;
4. Desenvolvimento do *kernel*;
5. Validação do *kernel*;
6. Definição do processo de geração de carga;
7. Desenvolvimento do gerador de carga;
8. Estudos de caso.

A Figura 4.2 ilustra o diagrama de atividades da metodologia.

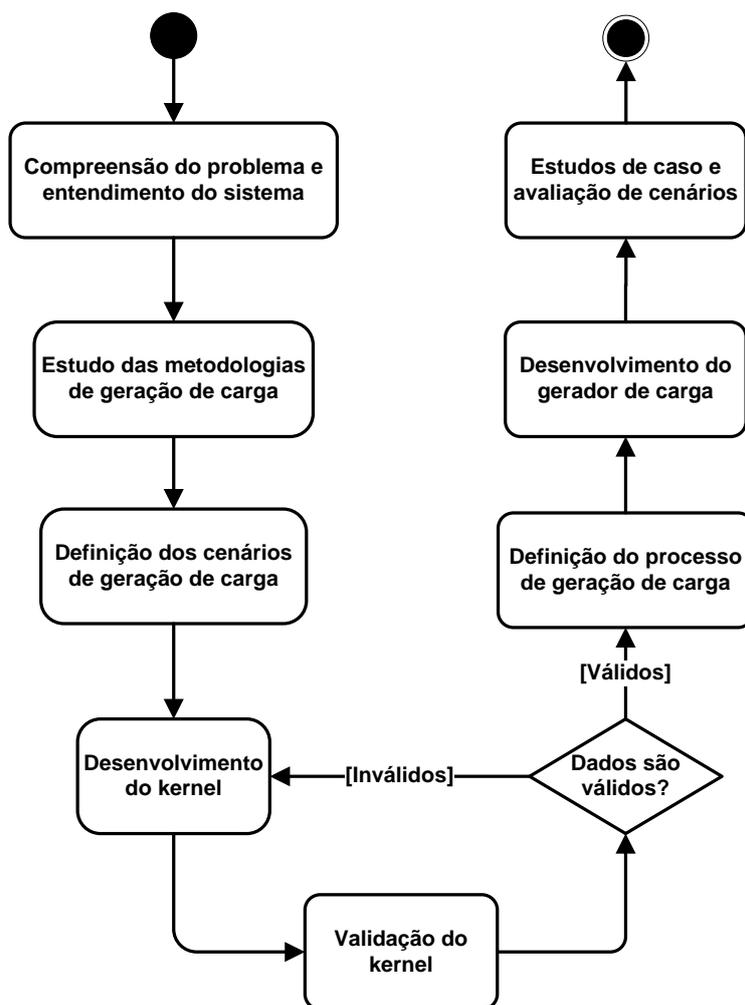


Figura 4.2: Metodologia adotada para a concepção do gerador de carga.

4.2.1 Compreensão do problema e entendimento do sistema

A primeira atividade da metodologia corresponde ao estudo e compreensão do ambiente onde o gerador de carga deve atuar, ressaltando o contexto em que a aplicação está inserida. Durante esta etapa realizou-se um estudo sobre as principais tecnologias de virtualização desenvolvidas pela HP (*Hewlett Packard*) para a obtenção do conhecimento de seus principais componentes, suas principais características, funcionalidades, potencialidades e limitações. Inicialmente foi detalhado o funcionamento do ambiente de servidores virtuais HP VSE (*Virtual Server Environment*) e posteriormente foi realizado um estudo aprofundado sobre um de seus principais componentes e foco deste trabalho, o HP *Capacity Advisor*.

Através deste estudo foi possível identificar as características mais importantes do ambiente VSE, o funcionamento e a arquitetura do *Capacity Advisor*, a formatação dos *traces* exportados atualmente pela ferramenta, suas principais limitações e as motivações para o desenvolvimento do gerador de carga. Desta forma, pôde ser definido o funcionamento do gerador de carga e as próximas atividades necessárias para a adequada execução do projeto, sendo esta fase da metodologia crucial para que não existam erros de interpretação durante o desenvolvimento do gerador de *traces* sintéticos.

4.2.2 Estudo das metodologias de geração de carga

A segunda atividade corresponde ao estudo das metodologias de geração de carga existentes. Durante esta etapa foi analisada toda a parte de fundamentação teórica e o estado da arte da pesquisa, com o intuito de se obter um conhecimento abrangente tanto na teoria quanto em trabalhos relacionados a geração de carga, para desta forma escolher as estratégias que melhor se adequem a pesquisa desenvolvida.

Entre os principais tópicos analisados podemos destacar: O estudo de diversas distribuições de probabilidade (contínuas e discretas), focando nas características principais de cada uma e a importância de sua utilização no processo de geração de carga; a geração de números e variáveis aleatórias (*Random Variates*) [Dev86, Cas08, Gen98], enfatizando o estudo de valores aleatórios baseados em distribuições de probabilidade; a análise das principais metodologias de *Distribution Fitting* [Gen98]; além de uma pesquisa detalhada sobre os principais trabalhos relacionados a geração de carga, publicados em eventos, revistas e periódicos importantes da comunidade científica.

4.2.3 Definição dos cenários de geração de carga

A terceira atividade corresponde a definição dos cenários de geração de carga aplicados ao contexto do problema. Durante esta fase realizou-se uma análise de requisitos com o objetivo de identificar as principais limitações do *Capacity Advisor*, e desenvolver uma estrutura de cenários onde o gerador de *traces* sintéticos deve atuar para suprir estas limitações.

Durante esta etapa foram definidos os três seguintes cenários:

- **Cenário 01 - *ReTrace***: *Geração de traces sintéticos baseados em traces reais;*

- **Cenário 02 - *Statistic Trace***: Geração de *traces* sintéticos baseados em resumos estatísticos e distribuições de probabilidade;
- **Cenário 03 - *Statistic ReTrace***: Geração de resumos estatísticos e *traces* sintéticos baseados em *traces* reais.

Os três cenários de geração de carga desenvolvidos são detalhados no Capítulo 5.

4.2.4 Desenvolvimento do *kernel*

A quarta atividade corresponde ao desenvolvimento do *kernel* (núcleo) para o gerador de carga. Durante esta etapa foi desenvolvido um gerador de variáveis aleatórias genérico, onde foram implementadas as metodologias para a geração de valores aleatórios baseados em algumas das principais distribuições de probabilidade existentes (contínuas e discretas), através da utilização das técnicas de geração de *random variates* [Dev86, Cas08, Gen98] estudadas na segunda atividade e descritas no Capítulo 2. Foram aplicadas e desenvolvidas metodologias para a geração de valores aleatórios baseados em dez das distribuições de probabilidades teóricas mais importantes, além da distribuição empírica, implementada com o intuito de se aplicar ao cenário de geração de *traces* sintéticos baseados em *traces* reais (*ReTrace*). A lista das distribuições de probabilidade implementadas durante esta etapa é mostrada a seguir:

- Empírica (Contínua)
- Erlang (Contínua)
- Exponencial (Contínua)
- Geométrica (Discreta)
- LogNormal (Contínua)
- Normal (Contínua)
- Pareto (Contínua)
- Poisson (Discreta)
- Triangular (Contínua)

- Uniforme (Contínua)
- Weibull(Contínua)

O desenvolvimento do *kernel* possui um papel fundamental para o desenvolvimento do gerador de *traces* sintéticos para o *Capacity Advisor*, uma vez que a sua função é gerar os valores pertencentes aos *traces* que representam o consumo de cada um dos recursos computacionais disponíveis pela ferramenta (processador, memória, disco e rede).

4.2.5 Validação do *kernel*

A quinta atividade corresponde a validação dos valores gerados pelo *kernel* do gerador de carga. Esta atividade analisa se os valores aleatórios gerados pela ferramenta realmente correspondem às distribuições de probabilidade teóricas originais.

A validação inicial foi realizada dentro do próprio *kernel*, através da implementação da estratégia de *distribution fitting Kolmogorov-Smirnov* [Gen98] aplicada nas amostras geradas pela ferramenta, além de um gráfico comparativo entre o histograma dos dados gerados e as distribuições de probabilidade correspondentes. Posteriormente, as amostras geradas pelo *kernel* foram avaliadas em algumas ferramentas estatísticas, tais como o *EasyFit* e o *Minitab*, onde foram realizados testes de *distribution fitting* mais complexos (ver Capítulo 2).

Para que o *kernel* seja considerado válido, é necessário que as amostras de cada uma das distribuições de probabilidade sejam validadas em pelo menos um dos testes realizados. Caso isso não ocorra com alguma amostra, é necessário revisar a implementação do *kernel*, visando encontrar erros cometidos nessa etapa.

Mais detalhes sobre o processo de validação do *kernel* são descritos na próxima seção.

4.2.6 Definição do processo de geração de carga

A sexta atividade corresponde a definição de todos os estágios necessários para o processo de geração dos *traces* sintéticos para o *Capacity Advisor*. Durante esta etapa foi estruturada a ordem de atividades que devem ser realizadas pelo gerador de carga para que possam ser gerados arquivos CSV contendo todas as propriedades, métricas e valores referentes aos *traces* sintéticos.

As seguintes atividades foram definidas durante esta fase: definição das propriedades

do *trace*, definição do período das amostras, seleção das métricas, seleção da metodologia de geração de carga e geração do *trace*. Estas etapas são detalhadas no Capítulo 5.

4.2.7 Desenvolvimento do gerador de carga

Uma vez definido o processo de geração de carga, iniciou-se a etapa correspondente ao desenvolvimento do gerador de *traces* sintéticos. Durante esta etapa foi desenvolvida uma aplicação completa, objetivando a geração de arquivos CSV compatíveis com o *Capacity Advisor*, contendo a representação da utilização dos recursos computacionais e os valores de seus respectivos *traces* sintéticos. Estes valores são gerados através do *kernel* desenvolvido, que foi acoplado a nova ferramenta durante a sua implementação.

O funcionamento da ferramenta baseia-se no processo de geração de carga realizado na etapa anterior, além de obedecer as necessidades observadas nos três cenários definidos, cumprindo desta forma o objetivo do gerador de carga. Sua estrutura é composta por dois módulos: *Módulo de geração de carga*, responsável pela geração de *traces* baseados em distribuições de probabilidade ou resumos estatísticos, suprimindo as necessidades dos cenários 01, 02 e 03; e o *Módulo de geração de resumos estatísticos*, responsável por extrair informações contidas em *traces* reais e gerar o seu respectivo resumo estatístico, suprimindo as necessidades do cenário 03. Desta forma, para que o cenário 03 seja utilizado, devem ser aplicados os dois módulos.

O funcionamento da ferramenta desenvolvida e a utilização dos módulos aplicados a cada cenário são descritos em detalhes no Capítulo 5.

4.2.8 Estudos de caso e avaliação de cenários

A última etapa da metodologia adotada corresponde a realização de estudos de caso através da utilização do ferramental desenvolvido. Durante esta etapa foram realizadas simulações com diversos *traces* sintéticos aplicados ao *Capacity Advisor*, com o intuito de avaliar cada um dos cenários de geração de carga desenvolvidos e analisar os resultados obtidos. Os detalhes desta atividade são descritos no Capítulo 6.

4.3 DESENVOLVIMENTO DO FERRAMENTAL

Conforme observado na seção anterior, o desenvolvimento do gerador de *traces* sintéticos para o *Capacity Advisor* foi realizado em duas etapas. A primeira etapa corresponde ao *desenvolvimento do kernel para o gerador* e a segunda etapa visa o *desenvolvimento do gerador de carga final*. Para cada uma destas etapas foi utilizado o mesmo processo de desenvolvimento, envolvendo as seguintes atividades: definição da arquitetura e funcionamento do ferramental (*kernel* e gerador de carga), implementação e a realização de testes para a validação dos resultados. É importante salientar que antes da execução de cada uma destas etapas foram realizadas duas atividades iniciais referentes a escolha da linguagem de programação e do ambiente de desenvolvimento a ser utilizado. O fluxograma com estas atividades é ilustrado na Figura 4.3.

A primeira atividade do fluxograma se refere a escolha da linguagem de programação utilizada para o desenvolvimento do ferramental. Visando criar uma ferramenta com maior extensibilidade, reusabilidade e flexibilidade, a linguagem de programação adotada para o seu desenvolvimento foi o *Java* [jav09], seguindo cuidadosamente as técnicas de orientação a objetos. Uma das características mais importante da linguagem é a sua portabilidade, que permite a independência da plataforma de sistema operacional utilizada para a execução do ferramental desenvolvido, possibilitando o seu funcionamento em qualquer sistema que suporte a máquina virtual *Java* (JVM). Além disso, a linguagem é distribuída com um vasto conjunto de ferramentas, *frameworks*, bibliotecas e APIs (*Application Program Interface*) de código aberto (*open-source*), que possibilitam o reuso de metodologias já consolidadas e um desenvolvimento mais ágil e consistente.

A segunda atividade do fluxograma corresponde a escolha do ambiente de desenvolvimento utilizado. Durante esta etapa foi selecionado o ambiente de desenvolvimento integrado *NetBeans IDE (Integrated Development Environment)* [net09]. O *NetBeans* é um ambiente multiplataforma escrito em *Java*, gratuito e de código aberto, utilizado para o desenvolvimento de *software* nas linguagens *Java*, *C/C++*, *PHP*, *Groovy*, *Ruby*, entre outras. Este ambiente visa auxiliar os desenvolvedores de *software* a editar, compilar, testar e depurar códigos, através de um conjunto integrado de ferramentas para a criação de aplicativos *desktop*, *web*, empresariais e móveis multiplataforma, reunindo diversas funcionalidades em uma única aplicação. Adicionalmente o *NetBeans* fornece um grande conjunto de bibliotecas, módulos e APIs, além de uma documentação bem organizada em diversos idiomas. A distribuição da ferramenta é realizada sob as condições da licença SPL (*Sun Public License*), que tem como objetivo garantir a redistribuição

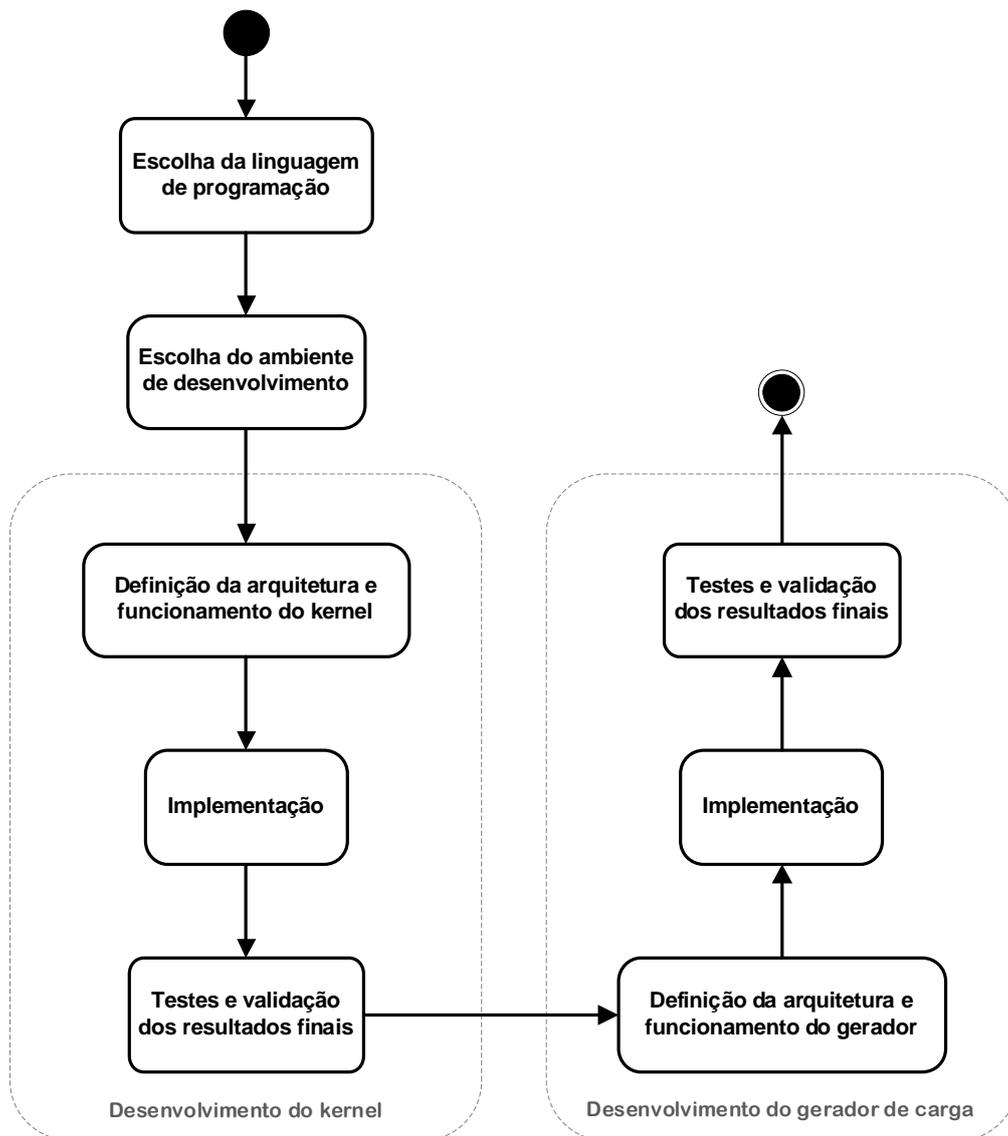


Figura 4.3: Fluxograma das atividades adotadas durante a implementação do ferramental.

de conhecimento à comunidade de desenvolvedores quando novas funcionalidades forem incorporadas à ferramenta.

A terceira atividade do fluxograma corresponde a definição da arquitetura e funcionamento do *kernel*. Durante esta etapa foram definidas e documentadas todas as atividades e papéis referentes ao funcionamento do *kernel* a ser implementado, através de diagramas de casos de uso, fluxograma de sequência de atividades e diagramas de classe. Este processo é extremamente importante para que se obtenha sucesso durante o desenvolvimento das aplicações, possibilitando o planejamento de todos os passos que serão executados

durante a etapa de implementação.

A quarta atividade do fluxograma se refere a implementação do *kernel*. Esta etapa é responsável por toda parte de codificação da aplicação, onde através da utilização do ambiente de desenvolvimento *NetBeans IDE*, são criadas interfaces, classes, objetos e módulos responsáveis por executar os papéis referentes a cada uma das atividade pertencentes ao ferramental.

A quinta atividade do fluxograma corresponde a realização de testes e validação dos resultados gerados pelo kernel. Durante esta etapa foram realizados testes nas classes e módulos das aplicações, com o intuito de garantir um bom desempenho e confiabilidade para a ferramenta. Posteriormente, foi realizada a validação dos dados gerados pelo *kernel*, seguindo os passos descritos em 4.2.5.

A sexta atividade do fluxograma corresponde a definição da arquitetura e funcionamento do gerador de carga. Durante esta etapa foram definidas e documentadas todas as atividades e papéis referentes ao funcionamento do gerador de *traces* sintéticos a ser implementados, através de diagramas de casos de uso, fluxograma de sequência de atividades e diagramas de classe, funcionando de maneira análoga a terceira atividade do fluxograma.

A sétima atividade do fluxograma se refere a implementação do gerador de carga para o *Capacity Advisor*. Esta etapa funciona de maneira análoga a quarta atividade do fluxograma, sendo responsável pela execução de toda parte de codificação do gerador, onde através da utilização do ambiente de desenvolvimento *NetBeans IDE*, são criadas interfaces, classes, objetos e módulos responsáveis por executar os papéis referentes a cada uma das atividade pertencentes ao ferramental.

A oitava e última atividade do fluxograma, corresponde a realização de testes e validação dos resultados gerados pelo gerador de carga. Durante esta etapa foram realizados testes nas classes e módulos das aplicações, com o intuito de garantir um bom desempenho e confiabilidade para a ferramenta. Por último, foi realizada uma avaliação dos dados gerados pelo gerador de carga através de estudos de casos, seguindo os passos descritos em 4.2.8.

4.4 VALIDAÇÃO

Para garantir a consistência dos valores gerados pelo gerador de *traces* sintéticos para o *Capacity Advisor* foi realizada uma validação das amostras geradas pelo *kernel* com

base nas distribuições de probabilidade implementadas. Conforme descrito na Seção 4.2.5, inicialmente foram desenvolvidos mecanismos para a validação dos dados dentro do próprio *kernel*, através da implementação da estratégia de *distribution fitting Kolmogorov-Smirnov* [Gen98], e a construção de gráficos comparativos entre o histograma dos dados gerados e as distribuições de probabilidade correspondentes. Estes mecanismos são mostrados no próximo capítulo, onde é apresentado o ferramental desenvolvido.

Além da validação construída dentro do *kernel*, foram realizados alguns testes de *distribution fitting* adicionais, com o intuito de avaliar os valores gerados de forma mais precisa. Estes testes foram executados através da utilização das ferramentas de cálculo estatístico *EasyFit* [eas09] e *Minitab* [min09, Min03]. Durante esta etapa foram geradas 10.000 amostras sintéticas de cada distribuição de probabilidade, e em seguida foram aplicadas a cada conjunto de amostras as seguintes metodologias de *distribution fitting: Kolmogorov-Smirnov, Anderson Darling* e *Chi-Squared* [Gen98]. Os resultados obtidos durante esta etapa são ilustrados nas tabelas 4.1, 4.2 e 4.3. Para cada teste utilizado foram calculadas estatísticas específicas com base nas amostras das distribuições de probabilidade, com o nível de significância “Alfa” = 0,01 (garantindo desta forma um alto grau de confiança estatística). Para que uma determinada distribuição seja considerada válida, o valor da estatística (coluna “Estatística” definida nas tabelas) não poderá ser maior que os valores críticos definidos para cada distribuição de probabilidade (coluna “Valor Crítico” definida nas tabelas). Observe que todas as amostras geradas foram aceitas em pelo menos um dos testes aplicados.

Tabela 4.1: Resultados do teste *Kolmogorov-Smirnov*.

<i>Distribuição</i>	<i>Estatística</i>	<i>Alfa</i>	<i>Valor Crítico</i>	<i>Rejeitar?</i>
<i>Erlang</i>	0,00256	0,01	0,00515	<i>Não</i>
<i>Exponencial</i>	0,0025	0,01	0,00515	<i>Não</i>
<i>LogNormal</i>	0,00167	0,01	0,00515	<i>Não</i>
<i>Normal</i>	0,00223	0,01	0,00515	<i>Não</i>
<i>Pareto</i>	0,00231	0,01	0,00515	<i>Não</i>
<i>Triangular</i>	0,00521	0,01	0,0163	<i>Não</i>
<i>Uniforme</i>	0,00576	0,01	0,0163	<i>Não</i>
<i>Weibull</i>	0,00266	0,01	0,00515	<i>Não</i>

Para realizar a validação da distribuição empírica, foi importado para a ferramenta um *trace* coletado de um sistema real, contendo 8.928 amostras com valores representando o consumo de seu processamento (CPU). Em seguida, a mesma quantidade de amostras

Tabela 4.2: Resultados do teste *Anderson-Darling*.

<i>Distribuição</i>	<i>Estatística</i>	<i>Alfa</i>	<i>Valor Crítico</i>	<i>Rejeitar?</i>
<i>Erlang</i>	1,117	0,01	3,9074	<i>Não</i>
<i>Exponencial</i>	0,82915	0,01	3,9074	<i>Não</i>
<i>LogNormal</i>	0,39719	0,01	3,9074	<i>Não</i>
<i>Normal</i>	0,37177	0,01	3,9074	<i>Não</i>
<i>Pareto</i>	2,6712	0,01	3,9074	<i>Não</i>
<i>Triangular</i>	0,31205	0,01	3,9074	<i>Não</i>
<i>Weibull</i>	0,81488	0,01	3,9074	<i>Não</i>

Tabela 4.3: Resultados do teste *Chi-Squared*.

<i>Distribuição</i>	<i>Estatística</i>	<i>Alfa</i>	<i>Valor Crítico</i>	<i>Rejeitar?</i>
<i>Erlang</i>	22,037	0,01	32,0	<i>Não</i>
<i>Exponencial</i>	16,419	0,01	32,0	<i>Não</i>
<i>Geometrica</i>	0,00183	0,01	0,17182	<i>Não</i>
<i>LogNormal</i>	15,77	0,01	32,0	<i>Não</i>
<i>Normal</i>	9,1523	0,01	32,0	<i>Não</i>
<i>Pareto</i>	12,426	0,01	32,0	<i>Não</i>
<i>Poisson</i>	0,00142	0,01	0,21043	<i>Não</i>
<i>Triangular</i>	6,3006	0,01	27,688	<i>Não</i>
<i>Weibull</i>	18,695	0,01	32,0	<i>Não</i>

foi gerada com base no *trace* importado, através da distribuição empírica. Por último foi gerado um gráfico comparativo com a função de distribuição acumulada de cada uma das amostras. Os resultados obtidos são mostrados na Figura 4.4. Através da análise do gráfico, pode-se observar que as duas curvas estão muito próximas, desta forma pudemos concluir que o conjunto de valores das duas amostras (real e empírica) estão distribuídos de maneira equivalente

Além da validação do *kernel* para o gerador de carga, foi realizada uma avaliação de seu desempenho com o objetivo de analisar o tempo médio de geração dos valores sintéticos. Foram geradas de 10.000 a 1.000.000 de amostras para cada uma das distribuições de probabilidade e calculado o tempo médio de geração, em milissegundos, para cada uma delas. Os resultados obtidos são mostrados nas figuras 4.5 e 4.6. Observe que o eixo *X* (horizontal) define a quantidade de amostras geradas, divididas em intervalos de 100.000 amostras; e o eixo *Y* (vertical) define o tempo gasto durante a geração

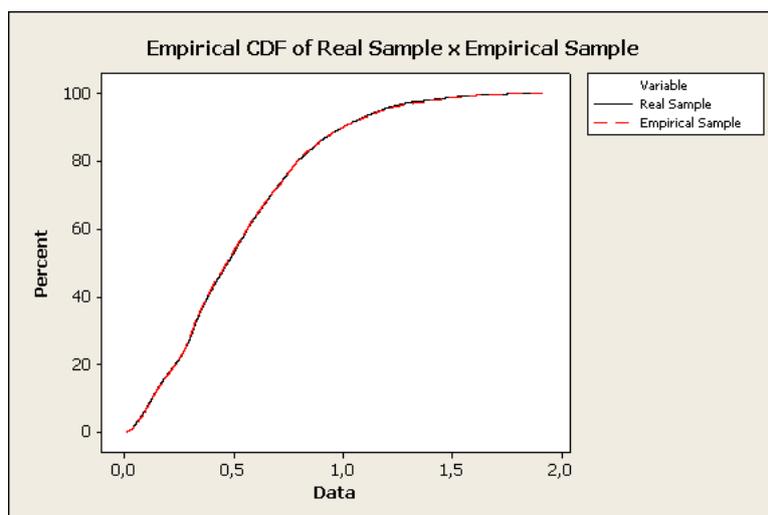


Figura 4.4: Função de distribuição acumulada - Amostra real x Amostra empírica.

de cada quantidade de amostras, medido em milissegundos.

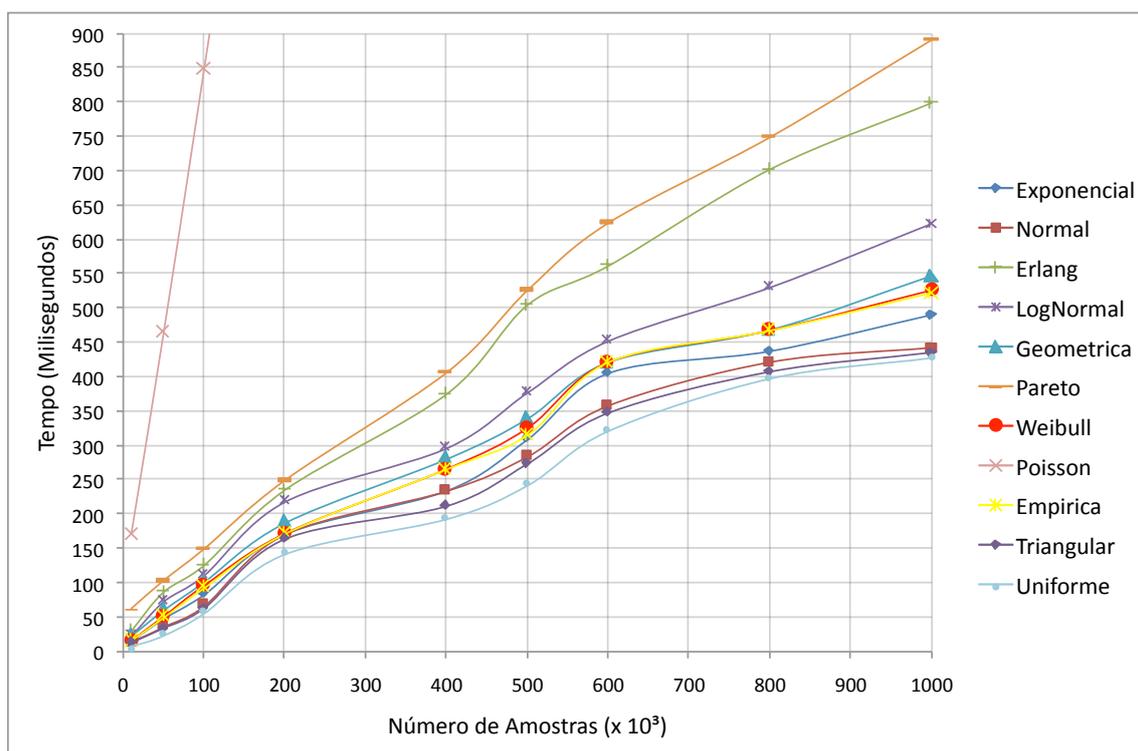


Figura 4.5: *Kernel* - Número de amostras x Tempo médio de geração.

Observe que todas as distribuições de probabilidade, com exceção da distribuição de Poisson, foram capazes de gerar até um milhão de amostras em menos de 900 milissegundos (0,9 segundos), e sete delas em menos de 600 milissegundos (0,6 segundos).

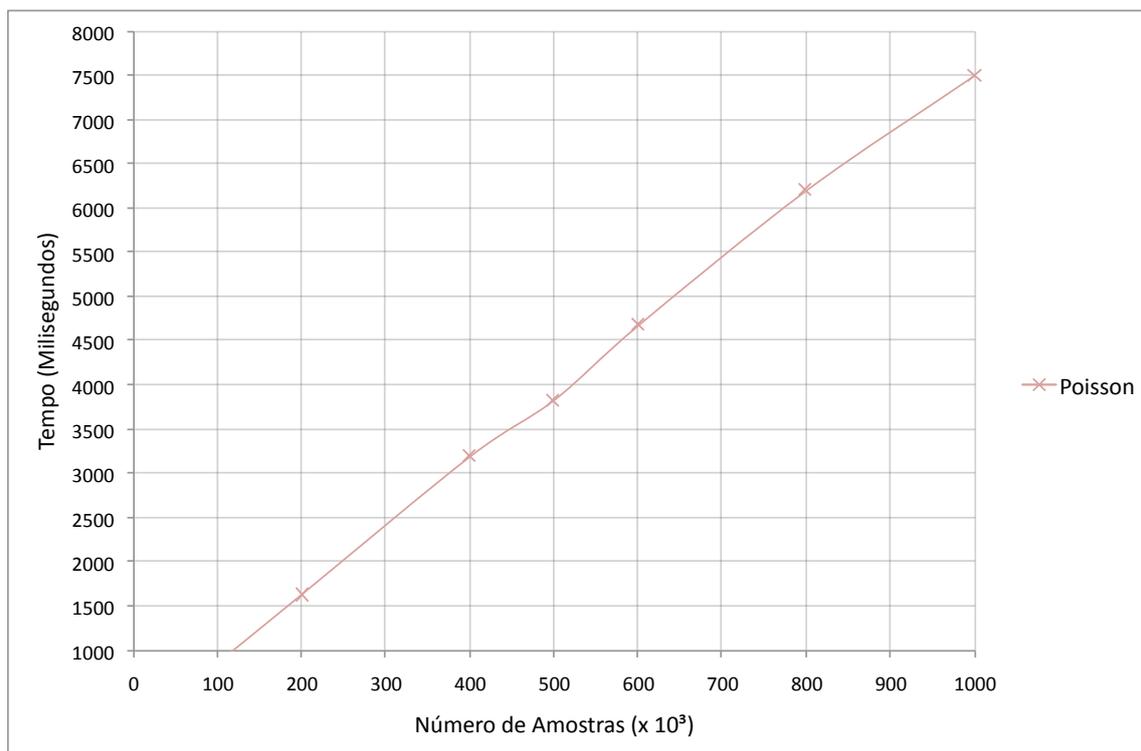


Figura 4.6: *Kernel* - Número de amostras x Tempo médio de geração (Cont. distribuição Poisson).

É importante salientar que a duração da execução do algoritmo de geração de valores aleatórios baseados na distribuição de Poisson é diretamente proporcional a sua taxa (*rate*), o que significa que, para a mesma quantidade de amostras, quanto maior a taxa, maior será o tempo de execução do algoritmo. Nesta avaliação, o valor definido pra a taxa foi igual a 50 e o tempo médio para a geração de um milhão de amostras foi de 7.507 milissegundos (7,5 segundos). No entanto, consideramos este valor aceitável para o gerador de carga, já que apenas algumas distribuições são usados a cada geração. Desta forma, podemos assegurar que o gerador de carga possui um bom desempenho.

4.5 SUMÁRIO

Este capítulo apresentou a metodologia adotada para a concepção do gerador de carga para o *Capacity Advisor*. Esta metodologia é composta por uma série de atividades extremamente importantes que vão desde o entendimento do ambiente e o estudo de metodologias de geração de carga existentes, passando pela definição dos cenários e do processo de geração de carga, chegando até o desenvolvimento do gerador de *traces*

sintéticos para o *Capacity Advisor*. Além da metodologia desenvolvida também foram descritos os passos executados durante o desenvolvimento do ferramental e os mecanismos utilizados para a sua validação.

CAPÍTULO 5

FERRAMENTAL DESENVOLVIDO

Este capítulo apresenta o ferramental desenvolvido para a geração de *traces* sintéticos para o *Capacity Advisor*. Inicialmente, são mostrados os cenários de geração de carga suportados pela ferramenta. Em seguida, é descrito o processo de geração de carga adotado, onde é definida a ordem com todos os passos executados durante a geração dos *traces* sintéticos. Posteriormente, é mostrado o *kernel* para o gerador de carga desenvolvido, onde são abordadas suas principais características e utilização. Por fim, é apresentada a ferramenta WGCap (*Workload Generator for Capacity Advisor*), foco deste trabalho.

5.1 CENÁRIOS DE GERAÇÃO DE CARGA

Conforme observado na Seção 4.2.3, a geração de *traces* sintéticos para o *Capacity Advisor* deve ocorrer de acordo com três cenários de geração de carga: *ReTrace* - Geração de *traces* sintéticos baseados em *traces* reais, *Statistic Trace* - Geração de *traces* sintéticos baseados em resumos estatísticos e distribuições de probabilidade e *Statistic ReTrace* - Geração de resumos estatísticos e *traces* sintéticos baseados em *traces* reais. Cada um destes cenários é detalhados a seguir.

5.1.1 Cenário 01 - *ReTrace*: Geração de *traces* sintéticos baseados em *traces* reais

Este cenário é responsável por permitir a geração de *traces* sintéticos a partir de *traces* reais. Através da utilização de arquivos contendo o histórico de *traces* coletados de sistemas reais em produção, pode ser realizada a geração de *traces* artificiais (sintéticos), com o comportamento equivalente ao *trace* real fornecido. Neste cenário pode-se, por exemplo, considerar um *trace* real referente a uma semana significativa para a geração de um *trace* sintético relativo ao período de um ano, seguindo o comportamento do *trace* real considerado.

A Figura 5.1 ilustra o fluxograma referente ao Cenário 01. Inicialmente, deve ser importado para a ferramenta (WGCap) um *trace* real contendo os valores referentes ao consumo de cada recurso a ser simulado. Em seguida, é gerado um *trace* sintético com base nos valores fornecidos para que finalmente possa ser utilizado pelo *Capacity Advisor*.

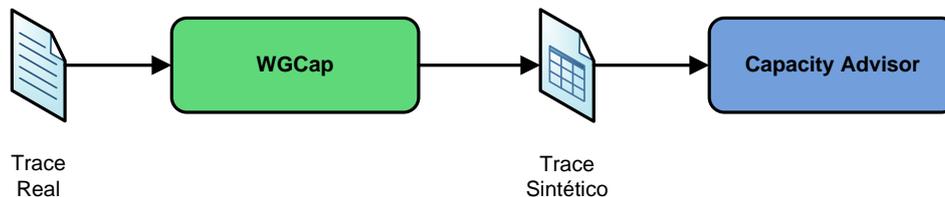


Figura 5.1: Cenário 01 - *ReTrace*.

5.1.2 Cenário 02 - *Statistic Trace*: Geração de traces sintéticos baseados em resumos estatísticos e distribuições de probabilidade

Este cenário é responsável por permitir a geração de *traces* sintéticos baseados em resumos estatísticos e distribuições de probabilidade. Neste cenário, o comportamento de cada recurso pertencente ao *trace* sintético (processador, disco, memória e rede) deve se basear em uma das diversas distribuições de probabilidade disponibilizadas pelo gerador de carga. O usuário tem a opção de gerar valores baseados em distribuições de probabilidade ou em resumos estatísticos. Caso a primeira opção seja escolhida, o usuário pode selecionar livremente a distribuição utilizada, devendo informar os valores dos parâmetros pertencentes a cada distribuição. Caso a segunda opção seja escolhida, devem ser fornecidos resumos estatísticos para que a ferramenta selecione uma lista com as distribuições de probabilidade mais adequadas, para em seguida gerar os valores associados.

A Figura 5.2 ilustra o fluxograma referente ao Cenário 02. Inicialmente, deve ser selecionada uma das opções para a geração dos valores: distribuições de probabilidade (*Distribuição*) ou resumos estatísticos (*Estatística*). Em seguida, devem ser fornecidos para a ferramenta os respectivos valores de acordo com a opção selecionada (resumo estatístico ou parâmetros para a distribuição de probabilidade). Por último, é gerado um *trace* sintético com base nos valores fornecidos para que finalmente possa ser utilizado pelo *Capacity Advisor*.

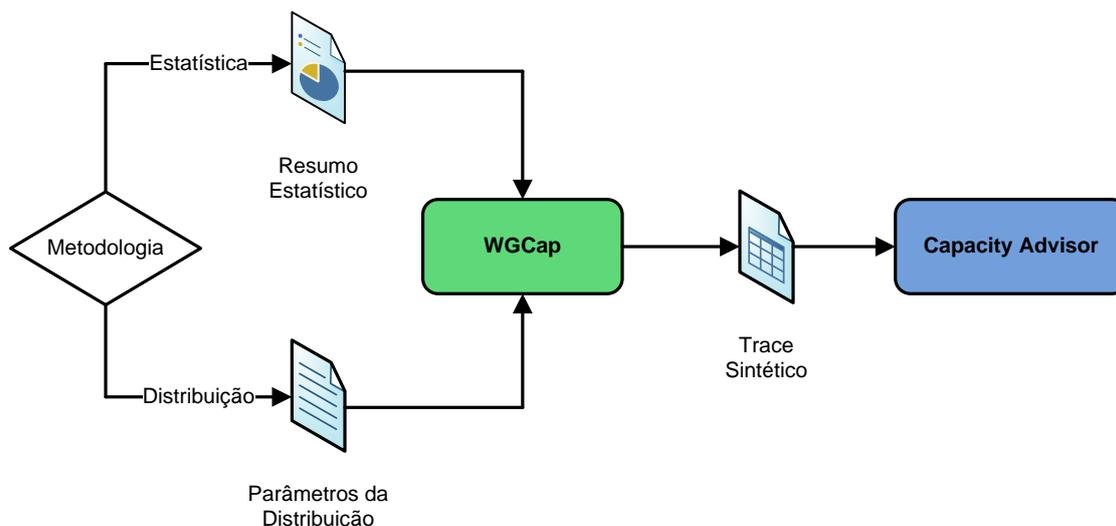


Figura 5.2: Cenário 02 - *Statistic Trace*.

5.1.3 Cenário 03 - *Statistic ReTrace*: Geração de resumos estatísticos e traces sintéticos baseados em traces reais

Este cenário é responsável por permitir a geração de resumos estatísticos baseados em *traces* reais, para que posteriormente sejam gerados *traces* sintéticos. Esta situação ocorre quando os administradores de sistemas não podem fornecer *traces* reais para que sejam diretamente utilizados pelo gerador de carga. Neste caso, a ferramenta pode ser fornecida, para que os próprios administradores gerem as estatísticas dos *traces* reais e as disponibilizem para a geração dos *traces* sintéticos.

A Figura 5.3 ilustra o fluxograma referente ao Cenário 03. Inicialmente, deve ser importado para a ferramenta um *trace* real contendo os valores referentes ao consumo de cada recurso a ser simulado para que seja gerado seu respectivo resumo estatístico. O resumo gerado pode ser utilizado posteriormente, onde deve ser utilizado pela ferramenta como dado de entrada para a geração de *traces* baseados em resumos estatísticos. A partir daí, é gerado um *trace* sintético com base nos valores fornecidos para que finalmente possa ser utilizado pelo *Capacity Advisor*.

5.2 PROCESSO DE GERAÇÃO DE CARGA

O processo de geração de *traces* sintéticos para o *Capacity Advisor* é composto por cinco atividades: definição das propriedades do *trace*, definição do período das amostras, seleção

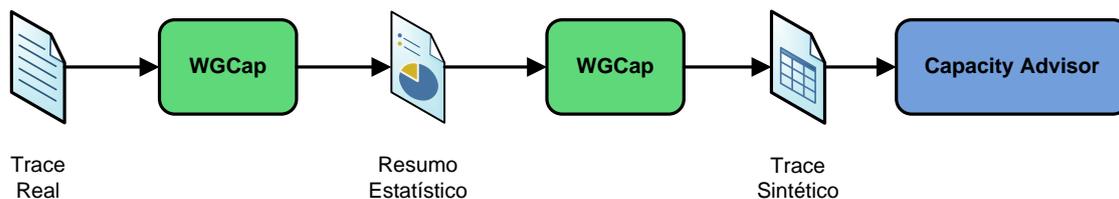


Figura 5.3: Cenário 03 - *Statistic ReTrace*.

das métricas, seleção da metodologia de geração de carga e geração do *trace*. A Figura 5.4 ilustra o fluxograma com as etapas descritas.

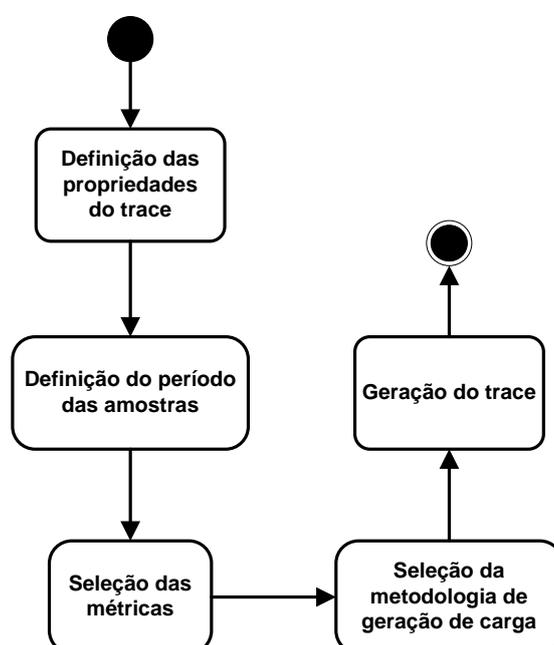


Figura 5.4: Fluxograma das atividades do processo geração de carga.

A primeira atividade do fluxograma se refere a definição das propriedades do *trace* sintético a ser gerado. Nesta etapa são definidas as características do ambiente a ser simulado, para que sejam incluídas no cabeçalho (*header*) do *trace*. Durante esta atividade são definidas as seguintes informações : *Profile name*, *Hostname*, *CPU_count*, *CPU_speed*, *Memory size*, *OS platform* e *model*. Estes campos serão detalhados na Seção 5.4.1.

A segunda atividade do fluxograma se refere a definição do período das amostras a serem geradas. Esta etapa é responsável pela definição do número de amostras que serão geradas pela ferramenta, além da representação do período referente a data inicial e final da geração dos *traces* sintéticos. Desta forma pode ser simulado o período em que as amostras sintéticas serão geradas, sendo incrementadas em intervalos de cinco minutos.

As datas geradas nesta etapa devem obedecer ao seguinte formato: YYYYMMDDhhmm, onde YYYY representa o ano, MM representa o mês (de 01 a 12), DD representa o dia (de 01 a 31), hh representa as horas (de 00 a 23) e mm representa os minutos (de 00 a 59).

A terceira atividade do fluxograma, corresponde a seleção das métricas referentes aos recursos computacionais selecionados para serem utilizados no *trace* sintético. Durante esta etapa deve ser selecionada pelo menos uma das seguinte métricas: CPU_UTIL, DISK_UTIL, MEM_UTIL e NET_UTIL.

A quarta atividade do fluxograma se refere a seleção da metodologia de geração de carga utilizada para a geração do *trace sintético*. Durante esta etapa deve ser selecionado um dos dois métodos de geração de *traces* desenvolvidos nesta pesquisa: *A geração de traces baseados em distribuições de probabilidade* ou *a geração de traces baseados em resumos estatísticos*. No primeiro caso podem ser gerados valores baseados em qualquer uma das distribuições teóricas existentes ou baseados em *traces* reais fornecidos pelo usuário (distribuição Empírica). Já no segundo caso, podem ser gerados valores a partir de resumos estatísticos (tais como média, desvio padrão, valores máximo e mínimo, etc), onde são indicadas as distribuições de probabilidade mais adequadas de acordo com as estatísticas fornecidas.

A quinta e última atividade do fluxograma, corresponde a geração do *trace* sintético final. Durante esta etapa, todas as informações adquiridas nas atividades anteriores, são utilizadas para que os valores dos *traces* sintéticos (referentes a cada uma das métricas selecionadas) sejam gerados e armazenados em um arquivo CSV compatível com o *Capacity Advisor*.

5.3 *KERNEL*

Conforme descrito anteriormente, antes da implementação do gerador de carga para o *Capacity Advisor* foi desenvolvido um *kernel* (núcleo), que serve como base para a geração dos valores que irão fazer parte dos *traces* sintéticos. O objetivo principal deste *kernel* é gerar amostras com valores aleatórios baseados em algumas das principais distribuições de probabilidade existentes (contínuas e discretas), através da aplicação de técnicas de geração de variáveis aleatórias (*random variates*) [Dev86, Cas08, Gen98]. Inicialmente foram aplicadas metodologias para a geração de variáveis aleatórias com base em dez distribuições de probabilidades teóricas importantes, além da distribuição empírica, cujo

objetivo é gerar valores aleatórios baseados em *traces* reais com qualquer comportamento observado. As dez distribuições teóricas implementadas pelo *kernel* são mostradas na Tabela 5.1.

Tabela 5.1: Distribuições teóricas implementadas pelo *kernel*.

<i>Erlang</i> (Contínua)	<i>Pareto</i> (Contínua)
<i>Exponencial</i> (Contínua)	<i>Poisson</i> (Discreta)
<i>Geométrica</i> (Discreta)	<i>Triangular</i> (Contínua)
<i>LogNormal</i> (Contínua)	<i>Weibull</i> (Contínua)
<i>Normal</i> (Contínua)	<i>Uniforme</i> (Contínua)

Além das distribuições de probabilidade acima, novas distribuições podem ser adicionadas à ferramenta, caso seja necessário. A Figura 5.5 ilustra a aba “*Projects*” do *NetBeans* IDE contendo o projeto do *kernel* desenvolvido, onde é mostrado o pacote contendo as classes responsáveis por implementar os métodos para a geração de números aleatórios. Podemos observar que para cada distribuição é criada uma classe, como por exemplo a classe para a distribuição empírica *EmpiricalRandomVariateGenerator.java*, onde são implementados seus respectivos mecanismos de geração de variáveis aleatórias. Desta forma, para acrescentar uma nova distribuição à ferramenta, basta a criação de uma nova classe contendo seus respectivos algoritmos para a geração de variáveis aleatórias, garantindo assim uma grande flexibilidade para a ferramenta.

Os principais recursos fornecidos pelo *kernel* desenvolvido são listados a seguir:

- Geração de valores aleatórios baseados em dez distribuições de probabilidade teóricas (Tabela 5.1);
- Geração de valores aleatórios baseados no comportamento de qualquer *trace* real, através da distribuição empírica;
- Visualização do histograma dos dados gerados e sua comparação gráfica com a função densidade da distribuição de probabilidade teórica selecionada;
- Geração de um resumo estatístico detalhado baseado nas amostras sintéticas geradas;
- Aplicação do teste de *distribution fitting* “*Kolmogorov-Smirnov*” nas amostras geradas;

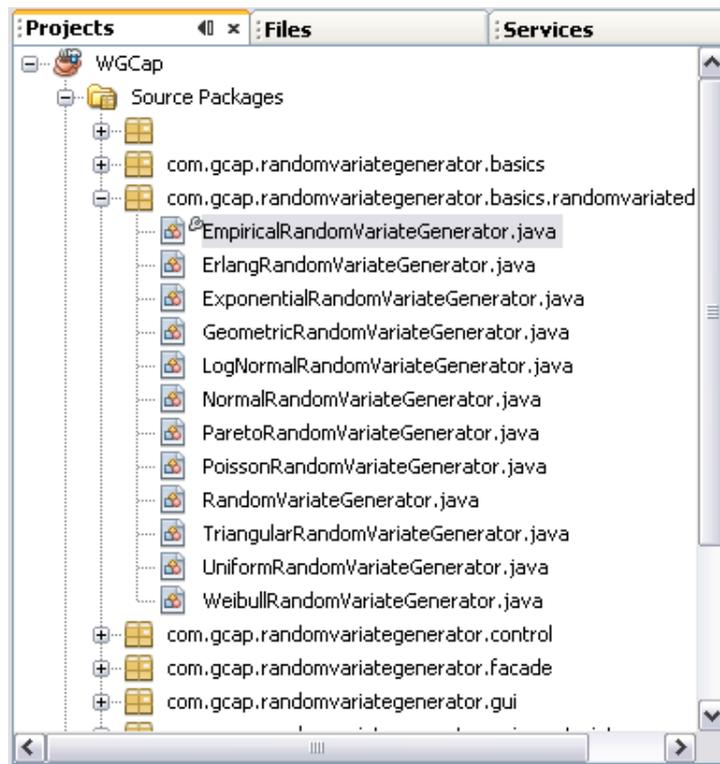


Figura 5.5: NetBeans IDE - Classes implementadas pelo *kernel*.

- Armazenamento dos valores das amostras e resumos estatísticos em arquivos “.txt” com o intuito de utilizá-los posteriormente.

A Figura 5.6 ilustra um *screenshot* do *kernel* desenvolvido para o gerador de carga, durante o processo de geração de variáveis aleatórias. Neste exemplo, é exibida a primeira aba da ferramenta (“Data Generator”), onde foram geradas 100.000 amostras baseadas na distribuição Normal com média igual a 10 e variância igual 1.

No lado direito da Figura 5.6, pode-se observar todas as características que definem o comportamento dos valores que serão gerados pela aplicação. O primeiro *frame* é responsável por definir a distribuição de probabilidade utilizada, além dos valores máximo e mínimo que representam os limites do eixo X do gráfico da função densidade de probabilidade desenhada. Caso seja selecionada a opção *AutoRange*, esses valores são gerados automaticamente, de maneira que o gráfico seja desenhado com o melhor ajuste possível na tela.

O *frame* seguinte é responsável pela definição dos valores de cada parâmetro das distribuições de probabilidade selecionadas. Neste exemplo são mostrados os parâmetros

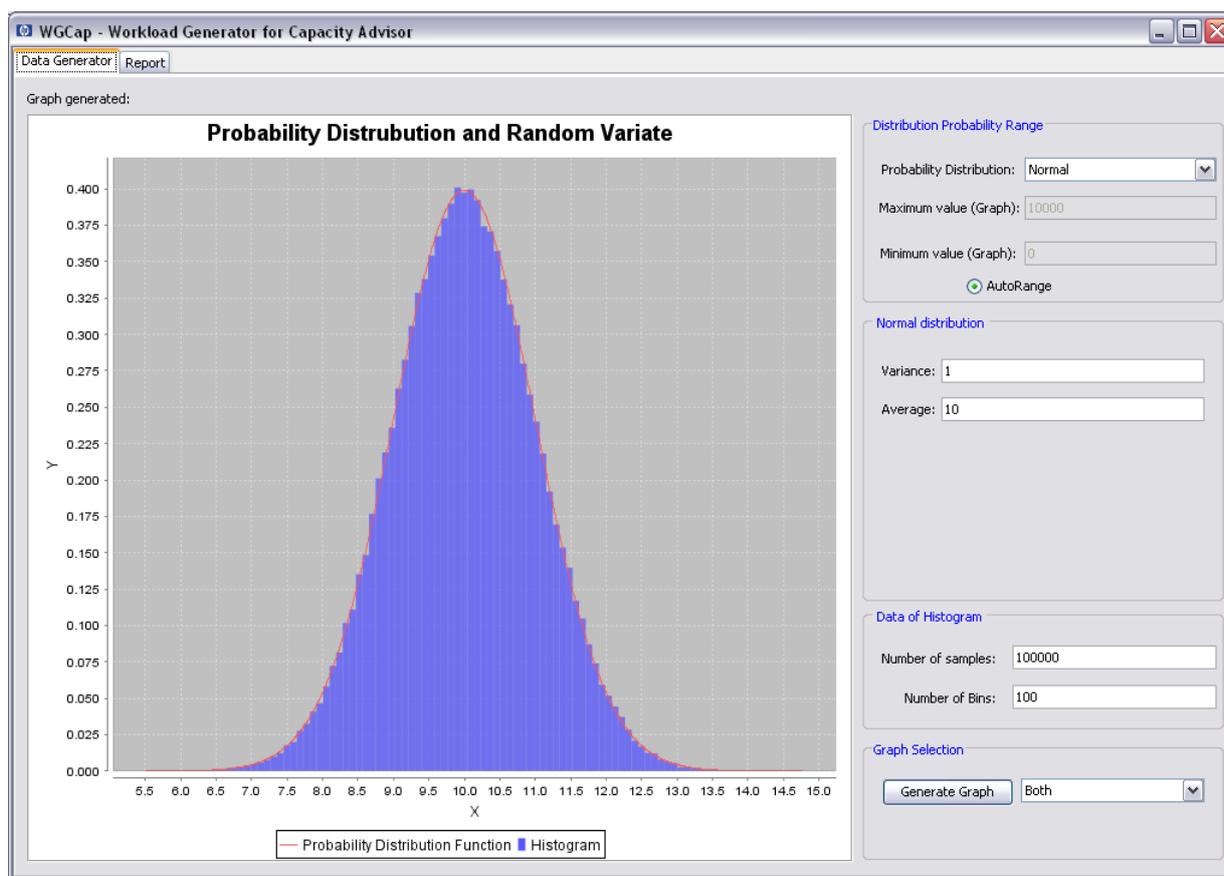


Figura 5.6: Screenshot do kernel - Geração de *random variates*.

da distribuição Normal: média (*average*) e variância (*variance*), no entanto, se for escolhida outra distribuição de probabilidade, o *frame* será atualizado e serão exibidos os parâmetros da nova distribuição. Caso seja selecionada a distribuição Empírica, surgirá um único parâmetro, com o objetivo de permitir realização do *upload* de um arquivo texto contendo um conjunto de valores que representem um determinado comportamento, a partir daí, serão gerados valores aleatórios com o mesmo comportamento observado.

O terceiro *frame* é responsável pela definição do número de amostras (*samples*) a serem geradas pelo *kernel*, além do número de *bins*, que representarão o número de classes que farão parte do histograma referente aos valores gerados. Neste exemplo, foram definidas 100.000 amostras e 100 *bins*.

O último *frame* é responsável pela escolha da representação gráfica a ser exibida pela ferramenta. Essa representação poderá ser feita de três formas diferentes: Geração gráfica apenas da função densidade de probabilidade referente a distribuição teórica selecionada, através da seleção da opção "*Probability Distribution*"; geração gráfica apenas

do histograma dos dados gerados, através da seleção da opção “*Random Variates*”; ou a geração de ambos os gráficos, com o intuito de compará-los, através da seleção da opção “*Both*”. Os gráficos gerados podem ser observados na Figura 5.6. Neste exemplo, foram criados ambos os gráficos, o primeiro referente a função densidade de probabilidade da distribuição normal, e o segundo representando o histograma dos dados gerados sinteticamente.

A Figura 5.7 mostra o *screenshot* da segunda aba do *kernel* (“*Report*”), onde é ilustrado o resumo estatístico detalhado das 100.000 amostras geradas com base na distribuição normal. Durante esta etapa, são calculadas diversas medidas estatísticas importantes, entre as quais podemos destacar: Média, variância, desvio padrão, mediana, valores máximo e mínimo, intervalo interquartil, *skewness* e *kurtosis*.

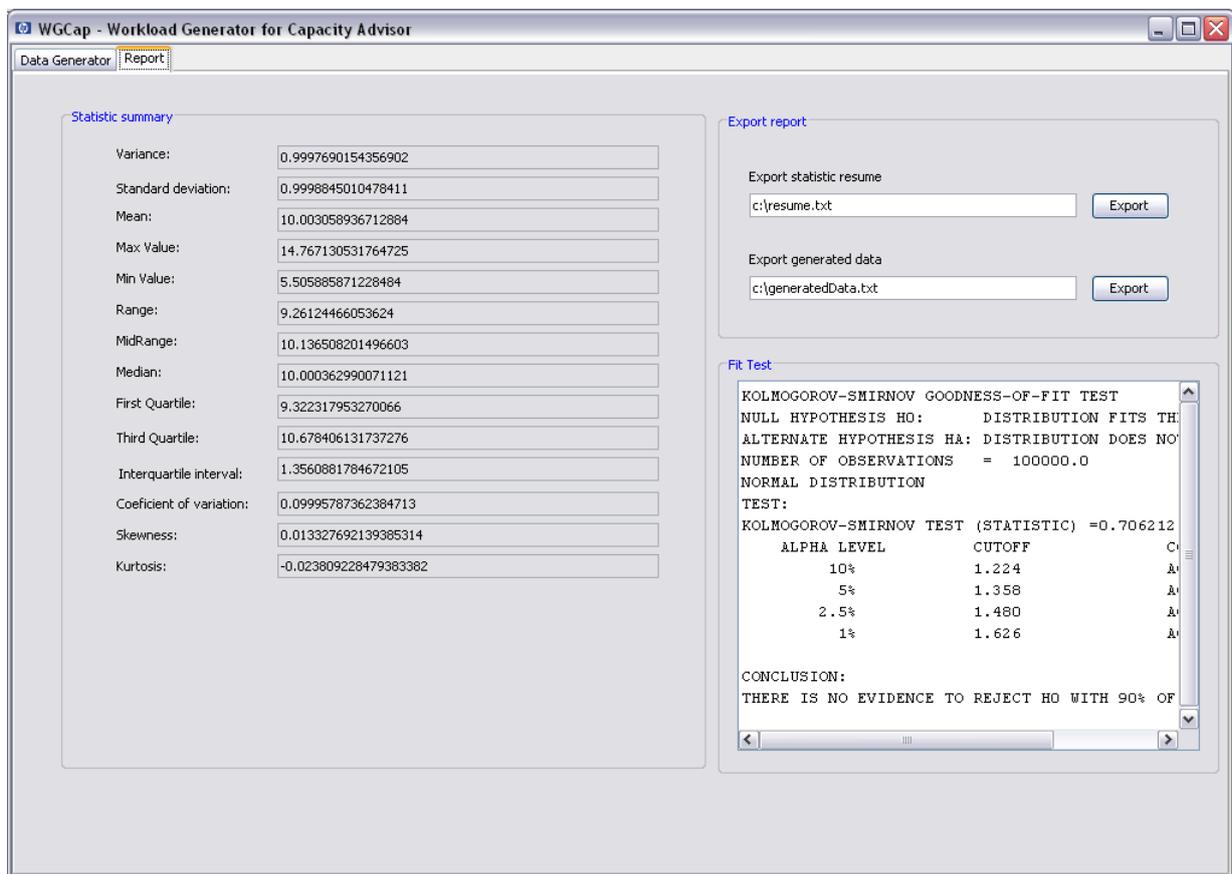


Figura 5.7: *Screenshot* do *kernel* - Resumo estatístico.

Além da visualização do resumo estatístico, a ferramenta permite que, tanto os dados gerados quanto o resumo estatístico, sejam armazenados e salvos em arquivos texto. No primeiro *frame* localizado ao lado direito da Figura 5.7 (“*Export report*”), pode-se

observar dois botões com a função de salvar os valores gerados pela ferramenta (“*Export generated data*”) e seu respectivo resumo estatístico (“*Export statistic resume*”).

Outra característica importante observada é o resultado do teste de *distribution fitting* “*Kolmogorov-Smirnov*” aplicado às amostras geradas (ver Figura 5.7). Este teste é utilizado para verificar se os valores gerados realmente correspondem a distribuição de probabilidade teórica. Como pode ser observado, as amostras geradas neste exemplo foram consideradas válidas.

5.4 WGCAP - WORKLOAD GENERATOR FOR CAPACITY ADVISOR

Após a implementação e validação do *kernel*, finalmente foi desenvolvido o gerador de *traces* sintéticos para o *Capacity Advisor*, intitulado **WGCap - Workload Generator for Capacity Advisor**. Como já foi observado, o principal objetivo da ferramenta, e foco deste trabalho, é a geração de *traces* sintéticos com o intuito de simular o consumo de recursos computacionais, tais como processador, memória, disco e rede, para que sejam importados ao *Capacity Advisor* através de arquivos com o formato CSV. Estes valores são gerados com base em *traces* reais ou em um conjunto significativo de distribuições de probabilidade e resumos estatísticos, através do utilização do *kernel*, que por sua vez, está integrado ao gerador de carga.

O WGCap é composto por dois módulos: O *módulo de geração de carga*, responsável pela geração de *traces* baseados em distribuições de probabilidade ou resumos estatísticos; e o *módulo de geração de resumos estatísticos*, responsável por extrair informações contidas em *traces* reais e gerar o seu respectivo resumo estatístico. A Figura 5.8 ilustra um *screenshot* com a tela inicial do WGCap, onde podem ser selecionados um dos dois módulos da ferramenta. O módulo de geração de carga é representado pela opção “*Generate Synthetic Traces*”, enquanto o módulo de geração de resumos estatísticos é representado pela opção “*Generate Statistical Summary*”.

5.4.1 Módulo de geração de carga

Caso seja selecionada a opção “*Generate Synthetic Traces*”, será iniciado o *módulo de geração de carga*, responsável pela geração dos *traces* sintéticos para o *Capacity Advisor*. É importante ressaltar que as atividades deste módulo executam na prática o processo de geração de carga descrito na Seção 5.2.

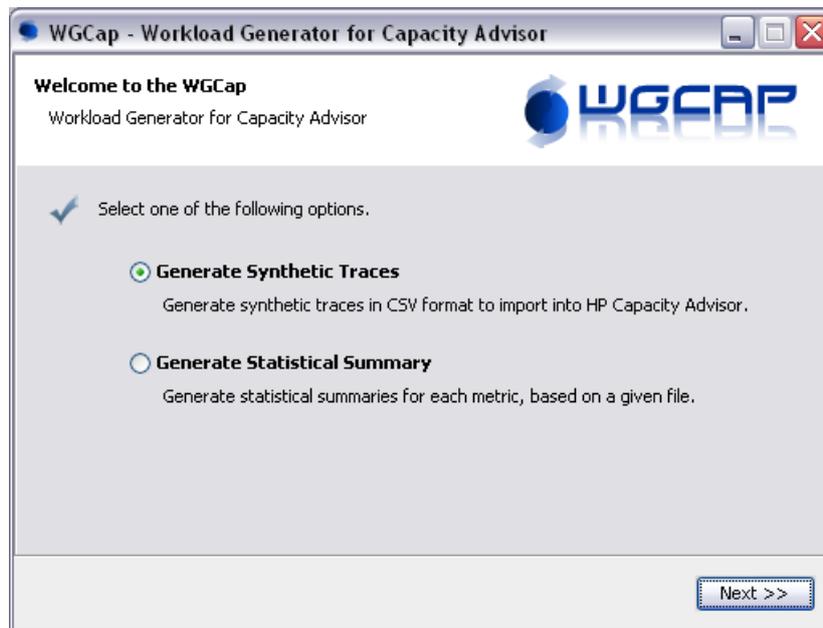


Figura 5.8: *Screenshot* do WGCap - Tela inicial.

A primeira tela desta atividade é ilustrada na Figura 5.9. Esta etapa é responsável pela definição das propriedades do *trace*, onde deverão ser definidas as principais características do ambiente a ser simulado, para que posteriormente possam ser incluídas no cabeçalho (*header*) do *trace* sintético.

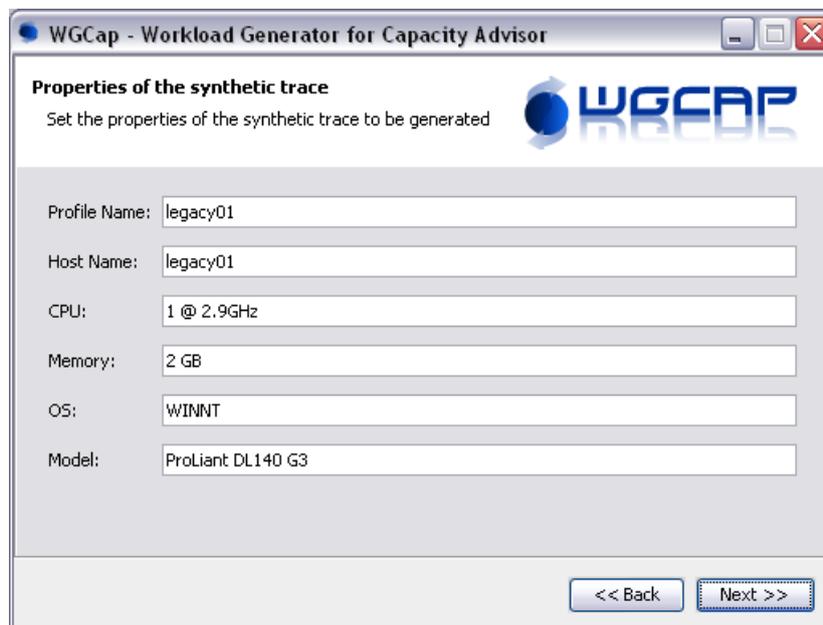


Figura 5.9: *Screenshot* do WGCap - Propriedades do *trace* sintético.

A descrição de cada um dos campos a ser preenchido durante esta etapa é mostrada a seguir:

#Profile: *name* - Nome definido para identificar o *Profile* (Identificação) do *trace* gerado;

#Host: *hostname* - Nome definido para identificar o sistema (*hostname*) simulado;

#CPU: *CPU_count@CPU_speedGHz* - Número de processadores (*CPU_count*) do sistema e as suas respectivas frequências de operação em *gigahertz*(*CPU_speedGHz*);

#Memory: *MEM_sizeMB* - Quantidade de memória RAM pertencente ao sistema em *Megabytes* (*MEM_sizeMB*);

#OS: *platform* - Sistema Operacional (OS) da plataforma a ser simulada;

#Model: *model* - Modelo (*model*) do *hardware* em que o sistema está configurado.

Após o preenchimento de todos os campos referentes às propriedades do *trace*, é iniciada a etapa de definição do período das amostras a serem geradas. A Figura 5.10 ilustra esta atividade, onde é definido o número de amostras geradas pela ferramenta, além da representação do período referente a data inicial e final da geração dos *traces* sintéticos. Conforme pode ser observado, a definição do período utilizado deverá ocorrer através da escolha de uma das seguintes opções: “*Define Period*” e “*Define Number of Samples*”. Caso seja selecionada a primeira opção, o número total de amostras é gerado automaticamente com base nas datas fornecidas; caso seja selecionada a segunda opção, a data final da simulação da coleta também é gerada automaticamente, com base no número de amostras fornecidas. Em ambos os casos, os valores gerados são incrementados em intervalos de cinco minutos, até que sejam totalizados o número de amostras. As datas fornecidas nesta etapa deverão obedecer ao seguinte formato: “DD/MM/YYYY hh:mm:ss”, onde YYYY representa o ano, MM representa o mês (de 01 a 12), DD representa o dia (de 01 a 31), hh representa as horas (de 00 a 23), mm representa os minutos (de 00 a 59) e ss representa os segundos (de 00 a 59). Posteriormente, estes períodos serão convertidos e incluídos em um arquivo CSV com a formatação utilizada pelo *Capacity Advisor*: “YYYYMMDDhhmm”.

Após a definição do período em que as amostras irão trabalhar, é iniciada a etapa de seleção das métricas referentes aos recursos computacionais que serão utilizados no *trace*

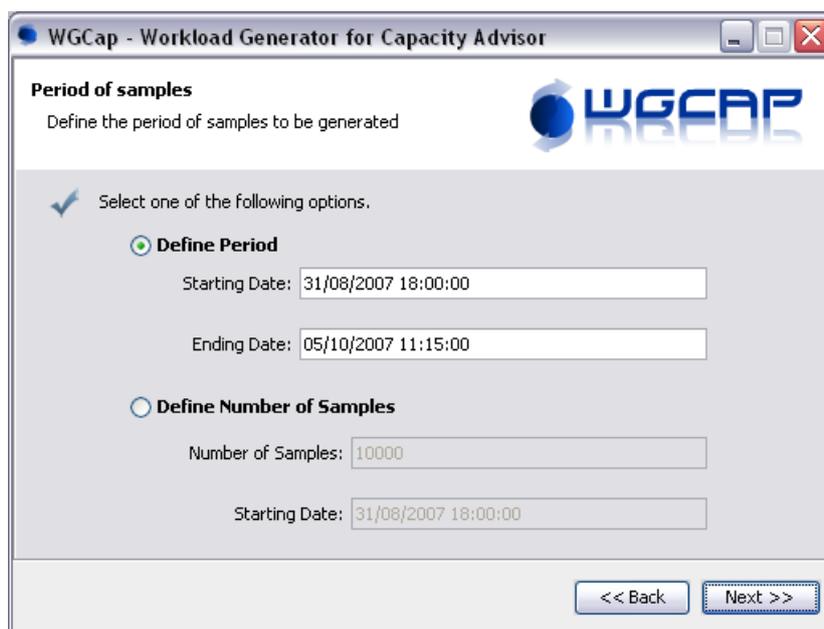


Figura 5.10: *Screenshot* do WGCap - Período das amostras.

sintético. A Figura 5.11 ilustra esta etapa, onde deve ser selecionada pelo menos uma das seguinte métricas disponíveis:

- *CPU Utilization* - CPU_UTIL;
- *Disk Utilization* - DISK_UTIL;
- *Memory Utilization* - MEM_UTIL;
- *Network Utilization* - NET_UTIL.

Uma vez selecionadas as métricas que farão parte do *trace* sintético, é iniciada a seleção da metodologia de geração de carga utilizada para a geração dos valores referentes a cada recurso. A Figura 5.12 ilustra um *screenshot* desta etapa, onde deve ser selecionado um dos dois métodos de geração de *traces* desenvolvidos nesta pesquisa: A geração de *traces* baseados em distribuições de probabilidade (*Probability distributions*) ou a geração de *traces* baseados em resumos estatísticos (*Statistical summaries*).

Caso seja escolhida a primeira opção, é iniciado o processo de seleção das distribuições de probabilidades utilizadas para cada uma das métricas selecionadas. A Figura 5.13 ilustra o *screenshot* desta etapa, onde podem ser gerados valores baseados em qualquer uma das distribuições teóricas existentes ou baseados em *traces* reais fornecidos pelo usuário,

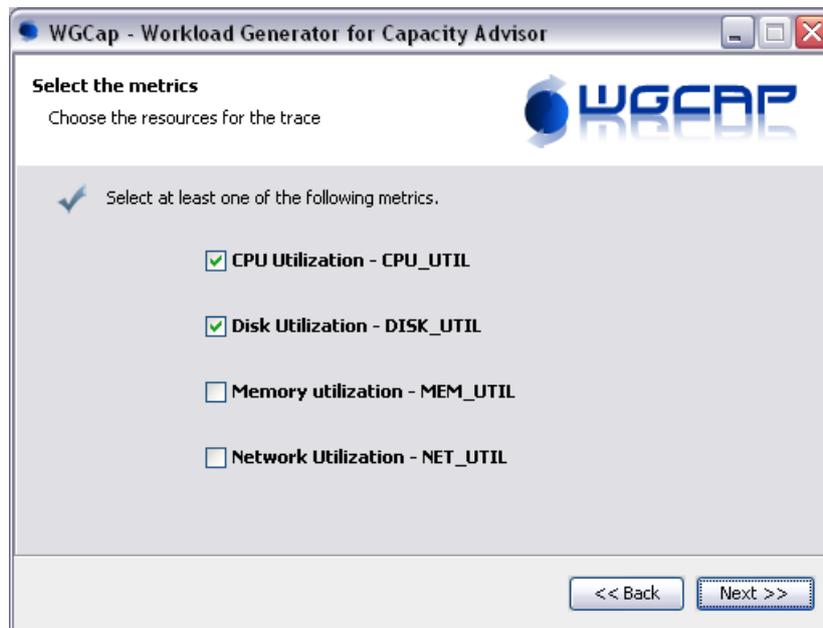


Figura 5.11: *Screenshot* do WGCAP - Seleção das métricas.

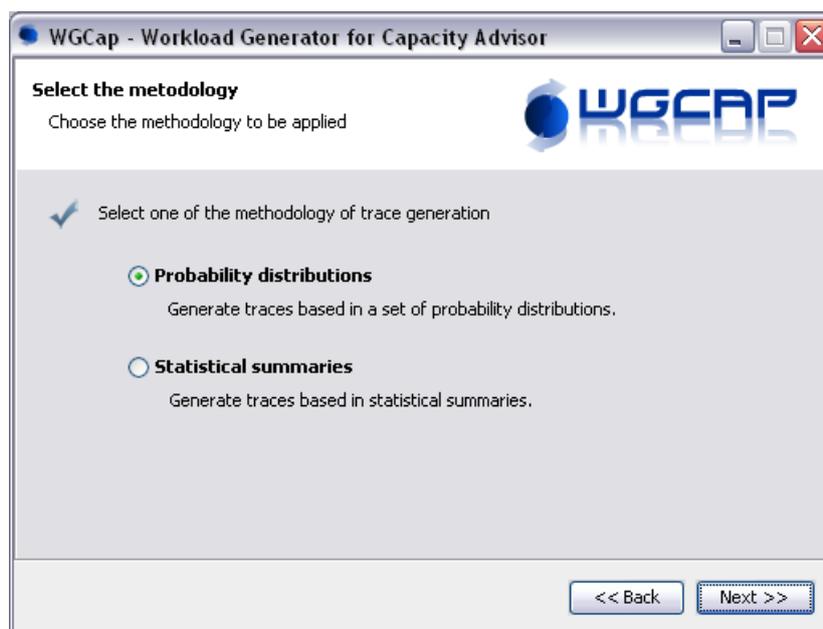


Figura 5.12: *Screenshot* do WGCAP - Seleção da metodologia de geração de carga.

através da utilização da distribuição Empírica. O funcionamento desta atividade é realizado com base no *kernel* descrito anteriormente, porém com algumas particularidades. Observe que o primeiro *frame*, ao lado direito da ferramenta, refere-se a métrica atual selecionada (“*Metric Selected*”), e será atualizado automaticamente a medida que as dis-

tribuições de probabilidade forem sendo selecionadas e aplicadas. Para cada métrica é necessária a escolha de uma das distribuições disponíveis, onde devem ser definidos os valores de cada parâmetro e o número de *bins* responsáveis por representar a quantidade de classes do histograma referentes aos valores gerados. Finalmente, para que os valores sejam gerados, o botão “*Generate Graph*” deve ser pressionado.

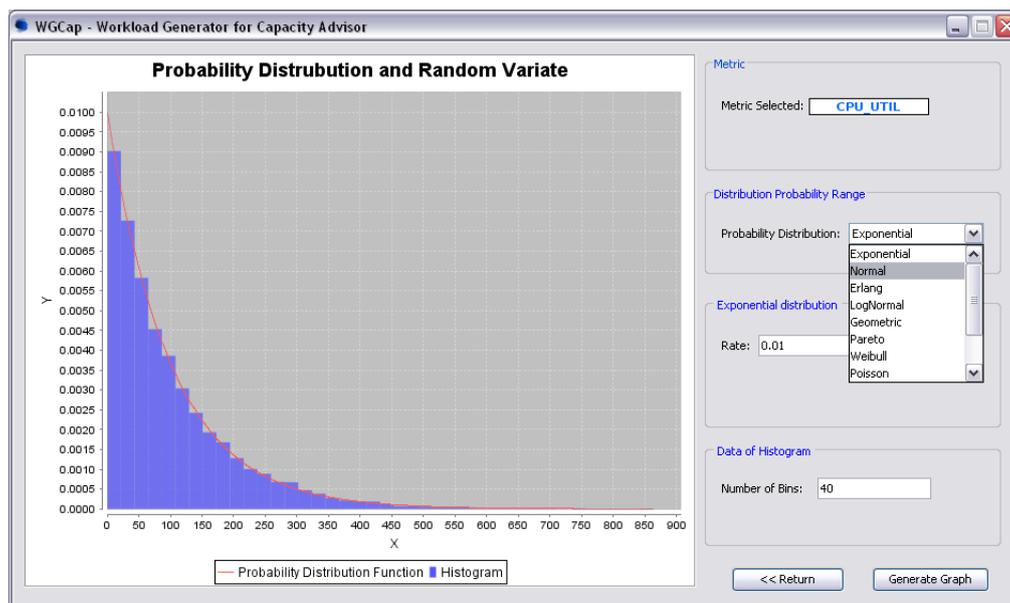


Figura 5.13: *Screenshot* do WGCap - Seleção da distribuição utilizada.

Caso seja escolhida a segunda opção, é iniciado o processo de definição dos resumos estatístico para cada uma das métricas selecionadas. Durante esta etapa podem ser gerados valores a partir de medidas estatísticas, tais como média, variância, valores máximo e mínimo, mediana, *skewness* e *kurtosis*, onde são indicadas as distribuições de probabilidade mais adequadas com base nos valores fornecidos. O *screenshot* desta atividade é ilustrado na Figura 5.14. Para cada métrica escolhida, é necessário o preenchimento do maior número de estatísticas possíveis para que possam ser definidas distribuições de probabilidade compatíveis com estes valores. Caso as estatísticas fornecidas não sejam suficientes, nenhuma distribuição de probabilidade poderá ser indicada e será solicitado que sejam introduzidos novos valores. Após o preenchimento das estatísticas, é exibida a mesma tela mostrada na Figura 5.13, no entanto só estarão disponibilizadas as distribuições de probabilidade que sejam compatíveis com as estatísticas fornecidas.

Após a seleção da metodologia de geração de carga e a definição das distribuições de probabilidade que representarão as métricas selecionadas, finalmente os *traces* sintéticos podem ser gerados e armazenados em um arquivo CSV para que possam ser importados

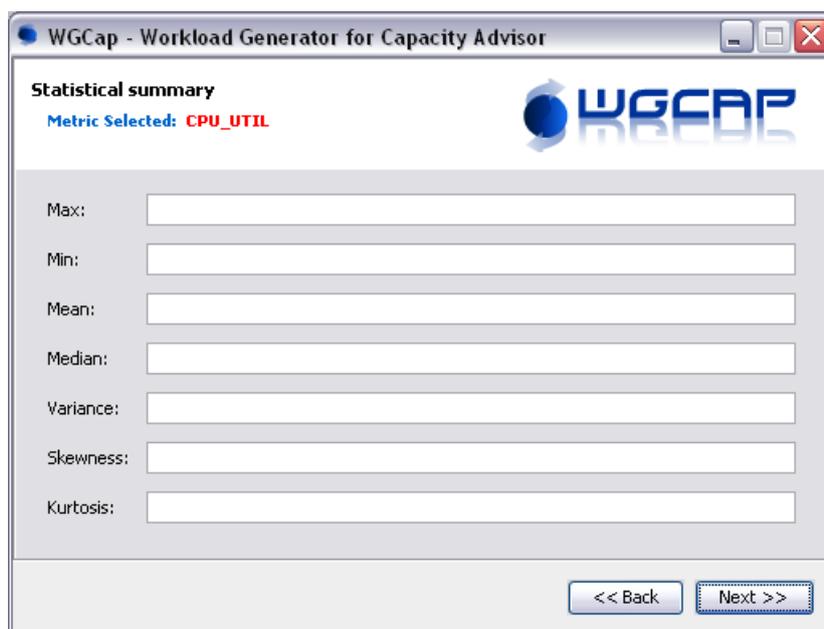


Figura 5.14: Screenshot do WGCAP - Definição do resumo estatístico.

ao *Capacity Advisor*. A Figura 5.15 ilustra um *screenshot* da última tela, responsável por indicar o andamento do processo enquanto o arquivo com os *traces* é gerado.

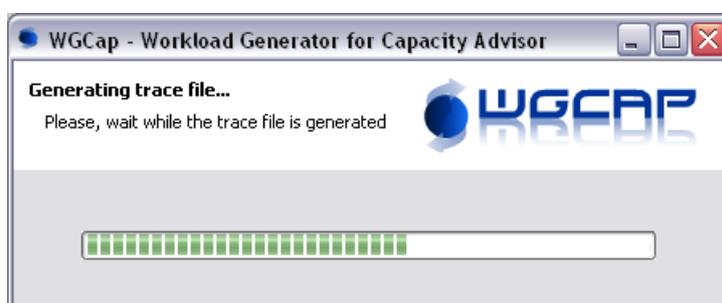


Figura 5.15: Screenshot do WGCAP - Geração do *trace* sintético.

5.4.2 Módulo de geração de resumos estatísticos

Caso seja selecionada a opção “*Generate Statistical Summary*”, é iniciado o *módulo de geração de resumos estatísticos*, cuja função é retirar informações contidas em arquivos contendo *traces* reais e gerar o seu respectivo resumo estatístico. Posteriormente, as estatísticas geradas podem ser utilizadas pelo módulo de geração de carga para a geração de valores baseados em resumos estatísticos. A utilização deste módulo deverá ser realizada quando não for possível a disponibilização de *traces* reais por parte dos administradores

de sistema. Neste caso, a ferramenta poderá ser fornecida, para que os próprios administradores gerem as estatísticas dos traces reais e as disponibilizem para a geração dos traces sintéticos.

A Figura 5.16 ilustra o *screenshot* desta etapa. Observe que para cada métrica selecionada deve ser realizado o *upload* de um arquivo contendo o *trace* referente ao recurso computacional escolhido. Em seguida é gerado um arquivo texto para cada métrica escolhida, contendo um resumo estatístico completo para que possa ser usado posteriormente.

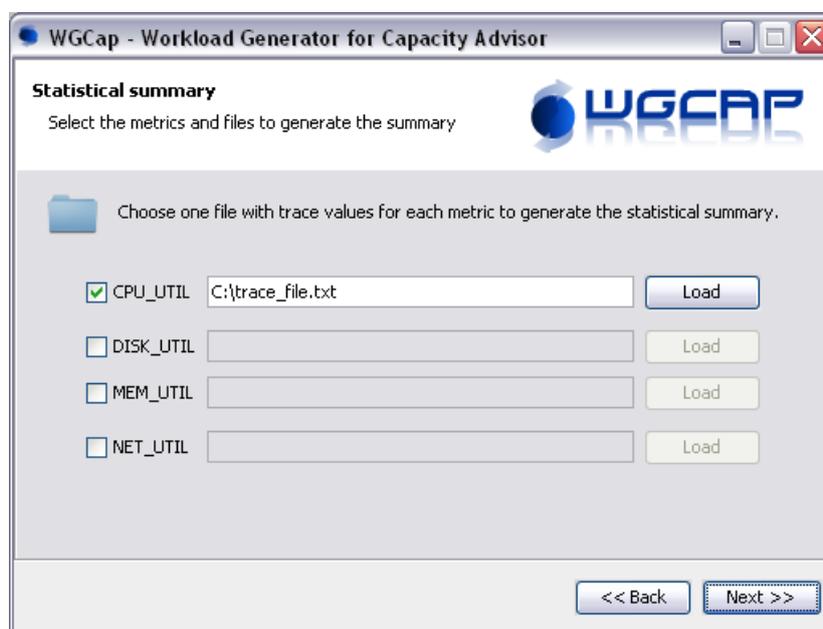


Figura 5.16: *Screenshot* do WGCAP - Geração de resumos estatísticos.

5.5 SUMÁRIO

Este capítulo apresentou utilização do ferramental desenvolvido para a geração de *traces* sintéticos para o *Capacity Advisor*. Inicialmente, foram descritos os principais cenários onde o gerador de carga pode ser aplicado, além das etapas executadas durante o processo de geração do *trace* sintético. Em seguida, foi mostrada a interface e a utilização do *kernel* desenvolvido, cuja função é gerar valores aleatórios baseados em diversas distribuições de probabilidade. Por último, foi apresentado o gerador de carga desenvolvido, intitulado WGCAP (*Workload Generator for Capacity Advisor*), onde foi descrito o seu funcionamento e todas as etapas executadas durante a geração de *traces* sintéticos representando o consumo de recursos computacionais (processador, disco, memória e rede)

para serem importados ao *Capacity Advisor*.

CAPÍTULO 6

ESTUDOS DE CASO

Neste capítulo, são apresentados os estudos de caso realizados com o intuito de avaliar o WGCap. O principal objetivo desta etapa é a aplicação do gerador desenvolvido em algumas situações que envolvam todos os cenários de geração de carga suportados pela ferramenta, mostrados na Seção 4.3. Serão gerados *traces* sintéticos baseados em *traces* reais, distribuições de probabilidade e resumos estatísticos, representando o consumo de todos os recursos computacionais disponibilizados pelo WGCap. Posteriormente, os arquivos CSV gerados serão importados para o *Capacity Advisor* e os resultados obtidos serão analisados.

6.1 ESTUDO DE CASO - *RETRACE*

O primeiro estudo de caso realizado refere-se ao Cenário 01 mostrado na Seção 4.3: “A geração de *traces* sintéticos baseados em *traces* reais”. Durante esta etapa foram realizadas coletas de *traces* em sistemas reais em produção, com a representação do consumo de cada recurso da infraestrutura utilizada. Em seguida, os *traces* reais foram importados ao WGCap e foram gerados seus respectivos *traces* sintéticos. Este estudo foi dividido em duas etapas: “A geração de um *trace* sintético baseado em um *trace* real utilizando o mesmo número de amostras” e “a geração de um *trace* sintético baseado em um *trace* real utilizando um número maior de amostras”. Estas etapas são detalhadas nas próximas subseções.

6.1.1 Geração de um *trace* sintético baseado em um *trace* real utilizando o mesmo número de amostras

Este estudo de caso mostra a geração de um *trace* sintético com amostras da métrica DISK_UTIL (utilização de disco rígido) baseado em amostras retiradas de um sistema real em produção. O principal objetivo desta etapa é a geração de valores sintéticos utilizando o mesmo número de amostras do *trace* real coletado. Estas amostras foram

coletadas e armazenadas através do *Utilization Provider* e, posteriormente, exportadas pelo *Capacity Advisor* como *traces* no formato CSV. As características do *trace* representando o sistema real utilizado foram definidas pelo seguinte cabeçalho (*profile header*):

```
#Profile: legacy11
#Host: legacy11
#CPU: 2 @ 2.7GHz
#Memory: 2 GB
#OS: WINNT
#Model: ProLiant DL140 G3
```

Neste *trace* foram coletadas 8.928 amostras com valores relativos ao consumo de cada recurso computacional utilizado pela infraestrutura. O identificador da data de coleta inicial (*timestamp* inicial), referente a primeira amostra coletada do sistema, é definido como segue: 200708311800 - YYYYMMDDhhmm (ver Seção 3.2.3). As amostras subsequentes foram coletadas em períodos com intervalos de 5 minutos (incrementos de 300 segundos).

Para a geração do *trace* sintético foram realizados os passos descritos no Capítulo 5.4, através da utilização do WGCap. Inicialmente, foram definidas as propriedades do *trace* sintético a ser gerado, onde foram utilizadas as mesmas características do sistema real definidas pelo *profile* “*legacy11*”. Em seguida, foi definido o *timestamp* inicial em que a simulação das amostras geradas terão início, durante esta etapa foi utilizado o mesmo *timestamp* inicial da amostra real coletada: 200708311800. Além do *timestamp* inicial, também foi definido o número de amostras a serem geradas, onde foi selecionada a mesma quantidade de valores coletados do *trace* real, 8.928 amostras. Posteriormente, foram escolhidas as métricas referentes a cada um dos recursos computacionais utilizados, neste caso, foi selecionada apenas a métrica DISK_UTIL.

Após a definição das principais características do *trace*, foi selecionado o módulo de geração de carga e escolhido o método de geração baseado na distribuição empírica, onde foram gerados novos valores baseados nas amostras referentes a métrica DISK_UTIL, retirada do *profile* “*legacy11*”. A Figura 6.1 ilustra o *screenshot* do WGCap com a conclusão desta etapa, onde podem ser observadas as características da distribuição empírica criada

e o histograma referente aos valores sintéticos gerados.

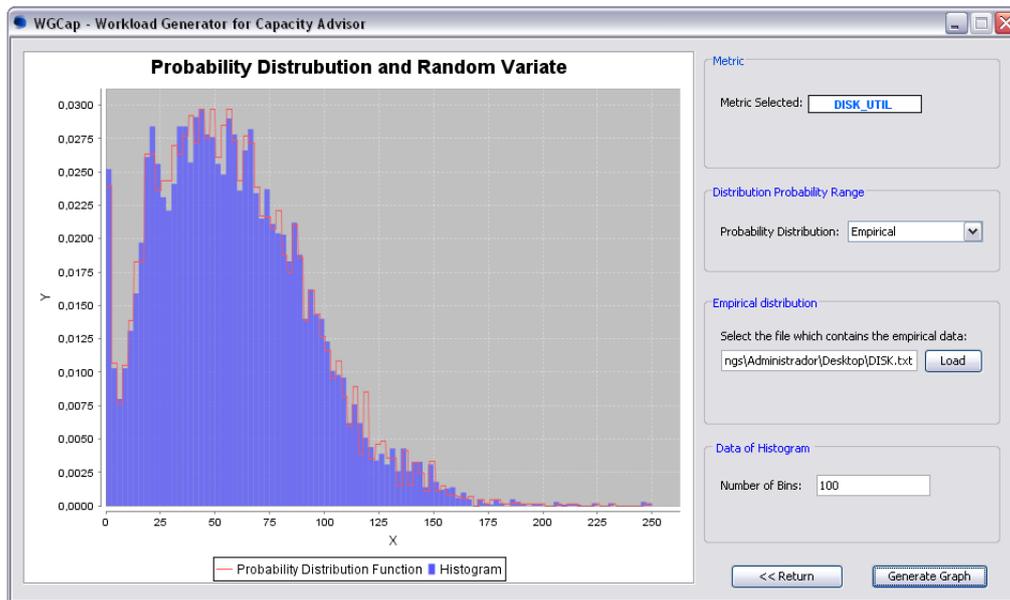


Figura 6.1: WGCap - Distribuição empírica (DISK_UTIL).

Finalmente, após a execução das etapas descritas, o *trace* sintético contendo o mesmo número de amostras do *trace* real pôde ser gerado. A Figura 6.2 ilustra os resultados comparativos com a função de distribuição acumulada de cada uma das duas amostras, real e empírica. Pode-se observar que ambas possuem comportamentos equivalentes.

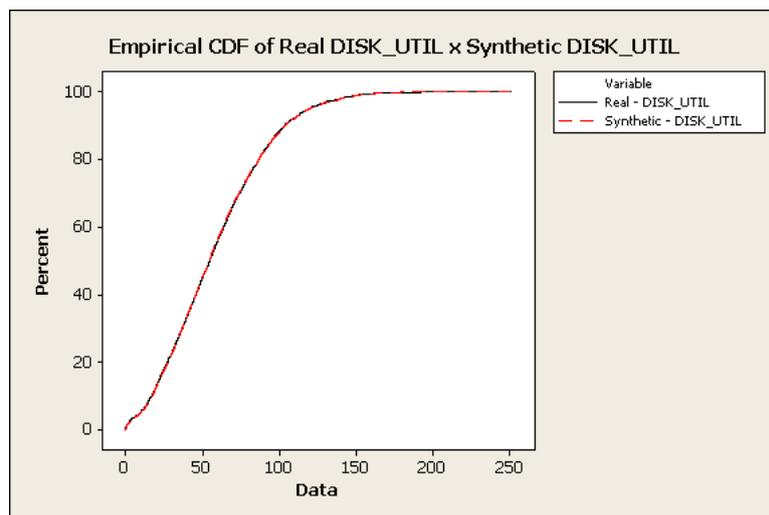


Figura 6.2: Função de distribuição acumulada - DISK_UTIL (Real) x DISK_UTIL (Sintético).

Uma vez gerado e validado, o *trace* sintético foi importado pelo o *Capacity Advisor* com sucesso. A Figura 6.3 ilustra os resultados obtidos após a inserção do *trace* na

ferramenta, onde pode ser visualizado graficamente a utilização da largura de banda do disco rígido em MB/s (*Disk I/O Capacity Utilization*) durante o período de 31/08/2007 a 03/10/2007.

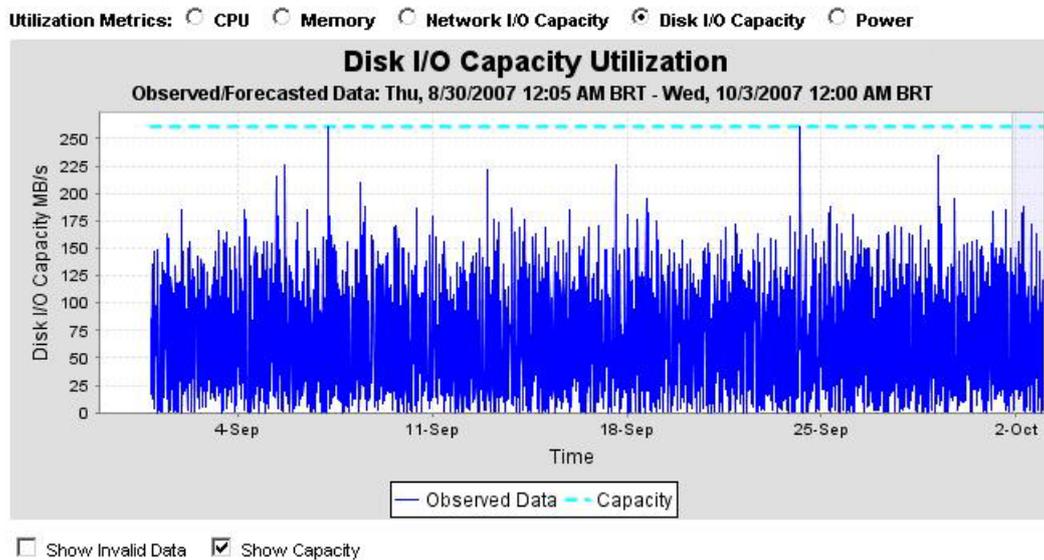


Figura 6.3: *Capacity Advisor* - Utilização da largura de banda do disco rígido em MB/s.

6.1.2 Geração de um *trace* sintético baseado em um *trace* real utilizando um número maior de amostras

Este estudo de caso mostra a geração de um *trace* sintético com amostras da métrica CPU_UTIL (utilização de CPU) baseado em amostras retiradas de um sistema real em produção, no entanto, desta vez foram gerados valores sintéticos utilizando um número maior de amostras com relação a quantidade de valores pertencentes ao *trace* real coletado, com o objetivo de simular o seu comportamento futuro. As características do *trace* representando o sistema real utilizado foram definidas pelo seguinte cabeçalho (*profile header*):

```
#Profile: legacy01
#Host: legacy01
#CPU: 1 @ 2.9GHz
#Memory: 2 GB
#OS: WINNT
```

```
#Model: ProLiant DL140 G3
```

Neste *trace*, também foram coletadas 8.928 amostras com valores relativos ao consumo de cada recurso computacional utilizado pela infraestrutura. O *timestamp* inicial (*initial timestamp*), referente a primeira amostra coletada do sistema, é definido como segue: 200708311800 (YYYYMMDDhhmm). As amostras subsequentes foram coletadas em períodos com intervalos de 5 minutos.

Para a geração do *trace* sintético foram realizados os mesmos passos descritos na subseção anterior. Inicialmente, foram definidas as propriedades do *trace* sintético a ser gerado, onde foram utilizadas as mesmas características do sistema real definidas pelo *profile* “*legacy01*”. Em seguida, foi definido o *timestamp* inicial em que a simulação das amostras geradas terão início. Como o intuito deste estudo é a geração de uma maior quantidade de *traces* sintéticos simulando um comportamento futuro do *trace* real, foi definido um novo *timestamp* inicial para representar a primeira amostra sintética. Para isso foi analisado o *timestamp* referente a última amostra do *trace* real (200710011755), e a partir dele definimos o *timestamp* inicial do *trace* sintético como segue: 200710011800. Observe que o *timestamp* inicial definido corresponde a última amostra do *trace* real com um incremento de cinco minutos, dando assim continuidade aos valores pertencentes ao *trace* real, simulando seu comportamento futuro.

Além do *timestamp* inicial, também foi definido o número de amostras a serem geradas pela ferramenta. Desta vez foram selecionadas 20.000 amostras, representando mais que o dobro do número de amostras pertencentes ao *trace* real. Posteriormente, foram escolhidas as métricas referentes a cada um dos recursos computacionais utilizados, neste caso, foi selecionada apenas a métrica CPU_UTIL.

Após a definição das principais características do *trace*, foi selecionado o módulo de geração de carga e escolhido o método de geração baseado na distribuição empírica, onde foram gerados novos valores baseados nas amostras referentes a métrica CPU_UTIL, retirada do *profile* “*legacy01*”. Finalmente, o *trace* sintético baseado em um *trace* real contendo um número maior de amostras pôde ser gerado. A Figura 6.4 ilustra os resultados comparativos com a função de distribuição acumulada de cada uma das duas amostras, real e empírica. Pode-se observar que ambas possuem comportamentos equivalentes.

Uma vez gerado e validado, o *trace* sintético foi importado pelo *Capacity Advisor* juntamente com o *trace* real. A Figura 6.5 ilustra os resultados obtidos após a inserção do *trace* real na ferramenta, enquanto a Figura 6.6 mostra os resultados referentes ao *trace*

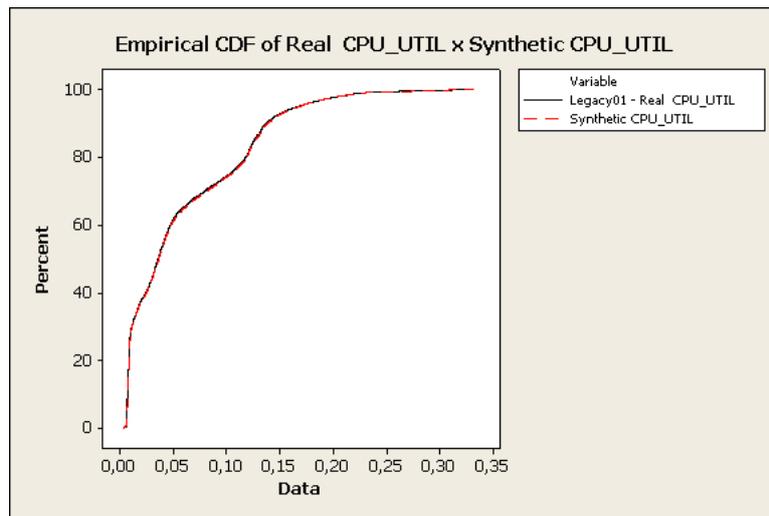


Figura 6.4: Função de distribuição acumulada - CPU_UTIL (Real) x CPU_UTIL (Sintético).

sintético, onde pode ser visualizado graficamente a utilização do CPU (*CPU Utilization*), definida pelo número de processadores utilizados durante o período de 01/10/2007 a 10/12/2007 (totalizando 20.000 amostras). Observe que o *trace* sintético gerado simula o *trace* real, mantendo suas principais características e níveis esperados de consumo do CPU.

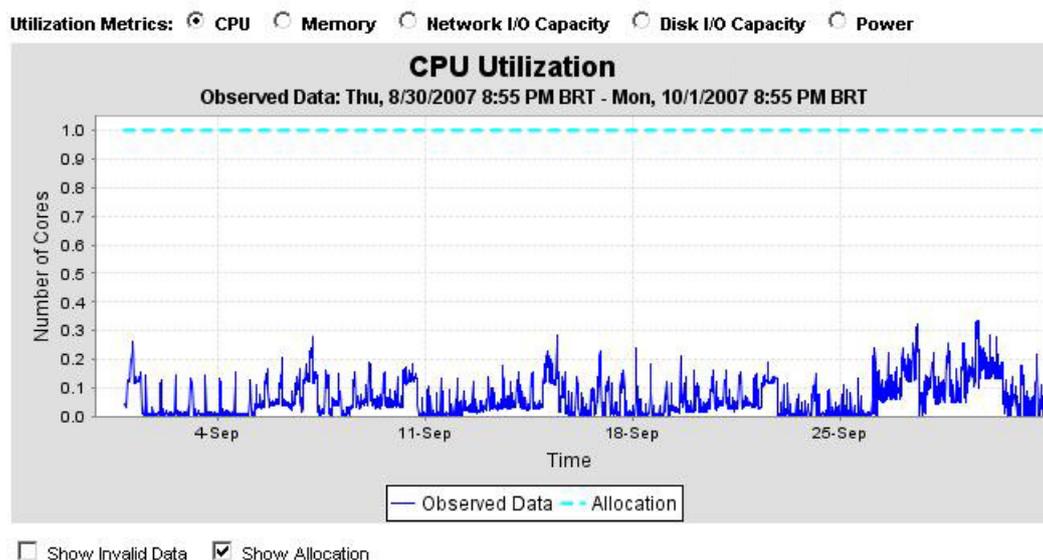


Figura 6.5: *Capacity Advisor* - Utilização de CPU (*trace* real).

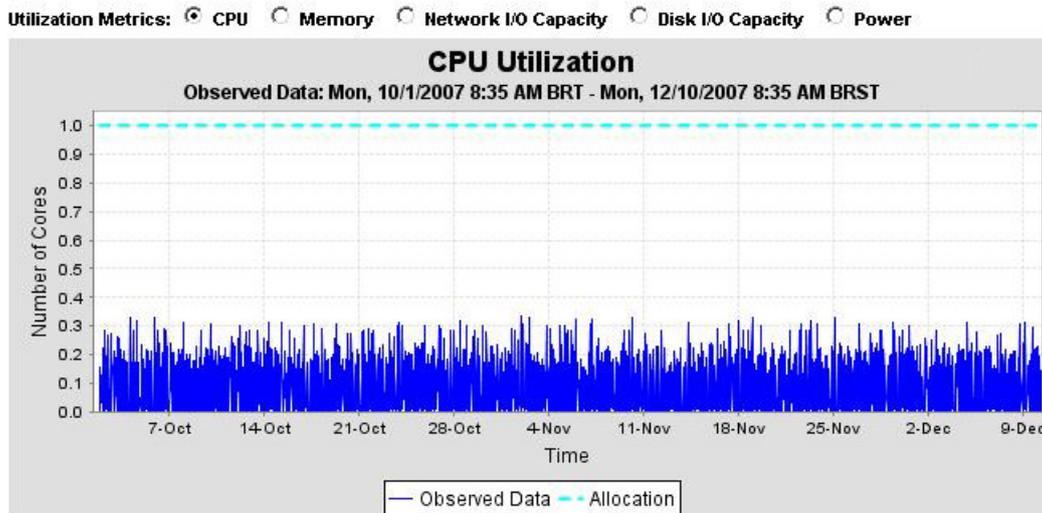


Figura 6.6: *Capacity Advisor* - Utilização de CPU (*trace* sintético).

6.2 ESTUDO DE CASO - *STATISTIC TRACE & STATISTIC RETRACE*

O segundo estudo de caso realizado refere-se aos Cenários 02 e 03 mostrados na Seção 4.3: “A geração de *traces* sintéticos baseados em distribuições de probabilidade e resumos estatísticos” e “a geração de resumos estatísticos e *traces* sintéticos baseados em *traces* reais”. O principal objetivo deste etapa é realizar a geração de *traces* sintéticos mesmo quando um *trace* real não estiver disponível para ser importado ao WGCap, através da utilização de resumos estatísticos e um conjunto de distribuições de probabilidade para a definição do comportamento das cargas a serem geradas. Para avaliar os cenários descritos, este estudo foi dividido em duas etapas: “A geração de um *trace* sintético baseado em distribuições de probabilidade” e “a geração de um *trace* sintético baseado em resumos estatísticos”. Estas etapas são detalhadas nas próximas subseções.

6.2.1 Geração de um *trace* sintético baseado em distribuições de probabilidade

Este estudo de caso mostra a geração de um *trace* sintético com amostras da métrica MEM_UTIL (utilização de memória RAM), baseadas na distribuição de probabilidade *Normal*. O intuito deste estudo é a simulação do consumo de memória RAM em um sistema com as mesmas características do *profile legacy11* mostrado na seção anterior, onde o comportamento das cargas geradas deverá ser distribuído normalmente. As características do *trace* sintético representando este sistema foram definidas pelo seguinte

cabeçalho (*profile header*):

```
#Profile: Synthetic_legacy11
#Host: Synthetic_legacy11
#CPU: 2 @ 2.7GHz
#Memory: 2 GB
#OS: WINNT
#Model: ProLiant DL140 G3
```

Para a geração do *trace* sintético, novamente foram realizados os passos descritos no Capítulo 5.4, através da utilização do WGCap. Inicialmente, foram definidas as propriedades do *trace* sintético a ser gerado, onde foram utilizadas as características mostradas acima, definidas pelo *profile* “*Synthetic_legacy11*”. Em seguida, foi definido o *timestamp* inicial em que a simulação das amostras geradas terão início, como este campo não terá influência na geração dos valores, durante esta etapa foi utilizado o mesmo *timestamp* inicial pertencente ao *profile legacy11*: 200708311800. Além do *timestamp* inicial, também foi definido o número de amostras a serem geradas pela ferramenta, durante esta etapa foram selecionadas 10.000 amostras. Posteriormente, foram escolhidas as métricas referentes a cada um dos recursos computacionais utilizados, neste caso, foi selecionada apenas a métrica MEM_UTIL.

Após a definição das principais características do *trace*, foi selecionado o módulo de geração de carga e escolhido o método de geração baseado na distribuição de probabilidade normal. Como o sistema a ser simulado possui 2GB de memória RAM, resolvemos verificar o seu comportamento com um consumo médio de 1,2GB e uma variância de 0,035, definindo assim os parâmetros da distribuição com estes respectivos valores. A Figura 6.7 ilustra o *screenshot* do WGCap com a conclusão desta etapa, onde podem ser observadas as características da distribuição normal criada, seus parâmetros e o histograma referente aos valores sintéticos gerados.

Finalmente, após a execução das etapas descritas, o *trace* sintético contendo 10.000 amostras baseadas na distribuição normal pôde ser gerado. A Figura 6.8 ilustra o histograma dos valores gerados juntamente com a função densidade da distribuição de probabilidade normal. Pode-se observar o que comportamento dos valores gerados realmente

obedecem a distribuição normal com média igual a 1,203 e variância igual a 0,3448 ($0,1857^2$), valores muito próximos aos parâmetros definidos inicialmente pelo WGCap.

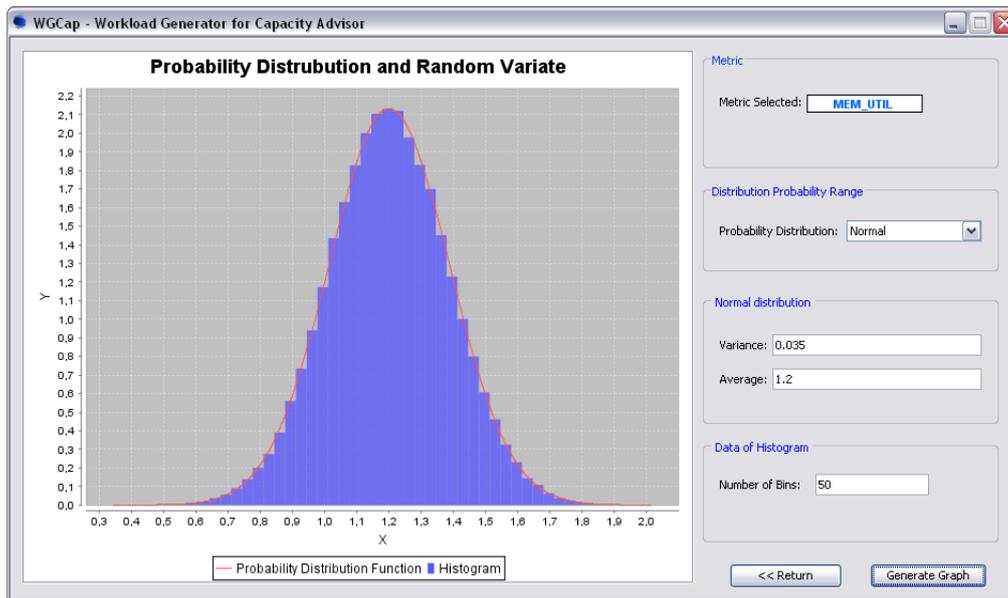


Figura 6.7: WGCap - Distribuição normal (MEM_UTIL).

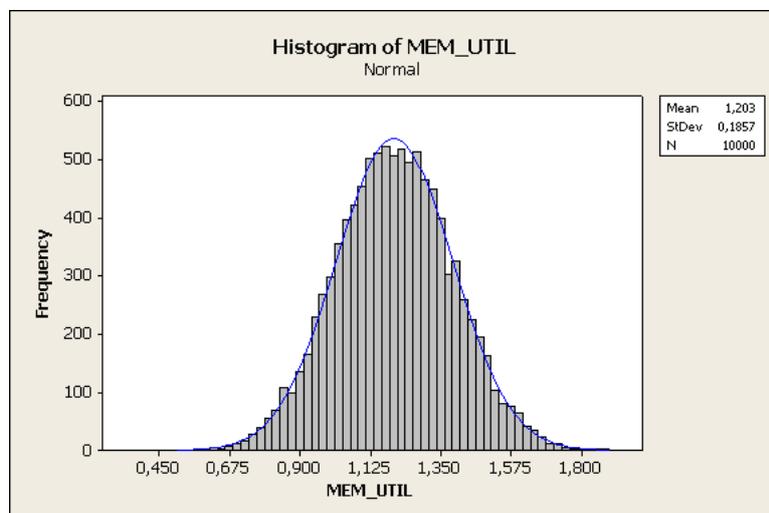


Figura 6.8: MEM_UTIL - Histograma dos valores gerados.

Uma vez gerado e validado, o *trace* sintético foi importado pelo *Capacity Advisor* com sucesso. A Figura 6.9 ilustra os resultados obtidos após a inserção do *trace* na ferramenta, onde pode ser visualizado graficamente a utilização da memória RAM em GB - *gigabytes* (*memory utilization*) durante o período de 31/08/2007 a 06/10/2007.

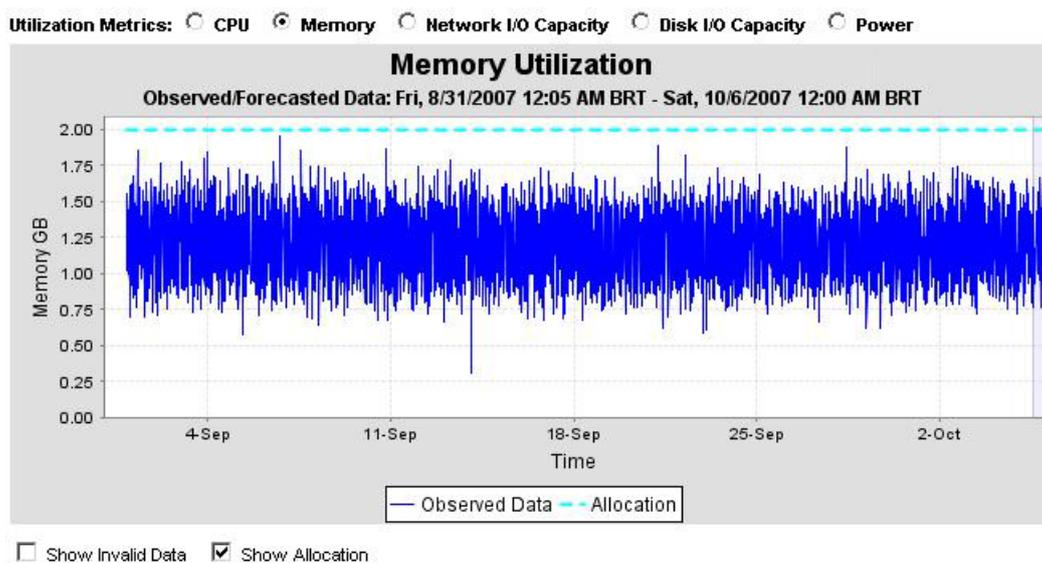


Figura 6.9: *Capacity Advisor* - Utilização de Memória RAM (Distribuição normal).

6.2.2 Geração de um *trace* sintético baseado em resumos estatísticos

Este estudo de caso mostra a geração de um *trace* sintético com amostras da métrica DISK_UTIL (utilização do disco rígido), baseadas em um resumo estatístico retirado de um sistema real em produção. Esta situação ocorre quando os administradores de sistemas não podem fornecer, ou dispor de *traces* reais para que sejam diretamente utilizados pelo gerador de carga. Neste caso, o WGCap é fornecido, para que os próprios administradores gerem as estatísticas dos *traces* reais e as disponibilizem para geração dos *traces* sintéticos. O principal objetivo deste estudo é a simulação do consumo de disco rígido através da geração de *traces* sintéticos baseados em estatísticas retiradas do *profile legacy01*, mostrado na seção anterior. As características do *trace* sintético representando este sistema foram definidas pelo seguinte cabeçalho (*profile header*):

```
#Profile: Synthetic_legacy01
#Host: Synthetic_legacy01
#CPU: 1 @ 2.9GHz
#Memory: 2 GB
#OS: WINNT
#Model: ProLiant DL140 G3
```

O primeiro passo desta atividade é a execução do módulo de geração de resumos estatísticos através do WGCap (ver Seção 5.4.2), onde deve ser inserido um arquivo com os valores referentes ao consumo de disco rígido pertencente ao *profile legacy01*, para que seja gerado seu respectivo resumo estatístico. O resultado obtido durante esta etapa é mostrado na Tabela 6.1.

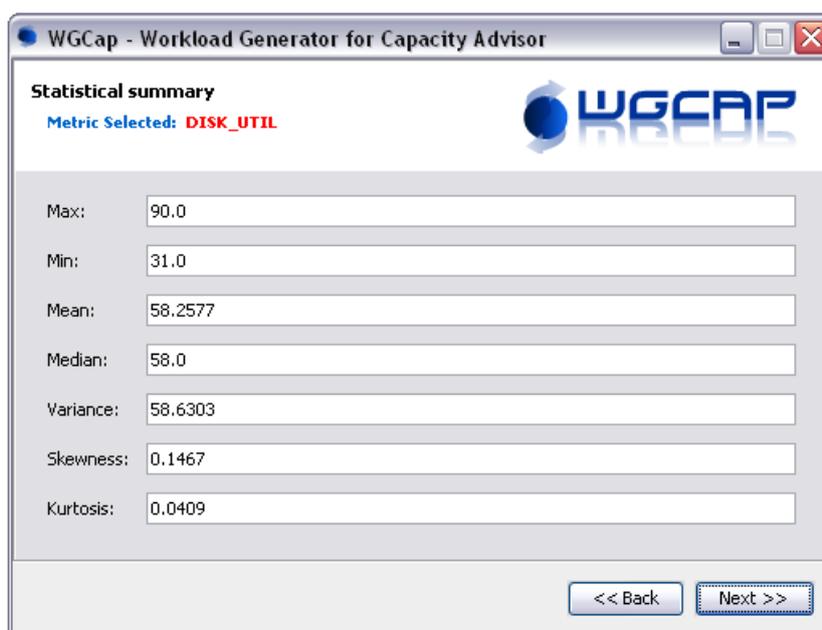
Tabela 6.1: Resumo estatístico - *profile legacy01* (DISK_UTIL).

<i>Samples</i>	8928
<i>Min. Value</i>	31,0
<i>Max. Value</i>	90,0
<i>Sum</i>	582577,0
<i>Mean</i>	58,2577
<i>Range</i>	59,0
<i>Standard Deviation</i>	7,6570
<i>Variance</i>	58,6303
<i>Midrange</i>	60,5
<i>Median</i>	58,0
<i>First Quartile</i>	53,0
<i>Third Quartile</i>	63,0
<i>Interquartile Interval</i>	10,0
<i>Kurtosis</i>	0,0409
<i>Skewness</i>	0,1467
<i>Coeficient variation</i>	0,1314

Após a geração do resumo estatístico é iniciado o processo de geração do *trace* sintético para o *Capacity Advisor*. Inicialmente, foram definidas as propriedades do *trace* a ser gerado, onde foram utilizadas as características mostradas acima, definidas pelo *profile "Synthetic_legacy01"*. Em seguida, foi definido o *timestamp* inicial em que a simulação das amostras geradas terão início, como este campo também não terá influência na geração dos valores, durante esta etapa foi utilizado o mesmo *timestamp* inicial pertencente ao *profile legacy01*: 200708311800. Além do *timestamp* inicial, também foi definido o número de amostras a serem geradas pela ferramenta, durante esta etapa foram selecionadas 10.000 amostras. Posteriormente, foram escolhidas as métricas referentes a cada um dos recursos computacionais utilizados, neste caso, foi selecionada apenas a métrica DISK_UTIL.

Uma vez definidas as principais características do *trace*, foi selecionado o módulo de geração de carga e escolhido o método de geração baseado em resumos estatísticos.

A Figura 6.10 ilustra o *screenshot* do WGCap durante a realização desta etapa, onde foram introduzidos os valores pertencentes ao resumo estatístico gerado. Após a definição destes valores, é iniciada a etapa de seleção da distribuição de probabilidade que será utilizada com base nas estatísticas fornecidas. Como todos os campos do resumo estatístico foram preenchidos, diversas distribuições de probabilidade se mostraram adequadas para a geração dos valores sintéticos, entretanto, optamos por selecionar a distribuição “triangular” pelo fato dela trabalhar com os parâmetros “média”, “valor máximo” e “valor mínimo”, situação importante quando não é possível obter um resumo estatístico completo e são apenas disponibilizados os valores médio, máximo e mínimo referentes ao consumo dos recursos selecionados. A Figura 6.11 ilustra esta etapa, onde podem ser observadas as características da distribuição triangular criada, seus parâmetros e o histograma referente aos valores sintéticos gerados.



Metric	Value
Max:	90.0
Min:	31.0
Mean:	58.2577
Median:	58.0
Variance:	58.6303
Skewness:	0.1467
Kurtosis:	0.0409

Figura 6.10: WGCap - Definição do resumo estatístico para a métrica DISK_UTIL.

O resumo estatístico das amostras geradas com base na distribuição triangular é mostrado na Tabela 6.2. Analisando o resumo estatístico das amostras geradas sinteticamente e comparando com o resumo da amostra real, observamos que a maioria das estatísticas calculadas ficaram muito próximas dos valores observados no *trace* real.

Finalmente, após a execução das etapas descritas, o *trace* sintético contendo 10.000 amostras baseadas no resumo estatístico fornecido em conjunto com a distribuição triangular, pôde ser gerado e importado pelo o *Capacity Advisor* com sucesso.

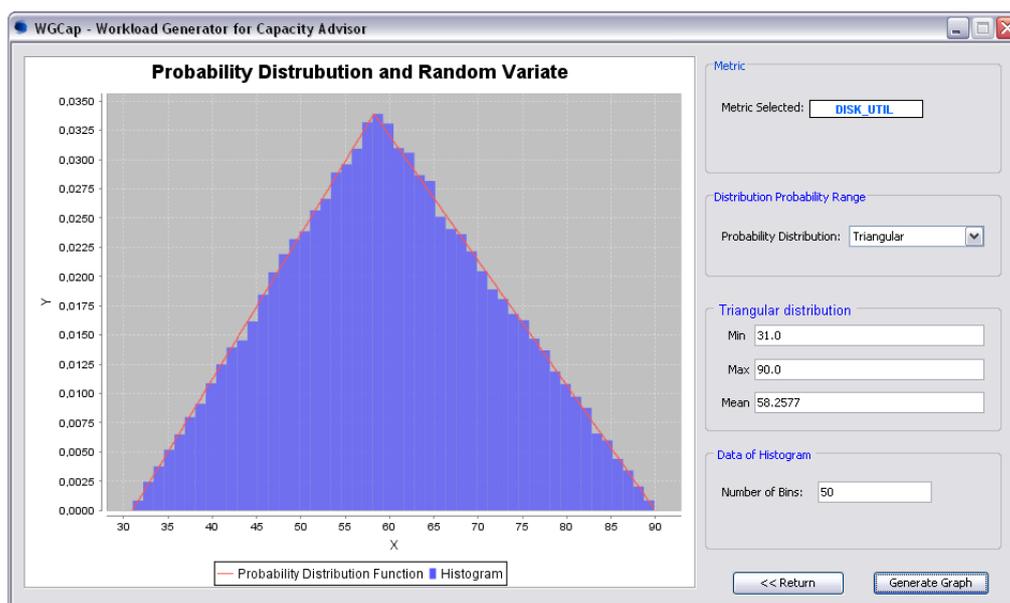


Figura 6.11: WGCap - Distribuição triangular (DISK_UTIL).

Tabela 6.2: Resumo estatístico - Distribuição triangular (DISK_UTIL).

<i>Samples</i>	10000
<i>Min. Value</i>	31,1394
<i>Max. Value</i>	89.8254
<i>Sum</i>	598680,1896
<i>Mean</i>	59,770
<i>Range</i>	58,6859
<i>Standard Deviation</i>	12,0424
<i>Variance</i>	145,0197
<i>Midrange</i>	60,4824
<i>Median</i>	59.4057
<i>First Quartile</i>	51,0810
<i>Third Quartile</i>	68,3305
<i>Interquartile Interval</i>	17,2494
<i>Kurtosis</i>	0,0594
<i>Skewness</i>	0,0749
<i>Coeficient variation</i>	0,2014

A Figura 6.12 ilustra os resultados obtidos após a inserção do *trace* na ferramenta, onde pode ser visualizado graficamente a utilização da largura de banda do disco rígido em MB/s (*Disk I/O Capacity Utilization*) durante o período de 31/08/2007 a 06/10/2007.

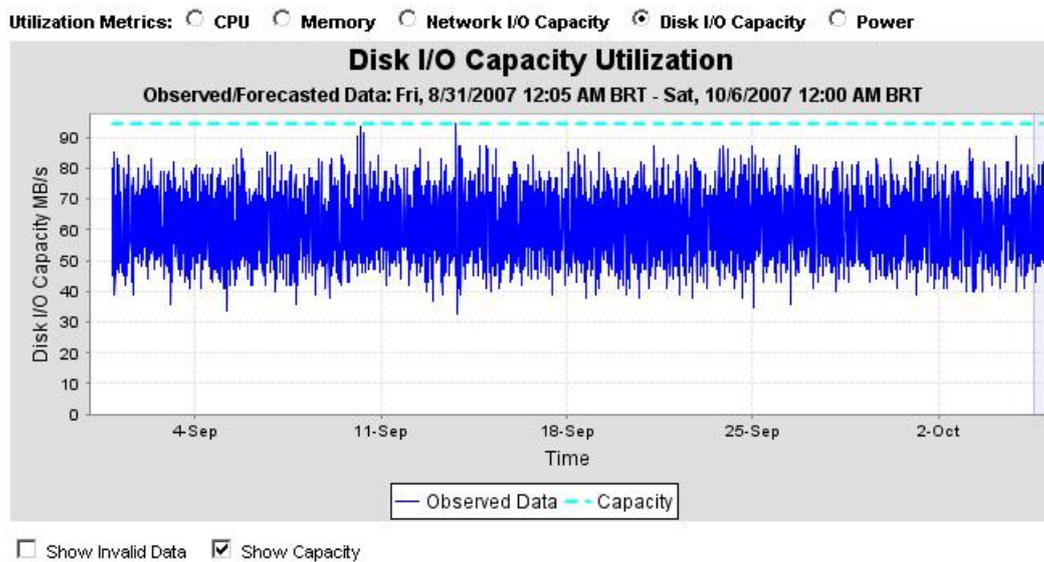


Figura 6.12: *Capacity Advisor* - Utilização da largura de banda do disco rígido em MB/s.

6.3 SUMÁRIO

Este capítulo apresentou os resultados obtidos durante a realização dos estudos de caso. Através dos estudos apresentados, foi possível avaliar e validar tanto a metodologia desenvolvida para a geração de carga quanto o ferramental desenvolvido. Durante esta etapa foram realizados estudos referentes aos cenários de geração de carga suportados pelo ferramental, mostrando na prática todas as possibilidades oferecidas pelo gerador de carga. Foram gerados diversos *traces* sintéticos baseados em *traces* reais, distribuições de probabilidade e resumos estatísticos, representando o consumo de vários recursos computacionais. Após a geração, os *traces* criados foram importados para o *Capacity Advisor* com sucesso, concluindo desta forma o objetivo da pesquisa.

CONCLUSÕES

Atualmente os sistemas de informação estão presentes em todas as atividades do nosso cotidiano, sendo aplicados nas mais diversas áreas de conhecimento, desde saúde, educação e comunicação até finanças, segurança e entretenimento. Dado o uso disseminado destes sistemas, a indisponibilidade ou desempenho insuficiente dos serviços providos têm sido objetos de atenção particular dos provedores de serviços, infra-estruturas, projetistas de aplicações e da comunidade científica. O planejamento de capacidade destas infraestruturas, permite que os serviços providos pelos sistemas atendam a níveis aceitáveis de qualidade (tais como tempo de resposta, disponibilidade e confiabilidade), fazendo com que os recursos sejam dimensionados de maneira que os custos associados sejam minimizados e os níveis de serviços estabelecidos sejam atendidos.

Grande parte das infra-estruturas computacionais existentes pelo mundo vêm utilizando cada vez mais a tecnologia de *virtualização* em seus sistemas. Esta tecnologia muda a forma sobre como são utilizados os recursos computacionais, permitindo a consolidação de múltiplas plataformas rodando diferentes sistemas operacionais e aplicações heterogêneas em uma única poderosa e confiável infraestrutura. Isso reduz o número de sistemas a serem gerenciados para a mesma tarefa e operação, o que ajuda a economizar energia e minimizar custos. Grandes empresas, tais como IBM, *Microsoft* e HP, estão investindo cada vez mais em diversas soluções de virtualização.

Este trabalho propôs o desenvolvimento de um gerador de *traces* sintéticos com o intuito de simular o consumo dos principais recursos computacionais, tais como processador, memória, disco e rede, para auxiliar o planejamento de capacidade de sistemas em um ambiente de servidores virtuais (*HP VSE - Virtual Server Environment*), através do *HP Capacity Advisor*. A principal vantagem da abordagem desenvolvida é a possibilidade de geração de cargas adaptáveis a diversas situações, onde podem ser gerados *traces* a partir de informações observadas em sistemas reais existentes ou baseados em distribuições de probabilidade, permitindo a geração de valores referentes a novas situações e hipóteses, sem que haja a necessidade de uma infraestrutura física.

Para o desenvolvimento do gerador de carga, foi criada uma metodologia composta por

diversas etapas importantes que envolvem desde o entendimento do ambiente e o estudo de metodologias de geração de carga existentes, passando pela definição dos cenários e do processo de geração de carga, chegando até o desenvolvimento e validação do gerador de *traces* sintéticos, intitulado **WGCap** - *Workload Generator for Capacity Advisor*. Os *traces* gerados pela ferramenta são armazenados em arquivos com o formato CSV compatíveis com o HP *Capacity Advisor*, permitindo a utilização da ferramenta para o estudo e a influência dos recursos computacionais no desempenho do sistema para o planejamento de capacidade da infra-estrutura computacional.

Após o desenvolvimento do WGCap, foram realizados alguns estudos de caso com o intuito de avaliar o gerador de carga desenvolvido, através da geração de alguns *traces* sintéticos, aplicados a cada um dos cenários de geração de carga. Foram gerados diversos *traces* baseados em *traces* reais, distribuições de probabilidade e resumos estatísticos, representando o consumo de vários recursos computacionais. Após a geração, os *traces* criados foram importados para o *Capacity Advisor* com sucesso, e os resultados observados mostraram que os valores gerados de fato conseguiram simular o consumo dos recursos computacionais utilizados, atendendo às especificações definidas durante o processo de geração dos *traces* sintéticos.

7.1 CONTRIBUIÇÕES, LIMITAÇÕES E DIFICULDADES

Entre as principais contribuições deste trabalho podemos destacar:

- A criação de uma metodologia para o desenvolvimento de um gerador de *traces* sintéticos voltado para o planejamento de capacidade em um ambiente de servidores virtuais. Esta metodologia é composta por uma série de atividades extremamente importantes, que permitem analisar todos os requisitos necessários para o desenvolvimento do gerador de carga e os principais mecanismos utilizados;
- O desenvolvimento de um gerador de variáveis aleatórias (*Random Variates*) genérico. Durante o desenvolvimento do *kernel* foram implementadas várias metodologias para a geração de *random variates*, incluindo algumas das principais distribuições de probabilidade contínuas e discretas. Para a visualização e validação dos valores gerados, foi criada uma ferramenta com uma interface gráfica bastante útil, que além de ser essencial para a geração dos valores no gerador de carga, poderá ser utilizada para diversas outras finalidades e aplicações;

- O desenvolvimento de um gerador de carga com suporte a geração de *traces* sintéticos baseados em *traces* reais. Através da implementação da distribuição de probabilidade *empírica*, o gerador de cargas permite a geração de *traces* sintéticos com o mesmo comportamento observado em arquivos contendo o histórico do consumo dos recursos computacionais de sistemas reais em produção;
- O desenvolvimento de um gerador de carga com suporte a geração de *traces* sintéticos baseados em distribuições de probabilidade e/ou resumos estatísticos. Para a simulação dos recursos computacionais, o gerador de carga desenvolvido permite a geração de valores com base em um conjunto de distribuições de probabilidade importantes ou em resumos estatísticos, permitindo a geração de valores referentes a novas situações e hipóteses, sem que haja a necessidade de uma infraestrutura física ou o monitoramento de um sistema real;
- Publicação de artigo em conferência internacional *Qualis B* referente ao processo de geração de carga desenvolvido nesta pesquisa [GSM⁺09].

Algumas limitações e dificuldades também foram observadas durante a execução deste trabalho. A principal delas deve-se ao fato de não ter sido possível o acesso ao ambiente de servidores virtuais HP (VSE), e conseqüentemente a utilização do HP *Capacity Advisor* para a realização de testes e estudos de casos mais complexos. Apesar desta pesquisa ter sido desenvolvida em parceria com a HP-Brasil, não foi possível a disponibilização do ambiente VSE e seus principais componentes, uma vez que a sua instalação necessita de uma infraestrutura de *hardware* bastante avançada e com custo elevado, além de serem necessárias aquisições de licenças para a utilização do ferramental. No entanto, foi disponibilizada uma documentação vasta contendo todas as informações necessárias para a execução desta pesquisa, envolvendo desde manuais para a execução do VSE, até documentos contendo as principais características e arquitetura de funcionamento do *Capacity Advisor*. Desta forma, o trabalho foi desenvolvido “desconectado” ao ambiente HP VSE, e após a sua conclusão, geramos alguns *traces* sintéticos que posteriormente foram enviados ao laboratório da HP para avaliar a sua execução no *Capacity Advisor*.

7.2 TRABALHOS FUTUROS

O presente trabalho não está esgotado. Durante a sua execução identificamos diversos trabalhos que poderão dar-lhe continuidade. O trabalho focou-se no desenvolvimento e

validação de um gerador de *traces* sintéticos para o HP *Capacity Advisor*, e na análise do seu funcionamento em estudos de caso. No entanto, não foi desenvolvida uma metodologia voltada para as possibilidades de geração de carga com a utilização da ferramenta por parte dos avaliadores (usuários). Como trabalho futuro é extremamente importante a criação de uma metodologia para o avaliador, focando nas possíveis estratégias para o planejamento de capacidade de sistemas com o uso do gerador de carga, além da avaliação da representatividade dos *traces* sintéticos gerados. Estudos de caso mais complexos devem ser realizados, ressaltando a utilização do gerador para o planejamento de infraestruturas em empresas e *datacenters* em produção, focando na melhoria da utilização dos recursos do ambiente.

Além disso, também podemos destacar a implementação de um novo gerador de carga (genérico), que possa contribuir para o planejamento de capacidade de qualquer infraestrutura computacional, seja ela composta por ambientes virtuais ou reais, e independente de plataforma e sistema operacional. Apesar desta pesquisa estar restrita a utilização do gerador de carga para o planejamento de capacidade com o HP *Capacity Advisor*, os mecanismos desenvolvidos para a geração de carga podem ser reaproveitados e utilizados para novas abordagens mais abrangentes. Uma abordagem interessante é a utilização desses mecanismos para o planejamento de capacidade em ambientes *Cloud Computing*.

Adicionalmente, pode ser realizado um trabalho de expansão do *kernel* desenvolvido, onde podem ser adicionados mecanismos para a geração de variáveis aleatórias (*random variates*) baseados em um novo conjunto de distribuições de probabilidade, ampliando o número de possibilidades para a geração de *traces* sintéticos.

REFERÊNCIAS BIBLIOGRÁFICAS

- [ABB⁺06] Kay S. Anderson, Joseph P. Bigus, Eric Bouillet, Parijat Dube, Nagui Halim, Zhen Liu, and Dimitrios E. Pendarakis. SWORD: scalable and flexible workload generator for distributed data processing systems. In L. Felipe Perrone, Barry Lawson, Jason Liu, and Frederick P. Wieland, editors, *Winter Simulation Conference*, pages 2109–2116. WSC, 2006.
- [All08] John Allspaw. *The Art of Capacity Planning*. O'Reilly Media, 2008.
- [AW97] Martin F. Arlitt and Carey L. Williamson. Internet web servers: workload characterization and performance implications. *IEEE/ACM Trans. Netw.*, 5(5):631–645, October 1997.
- [BA94] Luis Barriga and Rassul Ayani. Efficient generation of synthetic traces. In *Second Euromicro Workshop on Parallel and Distributed Processing*, 1994.
- [BC98] Paul Barford and Mark Crovella. Generating Representative Web Workloads for Network and Server Performance Evaluation. In *Measurement and Modeling of Computer Systems*, pages 151–160, 1998.
- [BDF⁺03] Paul Barham, Boris Dragovic, Keir Fraser, Steven Hand, Tim Harris, Alex Ho, Rolf Neugebauer, Ian Pratt, and Andrew Warfield. Xen and the art of virtualization. In *SOSP '03: Proceedings of the nineteenth ACM symposium on Operating systems principles*, pages 164–177, New York, NY, USA, 2003. ACM.
- [BW00] Ram Balakrishnan and Carey Williamson. The syntraff suite of traffic modeling toolkits. In *MASCOTS '00: Proceedings of the 8th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, page 333, Washington, DC, USA, 2000. IEEE Computer Society.

- [BW02a] G. Bai and C. Williamson. Workload characterization in web caching hierarchies. In *MASCOTS '02: Proceedings of the 10th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunications Systems*, page 13, Washington, DC, USA, 2002. IEEE Computer Society.
- [BW02b] M. Busari and C. Williamson. ProWGen: A Synthetic Workload Generation Tool for Simulation Evaluation of Web Proxy Caches. In *Computer Networks*, volume 38, pages 779–794, June 2002.
- [cap09] HP Capacity Advisor Version 4.1 User’s Guide, January 2009.
- [Cas08] Lafortune Stephane Cassandras, Christos G. *Introduction to Discrete Event Systems*. Springer, 2nd ed., 2008.
- [Chu03] Christopher A. Chung. *Simulation Modelling Handbook: A Practical Approach*. Routledge, USA, 2003.
- [CLR67] I. M. Chakravarti, R. G. Laha, and J. Roy. *Handbook of Methods of Applied Statistics*, volume I. John Wiley and Sons, USE, 1967.
- [CXZ07] Bin Cai, Changsheng Xie, and Guangxi Zhu. Performance evaluation of a load self-balancing method for heterogeneous metadata server cluster using trace-driven and synthetic workload simulation. *Parallel and Distributed Processing Symposium, International*, 0:405, 2007.
- [Dev86] Luc Devroye. *Non-Uniform Random Variate Generation*. Springer-Verlag, New York NY, 1986.
- [DRT04] Yves Denneulin, Emmanuel Romagnoli, and Denis Trystram. A synthetic workload generator for cluster computing. In *18th International Parallel and Distributed Processing Symposium (IPDPS'04)*, volume 15, page 243b, Los Alamitos, CA, USA, 2004. IEEE Computer Society.
- [dyn09a] HP Insight Dynamics - VSE and HP VSE Management Software 4.1 Getting Started Guide, January 2009.
- [dyn09b] HP Insight Dynamics - VSE Version 4.1 for ProLiant Installation and Configuration Guide, January 2009.

- [eas09] Easyfit - Distribution Fitting Tool. <http://www.mathwave.com/products/easyfit.html>, 2009. Acesso em Novembro de 2009.
- [EBDN00] Lieven Eeckhout, Koen De Bosschere, Koen De, and Henk Neefs. Performance analysis through synthetic trace generation. In *In The IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS-2000)*, pages 1–6, 2000.
- [FM03] Paul Fortier and Howard Michel. *Computer Systems Performance Evaluation and Prediction*. Digital Press, 2003.
- [Gan95] Gregory R. Ganger. Generating representative synthetic workloads: An unsolved problem. In *International Conference on Management and Performance Evaluation of Computer Systems (Nashville, TN)*, pages 1263–1269, 1995.
- [GBT06] Hermann de Meer Gunter Bolch, Stefan Greiner and Kishor S. Trivedi. *Queueing Networks and Markov Chains: Modeling and Performance Evaluation with Computer Science Applications*. John Wiley, second edition, New York, NY, 2006.
- [Gen98] James E. Gentle. *Random Number Generation and Monte Carlo Methods*. Springer, New York NY, 1998.
- [GS00] María E. Gómez and Vicente Santonja. A New Approach in the Modeling and Generation of Synthetic Disk Workload. In *MASCOTS '00: Proceedings of the 8th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, page 199, Washington, DC, USA, 2000. IEEE Computer Society.
- [GS03] Matteo Golfarelli and Ettore Saltarelli. The workload you have, the workload you would like. In *DOLAP '03: Proceedings of the 6th ACM international workshop on Data warehousing and OLAP*, pages 79–85, New York, NY, USA, 2003. ACM.
- [GSM⁺09] H. E. S. Galindo, W. M. Santos, P. R. M. Maceil, B. Silva, S. M. L. Galdino, and J. P. Pires. Synthetic workload generation for capacity planning of virtual server environments. In *IEEE International Conference on Systems, Man, and Cybernetics*, pages 2916–2921, San Antonio, TX, USA, 2009. IEEE Systems, Man, and Cybernetics Society.

- [Hav98] Boudewijn R. Haverkort. *Performance of Computer Communication Systems: A Model-Based Approach*. John Wiley & Sons, 1998.
- [Hel98] P. Hellekalek. Good random number generators are (not so) easy to find. *Math. Comput. Simul.*, 46(5-6):485–505, 1998.
- [Her06] B. Herington, D. & Jacquot. *The HP Virtual Server Environment: Making the Adaptive Enterprise Vision a Reality in your Data Center*. Prentice Hall, 2006. Cap. 17 & 18.
- [hpu10] HP-UX 11i. <http://h71028.www7.hp.com/enterprise/w1/en/os/hpux11i-overview.html>, 2010. Acesso em Dezembro de 2009.
- [ibm10] IBM Virtualization. <http://www-03.ibm.com/systems/virtualization/>, 2010. Acesso em Janeiro de 2010.
- [int09] Introduction to the HP Virtual Server Environment, January 2009.
- [Jai91] Raj Jain. *Art of Computer Systems Performance Analysis: Techniques For Experimental Design, Measurements, Simulation and Modeling*. Wiley Computer Publishing, John Wiley & Sons, 1991.
- [jav09] Java Technology. <http://www.sun.com/java/>, 2009. Acesso em Dezembro de 2009.
- [KRLM08] Rachid El Abdouni Khayari, Matthias Rucker, Axel Lehmann, and Adisa Musovic. Parasyntg: A parameterized synthetic trace generator for representation of www traffic. In *International Symposium on Performance Evaluation of Computer and Telecommunication Systems - SPECTS*, 2008.
- [LB06] Linda M. Laird and M. Carol Brennan. *Software Measurement and Estimation: A Practical Approach*. John Wiley & Sons, 2006.
- [L'E98] Pierre L'Ecuyer. Uniform random number generators. In *WSC '98: Proceedings of the 30th conference on Winter simulation*, pages 97–104, Los Alamitos, CA, USA, 1998. IEEE Computer Society Press.
- [Lil00] David J. Lilja. *Measuring Computer Performance: A Practitioner's Guide*. Cambridge University Press, 2000.

- [LK91] Averill M. Law and W. David Kelton. *Simulation Modeling & Analysis*. Second Edition, McGraw-Hill, 1991.
- [MA98] D.A. Menascé and V.A.F. Almeida. *Capacity Planning for Web Performance: Metrics, Models, and Methods*. Prentice-Hall, Inc. Upper Saddle River, NJ, USA, 1998.
- [MAD94] D.A. Menascé, V.A.F. Almeida, and L.W. Dowdy. *Capacity Planning and Performance Modeling: from Mainframes to Client-Server Systems*. Prentice-Hall, Inc. Upper Saddle River, NJ, USA, 1994.
- [Mah97] Bruce A. Mah. An Empirical Model of HTTP Network Traffic. In *INFOCOM '97: Proceedings of the INFOCOM '97. Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Driving the Information Revolution*, page 592, Washington, DC, USA, 1997. IEEE Computer Society.
- [MdOB06] P. A. Morettin and W. de O. Bussab. *Estatística Básica*. Saraiva, 2006.
- [mic10] Microsoft Virtualization. <https://lb1.www.ms.akadns.net/virtualization/en/us/default.aspx>, 2010. Acesso em Janeiro de 2010.
- [Min03] *Minitab Statistics*. Minitab Inc., 2003.
- [min09] Minitab statistical software. <http://www.minitab.com/>, 2009. Acesso em Dezembro de 2009.
- [Mon03] Douglas C. Montgomery. *Applied Statistics and Probability for Engineers*. John Wiley and Sons, 3rd Edition, 2003.
- [MR03] Douglas C. Montgomery and George C. Runger. *Applied Statistics and Probability for Engineers*. John Wiley & Sons, 2003.
- [MW02] Nayden Markatchev and Carey Williamson. Webtraff: A GUI for web proxy cache workload modeling and analysis. In *In Proceedings of the 10th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS '02)*, pages 356–363, 2002.
- [net09] NetBeans IDE. <http://netbeans.org/>, 2009. Acesso em Dezembro de 2009.

- [NIST03] National Institute of Standards and Technology NIST. Engineering Statistics Handbook, 2003. Acesso em Janeiro de 2010.
- [ovp09] HP OpenVMS Systems. <http://h71000.www7.hp.com/openvms/products/ovpa/index.html>, 2009. Acesso em Dezembro de 2009.
- [pmp09] HP Performance Management Pack. <http://software.hp.com/portal/swdepot/displayProductInfo.do?productNumber=HPPMP>, 2009. Acesso em Dezembro de 2009.
- [PZW⁺] Padala Padala, Xiaoyun Zhu, Zhikui Wang, Sharad Singhal, and Kang G. Shin. Performance evaluation of virtualization technologies for server consolidation. Technical report.
- [Rip90] B. D. Ripley. Thoughts on pseudorandom number generators. *J. Comput. Appl. Math.*, 31(1):153–163, 1990.
- [SC89] George W. Snedecor and William G. Cochran. *Statistical Methods, Eighth Edition*. Iowa State University Press, 1989.
- [SF02] E. S. Sorenson and J. K. Flanagan. Evaluating synthetic trace models using locality surfaces. In *Fifth IEEE Annual Workshop on Workload Characterization*, pages 23–33, 2002.
- [Sha05] Y. Shafranovich. RFC 4180: Common Format and MIME Type for Comma-Separated Values (CSV) Files `csvrfc`, 2005.
- [Sin04] Amit Singh. An Introduction to Virtualization. <http://www.kernelthread.com/publications/virtualization/>, January 2004. Acesso em Dezembro de 2009.
- [SMG01] W. Shi, M. H. Macgregor, and P. Gburzynski. Synthetic trace generation for the internet. In *in: The 4th IEEE Workshop on Workload Characterization (WWC-4*, pages 169–174, 2001.
- [SN05] James E. Smith and Ravi Nair. The architecture of virtual machines. *Computer*, 38(5):32–38, 2005.
- [Ste74] M. A. Stephens. Edf statistics for goodness of fit and some comparisons. *Journal of the American Statistical Association*, 69:730–737, 1974.

- [Ste76] M. A. Stephens. Asymptotic results for goodness-of-fit statistics with unknown parameters. *The Annals of Statistics*, 4:357–369, 1976.
- [Ste77] M. A. Stephens. Goodness of fit for the extreme value distribution. *Biometrika*, 64:583–588, 1977.
- [Tri05] M.F. Triola. *Introdução à estatística*. Livros técnicos e Científicos, 10 edition, 2005. Tradução Vera Regina de Farias e Flores; revisão de Ana Maria Lima de Farias e Flores.
- [Tri06] K.S. Trivedi. *Probability and Statistics with Reliability, Queuing, and Computer Science Applications*. John Wiley and Sons, 2nd Edition, 2006.
- [uti09] Utilization Provider. <http://software.hp.com/portal/swdepot/displayProductInfo.do?productNumber=UTILPROVIDER>, 2009. Acesso em Dezembro de 2009.
- [vmw09] Vmware esx. <http://www.vmware.com/products/esxi/>, 2009. Acesso em Dezembro de 2009.
- [vse09] Implementing a Virtual Server Environment: Getting Started, January 2009.
- [WST92] Joel L. Wolf, Harold S. Stone, and Dominique Thiébaud. Synthetic traces for trace-driven simulation of cache memories. *IEEE Trans. Comput.*, 41(4):388–410, 1992.