Autoescalonamento de máquinas virtuais baseado em séries temporais e thresholds.

Paulo Roberto Pereira da Silva

Orientador: Prof. Paulo Maciel
Coorientador: Prof. Jean Teixeira

Introdução

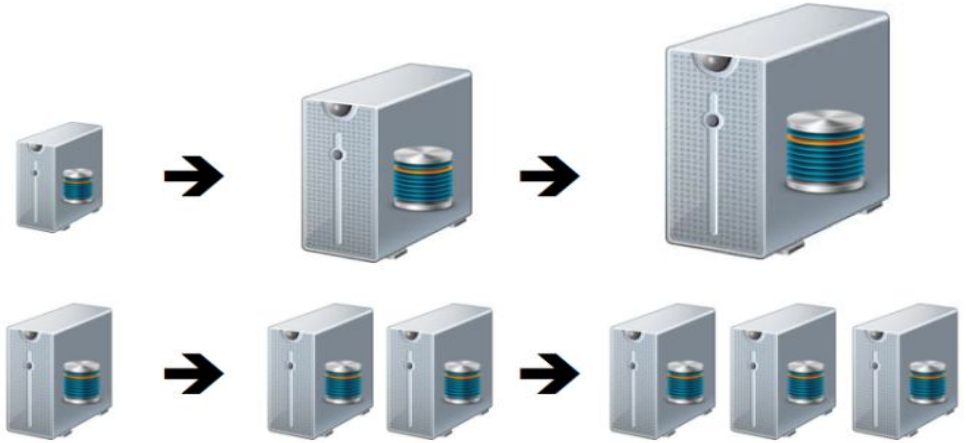
A computação em nuvem é uma tecnologia que está se tornando cada vez mais popular. Isso é devido principalmente à sua natureza de elasticidade: os usuários podem adquirir e liberar recursos sob demanda, e pagar apenas pelos recursos de que necessitarem (modelo de pagamento por uso ou pay-as-you-go).

Motivação

- Decidir a quantidade correta de recursos para um ambiente de *cloud computing* é uma espada de dois gumes, pois pode ocasionar um *over-provisioning* ou um *under-provisioning.*
- A saturação ou o desperdício de recursos está entre os maiores problemas enfrentados pela *cloud computing*.

Background

Alocação de recursos pode ser tanto **horizontalmente** quanto **verticalmente**. Porém, os sistemas operacionais mais comuns atualmente não suportam mudanças na CPU ou na quantidade de memória de forma *on-the-fly* (sem *reboot*).

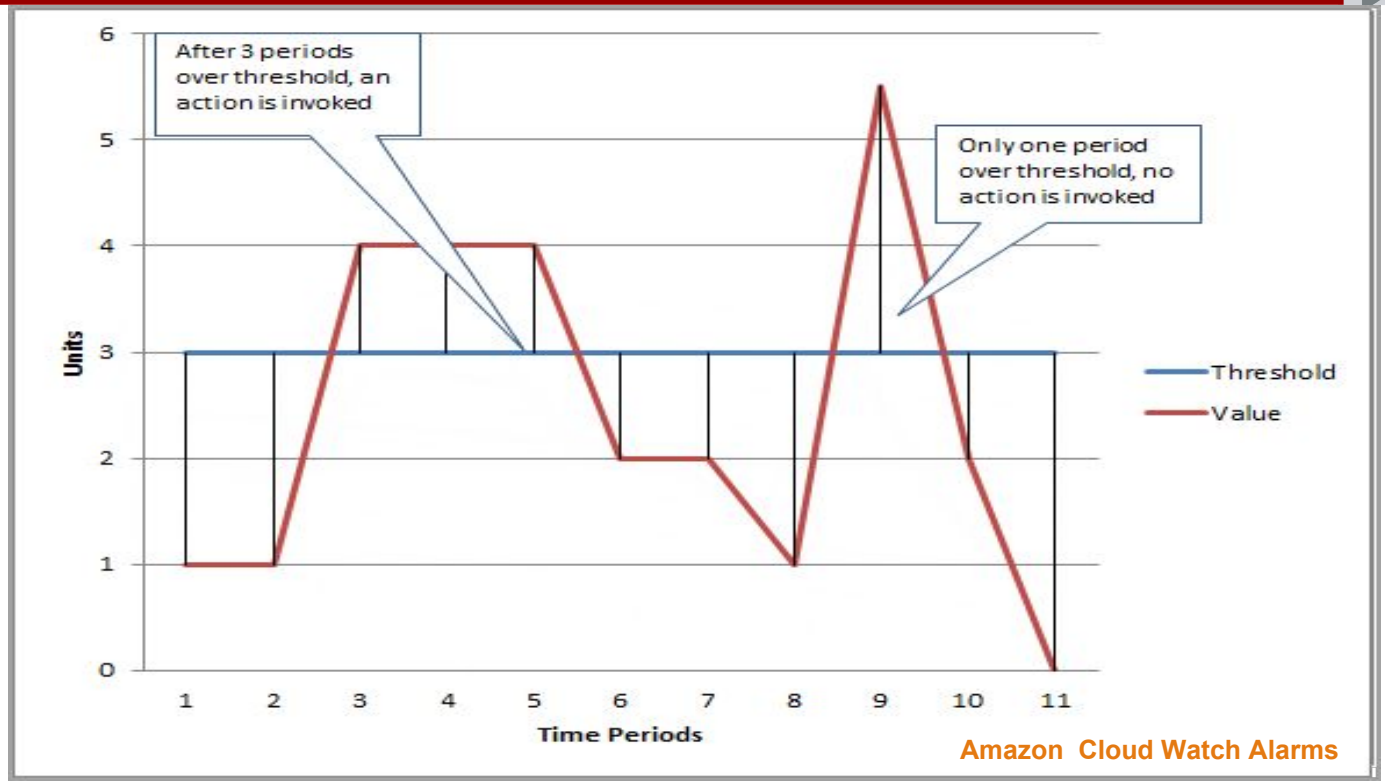


Background

A elasticidade é uma característica fundamental da *cloud computing* onde os recursos são alocados e executados de acordo com as demandas dos usuários. A elasticidade pode ser dividida em dois tipos:

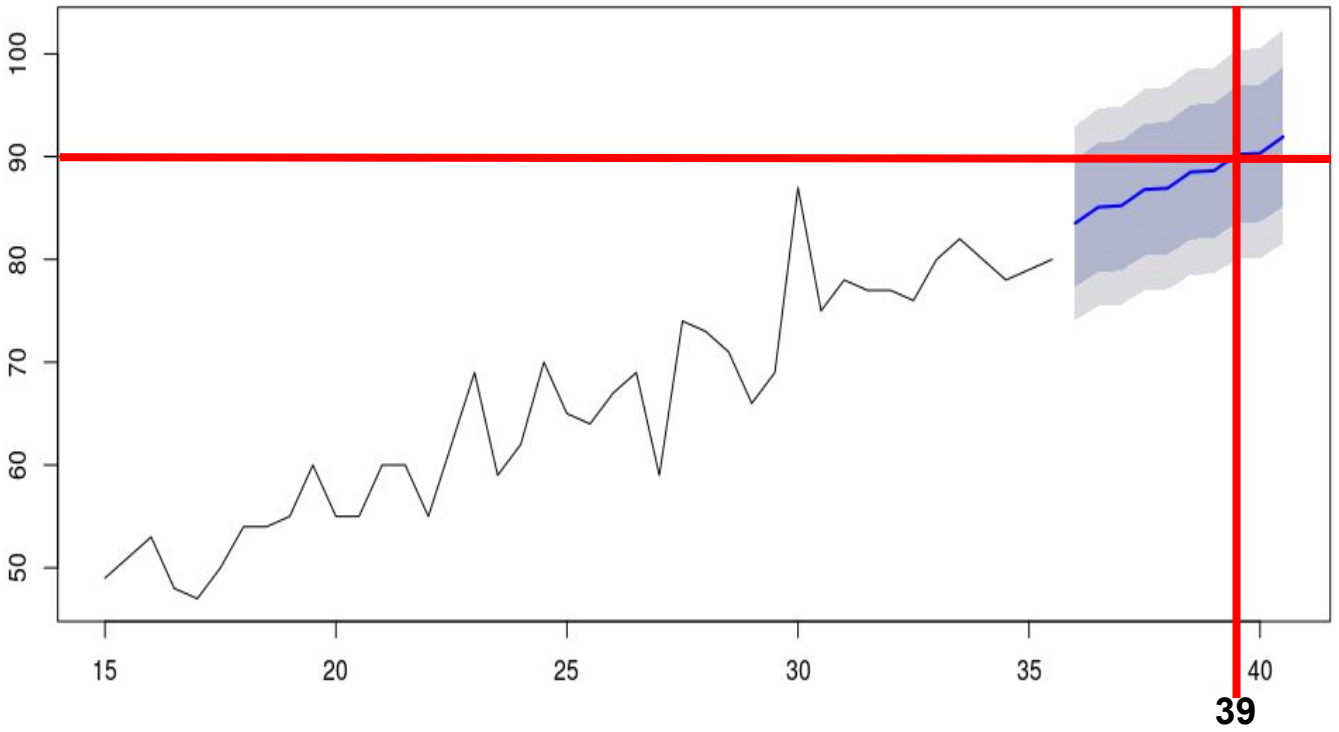
- **Reativo**
- **Proativo**

Reativo



Amazon Cloud Watch Alarms

Proativo

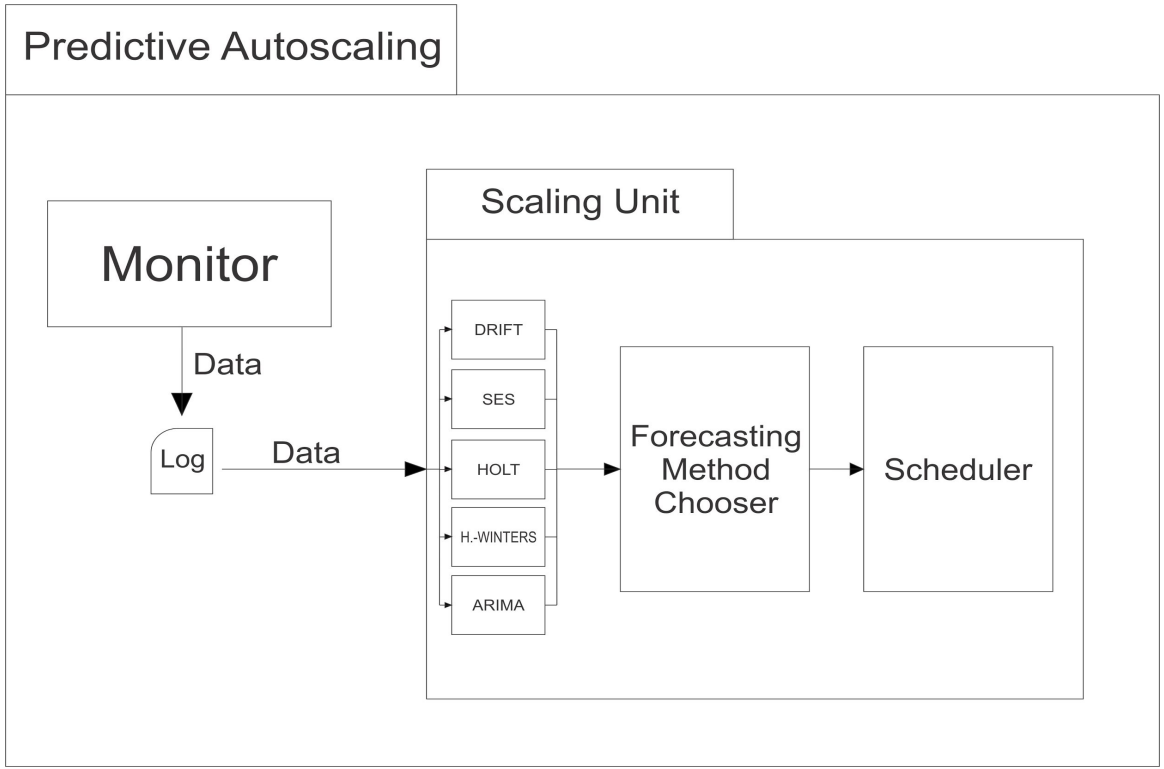


Proposta

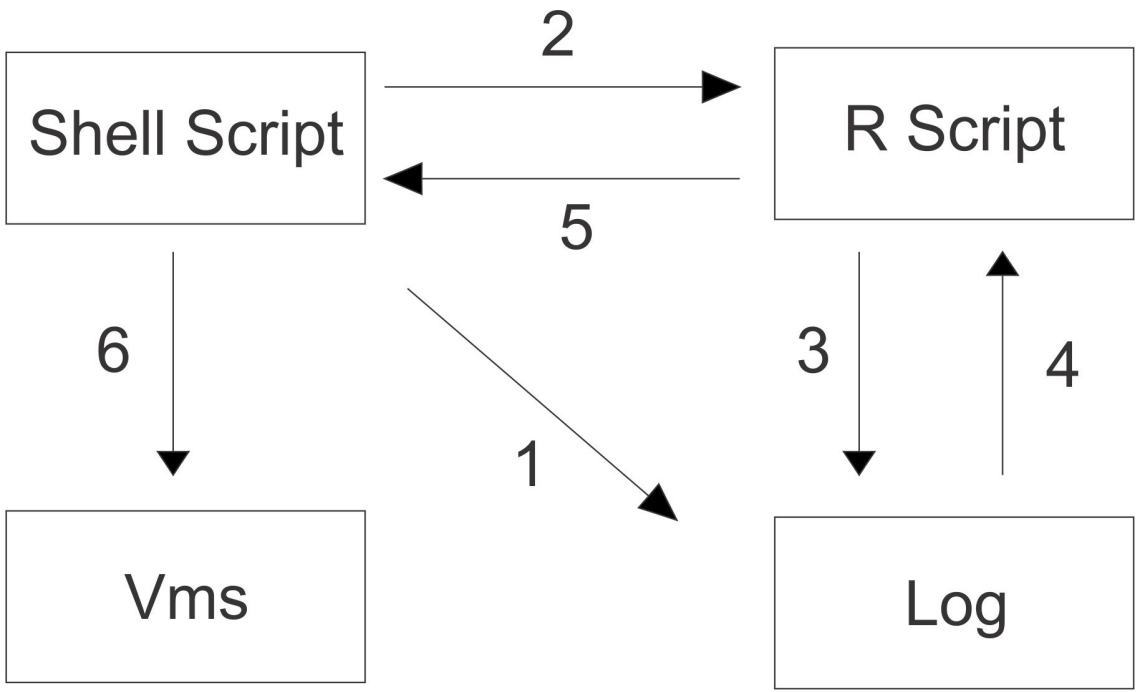
- Desenvolver uma estratégia de autoescalonamento de máquinas virtuais baseada em séries temporais e thresholds.

- A série temporal será utilizada para prever quando o **threshold** será **alcançado**, de forma que o **recurso** seja preparado com **antecedência** para que no momento que o **limiar** seja **atingido**, o **recurso** esteja **pronto** para a **utilização**.

Proposta


Predictive Autoscaling
Monitor
Data
Log
Data
Scaling Unit
DRIFT
SES
HOLT
H.-WINTERS
ARIMA
Forecasting Method Chooser
Scheduler

Proposta

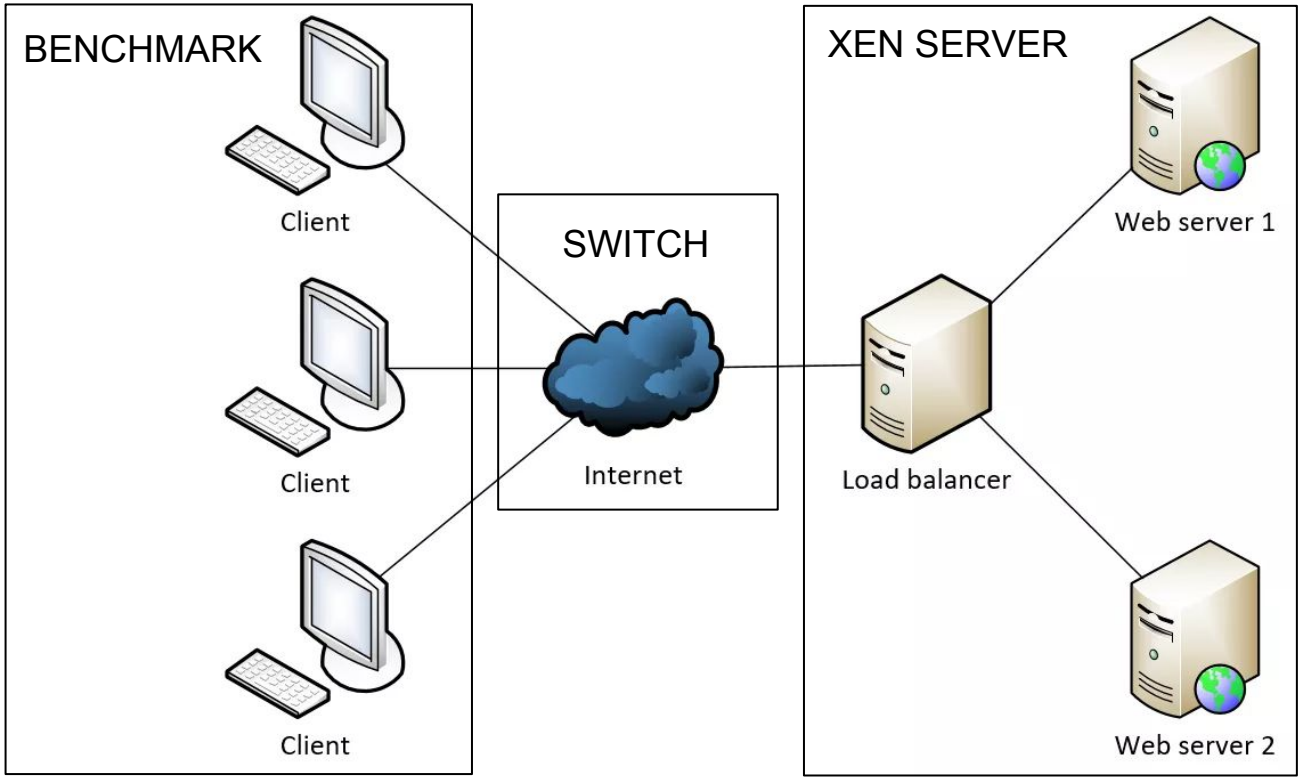


Objetivos específicos

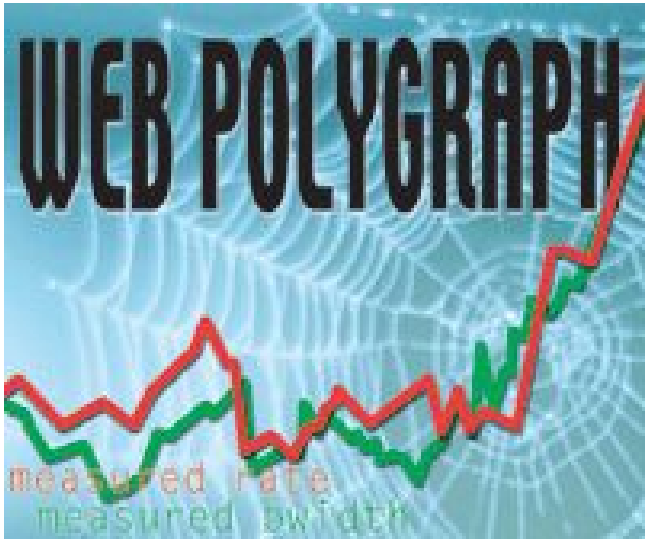
- Comparar diversas técnicas de predição de séries temporais para diferentes padrões de *workload*.

- Analisar o impacto da escolha da quantidade de amostras e do tamanho do horizonte de predição na acurácia das técnicas de predição.

Problemas Enfrentados



Próximos Passos



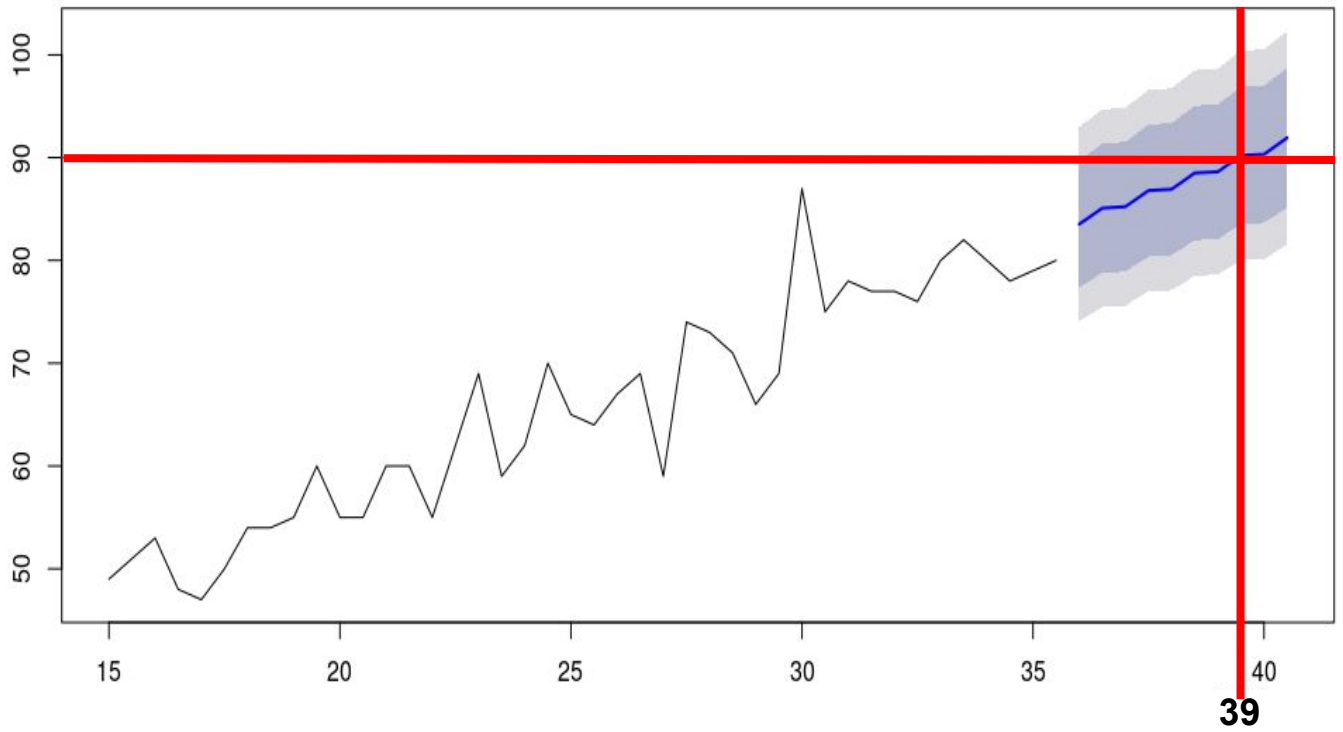
Testes Iniciais

- 2 VMs
  - 1 (núcleo) CPU
  - 1 GB RAM
  - 15 GB HD
- Experimento foi executado durante 40 minutos, onde foram coletadas amostras a cada 30 segundos, totalizando 80 amostras.
- O recurso medido foi o consumo de CPU.

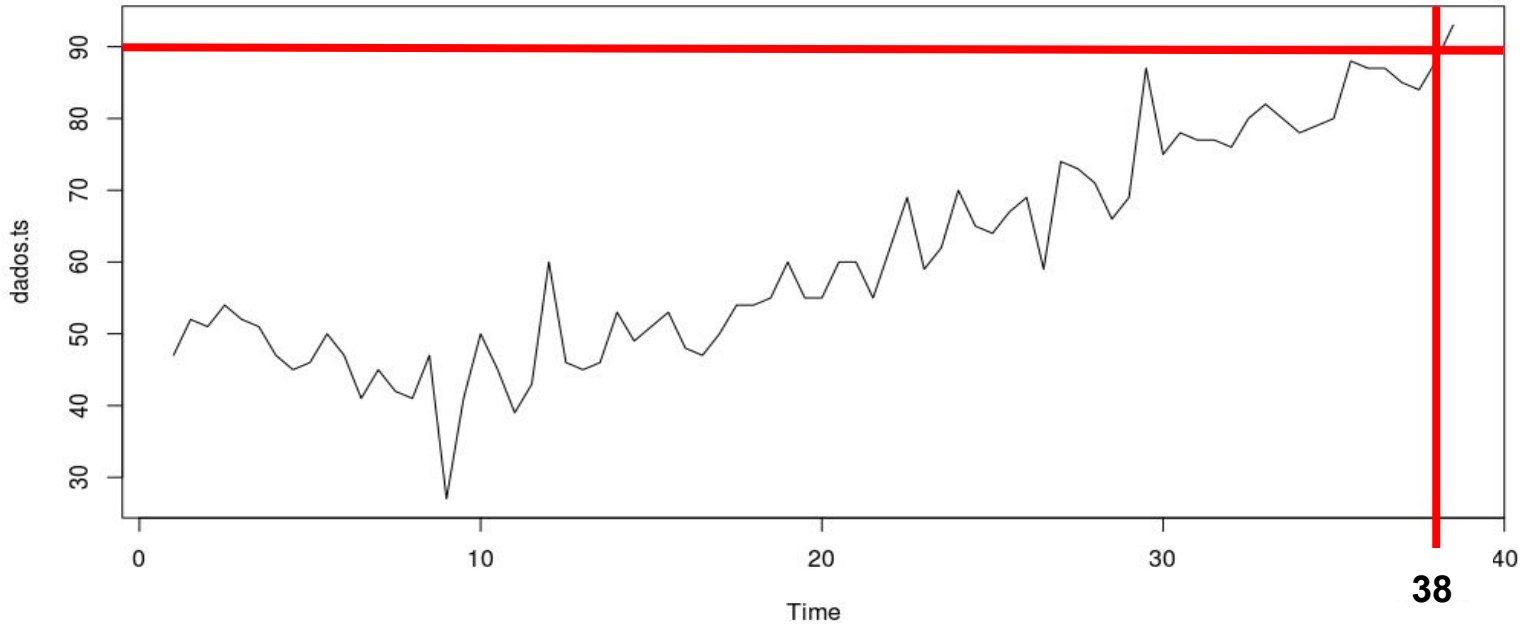
Testes Iniciais

- O threshold da utilização de CPU foi de 90%.
- A cada nova entrada uma nova predição era executada.
- 70 amostras eram utilizadas para treinamento e 10 para teste, e o horizonte de predição também era de 10 amostras (5 minutos).

Testes Iniciais



Testes Iniciais



Testes Iniciais

Mean Absolute Error				
Drift	SES	Holt	Holt-Winters	ARIMA
16.3066	2.9233	2.3684	**2.2341**	5.1434

FIM

Dúvidas e Comentários

Referências

- Lorido-Botran, Tania, Jose Miguel-Alonso, and Jose A. Lozano. "A review of auto-scaling techniques for elastic applications in cloud environments." Journal of Grid Computing 12.4 (2014): 559-592.
- Nikravesh, Ali Yadavar, Samuel A. Ajila, and Chung-Horng Lung. "Towards an autonomic auto-scaling prediction system for cloud resource provisioning." Proceedings of the 10th International Symposium on Software Engineering for Adaptive and Self-Managing Systems. IEEE Press, 2015.
- Dutreilh, Xavier, et al. "From data center resource allocation to control theory and back." Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on. IEEE, 2010.
- “Amazon Auto Scaling,” http://aws.amazon.com/autoscaling/