

Utilização de Redes Neurais para Gerência de Servidores Virtuais Web

Danilo Souza¹ Iago Medeiros¹

¹Universidade Federal do Pará

20 de Junho de 2013

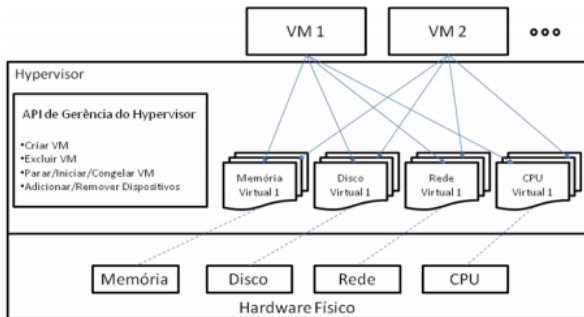


Introdução do Artigo

- Alto consumo de energia em *datacenters* (40% para equipamentos e 60% para infra-estrutura)
- Virtualizar servidores reduz o consumo de energia
 - Gera maior ociosidade
- Normalmente os equipamentos são superdimensionados (ociosidade)
- O autor propõe uma nova política de gerência de servidores *web*

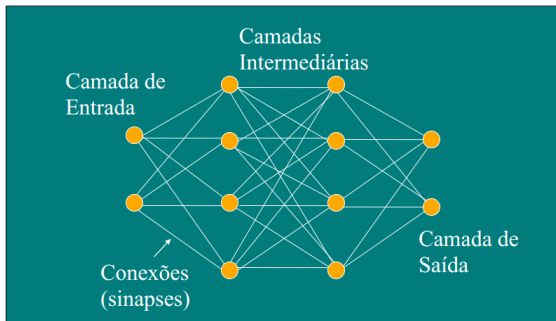
Arquitetura do Xen

- Utiliza para-virtualização
- Possui um *kernel* modificado (*dom0*)
- O *hypervisor* faz o mapeamento de recursos físicos em virtuais
- As VM's rodam no Domínio U (*domU*)



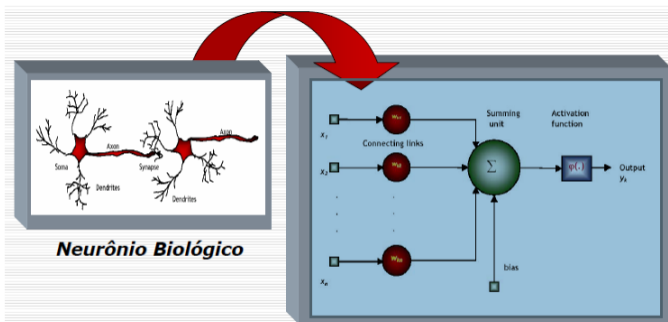
O que é uma RNA?

- São modelos computacionais
 - Adquirir, armazenar e utilizar conhecimento
 - Generalizar determinados sistemas
 - Agrupar e organizar dados
- Compostos por neurônios (unidades de processamento)
- Interligada por sinapses (conexões)
- Agrupados em camadas



Funcionamento da RNA

- Neurônios são utilizados para armazenar informações
- Cada sinapse é caracterizada por um peso
- O somador adiciona as entradas ponderadas
- A função de ativação restringe a saída entre $[0,1]$ ou $[-1,1]$
- O *bias* é utilizado para aumentar ou diminuir a entrada da função de ativação



Gerência de VM's

O autor cita três trabalhos sobre gerência inteligente de VM's

- Utilizando Balanceamento de carga
 - Analisa os recursos dos *hosts* [Megav 2007]
 - Migra as VM's de acordo com a quantidade de recursos disponíveis
- Utilizando um controlador *lookahead* [Kusic et al. 2008]
 - Analisa a quantidade de conexões e o tempo médio de resposta
 - Define a quantidade de *hosts* que devem ser ativados
- Utilizando múltiplos domínios [Ruth et al. 2006]
 - Um *host* e suas VM's formam um domínio
 - O sistema analisa a demanda de CPU e memória das VM's
 - O sistema migra as VM's caso os recursos de um domínio estejam esgotados

Balanceamento de Carga utilizando RNA

O autor cita três trabalhos referentes ao uso de RNA's para balanceamento de carga

- Controlar frequências de operação de CPU's [Souza et al. 2008]
 - Utiliza DFVS (*Dynamic Voltage and Frequency Scaling*) para controlar o *clock* das CPU's
 - Prove economia de energia mantendo o nível do serviço
 - Foram mapeadas frequência e taxa de ocupação das CPU's para associação com consumo de energia
- Classificar estados de um servidor *web* utilizando RNA [Simula et al. 2008]
 - Mapas auto-organizáveis com nove dimensões de entrada
 - Foram medidos uso de CPU, pacotes recebidos, blocos lidos/escritos do disco, etc.
 - A rede apresentou 4 estados de ocupação do servidor
 - Criação de uma política de tolerância a falhas baseada nesses estados

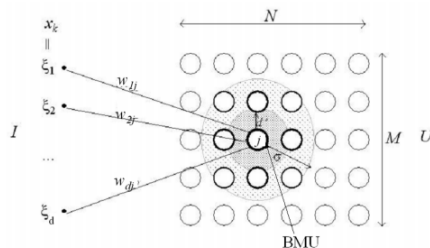
Balanceamento de Carga utilizando RNA

- Classificar uso de memória utilizando RNA [Lin 2006]
 - Utiliza 3 indicadores de consumo de memória como entrada
 - A rede indicou 6 estados de consumo de memória
 - Um módulo atuador foi projeto com base nos valores desses estados
 - Este módulo substitui o *OOM Killer* nativo do *Linux*

Mapas Auto-Organizáveis

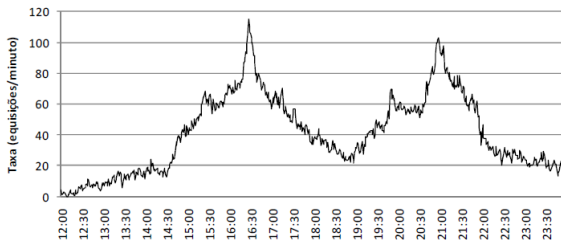
Faz analogia com a região do córtex cerebral

- Aloca regiões específicas para atividades específicas
- O grau de ativação dos neurônios diminui conforme se distancia da região de ativação inicial
- RNA com duas camadas (Entrada I e saída U)
- Geralmente são mapas $N \times M$



A RNA do Artigo

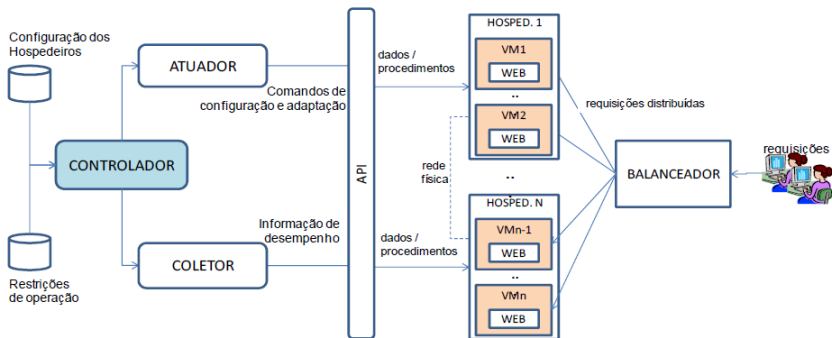
- 3 nós na interface de entrada (indicadores do cluster)
 - Tempo de resposta
 - Potência consumida
 - Taxa de requisições
- Uma mapa de 10x10 neurônios (captar o máximo de padrões possível)
- Treinamento *offline* usando *workload* de acesso ao site da Copa do Mundo de 1998



Abordagem

- Substituir a política de gerenciamento original por rede neural, aprimorando os mecanismos de atuação no cluster
- Definir os estados de operação no cluster de acordo com os parâmetros analisados
- Propor intervenções específicas da API de atuação de acordo com o estado do cluster
- Objetivo: economia de energia e manutenção do tempo de resposta das requisições

Sistema Proposto



Analizando Sistema Proposto

- Controlador: fornece a classificação do estado corrente conforme as medições coletadas
- Atuador: aciona as funções de gerência da API de acordo com a ordem do Controlador
- Coletor: obtém através da API os dados monitorados por agentes que atuam nos hospedeiros do cluster
- Balanceador: recebe requisições do usuário e as distribui para as VMs executadas nos servidores físicos hospedeiros. Deve reconhecer o estado atual e as diferentes localizações das VMs

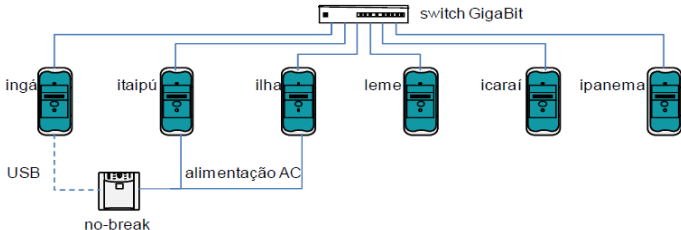
Políticas de Reconfiguração

- Região 1. TR 33,3%, POT 31,8%, REQ 22,2%
- Região 2. TR 56,3%, POT 48,9%, REQ 49,1%
- Região 3. TR 113,8%, POT 93,6%, REQ 91,2%
- Região 4. TR 73,5%, POT 68,1%, REQ 64,7%
- Região 5. TR 89,3%, POT 85,6%, REQ 81,6%

Intervenções da API

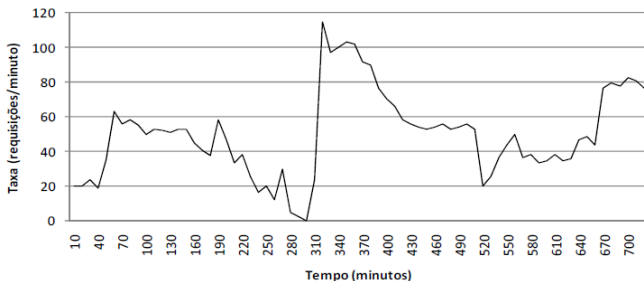
- Alterar frequência da CPU
- Ativar/Inativar núcleo da CPU
- Migrar VMs entre os servidores hospedeiros
- Ativar/Inativar servidores físicos

- Os testes foram feitos usando como infra-estrutura um conjunto de servidores executando Linux
- Os indicadores avaliados nos testes foram potência elétrica consumida e tempo de resposta da aplicação



- Ingá executa o sistema de gerência
- No-break monitora a potência elétrica consumida
- Icaraí simula solicitações de clientes Web
- Ipanema é o alvo das requisições
- Itaipú e Ilha são hospedeiros cluster e executam virtualizador Xen para hospedar VMs
- Leme coleta informações e exibe em tempo real

- Para avaliar o desempenho da rede neural, o log de acesso da página da NASA foi utilizado
- As requisições foram normalizadas para 115/minuto (similar ao workload da Copa de 98)



- A rede apresentou desempenho satisfatório
- Melhor desempenho em relação ao método sem RNA's
- Investigar suporte para aplicações heterogêneas