

Aplicação de Algoritmos de Provisionamento Baseados em Contratos de Nível de Serviço para Computação em Nuvem

Fernando Schubert¹, Carlos Oberdan Rolim¹, Carlos Becker Westphall²

¹URI – Universidade Regional Integrada do Alto Uruguai e das Missões – Campus Santo Ângelo – Av. Universidade das Missões s/n – Santo Ângelo – RS – Brasil

²UFSC – Universidade Federal de Santa Catarina – LRG – Laboratório de Redes e Gerência – Caixa Postal 476 - 88040-970 - Florianópolis – SC – Brasil

fernando.schubert@bea-services.com, ober@urisan.tche.br,
westphal@inf.ufsc.br

Abstract. *Cloud computing came to change the way that computing is delivered and used, turning it into a utility like water or electricity. In this context, many challenges and opportunities appear to make the Cloud a stable, accessible and trustful environment. Resource provisioning in the Cloud must be dynamic and adapt itself when the needs changes. In this paper, a provisioning method is proposed using neural networks to provide the desired quality of service and assure SLA.*

Resumo. *A computação em nuvem surgiu para mudar a forma que a computação é oferecida e utilizada, tornando a computação um serviço oferecido sob demanda como eletricidade, água, etc. Neste contexto desafios e oportunidades surgem para tornar a Nuvem um ambiente estável, acessível e confiável para seus potenciais usuários. O provisionamento de recursos na Nuvem deve ser dinâmico e atender aos requisitos dos consumidores. Propõe-se neste artigo um mecanismo de provisionamento dinâmico para ambientes em Nuvem, utilizando-se de máquinas de aprendizado estatísticas, através de redes neurais, obtendo-se como resultados do provisionamento as quantidades de recursos necessários para manutenção da qualidade de serviço e SLA.*

1. Introdução

Através da história da computação houve várias mudanças de paradigma. Dos *mainframes* para mini-computadores, dos mini-computadores ao micro-processamento e do micro-processamento aos computadores em rede. Enquanto as definições ainda estão em debate, fundamentalmente, computação em nuvem (*cloud computing*) pode ser definida como um modelo de serviços onde a informação é armazenada e processada na Internet, a Nuvem, usualmente em *datacenters* contendo uma grande capacidade de processamento e armazenamento, podendo ser acessado remotamente através de vários clientes e plataformas [Grimes, Jaeger e Lin 2008].

Computação em nuvem tem sido comumente referenciada como um conjunto de idéias como Software como Serviço (*SaaS – Software as a Service*, em inglês), Web 2.0, computação em grade, utilização maciça de virtualização e computação como utilidade (*utility computing*). Na essência, computação em nuvem é um conceito chave que busca sintetizar e encapsular este conceito de processamento e armazenamento ubíquo, mascarando dos seus usuários a real complexidade das camadas que compõem a Nuvem [Grimes, Jaeger e Lin 2008].

O modelo computacional em nuvem apresenta vários desafios e obstáculos a serem superados. Um dos mais importantes é proporcionar à Nuvem a elasticidade necessária mantendo os requisitos de qualidade de serviço (QoS) cumprindo assim os acordos de níveis de serviço (SLA – Service Level Agreement) requeridos e previamente estabelecidos com clientes. A elasticidade é a capacidade da Nuvem em se auto gerir, escalonando recursos para mais ou menos de acordo com a demanda. Esta elasticidade deve ser capaz de detectar mudanças na demanda de recursos e promover o provisionamento dinâmico de infraestrutura na Nuvem.

A proposta apresentada tem como objetivo o desenvolvimento de um modelo de provisionamento e alocação de recursos em ambientes de nuvem, que implementa o monitoramento e provisionamento de recursos utilizando como mecanismo de detecção e predição redes neurais artificiais. O escopo do trabalho está no mecanismo de provisionamento e sugestão de escalonamento de recursos, abstraindo-se a arquitetura complementar que se utiliza dos resultados e alimenta este sistema de predição.

O trabalho está organizado da seguinte forma: na seção 2 são apresentados trabalhos relacionados à computação em nuvem, provisionamento de recursos e qualidade de serviço; a seção 3 apresenta alternativas para o provisionamento dinâmico na Nuvem; na seção 4 é proposto um modelo de predição de recursos para a computação em nuvem, baseado em SLAs; na seção 5 são apresentadas as conclusões e trabalhos futuros.

2. Trabalhos relacionados

A computação em nuvem tornou-se a concretização de um objetivo buscado há muito tempo pela indústria de TI, da computação como uma utilidade. Este novo modelo tem a capacidade de mudar toda a forma de produção e implantação de software, além de guiar os rumos de como o *hardware* é desenvolvido e adquirido [Armbrust et al. 2009].

O advento da computação em nuvem não é um fato isolado, mas sim o êxito de várias tecnologias que têm sido combinadas para formar os ambientes computacionais em nuvem. Segundo [Foster et al. 2008] podem ser considerados três fatores principais que contribuíram para o desenvolvimento e aumento no interesse por computação em nuvem: 1) rápida redução nos custos de *hardware* e aumento no poder computacional e de armazenamento, o advento de tecnologias *multi-core* e os modernos supercomputadores constituídos de centenas de milhares de núcleos; 2) o crescimento exponencial na quantidade de dados científicos de simulação e instrumentação e aumento na publicação de dados na Internet; 3) a difusão da adoção da Web 2.0 e da arquitetura orientada a serviços.

Tabela 1. Níveis de serviço que a computação em nuvem é oferecida

<i>Categoria</i>	<i>Características</i>	<i>Tipo de Produto</i>	<i>Vendedores e Produtos</i>
SaaS	Os consumidores são providos de aplicações que são acessíveis em qualquer horário e qualquer lugar.	Aplicações Web e serviços (Web 2.0)	SalesForce.com (CRM), Clarizen.com (Gerenciamento de Projetos), Google Docs, Google Mail
PaaS	Os consumidores recebem uma plataforma para o desenvolvimento de aplicações hospedadas na Nuvem.	APIs para programação e frameworks.	Google AppEngine, Microsoft Azure, Manjrasoft Aneka.
IaaS/HaaS	Consumidores tem acesso a hardware ou armazenamento virtualizado. No topo do qual podem construir sua infraestrutura.	Máquinas Virtuais, gerenciamento de infraestrutura e de armazenamento.	Amazon EC2 e S3, GoGrid, Nirvanix.

A computação em nuvem é oferecida em diferentes camadas ou níveis. Cada nível apresenta uma abstração em relação ao anterior, visando oferecer recursos de maneira transparente, através de interfaces definidas, ocultando aos usuários a complexidade que está incluída no modelo. As principais camadas oferecidas são as de IaaS, PaaS e SaaS.

A tabela 1 [Buyya et al 2009] exhibe um resumo das características dos principais níveis em que a computação em nuvem está sendo oferecida e dividida. O escopo do trabalho está na camada de infraestrutura como serviço.

Amazon.com, um dos maiores provedores de ambientes de infraestrutura como serviço da atualidade [Youseff et al 2008] disponibiliza para seus usuários SLA relacionado à disponibilidade, ou seja, caso ocorram problemas de degradação de performance de processamento ou saturação da largura de banda o mecanismo de monitoramento informará o consumidor sobre o caso, sem tomar medidas de contingência proativas [Garfinkel 2007].

Eucalyptus é um *framework open source* de computação em nuvem focado na pesquisa acadêmica. Provê recursos para experimentos e estudo. Os usuários do Eucalyptus têm a possibilidade de inicializar, controlar e terminar máquinas virtuais integralmente [Endo et al 2009]. De acordo com [Nurmi et al. 2008] Eucalyptus proporciona um SLA baseado em regiões ou zonas, definindo o conceito de zonas, onde conjuntos de computadores com capacidades diferentes são agregados, sendo que o cliente pode definir em caso de violação do SLA migrar em tempo de execução sua instância virtual para um conjunto de recursos com maior capacidade.

De acordo com [Llorente 2009], o OpenNebula é um framework para computação em nuvem, utilizado para o desenvolvimento de nuvens privadas, públicas e híbridas. Em relação aos recursos de SLA o OpenNebula possibilita o escalonamento de novas instâncias virtuais no caso de violação de SLA.

Além destas soluções comerciais e científicas, existem outras técnicas, como a opacidade arquitetural (*architectural translucency*), proposta por [Stantchev e Malek 2006], que basicamente executa o provisionamento dinâmico a partir da replicação de componentes, criando um paralelismo na infraestrutura.

Pode-se observar, que as soluções existentes proporcionam certo nível de SLA e qualidade de serviço, porém não apresentam uma granularidade que possibilite ao consumidor definir requisitos estritos de performance, capacidade e demais atributos de QoS que podem ser necessários para certos tipos de aplicação na Nuvem. Essa carência acaba conduzindo à idéia de necessidade de uma solução que seja capaz de prover os requisitos esperados pelos consumidores. Esta habilidade de prever o comportamento dos sistemas na nuvem e prever uma nova configuração para atender esta mudança será abordada nas próximas seções.

3. Alternativas para o provisionamento dinâmico de recursos

Para atender a requisitos não funcionais que constituem os objetos do SLA, é necessário prover um mecanismo que possibilite avaliar o estado atual do serviço, e o volume de requisições e parâmetros de rede como latência, taxa de requisições, entre outros requisitos não funcionais e comparar estes dados com o SLA contratado.

Tabela 2. Atributos de QoS

Requisito	Especificação	Exemplo
Adaptabilidade	Proporcionar a reconfiguração dinâmica em tempo de execução.	Mudança nos requisitos exigem escalar mais uma máquina virtual ao ambiente.
Supervisão	Mensurar a QOS atual.	Monitorar parâmetros atuais.
Predição	Capacidade de prever comportamentos futuros baseados em dados atuais	Avaliar o processamento por um dado período e escalar mais recursos para atender à demanda crescente.
Granularidade	Possibilitar definição de parâmetros específicos para cada serviço	Definir requisitos em tempo de execução como tempo de resposta, throughput.

Avaliando o panorama de provedores e tecnologias de infraestrutura como serviço, na seção 2, pode-se observar que a maioria das soluções prove mecanismos de SLA e qualidade de serviço, porém estes ainda não conseguem expressar a granularidade e as especificidades que os consumidores corporativos e as aplicações necessitam. Faz-se necessário, portanto, uma avaliação dos algoritmos e mecanismos de provisionamento e de qualidade de serviço.

A granularidade referida é a capacidade de garantir a disponibilidade e confiabilidade dos dados e serviços hospedados na nuvem. Esta granularidade corresponde aos requisitos não funcionais característicos a cada aplicação ou serviço hospedado e depende da percepção e necessidade de cada cliente consumidor da nuvem.

A principal necessidade para a definição de tal algoritmo ou técnica é que este se enquadre nos requisitos de qualidade de serviço no contexto da computação em nuvem. A tabela 2 apresenta tais requisitos de qualidade de serviço necessários para um mecanismo de provisionamento dinâmico para a Nuvem. Estes requisitos foram extraídos a partir dos requisitos de SLA apontados por [Aib e Daheb 2007] como requisitos de SLA para redes IP, que podem ser aplicados para a computação em nuvem.

O levantamento de algoritmos de QoS da tabela 3, foi efetuado com o objetivo de buscar alternativas que já estão consolidadas e estáveis e que possam se adequar ao modelo de computação em nuvem. Foram avaliados os algoritmos de QoS utilizados para gerenciamento de rede, congestionamento, controle de fluxos e priorização de tráfego. Além destes foram buscadas soluções utilizadas em computação em grade e também propostas para a computação em nuvem [Balen e Westphall 2011].

A escolha por redes neurais deve-se ao fato de ser a solução que mais se enquadra nos requisitos propostos devido a sua capacidade de aprendizado com base no comportamento anterior e previsão de comportamento futuro. Este comportamento possibilita à nuvem um auto-controle evitando ou minimizando a ação de um administrador humano e reduzindo quebras nos contratos de SLA.

Tabela 3. Avaliação de algoritmos e técnicas de QoS

Técnica/Algoritmo	Adaptabilidade	Supervisão	Granularidade	Predição
FIFO	NÃO	NÃO	NÃO	NÃO
PQ	NÃO	NÃO	NÃO	NÃO
CQ	NÃO	NÃO	NÃO	NÃO
WFQ	NÃO	NÃO	NÃO	NÃO
CBWFQ	NÃO	NÃO	NÃO	NÃO
RED	NÃO	SIM	NÃO	SIM
WRED	NÃO	SIM	NÃO	SIM
FRED	NÃO	SIM	NÃO	SIM
Architectural Translucency	SIM	SIM	NÃO	SIM
Redes Neurais	SIM	SIM	SIM	SIM

É notório que existem outras soluções que abrangem o gerenciamento e manutenção de SLA e QoS em ambientes de computação em grade e em nuvem e que não foram abordados. Isto se deve ao fato que a solução proposta foca no uso de redes neurais como mecanismo de previsão e controle de uma arquitetura para computação em nuvem.

4. Modelo proposto

Faz-se necessária a definição de um modelo que implemente o provisionamento, monitoramento e previsão de recursos para ambientes de computação em Nuvem baseado em redes neurais. Segundo [Carvalho 2009], o desenvolvimento de uma aplicação em redes neurais deve seguir as etapas de coleta de dados e separação em conjuntos, configuração da rede, treinamento, teste e integração.

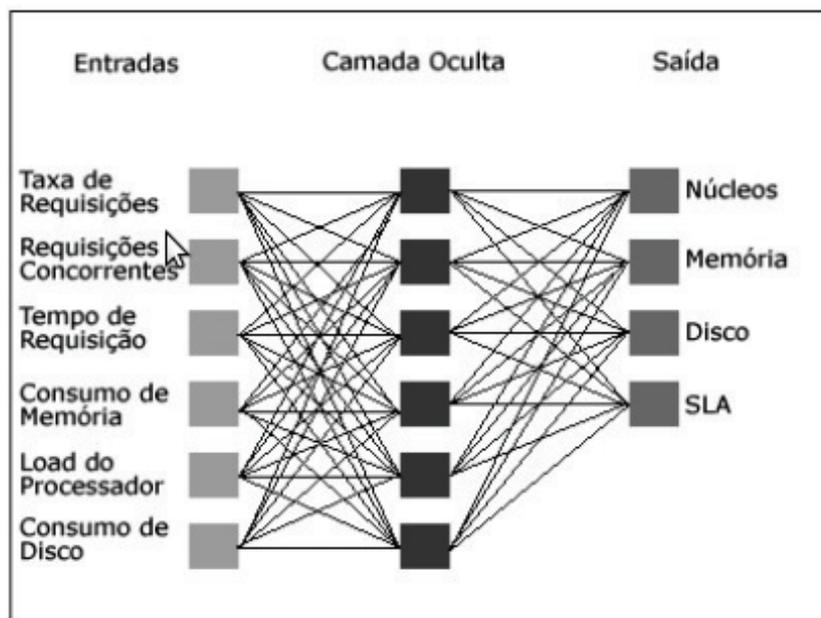


Figura 1. Rede neural artificial implementada pelo modelo

A topologia da rede contará conforme a figura 1 com seis entradas, sendo elas os requisitos não funcionais de QoS mais os indicadores do estado do sistema representados pelo *load* (carga) do sistema, uso de memória e de disco. A saída da rede contará com o resultado dos volumes de memória, núcleos de processamento e disco para o provisionamento, além do valor para o SLA, este valor é representado por uma variável booleana que determina se o SLA foi violado.

O cenário utilizado baseia-se em um ambiente de testes constituído por um servidor web virtual sob o virtualizador KVM, *Kernel-based Virtual Machine* [Kivity et al. 2007] que adiciona um monitor de máquinas virtuais (ou *hypervisor*) ao kernel Linux. Este servidor web por sua vez constitui-se em uma máquina virtual GNU/Linux com a distribuição Debian e com as seguintes características de hardware:

- Processador: 1 núcleo de 1.1 GHz
- Memória: 256 MB RAM
- Armazenamento: 8 Gb

Este ambiente virtual executa o servidor web Apache, e a linguagem de programação PHP. Este conjunto foi utilizado para simular um servidor web e suas requisições através da ferramenta de análise *ab* (apache benchmark). Com base nos relatórios obtidos foram gerados dados para alimentação da rede.

A partir deste cenário foi feita a modelagem da rede neural de predição e monitoramento.

As entradas da rede são os atributos de qualidade de serviço e o estado dos recursos de performance do sistema avaliado. Estes atributos que atuam sobre os recursos do sistema foram definidos levando-se em conta os valores que têm influência direta na performance e na disponibilidade do sistema. Atributos definidos como entrada da rede neural:

- *Taxa de requisições*: a taxa de requisições é a quantidade de requisições por unidade de tempo. No caso a métrica utilizada foi requisições por segundo (rps).
- *Requisições concorrentes*: é o volume de requisições simultâneas que o ambiente está recebendo em um momento.
- *Tempo de requisição*: é o tempo necessário para a requisição ser processada e exibir seu retorno. Mensurada em segundos.
- *Consumo de memória*: é o volume de memória dinâmica volátil (RAM) consumida pelas requisições no sistema em um momento específico. Exclui a memória utilizada pelo sistema operacional.
- *Load do processador*: o *load* de processador é o uso da capacidade de processamento. Os processos em um sistema ficam em espera ou em execução. O *load* representa portanto, a soma de todos os processos que estão atualmente em execução mais os processos que estão aguardando na fila de execução para serem executados. O *load* pode variar de 0 (sem processos rodando ou em fila) a 1.0, ou seja, a fila de execução está cheia, mas não existem processos aguardando o término de um processo para sua execução e valores maiores que 1.0 para o *load* significam que a fila de execução está cheia e existem processos aguardando execução.
- *Consumo de disco*: consumo em megabytes das requisições sobre o espaço de armazenamento do sistema.

Após o mapeamento das entradas do sistema, faz-se necessário efetuar a definição das saídas esperadas. Com o intuito de assegurar o SLA e demonstrar o provisionamento de recursos de forma dinâmica, as saídas devem avaliar as entradas e suas saídas devem demonstrar se houve quebra ou não do SLA, e caso houver sugerir os recursos necessários para que um novo ambiente seja criado e instanciado paralelamente ao que não atende mais aos requisitos de SLA.

- *Núcleos*: determina a quantidade de núcleos de processamento que devem ser alocados para a nova instância - ou instâncias - virtuais.
- *Memória*: determina a quantidade de memória a ser alocada para o novo ambiente.
- *Disco*: informa a quantidade de disco que deve ser alocada ao novo ambiente.

- *SLA*: valor booleano (verdadeiro ou falso) que determina se o SLA foi violado (verdadeiro) ou não (falso).

Uma vez definido o modelo, a próxima seção vai demonstrar quais os resultados obtidos a partir de testes que tentam simular cenários que exigem o uso de provisionamento de forma dinâmica demonstrando assim a efetividade deste trabalho.

5. Resultados

Para agregar maior confiabilidade à proposta, foi desenvolvido o modelo utilizando-se o simulador de redes neurais EasyNN plus. Após a criação da rede neural na ferramenta, visualizada na figura 2 procedeu-se ao treinamento da rede com dados históricos como entrada, e hipóteses esperadas para a saída.

Foi utilizado o algoritmo *backpropagation*, que possui uma estrutura direta multicamada com neurônios estáticos e seu aprendizado é supervisionado. Optou-se pelo algoritmo de *backpropagation* por ser uma solução consolidada, estável e amplamente utilizada no estudo e desenvolvimento de redes neurais.

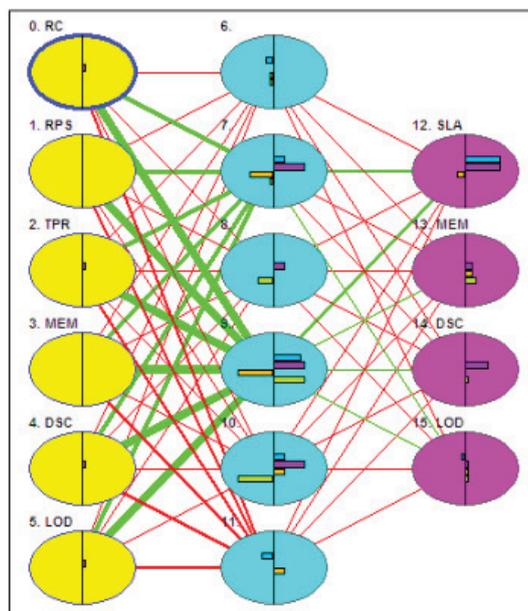


Figura 2. Rede neural no software EasyNN

Os dados hipotéticos foram gerados a partir de um servidor web Apache físico executando uma aplicação web, conforme definição na seção 4. Realizou-se testes de performance com o uso da ferramenta ab – Apache Benchmark, que possibilita gerar um número aleatório de requisições e retorna os tempos de resposta e custo de cada requisição.

Os resultados constituem-se na análise das consultas efetuadas na rede neural após seu treinamento. O objetivo desta avaliação é observar o comportamento da rede em relação a entradas aleatórias, diferentes dos valores utilizados no treinamento.

Após a finalização do treino, a rede está pronta para ser utilizada formalmente na avaliação de novas situações não previamente treinadas. Esta avaliação é o que irá

informar se as redes neurais podem ser utilizadas no contexto de computação em nuvem.

O treinamento englobou a realização de 36.976 ciclos de aprendizado, atingindo ao final do ciclo um erro de aprendizado médio de 0.000194. O erro resultante é proporcional aos ciclos de treinamento realizados. Um ciclo de aprendizado maior resultaria em um erro menor, o que torna a rede mais efetiva nas suas inferências.

Tabela 4. Casos de Teste

Atributo	Valor Cenário 1	Valor Cenário 2	Valor Cenário 3
Requisições Concorrentes	85	287	958
Requisições por Segundo	225	163	328
Tempo por requisição	62	193	230
Uso de memória	64.2	287	434
Uso de disco	85	287	958
Load	0.8	3.8	10.4
SLA Esperado	0	1	1
Memória Provisionada	0	347,84	562,56
Disco Provisionado	0	4383	5054
Núcleos Provisionados	0	4	10

Após o treinamento da rede foram efetuados testes utilizando-se por base alguns cenários de teste. Estes cenários constituem-se em dados de entrada que são introduzidos nas unidades de entrada da rede, que retorna os valores de saída correspondentes ao provisionamento efetuado. A tabela 4 exhibe os cenários de teste adotados e os valores desejados para a saída.

Os valores dos cenários foram inseridos na ferramenta de consulta do software EasyNN para coleta dos resultados previstos pela rede. Os resultados estão presentes na tabela 5. Esta tabela apresenta os valores de saída da rede neural.

Avaliando as entradas e saídas esperadas na tabela 4 e as saídas proporcionadas pela rede na tabela 5 verifica-se que as saídas não foram representativas em todos os casos. Isto indica que o período ou o conjunto de hipóteses utilizados no treinamento da rede não foram representativos para todas as possibilidades de saída. Mesmo com esta discrepância a rede previu corretamente a necessidade de provisionamento, tendo porém dimensionado os valores abaixo do que fora definido nos casos de teste.

Tabela 5. Resultado da rede neural para os cenários de teste.

Atributo	Valor Cenário 1	Valor Cenário 2	Valor Cenário 3
SLA	0	1	1
Memória Provisionada	0.0002	431.2512	408.3885
Disco Provisionado	0	4383	4557
Núcleos Provisionados	0	3	9

Com base nos resultados da tabela 5, os cenários serão avaliados individualmente. O Cenário 1, não violou o SLA. Pode-se constatar que os valores

esperados pela rede neural foram atingidos, sendo que a rede neural está devidamente treinada para identificar a violação ou não do SLA. Como neste caso não devem ser provisionados recursos, o resultado da rede também é válido de acordo com o esperado, ou seja, os valores ficaram em 0 ou muito próximos disso.

Os cenários 2 e 3 apontaram anomalias em relação ao resultado esperado. A validação do SLA ocorreu satisfatoriamente, sendo que nos dois casos esperava-se a quebra do contrato, o que ocorreu. Porém os valores previstos não seguiram o esperado pela definição da tabela 4.

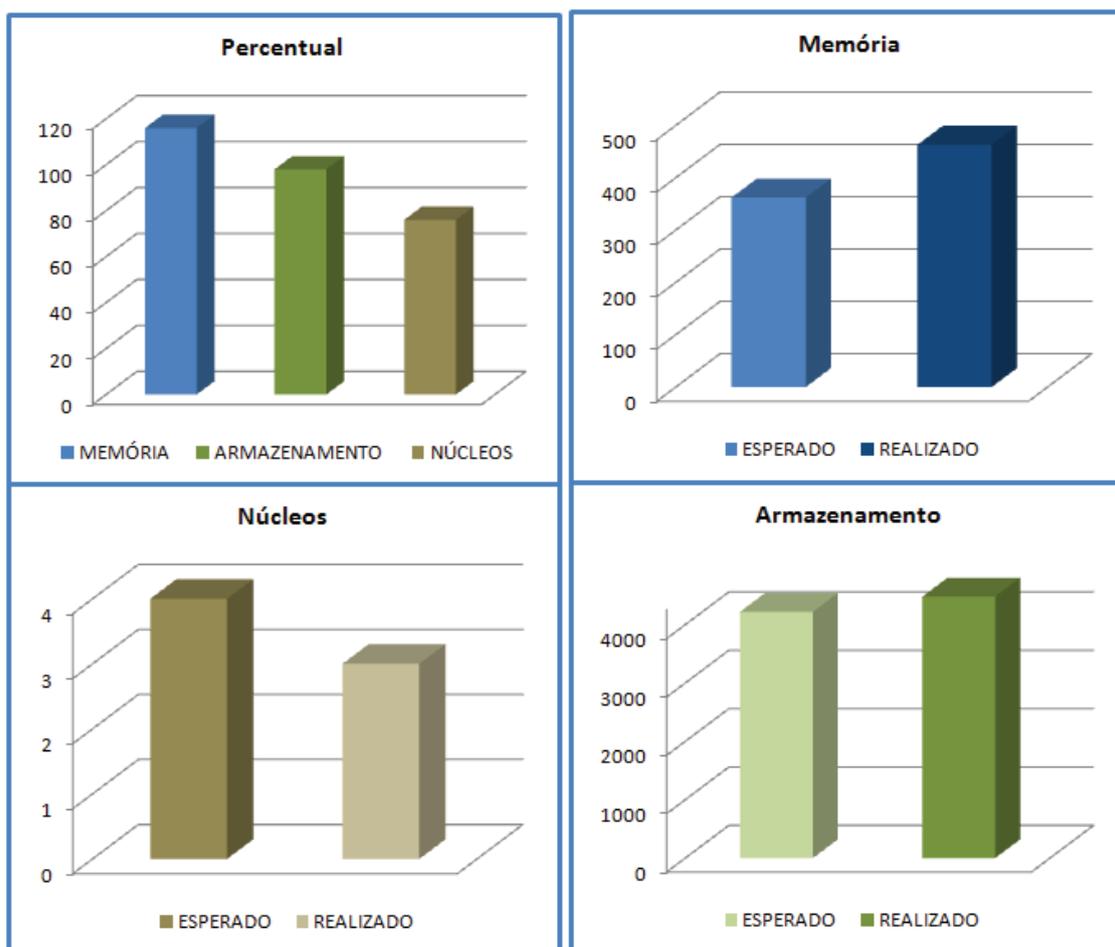


Figura 3. Gráficos de resultado do cenário de teste 2.

As figuras 3 e figura 4 apresentam uma representação gráfica dos cenários de teste 2 e 3, que apresentaram violação do SLA e portanto, fizeram com que a rede neural provesse uma sugestão de alocação de recursos para atender à demanda crescente por recursos. Os conjuntos de quatro gráficos estão agrupados e representam respectivamente: o gráfico esquerdo superior representa o percentual provisionado em relação ao esperado. Considera-se 100% como o valor esperado, sendo que a previsão pode ser superior ou inferior ao esperado. Os demais gráficos mostram os recursos previstos e os recursos esperados quantitativamente.

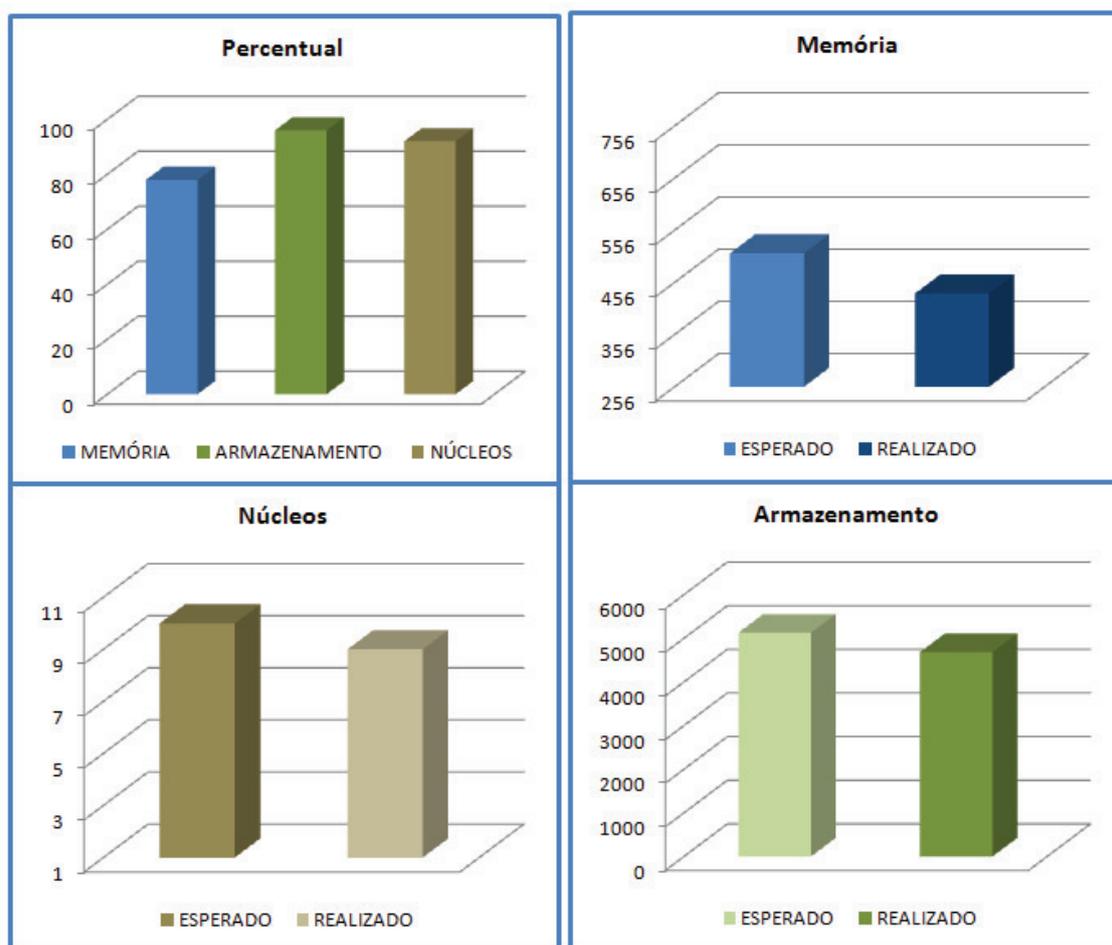


Figura 4. Gráficos de provisionamento do cenário 3.

Os gráficos e testes efetuados sob a rede neural apresentaram, no geral, resultados satisfatórios. O provisionamento de recursos foi efetuado, e variações no SLA detectadas. Pode-se atribuir a variação no provisionamento devido a limitações nos recursos computacionais e na coleta de dados de maior amplitude, além de possivelmente a rede necessitar de um treinamento maior e com um conjunto de dados mais abrangente.

Dessa forma, os resultados demonstram que a proposta abordada mostrou-se válida, no contexto do trabalho, redes neurais podem ser uma alternativa ao provisionamento em ambientes com constante variabilidade no desempenho e com uma ampla variação nos atributos mensurados, mais estudos são necessários, com a formulação de uma arquitetura em nuvem completa, ou a integração de redes neurais às existentes para verificar a eficácia do modelo em simulações ou ambientes reais. A computação como utilidade, e a computação em nuvem por extensão podem se beneficiar do uso de redes neurais para predição do comportamento das suas instâncias.

7. Conclusão

O provisionamento dinâmico de recursos segue sendo um desafio para a computação em nuvem. A utilização de redes neurais para o provisionamento e detecção de violação de SLA é uma das soluções propostas pela Universidade de Berkeley em seu relatório técnico sobre a computação em nuvem [Armbrust et al. 2009], onde o uso pervasivo e agressivo de máquinas de aprendizado estatísticas como ferramenta de diagnóstico, previsão e provisionamento de recursos é utilizado de forma a possibilitar escalonamento dinâmico, reação automática a problemas de desempenho e o gerenciamento automático dos sistemas em nuvem [Armbrust et al. 2009].

A implementação e testes mostraram o quanto o treinamento da rede tem influência nos seus resultados. O conjunto final de testes apresentando erro mínimo para o conjunto de dados de entrada foi considerado satisfatório.

Abrem-se diversas possibilidades para trabalhos futuros que estendam ou utilizem o escopo proposto no trabalho. A utilização de outros cenários e requisitos para a modelagem da rede pode ser proposta. Uma arquitetura em nuvem que contemple como núcleo de predição e provisionamento redes neurais pode ser desenvolvida.

No campo da economia da Nuvem e computação verde pode ser proposto uma rede neural que busque otimizar os recursos do provedor e adequar às necessidades dos clientes, visando a melhor relação de custo sem violar a qualidade desejada.

Outra proposta para trabalhos futuros é o desenvolvimento de um modelo neural integrado com simuladores de nuvem, como o CloudSim [Calheiros et al. 2009]. Esta integração possibilitaria mensurar a eficácia das redes neurais no provisionamento de ambientes próximos à realidade, fazendo parte de toda a arquitetura de provisionamento.

O futuro da computação em nuvem pode utilizar-se de máquinas de aprendizado estatístico, com as redes neurais desempenhando um papel importante no provisionamento, predição e avaliação da Nuvem. O campo é fértil para estudos e o presente trabalho objetivou propor este modelo integrando redes neurais à Nuvem.

Referências

- Aib, I., Daheb, B. (2007) "Management, Control and Evolution of IP Networks". In: ISTE 2007. Cap. "SLA Driven Network Management". 2007.
- Armbrust et al. (2009) "Above the Clouds: A Berkeley View of Cloud Computing". [S.l.], February 2009.
- Balen, D. O., Westphall, C. B., Westphall, C. M. (2011) "Experimental Assessment of Routing for Grid and Cloud". International Conference on Networking - ICN 2011.
- Buyya, R. et al. (2009) "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility". In: Future Generation Computer Systems, Elsevier B. V.
- Buyya, R. et al. (2010) "InterCloud: Utility-Oriented Federation of Cloud Computing Environments for Scaling of Application Services," Proceedings of the 10th International Conference on Algorithms and Architectures for Parallel Processing, LNCS, Springer, Germany, 2010., pp. 1–20, 2010.

- Calheiros R., et al. (2009). “CloudSim: A Novel Framework for Modeling and Simulation of Cloud Computing” Infrastructures and Services. GRIDS Laboratory 2009.
- Campedel B. (2007). “Management, control and evolution of IP networks”. In: . [S.l.]: ISTE, 2007. cap. Quality of Service: The Basics. 2007.
- Carvalho A.P. de L., (2009). “Redes Neurais Artificiais”. 2009. <http://www.icmc.usp.br/andre/research/neural/index.htm>.
- Chaouchi H., (2007) “Management, control and evolution of ip networks”. In: .[S.l.]: ISTE, 2007. cap. Mobility and Quality of Service. 2007.
- Endo, P.T., Gonçalves, G. E., Kelner, J., Sadok, D. (2010). “A Survey on Open-source Cloud Computing Solutions”. SBRC 2010, WCGA pages 3-16, 2010.
- Foster et al. (2008). “Cloud computing and grid computing 360-degree compared”. Elsevier, 2008.
- Garfinkel S. L. (2007). “An Evaluation of Amazon’s Grid Computing Services: EC2, S3 and SQS”. [S.l.], 2007.
- Grimes, J. M., Jaeger, P. T., e Lin J. (2008). “Weathering the storm: The policy implications of cloud computing”. 2008.
- Jin, L., Machiraju, V., Sahai, A. (2002). “Analysis on service level agreement of web services”. HPL-2002-180, 2002.
- Kivity, A., Kamay, Y., Laor, D., Lublin, U., Liguori, A., (2007) “kvm: The Linux Virtual Machine Monitor”. In: OLS, 2007. Ottawa, CA.
- Llorente, I.M. et al. (2009). “Open Nebula Cloud Computing Framework”. In: IEEE (Ed.), 2009.
- Lewicki, P., Hill, T. (2008). “Statistics methods and applications”. In: Statsoft, 2008. cap. Neural Networks. 2008.
- Nurmi D. et al. (2008). “Eucalyptus: A Technical Report on an Elastic Utility Computing Architecture Linking Your Programs to Useful Systems”. In: <http://www.eucalyptus.com>, 2008.
- Russel S., e Norvig, P. (2003). “Inteligência Artificial”. In: Inteligência Artificial. Campus, 2003. cap. Métodos Estatísticos de Aprendizagem. 2003.
- Stantchev e Malek M.(2006). “Architectural translucency in service-oriented architectures. IEEE Proceedings, p. 31–37, 2006.
- Youseff, L, Butrico, M., Da Silva, D. (2008) “Toward a Unified Ontology of Cloud Computing”. In: Grid Computing Environments Workshop, 2008. GCE '08, Austin, TX, USA.