Inferência de Desempenho: Uma Nova Abordagem para o Planejamento da Capacidade de Aplicações na Nuvem

Marcelo Gonçalves, Matheus Cunha, Américo Sampaio, Nabor C. Mendonça

[1]Programa de Pós-Graduação em Informática Aplicada (PPGIA)
Universidade de Fortaleza (UNIFOR)
Av. Washington Soares, 1321, Edson Queiroz, CEP 60811-905 Fortaleza, CE

{marcelocg,mathcunha}@gmail.com,{americo.sampaio,nabor}@unifor.br

Resumo. *Este trabalho propõe uma nova abordagem para apoiar o planejamento da capacidade de aplicações em nuvens que oferecem infraestrutura-como-serviço (IaaS). A abordagem proposta tem como premissa a existência de uma relação de capacidade entre diferentes configurações de recursos de um dado provedor de nuvem IaaS, com a qual é possível prever (ou "inferir"), com alta precisão, o desempenho esperado de uma aplicação para certas configurações de recursos e cargas de trabalho, tendo com base o desempenho da aplicação observado para outras configurações de recursos e cargas de trabalho neste mesmo provedor. Resultados empíricos preliminares, obtidos a partir da avaliação do desempenho de uma popular aplicação de* blogging *(WordPress) em um provedor de nuvem público (Amazon EC2), mostram que a nova abordagem consegue reduzir significativamente (acima de 85%) o número total de cenários de implantação da aplicação que precisam de fato ser avaliados na nuvem.*

Abstract. *This work proposes a novel approach to support application capacity planning in infrastructure-as-a-service (IaaS) clouds. The proposed approach relies on the assumption that there exists a capacity relation between different resource configurations offered by a given IaaS cloud provider, enabling one to predict (or "infer"), with high accuracy, an application's expected performance for certain resource configurations and workloads, based upon its observed performance for other resource configurations and workloads in that same provider. Preliminary empirical results, obtained from evaluating the performance of a well-known blogging application (WordPress) in a public cloud provider (Amazon EC2), show that the proposed approach can significantly reduce (over 85%) the total number of application deployment scenarios that need to be effectively tested in the cloud.*

1. Introdução

Um dos principais desafios enfrentados pelos usuários de nuvens que oferecem infraestrutura-como-serviço (IaaS) é planejar adequadamente a capacidade dos recursos da nuvem necessários para atender as demandas específicas de suas aplicações [Menascé and Ngo 2009]. Parte desse desafio envolve tentar descobrir a melhor maneira de implantar a aplicação na nuvem, considerando os vários tipos de recursos (em particular, máquinas virtuais) oferecidos pelo provedor, sob a perspectiva de diferentes requisitos e critérios de qualidade [Gonçalves Junior et al. 2015].

Em geral, provedores de nuvens IaaS cobram seus usuários em função do tempo de utilização dos recursos solicitados, cujos preços variam conforme a capacidade (normalmente medida por características técnicas como quantidade de núcleos de processamento, tamanho de memória e espaço de armazenamento) de cada recurso. Dessa forma, para calcular o custo de operação de uma aplicação na nuvem, é preciso estimar ou medir como a aplicação responderá a diferentes níveis de demanda, em termos de indicadores de desempenho como tempo de resposta ou vazão, quando executada sob diferentes configurações e perfis de máquinas virtuais. Na prática, isso significa que cabe ao usuário da nuvem identificar, dentre as possíveis configurações de máquinas virtuais ofertadas por um ou mais provedores de nuvem, aquelas de menor custo capazes de executar a aplicação mantendo-se níveis satisfatórios para os indicadores de desempenho.

Um grande problema começa a se desenhar para o usuário da nuvem ao seguir essa abordagem: a fase de avaliação da aplicação pode atingir patamares elevados de tempo e custo, em razão das necessidades de variação da demanda, da arquitetura de implantação e das configurações de recursos utilizadas para hospedar cada camada da aplicação [Silva et al. 2013]. Ainda que certos provedores IaaS ofereçam descontos ou pacotes de horas grátis para novos clientes, em geral esses incentivos, por estarem limitados a máquinas de pequeno porte, são insuficientes para suportar a carga de uma aplicação real em produção. Assim, executar uma aplicação real, tipicamente implantada em arquitetura de várias camadas [Jayasinghe et al. 2011], em máquinas virtuais de tamanho considerável e por longos períodos de tempo, apenas para estudar o seu comportamento, pode se traduzir em um custo alto que dificulte ou até mesmo inviabilize o próprio projeto de migração dessa aplicação para a nuvem [Beserra et al. 2012].

Vários trabalhos já foram propostos com o intuito de apoiar o planejamento da capacidade de aplicações em nuvens IaaS. Em linhas gerais, esses trabalhos podem ser classificados de acordo com duas abordagens distintas quanto à estratégia de avaliação do desempenho da aplicação. Trabalhos que seguem a primeira abordagem, referenciada neste trabalho como *abordagem preditiva*, visam estimar ou simular o desempenho esperado da aplicação para determinadas configurações de recursos e determinados níveis de carga, sem necessariamente ter que implantá-la na nuvem [Malkowski et al. 2010, Li et al. 2010, Li et al. 2011, Fittkau et al. 2012, Jung et al. 2013]. Apesar do baixo custo oferecido aos usuários, que não precisam pagar por recursos de nuvem durante a fase de avaliação, esse trabalhos têm como maior limitação a ainda baixa precisão das técnicas de predição de desempenho, particularmente daquelas baseadas em simulação [Fittkau et al. 2012]. Já os trabalhos que fazem parte da segunda abordagem, aqui referenciada como *abordagem empírica*, têm como objetivo medir o desempenho real da aplicação através de sua efetiva implantação na nuvem e da realização de testes de carga [Jayasinghe et al. 2012, Silva et al. 2013, Cunha et al. 2013a, Scheuner et al. 2014]. Por executarem a aplicação no próprio ambiente de nuvem, esses trabalhos conseguem resultados significativamente mais precisos no que diz respeito à seleção das melhores configurações de recursos para cargas de trabalho específicas. No entanto, uma limitação importante desses trabalhos é a necessidade de se testar exaustivamente uma grande quantidade de configurações de recursos e cargas de trabalho, implicando em altos custos durante a fase de avaliação.

Este trabalho propõe uma nova maneira de apoiar os usuários de nuvens IaaS a

identificarem as melhores (i.e., mais baratas) configurações de recursos capazes de satisfazer as demandas específicas de suas aplicações. A nova abordagem tem como premissa a existência de uma relação de capacidade entre diferentes configurações de recursos oferecidas por um dado provedor de nuvem, com a qual é possível prever (ou "inferir"), com alta precisão, o desempenho esperado da aplicação para determinadas configurações de recursos. A predição ou inferência é realizada com base no desempenho observado da aplicação para outras configurações de recursos e cargas de trabalho no mesmo provedor. Por exemplo, se a aplicação atendeu satisfatoriamente a demanda para uma configuração de recursos de determinada capacidade sob uma determinada carga de trabalho, é muito provável que ela também vá atendê-la para outras configurações de maior capacidade sob a mesma carga de trabalho. Analogamente, se a aplicação não atendeu a demanda para uma determinada configuração de recursos sob uma determinada carga de trabalho, muito provavelmente ela também não irá atendê-la para a mesma configuração sob cargas de trabalho maiores. Através do uso de inferência, a abordagem permite avaliar uma ampla variedade de cenários de implantação da aplicação, sendo que apenas uma parte relativamente pequena desses cenários precisa de fato ser implantada e executada na nuvem. Dessa forma, a abordagem consegue obter o melhor das duas abordagens previamente citadas, produzindo resultados de alta precisão (característicos da abordagem empírica) mas com significativa redução de custo (característica da abordagem preditiva).

A próxima seção apresenta um novo processo de avaliação de capacidade para aplicações na nuvem, fundamentado no conceito de inferência de desempenho. A Seção 3 descreve os resultados de uma avaliação preliminar do novo processo envolvendo a implantação de uma aplicação real em um provedor de nuvem IaaS público. A Seção 4 compara o processo proposto com outros trabalhos relacionados. Por fim, a Seção 5 oferece algumas conclusões e sugestões para trabalhos futuros.

2. Processo de Avaliação de Capacidade por Inferência de Desempenho

2.1. Conceitos e Terminologia

Antes de apresentarmos o processo, é necessário definirmos alguns conceitos importantes relacionados ao domínio da avaliação da capacidade de aplicações na nuvem (ver Tabela 1). A definição desses conceitos também serve para estabelecer a terminologia que será utilizada na descrição do processo, feita a seguir.

2.2. Dados de Entrada

O principal dado de entrada esperado pelo processo é o valor de referência (ou SLO), o qual será usado para determinar se a aplicação atingiu os requisitos mínimos de desempenho exigidos em cada cenário de execução. Além do SLO, o processo precisa também conhecer quais são as cargas de trabalho sob as quais o desempenho da aplicação deverá ser avaliado. Outro dado importante que deve ser passado como entrada para o processo é o espaço de implantação da aplicação. Para isso, o processo deve ser alimentado com três parâmetros: (i) uma lista de tipos de máquinas virtuais fornecidos pelo provedor no qual deseja-se hospedar a aplicação; (ii) a quantidade máxima de máquinas virtuais de cada tipo que irá compor cada configuração a ser avaliada; e (iii) um ou mais critérios para estabelecimento das relações de capacidade entre as configurações do espaço de implantação. A Seção 3 ilustra alguns critérios que podem ser usados para este fim.

Tabela 1. Conceitos e terminologia utilizados no artigo.

Conceito	Definição
Aplicação sob teste	Um sistema computacional, possivelmente implementado em uma arquitetura multicamadas, para o qual se deseja observar o comportamento em um ambiente de computação em nuvem e ao qual estão associadas uma ou mais *métricas de desempenho*.
Métrica de desempenho	Uma característica ou comportamento mensurável de forma automatizada e comparável a um *valor de referência*, capaz de indicar o grau de sucesso de uma execução da aplicação sob teste. É dependente do domínio da aplicação. Ex.: tempo de resposta, quadros por segundo.
Valor de referência	Um valor predefinido como minimamente aceitável para uma métrica de desempenho após uma execução da aplicação sob teste. Este valor, também referenciado neste trabalho como SLO (*Service Level Objective*), serve como base de comparação para que se classifique a aplicação como capaz de ser executada em uma certa *configuração* de máquinas virtuais e sob uma certa *carga de trabalho*.
Carga de trabalho	Representa o tamanho da demanda que será imposta à aplicação sob teste em uma execução. Sua unidade de medida é dependente do domínio da aplicação. Ex.: tamanho dos arquivos de entrada para uma aplicação de compactação de arquivos, quantidade de usuários concorrentes para uma aplicação web, etc.
Tipos de máquinas virtuais	Classificam as máquinas virtuais fornecidas por um provedor conforme suas características técnicas (e.g., núcleos de processamento, tamanho de memória, espaço em disco), permitindo que o provedor de nuvem mantenha uma linha de produtos discreta e finita.
Categorias de máquinas virtuais	Agrupam os tipos de máquinas virtuais de um provedor de acordo com suas características técnicas, plataforma e/ou arquitetura de hardware e a natureza do uso a que se destinam. Ex.: categorias que priorizam consumo de memória, acesso a disco, processamento gráfico, etc.
Configuração	Um conjunto de máquinas virtuais de um mesmo tipo e, portanto, de uma mesma categoria. *Configurações* são usadas para implantar uma ou mais camadas arquiteturais (ex.: apresentação, negócio, persistência) da aplicação sob teste.
Espaço de implantação	Denota um conjunto limitado de configurações de máquina virtuais nas quais a aplicação sob teste será implantada e executada durante uma sessão de avaliação.
Relações de capacidade	Relativizam o poder computacional das diversas configurações que compõem o espaço de implantação. As *relações de capacidade* definem um grafo orientado sobre o espaço de implantação onde os vértices correspondem às configurações e as arestas indicam a superioridade ou inferioridade (dependendo da direção da aresta) de uma configuração em relação a outra em termos de poder computacional.
Níveis de capacidade	Estabelecem uma hierarquia sobre as relações de capacidade definidas entre as configurações do espaço de implantação. Nessa hierarquia, configurações classificadas em um mesmo nível de capacidade seriam equivalentes (ou indistinguíveis) em termos de poder computacional.

2.3. Atividades

As principais atividades executadas pelo processo de avaliação de capacidade são ilustradas no diagrama da Figura 1. Nesse diagrama, atividades destacadas com o rótulo ≪A≫ são abstratas, devendo ser customizadas pelos usuários do processo de acordo com diferentes estratégias de avaliação (descritas na Seção 2.4). As demais atividades são concretas, sendo executadas independentemente da aplicação sob teste ou da estratégia de avaliação utilizada.

A execução do processo acontece de forma cíclica, com as atividades agrupadas em quatro fases distintas: (i) seleção do cenário de execução da aplicação; (ii) execução da aplicação; (iii) inferência de desempenho; e (iv) seleção do próximo cenário de execução. Cada uma dessas fases será detalhada a seguir.

2.3.1. Seleção do cenário de execução

A primeira atividade dessa fase é a escolha de uma carga de trabalho. Essa é uma atividade abstrata, significando que diferentes estratégias podem ser empregadas nessa escolha, por exemplo, selecionando um carga de trabalho maior ou menor dentre aquelas fornecidas como dados de entrada ao processo. Depois de selecionar a carga inicial, o processo seleciona uma categoria de máquinas virtuais. No caso da categoria, a ordem ou método utilizado na escolha é irrelevante para o processo, uma vez que todas as catego-

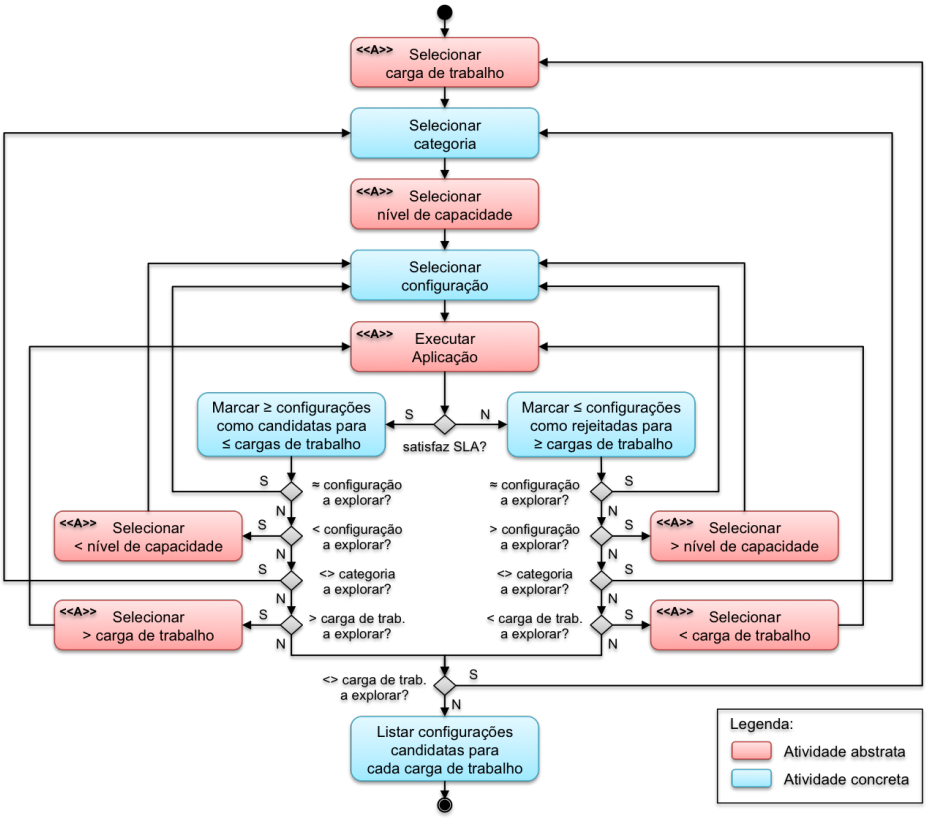


Figura 1. Diagrama de atividades do processo de avaliação de capacidade.

rias do espaço de implantação deverão ser avaliadas. Em seguida, o processo seleciona um nível de capacidade dentre aqueles presentes no espaço de implantação. Essa também é uma atividade abstrata, uma vez que níveis de capacidade mais acima ou mais abaixo na hierarquia podem ser escolhidos, a depender da estratégia de avaliação utilizada. Por fim, o processo seleciona uma configuração do nível de capacidade previamente selecionado. A ordem de seleção das configurações também é irrelevante, uma vez que todas as configurações daquele nível de capacidade devem ser avaliadas.

2.3.2. Execução da aplicação

Uma vez escolhidos uma carga de trabalho, uma categoria, um nível de capacidade e uma configuração, o processo está apto a executar a aplicação na nuvem. A execução da aplicação também é uma atividade abstrata do processo, pois depende de uma série de fatores que são específicos de cada aplicação ou plataforma de nuvem, como as tecnologias necessárias parar implantar os componentes da aplicação na nuvem bem como para submetê-los aos níveis de carga de trabalho desejados. Após a execução da aplicação, o processo analisa o resultado obtido e passa para a fase de inferência de desempenho.

2.3.3. Inferência de desempenho

Nesta fase, o processo se bifurca, atingindo seu primeiro ponto de decisão. A partir da análise do resultado da execução, que é feita comparando-se os indicadores obtidos para a

métrica de desempenho utilizada frente ao valor de referência (SLO) desejado, o processo determina se a aplicação é ou não capaz de atender à demanda imposta sobre ela com a atual configuração. Se a aplicação satisfaz o SLO, o processo assinala a configuração atual como uma *configuração candidata* para o atual nível de carga. Do contrário, o processo assinala a configuração atual como uma *configuração rejeitada* para esse nível de carga.

É neste momento que a abordagem de inferência de desempenho, proposta originalmente neste trabalho, entra em ação. Com base nas relações de capacidade presentes no espaço de implantação, o processo pode "inferir" o provável desempenho da aplicação para outras configurações e cargas de trabalho ainda não avaliadas. Se o processo identificou que uma certa configuração consegue satisfazer a demanda imposta à aplicação sob uma certa carga de trabalho, intuitivamente qualquer outra configuração de maior poder computacional também será capaz de fazê-lo sob a mesma carga de trabalho. Similarmente, é intuitivo concluir que a mesma configuração também será capaz de satisfazer o SLO da aplicação sob cargas de trabalho menores. Assim, usando as informações sobre as relações de capacidade existentes entre as configurações do espaço de implantação, o processo também assinala como candidatas para o atual nível de carga todas as outras configurações identificadas como sendo de "maior capacidade" que a configuração atual de acordo com o espaço de implantação. Da mesma forma, o processo também assinala a configuração atual como candidata para todos os níveis de carga inferiores ao nível de carga atual.

O caso em que a configuração atual não satisfaz o SLO da aplicação é tratado de modo análogo. Nesse caso, o processo assinala como rejeitadas para o atual nível de carga todas as outras configurações identificadas como sendo de "menor capacidade" que a configuração atual de acordo com o espaço de implantação. O mesmo acontece com a configuração atual, que também é assinalada como rejeitada para todos os outros níveis de carga superiores ao nível de carga atual.

2.3.4. Seleção do próximo cenário

Após a fase de inferência de desempenho, o processo seleciona os elementos que comporão o próximo cenário de execução a ser avaliado, ou encerra sua execução, caso não haja mais cenários a explorar. Nesse caso, o processo produz, como saída, uma lista contendo todas as configurações assinaladas como candidatas para cada carga de trabalho avaliada, em ordem crescente de preço.

A seleção do próximo cenário inclui a escolha de uma nova configuração do atual nível de capacidade, a escolha de um novo nível de capacidade (maior ou menor que o nível de capacidade atual), a escolha de uma nova categoria, ou a escolha de uma nova carga de trabalho (maior ou menor que o nível de carga atual). As escolha de um novo nível de capacidade ou de uma nova carga de trabalho vai depender do resultado da execução da aplicação no cenário atual, na medida em que o processo irá tentar diminuir (aumentar) o poder computacional da configuração atual ou, alternativamente, aumentar (diminuir) o nível de carga atual, caso a aplicação tenha alcançado (ou não) o SLO desejado. Por essa razão, essas escolhas também são consideradas atividades abstratas, a serem definidas como parte da customização do processo com diferentes estratégias de avaliação.

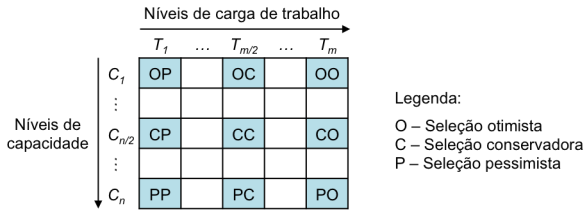


Figura 2. Heurísticas para seleção de configurações e cargas de trabalho.

2.4. Estratégias de Avaliação

Conforme mencionado anteriormente, todas as atividades abstratas do processo (com exceção da atividade de execução da aplicação na nuvem) devem ser customizadas de acordo com diferentes estratégias de avaliação. Essas atividades incluem, basicamente, a escolha de cargas de trabalho e níveis de capacidade. Tais escolhas influenciam diretamente a maneira através da qual o processo explora o espaço de implantação, tendo um forte impacto no alcance da inferência de desempenho.

Como exemplo, considere o caso de um espaço de implantação onde nenhuma configuração é capaz de atender a demanda da aplicação sob qualquer nível de carga. Nesse caso, iniciar o processo de avaliação pelas configurações do nível de capacidade mais baixo sob cargas de trabalho maiores não seria uma boa estratégia, uma vez que o número de configurações e cargas de trabalho para os quais o desempenho esperado da aplicação poderia ser inferido seria muito pequeno. Por outro lado, iniciar o processo pelas configurações de nível de capacidade mais alto sob cargas de trabalho menores seria um estratégia muito melhor, já que assim seria possível inferir o desempenho da aplicação para praticamente todas as outras configurações e todas as outras cargas de trabalho, representando uma grande economia de tempo e custo.

Esses dois extremos ilustram bem o desafio de se escolher os cenários de execução mais promissores do ponto de vista da inferência de desempenho. A fim de enfrentar esse desafio, este trabalho introduz o conceito das *heurísticas de seleção*, que agregam táticas a serem observadas no momento em que o processo, via alguma estratégia de avaliação, precisa escolher uma nova configuração ou uma nova carga de trabalho para compor um novo cenário de execução. Nesse sentido, foi inicialmente definido um conjunto de três táticas de seleção, denominadas *otimista*, *conservadora* e *pessimista*, respectivamente, aplicáveis tanto à escolha de novas cargas de trabalho quanto à escolha de novos níveis de capacidade. A combinação dessas três táticas na escolha de novos cenários de execução dá origem a nove heurísticas de seleção, ilustradas na Figura 2.

Nessa figura, as heurísticas são identificadas por diferentes pares de letras posicionados ao longo da matriz que representa o espaço de implantação. A primeira letra que identifica a heurística refere-se à tática usada na escolha da configuração (linha), enquanto a segunda letra refere-se à tática usada na escolha da carga de trabalho (coluna). Como pode-se observar, a tática otimista leva à escolha de configurações menores e cargas de trabalho maiores. Já a tática conservadora leva à escolha de configurações e cargas de trabalho de nível intermediário. Por fim, a tática pessimista leva à escolha de configurações maiores e cargas de trabalho menores. Cada heurística é aplicada recursivamente, de modo a explorar subconjuntos cada vez menores do espaço de implantação a cada nova iteração do processo. Nesse contexto, os termos menores, intermediários e maiores são

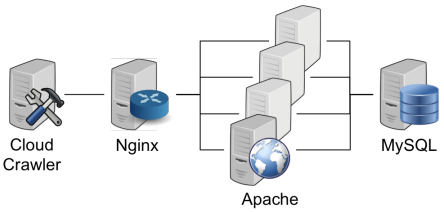


Figura 3. Arquitetura de implantação e avaliação do WordPress na Amazon EC2.

relativos, significando os elementos menores, intermediários e maiores, respectivamente, dentre aqueles ainda não explorados no espaço de implantação.

3. Avaliação Experimental

Esta seção descreve o experimento realizado como forma de verificação do processo de avaliação de capacidade apresentado anteriormente. Inicialmente, é apresentada a metodologia utilizada para a condução do experimento. Em seguida, são apresentados os resultados obtidos por cada uma das nove heurísticas de seleção propostas. Esses resultados são usados tanto para uma comparação qualitativa das heurísticas entre si, quanto para atestar a eficiência do processo proposto e de sua abordagem de inferência de desempenho.

É importante mencionar que o processo proposto foi implementado e está disponível na forma de uma ferramenta web,[1] a qual foi utilizada para executar o experimento descrito a seguir. Devido a restrições de espaço, os detalhes da implementação do processo bem como de sua ferramenta de apoio estão fora do escopo deste artigo.

3.1. Metodologia

O experimento consistiu na realização de sessões de avaliação de capacidade de uma aplicação web real (WordPress,[2] escolhida por ser uma das aplicações de criação e administração de *blogs* mais utilizadas atualmente) implantada em um provedor de nuvem também real (Amazon EC2,[3] escolhido por ser o líder de mercado entre provedores IaaS públicos). O WordPress foi implantado em duas camadas: uma para o banco de dados MySQL, e outra para o servidor de aplicação, executada pelo servidor Apache HTTPD. Como balanceador de carga, foi utilizada uma máquina dedicada executando o servidor web Nginx.

Devido a restrições de custo e tempo, o experimento limitou-se a variar apenas a camada de aplicação, usando de 1 a 4 servidores Apache executando o WordPress. A execução dos testes foi orquestrada pelo ambiente Cloud Crawler [Cunha et al. 2013b, Cunha et al. 2013a], que automatizou as tarefas de iniciar e parar todas as instâncias de máquinas virtuais, configurar o balanceador de carga de acordo com o número de instâncias testadas na camada de aplicação, iniciar e parar a execução dos testes, gerar as cargas de trabalho impostas à aplicação e, finalmente, coletar os dados de desempenho obtidos em cada teste. A Figura 3 ilustra a arquitetura utilizada para implantação e avaliação do WordPress na nuvem da Amazon.

[1] `http://cloud-capacitor.herokuapp.com/`.
[2] `https://wordpress.org/`.
[3] `http://aws.amazon.com/ec2`.

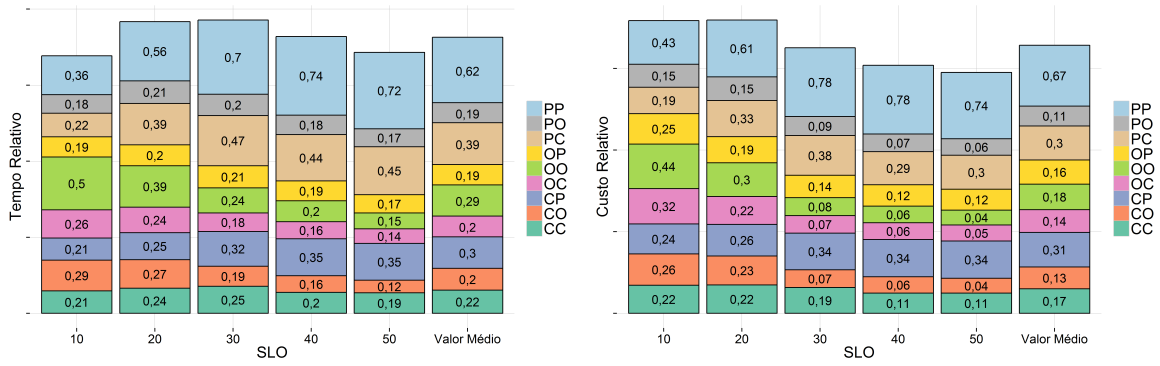
Para compor o espaço de implantação utilizado no experimento, foram escolhidos sete tipos de máquinas virtuais oferecidos pelo provedor Amazon EC2: *m3_medium*, *m3_large*, *m3_xlarge*, *m3_2xlarge*, *c3_large*, *c3_xlarge* e *c3_2xlarge*. Para cada um desses tipos, foram criadas configurações com 1, 2, 3 e 4 instâncias, levando a um total de 28 configurações diferentes no espaço de implantação, divididas em duas categorias distintas, "m3" e "c3". As relações de capacidade entre essas configurações foram definidas separadamente, para cada categoria, de modo a refletir o tipo e a quantidade de máquinas virtuais presentes em cada configuração. Assim, configurações com um certo número de máquinas virtuais de um determinado tipo eram consideradas de capacidade superior (inferior) a outras configurações contendo máquinas do mesmo tipo em menor (maior) quantidade. De maneira similar, configurações contendo um certo número de máquinas virtuais de um certo tipo eram consideradas de capacidade superior (inferior) a outras configurações com a mesma quantidade de máquinas mas de tipos diferentes se estes tipos fossem inferiores (superiores) ao tipo da primeira configuração, de acordo com a classificação dos tipos definidas pelo próprio provedor de nuvem. Por exemplo, a configuração composta por 3 máquinas to tipo *m3_2xlarge* era considerada superior a outra configuração composta por apenas 2 máquinas deste mesmo tipo. Da mesma forma, a configuração formada por 2 máquinas do tipo *c3_large* era considerada inferior a outra configuração com a mesma quantidade de máquinas do tipo *c3_xlarge*.

As cargas de trabalho utilizadas no experimento foram quantificadas em número de usuários concorrentes enviando requisições ao WordPress. Foi definido um total de 10 cargas de trabalho, representando 100, 200, 300, 400, 500, 600, 700, 800, 900 e 1000 usuários concorrentes, respectivamente.

De forma a estabelecer uma *baseline* para comparação da eficiência e da acurácia do processo proposto, especificamente de suas diferentes heurísticas de seleção, foram coletados dados de desempenho do WordPress na nuvem para cada um dos 280 cenários possíveis, ou seja, foram efetivamente realizados testes de desempenho da aplicação para cada uma das 28 configurações criadas sob cada uma das 10 cargas de trabalho especificadas. Esse conjunto de dados de execuções reais da aplicação foi denominado *oráculo*, e a estratégia necessária para gerar todos esses dados foi denominada heurística *Força Bruta* (em Inglês, *Brute Force – BF*). As nove heurísticas propostas foram então comparadas entre si e com a heurística BF.

Cada teste de desempenho consistiu em executar o WordPress utilizando uma das 28 configurações definidas para o espaço de implantação e então submetê-lo a uma das 10 cargas de trabalho especificadas durante um período de 1 hora. Durante os testes, um gerador de carga criava a quantidade de usuários corresponde à carga de trabalho sendo avaliada. Cada usuário realizava a seguinte sequência de requisições à aplicação: efetuar *logon*; inserir uma nova postagem; consultar a nova postagem; alterar a nova postagem; consultar postagens existentes por palavra-chave; alterar uma postagem existente; e, finalmente, efetuar *logoff*.

A métrica de desempenho utilizada no experimento foi o *tempo de resposta total*, ou seja, o tempo total decorrido entre o envio da primeira requisição da sequência acima e o momento em que o usuário recebeu a resposta para última requisição da sequência. Assim, para ser considerada como candidata para uma determinada carga de trabalho, uma configuração devia ser capaz de atender, sem erros, pelo menos 90% das sequências



(a) (b)

Figura 4. Eficiência das nove heurísticas de seleção em relação à heurística BF: (a) tempo de execução; (b) custo.

de requisições recebidas dos usuários da aplicação em um tempo total igual ou inferior ao valor do SLO, tal como definido no respectivo parâmetro de entrada do processo.

3.2. Resultados

3.2.1. Eficiência

Esta subseção apresenta os resultados de eficiência atingidos pelas heurísticas de seleção considerando-se duas métricas: *tempo de execução relativo* e *custo relativo*. Uma vez que a duração dos testes é igual em cada cenário, o tempo de execução relativo de uma determinada heurística é dado pela razão entre o número de vezes que a heurística executa a aplicação, e o número total de execuções da aplicação com a heurística BF. O custo relativo da heurística de seleção, por sua vez, é calculado pela razão entre a soma do custo de cada configuração efetivamente testada com essa heurística, e a soma dos custos de todas as configurações testadas com a heurística BF. Devemos notar que o custo de uma dada configuração depende do valor e da quantidade de máquinas virtuais que a compõem. Dessa forma, uma vez que os provedores podem fixar valores distintos para diferentes tipos de máquinas virtuais, o custo relativo de uma dada heurística de seleção será bastante influenciado pelas configurações específicas que a heurística selecionar para avaliar na nuvem.

A Figura 4 mostra os resultados para as duas métricas selecionadas, considerando os cinco SLOs investigados. Os resultados para a métrica tempo de execução relativo (Figura 4(a)) mostram que, sob SLOs mais brandos (ex: 50 segundos), as melhores heurísticas de seleção são OC e CO, oferecendo ganhos de 86% e 88%, respectivamente, com relação à BF. Porém, sob SLOs mais rígidos (ex: 10 segundos), as melhores heurísticas são PO e OP, com ganhos de 82% e 81%, respectivamente, com relação à BF. De fato, PO e OP, juntamente com CC, são em geral as melhores heurísticas para essa métrica, uma vez que seus resultados permanecem estáveis nos cinco SLOs, como indicado pelos valores médios (representados na coluna mais à direita dos dois gráficos). Os menores ganhos para essa métrica são obtidos com PP e PC, cujos ganhos, em média, podem chegar respectivamente a 38% e 61% com relação à BF.

No que diz respeito à métrica do custo relativo, uma análise da Figura 4(b) mostra que sob SLOs mais brandos os melhores resultados são obtidos com as heurísticas OO e

Tabela 2. Acurácia das heurísticas de seleção.

	SLO														
	10			20			30			40			50		
Heurística	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
CC	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,98	0,99	1,00	1,00	1,00	1,00	1,00	1,00
CO	1,00	1,00	1,00	1,00	1,00	1,00	0,99	1,00	0,99	1,00	1,00	1,00	1,00	1,00	1,00
CP	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,98	0,99	1,00	1,00	1,00	1,00	1,00	1,00
OC	1,00	1,00	1,00	1,00	1,00	1,00	0,99	0,99	0,99	1,00	1,00	1,00	1,00	1,00	1,00
OO	1,00	1,00	1,00	1,00	1,00	1,00	0,99	1,00	0,99	1,00	1,00	1,00	1,00	1,00	1,00
OP	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,98	0,99	1,00	1,00	1,00	1,00	1,00	1,00
PC	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,98	0,99	1,00	1,00	1,00	1,00	1,00	1,00
PO	1,00	1,00	1,00	1,00	1,00	1,00	0,99	1,00	0,99	1,00	1,00	1,00	1,00	1,00	1,00
PP	1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,98	0,99	1,00	1,00	1,00	1,00	1,00	1,00

CO, com ambas oferecendo ganhos de até 96% em comparação à BF. No entanto, essas duas heurísticas não têm um bom desempenho sob SLOs mais rígidos; nesse caso, os melhores resultados são obtidos com as heurísticas PO, PC e CC, com ganhos entre 78% e 85% com relação à BF. Em geral, as melhores heurísticas para esta métrica são PO, OC e CO, oferecendo ganhos médios entre 86% e 89% com relação à BF. Os menores ganhos médios são oferecidos pelas heurísticas PP, PC e CP, sendo que PP mais uma vez se destaca com o pior resultado entre todas as nove heurísticas avaliadas.

Uma análise abrangendo os resultados de ambas as métricas revela que, no geral, as melhores heurísticas são PO, OP e CC, todas oferecendo ganhos de ao menos 75% com relação à heurística BF em todos os cinco SLOs.

3.2.2. Acurácia

Para medir a acurácia do processo de avaliação de capacidade, foram calculados os valores médios de *Precision*, *Recall* e *F-Measure* [Baeza-Yates and Ribeiro-Neto 1999] para os resultados produzidos por cada uma das heurísticas de seleção sob os diferentes valores de SLO avaliados, tomando como base os dados do oráculo. Para isso, os dados do oráculo foram utilizados para determinar se as configurações identificadas como candidatas (resultados positivos) e rejeitadas (resultados negativos) por cada heurística para uma determinada carga de trabalho eram de fato verdadeiras (nesse caso, as predições teriam sido corretas) ou falsas (nesse caso, as predições teriam sido erradas).

Os valores dessas três métricas para uma carga de trabalho i, denotados por P_i, R_i e F_i, respectivamente, são dados pelas seguintes fórmulas:

$$P_i = \frac{\textit{no. resultados positivos verdadeiros}}{\textit{no. resultados positivos verdadeiros} + \textit{no. resultados positivos falsos}}$$

$$R_i = \frac{\textit{no. resultados positivos verdadeiros}}{\textit{no. resultados positivos verdadeiros} + \textit{no. resultados negativos falsos}}$$

$$F_i = \frac{2 \times P_i \times R_i}{P_i + R_i}$$

A Tabela 2 mostra os valores médios de P, R e F, considerando as 10 cargas de trabalho, calculados para cada heurística de seleção sob os cinco níveis de SLO. Nota-se que em apenas um dos cinco SLOs o processo deixou de obter 100% de acurácia nas predições, apresentando uma taxa de erro inferior a 3% para os valores de *Precision* e *Recall*, e de aproximadamente 1% para os valores de *F-Measure*, que estabelece uma média ponderada entre as duas primeiras métricas [Baeza-Yates and Ribeiro-Neto 1999].

Uma investigação mais minuciosa dos dados de desempenho da aplicação na nuvem revelou que essa pequena perda na qualidade das predições foi devida a flutuações ocasionais no desempenho de alguns dos tipos de máquinas virtuais disponibilizadas pelo provedor. Essas flutuações levaram algumas das configurações avaliadas a terem um desempenho superior ao de outras configurações consideradas de maior capacidade de acordo com o espaço de implantação. Tais flutuações afetaram particularmente o desempenho da aplicação para o SLO de 30 segundos, refletindo em erros de predição. De fato, oscilações no desempenho da infraestrutura virtualizada oferecida por provedores de nuvem IaaS são relativamente comuns, como já observado em outros trabalhos [Iosup et al. 2011, Jayasinghe et al. 2011, Cunha et al. 2011]. Vale destacar que o impacto dessa instabilidade poderia ter sido mitigado, caso um número maior de execuções para cada par de configuração e demanda tivesse sido realizado. Mesmo assim, no contexto deste trabalho o nível de instabilidade observado foi muito baixo, afetando um único nível de SLO com taxa de erro médio de 1%. Esses resultados reforçam a nossa confiança de que a abordagem de inferência de desempenho tem potencial para atingir alta acurácia mesmo quando utilizada em aplicações e plataformas de nuvem reais.

4. Trabalhos Relacionados

Esta seção analisa várias soluções existentes para apoiar os usuários de nuvens IaaS no planejamento da capacidade necessária às suas aplicações. Conforme mencionado previamente, essas soluções seguem duas abordagens principais, aqui denominadas de preditiva e empírica.

As soluções da abordagem preditiva utilizam diferentes técnicas de predição do desempenho da aplicação, com destaque para a analogia com os resultados obtidos através da execução de diversos *benchmarks* na nuvem, normalmente coletados *a priori* pelo provedor da solução [Malkowski et al. 2010, Li et al. 2010, Jung et al. 2013]; simulação do comportamento esperado da aplicação através de um simulador de nuvem [Fittkau et al. 2012]; e reprodução na nuvem de eventos relevantes do ponto de vista de desempenho, como utilização de CPU, memória e disco, capturados a partir da execução local da aplicação [Li et al. 2011]. As abordagens que fazem analogia e simulação possuem a vantagem de serem de baixo custo, ao contrário da solução descrita em [Li et al. 2011], que necessita adquirir recursos da nuvem para reproduzir os eventos da aplicação. No entanto, todos esses trabalhos ainda deixam a desejar em termos de acurácia, devido a limitações importantes das técnicas de predição adotadas. Mais especificamente, a predição por analogia tem pouca eficácia se os *bechmarks* disponíveis não possuem perfis de comportamento similares ao da aplicação sob teste. Já os simuladores de nuvem ainda não conseguem atingir um nível de fidelidade próximo ao comportamento real de uma aplicação implantada em um provedor de nuvem público, chegando a apresentar diferenças de desempenho superiores a 30% [Fittkau et al. 2012]. Um problema similar ocorre com a solução que reproduz eventos da aplicação na nuvem, cujo mecanismo de captura de eventos ainda possui sérias limitações de ordem prática [Li et al. 2011].

As soluções empíricas, por outro lado, oferecem alta acurácia na avaliação do desempenho da aplicação na nuvem, uma vez que são baseadas em dados de desempenho obtidos diretamente no provedor [Jayasinghe et al. 2012, Silva et al. 2013, Cunha et al. 2013a, Scheuner et al. 2014]. Além disso, essas soluções são muito mais flexíveis, no sentido em que permitem aos usuários avaliar diferentes combinações de

componentes da aplicação sob as mais variadas configurações de recursos e cargas de trabalho. O ponto negativo das soluções que adotam a abordagem empírica é a necessidade de executar cada um dos cenários definidos pelo usuário, uma vez que elas não oferecem nenhum mecanismo voltado especificamente para reduzir a quantidade de execuções da aplicação. Dessa forma, cabe exclusivamente aos usuários dessas soluções definirem as melhores estratégias de explorar o espaço de implantação da aplicação na nuvem.

Existem outros trabalhos que adotam estratégias de planejamento de capacidade de curto prazo na nuvem, conhecidas como *auto-scaling* (e.g., [Morais et al. 2013]). Tais trabalhos visam ajustar dinamicamente os recursos da nuvem alocados à a aplicação, com base em regras de escalabilidade definidas pelo usuário e métricas coletadas a partir do monitoramento do comportamento da aplicação (ex: uso de CPU e memória). Alguns problemas relacionados com estas soluções é que nem sempre as regras especificadas pelos usuários levam em consideração a alocação das melhores configurações de máquinas virtuais (ex: em termos de custo e desempenho) para atender a demanda atual da aplicação.

Nesse contexto, o novo processo de avaliação de capacidade apresentado neste trabalho segue uma abordagem híbrida, combinando aspectos positivos das abordagens preditiva e empírica. Em contraste às soluções da abordagem preditiva, o novo processo realiza predições com base em relações de capacidade definidas entre diferentes configurações de recursos de um mesmo provedor de nuvem, e em resultados empíricos obtidos a partir da execução da própria aplicação neste provedor. Com isso, o novo processo consegue alta acurácia nas predições ao mesmo tempo em que reduz significativamente a quantidade de cenários de implantação que precisam ser efetivamente testados na nuvem. Além disso, acreditamos que o processo de inferência de desempenho também possa ser útil para apoiar soluções baseadas em auto-scaling, por exemplo, relevando as melhores configurações de máquinas virtuais para diferentes faixas de demanda da aplicação.

5. Conclusão e Trabalhos futuros

A tarefa de escolher adequadamente os recursos computacionais (ex.: máquinas virtuais) de um provedor de nuvem, de forma a minimizar os custos necessários para atender diferentes níveis de demanda de uma aplicação, é um desafio importante para o qual ainda não existem soluções plenamente satisfatórias disponíveis. Este trabalho apresentou um novo processo de avaliação de capacidade por inferência de desempenho, que se mostrou uma solução ao mesmo tempo eficiente (em termos de custo e tempo) e eficaz (em termos da acurácia dos resultados) para apoiar o planejamento da capacidade de aplicações na nuvem.

Com relação aos trabalhos futuros, algumas possibilidades interessantes para melhoria ou extensão deste trabalho incluem: realizar novos experimentos visando investigar se os resultados reportados neste artigo são generalizáveis para outras aplicações e provedores de nuvem; investigar novas heurísticas de seleção de configurações e cargas de trabalho, que levem em conta dados sobre a utilização dos recursos da nuvem pela aplicação, como consumo de CPU e memória; e propor novos critérios para definir as relações de capacidade entre as diferentes configurações disponibilizadas pelo provedor de nuvem, por exemplo, considerando o custo de cada configuração, e investigar seu impacto no desempenho das heurísticas de seleção.

Referências

Baeza-Yates, R. A. and Ribeiro-Neto, B. (1999). *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Beserra, P. V. et al. (2012). Cloudstep: A Step-by-Step Decision Process to Support Legacy Application Migration to the Cloud. In *IEEE MESOCA 2012*, pages 7–16.

Cunha, M. et al. (2011). Investigating the impact of deployment configuration and user demand on a social network application in the Amazon EC2 cloud. In *IEEE CloudCom 2011*, pages 746–751.

Cunha, M. et al. (2013a). A Declarative Environment for Automatic Performance Evaluation in IaaS Clouds. In *IEEE CLOUD 2013*, pages 285–292.

Cunha, M. et al. (2013b). Cloud Crawler: Um Ambiente Programável para Avaliar o Desempenho de Aplicações em Nuvens de Infraestrutura. In *SBRC 2013*, pages 747–760.

Fittkau, F. et al. (2012). CDOSim: Simulating cloud deployment options for software migration support. In *IEEE MESOCA 2012*, pages 37–46.

Gonçalves Junior, R. et al. (2015). A Multi-Criteria Approach for Assessing Cloud Deployment Options Based on Non-Functional Requirements. In *ACM SAC 2015*.

Iosup, A. et al. (2011). On the performance variability of production cloud services. In *IEEE/ACM CCGrid 2011*, pages 104–113.

Jayasinghe, D. et al. (2011). Variations in performance and scalability when migrating n-tier applications to different clouds. In *IEEE CLOUD 2011*, pages 73–80.

Jayasinghe, D. et al. (2012). Expertus: A Generator Approach to Automate Performance Testing in IaaS Clouds. In *IEEE CLOUD 2012*, pages 73–80.

Jung, G. et al. (2013). CloudAdvisor: A Recommendation-as-a-Service Platform for Cloud Configuration and Pricing. In *IEEE SERVICES 2013*, pages 456–463.

Li, A. et al. (2010). CloudCmp: Comparing Public Cloud Providers. In *ACM SIGCOMM IMC 2010*, pages 1–14.

Li, A. et al. (2011). CloudProphet: Towards Application Performance Prediction in Cloud. In *ACM SIGCOMM 2011*, pages 426–427.

Malkowski, S. et al. (2010). CloudXplor: A tool for configuration planning in clouds based on empirical data. In *ACM SAC 2010*, pages 391–398.

Menascé, D. A. and Ngo, P. (2009). Understanding Cloud Computing: Experimentation and Capacity Planning. In *CMG 2009*.

Morais, F. J. A. et al. (2013). Autoflex: Service Agnostic Auto-scaling Framework for IaaS Deployment Models. In *IEEE/ACM CCGrid 2013*, pages 42–49.

Scheuner, J. et al. (2014). Cloud WorkBench – Infrastructure-as-Code Based Cloud Benchmarking. *arXiv preprint arXiv:1408.4565*.

Silva, M. et al. (2013). CloudBench: Experiment Automation for Cloud Environments. In *IEEE IC2E 2013*, pages 302–311.