Performance Inference: A Novel Approach for Planning the Capacity of IaaS Cloud Applications

Marcelo Gonçalves, Matheus Cunha, Nabor C. Mendonça, Américo Sampaio
Programa de Pós-Graduação em Informática Aplicada (PPGIA)
Universidade de Fortaleza (UNIFOR)
Fortaleza, CE, Brazil
Email: {marcelocg,mathcunha}@gmail.com, {nabor,americo.sampaio}@unifor.br

***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.**

I. Introduction

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 [1]. 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 [2].

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 [3]. 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 [4], 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 [5].

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 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 [6]–[11]. 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 [10]. Já os trabalhos que fazem parte da segunda abordagem, aqui referenciada como *abordagem empírica*, tem como objetivo medir o desempenho real da aplicação através de sua efetiva implantação na nuvem e da realização de testes de carga [3], [12]–[14]. 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.

Visando combinar as vantagens das abordagens preditiva e empírica, 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 "in-

ferir"), 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 pequena parte 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 III 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 IV compara o novo processo com outros trabalhos relacionados. Por fim, a Seção V oferece as conclusões e sugestões para trabalhos futuros.

II. The Proposed Capacity Evaluation Process

A. Concepts and Terminology

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

TABLE I. Concepts and terminology used in the paper.

Concept	Definition
Aplication under Test	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*.
Performance Metric	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.
Reference Value (SLA)	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 SLA (*Service Level Agreement*), 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*.
Cloud Provider	Uma empresa que fornece recursos computacionais como serviço cobrado financeiramente por fração de tempo de utilização. Neste trabalho, o foco será em provedores que disponibilizam recursos de infraestrutura, notadamente *máquinas virtuais*.
VM Types	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 mantenha uma linha de produtos discreta e finita.
VM Category	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.
Deployment Configuration	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.
Deployment Space	Denota um conjunto limitado de configurações de máquina virtuais nas quais a aplicação sob teste será implantada e executada durante uma sessão de avaliação.
Capacity Relations	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.
Capacity Level	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.
Workload	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.
Execution	Corresponde à execução da aplicação sob teste utilizando uma determinada configuração de máquinas virtuais e submetida a uma determinada carga de trabalho. O resultado de uma *execução* fornece indicadores que permitirão avaliar se a aplicação atingiu o valor de referência esperado para uma determinada métrica de desempenho naquele cenário.

B. Input Data

O principal dado de entrada esperado pelo processo é o valor de referência (SLA), o qual será usado para determinar se a aplicação atingiu os requisitos mínimos de desempenho exigidos em cada cenário de execução. Além do SLA, 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 III ilustra alguns critérios que podem ser usados para este fim.

A Figura 1 mostra um pequeno exemplo de um espaço de implantação, no qual 6 configurações, pertencentes a duas categorias distintas, foram classificadas em dois níveis de capacidade dentro de cada categoria. Nesse exemplo, os retângulos representam as configurações, com o rótulo de cada retângulo indicando o tipo e a quantidade de máquinas virtuais que compõem a configuração, e as setas que ligam as configurações representam a existência de uma relação de capacidade elas. A ausência de seta entre duas configurações implica a impossibilidade de se estabelecer uma relação de capacidade entre elas.

(... explicar melhor as configurações do exemplo e como foram definidas as suas relações de capacidade!)

C. Process Activities

As principais atividades executadas pelo processo de avaliação de capacidade são ilustradas no diagrama da Figura 2. As atividades destacas com o rótulo $\ll$A$\gg$ são atividades abstratas, devendo ser customizadas pelos usuários do processo de acordo com diferentes estratégias de avaliação

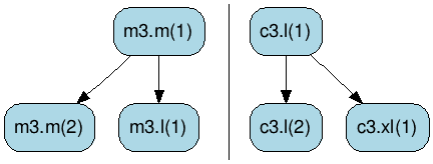


Fig. 1. Exemplo de relações e níveis de capacidade entre configurações.

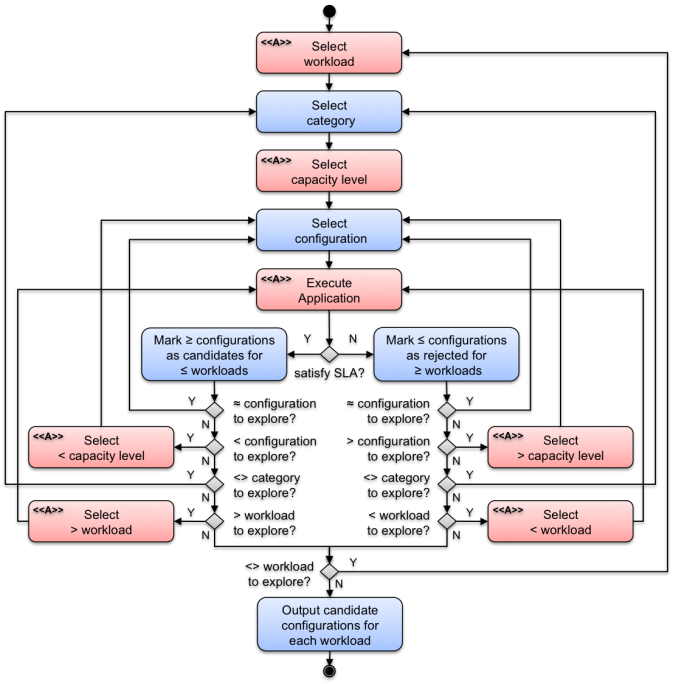


Fig. 2. Diagrama de atividades do processo de avaliação de capacidade.

(descritas na Seção II-D). As outras atividades são executadas de maneira idêntica independentemente de qual seja a aplicação sob teste ou de qual seja a estratégia de avaliação utilizada.

A execução do processo acontece em quatro fases bem distintas e cíclicas: seleção do cenário de execução da aplicação; execução da aplicação; inferência de desempenho; e seleção do próximo cenário. Cada uma dessas fases será detalhada a seguir.

1) Selection of an Initial Execution Scenario: 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 categorias 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) Application Execution: 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.

3) Performance Inference: 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 (SLA) 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 SLA, 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. Ora, 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 SLA 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 SLA 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.

O efeito da inferência de desempenho pode ser melhor visualizado através do exemplo de espaço de implantação mostrado na Figura 3. Nesse exemplo, o espaço de implantação está representado na forma de uma matriz, cujas linhas corre-

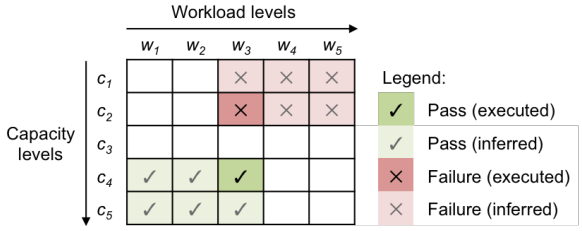


Fig. 3. Ilustração da marcação de configurações como *candidatas* (embaixo à esquerda) ou *rejeitadas* (no alto à direita) via inferência de desempenho.

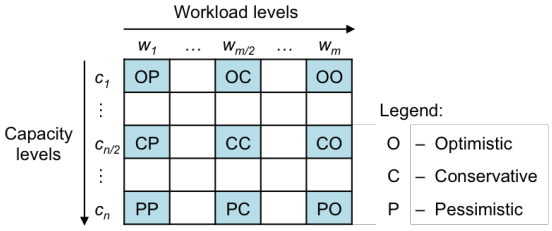


Fig. 4. Heurísticas para seleção de configurações e cargas de trabalho.

spondem às configurações e as colunas correspondem às cargas de trabalho. Note que o nível de capacidade das configurações cresce de cima para baixo na matriz, enquanto o tamanho da carga de trabalho cresce da esquerda para a direita. A parte inferior esquerda da matriz ilustra o caso em que o SLA é satisfeito em um determinado cenário de execução. Já a parte superior direita ilustra o caso oposto. As células marcadas com um "✓" ("×") indicam as configurações marcadas como candidatas (rejeitadas) para as respectivas cargas de trabalho. As duas células em destaque correspondem a execuções reais da aplicação, cujos resultados serviram de base para a inferência do desempenho da aplicação nos outros cenários.

Esse exemplo é representativo do grande potencial da abordagem de inferência de desempenho para reduzir os custos associados ao processo de planejamento de capacidade, tonando-o mais rápido e eficiente. Note que, considerando os dois casos ilustrados, o processo teria obtido resultados relativos à avaliação de 12 cenários de execução distintos, dos quais apenas dois teriam de fato sido executados na nuvem, o que representa uma economia de quase 90% em relação à abordagem empírica tradicional, onde todos os cenários de interesse devem ser sistematicamente avaliados.

Na seção III, apresentaremos resultados obtidos empiricamente mostrando que a abordagem de inferência de desempenho consegue prever o desempenho esperado de uma aplicação na nuvem com alta precisão.

4) Selection of the Next Execution Scenario: 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 (que deverá ser maior ou menor que o nível de capacidade atual, a depender do resultado da execução da aplicação no atual cenário), a escolha de uma nova categoria, ou a escolha de uma nova carga de trabalho (que também deverá ser maior ou menor que o nível de carga atual, novamente a depender do resultado da execução da aplicação no atual cenário). As escolhas do novo nível de capacidade e da nova carga de trabalho também são atividades abstratas, a serem definidas de acordo com a estratégia de avaliação utilizada para customizar o processo.

D. Evaluation Strategies

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 4.

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.

No contexto do processo de avaliação de capacidade proposto neste trabalho, cada heurística de seleção fornece uma "lógica" diferente para exploração do espaço de implantação, servindo como base para a customização do processo com diferentes estratégias de avaliação. A acurácia e a eficiência do processo proposto, em particular, da abordagem de inferência de desempenho, utilizando cada uma das noves heurísticas de seleção mencionas acima, serão analisadas na próxima seção.

III. Experimental Evaluation

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.

A. Method

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 [13], 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 5 mostra um panorama dessa arquitetura de implantação.

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*.

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

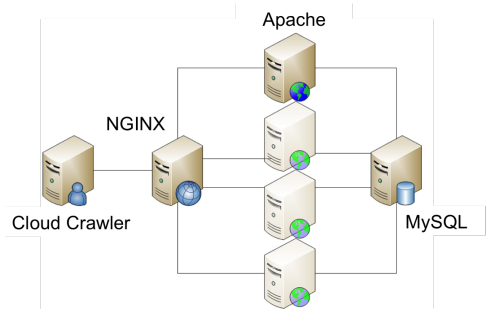


Fig. 5. Arquitetura de implantação do WordPress na Amazon EC2.

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

B. Results

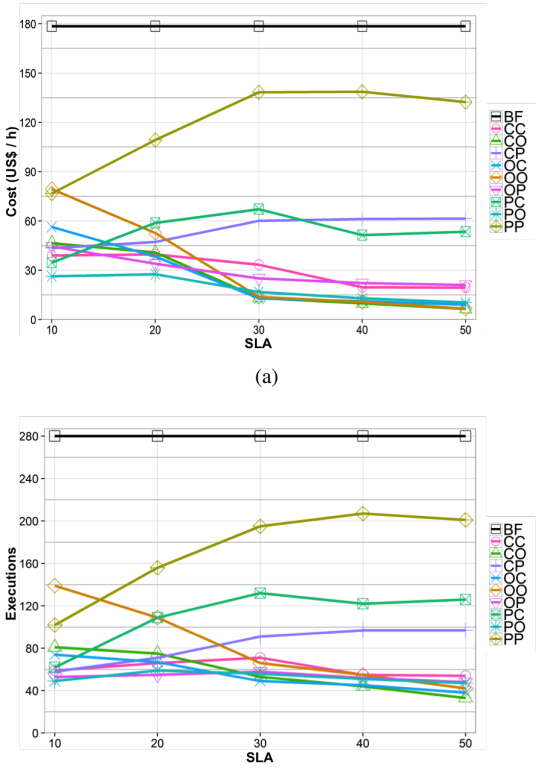
1) Efficiency: Esta subseção apresenta os resultados de eficiência atingidos pelas heurísticas de seleção usadas no processo sob dois aspectos distintos: o custo total da avaliação e a quantidade de execuções realizadas por cada heurística. Esse custo foi calculado somando-se o preço da hora de utilização, conforme a tabela de preços do provedor na data realização dos testes, para cada uma das configurações para as quais foram realizadas execuções reais na nuvem.

A Figura 6 mostra os gráficos dos resultados obtidos pelas nove heurísticas em relação a essas duas métricas, considerando SLAs de 10, 20, 30, 40 e 50 segundos. No topo de cada gráfico vê-se uma linha horizontal escura que representa os resultados da heurística BF. Note que, como essa heurística não efetua nenhuma inferência quanto ao desempenho da aplicação, seus resultados, tanto em termos de custo quanto em termos de número de execuções, são sempre constantes, independente do SLA requerido.

(... melhorar e expandir análise dos resultados!)

A análise do gráfico de custo (Figura 6a) mostra que mesmo a heurística com o pior desempenho no que se refere ao custo já apresenta uma redução considerável em relação à heurística BF. Por outro lado, as melhores heurísticas chegam a representar uma economia da ordem de 96% em comparação com o que seria gasto com a execução de todas as combinações de configurações e cargas de trabalho. Embora o comportamento das heurísticas varie em função do SLA, é possível notar que quando a exigência do SLA é mais moderada, o comportamento de todas as heurísticas se estabiliza, tornando possível identificar que algumas delas tendem a ser mais econômicas que as outras. Ainda que não seja possível afirmar que uma só heurística seja a melhor em todas as situações, pode-se considerar que a heurística Pessimista/Otimista (PO) se mostra como a mais econômica em geral. A heurística Conservadora/Otimista (CO) merece atenção para os SLAs mais brandos, com os menores custos absolutos nessas circunstâncias.

A análise do gráfico de execuções reais na nuvem (Figura 6b), por sua vez, mostra uma redução de até 88% em relação aos resultados da heurística BF. Os



(a)

(b)

Fig. 6. Eficiência das heurísticas de seleção: (a) custo e (b) execuções.

menores números de execuções são atingidos pelas heurísticas Otimista/Conservadora (OC) e Conservadora/Otimista (CO), sob os SLAs mais brandos. Porém, como não se saem tão bem sob SLAs mais rígidos, como 10 segundos, a heurística PO ganha destaque por ter comportamento mais estável, figurando entre as mais econômicas no aspecto de quantidade de execuções sob a maioria dos SLAs avaliados.

Vale ressaltar que a significativa redução do número de execuções necessárias durante o processo de planejamento de capacidade, decorrente da utilização da abordagem de inferência de desempenho, conforme mostrado nesta seção, implica não apenas em economia de tempo e horas de máquinas para os usuários da nuvem, mas também pode contribuir de forma decisiva para a reduzir outros tipos de custo típicos de qualquer projeto, como esforço e alocação de recursos humanos .

2) Accuracy: Para medir a acurácia do processo de avaliação de capacidade, foram calculados os valores médios de *Precision*, *Recall* e *F-Measure* [15] para os resultados produzidos por cada uma das heurísticas de seleção sob os diferentes valores de SLA avaliados, tomando como base os dados do oráculo.

(... explicar as três métricas!)

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).

TABLE II. ACCURACY OF THE PROPOSED SELECTION HEURISTICS.

Heuristic	SLA														
	10			20			30			40			50		
	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

Sejam $TP(w,s)$, $FP(w,s)$ e $FN(w,s)$ os resultados positivos verdadeiros, resultados positivos falsos e resultados negativos falsos, respectivamente, obtidos por uma heurística de seleção h sob uma carga de trabalho w e um SLA s. Os valores de *Precision*, *Recall* e *F-Measure* para a heurística h sob a carga de trabalho w e SLA s, denotados por $P(h,w,s)$, $R(h,w,s)$ e $F(h,w,s)$, respectivamente, são calculados pelas seguintes fórmulas [15]:

$$P(h,w,s) = \frac{TP(h,w,s)}{TP(h,w,s) + FP(h,w,s)}$$

$$R(h,w,s) = \frac{TP(h,w,s)}{TP(h,w,s) + FN(h,w,s)}$$

$$F(h,w,s) = 2 \times \frac{P(h,w,s) \times R(h,w,s)}{P(h,w,s) + R(h,w,s)}$$

Considerando m cargas de trabalho (denotadas por w_1, $w_2, \ldots, w_m$), as fórmulas para o cálculo das médias dessas três métricas para uma heurística h e SLA s, denotados por $P(h,s)$, $R(h,s)$ e $F(h,s)$, respectivamente, são as seguintes:

$$P(h,s) = \sum_{i=1}^{m} P(h,w_i,s) \,/\, m$$

$$R(h,s) = \sum_{i=1}^{m} R(h,w_i,s) \,/\, m$$

$$F(h,s) = \sum_{i=1}^{m} F(h,w_i,s) \,/\, m$$

A Tabela II mostra os valores de P, R e F calculados para as nove heurísticas de seleção sob os cinco níveis de SLA investigados no experimento. Nota-se que em apenas um dos cinco SLAs o processo deixou de obter 100% de acurácia nas predições, apresentando uma taxa de erro inferior a 3% para P e R, e de aproximadamente 1% para F.

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. (... detalhar quais configurações causaram o problema!)

Tais flutuações afetaram particularmente o desempenho da aplicação para o SLA 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 observados por [16] e [17]. O impacto dessas flutuações nos resultados observados no experimento, porém, foi mínimo (afetando um único nível de SLA com taxa de erro médio de 1%), o que reforça a confiança de que a abordagem de inferência de desempenho proposta neste trabalho pode atingir alta acurácia mesmo para aplicações e plataformas de nuvem reais.

IV. RELATED WORK

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 [6]–[8], [11]; simulação do comportamento esperado da aplicação através de um simulador de nuvem [10]; 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 [9]. Embora de baixo custo (com exceção da solução descrita em [9], que necessita adquirir recursos da nuvem para reproduzir os eventos da aplicação) e, no caso de soluções que suportam predições por analogia, de baixa complexidade, 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 pouco 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% [10]. 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 [9].

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 [3], [12]–[14]. 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.

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 entras 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 que reduz significativamente a quantidade de cenários de implantação que precisam ser efetivamente testados na nuvem.

V. Conclusion and Future Work

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 a planejar a 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 de 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.

Acknowledgment

This work is partially supported by Brazil's National Council for Scientific and Technological Development (CNPq), under grants 311617/2011-5 and 487174/2012-7.

References

[1] D. A. Menascé and P. Ngo, "Understanding Cloud Computing: Experimentation and Capacity Planning," in *CMG 2009*, 2009.

[2] R. Gonçalves Junior *et al.*, "A Multi-Criteria Approach for Assessing Cloud Deployment Options Based on Non-Functional Requirements," in *ACM SAC 2015*, 2015.

[3] M. Silva *et al.*, "CloudBench: Experiment Automation for Cloud Environments," in *IEEE IC2E 2013*, 2013, pp. 302–311.

[4] D. Jayasinghe *et al.*, "Variations in performance and scalability when migrating n-tier applications to different clouds," in *IEEE CLOUD 2011*, 2011, pp. 73–80.

[5] P. V. Beserra *et al.*, "Cloudstep: A Step-by-Step Decision Process to Support Legacy Application Migration to the Cloud," in *IEEE MESOCA 2012*, 2012, pp. 7–16.

[6] CloudHarmony, "CloudHarmony: Benchmarking the Cloud," 2014, http://goo.gl/IHDYxN.

[7] S. Malkowski *et al.*, "CloudXplor: A tool for configuration planning in clouds based on empirical data," in *ACM SAC 2010*, 2010, pp. 391–398.

[8] A. Li *et al.*, "CloudCmp: Comparing Public Cloud Providers," in *ACM SIGCOMM IMC 2010*, 2010, pp. 1–14.

[9] ——, "CloudProphet: Towards Application Performance Prediction in Cloud," in *ACM SIGCOMM 2011*, 2011, pp. 426–427.

[10] F. Fittkau *et al.*, "CDOSim: Simulating cloud deployment options for software migration support," in *IEEE MESOCA 2012*, 2012, pp. 37–46.

[11] G. Jung *et al.*, "CloudAdvisor: A Recommendation-as-a-Service Platform for Cloud Configuration and Pricing," in *IEEE SERVICES 2013*, 2013, pp. 456–463.

[12] D. Jayasinghe *et al.*, "Expertus: A Generator Approach to Automate Performance Testing in IaaS Clouds," in *IEEE CLOUD 2012*, 2012, pp. 73–80.

[13] M. Cunha *et al.*, "A Declarative Environment for Automatic Performance Evaluation in IaaS Clouds," in *IEEE CLOUD 2013*, 2013, pp. 285–292.

[14] J. Scheuner *et al.*, "Cloud WorkBench – Infrastructure-as-Code Based Cloud Benchmarking," *arXiv preprint arXiv:1408.4565*, 2014.

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

[16] A. Iosup *et al.*, "On the performance variability of production cloud services," in *IEEE/ACM CCGrid 2011*, 2011, pp. 104–113.

[17] M. Cunha *et al.*, "Investigating the impact of deployment configuration and user demand on a social network application in the Amazon EC2 cloud," in *IEEE CloudCom 2011*, 2011, pp. 746–751.