Heurísticas para Predição de Configurações de Custo Mínimo para Execução de Aplicações em Ambientes de Nuvem de Infraestrutura

1. Introdução e Motivação

O processo de decisão pela migração de aplicações para o ambiente de nuvens computacionais envolve uma série de análises que buscam, entre outras coisas, identificar que vantagens a mudança trará de fato. Ao comparar os serviços de provedores de computação em nuvem com a administração de um centro de dados próprio, a melhoria dos indicadores de desempenho e de custo, como redução de tempo de resposta, redução/otimização de custo de operação e melhores ferramentas com mais facilidades de gerenciamento, está entre os principais benefícios buscados a partir da adoção do ambiente de infraestrutura como serviço (IaaS – Infrastructure as a Service) [LI *et. al.*, 2011; RODERO-MERINO *et. al.*, 2010].

Em geral, provedores de IaaS cobram um valor em função do tempo de ocupação de uma máquina virtual, normalmente medido em horas, e esse valor unitário varia conforme o tamanho da máquina virtual (capacidade de processamento, memória e espaço de armazenamento). Dessa forma, a apuração do custo de operação da aplicação em um determinado período de tempo leva em conta a quantidade de máquinas virtuais utilizadas bem como seu perfil, ou seja, o tamanho e quantidade de recursos usados em cada uma.

Para prever o custo de operação de uma aplicação na nuvem, é preciso estimar ou medir como a aplicação responderá à demanda submetida em termos de indicadores de desempenho. A aplicação deve manter ou superar na nuvem o nível de desempenho apresentado quando executada em centro de dados próprio e com indicadores de custo menores, a fim de que se justifique o investimento feito na migração de ambiente. Para se chegar a essa conclusão, faz-se necessário conhecer o comportamento da aplicação na nova implantação para que se identifiquem quais perfis de máquinas virtuais oferecidos pelo provedor são capazes de executar a aplicação com níveis satisfatórios de desempenho. Somem-se a isso as variações da demanda exercida sobre a aplicação e as diversas possibilidades de variação de arquitetura de implantação por meio de procedimentos de escalabilidade.

Ao se tomarem procedimentos de escalabilidade vertical (variando-se a quantidade de recursos de cada máquina) e/ou de escalabilidade horizontal (variando-se a quantidade de máquinas em uma ou mais camadas, como dados, apresentação e negócio) chega-se a níveis de desempenho e de custo muito diversos. A variação da demanda exige que a aplicação também varie em tamanho da implantação, vertical ou horizontalmente, conforme a carga aplicada. Quanto mais acentuadas e mais frequentes as variações na demanda, mais variações de custo e desempenho serão observadas.

Assim, o custo apresenta-se entre os mais difíceis de prever, uma vez que depende necessariamente do tamanho da demanda exercida sobre a aplicação além do desempenho oferecido e preços cobrados pelo provedor de nuvem de infraestrutura contratado [CUNHA et. al., 2012]. Estrategicamente torna-se interessante identificar, entre as possíveis composições de máquinas virtuais ofertadas em um ou vários provedores, quais são as configurações de menor custo capazes de executar a aplicação mantendo-se os níveis satisfatórios para os indicadores de desempenho.

Para saber se uma determinada configuração de recursos do provedor é capaz de atender a uma demanda específica, é preciso antes enumerar os indicadores de desempenho que mais interessam à aplicação e a partir daí estabelecer os valores aceitáveis para esses indicadores. Uma vez estabelecidos os valores aceitáveis, pode-se implantar a aplicação sob essa configuração de recursos e então aplicar diferentes níveis de carga de trabalho sobre a aplicação. Ao comparar a resposta da aplicação com os valores dados como aceitáveis para os indicadores, é possível determinar se aquela configuração de recursos escolhida é capaz de executar a aplicação a contento e ainda calcular o custo mensal dessa implantação.

Porém, partindo-se do pressuposto de que o desempenho da aplicação foi satisfatório, o que se tem até agora é o custo de uma única configuração capaz de executar a aplicação estudada sob um único nível de carga de trabalho. Mas cargas de trabalho costumam variar em função do tempo em implantações reais, fazendo-se necessário, portanto, que esse efeito seja contemplado nos testes por meio da medição do desempenho da aplicação submetida a diferentes níveis de carga de trabalho.

Analogamente, as diversas configurações de máquinas e recursos, ainda que no mesmo provedor, podem responder de maneira muito diferente sob o mesmo nível de carga de trabalho a depender do momento em que sejam ativados [CUNHA et. al., 2011; IOSUP et. al. 2011; JAYASINGHE et. al. 2011]. Independente do motivo que leve a esse comportamento de certa forma imprevisível, é preciso levar em conta nos ensaios de avaliação de desempenho essa variabilidade e isso pode ser alcançado através da repetição dos cenários de teste em horários e dias diferentes.

Um grande problema começa a se desenhar ao seguir essa abordagem: a fase de ensaios pode atingir patamares elevados de custo, a depender das necessidades de variação da demanda, da arquitetura de implantação e das configurações utilizadas em cada arquitetura implantada [SILVA et. al., 2013]. Ainda que certos provedores *IaaS* ofereçam descontos ou pacotes de horas grátis para novos clientes, em geral esses incentivos são suficientes para custear apenas um mês de utilização de uma única máquina virtual muito pequena, provavelmente incapaz suportar a carga de uma aplicação real em produção. Assim, executar uma aplicação real, tipicamente implantada em arquitetura de várias camadas, 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 inviabilize o próprio projeto de migração dessa aplicação para a nuvem. Para evitar que sejam feitos testes com todas as combinações de provedores, configurações, horários, cargas de trabalho e métricas avaliadas, é possível lançar mão de técnicas de predição.

Através da predição, é possível estimar com razoável aproximação o desempenho que a aplicação apresentará ao ser executada em vários perfis de configuração diferentes, permitindo a determinação de qual configuração de menor custo capaz de executar a aplicação e sem a necessidade da realização completa dos testes. Como consequência, o custo dessa fase de ensaios pode ser reduzido sensivelmente.

Este trabalho propõe heurísticas de predição de custo mínimo para execução de aplicações em ambientes de nuvem de infraestrutura, bem como um arcabouço de programação que apoia a implementação dessas heurísticas. Além disso, o trabalho estuda os resultados apresentados pela aplicação das heurísticas propostas quanto ao custo total de execução da fase de ensaios para escolha da melhor configuração capaz de executar uma aplicação e quanto à acuidade dos resultados da predição em si.

2. Trabalhos Relacionados

CloudProphet – predição de desempenho baseado em record-replay

CloudBench – mecanismo de especificação e automação de testes

3. Formalização do Problema

A fim de que uma análise criteriosa possa ser feita, apresentamos a seguir um conjunto de definições e terminologias que possam basear o entendimento da construção do trabalho e também da avaliação dos resultados e de sua eficiência e eficácia. Os conceitos aqui explicitados são tomados de forma a permitir um estudo agnóstico quanto a aplicações, plataformas e provedores utilizados durante a execução das ferramentas desenvolvidas neste trabalho.

Apresentamos também um formalismo que visa à generalização da metodologia utilizada neste trabalho e a permitir a melhor descrição do raciocínio lógico envolvido no desenvolvimento das heurísticas criadas.

3.1 Definições e Terminologias

Apresentamos a seguir as definições que permeiam o conhecimento necessário para a análise dos problemas estudados e soluções propostas. Mostramos também as terminologias ou nomenclaturas que criamos para designar esses conceitos a fim de facilitar a comunicação e o entendimento por parte do leitor.

3.1.1 Aplicação sob Teste

A Aplicação sob Teste é um sistema computacional, possivelmente implementado em arquitetura multicamadas, para o qual se deseja observar o comportamento em um ambiente de computação em nuvem e ao estão ligadas uma ou mais Métricas de Desempenho.

3.1.2 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. Ex. *tempo de resposta*, *quadros por segundo*, etc. Métricas podem ser *minimizáveis* ou *maximizáveis*, a depender do objetivo da métrica quanto ao resultado desejado. Por exemplo, “tempo de resposta” é uma métrica minimizável, uma vez que geralmente se deseja que uma Aplicação responda a uma requisição com o menor tempo de resposta possível nos resultados. Contrariamente, uma métrica “quadros renderizados por segundo”, no domínio da computação gráfica, é uma métrica maximizável, pois quanto mais quadros são renderizados por unidade de tempo, maior a qualidade percebida pelo usuário.

3.1.3 Valor de Referência

Um valor predefinido como minimamente aceitável como resultado apresentado por uma Métrica após a execução da Aplicação sob Teste. Este valor, também referenciado neste trabalho como SLA (*Service Level Agreement*), serve como base de comparação para que as Heurísticas desenvolvidas e apresentadas saibam se a Aplicação é capaz de ser executada sobre um determinado arranjo de máquinas virtuais e sob uma determinada carga de trabalho a ela imposta.

3.1.4 Provedor

Consideramos neste trabalho a figura do provedor como representando uma empresa que fornece infraestrutura computacional como serviço cobrado financeiramente por fração de tempo de utilização. Alguns provedores fornecem conjuntamente a modalidade de plataforma como serviço. Nós, porém, não estamos considerando essa modalidade neste trabalho, interessando-nos apenas os serviços de infraestrutura, notadamente a disponibilização de máquinas virtuais.

3.1.5 Tipos de Máquinas Virtuais

Provedores costumam classificar as máquinas virtuais fornecidas conforme suas características, de modo a manter uma linha de produtos discreta e finita. Normalmente essa classificação se dá em termos de quantidade de memória RAM, quantidade de espaço em disco e capacidade computacional, neste caso, quer seja em termos relativos a um valor padrão tomado como base, quer seja em termos absolutos, como número de CPUs virtuais.

3.1.6 Categorias de Máquinas Virtuais

Tipos de Máquinas Virtuais podem ser agrupados em Categorias, conforme suas características físicas, plataforma e/ou arquitetura de hardware e a natureza do uso a que se destinam. Dentro de uma mesma Categoria, os Tipos de Máquinas Virtuais variam apenas na quantidade de cada um dos recursos especificados para a Categoria e no preço cobrado pelo uso das máquinas virtuais.

Como exemplo, podemos citar uma Categoria de máquinas destinadas a armazenamento de arquivos, onde as máquinas devem privilegiar o espaço de armazenamento em massa. Dentro dessa categoria, a principal diferença entre os Tipos de Máquinas Virtuais se dá em função da quantidade de espaço em disco disponibilizado, enquanto características como memória RAM e CPU teriam pequenas variações. Outras Categorias podem enfatizar o consumo de banda de rede ou processamento paralelo de alto desempenho.

3.1.7 Configurações

Chamamos de Configuração um conjunto de máquinas virtuais pertencentes ao mesmo Tipo de Máquinas Virtuais (e, portanto, de uma mesma Categoria) e que será aplicado a uma camada arquitetural da aplicação sob estudo (apresentação, negócio, persistência, etc.). Uma Configuração representa o estado de uma determinada camada da aplicação quanto à sua escalabilidade, vertical ou horizontal.

Por exemplo, poderíamos avaliar o comportamento de uma Aplicação sob Teste implantada com sua camada de aplicação em arquitetura de *cluster,* composta por duas, três ou quatro instâncias em paralelo, caracterizando a variação dessa camada em diferentes níveis de escalabilidade horizontal. Para esse teste, teríamos então três Configurações diferentes, a primeira com duas instâncias, a segunda com três e a terceira com quatro instâncias de máquinas do mesmo Tipo de Máquina Virtual.

As heurísticas desenvolvidas e apresentadas por este trabalho comparam implantações de diferentes Configurações em uma determinada camada da Aplicação sob Teste estudada. Isso permite que sejam feitas avaliações como a viabilidade financeira da escalabilidade vertical face ao desempenho possivelmente obtido com a escalabilidade horizontal.

3.1.8 Espaço de Implantação

Chamamos de Espaço de Implantação o conjunto limitado de Configurações tomadas para execução da Aplicação sob Teste em uma sessão de avaliação.

Idealmente, uma Aplicação deveria ser testada sob todos os Tipos de Máquinas Virtuais fornecidos pelo Provedor (cobrindo todo o espaço de escalabilidade vertical) com o maior número possível de combinações de quantidade de instâncias (cobrindo o espaço de escalabilidade horizontal). Porém, se muitos Tipos de Máquinas Virtuais forem necessários e se o intervalo de número de instâncias solicitado for muito grande, o tempo de duração da sessão e o custo da muitas execuções podem se tornar proibitivos.

Assim, o processo de especificação de um Espaço de Implantação consiste em selecionar uma lista de Tipos de Máquinas Virtuais entre os oferecidos pelo Provedor e designar o melhor valor para o número máximo de instâncias que serão usadas na criação das Configurações. Isso faz com que ambos os espaços de escalabilidade vertical e horizontal sejam limitados, de forma a controlar melhor os custos e permitir que sejam executados testes mais objetivos e de acordo com a meta de Carga de Trabalho a ser atendida pela Aplicação.

3.1.9 Carga de Trabalho

A Carga de Trabalho, ou *Workload*, representa o tamanho da demanda que será imposta à Aplicação sob Teste em uma execução. A unidade de medida da Carga é dependente do domínio da Aplicação, mas, para efeito deste trabalho, é irrelevante, uma vez que a responsabilidade pela execução dos testes e, por conseguinte, pela geração da carga, é delegada a um módulo à parte dentro sistema de avaliação.

3.1.10 Execução

Damos o nome de Execução ao evento de utilização de uma Configuração para executar a Aplicação sob Teste submetida a uma determinada Carga de Trabalho. Dessa forma, a avaliação dos Resultados de uma Execução nos dará uma ideia de como a Aplicação responderá às requisições de certo número de usuários (workload) após ser implantada num ambiente de nuvem com certo grau de escalabilidade horizontal (número de máquinas virtuais usadas).

3.2 Formalismos

Dada uma Aplicação sob Teste **A**, precisamos identificar, dentre um conjunto de cenários de implantação e execução da aplicação em ambiente de nuvem computacional sob a modalidade de infraestrutura como serviço, sob quais desses cenários a aplicação é executada com sucesso e, dentre esses cenários, quais os de menor custo.

3.2.1 Configurações

\*\*\*\*\*\*\*\*\*\* formalizar Espaco, TipoVM, e formalizar C em função destes

Seja **C** um conjunto de Configurações, onde **c1, c2, c3...c|C|** são identificadores de Configurações tomadas de um Espaço de Implantação definido a partir dos Tipos de Máquinas Virtuais fornecidas por um Provedor P.

3.2.2 Cargas de Trabalho

Seja **W** um conjunto de valores de cargas de trabalho, onde **w1, w2, w3... w|W|** são valores que determinam o tamanho, volume ou intensidade da carga de trabalho.

3.2.3 Métricas de Desempenho

Seja **M** um conjunto de métricas, onde **m1, m2, m3...m|M|** são identificadores de métricas usadas para avaliar o desempenho de uma aplicação.

3.2.4 Execuções e Resultados \*\*\*\*\*\*\*\*\*\*\* SEPARAR! \*\*\*\*\*\*\*\*\*\*\*\*\*

Seja **E** um conjunto de execuções, onde os elementos **e1, e2, e3...e|E|** são quíntuplas no formato **{s, w, m, r, l}** com **s S**, **w W**, **m M**, e onde **r** é o resultado de uma execução para a métrica **m**, no cenário **s** sob a carga de trabalho **w** e onde **l** é uma lista com o nível de utilização de recursos para essa execução em valores percentuais. Cada item da lista **l** representa um recurso físico utilizado durante a execução, como CPU ou memória.

Seja um valor de referência definido como parâmetro de sucesso para uma métrica quando da execução de um teste em um cenário.

Seja **atende(e, alfa),** onde **e** **E**, uma função tal que:

* se a métrica **e[m]** é minimizável
* se a métrica **e[m]** é maximizável

Em ambos os casos, representa a distância entre o resultado da execução **e** (ou seja, o valor de **e[r]**)

Seja **P** a matriz de desempenho definida como:

Cada elemento da matriz é definido como:

onde || é a contagem de execuções para o i-ésimo cenário S, a j-ésima carga de trabalho W e a n-ésima métrica M.

Cada elemento da matriz P informa se a execução da aplicação em um cenário de implantação foi bem sucedida sob uma determinada carga de trabalho. Assim, temos que a matriz de desempenho P informa o grau de sucesso de execução da aplicação estudada considerando todas as métricas avaliadas para cada cenário sob cada carga de trabalho.

4. Esquema de Solução

O objetivo deste trabalho é estudar heurísticas de preenchimento da matriz P descrita no capítulo anterior sem necessariamente ter que executar de fato todos os testes necessários. Esse preenchimento deverá ser feito, assim, por meio de um motor de predições que executará uma ou mais estratégias para preencher a matriz P com resultados calculados a partir dos dados de entrada.

Consideraremos como dados de entrada os resultados de uma ou mais execuções reais para um ou mais cenários de teste. A precisão dos resultados obtidos pelo motor de predição vai depender da qualidade da estratégia escolhida e da quantidade de dados de entrada. Quanto mais diversificadas quanto aos cenários a que forem aplicadas e quanto maior o número de execuções reais usadas para alimentar inicialmente o motor de predições, mais preciso tenderá a ser o resultado da predição, porém mais caro se tornará o processo, uma vez que os testes executados no ambiente de nuvem incorrem em custo financeiro.

Apresentaremos um motor capaz de executar heurísticas de predição e um arcabouço de implementação dessas heurísticas, de forma que nova inteligência de predição possa ser agregada ao trabalho futuramente. Apresentaremos também, como forma de validar a proposta do arcabouço e do motor de execuções, duas heurísticas para geração da matriz P preenchida e sugestão da configuração de menor custo para executar a aplicação alvo em ambiente de nuvem de infraestrutura.

A figura 1 mostra um diagrama em que é apresentada a arquitetura de alto nível da solução proposta neste trabalho. A imagem destaca os dados de entrada e saída, bem como a matriz P gerada e representada como a planilha de desempenho dos resultados intermediários. No centro pode-se observar a estrutura lógica da solução, tendo o motor de execução no nível mais alto, suportado abaixo pelo arcabouço de implementação de heurísticas e, na base, as heurísticas definidas pelo usuário.

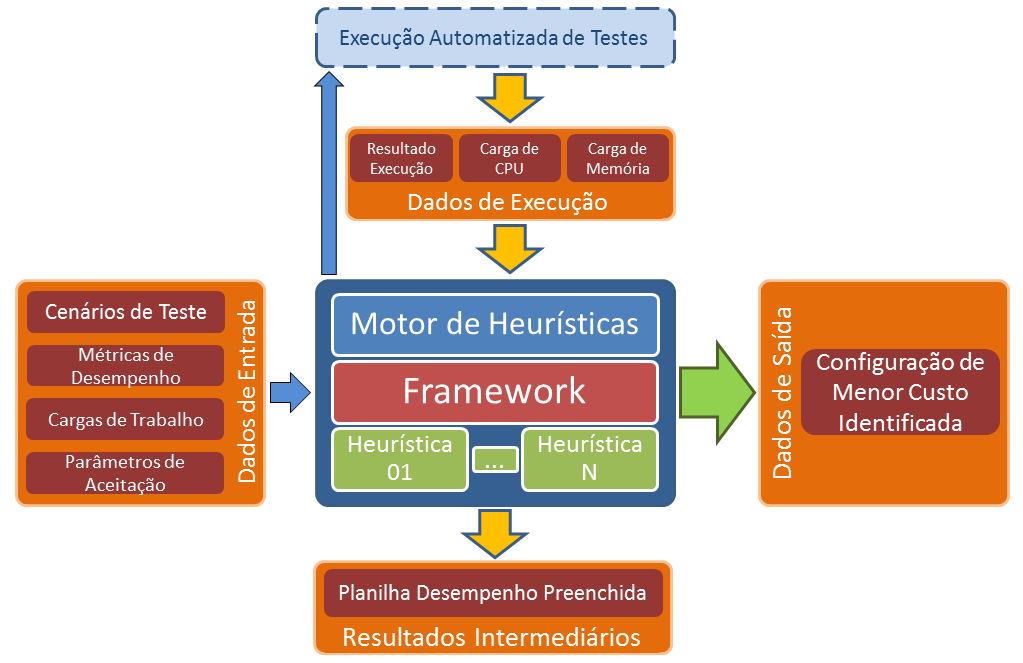


Figura 1 - Arquitetura de Alto Nível da Solução Proposta

A comunicação com o módulo externo de execução automatizada de testes será feita através de componentes plugáveis, que obedecerão a uma interface específica definida na solução. A implementação proposta oferecerá compatibilidade com o *Cloud Crawler* [CUNHA et. al., 2012] e com uma implementação abstrata que simulará a presença de um módulo externo. Essa implementação abstrata será útil quando já houver dados de execuções pré-existentes, que serão dados como resposta por essa implementação, de modo que não seja necessário repetir testes já executados.

5. Planejamento

[cronograma]

Bibliografia

CUNHA, M.; MENDONCA, N.; SAMPAIO, A. Investigating the Impact of Deployment Conﬁguration and User Demand on a Social Network Application in the Amazon EC2 Cloud. In: IEEE COMPUTER SOCIETY. Cloud Computing Technology and Science (CloudCom), 2011 IEEE Third International Conference on. [S.l.], 2011. p. 746-751

CUNHA, M.; MENDONÇA, N.; SAMPAIO, A. Um ambiente programável para avaliar o desempenho de aplicações em nuvens de infraestrutura. Dissertação de Mestrado. Universidade de Fortaleza, 2012.

IOSUP, A.; YIGITBASI, N.; EPEMA, D. On The Performance Variability of Production Cloud Services. 11th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid), 2011.

JAYASINGHE, D.; MALKOWSKI, S.; WANG, Q.; LI, J.; XIONG, P.; PU, C. Variations in Performance and Scalability When Migrating N-Tier Applications to Different Clouds. Cloud Computing (CLOUD), 2011 IEEE International Conference on. IEEE, 2011.

LI, A.; ZONG, X.; ZHANG, M.; KANDULA, S.; YANG, X. CloudProphet: Predicting Web Application Performance in the Cloud. Duke University Technical Report CS-2011-11, 2011.

RODERO-MERINO, L.; VAQUERO, L. M.; GIL, V.; GALÁN, F.; FONTÁN, J.; MONTERO, R. S.; LLORENTE, I. From Infrastructure delivery to service management in clouds. In: Future Generation Computer Systems, 2010, p. 1226 – 1240.

SILVA, M. A.; HINES, M. R.; GALLO, D.; LIU, Q.; RYU, K. D.; SILVA, D. CloudBench: Experiment Automation for Cloud Environments. IEEE International Conference on Cloud Engineering (IC2E), 2013.