

Um Framework para Descoberta de Recursos usando Computação nas Nuvens

Rodrigo Sforini Mota¹, Pablo Diego Silva da Silva¹,
Clézio Silva de Souza¹, Raphael de Aquino Gomes^{1,2}

¹Instituto Federal de Goiás - Campus Goiânia
Rua 75, n. 46, Centro. CEP: 74055-110 - Goiânia - GO - Brasil

²Instituto de Informática – Universidade Federal de Goiás
Câmpus Samambaia, Caixa Postal 131, Goiânia - GO, Brasil

primeironome.ultimonome@estudantes.ifg.edu.br, raphael.gomes@ifg.edu.br

Abstract. *The allocation of resources needed to deploy an application is usually performed in an ad hoc way. In this paper we propose a framework to facilitate cloud-based resources specification and discovery. The framework has been partially implemented, and its effectiveness was evaluated.*

Resumo. *A definição de recursos necessários para implantar uma aplicação é comumente executada de maneira ad hoc, o que pode ocasionar perda de qualidade. Neste trabalho propomos um framework para facilitar a especificação e descoberta de recursos em ambientes de computação nas nuvens. O framework foi parcialmente implementado, sendo sua eficácia comprovada.*

1. Introdução

Uma dificuldade comum no desenvolvimento de software é realizar sua implantação. Após a definição da infraestrutura necessária, a implantação comumente é feita mediante a alocação de recursos pré-existentes ou com alocação de capital para a aquisição de novos recursos. Contudo, em diversos cenários nenhuma destas alternativas é viável devido a limitações financeiras. Dessa forma, Computação nas Nuvens constitui uma alternativa promissora. Este modelo permite acesso sob demanda, através de rede, a um conjunto de recursos computacionais (rede, armazenamento, processamento e aplicações) configuráveis, que podem ser rapidamente oferecidos e liberados com mínimo esforço de gerenciamento e interação por seu provedor [Mell e Grance 2009].

Este trabalho apresenta um framework para definição e descoberta de recursos usando computação nas nuvens. É possível encontrar os recursos necessários utilizando múltiplos provedores. O principal objetivo deste trabalho é propor uma alternativa que abstraí detalhes da seleção de recursos e permite analisar a viabilidade de ambientes de nuvem para o provisionamento destes recursos. O framework proposto foi avaliado de forma qualitativa através de um estudo de caso utilizando uma aplicação de uma fábrica de software. O restante do artigo é organizado da seguinte forma: a Seção 2 discute alguns trabalhos relacionados; a Seção 3 apresenta o framework, ao passo que sua avaliação é discutida na Seção 4; na Seção 5 são apresentadas algumas considerações finais.

2. Trabalhos Relacionados

Computação nas Nuvens tem como tecnologia base Arquiteturas Orientadas a Serviços (SOA) para gerenciar os serviços oferecidos e controlar a distribuição, acesso e armazenamento dos dados. Desta forma, os trabalhos mais relacionados a ambientes de nuvem são aqueles que tratam ambientes de SOA. Dentre estes, Wright et al. [Wright et al. 2012] propuseram um modelo baseado em restrições para especificação e seleção de recursos em ambientes de nuvem. Assim como propomos, o modelo é focado na aplicação, de forma que os recursos são especificados usando uma linguagem independente do provedor. Contudo, a linguagem proposta é exclusiva do modelo, dificultando sua utilização de forma ampla. Em [Khajeh-Hosseini et al. 2011] é apresentado o *Cloud Adoption Toolkit* que consiste em um framework organizacional usado para identificar os aspectos relacionados à adoção de ambientes de nuvem com ferramentas para apoiar decisões do cliente. Neste trabalho não é apresentada nenhuma ferramenta que apoie a obtenção de recursos.

3. Framework para Descoberta de Recursos

O modelo proposto para representação dos recursos é um subconjunto do padrão *Open Cloud Computing Interface (OCCI)*¹, que consiste em um conjunto de especificações abertas desenvolvido pelo *Open Grid Forum*², uma comunidade voltada para padronização de computação nas nuvens.

Os tipos de recursos disponíveis no modelo inclui: i) *Resource* que representa um tipo abstrato para as demais classes; ii) *Network* que representa uma entidade de rede (ex. switch). Possui como atributos um identificador VLAN 802.1q; um *token* e o estado corrente da VLAN; iii) *Compute* que representa um recurso genérico de processamento (ex. máquina virtual). Possui como atributos a arquitetura da CPU, o número de núcleos, a frequência, o hostname DNS do recurso, a quantidade de memória RAM, e o estado; e iv) *Storage* que representa recurso que armazenam informação em um dispositivo de armazenamento de dados. Possui como atributo o tamanho do armazenamento em gigabytes.

O módulo responsável pela descoberta de recursos é denominado *Cloud Service Discovery (CSD)*. Sua arquitetura é apresentada na Figura 1. A linguagem utilizada para definição de recursos no CSD é uma implementação do padrão OCCI. Os demais componentes do CSD são descritos a seguir:

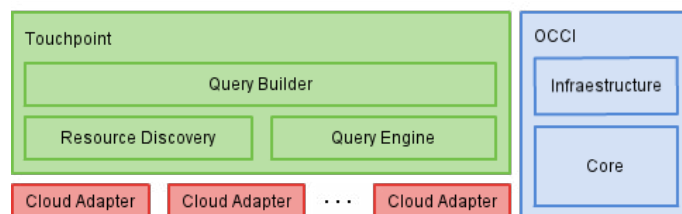


Figura 1. Arquitetura do framework *Cloud Service Discovery*.

O *Touchpoint* constitui o núcleo do módulo. Tem como responsabilidade a agregação dos vários provedores de nuvem, provendo um ponto único de acesso aos mesmos, além de gerar e processar as consultas a serem realizadas. Estas consultas são

¹<http://www.occi-wg.org>

²<http://www.ogf.org>

geradas pelo componente *Query Builder* a partir da definição de recursos da aplicação. A interpretação e processamento das consultas por sua vez são realizados pelo *Query Engine*. O gerenciamento dos provedores de nuvem é feito pelo componente *Service Discovery*. Para um provedor ser considerado nas buscas realizadas ele deve implementar a interface estabelecida pelo *Touchpoint* e se registrar no *Service Discovery*.

O *Cloud Service Provider Adapter* representa a abstração de um provedor de nuvem. Consiste em uma adaptador que deve ser implementada pelo provedor para que ele seja considerado nas consultas realizadas.

Os módulos do framework são executados na estação cliente, sendo a comunicação com o provedor de responsabilidade do adaptador. Além disso, as atividades de descoberta e especificação de recursos são feitas como uma etapa anterior à implantação da aplicação, não afetando na eficiência da mesma.

O framework foi implementado em Java. Para a implementação da descrição dos serviços em OCCI foi realizada uma adaptação da implementação OCCI4Java³. Para permitir a avaliação da abordagem, foi implementado um adaptador para provedor de nuvem. O provedor escolhido foi Amazon⁴, uma vez que este é um dos produtos dominantes na área e já oferece um SDK em Java⁵, permitindo sua fácil integração à implementação realizada. Neste protótipo também não foi considerada a implementação do componente *Query Builder*, de forma que as consultas são construídas manualmente.

4. Avaliação

A avaliação foi realizada através de uma aplicação que visa controlar as atividades pertinentes à realização de análises clínicas de um laboratório. O sistema possui interface Web e foi desenvolvido em PHP, sendo necessário portanto um servidor web e um sistema gerenciador de banco de dados para realizar sua implantação.

Com base nos requisitos descritos, e com a restrição de que a implantação deveria demandar o mínimo de gastos possível pela organização, foi obtida a descrição de recursos apresentada na Figura 2, de acordo com a qual seria necessário um banco de dados com espaço disponível de 10 GB e dois servidores, com arquitetura x86 e configuração mínima de processador com 8 núcleos e memória RAM com 7,5 GB. A duplicação dos servidores se deu visando redundância para atender o requisito de confiabilidade.

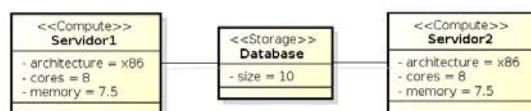


Figura 2. Descrição dos recursos necessários para a aplicação.

Com base nesta descrição de recursos foram geradas as consultas submetidas ao framework. O primeiro passo consiste em definir a consulta utilizando a API implementada pelo componente *Query Engine*. Depois, esta é submetida ao serviço de descobertas. Como resultado é retornado um mapa cujas chaves são identificadores dos provedores

³<https://github.com/occi4java>

⁴<http://aws.amazon.com>

⁵<http://aws.amazon.com/sdkforjava>

cadastrados no sistema; e os valores são recursos que satisfazem a consulta realizada. É importante observar que os recursos retornados tem o mínimo da configuração de hardware solicitada na consulta, sem necessariamente possuir o valor exato.

Realizamos um experimento para comparar a eficácia e eficiência da busca de recursos usando o framework ou usando diretamente a API oferecida pela Amazon.

A mesma consulta foi realizada 10 vezes em ambas as abordagens. Usando diretamente a API Java da Amazon foram encontradas 7248 imagens satisfatórias em um tempo médio de 34,23 segundos, com desvio padrão de 12,06 segundos. Usando o framework foram obtidas 7248 imagens em um tempo médio de 40,69 segundos, com desvio padrão de 10,43. A comparação dos dados permitiu verificar que os resultados eram os mesmos, comprovando a eficácia da proposta. Com relação à eficiência, foi verificada um aumento médio de 6,46 segundos com o uso do framework, o que não constitui um grande problema porque sua utilização é feita de maneira *offline*, sem comprometer a aplicação. Além disso, este resultado pode ter ocorrido em virtude de variações no provedor de serviços, o que pode ser verificado com o alto desvio padrão medido. Os principais resultados da consulta obtidos com o framework são apresentados na Tabela 1.

ID	Tipo de Instância	Cores	RAM (GB)	Custo por hora (\$)
ami-daf005b3	m4.2xlarge	8	32.0	0.54
ami-8ec035e7	m4.2xlarge	8	32.0	0.54
ami-987185f1	m4.2xlarge	8	32.0	0.54
ami-d59d6bbc	m4.2xlarge	8	32.0	0.54
ami-9f4082f6	m4.2xlarge	8	32.0	0.54

Tabela 1. Resultado da busca realizada no experimento (5 melhores instâncias).

5. Considerações Finais

Neste artigo foi apresentado um framework para descoberta de recursos em ambientes de nuvens. O framework pode ser usado para apoiar as atividades de implantação do produto desenvolvido. Ele foi avaliado com uma aplicação de exemplo que demonstrou sua eficácia e eficiência. Como melhorias a serem incluídas no framework propõe-se o suporte a um número maior de tipos de recursos e provedores de nuvem, além de um mecanismo mais aperfeiçoado de seleção dos recursos a serem utilizados.

Agradecimentos

Este trabalho recebe auxílio financeiro das fundações FAPEG (chamadas # 04/2011, 12/2012 e 03/2013) e CNPq (contratos # 249809/2013-3 e 473939/2012-6).

Referências

- Khajeh-Hosseini, A., Greenwood, D., Smith, J. W., e Sommerville, I. (2011). The Cloud Adoption Toolkit: supporting cloud adoption decisions in the enterprise. *Software: Practice and Experience*.
- Mell, P. e Grance, T. (2009). Draft NIST working definition of cloud computing.
- Wright, P., Sun, Y., Harmer, T., Keenan, A., Stewart, A., e Perrott, R. (2012). A constraints-based resource discovery model for multi-provider cloud environments. *Journal of Cloud Computing: Advances, Systems and Applications*, 1(1):6.