Capítulo

5

Elasticidade em Computação na Nuvem: Uma Abordagem Sistemática

Emanuel F. Coutinho, Flávio R. C. Sousa, Danielo G. Gomes
e José N. de Souza

Abstract

Elasticity is a key feature of cloud computing. This feature is the ability to add or remove resources without interruptions and at runtime to handle the load change. However, it difficult to understand the requirements of a specific elastic application, their workload and mainly as a provider must manage resources to meet these requirements. There have been some work on characterization of elasticity in the cloud. Typically these studies are not conducted in a systematic way, and generally are little broad and their results are hard to replicate. Moreover, a systematic review allows to identify, evaluate and interpret relevant results for a given search topic, phenomenon of interest or research question. The objective of this short course to present the state of the art of elasticity. based on an adaptation of a classic systematic review. It will be highlight different aspects of elasticity, such as definitions, metrics and benchmarks for measuring and evaluating the elasticity, strategies for use, and finally challenges and trends in the construction of elastic solutions.

Resumo

A elasticidade é uma das principais características da Computação em Nuvem. Esta característica consiste na capacidade de adicionar ou remover recursos, sem interrupções e em tempo de execução para lidar com a variação da carga. Contudo, é difícil compreender os requisitos de elasticidade de uma aplicação específica, de sua carga de trabalho e principalmente como um provedor deve gerenciar os recursos para atender esses requisitos. Existem alguns estudos sobre as características de elasticidade na nuvem. Normalmente esses estudos não são realizados de uma maneira sistemática e, em geral, são pouco abrangentes e difíceis de reproduzir. Por outro lado, uma revisão sistemática permite identificar, avaliar e interpretar resultados relevantes de um determinado tópico de pesquisa, fenômeno de interesse ou questão de pesquisa. O objetivo deste minicurso é,

a partir de uma proposta de adaptação de uma revisão sistemática clássica, apresentar o estado da arte sobre elasticidade. Serão destacados diferentes aspectos da elasticidade, tais como definições, métricas e benchmarks para a medição e avaliação da elasticidade, estratégicas de utilização, e por fim, desafios e tendências na construção de soluções elásticas.

5.1. Introdução

A Computação em Nuvem propõe a integração de diversos modelos tecnológicos para o provimento de infraestrutura de hardware, plataformas de desenvolvimento e aplicações na forma de serviços sob demanda com pagamento baseado em uso [Sá et al. 2011]. Neste novo paradigma de utilização de recursos computacionais, clientes abrem mão da administração de uma infraestrutura própria e dispõem de serviços oferecidos por terceiros, delegando responsabilidades e assumindo custos estritamente proporcionais à quantidade de recursos que utilizam.

Os serviços em nuvem apresentam diversas vantagens para os usuários, tais como: previsibilidade e custos mais baixos, proporcionais à qualidade de serviço (QoS) e cargas de trabalho reais; complexidade técnica reduzida, graças à interfaces de acesso unificado e administração simplificada; e elasticidade e escalabilidade, proporcionando a percepção recursos quase infinitos. Por outro lado, o provedor tem que garantir a ilusão de recursos infinitos sob cargas de trabalho dinâmicas e minimizar os custos operacionais associados a cada usuário [Sousa et al. 2010].

Os ambientes em nuvem caracterizam-se por serem intrinsecamente distribuídos e compostos por recursos heterogêneos, além de servirem concomitantemente a uma grande diversidade de clientes com requisitos de QoS distintos [Sousa et al. 2011]. Adicionalmente, os *datacenters* que compõem uma nuvem devem suportar uma grande quantidade de aplicações, o que os obriga a lidar com demandas variantes por processamento, armazenamento de dados e utilização de banda.

O gerenciamento do ambiente de nuvem possui algumas características peculiares que diferem de outros ambientes, dentre as quais pode-se destacar a intervenção humana limitada, carga de trabalho altamente variável e uma grande quantidade e variedade de recursos compartilhados. De acordo com as requisições dos usuários, estes recursos são expandidos ou reduzidos para manter a QoS [Sousa et al. 2010].

Apesar das limitações de rede e segurança, as soluções em nuvem devem fornecer um elevado desempenho, além de serem flexíveis para se adaptarem diante de uma determinada quantidade de requisições. Como, em geral, os ambientes de Computação em Nuvem possuem acesso público, torna-se imprevisível e variável a quantidade de requisições realizadas, dificultando fazer estimativas e fornecer garantias de QoS [Suleiman et al. 2012]. Essas garantias de qualidade são definidas entre o provedor do serviço e o usuário e expressas por meio de um acordo de nível de serviço (SLA) [Sousa and Machado 2012], que consiste de contratos que especificam um nível de desempenho que deve ser atendido e penalidades em caso de falha.

A elasticidade é ponto chave para implementar serviços com QoS para nuvem, pois permite adicionar ou remover recursos, sem interrupções e em tempo de execução

para lidar com a variação da carga. De acordo com [Mell and Grance 2009], estes recursos podem ser adquiridos de forma rápida, em alguns casos automaticamente, caso haja a necessidade de escalar com o aumento da demanda, e liberados, na retração dessa demanda. Para os usuários, os recursos disponíveis para uso parecem ser ilimitados e podem ser adquiridos em qualquer quantidade e a qualquer momento.

Contudo, a nuvem apresenta uma variabilidade de desempenho bastante elevada [Schad et al. 2010], principalmente devido à heterogeneidade de hardware, virtualização e compartilhamento de recursos, dificultando o desenvolvimento de soluções com QoS na nuvem. Além disso, é difícil estabelecer os requisitos de elasticidade de uma aplicação específica e de sua carga de trabalho. O provedor de serviços de nuvem não pode prever como os clientes irão utilizar seus serviços. Dessa forma, é essencial compreender as características da elasticidade na nuvem e sua relação com a qualidade dos serviços neste ambiente.

Apesar de vários trabalhos de Computação em Nuvem explorarem o tema da elasticidade, como em [Suleiman et al. 2012], [Islam et al. 2012], [Sharma et al. 2011], [Costa et al. 2011], [Galante and de Bona 2012], o processo de análise do estado da arte para a seleção de trabalhos correlatos frequentemente carece de uma sistemática em sua realização, o que pode implicar em uma revisão da literatura pouco abrangente, com trabalhos relacionados de difícil reprodução e cuja seleção depende fortemente de quem a realiza.

Assim, o desenvolvimento de uma abordagem sistemática de revisão auxilia no estabelecimento de um processo formal para este tipo de investigação. A revisão sistemática é um estudo conduzido de maneira a identificar, avaliar e interpretar resultados relevantes de um determinado tópico de pesquisa, fenômeno de interesse ou questão de pesquisa [Mafra and Travassos 2006].

Este minicurso tem como objetivo propor uma adaptação simplificada da revisão sistemática definida por [Kitchenham 2004] de maneira a identificar trabalhos relacionados à elasticidade em Computação em Nuvem. São destacados diferentes aspectos da elasticidade, tais como definições e o estado da arte da elasticidade em Computação em Nuvem. Para isso, é apresentada uma adaptação de uma revisão sistemática aplicada ao contexto de elasticidade em ambientes de Computação em Nuvem, destacando aspectos de análise de desempenho, métricas, estratégicas elásticas, *benchmarks* para a medição e avaliação da elasticidade, e desafios e tendências na construção de soluções elásticas.

Este minicurso é teórico, com uma descrição detalhada de todas as etapas do planejamento do estudo e uma análise crítica acerca dos aspectos encontrados nos trabalhos resultantes desta busca criteriosa e sistemática, suas diferenças, interseções, problemas mais relevantes e em aberto relacionados à elasticidade em ambientes de computação em nuvem.

5.2. Computação em Nuvem

A Computação em Nuvem está se tornando uma das palavras chaves da indústria de Tecnologia da Informação (TI). A nuvem é uma metáfora para a Internet ou infraestrutura de comunicação entre os componentes arquiteturais, baseada em uma abstração que oculta

à complexidade da infraestrutura. Cada parte desta infraestrutura é provida como um serviço os quais normalmente são alocados em centros de dados, utilizando hardware compartilhado para computação e armazenamento [Buyya et al. 2009].

A infraestrutura do ambiente de Computação em Nuvem normalmente é composta por um grande número, centenas ou milhares de máquinas físicas ou nós físicos de baixo custo, conectadas por meio de uma rede como ilustra a Figura 5.1. Cada máquina física tem as mesmas configurações de software, mas pode ter variação na capacidade de hardware em termos de CPU, memória e armazenamento em disco [Soror et al. 2010]. Dentro de cada máquina física existe um número variável de máquinas virtuais ou nós virtuais em execução, de acordo com a capacidade do hardware disponível na máquina física. Os dados são persistidos, geralmente, em sistemas de armazenamento distribuídos.

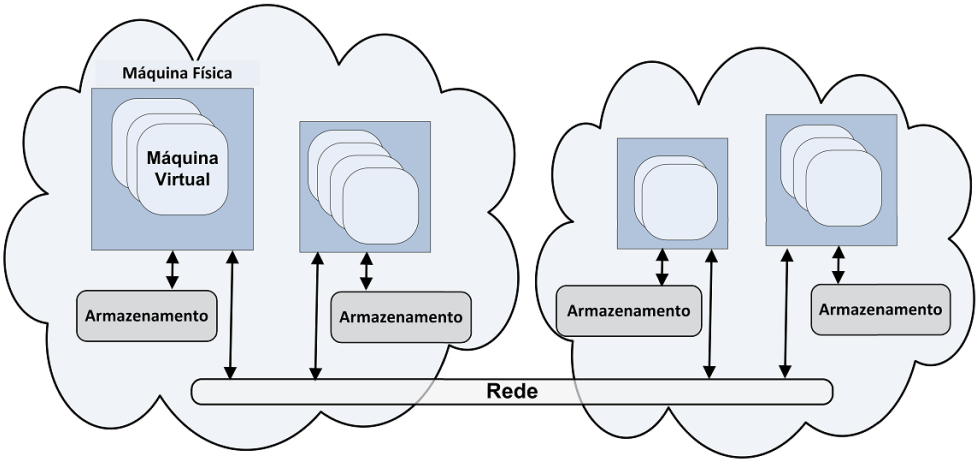

Figura 5.1. Ambiente de Computação em Nuvem

A Computação em Nuvem é uma evolução dos serviços e produtos de tecnologia da informação sob demanda, também chamada de *Utility Computing* [Brantner et al. 2008]. O objetivo da *Utility Computing* é fornecer componentes básicos como armazenamento, processamento e largura de banda de uma rede como uma "mercadoria" através de provedores especializados com um baixo custo por unidade utilizada. Usuários de serviços baseados em *Utility Computing* não precisam se preocupar com escalabilidade, pois a capacidade de armazenamento fornecida é praticamente infinita. A *Utility Computing* propõe fornecer disponibilidade total, isto é, os usuários podem ler e gravar dados a qualquer tempo, sem nunca serem bloqueados; os tempos de resposta são quase constantes e não dependem do número de usuários simultâneos, do tamanho do banco de dados ou de qualquer parâmetro do sistema. Os usuários não precisam se preocupar com *backups*, pois se os componentes falharem, o provedor é responsável por substituí-los e tornar os dados disponíveis em tempo hábil por meio de réplicas [Brantner et al. 2008].

Uma razão importante para a construção de novos serviços baseados em *Utility Computing* é que provedores de serviços que utilizam serviços de terceiros pagam apenas pelos recursos que recebem, ou seja, pagam pelo uso. Não são necessários grandes investimentos iniciais em TI e o custo cresce de forma linear e previsível com o uso. Dependendo do modelo do negócio, é possível que o provedor de serviços repasse o custo

de armazenagem, processamento e de rede para os usuários finais, já que é realizada a contabilização do uso.

Existem diversas propostas para definir o paradigma da Computação em Nuvem [Vaquero et al. 2009]. O *National Institute of Standards and Technology* (NIST) argumenta que a Computação em Nuvem é um paradigma em evolução e apresenta a seguinte definição: *"Computação em nuvem é um modelo que possibilita acesso, de modo conveniente e sob demanda, a um conjunto de recursos computacionais configuráveis (por exemplo, redes, servidores, armazenamento, aplicações e serviços) que podem ser rapidamente adquiridos e liberados com mínimo esforço gerencial ou interação com o provedor de serviços"* [Mell and Grance 2009]. Ainda segundo o NIST, a Computação em Nuvem é composta por cinco características essenciais, três modelos de serviço e quatro modelos de implantação, detalhados a seguir.

5.2.1. Características Essenciais

As características essenciais são vantagens que as soluções de Computação em Nuvem oferecem. Algumas destas características, em conjunto, definem exclusivamente a Computação em Nuvem e fazem a distinção com outros paradigmas. Por exemplo, a elasticidade rápida de recursos, amplo acesso e a medição de serviço são características básicas para compor uma solução de Computação em Nuvem.

- *Self-service sob demanda*: O usuário pode adquirir unilateralmente recurso computacional, como tempo de processamento no servidor ou armazenamento na rede, na medida em que necessite e sem precisar de interação humana com os provedores de cada serviço.
- *Amplo acesso*: Recursos são disponibilizados por meio da rede e acessados através de mecanismos padronizados que possibilitam o uso por plataformas do tipo *thin*, tais como celulares, *laptops* e PDAs.
- *Pooling de recursos*: Os recursos computacionais do provedor são organizados em um *pool* para servir múltiplos usuários usando um modelo *multi-tenant* ou multi-inquilino, com diferentes recursos físicos e virtuais, dinamicamente atribuídos e ajustados de acordo com a demanda dos usuários. Estes usuários não precisam ter conhecimento da localização física dos recursos computacionais, podendo somente especificar a localização em um nível mais alto de abstração, tais como o país, estado ou centro de dados.
- *Elasticidade rápida*: Recursos podem ser adquiridos de forma rápida e elástica, em alguns casos automaticamente, caso haja a necessidade de escalar com o aumento da demanda, e liberados, na retração dessa demanda. Para os usuários, os recursos disponíveis para uso parecem ser ilimitados e podem ser adquiridos em qualquer quantidade e a qualquer momento.
- *Serviço medido*: Sistemas em nuvem automaticamente controlam e otimizam o uso de recursos por meio de uma capacidade de medição. A automação é realizada em algum nível de abstração apropriado para o tipo de serviço, tais como armazenamento, processamento, largura de banda e contas de usuário ativas. O uso

de recursos pode ser monitorado e controlado, possibilitando transparência para o provedor e o usuário do serviço utilizado.

5.2.2. Modelos de Serviços

O ambiente de Computação em Nuvem é composto de três modelos de serviços. Estes modelos são importantes, pois eles definem um padrão arquitetural para soluções de Computação em Nuvem, conforme descrito na Figura 5.2.

- *Software como um Serviço (SaaS)*: O modelo de SaaS proporciona sistemas de software com propósitos específicos que são disponíveis para os usuários por meio da Internet e acessíveis a partir de vários dispositivos do usuário por meio de uma *interface thin client* como um navegador Web. No SaaS, o usuário não administra ou controla a infraestrutura subjacente, incluindo rede, servidores, sistema operacional, armazenamento ou mesmo as características individuais da aplicação, exceto configurações específicas. Como exemplos de SaaS podemos destacar os serviços de *Customer Relationship Management* (CRM) da Salesforce e o Google Drive.

- *Plataforma como um Serviço (PaaS)*: O modelo de PaaS fornece sistema operacional, linguagens de programação e ambientes de desenvolvimento para as aplicações, auxiliando a implementação de sistemas de software. Assim como no SaaS, o usuário não administra ou controla a infraestrutura subjacente, mas tem controle sobre as aplicações implantadas e, possivelmente, as configurações de aplicações hospedadas nesta infraestrutura. *Google App Engine* [Ciurana 2009] e *Microsoft Azure* [Azure 2012] são exemplos de PaaS.

- *Infraestrutura como um Serviço (IaaS)*: A IaaS torna mais fácil e acessível o fornecimento de recursos, tais como servidores, rede, armazenamento e outros recursos de computação fundamentais para construir um ambiente de aplicação sob demanda, que podem incluir sistemas operacionais e aplicativos. Em geral, o usuário não administra ou controla a infraestrutura da nuvem, mas tem controle sobre os sistemas operacionais, armazenamento, aplicativos implantados e, eventualmente, seleciona componentes de rede, tais como *firewalls*. O Amazon *Elastic Cloud Computing* (EC2) [Robinson 2008] e o Eucalyptus [Liu et al. 2007] são exemplos de IaaS.

5.2.3. Modelos de Implantação

Quanto ao acesso e à disponibilidade, há diferentes tipos de modelos de implantação para os ambientes de Computação em Nuvem. A restrição ou abertura de acesso depende do processo de negócios, do tipo de informação e do nível de visão desejado, conforme descrito a seguir:

- *Nuvem privada*: a infraestrutura de nuvem é utilizada exclusivamente por uma organização, sendo esta nuvem local ou remota e administrada pela própria empresa ou por terceiros.

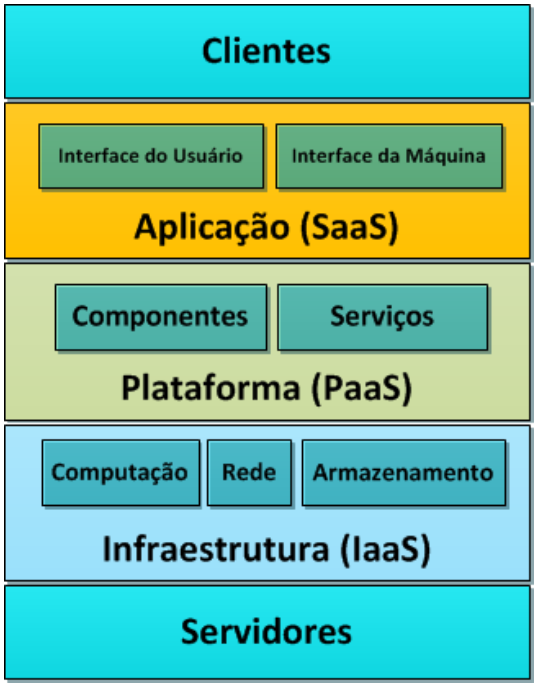

Figura 5.2. Modelo de serviço da Computação em Nuvem (baseado em [Verdi et al. 2010]).

- *Nuvem pública*: a infraestrutura de nuvem é disponibilizada para o público em geral, sendo acessado por qualquer usuário que conheça a localização do serviço.
- *Nuvem comunidade*: fornece uma infraestrutura compartilhada por uma comunidade de organizações com interesses em comum.
- *Nuvem híbrida*: a infraestrutura é uma composição de duas ou mais nuvens, que podem ser do tipo privada, pública ou comunidade e que continuam a ser entidades únicas, mas conectadas por meio de tecnologia proprietária ou padronizada que permite a portabilidade de dados e aplicações.

5.3. Análise de Desempenho

Muitos trabalhos executam atividades para avaliar o desempenho de diversos aspectos, tais como um ambiente, uma aplicação, uma política, um algoritmo, entre outros. Em Computação em Nuvem não é diferente, pois diversas áreas são objetivo de análises. No ambiente de Computação em Nuvem, a análise de desempenho pode ser utilizada para compreender a QoS e possibilitar o ajuste da elasticidade por meio de políticas de alocação de recursos. Será descrita nessa seção alguns conceitos sobre análise de desempenho comumente utilizados em trabalhos científicos.

Análise de desempenho envolve coleta de dados formais e informais para ajudar os clientes e patrocinadores definir e alcançar seus objetivos. Apresenta várias perspectivas sobre um problema ou oportunidade, determinando o direcionamento para barreiras ou desempenho bem sucedido, e propor um sistema de solução com base no que é descoberto. Inclui: medição, modelagem, estatística, projeto experimental, simulação, teoria das filas, etc. Em geral são identificados três aspectos:

- Sistema: Qualquer conjunto de hardware, software e firmware
- Métricas: Critérios utilizados para avaliar o desempenho do sistema ou componentes
- Cargas de trabalho: Requisições realizadas pelos usuários do sistema

É possível realizar diversas atividades com análise de desempenho, tais como especificar requisitos do desempenho, avaliar alternativas do projeto, comparar dois ou mais sistemas determinar o valor ótimo de um parâmetro (ajuste do sistema), encontrar gargalos de desempenho caracterizar a carga de trabalho no sistema, determinação o número e tamanho dos componentes (capacidade planejamento) e predizer o desempenho de futuras cargas. Raj Jain [Jain 1991] realiza um estudo detalhado sobre análise de desempenho. Neste estudo, uma abordagem para avaliação de desempenho é descrita:

- Defina metas e o sistema
- Liste os serviços e seus resultados
- Selecione métricas
- Liste parâmetros
- Selecione fatores para estudo
- Selecione técnica de avaliação
- Selecione carga de trabalho
- Projete os experimentos
- Analise e interprete os dados
- Apresente os resultados

Contudo, não existe a noção de plano de ação conforme os resultados vão sendo coletados e interpretados, e nem há a idéia do ciclo de vida da análise de desempenho. Além disso, Raj Jain aborda predominantemente o projeto do experimento e a intrepretação dos dados. Segundo Douglas Montgomery [Montgomery 2009], um experimento é um teste ou uma série de testes nas quais mudanças intencionais são realizadas para que variáveis de entrada de um processo ou sistema possam ser observadas e identificar as razões para mudanças que podem ser observadas nas respostas de saída. Em geral experimentos são utilizados para estudar o desempenho de processos e sistemas, e estes podem ser representados pelo modelo da figura 5.3. Geralmente um processo é visualizado como uma combinação de máquinas, métodos, pessoas e outros recursos que transformam alguma entrada em alguma saída que possui uma ou mais respostas observáveis. Algumas das variáveis desse sistema são controláveis enquanto que outras não. Os objetivos de um experimento podem incluir o seguinte:

- Determinação de quais variáveis influenciam mais na resposta
- Determinação de onde configurar a variável que mais influencia de maneira que a resposta seja quase sempre próxima ao resultado desejado
- Determinação de onde configurar a variável que mais influencia de modo que a variação na resposta seja pequena
- Determinação de onde configurar a variável que mais influencia de modo que os efeitos das variáveis não controladas sejam minimizados

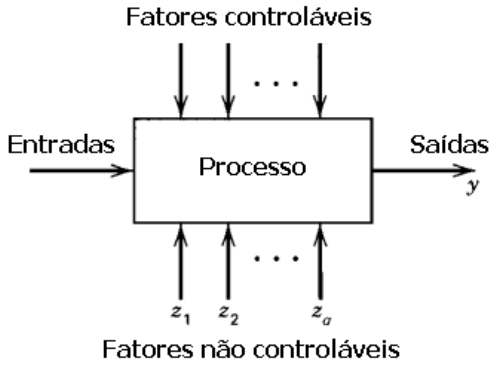

Figura 5.3. Modelo genérico de um processo ou sistema. Fonte: Adaptado de [Montgomery 2009].

Em geral, a abordagem de planejar e conduzir o experimento é chamado de estratégia de experimentação. Projeto experimental é uma importante ferramenta para melhorar o desempenho de processos e sistemas, assim como no desenvolvimento de novas aplicações. A aplicação de técnicas de projeto experimental cedo pode resultar em: rendimento do processo ou sistema melhorado, variabilidade reduzida e conformidade próximos com requisitos ou metas, tempo de desenvolvimento reduzido e redução dos custos globais. Métodos de projeto experimental também ajudam na tomada de decisão, como por exemplo: avaliação e comparação de configurações, avaliação de alternativas, seleção de parâmetros de projeto onde um produto trabalha bem sob uma grande variedade de condições, e determinação de parâmetros de projeto chave que impactam no desempenho do produto. O projeto estatístico de experimentos se refere ao processo de planejamento de maneira que os dados possam ser apropriadamente analisados por métodos estatísticos, resultando em conclusões objetivas e válidas.

Montgomery descreve algumas orientações para o projeto de experimentos:

- Reconhecimento e declaração do problema
- Escolha dos fatores, níveis e intervalos
- Seleção das variáveis de resposta

- Escolha do projeto experimental
- Realização do experimento
- Análise estatística dos dados
- Conclusões e recomendações

Em [Menasce et al. 2004] são descritas quais métricas podem ser agrupadas da seguinte maneira: tempo de resposta (tempo que o sistema leva para reagir a uma requisição humana, geralmente medido em segundos), *throughput* (taxa nas quais requisições são completadas a partir de um sistema computacional, e medido em operações por unidade de tempo), disponibilidade (fração de tempo que um sistema está ativo e disponível para clientes), confiabilidade (probabilidade de que o sistema funcione adequadamente e continuamente sobre um período fixo de tempo), segurança (combinação de confidencialidade, integridade de dados e não repudiação), escalabilidade (um sistema é dito escalável quando seu desempenho não se degrada significantemente com o aumento dos usuários ou carga), e extensibilidade (propriedade do sistema de evolução para lidar com os novos requisitos funcionais e de desempenho).

[Jain 1991] também agrupa as métricas de maneira semelhante, porém mais simplificada: tempo de resposta, *throughput*, utilização (fração de tempo que o recurso está ocupado atendendo requisições), confiabilidade, disponibilidade (*downtime* e *uptime*), aquisição de sistemas (custo e custo/desempenho). Algumas métricas são compartilhadas entre diferentes grupos, dependendo do contexto de utilização, por exemplo: tempo de alocação/desalocação de recursos é um tempo mas é também uma métrica de escalabilidade.

5.4. Elasticidade

Uma das principais razões para a utilização da Computação em Nuvem é a sua capacidade de aquisição de recursos de uma maneira dinâmica. Do ponto de vista do consumidor, a nuvem parece ser infinita pois ele pode adquirir mais ou menos poder computacional conforme for necessário para suas aplicações. Entretanto, é difícil compreender os requisitos de elasticidade de uma aplicação específica e de sua carga de trabalho, e a elasticidade provida por um provedor de Computação em Nuvem deve atender a esses requisitos, pois o provedor de serviços de nuvem não pode prever como os clientes irão utilizar o serviço.

Segundo o NIST, o conceito de elasticidade é a capacidade de rápido provisionamento e desprovisionamento, com capacidade de recursos virtuais praticamente infinita e quantidade adquirível sem restrição a qualquer momento [Mell and Grance 2009].

Elasticidade e escalabilidade são termos que se confundem. Enquanto a escalabilidade tem relação com a capacidade do sistema ser expandido quando necessário, a elasticidade pressupõe a capacidade dos recursos se ajustarem à carga necessária (tanto o aumento quanto a diminuição deles) [Geotecnologias 2012].

Segundo [Taurion 2012], elasticidade é a capacidade do ambiente computacional da nuvem aumentar ou diminuir de forma automática os recursos computacionais demandados e provisionados para cada usuário. É a escalabilidade em duas direções: tanto

cresce quanto diminui a capacidade ofertada. Ao longo da pesquisa diversas definições de elasticidade foram identificadas, conforme Tabela 5.1.

Tabela 5.1. Definições de elasticidade identificadas durante a revisão

Autor	Definição de Elasticidade
[Cooper et al. 2010]	Capacidade de adicionar novas instâncias e distribuir a carga de trabalho para estas instâncias.
[Fito et al. 2010]	Habilidade de adquirir e liberar recursos de granularidades variadas de acordo com a carga da trabalho em um curto intervalo de tempo.
[Aisopos et al. 2011]	Capacidade do provedor alterar dinamicamente a quantidade de recursos de CPU, memória e espaço em disco para uma determinada tarefa.
[Espadas et al. 2011]	Habilidade de criar um número variável de instâncias de máquinas virtuais que dependem da demanda da aplicação.
[Garg et al. 2011] [Garg et al. 2012]	Capacidade de um serviço escalar durante períodos de pico caracterizada pelo tempo médio para expandir ou contrair a capacidade do serviço e capacidade máxima do serviço.
[Li et al. 2011a]	Habilidade do sistema de se adaptar à mudanças repentinas na carga de trabalho.
[Perez-Sorrosal et al. 2011]	Capacidade de aumentar e diminuir a quantidade de réplicas sem interromper o processamento em andamento.
[Han et al. 2012]	Habilidade de escalar recursos de maneira adaptável para cima e para baixo para atender à variação da demanda das aplicações.
[Islam et al. 2012]	Capacidade de provisionar recursos automaticamente e rapidamente.
[Pandey et al. 2012]	Habilidade de um sistema expandir e contrair sem problemas.

Com base nestas definições, este trabalho propõe a seguinte definição para elasticidade *"Capacidade de adicionar e remover recursos de forma automática de acordo com a carga de trabalho sem interrupções e utilizando os recursos de forma otimizada."*. Para implementar a elasticidade, as principais formas utilizadas são técnicas de replicação e redimensionamento de recursos [Sharma et al. 2011].

- Replicação de recursos: Técnicas de replicação são usadas para melhorar a disponibilidade, o desempenho e a escalabilidade em diversos ambientes. Assim, a replicação pode ser utilizada para implementar a elasticidade do ajuste automaticamente da quantidade de número de réplicas para a carga de trabalho atual [Sousa and Machado 2012]. Elasticidade é alcançada por meio de auto provisionamento, que adiciona novas réplicas se o sistema não consegue lidar com a carga de trabalho atual ou remove réplicas desnecessárias.

- Redimensionamento de recursos: O redimensionamento permite a ajuste da quantidade de recursos de acordo com a carga de trabalho. Por exemplo, pode-se incrementar ou decrementar a quantidade de CPU nas máquinas virtuais de forma automática, garantindo a qualidade e reduzindo os custos [Rego et al. 2011].

A elasticidade é vista de forma diferente pelo consumidor de serviços da nuvem e pelo provedor destes serviços. O usuário ou consumidor de serviços da nuvem não olha o interior das tecnologias da nuvem, mas apenas visualiza a sua interface. Ele interage com uma nuvem apenas pelo portal de acesso no qual solicita provisionamento e alocação dos recursos computacionais, e os detalhes técnicos ficam escondidos. Já o provedor precisa colocar em operação toda a tecnologia necessária para que a elasticidade aconteça.

Em nuvens privadas, a empresa precisa adquirir e implementar os recursos computacionais que serão alocados através do modelo da nuvem para os seus usuários. A elasticidade é sentida apenas no nivel dos usuários internos da nuvem privada, mas não no datacenter, que precisa ainda investir em capital, como servidores e sistemas operacionais. De qualquer forma esta infraestrutura dinâmica permitida pela nuvem privada é um grande benefício quando comparado ao modelo tradicional de gestão de recursos computacionais, como realizado hoje pelos datacenters. Uma gestão dinâmica de recursos diminui em muito a ociosidade média dos servidores (em torno de 85%) e acelera a velocidade com que estes recursos são provisionados para seus usuários [Taurion 2012].

5.5. Adaptação da Revisão Sistemática

Neste trabalho, uma adaptação de uma revisão sistemática é proposta como uma maneira simplificada de realizar estudos em uma determinada área de pesquisa, com a descrição de suas atividades e a dinâmica do funcionamento. Para o estudo de trabalhos relacionados e revisão bibliográfica, foi adotada uma sistemática de revisão. Essa sistemática consiste na criação de consultas em bibliotecas digitais nas quais é possível criar consultas mais detalhadas. Assim, foi elaborada uma revisão sistemática para relacionar os trabalhos que utilizaram elasticidade em ambientes de Computação em Nuvem, com foco em análise de desempenho.

A revisão sistemática é uma estratégia para a identificação, avaliação e interpretação das pesquisas relevantes disponíveis para uma questão de pesquisa ou algum interesse em particular [Kitchenham 2004]. Por meio da utilização de um processo controlado e formal de pesquisa bibliográfica, espera-se que os resultados retornem os tópicos mais pesquisados, lacunas, desafios, processos, ferramentas e técnicas. Para a execução da revisão foi utilizado como base o guia para revisão sistemática de [Kitchenham 2004] com algumas adaptações. Uma ferramenta que apóia o desenvolvimento de revisões sistemáticas pode ser encontrada em [LaPES 2013], onde é possível realizar o *download* da ferramenta StArt. A Figura 5.4 exibe essa adaptação utilizada neste trabalho.

5.5.1. Atividade 1: Planejar Revisão

Descreve as atividades necessárias para o planejamento da revisão.

- **Identificar Necessidade da Revisão**

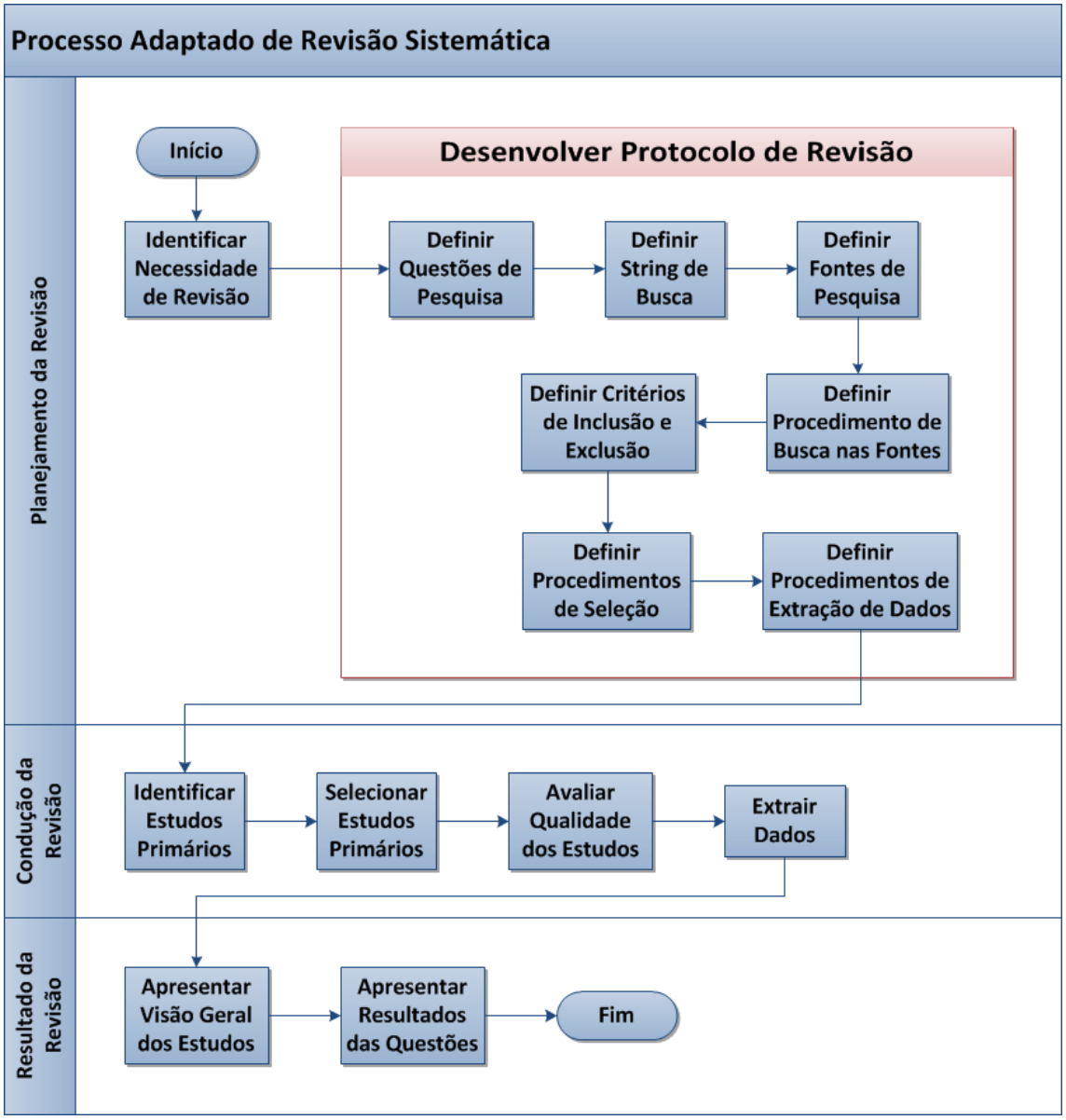

Figura 5.4. Fluxo do protocolo da revisão. Fonte: Adaptado de [Kitchenham 2004].

Elasticidade é comumente citada como uma das vantagens da Computação em Nuvem, porém não é fácil encontrar trabalhos que se aprofundem no assunto.

- **Desenvolver Protocolo da Revisão**

  O protocolo de revisão está descrito a seguir em sete subatividades:

  (i) Definir Questões de Pesquisa

  Após a leitura dos artigos selecionados, esta revisão sistemática teve como objetivo responder às seguintes questões:

- Questão Principal (QP) Qual o estado da arte sobre elasticidade em ambientes de

Computação em Nuvem?

- Questão Secundária 1 (QS1) Como está sendo realizado análise de desempenho para elasticidade em ambientes de Computação em Nuvem?
- Questão Secundária 2 (QS2) Quais ferramentas, *benchmarks* ou cargas de trabalho são utilizados para avaliar a elasticidade de ambientes de Computação em Nuvem?
- Questão Secundária 3 (QS3) Quais métricas são mais utilizadas para avaliar a elasticidade de ambientes de Computação em Nuvem?
- Questão Secundária 4 (QS4) Quais as tendências de pesquisa em Computação em Nuvem do ponto de vista de elasticidade?

(ii) Definir String de Busca

Em todas as consultas foram utilizadas as seguintes palavras chave: *"cloud computing"*, *elasticity*, *"performance analysis"*, *"performance evaluation"*, *metric*, *benchmark* e *tool*. Após alguns refinamentos, a seguinte *string* de busca foi gerada: ("*cloud computing*") AND ("*elasticity*") AND ("*performance analysis*" OR "*performance evaluation*") AND ("*metric*" OR "*benchmark*" OR "*tool*").

(iii) Definir Fontes de Pesquisa

Para esta revisão, os artigos foram pesquisados nos indexadores Science Direct [ScienceDirect 2012], ACM Digital Library [ACM 2012] e IEEExplorer [IEEE 2012].

(iv) Definir Procedimento de Busca nas Fontes

A mesma *string* de busca foi utilizada nos três indexadores por meio do mecanismo de busca avançada existente em cada um.

(v) Definir Critérios de Inclusão e Exclusão

Algumas restrições foram utilizadas para limitar a busca. Foram pesquisados trabalhos do tipo periódico ou conferência e os trabalhos deveriam ter sido publicados entre os anos de 2009 a 2012. A palavra chave "*elasticity*" deve constar no trabalho, já que todos os trabalhos estão em inglês. Novamente, os trabalhos selecionados e não selecionados são verificados para se garantir que nenhum trabalho tenha sido incluído ou excluído erroneamente.

(vi) Definir Procedimento de Seleção

Etapa 1: A estratégia de busca é aplicada nas fontes.

Etapa 2: Para selecionar um conjunto inicial de estudos, os títulos e resumos de todos os artigos obtidos foram lidos e confrontados com os critérios de inclusão e exclusão.

Etapa 3: Todos os artigos selecionados na etapa 2 foram lidos por completo e novamente confrontados com os critérios do item (v). Os artigos incluídos são documentados e encaminhados para extração dos dados.

(vii) Definir Procedimento de Extração dos Dados

A extração das informações dos artigos foi realizada com base em um formulário com perguntas direcionadas a obter respostas para as questões de pesquisa da revisão. O formulário consistia de uma planilha com os seguintes itens a serem preenchidos para cada trabalho lido: título, ano de publicação, veículo de publicação, autores, país, grupo, palavras chave, proposta, observações, tipo de análise, métricas, métricas para elasticidade, carga de trabalho, ferramentas, trabalhos futuros e reprodução.

5.5.2. Atividade 2: Condução da Revisão

A condução da revisão consistiu de quatro subatividades, descritas a seguir:

- **Identificar Estudos Primários**

A coleta de informações desta revisão sistemática ocorreu no mês de julho de 2012. A execução das demais atividades ocorreram no segundo semestre de 2012. Como resultado obtido, na Science Direct foram encontrados 24 trabalhos, na ACM Digital Library 32 e no o IEEExplorer 88 trabalhos, totalizando 144 trabalhos.

- **Selecionar Estudos Primários**

Para um refinamento dos resultados, como critério de inclusão os resumos dos 144 trabalhos foram lidos para que fosse confirmado se eles realmente estavam alinhados com o tema, e se de alguma forma o termo elasticidade era utilizado no trabalho, restando 48 trabalhos divididos em 15 do Science Direct, 14 da ACM Digital Library e 19 do IEEExplorer.

- **Avaliar da Qualidade dos Estudos**

A avaliação da qualidade dos artigos dos estudos primários foi feita de forma simplificada verificando apenas a presença ou não de alguma forma de aplicação do conceito elasticidade em Computação em Nuvem, ou métricas e ferramentas específicas para elasticidade.

- **Extrair Dados**

Uma vez definido o conjunto dos trabalhos selecionados para leitura completa, efetuou-se o processo de extração dos dados conforme planejado no item de procedimento de extração de dados. Para isto uma planilha para cada artigo foi preenchida com as informações definidas no item "Definir Procedimento de Extração dos Dados". Esta atividade durou aproximadamente quatro meses para ser concluída.

5.5.3. Atividade 3: Resultados da Revisão

Apresenta os resultados da revisão de maneira geral e para cada questão planejada.

- **Apresentar Visão Geral dos Estudos**

Após a leitura dos artigos selecionados, os resultados foram consolidados conforme o planejamento. Algumas informações foram identificadas antes da consolidação das questões pesquisadas. Estas informações permitiram obter uma visão geral sobre os estudos de elasticidade em Computação em Nuvem.

A Tabela 5.2 exibe os valores para cada ano, considerando tanto periódicos quanto em conferências. O ano de publicação considerado foi o que estava registrado no próprio trabalho como data de publicação. O período considerado foram os anos de 2009 a 2012. No momento da obtenção dos artigos, alguns trabalhos ainda estavam na situação de "*in press*". É provável que ao longo deste trabalho estes artigos já tenham sido publicados. Pode-se perceber que houve um acréscimo nas publicações no ano de 2011. Talvez esse aumento tenha ocorrido devido ao fato que muitos grupos de trabalho finalizaram suas pesquisas em Computação em Nuvem, mostrando os primeiros resultados. Em 2012 há apenas 11 trabalhos publicados, porém esse número deve aumentar em consequencia da publicação dos trabalhos *in press*, e considerando que os artigos foram selecionados em meados do ano de 2012, provavelmente haverão mais trabalhos no restante do ano.

Tabela 5.2. Quantidade de trabalhos publicados por ano de publicação

Ano	Quantidade
2009	2
2010	5
2011	23
2012	11
in press	7

A maioria dos trabalhos foram originados do periódico *Future Generation Computer Systems*[1], fator de impacto igual a 1.978 e Qualis A2 em Ciência da Computação, com 12 publicações. Estas publicações são recentes, tendo em vista que a maioria de seus trabalhos são de 2012 (4) e 7 deles estão com o estado de *in press*. Muitas conferências estão começando a publicar trabalhos em nuvem nos últimos anos e a tendência é que apareçam mais conferências específicas. Os veículos de publicação retornados pela pesquisa foram bem diversificados. A exceção do *Future Generation Computer Systems*, os demais veículos obtiveram uma ou duas publicações. A Tabela 5.3 exibe apenas os veículos com pelo menos duas publicações.

A contabilização dos países foi realizada da seguinte forma: para cada país de cada grupo de pesquisa em uma publicação, seria incrementada a quantidade de publicação do país. Portanto, alguns artigos contabilizaram mais de um país. A Figura 5.5 apresenta as publicações por país. Três países se destacaram nas publicações relacionadas à elasticidade em Computação em Nuvem: Austrália (9), Espanha (9) e Estados Unidos (10).

[1] http://www.journals.elsevier.com/future-generation-computer-systems/

Tabela 5.3. Quantidade de publicações por veículo de publicação

Ano	**Quantidade**	Qualis
Future Generation Computer Systems	12	A2
IEEE International Conference on Utility and Cloud Computing	2	-
IEEE International Parallel & Distributed Processing Symposium	2	A1
IEEE International Symposium on Service Oriented System Engineering	2	B2
International Conference on Cloud Computing	2	B2
International Conference on Parallel Processing	2	A2

Apesar dos Estados Unidos possuírem mais publicações, estas foram de diferentes grupos de pesquisas e associadas a empresas.

Ao todo foram identificados 71 grupos de pesquisas distribuídos entre 17 países. Os grupos que apresentaram mais publicações relacionadas à pesquisa foram Austrália (5) e Espanha (5), seguidos da China (3) e da Suécia (3), com destaque para os grupos *Cloud Computing and Distributed Systems (CLOUDS) Laboratory* [2] e *Dept. Arquitectura de Computadores y Automática* [3], que possuem muitas publicações devido a atuação em diversas áreas de pesquisa em Computação em Nuvem.

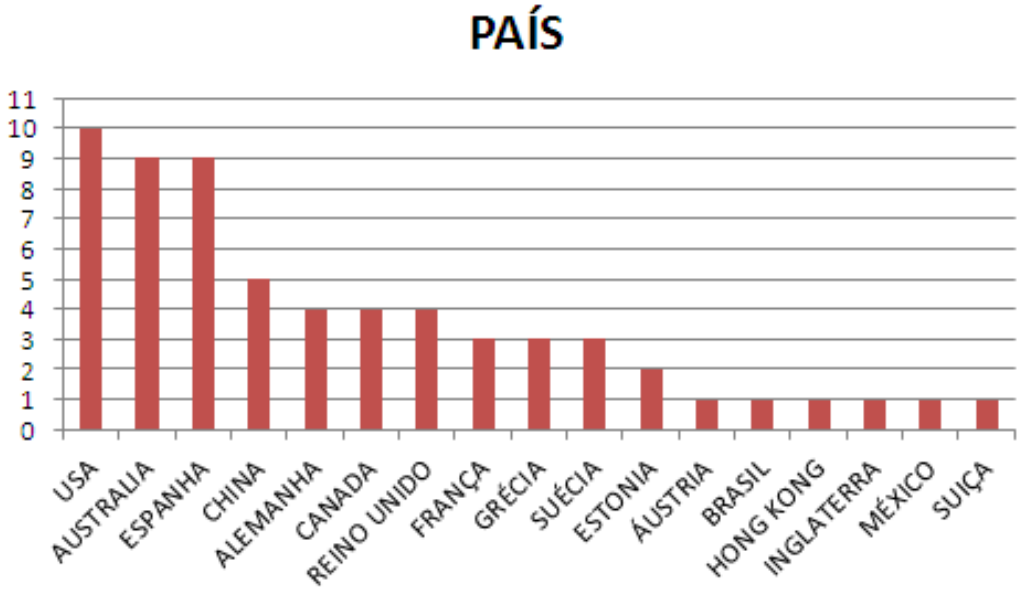

Figura 5.5. Gráfico de publicações por país.

A Figura 5.6 apresenta as publicações por autor. Coincidentemente os autores com maior quantidade de publicações são de grupos consolidados em pesquisa em Com-

[2]http://www.cloudbus.org
[3]http://dsa-research.org

putação em Nuvem. Estes autores são de grupos da Austrália e Espanha. Seus trabalhos não se restringem somente à elasticidade, mas atuam em diversas linhas de pesquisa em Computação em Nuvem. Foram identificados 166 autores diferentes.

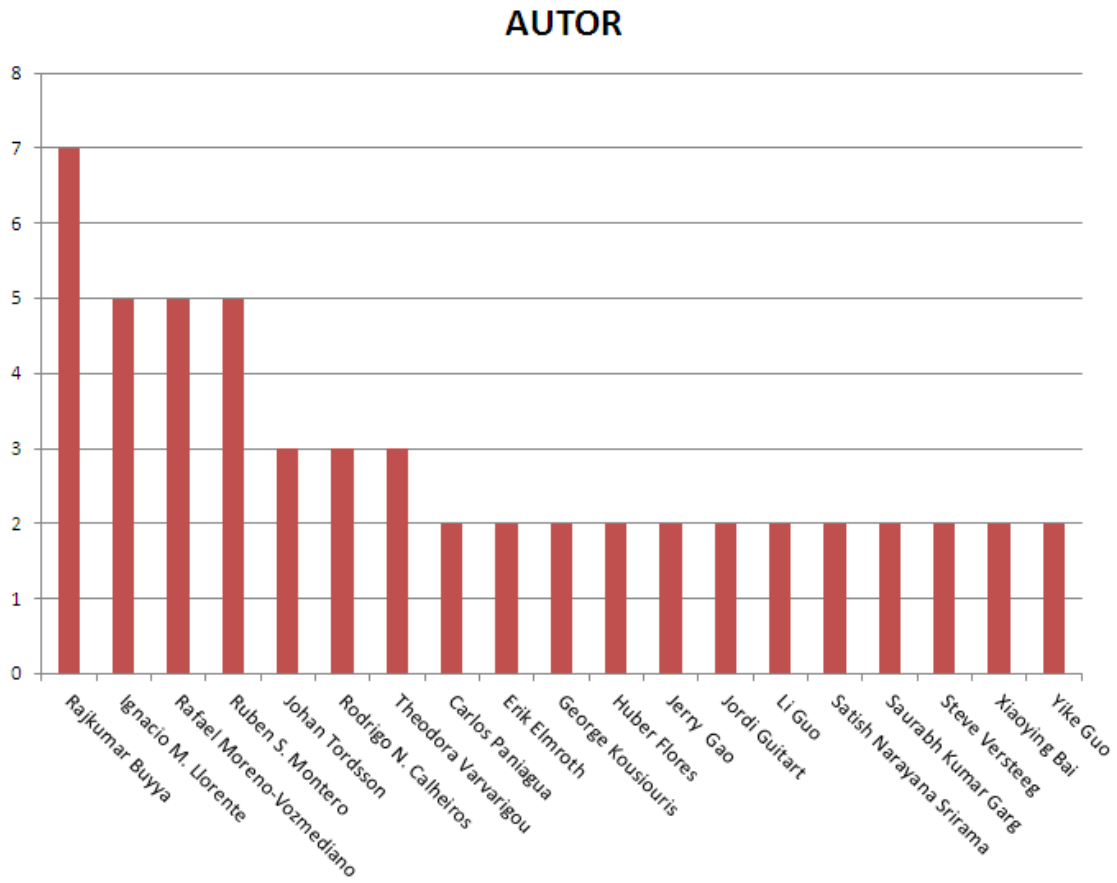

Figura 5.6. Gráfico de publicações por autor.

- **Apresentar Resultados das Questões**

O resultado e análise das questões de pesquisa estão descritos a seguir:

Questão Principal (QP): Qual o estado da arte sobre elasticidade em ambientes de Computação em Nuvem?

Em todos os trabalhos os termos elasticidade e análise de desempenho ocorreram. Alguns trabalhos desenvolveram experimentos especificamente para a elasticidade, enquanto que a maioria utilizou recursos de ambientes para a elasticidade ou propuseram soluções elásticas.

De maneira geral, a maioria dos trabalhos em Computação em Nuvem relacionados à elasticidade englobaram os seguintes temas:

- Arquiteturas para Computação em Nuvem

- Estratégias para alocação de recursos
- Ferramentas para proposíto geral
- Definição da tarifação de serviços na nuvem
- Aplicações de alto desempenho e web
- Ambientes diversificados (predominantemente Amazon)

De maneira mais específica, algumas categorias de áreas pesquisa puderam ser identificadas nos trabalhos:

- Elasticidade: alguns trabalhos desenvolveram soluções elásticas baseadas em replicação [Sousa and Machado 2012] ou redimencionamento dos recursos de forma automática [Rego et al. 2011]. Alguns Métricas para medir a elasticidade foram propostas por [Islam et al. 2012] [Li et al. 2011a] [Calheiros et al. 2012]. Para avaliar a elasticidade, os experimentos realizados utilizaram cargas de trabalho variáveis [Tirado et al. 2011].
- Alocação de recursos: estratégias e políticas para à alocação de recursos foram propostas, variando desde modelos matemáticos, análise dos recursos disponíveis, modelos multi-inquilinos e localização geográfica, dentre outros [Espadas et al. 2011], [Paniagua et al. 2011], [Bryant et al. 2011], [Li et al. 2011b], [Otto et al. 2012] e [Calheiros et al. 2011a]. Geralmente métricas estavam associadas às políticas, sejam para análise dos recursos, seja para tomada de decisão para a alocação.
- SLA: os trabalhos pesquisados utilizaram a elasticidade para garantir QoS. Alguns *frameworks* foram identificados para avaliação do SLA: YCSB [Cooper et al. 2010], OPTIMIS [Ferrer et al. 2012] e SMICloud [Garg et al. 2011] e [Garg et al. 2012]. Também foram associados a variações em cargas de trabalho distintas para analisar os efeitos sobre a manutenção do SLA.
- Tarifação: as pesquisas em tarifação de serviços na nuvem aparecem ainda de maneira muito variada, geralmente propostas de tarifação baseadas na utilização de recursos [Han et al. 2012]. Um ponto a ser considerado é a minimização dos custos por entre os modelos de serviços dos provedores de nuvem [He et al. 2010] e [Hong et al. 2012].
- *High Performance Computing* (HPC): a computação de alto desempenho surgiu em vários trabalhos, tanto do ponto de vista de simplesmente executar aplicações de HPC na nuvem quanto ao desenvolvimento de metodologias [Mauch et al. 2012], [Lu et al. 2010] e [Zhai et al. 2011]. [Raveendran et al. 2011] utiliza aplicações *Message Passing Interface* (MPI) elásticas para um *framework* de nuvem. O trabalho [Moreno-Vozmediano et al. 2011] utiliza o acesso ubíquo de recursos na nuvem para implantar um *cluster* no topo de uma infraestrututra de nuvem para resolver aplicações do tipo *Many-Task Computing* (MTC).

- Computação Autonômica: muitos trabalhos indiretamente utilizaram recursos ou princípios de computação autonômica [Pandey et al. 2012], [Etchevers et al. 2011], [Ghanbari et al. 2011] e [Niu et al. 2012], tentando reduzir a intervenção humana. A utilização de agentes para coleta de informações, principalmente métricas associadas a algum recurso computacional do ambiente, para posterior tomada de decisão baseada em regras foi um recursos muitas vezes utilizado para promover elasticidade.

- Computação Móvel: a nuvem pode ser utilizada para melhorar o desempenho de dispositivos móveis através do processamento e armazenamento de maneira elástica na nuvem [Niehorster et al. 2011]. A *Internet of Things* (IoT) deve aumentar a interação entre a nuvem e os diversos dispositivos móveis [Paniagua et al. 2011] e [Flores et al. 2011].

- Análise de Desempenho: avaliações de desempenho orientadas a cargas de trabalho em nuvens privadas e nuvens públicas foram alvos de algumas pesquisas em [Tudoran et al. 2012] e [Gao et al. 2011].

- Arquitetura: abordagens para segurança [Li et al. 2012] e para o gerenciamento de máquinas virtuais [Montero et al. 2011] foram propostas. Muitos trabalhos utilizaram aspectos de integração entre ambientes, como GRID e nuvens.

  Alguns *frameworks* e arquiteturas foram propostas para gerenciar a tratar a alocação de recursos utilizando balanceadores de carga e *brokers* [Tordsson et al. 2012]. Em [Dawoud et al. 2011] foi proposta uma arquitetura de escalonamento para máquinas virtuais elásticas. Em [Etchevers et al. 2011] foi utilizado um modelo arquitetural de aplicação e protocolo de auto configuração que automatiza a distribuição de aplicações distribuídas legadas. [Lucas-Simarro et al. 2012] propuseram uma arquitetura que trabalha diferentes estratégias de escalonamento através de várias nuvens baseada em um critério de otimização, restrições do usuário e do ambiente.

Questão Secundária 1 (QS1): Como está sendo realizado análise de desempenho para elasticidade em ambientes de Computação em Nuvem?

Conforme as definições de [Jain 1991] para os tipos de experimentos, foram identificados três tipos para os artigos da pesquisa: experimentação, modelagem analítica e simulação, sendo que estes trabalhos podem ocorrer combinados. A Tabela 5.4 descreve o quantitativo dos tipos de experimentos por publicação. Dos 48 artigos pesquisados, dois eram do tipo *survey* e não possuiam nenhum experimento. Estes trabalhos não estão inclusos na Tabela 5.4.

A maioria dos trabalhos faz algum tipo de experimentação, seja experimentação simples ou combinada com outra técnica. É comum que os experimentos em elasticidade em Computação em Nuvem ocorram na Amazon EC2. Diversos artigos fizeram os experimentos combinando tipos de experimentos. Isso implica em uma qualidade maior dos trabalhos, pois eles tornam-se mais completos e é possível a comparação entre trabalhos iguais mas com visões diferentes. Os trabalhos utilizaram o CloudSim [Calheiros et al. 2011b] ou utilizam *traces* de aplicações para realizar as simulações.

Apesar dos experimentos serem bem variados, existe deficiências no projeto dos experimentos. Os objetivos, serviços e resultados dos experimentos são informados. Contudo, o planejamento não é detalhado nas publicações. Em alguns trabalhos, a justificativa para a seleção das métricas, os parâmetros e ferramentas utilizadas na configuração dos experimentos não são informadas. Em relação aos resultados, estes são apenas dispostas em tabelas ou gráficos e comentários relacionados são superficiais.

O projeto da carga de trabalho utilizada nos experimentos está diretamente associado ao *benchmark* ou carga de trabalho utilizado, o que significa que se estes não possuírem algum mecanismo de configuração da carga, o experimento fica muito atrelado às ferramentas. Em geral os experimentos dos trabalhos identificados utilizaram o SPECweb2005 [Wee and Liu 2010] [Perez-Sorrosal et al. 2011] [Bryant et al. 2011] ou o TPC-W [Han et al. 2012] [Kossmann et al. 2010] [Islam et al. 2012], que permitem configurações das cargas de trabalho.

A forma como os experimentos foram conduzidos requer um projeto mais detalhado, pois está diretamente associado ao esforço e custo do experimento. A análise e interpretação dos dados em vários trabalhos foi bastante completa, porém em alguns casos foi meramente descritiva e geral, o que pode deixar conclusões em aberto. Todos esses fatores prejudicam a reprodução dos experimentos e consequente comparação entre trabalhos.

Por outro lado, a quantidade de 23 trabalhos com técnicas de avaliação combinadas proporciona uma qualidade maior dos trabalhos. Essa combinação de técnicas de avaliação permite que se possa ter uma visão teórica, muitas vezes do ponto de vista matemático, e prática dos experimentos.

Tabela 5.4. Tipos de experimentos identificados nas publicações

Tipo de Experimento	SD	ACM	IEEE	Total por Tipo de Experimento
Experimentação	6	9	4	19
Modelagem analítica	0	1	2	3
Simulação	0	0	1	1
Experimentação e Modelagem analítica	8	2	5	15
Simulação e Modelagem analítica	1	0	5	6
Experimentação e Simulação	0	1	1	2
Total por Indexador	15	13	18	46

Questão Secundária 2 (QS2): Quais ferramentas, *benchmarks* ou cargas de trabalho são utilizados para avaliar a elasticidade de ambientes de Computação em Nuvem?

Para facilitar a análise das ferramentas, estas foram divididas em grupos. Algumas ferramentas provêm um serviço, principalmente infraestrutura. A Tabela 5.5 exibe as ferramentas identificadas na pesquisa.

A maioria dos trabalhos utilizou o ambiente da Amazon, principalmente instâncias de máquinas virtuais (EC2), banco de dados (S3, MySQL, EBS, RDS, SimpleDB), escalonamento automático (AutoScaling) e o de monitoramento (CloudWatch). Contudo, apenas 2 trabalhos utilizaram o AutoScaling e nenhum trabalho utilizou o Elastic Load Balancing. Outros ambientes identificados foram o Microsost Azure e RackSpace. Ferramentas

Tabela 5.5. Principais ferramentas identificadas para a provisão de serviçoes na nuvem

Descrição	Quantidade
Amazon EC2	22
Amazon S3	6
OpenNebula	4
Azure	3
Eucalyptus	3
Amazon AutoScaling	2
Amazon CloudWatch	2
Aneka	2
AWS MySQL	2
ElasticHosts	2
Google APP Engine	2
RightScale	2
SciCloud	2
Amazon EBS	1
AWS RDS	1
AWS SimpleDB	1
AzureBlobs	1
Cloud Hosting Provider (CHP)	1
Cloud9	1
Cumulus	1
Emotive	1
Flexiscale Cloud	1
GoGrid	1
Grid5000	1
HPC Cluster	1
IBM Research Compute Cloud (RC2)	1
Nimbus	1
OCT	1
Plataforma iVic	1
RackSpace	1

livre tais como o OpenNebula também foram utilizadas para a criação de nuvens privadas e públicas em alguns experimentos, como [Mauch et al. 2012] [Montero et al. 2011] [Tordsson et al. 2012] [Moreno-Vozmediano et al. 2009].

O Amazon CloudWatch[4] oferece monitoramento de recursos em nuvem. Tanto desenvolvedores quanto administradores do sistema podem utilizá-lo para coletar e monitorar métricas, que podem ser utilizadas para reagir imediatamente à situações e manter serviços funcionando de acordo com o SLA. Assim é possível obter uma visibilidade geral da utilização de recursos e do desempenho dos serviços e aplicativos.

[4]Amazon CloudWatch - http://aws.amazon.com/pt/cloudwatch/

O Auto Scaling[5] é um serviço permite escalar a capacidade da instância para mais ou para menos de forma automática. Média de utilização de CPU, rede e utilização de disco são exemplos de métricas utilizadas para escalar um serviço. Assim, o número de instâncias utilizadas aumenta facilmente durante picos de demanda para manter o desempenho e diminui automaticamente durante a redução da demanda para minimizar custos. O Auto Scaling é especialmente útil para aplicativos que experimentam variabilidade de uso por hora, dia ou semana e é ativado pelo Amazon CloudWatch.

O Elastic Load Balancing[6] distribui automaticamente o tráfego de entrada dos aplicativos em várias instâncias do Amazon EC2. Ele permite que se atinja uma maior tolerância a falhas nos serviços, fornecendo uma capacidade de equilíbrio entre a carga de trabalho e a resposta ao tráfego de entrada dos aplicativos. Ele detecta instâncias com problemas de integridade dentro de um conjunto e redireciona automaticamente o tráfego para instâncias íntegras até que as instâncias com problemas sejam restauradas. Além disso é possível utilizar o Elastic Load Balancing entre zonas de disponibilidade variadas, melhorando a disponibilidade.

Trabalhos relacionados à análise de desempenho utilizaram *benchmarks* e cargas de trabalho para seus experimentos. Diversos *benchmarks* e cargas de trabalho foram identificados na revisão, citados na Tabela 5.6. Dentre estes destacam-se os *benchmarks* TPC-W [Han et al. 2012] [Kossmann et al. 2010] [Islam et al. 2012] e o SPECweb2005 [Wee and Liu 2010] [Perez-Sorrosal et al. 2011] [Bryant et al. 2011] [Fito et al. 2010]. O Embarrassingly Distributed (ED) o NAS Grid Benchmarks (NGB) estiveram entre as mais utilizadas na pesquisa, sendo que em alguns trabalhos ambos foram utilizados, como em [Moreno-Vozmediano et al. 2009] [Moreno-Vozmediano et al. 2011] [Montero et al. 2011] [Tordsson et al. 2012].

Algumas dessas ferramentas são utilizadas para testes de carga em aplicações e se demonstraram eficiente para os experimentos de cargas de trabalho para aplicações em ambientes de Computação em Nuvem: Httperf [Fito et al. 2010] [Bryant et al. 2011] [Lucas-Simarro et al. 2012], JMeter [Espadas et al. 2011] [Islam et al. 2012] e o Tsung [Flores et al. 2011].

Uma forma de analisar o comportamento de um ambiente é por meio da execução de aplicações. A Tabela 5.7 exibe as aplicações identificadas na pesquisa. Tais aplicações eram das mais variadas áreas: computação gráfica [Emeakaroha et al. 2012], redes sociais [Otto et al. 2012], aplicações web [Lucas-Simarro et al. 2012] e funções matemáticas [Gao et al. 2011]. Todas as aplicações apareceram apenas uma vez na pesquisa.

Em relação à utilização de *traces*, estes foram utilizados em diversos experimentos. Muitas vezes são extrações de dados de sistemas, de *logs* de aplicações, de forma que aplicações que as utilizem ou a sua utilização para experimentos seja uma projeção ou simulação da realidade de algum outro sistema ou aplicação em plena utilização real. A Tabela 5.8 descreve as fontes de *traces* identificados. Todos os *traces* informados apareceram apenas uma vez na pesquisa.

[5]Auto Scaling - http://aws.amazon.com/pt/autoscaling/

[6]Elastic Load Balancing - http://aws.amazon.com/pt/elasticloadbalancing/

Tabela 5.6. *benchmarks* e cargas de trabalho identificados

Descrição	Quantidade
SPECweb2005	6
TPC-W	6
Embarrassingly Distributed (ED)	4
NAS Grid Benchmarks (NGB)	4
Httperf	3
JMeter	2
Linpack	2
YCSB	2
Apache Server Benchmarking Tool	1
Blast	1
BT-IO	1
Cloud Testing ETHZ	1
Cloudstone	1
Faban	1
HadoopUnit	1
Intel MPI Benchmark	1
IOR	1
JProfiler	1
MalStone	1
MPI-BLAST	1
NAS MPB Benchmark	1
Numerical Aerodynamic Simulation (NGB) suite	1
Olio	1
Prefail	1
RUBIS	1
SIGAR	1
Skampi	1
SMICloud	1
SOASTA	1
SpecJAppServer	1
Terasort	1
TPC-H	1
TSUNG	1
WBOX	1
YETI	1

Considerando as ferramentas para a simulação de ambientes de Computação em Nuvem, destacam-se o CloudSim [Calheiros et al. 2011b] e o D-Cloud [Banzai et al. 2010]. Os *hypervisors* identificados foram o KVM, com apenas uma referência, e o XEN com três, apesar de que praticamente todos os trabalhos fizeram uso de algum *hypervisor*, não citando qual foi ao longo do texto. Aplicações de *proxy* e balanceadores de carga também foram utilizadas: HAProxy (3), Nginx (2) e Squid e Amoeba com apenas uma ocorrência.

Tabela 5.7. Aplicações identificadas

Descrição	Descrição
A-Brain [Tudoran et al. 2012]	Java Fast Fourier Transform (FFT) [Gao et al. 2011]
Aplicação WEB [Lucas-Simarro et al. 2012]	NAMD [Wong and Goscinski 2012]
Croudstag [Flores et al. 2011]	Plugin do Firefox [Calheiros et al. 2012]
Emulador em cliente próprio [Han et al. 2012]	POP [Zhai et al. 2011]
FFMPEG [Kousiouris et al. 2012]	POVRAY [Emeakaroha et al. 2012]
Google Analytics Service [Paniagua et al. 2011]	TWITTER [Otto et al. 2012]
GRAPES [Zhai et al. 2011]	VMD [Wong and Goscinski 2012]
Gromacs [Niehorster et al. 2011]	Servidor de Wikipedia [Kousiouris et al. 2012]
IMD plugin [Wong and Goscinski 2012]	Zompopo [Flores et al. 2011]
Informações via GPS para pontos próximos de e-learning [Kousiouris et al. 2012]	

Tabela 5.8. Principais *traces* utilizados

Descrição	Descrição
Amazon EC2 [Garg et al. 2011]	Rackspace [Garg et al. 2011]
Carga de simulação de virus [Wong and Goscinski 2012]	UC Berkeley [Hong et al. 2012]
Clarknet [Hong et al. 2012]	*Urdue University's College of Engineering website* [Hong et al. 2012]
Fifa 98 [Ghanbari et al. 2011]	Vod UUSee [Niu et al. 2012]
Grid Workloads Archive (AuverGrid) [Nie and Xu 2009]	*Wikimedia Web Traces* [Hong et al. 2012]
Grid Workloads Archive (LCG) [Nie and Xu 2009]	Wikipedia [Ferrer et al. 2012]
Lastfm [Tirado et al. 2011]	Windows Azure [Garg et al. 2011]
NASA [Hong et al. 2012]	

Alguns trabalhos fizeram uso de modelagem analítica e utilizaram *solvers* para a resolução de problemas matemáticos. A Tabela 5.9 contabiliza os *solvers* utilizados.

Alguns trabalhos fizeram uso de servidores web. A Tabela 5.10 identifica os servidores identificados. O Apache Web Server e o Apache Tomcat foram os mais utilizados, totalizando somente eles 13 ocorrências, enquanto que a soma dos demais totalizou apenas 4 ocorrências.

A Tabela 5.11 exibe linguagens de programação, componentes ou bibliotecas identificadas na pesquisa com a respectiva quantidade de trabalhos nas quais foram uti-

Tabela 5.9. Solvers utilizados

Descrição	Quantidade
AMPL [Li et al. 2011b] [Lucas-Simarro et al. 2012] [Tordsson et al. 2012]	3
CPLEX [Tordsson et al. 2012]	1
Gurobi [Li et al. 2011b]	1
Matlab [Kousiouris et al. 2012]	1
MINOS [Lucas-Simarro et al. 2012]	1
MKL Library [Mauch et al. 2012]	1
Octave [Kousiouris et al. 2012]	1
Optimj [Paniagua et al. 2011]	1

Tabela 5.10. Servidores web utilizados

Descrição	Quantidade
Apache Web Server	7
Apache Tomcat	6
Jonas	2
Glassfish	1
JBOSS	1

lizadas. Na maioria das vezes essas linguagens foram utilizadas para a construção de *microbenchmarks*, *scripts* de apoio à tarefas como coleta de informações e comunicação. Não necessariamente estão diretamente associadas à elasticidade, mas à funcionalidades do sistemas e apoio. Um exemplo é o Java, que foi a linguagem mais utilizada. Vários itens da lista são também relacionados com Java, mas foram contabilizados de maneira separada.

Tabela 5.11. Linguagens de programação identificadas

Descrição	Quantidade	Descrição	Quantidade
Java	7	C++	1
Android	2	Java Library Openforecast	1
Jets3t	2	Java script	1
Json	2	JNI	1
Libvirt	2	JVM	1
Python	2	Jvmti	1
Servlet	2	Mpich2	1
Shell script	2	Pnuts	1
Ac2dm	1	Scripts ruby	1
Bcel	1	TCL/TK	1
C	1	XML	1

Uma categoria considerada foi a de técnicas e padrões utilizados, conforme descrito na Tabela 5.12. Algumas são apenas padrões utilizados através de alguma ferramenta. Outras são técnicas matemáticas específicas. Destaca-se o *Message Passing Interface* (MPI), um padrão para comunicação de dados em computação paralela, e que

aparece em primeiro lugar nesta categoria. Sua aplicação vem aumentando em trabalhos com Computação em Nuvem, não apenas relacionado a elasticidade, mas na utilização de MPI em ambientes de nuvem. Alguns *benchmarks* para MPI foram identificados na pesquisa, como o Intel MPI Benchmark e o MPI-Blast, ambos em [Zhai et al. 2011].

Tabela 5.12. Padrões e técnicas identificados na pesquisa

Descrição	Quantidade
MPI	4
ADL	1
AHP	1
Autoregressive Moving Average with Exogenous Inputs Model (ARMAX)	1
Coeficiente de Jaccard	1
Conjugate Gradient (CG)	1
DPPC in water	1
Jacobi	1
Modelagem	1
Modelo ARIMA	1
Modelo GARCH	1
Multiple Criteria Decision Making (MCDM)	1
OVF	1
Quadratic Response Surface Model (QRSM)	1
Redes Neurais	1
Regressão Linear Multivariada	1
Sparse Periodic Auto-Regression (SPAR)	1
Weibull Distributions	1
WS-BPEL	1

Para o armazenamento dos dados normalmente servidores de banco de dados são utilizados, que estão citados na Tabela 5.13. Alguns *benchmarks* e geradores de carga de trabalho necessitam de um banco de dados para o povoamento ou registro de operações. Também em aplicações web geralmente havia um acesso a dados. A elasticidade em alguns casos era avaliada pela capacidade de processamento de consultas no banco de dados, geralmente submetida pelos *benchmarks* e geradores de carga de trabalho.

Tabela 5.13. Banco de dados utilizados

Descrição	Quantidade
MYSQL	7
Cassandra	2
Hadoop	2
Postgreesql	1
Hbase	1
JDBC	1
JoSQL	1

Outra possibilidade para cargas de trabalho e *benchmarks* são os *microbenchmarks*, que são *benchmarks* simplificados, geralmente programas escritos e executados para gerar determinada carga de trabalho, podendo coletar algumas métricas. Os *microbenchmarks* identificados estão citados na Tabela 5.14. Como estes são específicos para um experimento, não foram encontrados em mais de um trabalho diferente. Por isso, todos os *microbenchmarks* identificados tiveram apenas uma ocorrência.

Tabela 5.14. *Microbenchmarks* identificados

Descrição
Benchmarks sintéticos [Tudoran et al. 2012]
Microbenchmarks [Zhai et al. 2011]
Benchmarks específicos construídos [Li et al. 2012]
Bubble sort [Lucas-Simarro et al. 2012]
Testes construídos com RIB [Aisopos et al. 2011]
Testes de carga específica escritos [Niehorster et al. 2011]
Carga de trabalho para *Bag-of-Tasks* (BoT) e aplicações GRID [Calheiros et al. 2011a]
Carga de trabalho simulados [Li et al. 2012]
Carga de trabalho sintéticos baseados em Poisson [Nie and Xu 2009]

Questão Secundária 3 (QS3): Quais métricas são mais utilizadas para avaliar a elasticidade de ambientes de Computação em Nuvem?

As métricas que foram identificadas nos trabalhos resultantes da pesquisa foram agrupadas em temas. Essas métricas são encontradas em diversas áreas relacionadas à redes de computadores e sistemas distribuídos, e não se limitam a apenas às comentadas neste trabalho. As principais métricas são orientadas a recursos, tais como CPU e memória. O percentual de alocação da CPU foi a métrica mais frequente, mas métricas de uso de memória, disco, erros e violações de QoS também foram identificadas. Métricas de desempenho também foram identificadas, por exemplo, a vazão, expresa em requisições ou MB por segundo.

Conforme a classificação de [Jain 1991], as métricas mais citadas foram: tempo de resposta (latência, tempo de alocação/desalocação, tempo de acesso, tempo de ociosidade, tempo de resposta), *throughput* (requisições/s, MB/s), utilização (percentual de recursos, percentual de CPU), confiabilidade, disponibilidade (*downtime* e *uptime*) e aquisição de sistemas (custo e custo/desempenho). Escalabilidade está incluída na classificação de [Menasce et al. 2004]. Nesta categoria as métricas mais citadas foram as seguintes: demanda, SLA, *overhead*, capacidade total da infraestrutura, custo e energia.

Algumas métricas são compartilhadas entre grupos diferentes dependendo do contexto de utilização, por exemplo, tempo de alocação/desalocação de recursos pode estar relacionados a métrica de tempo de resposta e escalabilidade. Algumas técnicas estatísticas são muito utilizadas na exposição e interpretação dos resultados de experimentos. As mais encontradas foram média, desvio padrão, variância e erro.

O custo e a tarifação são aspectos que muitas vezes estão associados à nuvens públicas devido ao seu caráter de aquisição de recursos. A definição de uma forma de se precificar a utilização de recursos na nuvem ainda não é padronizada, e normalmente

cada provedor de serviços possui a sua própria estratégia. Custos totais, custos da infraestrutura, custos por hora são exemplos identificados. A tarifação dos serviços de um provedor muitas vezes está associada à utilização de recursos, como CPU, memória e armazenamento.

O consumo de energia é um os temas principais em ambientes de Computação em Nuvem. Nos trabalhos analisados foram identificadas as seguintes métricas para medir eficiência energética: *Data Center Infrastructure Efficiency* (DCiE), *Power Usage Efficiency* (PUE) e *Data Center Performance per Energy* (DPPE) [Garg et al. 2012] e [Tudoran et al. 2012]. DCiE é definido como o percentual da potência total do equipamento de TI (computadores, *storages* e rede). O PUE é inversamente proporcional ao DCiE. DPPE correlaciona o desempenho do datacenter com emissões de carbono. O *overhead* de energia foi analisado em um trabalho como uma maneira de se comparar experimentos, com a medição da potência e consumo de energia sendo realizada através de um dispositivo físico [Li et al. 2012].

Considerando a latência como a diferença entre o início de um evento e o início real do evento, onde seus efeitos se tornam perceptíveis, muitos eventos na nuvem utilizam esta métrica como uma maneira de se medir esse tempo. Nesta pesquisa foram identificadas latências relaciadas à migração de máquinas virtuais, atualização e leitura de dados. Aspectos de QoS foram identificados e quantificados, sendo relacionados. a um SLA ser mantido pela aplicação ou ambiente. Geralmente o QoS está associados a regras de monitoramento. Por fim, métricas associadas ao *overhead* e confiabilidade foram também identificadas.

Em relação à medidas específicas para elasticidade, a pesquisa retornou as seguintes métricas descritas na Tabela 5.15. Muitas dessas métricas são de recursos do sistema associadas à elasticidade, ou seja, para medir ou promover a elasticidade. Algumas são específicas para a elasticidade, como é o caso de métricas relacionadas à sobre-utilização e sub-utilização. O trabalho [Islam et al. 2012] definiu métricas de elasticidade baseadas em penalidades para a sobre-utilização e a sub-utilização dos recursos. [Gao et al. 2011] descreveram um conjunto de métricas (CRAM, CRUM, SPM, SLM, SCM, SEC, ESS e ESR) para analisar a escalabilidade e elasticidade em ambientes de SaaS. Estas métricas estão associadas à alocação de recursos, utilização dos recursos, desempenho do sistema, carga do sistema, capacidade efetiva do sistema, escalabilidade efetiva do sistema, faixa efetiva de escalabilidade.

O trabalho [Garg et al. 2012] descreveu uma métrica para elasticidade como o quanto um serviço de nuvem pode ser escalado durante intervalos de pico, que pode ser definido por dois atributos tempo médio para expandir ou contrair a capacidade do serviço e a capacidade máxima do serviço (número máximo de unidades computacionais que podem ser providas em períodos de pico. Em [Bai et al. 2011], a métrica *Performance Resource Ratio* (PRR), ou a Taxa de Desempenho de Recursos (TDR) reflete o relacionamento entre o desempenho e os resursos utilizados, baseada no tempo de espera, tempo de execução, e alocação dos recursos, como CPU, largura de banda e memória.

Questão Secundária 4 (QS4): Quais as tendências de pesquisa em Computação em Nuvem do ponto de vista de elasticidade?

Tabela 5.15. Métricas associadas à elasticidade

Descrição	Descrição
% de Violações	*Performance Resource Ratio* (PRR)
Capacidade de Computação	*Scale-up*
Capacidade de Serviço Máxima	SLA
Capacidade Total da Infraestrutura	Sobreutilização
Computing Resource Allocation Meter (CRAM)	Subutilização
Computing Resource Utilization Meter (CRUM)	*Svstem Capacity Meter* (SLM)
Custo de Implantação Total	*System Capacity Meter* (SCM)
Custo/taxa de Desempenho	*System Effective Capacity Meter* (SEC)
Demanda	*System Performance Meter* (SPM)
Effective Scalable Range (ESR)	Taxa de Sobre-provisionamento
Effective System Scalability (ESS)	Tempo de Alocação/Desalocação de Recursos
Elasticidade	Tempo Médio para Expandir ou Contrair a Capacidade do Serviço
Escalabilidade	Tempo x Recursos no Tempo
Fornecimento Disponivel	*Throughput* (Requisições/tempo)
Overhead de Reconfiguração da Infraestrutura	Variância de maneira geral
Preço Total da Infraestrutura	

De maneira geral, muitos trabalhos foram identificados como tendências de pesquisa em Computação em Nuvem relacionados à elasticidade, onde se destacam:

- *Benchmarks* para Computação em Nuvem: na literatura existem muitos *benchmarks* para a geração de diversos testes e cargas de trabalho. Porém, *benchmarks* específicos para ambientes de Computação em Nuvem, onde se possa avaliar a elasticidade de um ambiente para simular cargas variadas com aumento e diminuição da carga ainda não são comuns. Um exemplo de *benchmark* utilizado para medir a elasticidade em um ambiente de Computação em Nuvem é o *framework* Yahoo! Cloud Serving Benchmark (YCSB) [Cooper et al. 2010], que permite comparar desempenho para serviço de dados na nuvem.

- Métricas específicas para elasticidade: apesar de que alguns trabalhos citam métricas e modelos específicos para a medição da elasticidade ([Islam et al. 2012], [Garg et al. 2012] e [Gao et al. 2011]), ainda existem poucos trabalhos que realmente medem a elasticidade. É mais comum identificar trabalhos que utilizam ou provêm a elasticidade.

- Redes autonômicas em ambientes de nuvem: os princípios de redes autonômicas já são encontrados em trabalhos de elasticidade em nuvem, principalmente pela utilização de agentes para coleta de dados e definição de regras para a tomada de

decisão. Operações de auto escalonamento de recursos são baseados em regras. Contudo, este ambiente não possui padronização. A automação de alguns recursos nos provedores, de maneira que não seja necessária a intervenção humana e que se adéque às necessidades dos usuários é outro ponto a ser considerado.

- Testes em ambientes de Computação em Nuvem: testes com o objetivo de verificar se o ambiente de Computação em Nuvem está apto a suportar cargas de trabalho variadas, alterações de cargas em curtos intervalos de tempo, simulação de usuários. O quanto um ambiente é elástico, o quanto ele suporta a elasticidade dos recursos mantendo um nível de serviço adequado ao usuário. Em [Bai et al. 2011] diversos aspectos de testes em ambientes de Computação em nuvem são abordados.

- Geração de cargas de trabalho em nuvem: diversos geradores de carga de trabalho existem na literatura, assim como os *benchmarks* e muitas vezes ambos em conjunto.

- Tarifação baseado em elasticidade: a tarifação de serviços na nuvem é alvo de muita pesquisa, e muitos trabalhos propõem estratégias de precificação. A padronização ainda é uma área a ser pesquisada, pois os provedores são variados, os tipos de serviços e modelos adotados são diferentes.

- Computação móvel: com a facilidade de acesso através de dispositivos móveis faz-se necessário a integração entre os dispositivos e a nuvem, responsável pelo processamento. A elasticidade está no lado servidor das aplicações, ou seja, nos datacenters, onde mecanismos que provêm a elasticidade estão em execução.

- Aplicações orientadas ao consumo de energia: com os datacenters investindo em mecanismos de redução do consumo energético, a Computação em Nuvem pode colaborar com a migração de aplicações, que atendendo às necessidades dos clientes através da elasticidade, pode evitar o desperdício de uma infraestrutura ociosa. A "computação verde" pode ser utilizada junto a Computação em Nuvem. O fato da elasticidade estar em execução em provedores de Computação em Nuvem pode promover uma economia de energia. Estratégias de alocação de recursos baseadas no consumo energético podem regular e consolidar servidores, desativando os que não estão em utilização.

Um conjunto de trabalhos futuros foram identificados de acordo com esta revisão. Muitos desses trabalhos podem servir como subsídio para futuras pesquisas, aprofundamento de tecnologias ou práticas didáticas. As referências associadas aos itens a seguir são de trabalhos que podem ser utilizados como base para pesquisa, e não necessariamente realizaram a pesquisa em questão. Alguns trabalhos, inclusive, estão repetidos pois a partir dele pode-se derivar diferentes trabalhos futuros.

- Elasticidade: os principais trabalhos futuros estão relacionados a alocação de recursos, políticas e medições. Não especificamente derivado dos trabalhos analisados, pesquisas em elasticidade para escalonamento dinâmico, medições com modelos conhecidos, regras para definir o comportamento do ambiente, redes autonômicas e controles heurísticos podem ser considerados.

  - Desenvolver uma arquitetura de coordenador e troca de recursos entre provedores [Calheiros et al. 2012]
  - Utilização de teoria das filas e balanceadores de carga em servidores web [Han et al. 2012]
  - Implementar um controle para a elasticidade e para balanceadores de carga [Vaquero et al. 2011]
  - Medir elasticidade de um ambiente de nuvem, medir o custo associadas as penalidades ou não cumprimento do SLA; testes com *benchmarks* de caracteristicas diferentes, e utilizar o AutoScaling da Amazon [Islam et al. 2012]
  - Aplicações que integrem GRID e nuvens [Nie and Xu 2009]
  - Definir funções de satisfação de SLA e aplicar a um modelo e arquitetura. Aplicações com regras aplicadas a um *proxy* para redirecionar servidores baseado em limites de regras [Fito et al. 2010]
  - Implementar técnicas de replicação elásticas para lidar com mudanças na carga de trabalho [Özsu and Valduriez 2011].
  - Gerenciar de forma automática os recursos disponíveis e a carga de trabalho do sistema para garantir a QoS e melhorar a utilização destes recursos [Sousa et al. 2011].

- Alocação e provisionamento de recursos: diversas técnicas foram propostas, outras foram utilizadas como estratégias de alocação. Contudo, este aspecto é bastante amplo e muitas oportunidades de pesquisa ainda estão em aberto.

  - Problema da mochila [Aisopos et al. 2011]
  - Balanceamento de carga [Espadas et al. 2011]
  - Implementar um *cloud broker* e modelos matemáticos para alocação em zonas distintas da Amazon [Lucas-Simarro et al. 2012]
  - Aplicação móvel que utiliza recursos da nuvem e modelo matemático de predição de cargas [Paniagua et al. 2011]
  - Aplicações com o SPECweb2005 e balanceamento de carga [Wee and Liu 2010]
  - Testes com elasticidade envolvendo migração e clonagem de máquinas virtuais [Bryant et al. 2011]
  - Modelo de alocação em múltiplas nuvens [Li et al. 2011b]
  - Predição de cargas de trabalho [Niu et al. 2012]
  - Definir funções de satisfação de SLA, aplicar a um modelo e a uma arquitetura, gerar aplicações com regras aplicadas a um *proxy* para redirecionamento de servidores baseado em limites de regras [Fito et al. 2010]
  - Modelos de tarifação de elasticidade e modelos de filas para alocação de máquinas virtuais [Otto et al. 2012]
  - Modelar filas para alocação de recursos na nuvem, criar políticas de alocação e realizar experimentos de simulação de alocação com o CloudSim com grandes quantidades de máquinas virtuais [Calheiros et al. 2011a]

- Análise de desempenho: dois trabalhos focam em realizar a análise de desempenho comparando dois ambientes.
  - Instalação e configuração do Nimbus e Azure, e utilizar métricas de custo para a análise [Tudoran et al. 2012]
  - Utilizar gráfico de radar para avaliar ambientes de nuvem em aplicações que utilizem o CloudWatch e AutoScaling da Amazon [Gao et al. 2011]
- Arquiteturas: diferentes abordagem para o gerenciamento de algum aspecto do ambiente foram propostas.
  - Alterações em *hypervisors* [Li et al. 2012]
  - *Cloud broker* em zonas distintas da Amazon [Lucas-Simarro et al. 2012]
  - Desenvolver uma arquitetura com um *cloud broker* [Tordsson et al. 2012]
  - Aplicações em banco de dados na nuvem em provedores diferentes e aplicações com o PaaS [Kossmann et al. 2010]
  - Aplicações com o SPECweb2005 e balanceadores de carga [Wee and Liu 2010]
- Métricas e SLA: tanto para a coleta de dados quanto para avaliar algum aspecto do ambiente.
  - Coletar métricas a nível SaaS e converter para IaaS [Emeakaroha et al. 2012]
  - Utilizar funções utilitárias e calcular as métricas [Garg et al. 2012]
  - Aplicar elementos do *Key Performance Indicators* (KPI) para análise da nuvem [Garg et al. 2011]
  - Analisar o *overhead* da virtualização e da nuvem
  - Utilizar gráfico de radar para avaliar ambientes de nuvem em aplicações que utilizem o CloudWatch e AutoScaling da Amazon [Gao et al. 2011]
  - Utilizar *High Throughput Computing* (HTC) no OpenNebula e Amazon e analisar o *overhead* da nuvem e virtualização [Montero et al. 2011]
- Computação Científica: trabalhos que utilizam aplicações de computação científica no ambiente de nuvem.
  - Aplicações do tipo *Many-Task Computing* (MTC) com *benchmarks* e aplicações [Moreno-Vozmediano et al. 2011]
  - Desenvolver aplicações HPC na nuvem [Mauch et al. 2012]
  - Aplicações MPI na nuvem [Zhai et al. 2011] [Raveendran et al. 2011]
  - Utilizar aplicações do tipo *High Throughput Computing* (HTC) no OpenNebula e Amazon [Montero et al. 2011]
- Redes Autonômicas: a utilização de princípios de redes autonômicas, como *loops* de controle, regras e tomada de decisão.

  - Criar agentes para IaaS e criar um mecanismo no SaaS que utilize os agentes do IaaS [Niehorster et al. 2011]
  - Criar aplicações para controle teórico e heurístico da escalabilidade com utilização de regras de controle para métricas [Ghanbari et al. 2011]
  - Predição de cargas de trabalho [Niu et al. 2012]

- Diversificados: trabalhos futuros isolados, que não se encaixaram nas categorias anteriores.
  - Avaliar e testar ambientes de nuvem com as métricas de escalabilidade. Definir pontos de testes em um ambiente de nuvem. Simular ambientes com o CloudSim. Definir maneiras de se testar elasticidade [Bai et al. 2011]
  - Aplicação móvel que acessa uma nuvem e utiliza um balanceador de carga [Flores et al. 2011]
  - Criação de modelos de custo para tarifação para cobrança pela utilização dos recursos [Hong et al. 2012]
  - Aplicações no Azure com o BLAST [Lu et al. 2010]
  - Aplicações com redes sociais [Tirado et al. 2011]
  - Workflows na nuvem [Pandey et al. 2012]
  - Aplicações ou *benchmarks* em nuvens privadas, publicas e híbridas, com balanceador de carga [Moreno-Vozmediano et al. 2009]

Além das categorias acima, um grupo identificado refere-se aos trabalhos que desenvolveram ou utilizaram *frameworks*. Como extensão a partir destes trabalhos, análise de desempenho, evolução e aplicações em contextos diferentes podem sem utilizados. Os *frameworks* foram LoM2His [Emeakaroha et al. 2012], OPTIMIS [Ferrer et al. 2012], MCM [Flores et al. 2011], YCSB [Cooper et al. 2010], e SMICloud [Garg et al. 2011] [Garg et al. 2012].

Provedores de acesso possuem diversos aspectos que podem influenciar no desempenho do serviço utilizado, por exemplo o tamanho das instâncias utilizadas ou pela localização geográfica. Pesquisas em aspectos de localização geográfica, provedores diferentes, com hardware heterogêneo, nuvens públicas, privadas e híbridas podem ser consideradas.

5.6. Estratégias para Análise de Desempenho com Elasticidade em Computação em Nuvem

Algumas estratégias serão discutidas a seguir para o provimento de soluções elaásticas, baseadas em algumas soluções dos trabalhos analisados. Destaca-se que as soluções não se limitam às citadas, sendo apenas estratégias que devem ser detalhadas e adaptadas a cada caso.

A computação autonômica é inspirada em sistemas biológicos para lidar com desafios de complexidade, dinamismo e heterogeneidade [Kephart and Chess 2003], características presentes nos ambientes de Computação em Nuvem e, assim, fornece uma abordagem promissora neste contexto. Embora a Computação em Nuvem apresente certas

características autonômicas, como o provisionamento automático de recursos, seu objetivo é reduzir o custo dos recursos ao invés de reduzir a complexidade do sistema [Zhang et al. 2010]. A utilização de regras para a tomada de decição surge como uma solução para que um ambiente se torne elástico em relação à utilização dos recursos. No ambiente de nuvem, as políticas, regras ou requisitos são orientados ao negócio. Entretanto não existe uma padronização destas regras. O grau de automação, abstração e customização das políticas e regras que regem um serviço podem variar bastante. Alguns sistemas oferecem aos usuários a possibilidade de construção de condições simples com base em certas métricas, por exemplo, CPU, memória, disco e rede, enquanto outros utilizam métricas no nível do serviço, por exemplo, relação custo/benefício e permitem estratégias mais complexas [Vaquero et al. 2011].

A utilização de um mecanismo de *brokering* em um provedor de serviços na nuvem é uma opção de estratégia. Eles fornecem mecanismos de escalonamento necessários para otimizar a alocação de máquinas virtuais entre várias nuvens e oferecem uma interface uniforme de gerenciamento para operações de, por exemplo, implantação, monitoramento e finalização de máquinas virtuais, independente da tecnologia em particular do provedor de nuvem [Tordsson et al. 2012]. Nesse mesmo trabalho, um mecanismo de utilização de um *brokering* é descrita. Esses mecanismos de escalonamento necessários para otimizar a de seleção de recursos virtuais podem ser independentes ou pertencer a um serviço multi-componente, entre diferentes nuvens. Ele deve considerar exigências como a configuração de recursos individuais, desempenho do serviço agregado e o custo total. Além disso, o usuário pode especificar restrições sobre balanceamento de carga e configuração de serviço, a fim de evitar que um conjunto de recursos sejam alocados na mesma nuvem ou em nuvens diferentes. O escalonador encontra uma alocação de recursos virtuais entre os diferentes provedores de nuvem que otimiza os critérios do utilizador e atende às restrições de alocação. Essa arquitetura está representada na Figura 5.7.

Alguns trabalhos utilizaram *proxies* para realizar o balanceamento de carga nos ambientes de Computação em Nuvem. Com a distribuição da carga de trabalho de entrada, é possível através de alguma política de balanceamento que seja distribuído entre instâncias diferentes, promovendo uam solução elástica. Na pesquisa foram identificados aplicações na maioria dos casos com o HAProxy[7] e o NGINX[8]. Soluções com balanceadores de carga podem ser associadas à soluções autonômicas e assim prover uma solução elástica mais eficiente.

Em nuvens com instâncias de capacidades diferentes é possível se promover a elasticidade de diversas formas. Em abordagens reativas, é possível se gerar um mecanismo que colete informações do ambiente (agente), e se defina regras onde seja possível definir limites a serem atendidos para garantir o SLA. No caso de ultrapassar esses limites, alguma ação ocorre no ambiente, podendo ser a geração de novas instâncias, a adição de novas instâncias em um balanceador de carga, o aumento ou redução da capacidade das instâncias atuais, a migração de instâncias de um servidor para outro (consolidação), dentre outras opções.

Para a utilização de estratégias elásticas em ambientes públicos comerciais, uma

[7]HAProxy - http://haproxy.1wt.eu/
[8]NGINX - http://nginx.org/

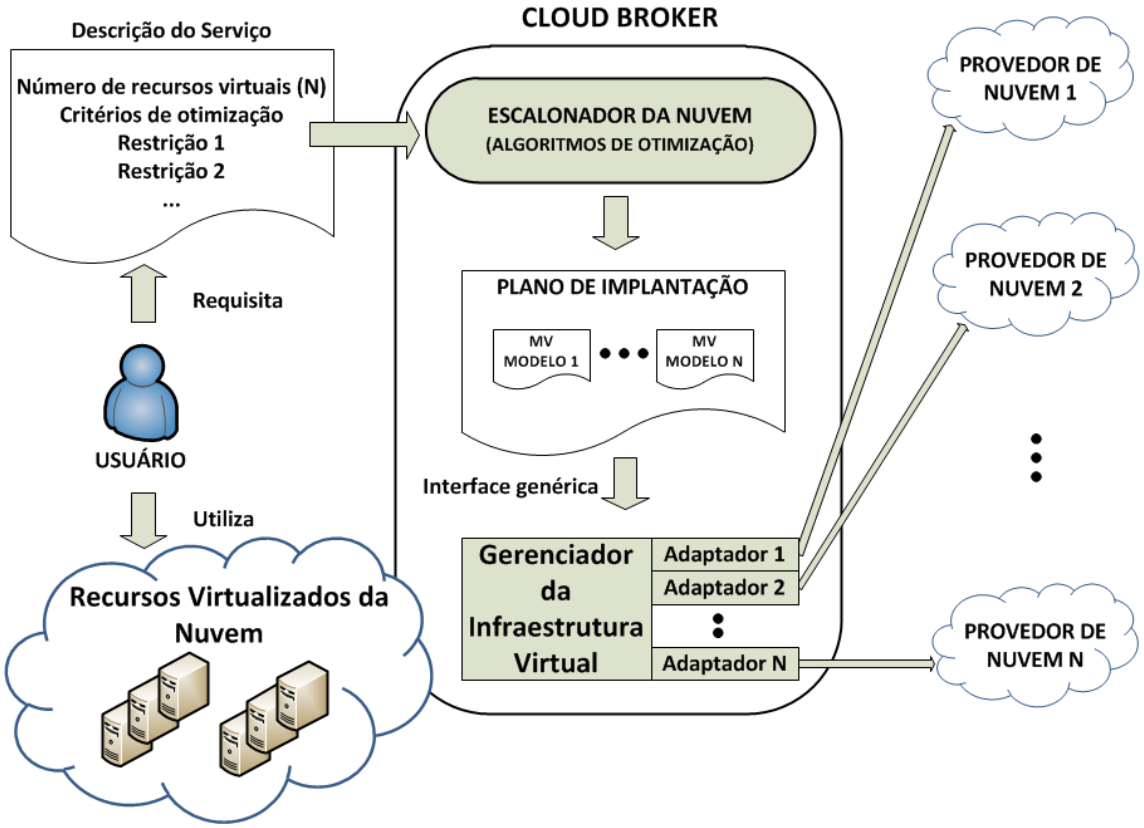

Figura 5.7. Arquitetura com um *cloud broker* (baseado em [Tordsson et al. 2012]).

estratégia interessante é utilizar os serviços da Amazon, além das instâncias. Estes serviços são o Auto Scaling, Amazon CloudWatch e o Elastic Load Balancing. Muitos trabalhos utilizaram o ambiente da Amazon para seus experimentos, mas podem ser replicados em outros provedores que possuam os mesmos mecanismos. Este provedor possui diversos mecanismos que podem ser bastante úteis na elaboração de soluções elásticas. Por ser um ambiente comercial, a utilização dos serviços é paga. Alguns desses serviços possuem uma cota de utilização livre, que após ultrapassada, passa-se a ser tarifada.

Combinando esses recursos, é possível a construção de soluções elásticas. Por exemplo, utilizando o Auto Scaling e o Elastic Load Balancing pode-se tratar as seguintes situações: para se ter certeza de que o número de instâncias saudáveis sob o Elastic Load Balancer nunca seja menor do que duas, quando o Auto Scaling detectar essa condição, automaticamente acrescentará o número requisitado de instâncias ao seu grupo; para ter certeza de acrescentar instâncias do Amazon EC2 quando a latência de alguma das instâncias exceder em 4 segundos durante um período de 15 minutos, basta gerar essa condição e o Auto Scaling executará as medidas apropriadas nas instâncias.

5.7. Conclusão

Elasticidade é uma das principais características da Computação em Nuvem, consistindo na capacidade de adicionar ou remover recursos, sem interrupções e em tempo de execução para lidar com a variação da carga. É difícil compreender os requisitos de elasticidade

de uma aplicação específica, de sua carga de trabalho e principalmente como um provedor deve gerenciar os recursos para atender esses requisitos. Alguns estudos sobre as características de elasticidade na nuvem são encontrados na literatura.

Este trabalho sugeriu diferentes maneiras de se trabalhar análise de desempenho em elasticidade em ambientes de Computação em Nuvem, sob a forma de uma adaptação de uma revisão sistemática. Não foi intenção desta revisão afirmar que o estado da arte no tema é apenas o que foi descrito, mesmo até por causa da limitação de indexadores de artigos e do período utilizado para a coleta de artigos. Também não se espera de forma alguma ter esgotado a lista de trabalhos que respondem às questões de pesquisa, mas atendeu-se ao objetivo de comentar sobre o estado da arte sobre elasticidade em Computação em Nuvem.

Foi apresentada uma metodologia para revisão sistemática simplificada onde foi realizada todo o planejamento e em seguida a consolidação dos resultados. Os benefícios da revisão foram enormes. Uma visão geral da elasticidade em Computação em Nuvem foi vista, com aspectos gerais, e algumas questões mais específicas foram respondidas. Aspectos relacionados à formas de realizar análise de desempenho, ferramentas, *benchmarks*, cargas de trabalho, métricas e tendências de pesquisa em elasticidade em Computação em Nuvem, com aspectos de análise de desempenho, puderam ser listados.

Como trabalho futuro existe sempre a atualização da revisão com novos artigos posteriores à coleta. Além disso, a adição de novos indexadores de busca. Como a revisão é reproduzível, e há o registro dos dados, então não há perda de informação, podendo esta ser sempre atualizada e gerar resultados mais atuais.

Referências

[ACM 2012] ACM (2012). Acm digital library. http://dl.acm.org/. Online; acessado em fevereiro-2012.

[Aisopos et al. 2011] Aisopos, F., Tserpes, K., and Varvarigou, T. (2011). Resource management in software as a service using the knapsack problem model. *International Journal of Production Economics*, (0):–.

[Azure 2012] Azure (2012). *Microsoft Azure*. http://www.microsoft.com/azure/.

[Bai et al. 2011] Bai, X., Li, M., Chen, B., Tsai, W.-T., and Gao, J. (2011). Cloud testing tools. In *Service Oriented System Engineering (SOSE), 2011 IEEE 6th International Symposium on*, pages 1 –12.

[Banzai et al. 2010] Banzai, T., Koizumi, H., Kanbayashi, R., Imada, T., Hanawa, T., and Sato, M. (2010). D-cloud: Design of a software testing environment for reliable distributed systems using cloud computing technology. In *Cluster, Cloud and Grid Computing (CCGrid), 2010 10th IEEE/ACM International Conference on*, pages 631 –636.

[Brantner et al. 2008] Brantner, M., Florescu, D., Graf, D., Kossmann, D., and Kraska, T. (2008). Building a database on s3. In *Proceedings of the 2008 ACM SIGMOD*

international conference on Management of data - SIGMOD '08, page 251, New York. ACM Press.

[Bryant et al. 2011] Bryant, R., Tumanov, A., Irzak, O., Scannell, A., Joshi, K., Hiltunen, M., Lagar-Cavilla, A., and de Lara, E. (2011). Kaleidoscope: cloud micro-elasticity via vm state coloring. In *Proceedings of the sixth conference on Computer systems*, EuroSys '11, pages 273–286, New York, NY, USA. ACM.

[Buyya et al. 2009] Buyya, R., Yeo, C. S., Venugopal, S., Broberg, J., and Brandic, I. (2009). Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Gener. Comput. Syst.*, 25(6):599–616.

[Calheiros et al. 2011a] Calheiros, R., Ranjan, R., and Buyya, R. (2011a). Virtual machine provisioning based on analytical performance and qos in cloud computing environments. In *Parallel Processing (ICPP), 2011 International Conference on*, pages 295 –304.

[Calheiros et al. 2012] Calheiros, R. N., Toosi, A. N., Vecchiola, C., and Buyya, R. (2012). A coordinator for scaling elastic applications across multiple clouds. *Future Generation Computer Systems*, 28(8):1350 – 1362. Including Special sections SS: Trusting Software Behavior and SS: Economics of Computing Services.

[Calheiros et al. 2011b] Calheiros, R. N., Vecchiola, C., Karunamoorthy, D., and Buyya, R. (2011b). The aneka platform and qos-driven resource provisioning for elastic applications on hybrid clouds. *Future Generation Computer Systems*, (0):–.

[Ciurana 2009] Ciurana, E. (2009). *Developing with Google App Engine*. Apress, Berkely, CA, USA.

[Cooper et al. 2010] Cooper, B. F., Silberstein, A., Tam, E., Ramakrishnan, R., and Sears, R. (2010). Benchmarking cloud serving systems with ycsb. In *Proceedings of the 1st ACM symposium on Cloud computing*, SoCC '10, pages 143–154, New York, NY, USA. ACM.

[Costa et al. 2011] Costa, R., Brasileiro, F., Lemos, G., and Mariz, D. (2011). Sobre a amplitude da elasticidade dos provedores atuais de computação na nuvem. In *SBRC 2011*.

[Dawoud et al. 2011] Dawoud, W., Takouna, I., and Meinel, C. (2011). Elastic vm for rapid and optimum virtualized resources' allocation. In *Systems and Virtualization Management (SVM), 2011 5th International DMTF Academic Alliance Workshop on*, pages 1 –4.

[Emeakaroha et al. 2012] Emeakaroha, V. C., Netto, M. A., Calheiros, R. N., Brandic, I., Buyya, R., and Rose, C. A. D. (2012). Towards autonomic detection of sla violations in cloud infrastructures. *Future Generation Computer Systems*, 28(7):1017 – 1029. Special section: Quality of Service in Grid and Cloud Computing</ce:title>.

[Espadas et al. 2011] Espadas, J., Molina, A., Jiménez, G., Molina, M., Ramírez, R., and Concha, D. (2011). A tenant-based resource allocation model for scaling software-as-a-service applications over cloud computing infrastructures. *Future Generation Computer Systems*, (0):–.

[Etchevers et al. 2011] Etchevers, X., Coupaye, T., Boyer, F., de Palma, N., and Salaun, G. (2011). Automated configuration of legacy applications in the cloud. In *Utility and Cloud Computing (UCC), 2011 Fourth IEEE International Conference on*, pages 170 –177.

[Ferrer et al. 2012] Ferrer, A. J., Hernández, F., Tordsson, J., Elmroth, E., Ali-Eldin, A., Zsigri, C., Sirvent, R., Guitart, J., Badia, R. M., Djemame, K., Ziegler, W., Dimitrakos, T., Nair, S. K., Kousiouris, G., Konstanteli, K., Varvarigou, T., Hudzia, B., Kipp, A., Wesner, S., Corrales, M., Forgó, N., Sharif, T., and Sheridan, C. (2012). Optimis: A holistic approach to cloud service provisioning. *Future Generation Computer Systems*, 28(1):66 – 77.

[Fito et al. 2010] Fito, J., Goiri, I., and Guitart, J. (2010). Sla-driven elastic cloud hosting provider. In *Parallel, Distributed and Network-Based Processing (PDP), 2010 18th Euromicro International Conference on*, pages 111 –118.

[Flores et al. 2011] Flores, H., Srirama, S. N., and Paniagua, C. (2011). A generic middleware framework for handling process intensive hybrid cloud services from mobiles. In *Proceedings of the 9th International Conference on Advances in Mobile Computing and Multimedia*, MoMM '11, pages 87–94, New York, NY, USA. ACM.

[Galante and de Bona 2012] Galante, G. and de Bona, L. C. E. (2012). A survey on cloud computing elasticity. In *Proceedings of the 5th IEEE/ACM International Conference on Utility and Cloud Computing (UCC '12)*.

[Gao et al. 2011] Gao, J., Pattabhiraman, P., Bai, X., and Tsai, W. T. (2011). Saas performance and scalability evaluation in clouds. In *Service Oriented System Engineering (SOSE), 2011 IEEE 6th International Symposium on*, pages 61 –71.

[Garg et al. 2011] Garg, S., Versteeg, S., and Buyya, R. (2011). Smicloud: A framework for comparing and ranking cloud services. In *Utility and Cloud Computing (UCC), 2011 Fourth IEEE International Conference on*, pages 210 –218.

[Garg et al. 2012] Garg, S. K., Versteeg, S., and Buyya, R. (2012). A framework for ranking of cloud computing services. *Future Generation Computer Systems*, (0):–.

[Geotecnologias 2012] Geotecnologias, S. . (2012). Cloud computing and gis - sadeck - geotecnologias. http://geotecnologias.wordpress.com/2011/06/28/cloud-computing-and-gis/. Online; acessado em agosto-2012.

[Ghanbari et al. 2011] Ghanbari, H., Simmons, B., Litoiu, M., and Iszlai, G. (2011). Exploring alternative approaches to implement an elasticity policy. In *Cloud Computing (CLOUD), 2011 IEEE International Conference on*, pages 716 –723.

[Han et al. 2012] Han, R., Ghanem, M. M., Guo, L., Guo, Y., and Osmond, M. (2012). Enabling cost-aware and adaptive elasticity of multi-tier cloud applications. *Future Generation Computer Systems*, (0):–.

[He et al. 2010] He, Q., Zhou, S., Kobler, B., Duffy, D., and McGlynn, T. (2010). Case study for running hpc applications in public clouds. In *Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing*, HPDC '10, pages 395–401, New York, NY, USA. ACM.

[Hong et al. 2012] Hong, Y.-J., Xue, J., and Thottethodi, M. (2012). Selective commitment and selective margin: Techniques to minimize cost in an iaas cloud. In *Performance Analysis of Systems and Software (ISPASS), 2012 IEEE International Symposium on*, pages 99 –109.

[IEEE 2012] IEEE (2012). Ieee xplore - home. http://ieeexplore.ieee.org. Online; acessado em fevereiro-2012.

[Islam et al. 2012] Islam, S., Lee, K., Fekete, A., and Liu, A. (2012). How a consumer can measure elasticity for cloud platforms. In *Proceedings of the third joint WOSP/SIPEW international conference on Performance Engineering*, ICPE '12, pages 85–96, New York, NY, USA. ACM.

[Jain 1991] Jain, R. (1991). *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling*. John Wiley and Sons, INC, 1st edition.

[Kephart and Chess 2003] Kephart, J. O. and Chess, D. M. (2003). The vision of autonomic computing. *Computer*, 36(1):41–50.

[Kitchenham 2004] Kitchenham, B. (2004). Procedures for performing systematic reviews. Technical report, Keele University and NICTA.

[Kossmann et al. 2010] Kossmann, D., Kraska, T., and Loesing, S. (2010). An evaluation of alternative architectures for transaction processing in the cloud. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, SIGMOD '10, pages 579–590, New York, NY, USA. ACM.

[Kousiouris et al. 2012] Kousiouris, G., Menychtas, A., Kyriazis, D., Gogouvitis, S., and Varvarigou, T. (2012). Dynamic, behavioral-based estimation of resource provisioning based on high-level application terms in cloud platforms. *Future Generation Computer Systems*, (0):–.

[LaPES 2013] LaPES (2013). Start. http://lapes.dc.ufscar.br/tools/start-tool/. Online; acessado em Janeiro-2013.

[Li et al. 2012] Li, J., Li, B., Wo, T., Hu, C., Huai, J., Liu, L., and Lam, K. (2012). Cyberguarder: A virtualization security assurance architecture for green cloud computing. *Future Generation Computer Systems*, 28(2):379 – 390.

[Li et al. 2011a] Li, M., Ye, F., Kim, M., Chen, H., and Lei, H. (2011a). A scalable and elastic publish/subscribe service. In *Parallel Distributed Processing Symposium (IPDPS), 2011 IEEE International*, pages 1254 –1265.

[Li et al. 2011b] Li, W., Tordsson, J., and Elmroth, E. (2011b). Modeling for dynamic cloud scheduling via migration of virtual machines. In *Cloud Computing Technology and Science (CloudCom), 2011 IEEE Third International Conference on*, pages 163 –171.

[Liu et al. 2007] Liu, S., Liang, Y., and Brooks, M. (2007). Eucalyptus: a web service-enabled e-infrastructure. In *CASCON '07: Proceedings of the 2007 conference of the center for advanced studies on Collaborative research*, pages 1–11, New York, NY, USA. ACM.

[Lu et al. 2010] Lu, W., Jackson, J., and Barga, R. (2010). Azureblast: a case study of developing science applications on the cloud. In *Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing*, HPDC '10, pages 413–420, New York, NY, USA. ACM.

[Lucas-Simarro et al. 2012] Lucas-Simarro, J. L., Moreno-Vozmediano, R., Montero, R. S., and Llorente, I. M. (2012). Scheduling strategies for optimal service deployment across multiple clouds. *Future Generation Computer Systems*, (0):–.

[Mafra and Travassos 2006] Mafra, S. N. and Travassos, G. H. (2006). Estudos primários e secundários apoiando a busca por evidência em engenharia de software. Technical report, Programa de Engenharia de Sistemas e Computação - COPPE/UFRJ.

[Mauch et al. 2012] Mauch, V., Kunze, M., and Hillenbrand, M. (2012). High performance cloud computing. *Future Generation Computer Systems*, (0):–.

[Mell and Grance 2009] Mell, P. and Grance, T. (2009). The nist definition of cloud computing. *National Institute of Standards and Technology*, 53(6):50.

[Menasce et al. 2004] Menasce, D. A., Dowdy, L. W., and Almeida, V. A. F. (2004). *Performance by Design: Computer Capacity Planning By Example*. Prentice Hall PTR, Upper Saddle River, NJ, USA.

[Montero et al. 2011] Montero, R. S., Moreno-Vozmediano, R., and Llorente, I. M. (2011). An elasticity model for high throughput computing clusters. *Journal of Parallel and Distributed Computing*, 71(6):750 – 757. Special Issue on Cloud Computing.

[Montgomery 2009] Montgomery, D. C. (2009). *Design and analysis of experiments*. John Wiley and Sons, INC, 7th edition.

[Moreno-Vozmediano et al. 2011] Moreno-Vozmediano, R., Montero, R., and Llorente, I. (2011). Multicloud deployment of computing clusters for loosely coupled mtc applications. *Parallel and Distributed Systems, IEEE Transactions on*, 22(6):924 –930.

[Moreno-Vozmediano et al. 2009] Moreno-Vozmediano, R., Montero, R. S., and Llorente, I. M. (2009). Elastic management of cluster-based services in the cloud. In *Proceedings of the 1st workshop on Automated control for datacenters and clouds*, ACDC '09, pages 19–24, New York, NY, USA. ACM.

[Nie and Xu 2009] Nie, L. and Xu, Z. (2009). An adaptive scheduling mechanism for elastic grid computing. In *Semantics, Knowledge and Grid, 2009. SKG 2009. Fifth International Conference on*, pages 184 –191.

[Niehorster et al. 2011] Niehorster, O., Krieger, A., Simon, J., and Brinkmann, A. (2011). Autonomic resource management with support vector machines. In *Proceedings of the 2011 IEEE/ACM 12th International Conference on Grid Computing*, GRID '11, pages 157–164, Washington, DC, USA. IEEE Computer Society.

[Niu et al. 2012] Niu, D., Xu, H., Li, B., and Zhao, S. (2012). Quality-assured cloud bandwidth auto-scaling for video-on-demand applications. In *INFOCOM, 2012 Proceedings IEEE*, pages 460 –468.

[Otto et al. 2012] Otto, J., Stanojevic, R., and Laoutaris, N. (2012). Temporal rate limiting: Cloud elasticity at a flat fee. In *Computer Communications Workshops (INFOCOM WKSHPS), 2012 IEEE Conference on*, pages 151 –156.

[Özsu and Valduriez 2011] Özsu, M. T. and Valduriez, P. (2011). *Principles of Distributed Database Systems, 3rd Edition*. Springer.

[Pandey et al. 2012] Pandey, S., Voorsluys, W., Niu, S., Khandoker, A., and Buyya, R. (2012). An autonomic cloud environment for hosting ecg data analysis services. *Future Generation Computer Systems*, 28(1):147 – 154.

[Paniagua et al. 2011] Paniagua, C., Srirama, S. N., and Flores, H. (2011). Bakabs: managing load of cloud-based web applications from mobiles. In *Proceedings of the 13th International Conference on Information Integration and Web-based Applications and Services*, iiWAS '11, pages 485–490, New York, NY, USA. ACM.

[Perez-Sorrosal et al. 2011] Perez-Sorrosal, F., Patiño Martinez, M., Jimenez-Peris, R., and Kemme, B. (2011). Elastic si-cache: consistent and scalable caching in multi-tier architectures. *The VLDB Journal*, 20(6):841–865.

[Raveendran et al. 2011] Raveendran, A., Bicer, T., and Agrawal, G. (2011). A framework for elastic execution of existing mpi programs. In *Parallel and Distributed Processing Workshops and Phd Forum (IPDPSW), 2011 IEEE International Symposium on*, pages 940 –947.

[Rego et al. 2011] Rego, P. A. L., Coutinho, E. F., Gomes, D. G., and de Souza, J. N. (2011). Architecture for allocation of virtual machines using processing features. In *1st International Workshop on Cloud Computing and Scientific Applications (CCSA)*.

[Robinson 2008] Robinson, D. (2008). *Amazon Web Services Made Simple: Learn how Amazon EC2, S3, SimpleDB and SQS Web Services enables you to reach business goals faster*. Emereo Pty Ltd, London, UK, UK.

[Sá et al. 2011] Sá, T. T., Soares, J. M., and Gomes, D. G. (2011). Cloudreports: Uma ferramenta gráfica para a simulação de ambientes computacionais em nuvem baseada no framework cloudsim. In *IX Workshop em Clouds e Aplicações - WCGA*.

[Schad et al. 2010] Schad, J., Dittrich, J., and Quiané-Ruiz, J.-A. (2010). Runtime measurements in the cloud: Observing, analyzing, and reducing variance. *PVLDB*, 3(1):460–471.

[ScienceDirect 2012] ScienceDirect (2012). Sciencedirect.com | search through over 10 million science, health, medical journal full text articles and books. http://www.sciencedirect.com/. Online; acessado em fevereiro-2012.

[Sharma et al. 2011] Sharma, U., Shenoy, P. J., Sahu, S., and Shaikh, A. (2011). A cost-aware elasticity provisioning system for the cloud. In *International Conference on Distributed Computing Systems (ICDCS)*, pages 559–570.

[Soror et al. 2010] Soror, A. A., Minhas, U. F., Aboulnaga, A., Salem, K., Kokosielis, P., and Kamath, S. (2010). Automatic virtual machine configuration for database workloads. *ACM Trans. Database Syst.*, 35(1):1–47.

[Sousa and Machado 2012] Sousa, F. R. C. and Machado, J. C. (2012). Towards elastic multi-tenant database replication with quality of service. In *Proceedings of the 5th IEEE/ACM International Conference on Utility and Cloud Computing (UCC '12)*, pages 168–175.

[Sousa et al. 2010] Sousa, F. R. C., Moreira, L. O., Macêdo, J. A. F., and Machado, J. C. (2010). Gerenciamento de dados em nuvem: Conceitos, sistemas e desafios. In *SBBD*, pages 101–130.

[Sousa et al. 2011] Sousa, F. R. C., Moreira, L. O., and Machado, J. C. (2011). Computação em nuvem autônoma: Oportunidades e desafios. In *Proceedings of the WoSiDA, collocated with SBRC 2011*.

[Suleiman et al. 2012] Suleiman, B., Sakr, S., Jeffery, D. R., and Liu, A. (2012). On understanding the economics and elasticity challenges of deploying business applications on public cloud infrastructure. *J. Internet Services and Applications (JISA)*, 3(2):173–193.

[Taurion 2012] Taurion, C. (2012). O que é elasticidade em cloud computing? (software, open source, soa, innovation, open standards, trends). http://goo.gl/qidBs. Online; acessado em agosto-2012.

[Tirado et al. 2011] Tirado, J., Higuero, D., Isaila, F., and Carretero, J. (2011). Predictive data grouping and placement for cloud-based elastic server infrastructures. In *Cluster, Cloud and Grid Computing (CCGrid), 2011 11th IEEE/ACM International Symposium on*, pages 285 –294.

[Tordsson et al. 2012] Tordsson, J., Montero, R. S., Moreno-Vozmediano, R., and Llorente, I. M. (2012). Cloud brokering mechanisms for optimized placement of virtual

machines across multiple providers. *Future Generation Computer Systems*, 28(2):358 – 367.

[Tudoran et al. 2012] Tudoran, R., Costan, A., Antoniu, G., and Bougé, L. (2012). A performance evaluation of azure and nimbus clouds for scientific applications. In *Proceedings of the 2nd International Workshop on Cloud Computing Platforms*, CloudCP '12, pages 4:1–4:6, New York, NY, USA. ACM.

[Vaquero et al. 2011] Vaquero, L. M., Rodero-Merino, L., and Buyya, R. (2011). Dynamically scaling applications in the cloud. *SIGCOMM Comput. Commun. Rev.*, 41(1):45–52.

[Vaquero et al. 2009] Vaquero, L. M., Rodero-Merino, L., Caceres, J., and Lindner, M. (2009). A break in the clouds: towards a cloud definition. *SIGCOMM Comput. Commun. Rev.*, 39(1):50–55.

[Verdi et al. 2010] Verdi, F. L., Rothenberg, C. E., Pasquini, R., and Magalhães, M. F. (2010). *Novas Arquiteturas de Data Center para Cloud Computing*, in Minicursos do XXVIII Simpósio Brasileiro de Redes de Computadores - SBRC2010, pages 103–152. SBC, Gramado, RS.

[Wee and Liu 2010] Wee, S. and Liu, H. (2010). Client-side load balancer using cloud. In *Proceedings of the 2010 ACM Symposium on Applied Computing*, SAC '10, pages 399–405, New York, NY, USA. ACM.

[Wong and Goscinski 2012] Wong, A. K. and Goscinski, A. M. (2012). A vmd plugin for namd simulations on amazon ec2. *Procedia Computer Science*, 9(0):136 – 145. Proceedings of the International Conference on Computational Science, ICCS 2012.

[Zhai et al. 2011] Zhai, Y., Liu, M., Zhai, J., Ma, X., and Chen, W. (2011). Cloud versus in-house cluster: evaluating amazon cluster compute instances for running mpi applications. In *State of the Practice Reports*, SC '11, pages 11:1–11:10, New York, NY, USA. ACM.

[Zhang et al. 2010] Zhang, Q., Cheng, L., and Boutaba, R. (2010). Cloud computing: state-of-the-art and research challenges. *Journal of Internet Services and Applications*, 1:7–18. 10.1007/s13174-010-0007-6.