

UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA
CURSO DE MESTRADO EM CIÊNCIA DA COMPUTAÇÃO

PROPOSTA DE DISSERTAÇÃO DE MESTRADO

**PROVISÃO DINÂMICA DE RECURSOS BASEADA EM
MÉTRICAS DE NEGÓCIO PARA APLICAÇÕES *SaaS***

MESTRANDO

RICARDO ARAÚJO SANTOS

ORIENTADORA

RAQUEL VIGOLVINO LOPES

CAMPINA GRANDE
FEVEREIRO - 2011

Lista de Siglas e Abreviaturas

CRM Customer Relationship Management

IAAS Infrastructure as a Service

PAAS Platform as a Service

SAAS Software as a Service

SLA Service Level Agreement

SOA Service Oriented Architecture

TI Tecnologia da Informação

Lista de Figuras

1	Interação entre clientes e provedores	8
---	-------------------------------------------------	---

Lista de Tabelas

1	Atividades Planejadas	12
2	Cronograma	16

Sumário

1	Introdução	5
1.1	Características de aplicações SaaS	5
1.2	Vantagens da metodologia <i>SaaS</i>	7
1.3	Provisão de Recursos	8
2	Objetivo da Proposta	11
3	Relevância da Proposta	11
4	Metodologia de Trabalho	12
4.1	Cronograma	16

1 Introdução

Nos últimos anos, a computação na nuvem tem modificado vários setores da indústria e academia e, com isso, a ideia de oferecer computação como um serviço utilitário, como energia elétrica, vem se tornando uma realidade. Dessa forma serviços como processamento, armazenamento, infraestrutura ou mesmo o acesso a um software através da internet têm se tornado mais comum tanto para o departamento de Tecnologia da Informação (TI) das empresas como para os usuários finais.

Os serviços oferecidos como computação utilitária abrangem vários níveis, porém não há um consenso na classificação deles. A classificação em [VRMCL08] é a mais comum na prática e divide os serviços em: (i) Infraestrutura como Serviço (do inglês *Infrastructure as a Service* - *IAAS*), na qual estão serviços como processamento e armazenamento, tendo como exemplos mais conhecidos o Amazon Elastic Compute Cloud [ama11b] e GoGrid [gog11]; (ii) Plataforma como Serviço (do inglês *Platform as a Service* - *PAAS*), englobando ferramentas para auxiliar desde o desenvolvimento de aplicações até a manutenção do ciclo de vida delas, representada aqui pelo Google App Engine [goo11]; e (iii) Software como Serviço (do inglês *Software as a Service* - *SAAS*), ofertando aplicações prontas para o usuário final, tendo como exemplos mais conhecidos aplicações para *Customer Relationship Management* (*CRM*) e editores de texto.

1.1 Características de aplicações SaaS

A ideia de prover acesso a aplicações por meio de *SaaS* não é nova. Brereton e Budgen [BB00] argumentaram em 1999 a favor do desenvolvimento de software em componentes independentes que juntos entregariam um serviço de mais alto nível ao usuário. Já a ideia de separar os papéis de dono e usuário do software é defendida por Turner et al. [TBB03]. Essas duas características ajudam a formar um conceito mais geral de *SaaS* no qual serviços são oferecidos possivelmente por diferentes donos, agregados formando um novo serviço e oferecido ao usuário final que, por sua vez, utiliza o software por meio de uma assinatura e não é responsável por manter o serviço.

Além disso, algumas outras características marcantes de *SaaS* foram identificadas em [Nat07] como:

Cobrança em um grão pequeno: As aplicações *on-premise*¹ possuem um sistema de licença única no momento da aquisição da aplicação. No caso das aplicações *SaaS*, licenças temporárias são adquiridas com base na quantidade de usuários aos quais se quer prover acesso e a cobrança é feita de acordo com a quantidade usada do serviço (processamento utilizado, quantidade de dados armazenados ou transferidos, por exemplo). Dessa forma, tanto o usuário da aplicação acaba por economizar já que a cobrança é mais justa com relação ao tempo de uso da aplicação como o provedor da aplicação lucra provendo uma mesma aplicação para diferentes usuários.

Fácil composição de aplicações: É muito comum utilizar resultados de um serviço como entrada de outros serviços de forma a produzir um serviço mais especializado agregando valor a um ou mais outros. Isso tem sido possível atualmente com a utilização de tecnologias de Arquitetura Orientada a Serviços (do inglês *Service Oriented Architecture - SOA*). Um exemplo pode ser uma aplicação que oferece serviço de edição de texto com correção automática provida por um serviço responsável somente por correção de texto.

Multi-localização: Que significa prover acesso a uma mesma aplicação para diferentes usuários de diferentes empresas, impedindo que um acesse informações do outro. Nesse caso, as aplicações podem ser genéricas o suficiente para permitir uma adaptação e customização do serviço oferecido a diferentes clientes. É necessário também uma gerência mais específica dos recursos utilizados para executar as diferentes instâncias do serviço impedindo que o alto processamento solicitado numa instância possa resultar num atraso no tempo de resposta de outra instância.

Alta escalabilidade: Durante o desenvolvimento da aplicação, há um esforço por parte dos desenvolvedores para que a aplicação funcione eficientemente ao atender diferentes demandas de usuários.

Facilidade de disponibilizar a aplicação na infraestrutura da empresa: Como o serviço é acessado através da internet, mudar a infraestrutura onde a aplicação está hospede-

¹*on-premise* significa executar a aplicação in loco, na infraestrutura local da empresa, em oposição a *off-premise*, que significa executar numa infraestrutura remota. Aplicações *SaaS* são comumente chamadas de *off-premise applications*.

dada não deve ser uma tarefa difícil. Dessa forma, uma empresa pode utilizar recursos próprios disponíveis para executar a aplicação e em momentos de sobrecarga de requisições balancear a carga executando novas instâncias da aplicação em infraestruturas remotas.

Internacionalização: Disponibilizar a aplicação para usuários acessando a partir de diferentes regiões geográficas. Isso pode requerer uma adaptação do serviço ao idioma nativo do cliente e sistemas de medida e moedas diferentes, como também a personalização do próprio servido. Como exemplo disso, pode-se pensar num sistema de busca por restaurantes que oferece resultados diferentes dependendo da região geográfica a partir da qual a requisição foi originada. Essa preocupação surge ainda durante o design da aplicação.

1.2 Vantagens da metodologia SaaS

Muitas são as vantagens desse modelo de entrega de serviço, tanto para o usuário final como para quem provê o serviço. Do lado de quem usa, as pequenas e médias empresas têm seus custos reduzidos com o uso do modelo *pay-as-you-go* pois não há mais a necessidade de adquirir e manter a aplicação muito menos infraestrutura para sua execução [LT08]. Grandes empresas, por sua vez, tiram proveito do fato de não precisar desenvolver uma nova aplicação. Além disso, há também a possibilidade de executar na própria infraestrutura da empresa reduzindo preocupações com segurança de dados críticos.

Na academia, iniciativas como Globus Online [glo11] e R-PHP [r-p11] mostram a viabilidade na utilização desse modelo para prover serviços que auxiliem os pesquisadores durante o ciclo da pesquisa. Serviços como edição de texto, processadores de dados, classificação de artigos são de interesse de muitos pesquisadores e o uso de ferramentas que não necessitem de instalação reduz bastante o trabalho adicional e permitem um maior foco no próprio processo da pesquisa.

Quanto aos provedores, a mesma aplicação pode ser provida a vários usuários de empresas diferentes. Dessa forma, custos com manutenção da aplicação e infraestrutura são compensados com a economia de escala já que a mesma infraestrutura pode ser utilizada para manter o mesmo serviço para milhares de clientes independentes. No entanto vários

outros problemas surgem com o emprego desse modelo: (i) preocupações com a modelagem, design, construção e manutenção de aplicações SaaS, tem sido um alvo de trabalhos como [MMLP09, ECM08]; (ii) segurança privacidade de dados críticos de empresas; (iii) como prover internacionalização; (iv) como fazer balanceamento de carga entre servidores; entre muitos outros.

1.3 Provisão de Recursos

Um problema interessante e que merece uma atenção especial por impactar diretamente nos lucros do provedor diz respeito a como o provedor executa a aplicação. A figura 1 mostra como os diferentes papéis interagem nesse processo. Clientes usam os serviços disponibilizados pelo provedor de *SaaS* que, por sua vez, adquire recursos de um ou mais provedores de *PaaS* ou diretamente de provedores de *IaaS*. Tanto a forma com a qual esses recursos são adquiridos como a forma como são utilizados impactam diretamente no custo de manter os serviços e, consequentemente, no lucro do provedor.

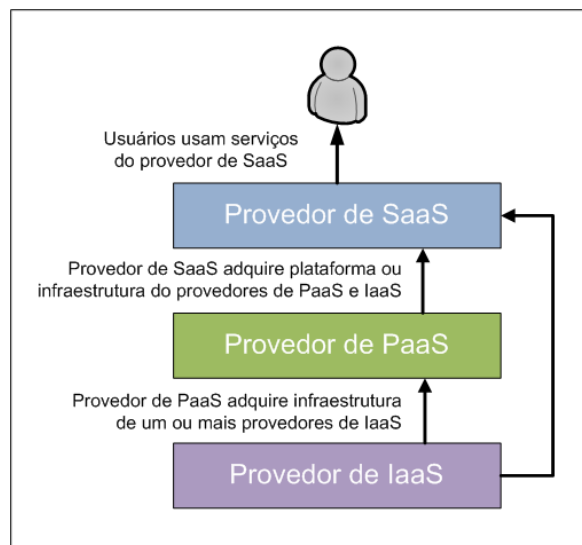


Figura 1: Interação entre clientes e provedores

Atualmente, o provedor de *IaaS* mais conhecido é o Amazon EC2 [ama11b]. Diversos tipos de máquinas virtuais são oferecidos a preços diferentes. O usuário paga pelo tempo que usa, pela quantidade de dados transferida e armazenada nas instâncias. Existem três formas de aquisição de máquinas virtuais [ama11a]:

Mercado sob demanda: As máquinas são compradas sem a necessidade de um contrato ou reserva. Dessa forma o usuário pode facilmente aumentar e diminuir a capacidade da aplicação. Em momentos de sobrecarga, pode ser que a Amazon não tenha máquinas para oferecer e, portanto, o pedido de aquisição de instâncias sob demanda ser negado;

Mercado de reserva: Nessa modalidade as máquinas são reservadas pelo intervalo de 1 a 3 anos a um preço mais baixo que o do mercado sob demanda. Dessa forma, quando o usuário precisar usar uma máquina reservada, a quantia paga pela hora utilizada é inferior ao do mercado sob demanda. Existe aqui a garantia de que as máquinas reservadas vão estar disponíveis quando solicitadas para uso.

Mercado baseado em lances de oferta: Essa modalidade de mercado é adequada para quem precisa de máquinas num grão menor que o do mercado sob demanda. O usuário especifica um preço limite para a máquina-hora. Como o preço da instância flutua baseado na oferta e demanda por máquinas desse tipo, quando o preço da instância está abaixo do limite especificado pelo usuário, a máquina é vendida. Se o preço da instância ultrapassar o limite do usuário, a máquina é encerrada.

Dessa forma um provedor pode, por exemplo, planejar um ano de utilização de recursos reservando instâncias no mercado de reserva, atender picos de requisições com instâncias sob demanda e reduzir custos comprando máquinas no mercado baseado em lances. Nesse caso, problemas interessantes surgem no momento de escalonar a aplicação na infraestrutura disponível no momento, mesmo não havendo um planejamento a ser seguido.

Alguns objetivos, não mutuamente excludentes, de quem é responsável por escalonar a aplicação na infraestrutura são: (i) maximizar o número de usuários atendidos; (ii) assegurar que os Acordos de Nível de Serviço (do inglês *Service Level Agreement* - *SLA*) com os clientes estão sendo mantidos; e (iii) minimizar os custos com a execução da aplicação.

O problema de provisão de recursos não é novidade na comunidade acadêmica muito menos no departamento de TI das empresas. Urgaonkar et al. [USC08], trata desse problema para aplicações Web multicamadas, enquanto em [FGG10] o escalonamento de aplicações web simples é otimizado com foco no *SLA* estabelecido com o usuário, levando-se em consideração picos de demanda. No campo das aplicações de e-ciência, Kim et al. [KPFY09] otimizam o escalonamento de aplicações com base no tempo de resposta.

Essas abordagens possuem um ponto em comum, usam métricas muito técnicas e que nem sempre são conhecimento comum de quem é responsável por essa tarefa. Ou ainda, as ferramentas propostas precisam de que o usuário configure certos parâmetros com uma certa acurácia para produzir bons resultados. Visando facilitar mais essa tomada de decisão, o primeiro ponto é o uso de técnicas baseadas em métricas de negócio. Com isso o conhecimento fica mais acessível a quem é responsável pela tomada de decisão no provedor. Naturalmente as métricas mais técnicas serão levadas em consideração, porém acima deste sistema baseado em métricas técnicas estará o sistema baseado em métricas de negócio que irá avaliar diferentes configurações à luz dessas métricas identificando a que trouxer maior retorno para o negócio. O segundo ponto é tornar essa decisão o mais automatizada possível, ou prover uma configuração inicial que permita o uso dessa abordagem sem a necessidade de configurar inicialmente parâmetros sensíveis com tamanha acurácia.

Pretende-se construir um escalonador de aplicações *SaaS* baseado em métricas de negócio que venha a tirar proveito desse tipo de métrica para reduzir os custos do provedor sem comprometer o serviço oferecido. Inicialmente é necessário capturar características dessas aplicações que possam ser usadas na construção do modelo. A partir disso, é necessária uma forma de valorar essas características capturadas e uma maneira bastante comum é o uso de funções de utilidade. Essas funções mapeiam o valor que os usuários atribuem a execução de aplicações [Wil08], podendo ser definidas em termos de tempo de execução, por exemplo. A forma da curva de utilidade, no entanto, depende dos objetivos dos usuários [LS06]. Em se tratando de aplicações Web, como são todas as aplicações *SaaS*, não existe um tempo delimitado de execução da aplicação. A aplicação é de fato um serviço, que está sempre à espera de requisições vindas de diferentes usuários. Um novo modelo que capture esta característica de serviço além dos objetivos dos usuários deve ser criado.

Uma vez caracterizadas as aplicações *SaaS* e seu modelo de valoração, escalonadores cientes dessas métricas de negócio podem tirar proveito dependendo do objetivo escolhido: reduzir custo, reduzir tamanho de fila de espera, aumentar número de requisições atendidas por segundo, por exemplo. Para isso, pretende-se avaliar o desempenho de diferentes heurísticas em diversas situações, inclusive comparando com outras alternativas que tenham sido encontradas na literatura.

2 Objetivo da Proposta

O objetivo principal dessa pesquisa é: **Investigar a viabilidade do uso de técnicas dirigidas por métricas de negócio para a provisão dinâmica de recursos para execução de aplicações SaaS.**

Para cumprir esse objetivo, é necessário atingir alguns objetivos secundários:

- Capturar características importantes das aplicações em estudo para que o modelo construído possa tirar proveito dessas especialidades;
- Criar e avaliar um modelo de escalonador que automatize a alocação de recursos;
- Traçar conclusões gerais a respeito do modelo para que ele não seja útil somente nos cenários avaliados.

3 Relevância da Proposta

O uso de aplicações SaaS está cada vez mais comum nas empresas assim como a academia também tem como tirar proveito desse modelo. Espera-se apontar novos caminhos na área de provisão dinâmica de recursos para esse tipo de aplicação e a sugestão ou amadurecimento de métricas de negócio, bem como a proposta de um escalonador que possa automatizar essa atividade.

Com o sucesso dessa pesquisa, provedores de SaaS poderão tirar um melhor proveito de sua infraestrutura local ou da infraestrutura adquirida em outros provedores fazendo um uso mais eficiente dos recursos atendendo à demanda dos usuários da aplicação. O uso de métricas de negócio torna mais simples a gerência da aplicação uma vez que os resultados de uma decisão refletem diretamente em métricas compreensíveis por quem tem o poder de decisão na empresa. Uma vez bem estudadas e documentadas, essas métricas podem, além de guiar o processo de tomada de decisão, atuar na automação desse processo.

4 Metodologia de Trabalho

Através dos objetivos levantados na seção 2, algumas atividades foram elencadas para o sucesso da pesquisa. Essas atividades estão resumidas na tabela 1 e abaixo segue uma descrição detalhada de cada atividade e dos resultados esperados. O cronograma planejado é apresentado na subseção 4.1.

Atividades	Descrição
A1	Levantamento bibliográfico sobre caracterização e valoração de aplicações <i>SaaS</i>
A2	Caracterização de aplicações <i>SaaS</i>
A3	Elaborar um modelo de valoração de aplicações <i>SaaS</i>
A4	Levantamento bibliográfico sobre provisão dinâmica de recursos, apoiadas em métricas de negócio, para aplicações <i>SaaS</i>
A5	Elaboração de heurísticas para provisão dinâmica de recursos para aplicações <i>SaaS</i>
A6	Avaliação das heurísticas propostas
A7	Escrita da dissertação
A8	Escrita de um artigo para uma conferência internacional ou periódico
A9	Defesa da dissertação

Tabela 1: Atividades Planejadas

A1 - Levantamento bibliográfico sobre caracterização e valoração de aplicações *SaaS*

Essa atividade é fundamental para se ter conhecimento do estado da arte da área. Esse levantamento bibliográfico é fundamental e mais demorado no início da pesquisa mas é retomado a cada ciclo completo da pesquisa. Com isso a revisão da área não fica desatualizada. Essa atividade tem como resultados: (i) parte da seção de trabalhos relacionados da dissertação e do artigo científico; (ii) conhecimento para auxílio nas atividades A2 e A3.

A2 - Caracterização de aplicações *SaaS*

Com base no primeiro levantamento bibliográfico feito, dá-se início ao processo de caracterização de aplicações *SaaS*. Essa atividade tem como resultado a documentação de características importantes a serem levadas em conta na elaboração do modelo de valoração. Essa atividade também pode vir a reduzir o escopo de alcance do trabalho e a pesquisa se focar nas classes mais representativas de aplicações encontradas.

A3 - Elaborar um modelo de valoração de aplicações *SaaS*

Como o foco do trabalho reside nas métricas de negócio, essa atividade usa o conhecimento coletado na atividade A1 e A2 na construção de um modelo que leve em conta as características levantadas. Um modelo já existente pode ser usado, ou ainda, aperfeiçoado. Um objetivo importante dessa fase está na identificação das métricas de negócio que serão alvo na fase de avaliação do escalonador. O resultado dessa fase é um modelo de valoração de aplicações *SaaS* com base em métricas de negócio identificadas.

A4 - Levantamento bibliográfico sobre provisão dinâmica de recursos, apoiadas em métricas de negócio, para aplicações *SaaS*

Essa atividade dá início a linha principal da pesquisa. Com essa revisão, será possível identificar escalonadores/heurísticas/algoritmos para provisão dinâmica que possam ser usados para comparação com as heurísticas propostas. É desejável que os algoritmos usados na comparação sejam baseados em métricas de negócio também. Essa atividade tem como resultados: (i) parte da seção de trabalhos relacionados da dissertação e do artigo; (ii) uma fundamentação para a análise do escalonador proposto; e (iii) uma formulação matemática do problema analisado.

A5 - Elaboração de heurísticas para provisão dinâmica de recursos para aplicações *SaaS*

Uma vez modelada a valoração das aplicações, essa atividade tem por finalidade completar essa modelagem propondo um novo escalonador baseado nas métricas de negócio modela-

das. Espera-se levantar várias propostas de heurística que considerem situações diferentes ou que priorizem métricas diferentes. Os resultados dessa atividade são: (i) uma modelagem analítica de um escalonador baseado em métricas de negócio; (ii) um simulador desse escalonador proposto; (iii) uma modelagem analítica de uma ou mais heurísticas; e (iv) implementações dessas heurísticas para serem avaliadas no simulador.

A6 - Avaliação das heurísticas propostas

Com uma modelagem do problema e das heurísticas é possível tirar conclusões através de uma avaliação analítica de situações bem conhecidas. Essas conclusões podem ser usadas no processo de verificação do modelo de simulação. Com um modelo de simulação implementado, novas situações podem ser analisadas e conclusões mais gerais sobre a solução podem ser traçadas. Nesse caso, deve ser feito antes de cada seção de experimentação um design do experimento com a finalidade de guiar o processo de seleção de fatores e tratamentos importantes para serem analisados. Os resultados dessa atividade são: (i) a seção de modelagem e análise de resultados da dissertação e do artigo científico. As conclusões extraídas dessa análise podem requerer um refinamento do modelo e portanto um retorno às atividades anteriores.

A7 - Escrita da dissertação

Essa atividade será realizada durante todo o processo da pesquisa documentando os resultados de todas as outras atividades.

A8 - Escrita de um artigo para uma conferência internacional ou periódico

No final dos últimos ciclos da pesquisa (estudo - modelagem - avaliação), será estudada a possibilidade de publicação das conclusões extraídas dos resultados parciais do trabalho numa conferência internacional ou periódico. O resultado dessa atividade é a publicação dos resultados na comunidade internacional.

A9 - Defesa da dissertação

Essa atividade põe fim ao ciclo dessa pesquisa, apresentando os resultados a comunidade científica e contribuindo com o estado da arte da área pesquisada.

4.1 Cronograma

O planejamento de execução das atividades identificadas na tabela 1 está representado na tabela 2.

	Mar/2011	Abr/2011	Mai/2011	Jun/2011	Jul/2011	Ago/2011
A1	■	■				
A2		■			■	
A3			■		■	
A4				■		■
A5				■		■
A6					■	■
A7	■	■	■	■	■	■
A8						
A9						
	Set/2011	Out/2011	Nov/2011	Dez/2011	Jan/2012	Fev/2012
A1						
A2	■		■			
A3	■		■			
A4		■		■		
A5		■		■		
A6	■	■	■	■	■	
A7	■	■	■	■	■	■
A8			■			
A9						■

Tabela 2: Cronograma

Referências

- [ama11a] Amazon ec2 instance purchasing options. <http://aws.amazon.com/ec2/purchasing-options/>, Fevereiro 2011.
- [ama11b] Amazon Elastic Compute Cloud (Amazon EC2). <http://aws.amazon.com/ec2/>, Fevereiro 2011.
- [BB00] Pearl Brereton and David Budgen. Component-based systems: A classification of issues. *Computer*, 33:54–62, November 2000.
- [ECM08] Javier Espadas, David Concha, and Arturo Molina. Application Development over Software-as-a-Service Platforms. *2008 The Third International Conference on Software Engineering Advances*, pages 97–104, Outubro 2008.
- [FGG10] J.O. Fito, I. Goiri, and J. Guitart. Sla-driven elastic cloud hosting provider. In *Parallel, Distributed and Network-Based Processing (PDP), 2010 18th Euro-micro International Conference on*, pages 111 –118, Fevereiro 2010.
- [glo11] Globus Online. <http://www.globusonline.org/>, Fevereiro 2011.
- [gog11] Cloud Hosting, Cloud Servers, Hybrid Hosting, Cloud Infrastructure from Go-Grid. <http://www.gogrid.com/>, Fevereiro 2011.
- [goo11] Google App Engine - Google Code. <http://code.google.com/appengine/>, Fevereiro 2011.
- [KPFY09] Hyunjoo Kim, M. Parashar, D.J. Foran, and Lin Yang. Investigating the use of autonomic cloudbursts for high-throughput medical image registration. In *Grid Computing, 2009 10th IEEE/ACM International Conference on*, pages 34 –41, Outubro 2009.
- [LS06] Cynthia Bailey Lee and Allan Snaveley. On the user-scheduler dialogue: Studies of user-provided runtime estimates and utility functions. *Int. J. High Perform. Comput. Appl.*, 20:495–506, Novembro 2006.

- [LT08] HanCheng Liao and ChangQi Tao. An Anatomy to SaaS Business Mode Based on Internet. *2008 International Conference on Management of e-Commerce and e-Government*, pages 215–220, October 2008.
- [MMLP09] Ralph Mietzner, Andreas Metzger, Frank Leymann, and Klaus Pohl. Variability modeling to support customization and deployment of multi-tenant-aware Software as a Service applications. *2009 ICSE Workshop on Principles of Engineering Service Oriented Systems*, 215483:18–25, Maio 2009.
- [Nat07] Yefim V. Natis. Introducing saas-enabled application platforms: Features, roles and futures, 2007.
- [r-p11] R-PHP. <http://dssm.unipa.it/R-php/>, Fevereiro 2011.
- [TBB03] Mark Turner, David Budgen, and Pearl Brereton. Turning software into a service. *Computer*, 36:38–44, October 2003.
- [USC08] Bhuvan Urgaonkar, Prashant Shenoy, and Abhishek Chandra. Agile dynamic provisioning of multi-tier internet applications. *ACM Transactions on*, 2008.
- [VRMCL08] Luis M. Vaquero, Luis Roderio-Merino, Juan Caceres, and Maik Lindner. A break in the clouds: towards a cloud definition. *SIGCOMM Comput. Commun. Rev.*, 39:50–55, December 2008.
- [Wil08] John Wilkes. Utility functions, prices, and negotiation, 2008.