

**UNIVERSIDADE FEDERAL DE SANTA CATARINA  
CENTRO TECNOLÓGICO  
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA  
COMPUTAÇÃO**

Hans Alberto Franke

**UMA ABORDAGEM DE ACORDO DE NÍVEL DE SERVIÇO  
PARA COMPUTAÇÃO EM NUVEM**

FLORIANÓPOLIS-SC  
2010

Hans Alberto Franke

**UMA ABORDAGEM DE ACORDO DE NÍVEL DE SERVIÇO  
PARA COMPUTAÇÃO EM NUVEM**

Dissertação apresentada ao Programa de Pós Graduação em Ciências da Computação da Universidade Federal de Santa Catarina como parte dos requisitos para a obtenção do grau de Mestre em Computação

Mestrando: Hans Alberto Franke

Orientador: Prof. Dr. Carlos Becker Westphall

Área de Concentração: Ciências da Computação

FLORIANÓPOLIS-SC  
2010

**Catálogo na fonte elaborada pela biblioteca da  
Universidade Federal de Santa Catarina**

F829a Franke, Hans Alberto

Uma abordagem de acordo de nível de serviço para  
Computação em nuvem [dissertação] / Hans Alberto Franke ;  
orientador, Carlos Becker Westphall. - Florianópolis, SC,  
2010.

**1 v. : il., grafs., tabs.**

Dissertação (mestrado) - Universidade Federal de Santa  
Catarina, Centro Tecnológico. Programa de Pós-Graduação em  
Ciência da Computação.

Inclui referências

1. Informática. 2. Ciência da computação. 3. Qualidade  
dos serviços. I. Westphall, Carlos Becker. II. Universidade  
Federal de Santa Catarina. Programa de Pós-Graduação em  
Ciência da Computação. III. Título.

CDU 681

Hans Alberto Franke

Esta Dissertação foi julgada adequada para a obtenção do título de Mestre em Ciência da Computação - Área de Concentração Ciências de Computação e aprovada em sua forma final pelo Programa de Pós-Graduação em Ciência da Computação.

**UMA ABORDAGEM DE ACORDO DE NÍVEL DE SERVIÇO  
PARA COMPUTAÇÃO EM NUVEM**

Florianópolis, 26 de fevereiro de 2010.

\_\_\_\_\_  
Prof. Dr. Mário Antônio Ribeiro Dantas  
Coordenador do Programa de

**Banca Examinadora:**

\_\_\_\_\_  
Prof. Dr. Carlos Becker Westphall  
Orientador

\_\_\_\_\_  
Prof. João Bosco Mangueira Sobral

\_\_\_\_\_  
Prof. Dr. Mário Antônio Ribeiro Dantas

\_\_\_\_\_  
Prof. Dr. Carla Merkle Westphall

\_\_\_\_\_  
Prof. Dr. Bruno Richard Schulze

*"The mere formulation of a problem is far more essential than its solution, which may be merely a matter of mathematical or experimental skills. To raise new questions, new possibilities, to regard old problems from a new angle requires creative imagination and marks real advances in science."  
(Albert Einstein)*

## AGRADECIMENTOS

*Agradeço acima de tudo a meus pais que sempre incentivaram e acreditaram que chegaria até aqui. Nos momentos de dificuldade, dúvidas, incertezas, foram eles que apoiaram. Eles me propiciaram todos os recursos necessários para me dedicar a estudar, me dando conforto e tranquilidade. Não poderia esquecer-se da minha irmã, que acha que tem o melhor irmão do mundo!*

*Ao professor Westphall pelo apoio e oportunidade. Por diversas vezes precisei do seu auxílio, companheirismo e tempo, principalmente tempo. Sou grato por todo o tempo dividido comigo, desde a graduação, bolsa e mestrado. Seus conhecimentos e ajuda incondicional em momentos difíceis propiciaram que estas linhas fossem escritas.*

*Ao grande amigo e ouvinte Kleber. Sem a sua ajuda, motivação, ideias e discussões eu teria tido um caminho bem mais árduo, e talvez sem tanto mérito. Ao também importante Diego que me guiou durante o processo de simulação, com ajuda irrestrita.*

*Ao time do LRG, que revisou, deu ideias, críticas, sugestões e suporte para entrar no universo das Nuvens Computacionais.*

*E o agradecimento especial a minha namorada Tamires que entendeu os fins de semana, os feriados, as madrugadas estudando e escrevendo ao meu lado. Sua participação e principalmente o apoio foram fundamentais para conciliar, trabalho e estudo; e por muitas vezes esquecer o namoro. Todos os puxões de orelha e cobranças foram fundamentais!*

## RESUMO

Computação em Nuvem é baseada na visão de prover serviços como utilidades (e.g. água, luz), onde consumidores podem acessar os serviços em qualquer lugar do mundo e, por demanda, pagar apenas pela quantidade que consomem. O tamanho, complexidade e escopo desse paradigma computacional faz com que surja a necessidade dos provedores de serviço implementarem uma forma mais escalável para negociar, orquestrar e automatizar o processo de Acordo de Nível de Serviço.

Como prover um serviço onde o usuário contrate apenas aquilo que necessite, pagando apenas pelo uso, de maneira eficiente, simples, transparente e da forma mais barata possível? Como decidir entre Nuvem ou estrutura tradicional com micros dedicados? Tendo Acordos de Nível de Serviço (SLA) bem definidos e previamente acordados entre usuários e fornecedor, que possam mudar de acordo com a necessidade do usuário; usando mais ou menos recursos de maneira automática baseada na demanda; utilizando de maneira colaborativa toda a infraestrutura disponível, reduzindo custos e preservando recursos; como descobrir os recursos/serviços de maneira eficiente com o menor custo possível;

Neste trabalho propõe-se uma abordagem para mapeamento de SLA entre processos de negócio e infraestrutura de TI, garantindo que os acordos de nível de serviço sejam endereçados de maneira dinâmica, baseado na demanda do usuário. Isto se dá através de uma renegociação dinâmica entre os clientes e uma lista de provedores, gerando o menor custo possível ao usuário e otimizando os recursos dos provedores.

É proposto uma arquitetura que utiliza parâmetros definidos em um arquivo de Objetivo de Nível de Serviço – SLO, para negociar e escolher os melhores provedores de nuvem. A proposta foi simulada no ambiente CloudAnalyst, comparando-se abordagens de SLA estáticas e dinâmicas. Os resultados apresentados nas simulações com negociação dinâmica provam que a proposta de endereçar processos de negócio em infraestrutura de TI de maneira dinâmica gera um ganho ao ambiente e ao usuário.

**Palavras-Chaves:** Computação em Nuvem, Acordo de Nível de Serviço, Qualidade de Serviço.

## ABSTRACT

Cloud Computing is based on the vision of providing services such as utilities (e.g. water, electricity), where consumers can access services anywhere in the world and, on demand, pay only the amount they consume. The size, complexity and scope of this computational paradigm means that there is a need for service providers to implement a more scalable way to negotiate, orchestrate and automate the process of Service Level Agreement.

How to provide a service where the user engages only what you need, paying only for the use of efficient, simple, transparent and as cheaply as possible? How to decide between Cloud or traditional structure with dedicated micros? With Service Level Agreements (SLA) well defined and agreed between user and supplier, which may change depending on the user's needs, using more or less automatic way of resources based on demand, using a collaborative manner throughout the infrastructure structure available, reducing costs and preserving resources, how to find the resources / services effectively at the lowest possible cost.

In this work we propose an approach for mapping SLA between business processes and IT infrastructure, ensuring that service level agreements are addressed proactively, based on user demand. This occurs through a dynamic renegotiation between customers and a list of providers, generating the lowest possible cost to the user and optimizing the resources of the providers.

It proposed an architecture that uses parameters defined in a file of the Service Level Goal - SLO, to negotiate and choose the best providers of cloud. The proposal was simulated in a CloudAnalyst environment, comparing the approaches of static and dynamic SLA. The results presented in the negotiating dynamics simulations show that the proposal to address business processes in IT infrastructure dynamically generates a gain to the environment and the user.

**Keywords:** Cloud Computing, Service Level Agreement, Quality of Service.

## LISTA DE FIGURAS

FIGURA 2.1 VISÃO DA RELAÇÃO ENTRE GRID, UTILITY E ON-DEMAND COMPUTING. ....	21
FIGURA 2.3.1 - UNIVERSO DO SLA.....	28
FIGURA 2.3.7 - PIRÂMIDE MODELO DO NÍVEL DE SERVIÇO .....	34
FIGURA 3.1. MIDDLEWARE DE META NEGOCIAÇÃO.....	39
FIGURA 3.7A MODELAGEM DE UM SLA.....	48
FIGURA 3.7B FLUXOGRAMA DE ANÁLISE DE NEGÓCIO. ....	49
FIGURA 3.7C: DTD PARA COMUNICAÇÃO ENTRE OS PARTICIPANTES DA NEGOCIAÇÃO. ....	50
FIGURA 3.7D. RELAÇÃO ENTRE CONSUMIDOR E PROVEDOR DE SLA.....	50
FIGURA 3.11 MODELO DE FALHA USADO EM UMA ESTRUTURA DE SLO.....	55
FIGURA 3.12A MODELO DE NEGOCIAÇÃO DINÂMICA SIMULADA. ...	57
FIGURA 3.12B EXEMPLO DE EXECUÇÃO: COMUNICAÇÃO ENTRE DIVERSAS BASES DE USUÁRIOS (UB) E SERVIDORES (DC) ALOCADOS NO GLOBO. ....	58
FIGURA 3.12C CONFIGURADOR CRIADO PARA FACILITAR O AJUSTE DE PARÂMETROS .....	60
FIGURA 3.12D RESUMO EXECUTIVO COMPARANDO-SE: ESTÁTICA X DINÂMICA POR PERÍODO: .....	64
FIGURA 4.2A - ARQUIVO COM SLA SIMPLES. ....	75
FIGURA 4.2B. SERVICES LAYERS NO AZURE.....	76

## LISTA DE TABELAS

TABELA 2.3.6 - PRINCIPAIS COMPONENTES DE UM SLA .....	31
TABELA 2.4: COMPARATIVO ENTRE DIFERENTES PLATAFORMAS PARA CLOUD.....	37
TABELA 3.8 ESTRUTURA DOS OBJETIVOS DE NÍVEL DE SERVIÇO...	52
TABELA 3.12A CONFIGURAÇÃO DOS CLIENTES NO AMBIENTE .....	58
TABELA 3.12B. CONFIGURAÇÃO DOS SERVIDORES NO AMBIENTE.	59
TABELA 3.12C CUSTO EM UMA SIMULAÇÃO ESTÁTICA SEMANAL.	61
TABELA 3.12D CUSTO EM UMA SIMULAÇÃO DINÂMICA SEMANAL. .....	61
TABELA 3.12E TAXA DE SUCESSO EM UMA SIMULAÇÃO DINÂMICA SEMANAL.....	62
TABELA 3.12F TAXA DE SUCESSO EM UMA SIMULAÇÃO ESTÁTICA SEMANAL.....	62
TABELA 3.12G CUSTO EM UMA SIMULAÇÃO ESTÁTICA SEMESTRAL. .....	62
TABELA 3.12H CUSTO EM UMA SIMULAÇÃO DINÂMICA SEMESTRAL.....	63
TABELA 3.12I TAXA DE SUCESSO EM UMA SIMULAÇÃO ESTÁTICA SEMESTRAL.....	63
TABELA 3.12J TAXA DE SUCESSO EM UMA SIMULAÇÃO DINÂMICA SEMESTRAL.....	63
TABELA 4.1.5A SISTEMAS OPERACIONAIS DISPONÍVEIS NO AMAZON .....	70
TABELA 4.1.5B – SOFTWARES DISPONÍVEIS.....	70
TABELA 4.1.6A – PREÇOS DE INSTÂNCIAS ON-DEMAND DO AMAZON .....	71
TABELA 4.1.6B – PREÇO DE TRANSFERÊNCIA DE DADOS NA INTERNET ATRAVÉS DO AMAZON .....	71

**LISTA DE ABREVIATURAS E SIGLAS**

<b>SaaS</b>	Software como Serviço
<b>PaaS</b>	Plataforma como Serviço
<b>IaaS</b>	Infraestrutura as a Service
<b>NFP</b>	Propriedades não funcionais
<b>WSN</b>	Wireless Sensor Networks
<b>SSL</b>	<i>Secure Socket Layers</i>
<b>SOA</b>	Arquitetura orientada a serviços
<b>VM</b>	Máquinas Virtuais

## SUMÁRIO

<b>1. INTRODUÇÃO.....</b>	<b>14</b>
1.1 TEMA E PROBLEMA DA PESQUISA .....	14
1.2 MOTIVAÇÃO.....	15
1.3 ORGANIZAÇÃO DO TRABALHO.....	18
<b>2. CONCEITOS E MOTIVAÇÃO .....</b>	<b>20</b>
2.1 GRIDS COMPUTACIONAIS.....	20
2.2 CLOUD COMPUTING .....	22
2.2.1 Características.....	24
2.2.2 Software como Serviço (SaaS).....	25
2.2.3 IaaS – Infra-estrutura como Serviço.....	26
2.2.4 PaaS – Plataforma como Serviço .....	26
2.2.5 Exemplos de aplicações em Cloud Computing .....	26
2.3 SLA – SERVICE LEVEL AGREEMENT .....	27
2.3.1 Universo do SLA .....	28
2.3.2 SLA interno .....	29
2.3.3 SLA do serviço .....	29
2.3.4 SLA fornecedor .....	29
2.3.5 Relação entre o SLA e o Plano de Negócios.....	30
2.3.6 Componentes de um SLA .....	30
2.3.7 Processo do SLA .....	32
2.4 TRABALHOS RELACIONADOS .....	34
<b>3. PROPOSTA.....</b>	<b>38</b>
3.1 NEGOCIAÇÃO DE SLA .....	38
3.2 ORQUESTRAÇÃO .....	40
3.3 DESAFIOS NA AUTOMAÇÃO DE ACORDO DE NÍVEL DE SERVIÇO..	42
3.4 MECANISMOS E ALGORITMOS PARA ALOCAÇÃO DE RECURSOS ..	44
3.5 ARQUITETURA.....	45
3.6 UMA ABORDAGEM PARA MAPEAMENTO DE SLA DINAMICAMENTE .....	46
3.7 ARQUITETURA DE UM MODELO DE NEGOCIAÇÃO DINÂMICA DE SLA .....	47
3.8 FORMALIZAÇÃO E NEGOCIAÇÃO DE SLA .....	51
3.9 APLICAÇÃO DE QOS DE SLA EM NUVENS .....	52
3.10 DESEMPENHO.....	53
3.11 FIDELIDADE .....	54
3.12 AVALIAÇÃO EXPERIMENTAL .....	56

3.13 OBSERVAÇÕES SOBRE O CAPÍTULO .....	64
<b>4. ESTUDO DE CASO.....</b>	<b>66</b>
4.1 ESTUDO DE CASO – OPEN UTILITY CLOUD COMPUTING.....	66
4.1.1 Coleta de dados.....	67
4.1.2 Visão Geral.....	67
4.1.3 Serviços de destaque .....	68
4.1.4 Tipos de instâncias.....	69
4.1.5 Sistemas operacionais e softwares.....	69
4.1.6 Preços .....	71
4.1.7 Análise.....	72
4.1.8 Resultados e contribuições.....	73
4. 2 SISTEMA OPERACIONAL EM CLOUD: AZURE.....	74
<b>5. CONCLUSÕES .....</b>	<b>78</b>
5.1 TRABALHOS FUTUROS.....	79
<b>REFERÊNCIAS .....</b>	<b>80</b>
<b>ANEXO A – CÓDIGO FONTE - CLOUDANALYST .....</b>	<b>87</b>
<b>ANEXO B – RESULTADOS SIMULAÇÃO.....</b>	<b>88</b>

# 1. INTRODUÇÃO

## 1.1 Tema e Problema da Pesquisa

O aumento progressivo das necessidades por capacidade de processamento e recursos computacionais resultou no compartilhamento em grande escala entre instituições diferentes que possuem afinidades entre si e que podem, em momentos de ociosidade dos recursos, ajudar a suprir as necessidades de outra organização. Desta forma, em meados da década de 90, começou a surgir computação em grid, um conceito criado para definir um ramo de pesquisa dentro da computação distribuída.

Foster e Kesselman (Foster 2000), definem Grid como um sistema heterogêneo com uma grande diversidade de hardware e software, distribuído geograficamente, possuindo dinamicidade onde recursos conectam e desconectam do Grid, ao longo do tempo, com um controle distribuído de políticas em que cada organização é responsável por seu domínio.

A solução fornecida pela computação em Grid vai além de criar redes de processamento e armazenamento distribuídos. Argumenta-se que é também uma técnica poderosa para promover a homogeneização ou a virtualização criando redes de fornecedores de serviço onde cada nó compartilha serviços ao Grid. Executam em ambientes heterogêneos, como nas Wireless Sensor Networks (WSN), em hardware (desktop, PDA, dispositivos móveis). As aplicações devem lidar com as diversidades e entregar um serviço de qualidade. Consequentemente, a pergunta está em Como integrar estes ambientes em uma infraestrutura comum para fornecer serviços? ou; Qual a plataforma ideal para promover a homogeneização em um Grid de Wireless Sensor Networks (Franke 2007)?

Diversos estudos estão sendo realizados no âmbito desta pergunta. Por exemplo, na comunicação de dados simples, rápida e mais confiável; em redes de transmissão de dados (Uskela 2003); no hardware, novos dispositivos com mais recursos e em plataformas mais simples de desenvolvimento, e; na tecnologia de programação há a criação de metodologias do desenvolvimento capazes de lidar com as restrições de recursos existentes (Guiguere 2001). Entretanto, sugere-se que ainda falta na literatura uma solução completa e reutilizável para integrar Wireless Sensor Networks, provedores de serviços, dispositivos estáticos (workstations) e móveis de computação. Sugere-se que esta

abertura na literatura apresenta uma oportunidade de contribuir com uma estrutura para proporcionar o autogerenciamento do grid e uma interface e estruturas que possibilitem a aplicação de inteligência artificial para que o middleware de grid seja capaz de adaptar sua estrutura de forma autônoma de acordo com cada situação descobrindo e provendo os serviços respeitando as limitações dos dispositivos móveis.

A evolução do conceito de Grid Computing aparece com o surgimento da computação em Nuvem ou “Cloud Computing”. Nela temos uma abstração maior que o Grid, onde o usuário passa a contratar um “serviço”, seja um SO, Software, armazenamento, mobilidade, enfim; ele não enxerga as camadas inferiores, somente o seu produto.

Como prover um serviço onde o usuário contrate apenas aquilo que necessite, pagando apenas pelo uso, de maneira eficiente, simples, transparente e da forma mais barata possível? Como decidir entre Nuvem ou estrutura tradicional com micros dedicados?

- Tendo Acordos de Nível de Serviço (SLA) bem definidos e previamente acordados entre usuários e fornecedor, que possam mudar de acordo com a necessidade do usuário;

- Escalabilidade: possa usar mais ou menos recursos de maneira automática baseada na demanda;

- Utilizando de maneira colaborativa toda a infraestrutura disponível, reduzindo custos e preservando recursos;

- Como descobrir os recursos/serviços de maneira eficiente com o menor custo possível;

## 1.2 Motivação

Computação em nuvem, é uma maneira nebulosa para fornecer infraestrutura barata, parece encontrar acima uma barreira muito sólida. Serviços baseados na Internet como infraestrutura como a EC2, *Amazon.com 's Elastic Cloud Computing*, ainda não são capazes de suprir os acordos de nível de serviço (SLAs).

A *Cloud computing* corta drasticamente os custos, permitindo que os clientes utilizem servidores, virtualização, armazenamento, processamento e paguem por eles somente o necessário. Isso funciona muito bem para iniciantes, principalmente aqueles que prestam um serviço online, por exemplo, um software genérico de gestão de estoque. Mas, para empresas com necessidades de computação crítica e madura de Software as Service (SaaS), a nuvem, em virtude de toda a sua

elasticidade, não pode fornecer a personalização necessária e acordos de nível de serviço.

Esse novo paradigma tornou-se um valioso serviço para elevar a inteligência de negócios, mas as empresas mantiveram as funções mais críticas, como aplicações de planejamento de recursos corporativos.

Como exemplo, podemos citar a *Visible Measures Boston Corp* (Visible 2009) que utiliza o EC2 como uma extensão do seu centro de armazenamento. A nuvem apoia o desenvolvimento e serve como um backup em caso de uma falha ou um aumento súbito da procura que excede a capacidade de seus servidores. Porém não deixa as funções principais rodando na nuvem porque necessidade de um controle muito grande na coleção de dados, que surgem do compartilhando de vídeos e publicidades entre os clientes.

Surgem então 4 questões chaves que devem ser consideradas para se garantir SLA:

- Controle: o acordo de nível de serviço deve garantir qualidade e eficiência das funções operacionais como: disponibilidade, confiabilidade, desempenho, manutenção, backup, recuperação de falhas; estas eram usadas sob controle do TI locais, executando com premissas e gestão próprias, porém agora são executadas sob o controle do prestador de serviço, visto que as aplicações são executadas dentro da nuvem.

- Risco operacional: deve resolver riscos inerentes a ambientes que são compartilhados, como: segurança, privacidade e propriedade de dados a fim de garantir a disponibilidade do serviço.

- Risco de negócio: as companhias de computação em nuvem vão ficando mais eficientes com a maturidade, conseguindo agregar valor e sucesso, logo passarão a acrescentar garantias de sucesso de negócios nos SLA's – como garantia de tempo de implementação, qualidade do suporte técnico, valor de negócio recebido e até garantia de dinheiro devolvido caso o cliente não esteja satisfeito com o serviço prestado. Serviços de nuvem podem contornar os riscos de negócios baseados nas suas experiências bem sucedidos, uma vez que seus projetos bem sucedidos diluem o custo envolvido com a infraestrutura.

- Penalidades, recompensas e transparência: o SLA deve garantir penalidades financeiras quando ocorrerem violações no nível de serviço acordados. Se não houver uma despesa para o prestador de serviço em caso de descumprimento de um acordo, o SLA não tem valor. Igualmente, o comprador deve pagar uma recompensa em caso de serviços extraordinários prestados que trouxeram benefícios. Se 100% de disponibilidade é uma meta importante, deve-se pagar um bônus ao

prestador em caso de ser cumprido. Os prestadores de serviços devem manter um website com a real performance e atingimento de seus SLA's, garantindo de forma clara e transparente aos compradores o serviço prestado.

Estas ideias levam a especificar termos de SLA entre clientes e prestadores:

#### 1. Controle dos SLAs:

- \* Estabelecimento de um sistema compatível com o SLA, baseando-se em uma média mensal, com bônus para execuções e aumento por *downtime* sob os níveis;

- \* Estabelecimento de um sistema responsório por tempo SLA, baseado na média mensal de respostas por tempo, que aponte as perdas nas baixas performances do sistema;

- \* Sob a falta de servidor por algum erro, a recuperação do SLA nos casos de falha na estrutura da entidade.

#### 2. Riscos Operacionais de SLA:

- \* Certificar-se de que o usuário pode ter a informação recuperada nos casos em que ele decidir sair e que nenhuma ambiguidade ocorra em suas configurações;

- \* Certificar-se de que a entidade irá prestar assistência nos casos de migração, considerando uma adequada taxa de serviço por isso;

- \* Certificar-se de que o usuário pode reter sua própria configuração e dados no sistema do fornecedor, considerando uma taxa de serviços, para os casos em que o usuário não deseje perder o acesso a suas configurações e acesso às suas informações;

- \* Verificar a política de privacidade do fornecedor e se certificar de que a mesma é clara de acordo com a SLA e que não existe nenhuma brecha;

- \* Verificar se o fornecedor se submete a auditorias anuais sobre a SAS70 Type II (Sas70 2009), e se além dessa o fornecedor passa pela verificação de segurança anual.

#### 3. Riscos de Negócios nos níveis de SLA:

- \* Estabelecimento de uma resolução de erro, considerando diferentes servidores para diferentes níveis de serviço e novamente penalidades por não receber resposta;

- \* Estabelecimento de critérios para qualidade e tempo médio para a apresentação do profissional para implementação da melhora do funcionamento;

\* Estabelecimento das garantias referente à resposta de comunicação – analisando por meses as características e atualização do produto a fim de entender como as solicitações do usuário por novas características e alteração no produto serão priorizadas.

#### 4. Multas / Recompensas e transparência nos níveis do SLA:

\* Em cada uma das partes citadas acima, é do fornecedor, a responsabilidade pela documentação e metodologia para mensurar o cálculo das multas e recompensas;

\* O usuário receberá automaticamente o crédito no caso de ocorrer uma falha no sistema que afete seu acesso, ou o usuário precisa contatar o fornecedor para solicitar o crédito de recompensa;

\* O fornecedor deve prover ao usuário notificações de vulnerabilidade do serviço e informar sobre previsões de problemas gerados por atualizações;

\* Comunicados e Notas do fornecedor em tempo real, atualização do website que informe a performance profissional e problemas inesperados em tempo real. Se não houver a manifestação do fornecedor por esses meios em caso de problemas, o usuário deve procurar por outro fornecedor;

O segredo por trás de tudo isso é que a nuvem pode tornar tudo mais barato e muito melhor para todos os clientes. Pois, mesmo expandindo os custos, os fornecedores podem fazer melhor por milhões de usuários. A melhor nuvem seria se os fornecedores pudessem identificar que tudo é uma enorme vantagem competitiva e que isso influencia significativamente no valor para seus clientes.

### 1.3 Organização do Trabalho

Este trabalho está dividido da seguinte forma:

- O primeiro capítulo apresenta uma introdução ao tema, contexto e problema da pesquisa, colocando suas questões e objetivos, bem como, sua justificativa de realização e a organização do estudo;
- O segundo capítulo apresenta a motivação que impulsiona este tipo de pesquisa, a definição de Grid, nuvem e SLA;
- O terceiro capítulo é referente a esta proposta de pesquisa e a solução dos problemas apresentados no capítulo 1. Apresentação de uma proposta de acordo de nível de serviço dinâmico. Simulação da proposta

no ambiente CloudAnalyst e apresentação dos resultados para validação da proposta

- O quarto capítulo apresenta dois estudos de caso: *Open Utility Computing* e como prover SLA em um Sistema Operacional em Nuvem; introduzindo a ideia de Computação em Nuvem completando a proposta, apresentando exemplos e resultados;
- O quinto capítulo aborda as considerações finais, mostrando como e onde foram alcançados os objetivos, descreve as limitações da pesquisa e propõe novas direções de estudo em função da complementação do problema enfocado.

## 2. CONCEITOS E MOTIVAÇÃO

### 2.1 Grids Computacionais

A tecnologia de Grades de computadores ou Grids, segundo (ASSUNÇÃO 2004), começou a ser difundida no final da década de 90 com a descrição de uma arquitetura que possibilitava um conjunto de recursos geograficamente distribuídos serem utilizados para processar grandes volumes de dados.

Em 1998, Foster e Kesselman (FOSTER 1998) definiram Grids como sendo “uma infraestrutura de hardware e software que fornece um pervasivo, consistente, acessível e confiável acesso a capacidades computacionais de alto desempenho”.

Em um trabalho mais recente (FOSTER 2001) define Grid como sendo “compartilhamento coordenado de recursos e solucionador de problemas em instituições ou organizações virtuais dinâmicas”.

Em outro artigo (FOSTER 2002) redefine o conceito de Grid ao descrever três pontos importantes e enfatiza que Grid é um sistema que:

a) coordena recursos que não estão sujeitos a um controle centralizado, ou seja, recursos em diferentes unidades administrativas são coordenados e integrados por meio de uma Grid;

b) usa de protocolos e interfaces de padrão abertos e para propósito geral, ou seja, uma Grid é construída e gerida por protocolos e interfaces de propósito geral que atendem a questões fundamentais como autenticação, autorização e descobrimento e acesso aos recursos, e;

c) entrega qualidade de serviço não trivial, ou seja, uma Grid permite por meio de seus recursos combinados entregarem diferentes qualidades de serviço, como tempo de resposta, disponibilidade, banda passante, segurança entre outros.

Baseado nesta evolução das definições de grid computing, presentemente Grid possui definições bem próximas às anteriores, sendo uma delas, “Grid Computing habilita organizações a trocar e compartilhar informações e recursos computacionais entre departamentos e organizações de forma segura e eficiente” (GRIDFORUM 2005).

Logo, grades de computadores possuem um conceito intrínseco que permeia este trabalho, pois além de possibilitar troca de informações e compartilhamento de recursos, possuem uma característica essencial que é a homogeneização de dispositivos, ou seja, todos os componentes da grade de computadores são nodos, seja um dispositivo móvel como um

PDA, seja um dispositivo de armazenamento (*storage device*) ou um sensor. A comunicação entre estes e o compartilhamento de recursos torna o uso da tecnologia de grades de computadores fundamental para solucionar estas questões de forma eficaz.

O termo Grid é o que compreende melhor esse novo paradigma de serviços Tecnologia de Informação (TI), que tende a substituir a verticalização, paradigma onde as aplicações eram suportadas por vários servidores e estes quando uma nova aplicação necessitava de mais poder de processamento seriam então necessárias novas aquisições. Fazendo uma alusão a adicionar, colocar em cima mais recursos, ou seja, novos servidores para as novas aplicações.

Atualmente tem-se um novo paradigma: a integração horizontal dos recursos, onde os recursos de hardware, como os servidores entre outros, são integrados e gerenciados por meio de um sistema Grid que faz alocação de recursos para determinada aplicação, gerenciamento desta aplicação e possui interfaces padronizadas para alocação de outras aplicações, gerando um menor custo e uma maior otimização dos recursos disponíveis.

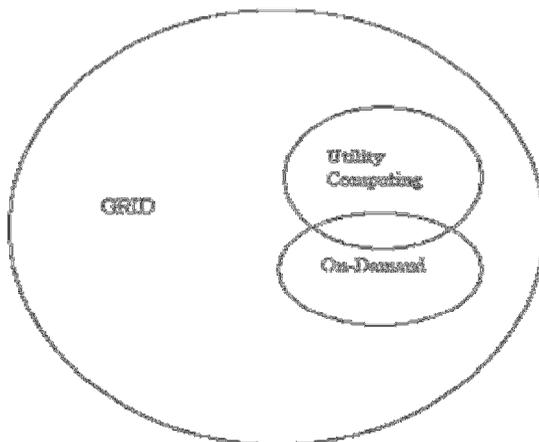


Figura 2.1 Visão da relação entre Grid, Utility e On-demand Computing.  
FONTE: Adaptado de FOSTER (2005).

O termo Grid além destas características acima descritas ainda suplanta termos como “on-demand”, um termo muito usado para denotar sistemas e tecnologias que permitem usuários ou aplicações adquirirem recursos adicionais para satisfazer mudanças de requisitos, e termos

como “Utility Computing” que se refere a uma separação entre provedores de serviços e consumidores e uma capacidade para negociar níveis de qualidade de serviços. Esta superioridade pode ser visualizada na Figura 2.1.

## 2.2 Cloud Computing

Segundo (Armbrust 2009) e (Parkhill 1966), define-se:

*Cloud Computing, o sonho de longa data de computação como utilidade, tem o potencial de transformar uma grande parte da Indústria de TI, tornando o software ainda mais atraente como um serviço e mudando a forma como o hardware é concebido e comprado.*

Desenvolvedores com ideias inovadoras para novos serviços de Internet já não precisam ter grandes gastos de capital em hardware para implementar o seu serviço ou a despesa de mão-de-obra para operá-lo. Eles não precisam se preocupar com overprovisioning de um serviço cuja popularidade não cumprir as suas previsões, assim, o desperdício de recursos onerosos, ou underprovisioning para um que se torna popular, perdendo potenciais clientes e receitas. Além disso, as empresas com grande lote de tarefas pode obter resultados mais rapidamente tanto quanto seus programas possam ser escaláveis, uma vez que utilizar 1000 servidores por uma hora não custa mais do que usar um servidor por 1000 horas. Esta elasticidade dos recursos, sem pagar um prêmio por larga escala, não tem precedentes na história da TI.

Computação em nuvem refere-se às aplicações entregues como serviços através da Internet e do hardware e sistemas de software nos datacenters que oferecem esses serviços (Bechtolsheim 2008). Os próprios serviços têm sido referidos como Software as a Service (SaaS). O hardware e software é o que chamamos de nuvem. Quando uma nuvem é disponibilizada em sistema *pay-as-you-go* para o público em geral, podemos chamá-lo de uma nuvem pública, o serviço que está sendo vendido é *Utility Computing*. Usa-se o termo nuvem privada para se referir a centros de dados internos de uma empresa ou outra organização, que não sejam disponibilizados ao público em geral. Assim, o *Cloud Computing* é a soma de *Utility Computing* e SaaS, mas não incluem nuvens privadas. As pessoas podem ser usuários ou fornecedores de SaaS, ou usuários ou prestadores de *Utility Computing*.

De um ponto de vista do hardware, três aspectos são novos em Cloud Computing (Vogels 2008).

1. A ilusão de infinitos recursos de computação disponíveis sob demanda, eliminando assim a necessidade de usuários de Cloud Computing de planejar muito à frente o provisionamento de recursos.
2. A eliminação de um compromisso por usuários Cloud, permitindo que as empresas comecem pequenas e aumentem seus recursos de hardware somente quando há um aumento de suas necessidades.
3. A capacidade de pagamento por utilização dos recursos computacionais em uma base de curto prazo conforme necessidade (e.g. processamento por hora e armazenagem por dia) e liberá-los quando necessário.

Computação em Nuvem se refere, essencialmente, à ideia de utilizarmos, em qualquer lugar e independente de plataforma, as mais variadas aplicações através da internet com a mesma facilidade de tê-las instaladas em nossos próprios computadores.

Habitou-se a utilizar aplicações instaladas em nossos próprios computadores, assim como a armazenar arquivos e dados dos mais variados tipos neles. No ambiente corporativo esse cenário é um pouco diferente, já que nele é mais fácil encontrar aplicações disponíveis em servidores e que podem ser acessadas por qualquer terminal com permissão através de uma rede.

A principal vantagem desse modelo está no fato de que é possível, pelo menos na maioria das vezes, utilizarmos as aplicações mesmo sem acesso à internet ou à rede. Em outras palavras, é possível utilizar esses recursos de maneira off-line. Entretanto, todos os dados gerados estarão restritos a esse computador, exceto quando compartilhados em rede, coisa que não é muito comum no ambiente doméstico. Mesmo no ambiente corporativo, isso pode gerar algumas limitações, como a necessidade de se ter uma licença de um determinado software para cada computador, por exemplo.

A evolução constante da tecnologia computacional e das telecomunicações está fazendo com que o acesso à internet se torne cada vez mais amplo e cada vez mais rápido. Em países desenvolvidos, como Japão, Alemanha e Estados Unidos, é possível ter acesso rápido à internet pagando muito pouco. Esse cenário cria a situação perfeita para a popularização do Cloud Computing, embora esse conceito esteja se tornando conhecido no mundo todo, inclusive no Brasil.

Com o Cloud Computing, muitos aplicativos dos usuários, assim como seus arquivos e dados relacionados, não precisam mais estar instalados ou armazenados em seu computador. Elas ficam disponíveis na "nuvem", isto é, na internet. Ao fornecedor da aplicação cabem todas as tarefas de desenvolvimento, armazenamento, manutenção, atualização, backup, escalonamento, etc. O usuário não precisa se preocupar com nada disso, apenas com acessar e usar.

Um exemplo prático dessa nova realidade é o Google Docs, serviço onde os usuários podem editar textos, fazer planilhas e elaborar apresentações de slides pela internet, sem necessidade de ter programas como o Microsoft Office ou OpenOffice.org instalados em suas máquinas. Tudo o que os usuários precisam fazer é abrir o navegador de internet e acessar o endereço do Google Docs para começar a trabalhar, não importa qual seja o seu sistema operacional ou o computador utilizado. Neste caso, o que ele precisa é apenas de um navegador de internet compatível, o que é o caso da maioria dos browsers da atualidade.

### **2.2.1 Características**

Conforme já dito, uma das vantagens do Cloud Computing é a possibilidade de utilizar aplicações diretamente da internet, sem que estas estejam instaladas no computador do usuário. Mas, há outras significativas vantagens:

- Na maioria dos casos, o usuário pode acessar determinadas aplicações independente do seu sistema operacional ou de hardware;
- O usuário não precisa se preocupar com a estrutura para executar a aplicação: hardware, procedimentos de backup, controle de segurança, manutenção, entre outros, ficam a cargo do fornecedor do serviço;
- Compartilhamento de dados e trabalho colaborativo se tornam mais fáceis, uma vez que todos os usuários acessam as aplicações e os dados do mesmo lugar (a nuvem). Além disso, muitas aplicações nas nuvens já são feitas considerando essas possibilidades;
- Dependendo do fornecedor, o usuário pode contar com alta disponibilidade, já que se, por exemplo, um servidor parar de funcionar, os demais que fazem parte da estrutura continuam a oferecer o serviço;

- O usuário conta com melhor controle de gastos. Muitas aplicações em *Cloud Computing* são gratuitas e, quando é necessário pagar, o usuário só o fará em relação aos recursos que usar ou ao tempo de utilização. Não será necessário pagar por uma licença integral de uso, tal como acontece no modelo tradicional de fornecimento de software.

Note que, independente da aplicação, com o *Cloud Computing* o usuário não precisa conhecer toda a estrutura que há por trás, ou seja, ele não precisa saber quantos servidores executam determinada aplicação, quais as configurações de hardware utilizadas, como o escalonamento é feito, onde está a localização física do datacenter, enfim. O que importa ao usuário é saber que a aplicação está disponível na nuvem, isto é, na internet, não importa de que forma.

### **2.2.2 Software como Serviço (SaaS)**

Intimamente ligado ao Cloud Computing está o conceito de Software as a Service (SaaS) ou, Software como Serviço. Em sua essência, trata-se de uma forma de trabalho onde o software é oferecido como serviço, assim, o usuário não precisa adquirir licenças de uso para instalação ou mesmo comprar computadores ou servidores para executá-lo. Nessa modalidade, no máximo paga-se um valor periódico - como se fosse uma assinatura - somente pelos recursos que utilizar e/ou pelo tempo de uso.

Para entender melhor os benefícios do SaaS, suponha que uma empresa que tem vinte funcionários necessita de um software para gerar folhas de pagamento. Há várias soluções prontas para isso no mercado, no entanto, a empresa terá que comprar licenças de uso do software escolhido e, dependendo do caso, até mesmo hardware para executá-lo. Em muitos casos, o preço da licença ou mesmo dos equipamentos pode gerar um custo alto e não compatível com a condição de porte pequeno da empresa.

Se, por outro lado, a empresa encontrar uma fornecedora de software para folhas de pagamento que trabalha com o modelo SaaS, a situação pode ficar mais fácil: essa fornecedora pode, por exemplo, oferecer esse serviço através de Cloud Computing e cobrar apenas pelo número de usuários e/ou pelo tempo de uso.

Dessa forma, a empresa interessada paga um valor baixo pelo uso da aplicação. Além disso, hardware, instalação, atualização, manutenção, entre outros, ficam por conta do fornecedor. Também é importante levar em conta que o intervalo entre a contratação do serviço e o início de sua

utilização é extremamente baixo, o que não aconteceria se o software tivesse que ser instalado nos computadores da empresa-cliente. Esta última só precisa se preocupar com o acesso ao serviço (no caso, uma conexão à internet) ou, se necessário, com a simples instalação de algum plug-in no navegador de internet de suas máquinas.

### **2.2.3 IaaS – Infraestrutura como Serviço**

Infraestrutura como serviço (IaaS) oferece infraestrutura de computação, normalmente um ambiente de virtualização de plataforma como um serviço. Em vez de comprar servidores, software, espaço do centro de dados ou equipamentos de rede, os clientes compram esses recursos como um serviço totalmente terceirizado. O serviço normalmente é cobrado sobre uma base de computação utilitária e a quantidade dos recursos consumidos e; portanto, o custo normalmente reflete o nível de atividade. É uma evolução de ofertas de servidor virtual privado.

### **2.2.4 PaaS – Plataforma como Serviço**

Serviços de plataforma Cloud ou "plataforma de serviço (PaaS)" fornecem uma plataforma de computação como um serviço, muitas vezes o consumo da infraestrutura e manutenção de aplicações em Cloud. Isso facilita o desenvolvimento de aplicações sem o custo e a complexidade de compra e gestão do hardware subjacente e camada de software.

### **2.2.5 Exemplos de aplicações em Cloud Computing**

Os termos Cloud Computing e Computação nas Nuvens são recentes, mas se analisando bem, veremos que a ideia não é, necessariamente, nova. Serviços de webmail como Gmail e Yahoo! Mail, discos virtuais na internet, sites de armazenamento e compartilhamento de fotos ou vídeos como Flickr e YouTube, são exemplos de aplicações que, de certa forma, contém o conceito de Cloud Computing. Note que todos esses serviços não executam no computador do usuário e este pode acessá-los de qualquer lugar, sem necessidade de instalar aplicativos em sua máquina ou de pagar licença de software. No máximo, paga-se um valor periódico pelo uso do serviço.

- *Google Apps*: esse é um pacote de serviços que o Google oferece que contém aplicativos de edição de texto, planilhas e apresentações

(Google Docs), serviço de agenda (Google Agenda), comunicador instantâneo integrado (Google Talk), e-mail com o domínio da empresa, entre outros. Todos esses serviços são processados pelo Google e o cliente só precisa criar as contas do usuário. O Google Apps oferece pacotes gratuitos e pagos, de acordo com o número de usuários. Um dos maiores clientes do Google Apps é a Procter & Gamble, que contratou os serviços para mais de 130 mil colaboradores;

- *Amazon*: é um dos maiores serviços de comércio eletrônico do mundo. Para suportar o volume de vendas no período de Natal, a empresa montou uma superestrutura de processamento e armazenamento de dados, que acaba ficando ociosa na maior parte do ano. Foi aí que a empresa teve a ideia de "alugar" esses recursos, com serviços como o *Simple Storage Solution* (S3), para armazenamento de dados, e Elastic Compute Cloud (EC2), para uso de máquinas virtuais.

- *Live Mesh*: esse é um serviço da Microsoft ainda em estágio inicial. Sua proposta principal é a de permitir que o usuário acesse o seu desktop de qualquer computador, com a diferença de que todos os seus arquivos ficam na nuvem, isto é, no servidores da Microsoft;

- *Datasul By You*: a brasileira Datasul (agora integrada à TOTVS), dispõe de um conjunto de soluções de ERP chamado *By You* que utiliza os conceitos de Cloud Computing e SaaS;

- *Aprex*: também brasileiro, oferece um conjunto de ferramentas para uso profissional, como calendário, gerenciador de contatos, lista de tarefas, disco virtual, blog, serviço de e-mail marketing, apresentações, entre outros. Tudo é feito pela Web e, no caso de empresas, é possível até mesmo inserir logotipo e alterar o padrão de cores das páginas. Há opções de contas gratuitas e pagas.

### 2.3 SLA – Service Level Agreement

O SLA é um acordo entre provedor de serviço e seu cliente, determinando a qualidade mínima de serviço aceitável pelo cliente (HILES, 2000).

Os acordos do nível de serviço (SLA) foram inicialmente desenvolvidos para o fornecimento de serviços internos das empresas, sendo que os pioneiros foram os serviços de informática. A seguir, foram utilizados para a regulamentação da conectividade de rede entre grandes provedores do mesmo nível e, posteriormente, nos serviços externos, onde passaram a ser tratados por meio de contratos, com as especificações de níveis de serviços requeridas pelo cliente.

O SLA pode ser considerado em duas partes distintas: uma, contendo as informações que não dependem de particularidades dos serviços, tais como o módulo de autenticação de usuários, as informações sobre disponibilidade e confiabilidade do serviço, a privacidade e os aspectos de segurança. A outra, contendo dados específicos do serviço, tais como a tecnologia de rede utilizada, o protocolo de transporte, a forma de atendimento no acesso dentro do cliente, o gerenciamento e as formas de acesso ao gerenciamento da rede a ser fornecida (ANTONIO, 2004).

### 2.3.1 Universo do SLA

A Figura. 2.3.1 visualiza o universo dos referidos SLAs (BEST, 2002) dentro da cadeia de valor de fornecimento de um serviço, onde o alinhamento entre os três SLAs permite que sejam satisfeitas todas as partes envolvidas no modelo do negócio, quais sejam:

- O cliente, aquele que compra ou contrata um determinado serviço.
- O provedor de serviço, aquele que está fornecendo o serviço ou produto, o qual pode estar associado a outros parceiros a fim de prover o que foi solicitado pelo cliente.
- A rede, o meio pelo qual o serviço está sendo prestado do fornecedor ao cliente. Ou neste caso a infraestrutura, no caso a nuvem.
- O negócio, que é o elo que envolve as duas partes, cliente e fornecedor.

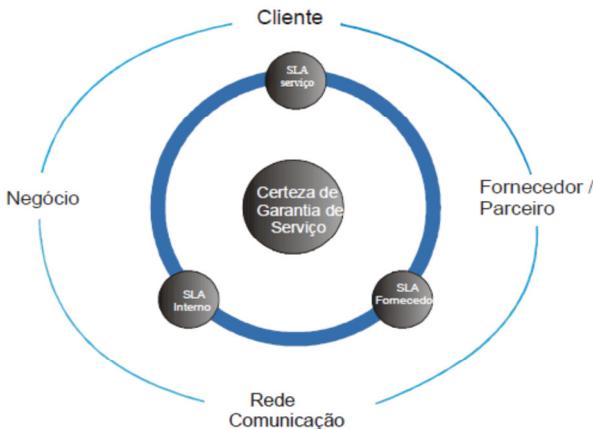


Figura 2.3.1 - Universo do SLA.  
FONTE: BEST (2002).

### **2.3.2 SLA interno**

É chamado de SLA interno todo e qualquer acordo de nível de serviço que abrange os limites interno de uma unidade de negócio ou empresa. Em muitos casos, a área que está geralmente sob foco é a dos SLAs internos.

Os SLAs internos são frequentemente focados no gerenciamento de componentes da cadeia de entrega do serviço, sendo que estes componentes são definidos em função da necessidade da operação interna do negócio, por exemplo: disponibilidade interna de ativos da rede, disponibilidade interna dos servidores de aplicativos, disponibilidade operacional dos aplicativos, critérios de segurança no acesso físico e lógico dos sistemas, listas, dados. Assim, são agrupados os componentes da cadeia de entrega do serviço para modelar do início ao fim um serviço mensurável por meio de indicadores definidos a fim de se obter a qualidade solicitada pelo usuário interno.

Por sua natureza, os SLAs internos não são direcionados aos clientes externos, portanto não estão adequados a termos compreensíveis pelo cliente final. Os principais beneficiados deste tipo de SLA são as funções básicas voltadas para o cliente interno, que são responsáveis pelo gerenciamento dos componentes, ou seja, a função da disponibilidade, segurança e desempenho de serviço em comparação com os acordados no contrato de qualidade do serviço prestado.

### **2.3.3 SLA do serviço**

O SLA do produto é que proporcionará a confiança do cliente no SLA interno para uma melhor e eficiente condução do serviço prestado e também para entendimento do desempenho da entrega do serviço dentro das funções do negócio. Percebe-se que o acordo de nível de serviço não está restrito à rede de dados, mas também se estende aos componentes de serviço, além de ser aplicável aos serviços fora da rede, por exemplo, o billing.

### **2.3.4 SLA fornecedor**

Os clientes tornam-se cada vez mais e mais confiantes na terceirização de serviços e conteúdo, e a implantação de SLAs subsidia os fornecedores/parceiros, que representarão papel importante no modelo de negócio. De fato, a implementação de SLA nesta área melhora a qualidade do serviço ou conteúdo entregue, fato este que

permitirá ao cliente repassar ao terceirizado algum risco financeiro devido à degradação do serviço.

Estes SLAs são muito similares ao modelo do SLA interno discutido acima, mas pode também ter impacto dentro do processo contábil para pagamento, ou seja, existirão penalidades caso o fornecedor não cumpra o acordado, principalmente no que tange à prestação de serviços de conteúdo.

### **2.3.5 Relação entre o SLA e o Plano de Negócios**

Para que seja fornecido um SLA adequado, devem-se identificar quais são os fatores críticos de sucesso no modelo de negócio do cliente e de que forma o SLA proposto sobre o produto ou serviço contratado atenderá às condições do plano de negócio (business plan) da empresa.

O alinhamento adequado de um SLA às necessidades do plano estratégico da empresa é a chave do sucesso no fornecimento de um produto ou serviço com elevado nível de qualidade.

### **2.3.6 Componentes de um SLA**

Os componentes de um SLA destacam-se na especificação dos serviços ofertados e nos seus respectivos níveis de qualidade, conforme o exemplo demonstrado na Tabela 2.3.6. Dessa forma, o contrato deverá especificar níveis de serviço que atendam às necessidades do usuário, concomitantemente aos limites técnicos e comerciais do fornecedor do serviço.

Um SLA deve conter parâmetros objetivos e mensuráveis, os quais o fornecedor dos serviços se compromete a atender junto ao cliente interno ou externo.

O contrato de SLA tem por objetivo resguardar as partes envolvidas, ou seja, fornecedor e cliente. Busca garantir que o serviço seja realizado de forma adequada e compatível com o que foi previamente acordado.

Tabela 2.3.6 - Principais componentes de um SLA

<b>Item</b>	<b>Descrição</b>
Descrição do ambiente	Contem informações que descrevem por que existe a necessidade de criação do SLA e quais vantagens ele irá trazer.
Partes	Identifica as partes envolvidas no acordo.
Serviços	Responsável por exibir as particularidades do serviço.
Parâmetros	Especifica em quais níveis os usuários ou clientes podem esperar a prestação do serviço.
Disponibilidade	Apresenta a porcentagem mínima de tempo que o serviço negociado deve manter-se em operação.
Limitações	Coloca as possíveis limitações dos serviços em momentos críticos que independem de quem oferece.
Compensações	Caso o acordo negociado seja quebrado, este item menciona as consequências de cada parte.
Medições	Como o serviço ou parâmetro disponibilizado é monitorado e avaliado.

Uma das premissas de um SLA é a identificação de métricas e a formalização de parâmetros, para que se possa medir o que se fornece, de forma que as premissas acordadas não tenham como objetivo exigir do fornecedor um desempenho que não seja possível ser cumprido por ele.

Um contrato ideal de SLA deve sempre prever ou ser capaz de se adequar a problemas que vão surgindo no dia-a-dia da operação. Para se obter essa condição, sua aplicação deve se dar de forma flexível.

Note que um contrato bem elaborado de SLA deve estabelecer muitas para o fornecedor de serviços, caso ele não cumpra os indicadores de desempenho previamente estabelecidos. Isso, obviamente, desde que a organização/cliente forneça todos os elementos e condições necessárias para que o fornecedor desempenhe suas atividades.

Nos serviços que podem comprometer a disponibilidade da informação, a área que gerencia a segurança precisa envolver-se a fim

de identificar a melhor forma de atenuar e eliminar riscos, por meio do monitoramento dos serviços prestados por terceiros.

### 2.3.7 Processo do SLA

Existem duas partes principais em um SLA: o documento redigido e o processo. O documento de SLA é o elo entre o cliente e o fornecedor de serviço. Esse documento descreve os serviços e os níveis exatos de serviço, com detalhes sobre todos os acordos. O processo de SLA representa os métodos que o fornecedor utilizará como suporte ao documento do SLA. Os métodos que dão suporte ao documento de SLA são deixados geralmente a cargo do fornecedor da solução para sua identificação. Esses processos devem ser identificados e discutidos durante a negociação do contrato de SLA. É importante que as partes compreendam os processos e os métodos de suporte, assim como a gerência e as ferramentas de relatórios.

A gerência de sistema e a tecnologia da automação do serviço interno podem fornecer um ambiente de suporte para seguir, delegar e efetuar o gerenciamento das métricas (SLs2) dos serviços. Os relatórios de satisfação do usuário das pontas podem também fornecer a entrada que ajudará a dirigir níveis adequados de serviço e efetivos controles de custo.

Tipos de SLA:

- **Básico:** É definido como um único acordo de nível de serviço nivelado em uma área específica. As métricas são estabelecidas e mensuradas, requerendo possivelmente a seleção manual de dados para o relatório da gerência. Tem por objetivo justificar o suporte técnico de operação.

Um exemplo desse tipo de acordo é o fornecimento de suporte na operação, ou um SLA interno específico, em que são avaliadas somente as condições de fornecimento do serviço por parte do fornecedor, respeitando sempre os limites estabelecidos pelo acordo firmado, não tendo, como foco principal a satisfação do cliente, e sim o alcance dos parâmetros de suporte da operação e manutenção do serviço.

A vantagem deste tipo de acordo é o fácil gerenciamento em função de esse acordo possuir parâmetros e métricas voltadas para a área de fornecimento do suporte de operação do serviço

A desvantagem é que, requerendo-se a seleção manual de dados para o relatório da gerência, pode haver demora na identificação dos problemas e na correção destes, o que prejudica o nível de qualidade do

serviço prestado. Também podem ocorrer erros nas coletas de dados, o que gera um mascaramento do relatório e que provoca a insatisfação do usuário final.

- **Médio:** É definido como acordo de múltiplos níveis de serviço em função do custo, conforme a característica de determinado serviço. Pode-se chamar este de SLA estático. Tem o objetivo de combinar serviço e níveis de custo, visando, a longo prazo, ao aumento dos níveis de serviço e à redução dos custos com o passar do tempo.

Um exemplo desse tipo de acordo é a determinação de SLAs diferenciados por classes de serviços específicos dentro de uma unidade de negócio.

A vantagem desse tipo de acordo é a automatização das métricas de dados, o que permite mais detalhes e trabalho menos intenso na realização de relatório, obtendo-se melhor gerenciamento do serviço prestado e uma visão da unidade de negócio em relação ao SLA aplicado.

A desvantagem é que não possui provisionamento de forma automática de seus recursos internos, e geralmente o valor depende do nível de serviço exigido pelo cliente, o que pode se tornar proibitivo financeiramente. O gerenciamento do serviço envolve somente cliente e fornecedor, não se observando as mudanças na condição do negócio.

- **Avançado:** É definido como acordo de nível de serviço dinâmico (SLA dinâmico). Os níveis de serviço são encaixados em processos totais no serviço interno, permitindo a alocação dinâmica de recursos, tanto externos como internos, para encontrar mudança nas condições do negócio.

Tem por objetivo fornecer, sem emendas, uma mescla dos serviços, dos custos e dos fornecedores de serviço, oferecendo melhores valores na relação custo benefício. Um exemplo desse tipo de acordo é a integração total do modelo de negócio com sua cadeia de valor.

As vantagens desse tipo de acordo são que, frequentemente, as empresas que atingem um determinado nível de maturidade estão prontas para estender serviços a sua cadeia de valor e, conseqüentemente, têm maior competitividade e uma visão macro do gerenciamento e do planejamento do negócio.

As desvantagens são que, para a implementação desse tipo de SLA, todo modelo de negócio deverá estar com os processos alinhados com sua cadeia de valor, o que demanda grande esforço e maturidade. O gerenciamento deste SLA é mais complexo e necessita um controle mais rígido; na maioria dos casos, existe integração entre diferentes

plataformas e diferentes sistemas, principalmente quando envolve sistemas legados, o que gera grande dificuldade.

Na Figura 2.3.7, tem-se um comparativo entre as classificações dos níveis de serviço supracitadas e pode-se constatar que o custo do serviço é inversamente proporcional ao nível de risco de paralisação do serviço fornecido.

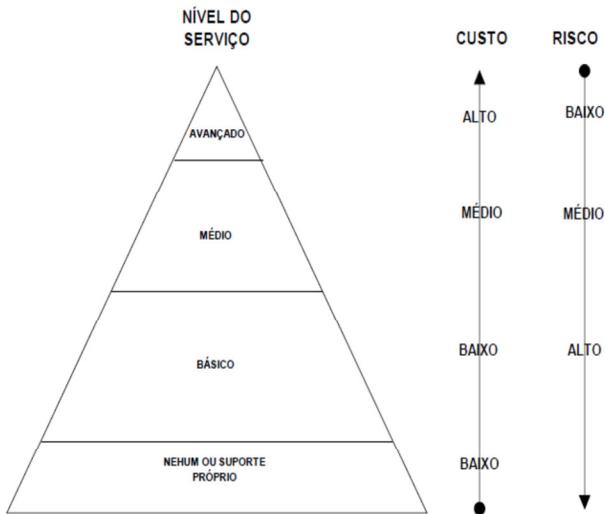


FIGURA 2.3.7 - Pirâmide modelo do nível de serviço  
 FONTE: HILES (2000).

## 2.4 Trabalhos relacionados

Os analistas industriais fizeram projeções de tendência sobre como a *Cloud* transformará a indústria da computação. De acordo com uma recente da pesquisa de Merrill Lynch (Hamilton 2008), *Cloud* é esperada ser “uma oportunidade de mercado de \$160- bilhões, incluindo \$95- bilhões em aplicações do negócio e de produtividade, e outro de \$65-bilhões na publicidade online”.

Outro estudo da pesquisa, desenvolvido pela Morgan Stanley (Morgan 2008) igualmente identificou *Cloud* como uma das tendências de tecnologia proeminentes. Enquanto a indústria de computação

contribui para o fornecimento de plataforma como um serviço (PaaS) e o software como um serviço (SaaS) para que consumidores e as empresas acessem por demanda independente de tempo e localização; então haverá um aumento no número de plataformas de *Cloud* disponíveis.

Recentemente, diversas organizações acadêmicas e industriais começaram o desenvolvimento de tecnologias e de infraestrutura para a computação em Nuvem. Os esforços acadêmicos incluem os espaços de trabalho virtuais (Keahey 2005) e o OpenNebula (Llorente 2008).

*Amazon Elastic Compute Cloud* (EC2) (Amazon 2009) fornece um ambiente de computação virtual que permite um usuário de utilizar aplicações baseada em Linux. O usuário pode criar uma nova *Amazon Machine Image* (AMI) contendo aplicações, bibliotecas, dados e os ajustes associados da configuração, ou seleções de uma biblioteca de AMIs disponível globalmente. O usuário precisa então fazer o *upload* dos arquivos AMIs no *Amazon Simple Storage Service* (S3), antes que possa começar, parar e monitorar instancias dos AMIs. A Amazon EC2 cobra o usuário pelo tempo em que a instancia está rodando, enquanto o Amazon S3 cobra por transferência de dados (*download* ou *upload*).

*Google App* (Google 2009) permite que um usuário execute aplicações Web escritas utilizando a linguagem de programação Python. Além de suportar a biblioteca do Python, o Google Apps também suporta API's para *datastore*, contas do Google, URL *fetch*, manipulação de imagens e serviços de e-mail. O Google App igualmente fornece um suporte de administração Web, para que o usuário controle facilmente suas aplicações Web. Atualmente, o Google App está livre para ser usado com até 500MB de armazenamento e tem aproximadamente 5 milhões de acesso à página por mês.

A *Microsoft Live Mesh* (Microsoft 2009) fornece uma solução integrada para acesso a aplicações e dados através dos dispositivos (tais como computadores e telefones celulares) em qualquer lugar do mundo. O usuário pode acessar as aplicações e os dados com um *Dekstop* com suporte a internet ou seus próprios dispositivos que possuam software *Live Mesh* instalado. Cada usuário possui uma senha e uma autenticação através de uma sessão *Windows Live Login*, com todas as transferências protegidas pelo *Secure Socket Layers* (SSL).

*Sun Network.com* (Sun grid) (Sun 2009) permite ao usuário rodar Solaris, Java, C, C++, e aplicações baseadas em FORTRAN. Primeiramente, o usuário tem que construir e eliminar erros das suas aplicações e *runtime scripts* em um ambiente de desenvolvimento local que seja configurado para ser similar a Sun Grid. Então, é preciso criar

um arquivo ZIP (que contem todos os certificados relacionados, bibliotecas, binários executáveis e dados de entrada) e transferir à Sun Grid. Finalmente, pode executar e monitorar a aplicação usando o portal web da Sun ou a API. Após a conclusão da aplicação, o usuário precisará transferir os resultados da execução a seu ambiente de desenvolvimento local para a visualização dos resultados.

*GRIDS Lab Aneka* (Chu 2007), está sendo comercializado pela Manjrasoft, é uma plataforma orientada a serviço, *NET-based* para construir Grids. Ela é projetada para suportar modelos da aplicação múltipla, soluções de persistência e de segurança, protocolos de comunicação tais que a seleção preferida pode ser mudada a qualquer hora sem afetar um sistema Aneka. Para criar uma Nuvem na empresa, o fornecedor de serviço precisa somente instanciar um container Aneka configurável que hospeda serviços exigidos em cada computador selecionado. A finalidade de Aneka é inicializar serviços e atuar como um único ponto para a interação com o resto da empresa. Aneka fornece SLA visto que o usuário define parâmetros de QoS, por exemplo: *Deadline* ( período de tempo máximo que a aplicação precisa para estar completa ) e o orçamento ( preço máximo que o usuário está disposto a pagar para respeitar o deadline ). O usuário pode acessar o Aneka remotamente através do *GridBus Broker*. O *Gridbus broker* permite ao usuário negociar e concordar com as exigências de QoS feitas ao fornecedor de serviços.

Podemos ver abaixo que nenhum dos projetos pesquisados oferece soluções para SLA de maneira dinâmica, exceção ao Aneka, porém é feito de maneira estática uma vez que os parâmetros são pré-definidos e não há possibilidade mudança sob demanda. Visando fechar esse GAP na literatura, este trabalho é propõe uma abordagem para obter acordo de nível de serviço de maneira dinâmica, durante a execução e que seja sob demanda, ou seja, mude de acordo com as necessidades do usuário.

Tabela 2.4: Comparativo entre diferentes plataformas para Cloud.

<b>Sistemas</b>	<b>Amazon EC2)</b>	<b>Google APP</b>	<b>Microsoft Live Mesh</b>	<b>Sun Network.com (Sun Grid)</b>	<b>GRIDS Lab Aneka</b>
<b>Propriedades</b>					
<b>Foco</b>	Infraestrutura	Plataforma	Infraestrutura	Infraestrutura	Plataforma Software Nuvem
<b>Tipo Serviço</b>	Computação Armazenamento	Aplicação web	Armazenamento	Computação	Computação
<b>Virtualização</b>	Execução em nível de SO ou Xen	Container de aplicações	Nível de SO	Sistema de gerenciamento de JOBS	Gerenciar e escalar recursos
<b>Negociação dinâmica de parâmetros de QoS</b>	Não suporta	Não suporta	Não suporta	Não suporta	Reserva de recursos de SLA
<b>Acesso a Interface</b>	Utilitário de linha de comando Amazon EC2	Console de administração Web	Desktop ou qualquer dispositivo com acesso Live Mesh	Submissão de tarefas através do portal Sun Grid	Portal web
<b>Web APIs</b>	Sim	Sim	Desconhecido	Sim	Sim
<b>Value-added Service</b>	Sim	Não	Não	Sim	Não
<b>Linguagem</b>	Amazon Machine Image (AMI)	Python	Não aplicável	Solaris OS, Java, C, C++, FORTRAN	C# e .NET

### 3. PROPOSTA

Antigamente, serviços (sistemas) rodavam *in-house* e eram desenvolvidos por companhias individuais. Entretanto, com *outsourcing* e equipes multidisciplinares se tornando populares, serviços agora estão sendo providos por provedores terceirizados de boa reputação, como a Amazon e Microsoft, assim as companhias podem se concentrar nas suas competências principais e não serem atrasados por operações secundárias de escritório. Enquanto as companhias migram para essa arquitetura orientada a serviços (SOA), a qualidade e confiabilidade dos serviços que são providos pelos provedores é um ponto chave, já que os processos críticos de negócio serão suportados por esses sistemas 24/7 (24 horas por dia, 7 dias por semana). O SLA, um contrato formal negociado que define as obrigações (tipicamente as obrigações do provedor de manter um nível garantido de serviço), serve como fundação para assegurar um entendimento comum das expectativas do serviço entre o consumidor (companhia) e o provedor de serviço.

Nos últimos anos, com Cloud Computing emergindo a computação teve seu cenário de arquiteturas de serviço modificado. Cloud Computing é baseado na visão de prover serviços como utilidades (e.g. água, luz), onde consumidores podem acessar os serviços em qualquer lugar do mundo e, por demanda, pagar apenas pela quantidade que consomem. O tamanho, complexidade e escopo desse paradigma computacional faz com que surja a necessidade dos provedores de serviço de implementar uma forma mais escalável para negociar, orquestrar e automatizar o processo de SLA.

#### 3.1 Negociação de SLA

Negociação é o processo onde o provedor do serviço e o consumidor discutem, debatem e concordam com as expectativas do fornecimento do serviço. Os consumidores do serviço irão requerer uma Qualidade de Serviço (QoS) específica, por exemplo, consumidores com operações críticas irão necessitar um tempo de resposta mais rápido e suas expectativas podem mudar com o tempo devido a mudanças contínuas no ambiente. Provedores de Serviço devem assegurar essa flexibilidade na sua entrega de serviço para satisfazer melhor seu consumidor, e podem fazer isso permitindo SLA's customizadas e dinâmicas sempre levando em consideração fatores como lucratividade e factibilidade. Os provedores devem também adotar uma estratégia de

preço flexível que diferencia os serviços baseado nos seus requerimentos para prover incentivos financeiros tanto para o consumidor como para eles mesmos (Kankanhalli 2009).

Assim, muitos pesquisadores já propuseram diferentes protocolos e estratégias para suportar negociação automatizada, como *Web Service Level Agreement*(WSLA), *Web Services Agreement Specification*(WS-Agreement), SLang etc. Essas ferramentas criam uma linguagem comum que facilita a comunicação automatizada entre o provedor e o consumidor, mas coloca uma desvantagem que é a necessidade das duas partes entenderem uma linguagem comum. O consumidor pode querer obter serviços de vários provedores e, caso os provedores empreguem protocolos diferentes, isso faria com que o consumidor tivesse que adotar protocolos separados para cada provedor. O middleware de meta negociação (*Meta-Negotiation Middleware MNM*) (Buyya 2008) apresenta uma solução em potencial para o problema, criando um 'mercado público' de serviço para todos os consumidores.

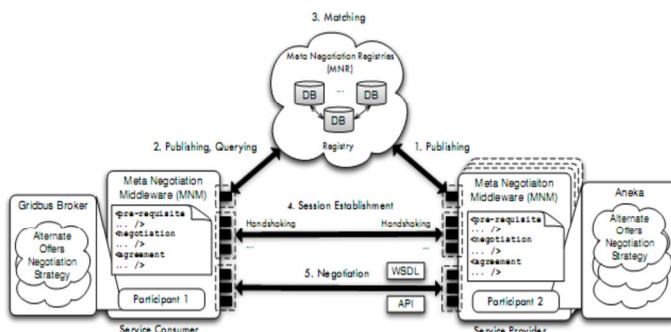


Figura 3.1. Middleware de Meta Negociação.

FONTE: Buyya (2008).

Gridbus Broker e Aneka são ferramentas usadas pelo consumidor e para o provedor respectivamente para determinar a estratégia de negociação, que é a lógica usada por um parceiro para decidir qual provedor ou consumidor satisfaz melhor suas necessidades. O processo pode começar com os provedores de serviço publicando seus pré-requisitos, que serão traduzidos em um formato comum e exibidos em um registro. Quando um consumidor deseja obter um serviço em particular, ele submete seu pedido de serviço para o registro, e o registro irá retornar uma lista de documentos publicados por provedores que satisfazem as necessidades do consumidor. O consumidor, então,

seleciona o serviço apropriado baseado na sua estratégia de negociação e uma sessão de negociação é iniciada e pode ser estabelecida entre as duas partes.

Mesmo que desejável, o estado-da-arte atual em Cloud Computing, entretanto, deve amadurecer para tal estágio. De fato, a maioria dos provedores de aplicações em cloud computing, como Amazon, Google e IBM não suporta nem a negociação dinâmica de SLAs, ou seja, todos consumidores adotam uma SLA pré determinada pelo provedor.

### 3.2 Orquestração

Segundo (Kankanhalli 2009) orquestração é o arranjo, a coordenação e gerenciamento de sistemas de computador complexos, middleware e serviços. Em Cloud Computing precisa-se de um sistema que faça a negociação de SLAs com o mínimo de intervenção humana. Precisa-se disso pelo fato de que o serviço será oferecido para consumidores do mundo inteiro que podem estar precisando do serviço simultaneamente pela internet. Entretanto, como foi mencionado, ainda não há suporte para negociação de SLAs dinâmicas para provedores de Cloud Computing. Ademais, cada companhia é única; e é difícil, ou até mesmo impossível, ter uma SLA que satisfaz os requerimentos e necessidades de todas as companhias.

Negociações dinâmicas não são suportadas devido a razões, como a falta de padrões, leis e regulamentações diferentes em vários países e as muitas indústrias diferentes nas quais as companhias podem estar. Para estabelecer o sistema de negociação de SLAs dinâmicas, provedores de cloud computing deveriam ter uma base de dados relativamente grande em cada indústria. Entretanto, apesar de que cloud computing tenha apenas ganhado o interesse das companhias recentemente, muitas ainda não sentem confiança o bastante para deixar seus negócios na cloud, pelo fato de cloud computing ser imatura e não ter uma SLA bem definida.

Provedores devem primeiramente fazer um sistema de negociação de SLA dinâmico limitado, que irá focar no serviço ao consumidor. O consumidor deve poder escolher uma variedade aceitável de um conjunto de métricas de SLA pré-definidas, como tempo de resposta, *uptime*, segurança, etc. O consumidor iria ter que informar alguns detalhes como o preço que ele poderá pagar por terabyte por segundo, ou seja, seu custo x benefício aceitável que contemple seus requisitos. Depois o sistema deveria poder combinar as preferências do consumidor

e retornar algumas SLA que atenderiam aos requisitos do consumidor (Kankanhalli 2009).

O consumidor poderia, então, escolher aceitar uma das SLA ou escolher alguns atributos a mais e continuar a refinar e combinar resultados até que uma SLA o satisfaça. Nos bastidores o sistema iria checar a história de crédito do consumidor na sua base de dados, na base de dados de outras nuvens e bancos. Também seriam checados contratos e recurso disponível para saber se é possível prover os recursos desejados.

Os recursos, então, seriam colocados como reserva. Após isso, iria ser checada uma lista de preços e o preço seria ajustado (preço máximo que o consumidor deseja pagar para assegurar o nível de serviço que é garantido durante os horários de pico) de acordo com descontos e retornaria as SLAs recomendadas ao consumidor. Depois de selecionar a SLA, os recursos reservados da SLA seriam alocados ao consumidor e o restante seria disponibilizado de volta ao sistema. Isso é principalmente para a configuração inicial da conta do consumidor. Consumidores com demandas especiais também poderiam contratar o serviço ao consumidor do provedor através de vídeo conferência e e-mails para negociar melhor os termos contratuais (Kankanhalli 2009).

Consumidores deveriam ser segmentados em regiões, como Ásia, Europa ou Estados Unidos, para prevenção de casos inesperados. Deste modo, servidores para provedores deveriam estar em várias regiões para assegurar que os termos e condições ainda sejam cumpridos mesmo nesse tipo de situação. O sistema deveria poder detectar uma queda no tempo de resposta e ajustar a alocação de recursos de acordo com a necessidade, redirecionando o pedido ou abrindo e fechando servidores em diferentes regiões de acordo com a situação.

Como Computação em Nuvem trata o uso de TI como uma utilidade, onde as pessoas pagam pelo uso é difícil determinar como será o uso de um consumidor novo nos recursos. Como mencionado anteriormente o preço em uma SLA é um indicador de quanto o consumidor aceita pagar para manter o nível de serviço durante o horário de pico de uso. Como o serviço seria desperdiçado quando não usado nos horários fora do pico de uso, o preço desses horários deveria ser mais baixo que os horários de pico. Para atrair mais usuários uma aplicação deveria ser provida para rodar serviços de baixa prioridade durante os horários que não são de pico, mas como o serviço da *cloud* é oferecido no mundo todo, é impossível para um humano determinar quando são os horários de pico, levantando a necessidade da automação. Os horários fora de pico seriam determinados pelo sistema de *cloud*,

quando a atividade atingisse uma determinada baixa e os serviços de baixa prioridade rodassem. Entretanto, determinar qual serviço rodaria primeiro seria um desafio. Isso ajudaria a garantir os requerimentos de SLA, enquanto minimiza o gasto de recursos para manter certo nível de serviço. Além disso, os servidores deveriam ficar *online* e *off-line* na cloud, de acordo com a atividade e capacidade da *cloud* de garantir as demandas de SLA (Kankanhalli 2009).

Quando estiver monitorando os requerimentos de SLA, o sistema deve poder automaticamente compensar os consumidores caso alguma violação ocorra. Compensação poderia ocorrer na forma de descontos. Entretanto, violações, como o vazamento de dados do consumidor, são extremamente difíceis de medir a compensação necessária. Dessa maneira, deveria haver uma autoridade independente da cloud computing no futuro, pois as leis e regulamentações de cada país são diferentes e podem favorecer seus cidadãos. Então, com uma autoridade independente para monitorar e resolver conflitos asseguraria igualdade de tratamento para ambas às partes.

Computação em Nuvem é provido como utilidade, logo é realmente difícil oferecer um bom preço para novos consumidores. Um modelo de pagar menos por unidade quando o consumo aumenta poderia ser implementado pelos provedores, já que não haveria como determinar quanto de recursos o consumidor iria necessitar antes de um período prolongado de uso, no qual deveria ser feito mineração de dados para inferir a quantidade necessária de recursos. Quanto mais companhias adotam cloud computing, mais informações de consumo serão usadas para detectar padrões de uso que servirão para ajudar a fazer a orquestração em uma SLA dinâmica. Os provedores de cloud computing terão que trabalhar juntos para fazer o sistema de SLAs dinâmicas possível.

### 3.3 Desafios na automação de Acordo de Nível de Serviço

(Kankanhalli 2009) enumera alguns dos principais desafios em automação de SLA em Nuvem: Para clientes de cloud computing, na competição por recursos do ambiente de negócios do mundo real, os requerimentos de QoS podem mudar com o tempo, já que há repetitivas mudanças nas operações de negócios e ambientes. Uma abordagem dinâmica para negociação de SLA se mostra importante nesse cenário. Entretanto a tecnologia corrente não possui (ou possui suporte limitado) para negociação dinâmica de SLA para serviços de cloud. Isso trás um

risco para os consumidores conforme foi explicitado na seção anterior. Há desafios para implementar as soluções sugeridas na seção anterior.

Primeiramente, não há registro de serviços para os provedores de cloud, onde eles possam publicar seus *Service Level Objectives* (SLO) para os diferentes níveis de serviço que eles oferecem. Para um registro funcionar, um protocolo comum deve ser desenvolvido e utilizado por todos os provedores de serviço. Entretanto, diferentemente do caso dos *web services*, no qual os servidores adotaram uma arquitetura orientada a serviços (SOA), os provedores de cloud implementaram suas arquiteturas de cloud próprias para poder atender seus clientes específicos. O fato de haver muitas arquiteturas diferentes limita a possibilidade do cliente de trocar de um provedor de cloud para outro, ou com baixo impacto na migração e nos custos de migração. Com um sistema de registro global para o mercado de *cloud computing* onde os consumidores podem selecionar e trocar de provedor que oferecem o mesmo serviço parece redundante nesse caso.

Em segundo lugar, para possibilitar uma negociação de SLA automática e dinâmica há uma falta de ferramentas para localizar e verificar a disponibilidade de recursos e capacidade que atendem a demanda do consumidor. Para a Nuvem, diferentes arquiteturas de aplicação e protocolos podem ser implementados em diferentes infraestruturas virtuais que estão numa mesma infraestrutura física. Para determinar se o provedor possui os recursos disponíveis e capacidade necessária para atender às expectativas do consumidor e evitar uma sobre-alocação, novos mecanismos automatizados devem ser capazes de prever e reservar recursos e capacidade para usuários reais em tempo real.

Em terceiro lugar, há uma falta de ferramentas para avaliar e viabilizar os pedidos de serviço. Há parâmetros críticos de QoS que devem ser considerados e atendidos para cada consumidor individualmente, conforme negociado nas suas SLAs específicas. Mecanismos para dar o preço de maneira flexível e dinâmica para os diferentes parâmetros de QoS requisitados também são necessários para competir com outros provedores. Como há um volume consideravelmente grande de pedidos diariamente é necessário avaliar quais pedidos são mais rentáveis e quais não são. Dessa maneira, é importante determinar a importância de um novo pedido de serviço e decidir aceitar ou recusar contra propostas automaticamente.

### 3.4 Mecanismos e algoritmos para Alocação de Recursos

A arquitetura de *cloud computing* difere de outras pelo uso de máquinas virtuais para criar várias infraestruturas virtuais a partir de uma única infraestrutura física. Como cada VM pode se configurar e designar um percentual de recursos físicos, isso levanta vários desafios como a alocação inteligente de recursos para gerenciar a competição de uso dos recursos pelos consumidores. Uma vez que as SLAs foram negociadas e aceitas por ambas as partes, mecanismos também são necessários para fazer com que as VMs correspondentes sejam configuradas para atender os requerimentos de SLA automaticamente. A tecnologia corrente não possui suporte para esse uso, pois as ferramentas existentes são incapazes de gerenciar o planejamento de capacidade para esse novo paradigma de *cloud computing*. Para atender os picos e vales de demanda nos diferentes níveis de requerimento de serviços de diferentes consumidores, novos mecanismos e algoritmos devem ser desenvolvidos para monitorar os recursos e requerimentos de capacidade e depois balancear e configurar as VMs em tempo real (Kankanhalli 2009).

As diferenças de arquitetura precisam de novas ferramentas para fazer o monitoramento em tempo real de todas as diferentes VMs e rastrear métricas de performance, máximos, mínimos e detectar anomalias. Ferramentas também precisam possibilitar que serviços confiáveis de auditoria possam monitorar externamente e assegurar que as SLAs estão sendo cumpridas durante todo o tempo. Quando anomalias são detectadas, o sistema deve ser capaz de resolver os problemas automaticamente. Em casos onde o sistema não consegue resolver o problema sozinho, deve disparar um alarme apropriado para que as medidas necessárias sejam tomadas. Para minimizar o *downtime* do serviço, novas ferramentas devem ser desenvolvidas para que seja fácil replicar, migrar ou mudar de VM automaticamente de uma infraestrutura física com problema para outra infraestrutura física que não apresente problemas, para o caso de falhas físicas na infraestrutura (como queda de energia, incêndios, etc.).

O paradigma de *cloud computing* ainda está na sua infância e apresenta muitos desafios, como descritos acima, que precisam ser resolvidos. Como o estado da arte corrente ainda não consegue cumprilos, os provedores de serviço de *cloud computing* atuais estão adotando SLAs pré-definidas e simples, que não dão muitas garantias para os

consumidores. Abaixo apresentaremos uma arquitetura que busca suprir esse GAP na literatura e resolver os problemas apresentados acima.

### 3.5 Arquitetura

*Service-Oriented Architecture* (SOA) é uma arquitetura que combina elementos de arquitetura de software e arquitetura empresarial. É baseado na interação com os serviços autônomos e interoperáveis que oferecem funcionalidade de negócio reutilizáveis através de interfaces padronizadas. Os serviços podem existir em todas as camadas de uma aplicação de sistema de processo (empresas, apresentação, lógica de negócio, gestão de dados). Eles podem ser compostos de serviços de camadas inferiores, componentes embarcados ou serem implementados a partir do zero. Normalmente, os serviços na camada de processo de negócio são descritos como *business services*, enquanto os serviços no menor nível são descritos como serviços técnicos (Stanchev 2009).

Uma arquitetura orientada a serviço tem dois objetivos principais: prover a funcionalidade necessária e prover a Qualidade de Serviço (QoS). Parâmetros de QoS são parte do *runtime* relacionado com propriedades não-funcionais (NFPs) de um serviço e apresenta um dos desafios de pesquisa principais em computação direcionada a serviço. Ao contrário de *design-time* NFPs (e.g., linguagem de serviço ou complacência), a execução de NFPs é orientada a desempenho (e.g., tempo de resposta, taxa de transação, disponibilidade). Eles podem mudar durante a execução quando se tem uso simultâneo; hora extenso por muitos usuários e em seguida, tempos de uso raro, ou quando falhas acontecem (Stanchev 2009).

Uma proposta para medir e dinamicamente adaptar desempenho de serviço em grade e ambientes de computação de nuvem e assegurar níveis de serviço definida ao nível de negócio. Isto é uma tarefa mais desafiadora em tal infraestrutura que não são definidas ou controladas diretamente pela organização. Especificamente, tal abordagem deveria considerar reconfiguração de serviço na execução, devido a mudanças na implementação de serviço não ser realístico.

São especificados NFPs de serviços (técnico e humano, como também as combinações deles) tipicamente em Acordo de Nível de Serviço (SLAs). Eles estão tipicamente definidos ao nível de negócio (*business level*), mas precisam ser resolvidos no nível de infraestrutura de TI. Assim são orquestrados vários serviços técnicos para prover *business services* para um *business process*. São negociados SLAs entre

o dono do processo e o provedor de serviço que têm que concordar neles (Stanchev 2009).

Este trabalho propõe um modo dinâmico para negociar SLAs entre um cliente e um provedor de serviço, garantindo o cumprimento deste SLAs ao nível de grade e infraestruturas de computação de nuvem, formalizando ambos os requerimentos de nível de serviço do cliente e a capacidade dos serviços da grade ou nuvem usando uma estrutura similar consegue-se compará-los de uma maneira automática. Baseado nessas comparações consegue-se prover serviços de configuração otimizados na nuvem e através disso garantir os SLAs dos processos de negócio em características de Qos e níveis de serviço desses serviços técnicos.

### 3.6 Uma abordagem para mapeamento de SLA dinamicamente

Quando uma companhia controla a sua estrutura de Ti, analistas de projetos e desenvolvedores podem definir os requisitos de nível de serviço durante o design e selecionar e definir componentes para suprir esses requisitos. Entretanto em ambientes de computação em nuvens, SLA são geralmente providos por plataforma de serviços básicos (e.g.: system uptime. Throughput...). Processos de negócio usualmente esperam níveis de serviço dos serviços que eles integram (e.g. Envio de ordens em menos de 1 seg.). Como trazer de encontro esses dois mundos?

Neste trabalho propõe-se uma abordagem para mapeamento de SLA entre processos de negócio e infraestrutura de TI. É baseado em um método que garante as NFPs e inclui três tarefas principais:

- **Formalização** de requerimentos de processos de negócio do lado do negócio e das capacidades do serviço na infraestrutura de TI. Ambas especificadas de uma maneira formal, usando uma estrutura pré-definida de SLO e termos de NFP predefinidos.
- **Negociação** de capacidades de serviço na infraestrutura de TI que correspondem aos requerimentos de processo de negócio formais: Aqui, estima-se se os serviços técnicos agregados provêm os níveis de serviço esperados para corresponder aos requerimentos de processo de negócio sob diferentes cargas hipotéticas. Nesta comparação calcula-se o nível de serviço agregado, usando métricas de performance de serviços técnicos individuais. Baseado no resultado dessas comparações pode-se decidir onde aplicar a replicação no próximo passo. Uma unidade de comparação deve entender tanto a estrutura das declarações e NFPs usadas no negócio e na infraestrutura.

- **Aplicação** das SLAs dos processos de negócio no nível da infraestrutura de TI: paralelização translúcida de processamento de serviços usando múltiplos nodos em um ambiente de datacenter. Replicação pode ser estabelecida para aprimorar os níveis de serviço referentes a tempo de resposta, nível de transação, *throughput*, disponibilidade e confiabilidade.

Os principais aspectos da arquitetura de SOA (Service Oriented Architecture), tais como a visualização do serviço, a integração, e a seleção do serviço foram endereçadas consistentemente por pesquisadores. Os padrões existentes para a especificação de características de QoS em Web Services nos ambientes podem ser agrupados de acordo com seu principal foco: projeto de software (Um perfil UML para QoS e QML - QoS Moderador de linguagem), a prestação de serviços de manutenção padrão (WS) e SLAcentric aproximações (WSLA - acordos de nível de serviço, WSOL - Web Service Offerings Language, Slang - definição do acordo do nível de serviço de linguagem língua e acordo WS) (Stanchev 2009).

Muitos trabalhos foram desenvolvidos na área de descoberta de web services que se preocupavam com QoS (Makripoulis 2005) Plataformas e Middlewares que se preocupam com QoS (Yau 2003, Zeng 2004, Canfora 2005, Soldberd 2004) e serviços que se preocupam com o contexto (Tokairin 2007). Entretanto, todas essas abordagens não atendem aplicação adaptativa de NFPs, mas lidam com a composição de serviços onde a agregação de níveis de NFPs pré definidos que satisfazem um requerimento específico. De particular interesse são abordagens que permitem a 'gridificação' de aplicações específicas, e.g. em (Abdennadher 2007) onde um conjunto de programas para inferir árvores evolucionárias é portada para uma plataforma em grid XtremWeb-CH (Abdennadher 2007b). Abordagens como memória compartilhada também já foram propostas para tais tarefas (Ibach 2006). Há uma pesquisa acontecendo na área de otimização adaptativa, mais especificamente na área de grids (e.g., otimização de workflow de grids (Wanek 2008)) e operações paralelas de banco de dados (Schikuta 2008).

### 3.7 Arquitetura de um modelo de negociação dinâmica de SLA

Para um modelo de negociação dinâmica de SLA funcionar, um protocolo único deve ser estabelecido entre provedores e clientes da cloud. Antes, porém, deve ser estabelecida uma arquitetura padrão para ser seguida. Essa arquitetura deixará explícito o modelo de uma SLA

para que os participantes possam negociar sem haver problemas de comunicação. Para que esse modelo seja feito é uma boa ideia que se parta de uma modelagem minimalista e que o modelo vá evoluindo com o uso ao longo do tempo.

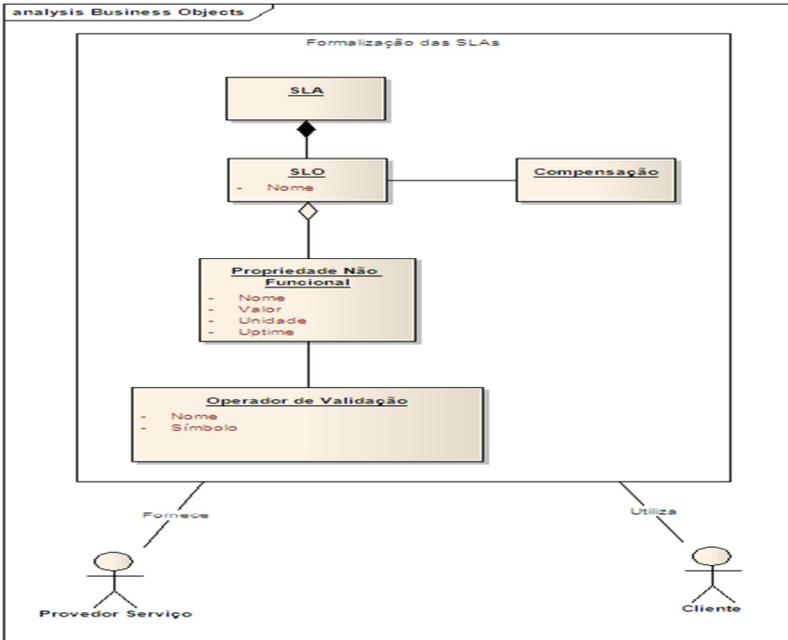


Figura 3.7a Modelagem de um SLA

Na figura 3.7a é demonstrada uma modelagem da definição de uma SLA. Um SLA seria composto por várias SLO (*service layer objective*). Cada SLO define a performance de uma característica única do serviço necessária para atingir um determinado objetivo. Exemplos de SLO seriam: tempo necessário para resolução de problemas, tempo de resposta de aplicação e disponibilidade do sistema. Cada SLO teria uma compensação financeira caso o serviço não consiga cumprir sua responsabilidade contratual no fornecimento do serviço.

Cada SLO, por sua vez, é composto por várias propriedades não funcionais que definem no nível mais baixo a qualidade mínima aceitável do serviço para uma determinada característica. Cada propriedade não funcional deverá ter um operador de validação para ser utilizada em cima de seu valor, toda vez que o operador de validação não conseguir validar o valor da propriedade significa que o serviço

falhou em prover aquela propriedade não funcional, acarretando na falha de uma SLO. Por exemplo, uma propriedade não funcional pode ser 'Tempo de Resposta', com o operador 'Menor que' e um valor de 90 e unidade milissegundos. Quando o tempo de resposta atingir valores maiores que 90 ms essa propriedade não funcional será considerada como não cumprida acionando a compensação financeira devida ao cliente. Há também o 'Uptime' da propriedade, que é o percentual de tempo no dia que aquela propriedade é garantida. Um 'Uptime' de 100 significa que a propriedade não funcional não pode ser descumprida nunca. Quando Uptime é menor que 100, como por exemplo, 99, significa que o provedor pode descumprir a propriedade não funcional 1% do tempo sem acionar a compensação financeira.

Com essa arquitetura bem estabelecida seria fácil definir um modelo de prospecção de provedores tendo uma SLA mínima requerida em mãos para fazer uma busca online em um servidor centralizado que buscaria serviços com suas respectivas SLAs e valores.

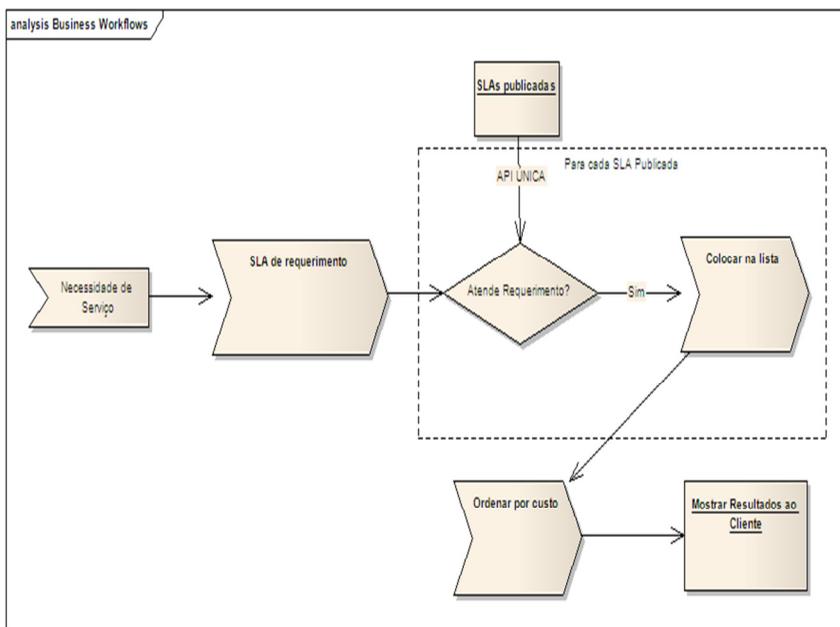


Figura 3.7b Fluxograma de análise de negócio.

Havendo uma necessidade de serviço o cliente colocaria sua SLA de requerimento no sistema que iria prospectar dentre todas as SLAs

publicadas, e retornaria ordenadamente as mais vantajosas para o cliente que atendem a SLA mínima.

No serviço de publicação a comunicação deveria ser feita de uma maneira que todos compreendessem, para isso seria perfeito o uso de um XML padrão para uniformizar a comunicação entre os participantes.

```

<!ELEMENT SLA (SLO)>
<!ELEMENT SLO (Name, NfoList, Compensation)>
<!ELEMENT Name (#PCDATA)>
<!ELEMENT NfoList (Nfo)>
<!ELEMENT Nfo (Name, Value, Metric, Uptime)>
<!ELEMENT Name (#PCDATA)>
<!ELEMENT Value (#PCDATA)>
<!ELEMENT Metric (#PCDATA)>
<!ELEMENT Uptime (#PCDATA)>
<!ELEMENT Compensation (#PCDATA)>

```

Figura 3.7c: DTD para comunicação entre os participantes da negociação.

Na figura acima (3.7c) está descrito o DTD que seria utilizado para validar os XMLs que serviriam para realizar a comunicação entre os clientes e provedores. Um XML seria a escolha ideal, pois poderia ser utilizado tanto para comunicação via Web Services quanto RCP quanto várias soluções de comunicação SOAM.

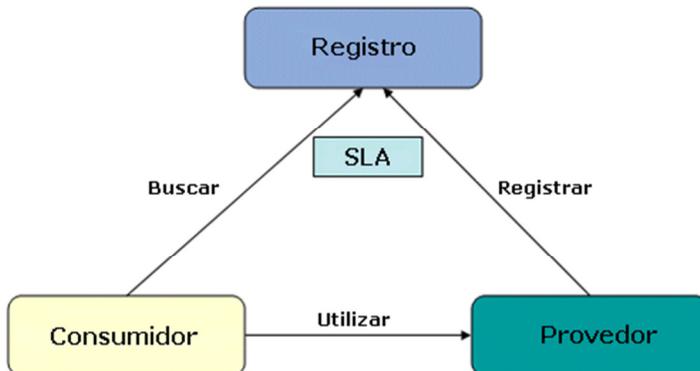


Figura 3.7d. Relação entre consumidor e provedor de SLA.

Tendo essa arquitetura em mente o ideal seria uma autoridade reguladora para aplicá-la, assim como a W3C regulamenta os padrões da Web deveria haver uma autoridade de igual porte para padronizar os serviços de Cloud, o que seria bom tanto para clientes quanto para provedores. Os clientes teriam a segurança de estar escolhendo os melhores serviços com o melhor custo para seus negócios. Os provedores teriam seus serviços ofertados de maneira mais dinâmica e conseguiriam atingir um público maior, apenas se esforçando em ter melhores preços e oferecer um melhor serviço e economizando no marketing.

### 3.8 Formalização e negociação de SLA

A tabela 3.8 mostra uma estrutura proposta recentemente para a formalização de negócios do nível de serviço do processo (SLOs) e do serviço técnico de capacidades. As figuras também contêm indicações do nível de serviço e amostras sobre o tempo de resposta, *throughput* e de taxa transação. Estas indicações são armazenadas então com a descrição do serviço (de acordo com as capacidades de serviço) respectivamente com a definição do negócio, a definição de processo (processo de negócio SLO) e é o ponto de partida para a negociação de SLAs. Um exemplo para uma indicação sobre a capacidade de serviço é "a taxa de transação do serviço, que é mais elevada por 90s das transações em 98% dos casos constatado que o *throughput* for mais elevada de 500 kB/s." Um exemplo para o nível do processo de negócio da exigência é "A taxa de transação do processo deve ser mais elevada por 50s das transações em 97% dos casos quando *throughput* for mais elevada que 500 kB/s." Neste trabalho trata-se a negociação e a aplicação de performance NFPs como parte de SLAs e consequentemente foco na sua formalização.

Entretanto, nosso esquema pode igualmente ser usado para descrever uns aspectos adicionais de SLAs, por exemplo, o tempo de desempenho relacionado ao NFPs, tais como cultural (e.g. língua), legal (e.g. conformidade SOX, Basileia-II), de organização, usabilidade (e.g. simplificação da GUI) e do relacionamento de confiança (por exemplo, avaliação de cliente, experiência do fornecedor).

Tabela 3.8 Estrutura dos Objetivos de Nível de Serviço.

E x e m p l o s  D e S L O	NFP	Predicado	Métrica	%	Condições		
					NFP	Predica do	Métrica
	Tempo Respos ta	Menor que	100 ms	95%	Taxa Transfer ência	Menor que	10 tps
	Throug hput	Maior que	1000Kb/ s	95%	Taxa Transfer ência	Menor que	10 tps
	Taxa Transfe rência	Maior que	90 tps	98%	Throug hput	Maior que	500 kB/s

Fonte: STANCHEV (2009).

Em nossa abordagem, o dono do processo faz as exigências do nível de serviço esperado no nível de negócio. E nessa infraestrutura são avaliadas as diferentes configurações de serviços técnicos em grids e ambientes de computação em nuvem, tais como o *Amazon EC2*. Ambas as exigências e capacidades de serviço com configurações diferentes então são formalizadas e comparadas. Além disso, podemos começar uma negociação entre o proprietário do processo e o serviço do fornecedor baseado nesta comparação. Desse modo, a configuração que melhor supre as exigências do negócio é selecionada e é o ponto de partida para SLA. O modelo de custo transparente de uma Open Cloud Computing permite a aplicação de preços diferentes para os níveis de serviço exigido (e.g. um processo de negócio que precise mais transações exige que se pague mais), assim permite atividade baseando-se nos reais custos de TI. Além disso, pode-se utilizar a estrutura do SLO em uma maneira de *supply-oriented*, contrariamente a *demand-oriented*. Desse modo, pode-se especificar combinações de níveis do NFP que representam os SLAs genéricos diferentes (e.g. *Gold, Standard, Cost-optimized, Time-Critical, Load-Critical, Dependability-Critical*).

### 3.9 Aplicação de QoS de SLA em Nuvens

A fim de satisfazer os SLAs de um processo de negócio devem-se olhar maneiras de representar e controlar NFPs no nível de estrutura TI. Quando a computação especifica os ambientes Clouds, os níveis de

serviço para operações básicas da plataforma (e.g. *uptime*), concentram-se sobre a melhoria de níveis de serviço para serviços técnicos específicos, isso é, compor para fornecer os serviços a empresas que as necessitam. Um exemplo é a composição dos serviços técnicos *GetOrder* e *ClearPayment* que forneça a colocação da ordem dos serviços a empresas (Stanchev 2009).

### 3.10 Desempenho

Uma definição geral e amplamente aceita de desempenho deve observar o sistema output o  $\omega(\delta)$  que representa o número de pedidos com transações completas de um total de pedidos da entrada ( $\delta$ ) durante um período de tempo.

$$\omega(\delta) = f(u(\delta)) \quad (\text{Stanchev 2009})$$

Essa definição de desempenho corresponde à taxa de transação como o NFP - garantia do sistema processar N pedidos durante o período de tempo T. O desempenho de um serviço é composto por uma série de outros processos e é determinada pelo desempenho do processo com mais baixo desempenho.

Supondo que uma rede forneça o serviço 1, o serviço 2 e o serviço 3. Se o serviço 1 e o serviço 3 estão fornecendo altas a taxa de transação (e.g., 500 requisições por segundo) e o serviço 2 está fornecendo uma mais baixa taxa de transação (e.g. 50 por segundo), o serviço compõe somente 50 transações (ou realmente menos de 50 requisições) por segundo total. Pode-se facilmente calcular o tempo de resposta médio da taxa de transação, dividindo o período de tempo pelo número de pedidos. Além disso, pode-se igualmente mensurar a média do desempenho, tal como o verificar as hipóteses de tempo de execução (WCET) utilizando especificações do SLA.

Quando o paralelismo é aplicado através da réplica funcional, pode-se idealmente dobrar o desempenho de processamento. A réplica do serviço com a mais baixa taxa de transação (serviço 2) conduz a um aumento total da taxa de transação para compor o serviço. Consequentemente, a réplica tem efeitos vantajosos no desempenho do serviço, quando nenhuma sincronização da réplica for exigida. Isto inclui taxa de transação, throughput e tempo de resposta como parâmetros de SLAs.

### 3.11 Fidelidade

A fidelidade possui diversos atributos: disponibilidade, confiabilidade, segurança, integridade e sustentabilidade. Estes são definidos como segue:

- Disponibilidade da prontidão para proporcionar um serviço correto,
- Confiabilidade para a continuidade da provisão de serviço,
- Segurança é um atributo que impede que haja alguma consequência catastrófica ligada ao usuário e ao ambiente,
- Integridade que não há nenhuma mudança imprópria do sistema.
- Sustentabilidade de um sistema pode-se submeter a mudanças e a reparos.

Há quatro categorias de aproximações para alcançar a confiança: prevenção da falha, tolerância de falha, remoção da falha, e previsão da falha (Stanchev 2009).

No contexto deste trabalho centramos na disponibilidade como o atributo e sobre a tolerância de falha como a aproximação para alcançá-la. A disponibilidade precisa de ser determinada no SLAs de modo que se possa negociar e reforçá-la. Foi definida tradicionalmente como uma métrica binária que descreve se um sistema está "ativo" ou "inativo" em um único momento. Uma extensão comum desta definição é computar a porcentagem média do tempo que um sistema é disponível durante um determinado período - esta é uma medida típica da disponibilidade, descrever um sistema como tendo a disponibilidade 99,999%, por exemplo. Há diversas definições de disponibilidade que levam a inerentes limitações da definição tradicional - a disponibilidade deve ser considerada como um espectro, em vez de uma métrica binária, como os sistemas que podem estar degradados, mas operacionalmente em estados entre o "*up*" e "*down*" (Stanchev 2009). Além disso, a definição não considera aspectos de QoS.

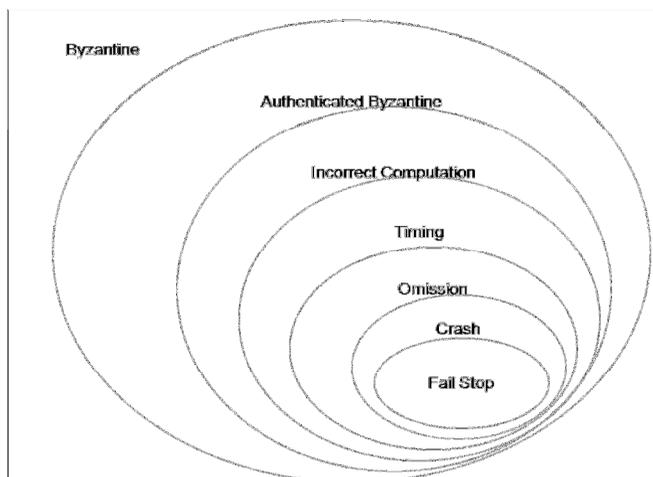


Figura 3.11 Modelo de falha usado em uma estrutura de SLO.  
Fonte: POLZE (2000).

Uma possibilidade de mensurar a disponibilidade é examinando variações no medidor do sistema QoS sobre o tempo. Os autores indicam que a escolha particular do medidor de QoS depende do tipo de sistema e sugere o desempenho e o grau de tolerância de falha para sistemas de usuário. Neste caso, o desempenho significaria os pedidos satisfeitos. Isto corresponde a nossa definição do desempenho na subseção precedente. A fim de especificar graus de tolerância de falha precisa-se um modelo subjacente da falha. É possível utilizar um modelo em que seja usado igualmente (veja Figura 3.11). Incorpora diversos aspectos que são típicos para serviços técnicos em ambientes de computação em Nuvem, em comparação aos sistemas tradicionais (e.g. edições da confiança). Uma adaptação recente de métodos tradicionais para a melhor disponibilidade é propor dentro do SOA que envolve a réplica de serviços técnicos através de múltiplos locais. Tipicamente, precisa-se fornecer a consistência forte entre as réplicas a fim de fornecer a melhor disponibilidade. Isto faz a abordagem problemática, particularmente em ambientes de computação de nuvem - as despesas gerais que introduzimos para se assegurar de que a consistência tenha geralmente um impacto negativo no desempenho.

Há diferentes maneiras onde se pode replicar serviços técnicos em ambientes de computação de nuvem. O conceito de arquitetura de abstração define três níveis para a replicação em plataformas de SOA (hardware, sistema operacional, e serviceware) e propor técnicas e

mecanismos de replicação para a avaliação dos seus efeitos nos NFPs. Usando tais conceitos pode-se avaliar diferentes réplicas a nível de infraestrutura de TI, formalizando os resultados destas avaliações como SLOs, e selecionando a configuração que melhor supre as exigências do processo de negócio.

### 3.12 Avaliação Experimental

Visando comprovar as ideias apresentadas no trabalhado foram construídas algumas simulações utilizando o CloudAnalyst (WICKREMASINGHE 2009), modificando alguns de seus parâmetros para termos negociação dinâmica. O CloudAnalyst é uma plataforma livre que simula um ambiente em nuvem com vários parâmetros e considerações sobre um ambiente real que podem ser modificados, como: Banda entre regiões, horários de picos, taxa transferência e entre outras variâncias. O seu código foi levemente modificado para suportar um configurador mais amigável. A modificação para a saída de dados de um arquivo único para cada tipo de interação (dinâmica ou estática) em formato CSV, facilitando a manipulação dos dados.

Primeiro passo foi definir os arquivos SLO com os parâmetros que seriam considerados na simulação. Foram escolhidos tempo de resposta e uptime, porque ambos podem demonstrar exigências interessantes em um ambiente que deve garantir performance x estabilidade.

O segundo passo foi configurar o ambiente da simulação, para que refletisse da melhor maneira o ambiente global e atual da demanda por serviços de computação em nuvem. Foi definido 5 regiões cada uma com a sua largura de banda, horários de pico, taxa de transferência, etc.. Esse paradigma foi usado uma vez, que hoje em dia encontra-se diversos provedores de nuvem espalhados pelo globo.

O terceiro passo foi definir o cenário: a fim de recriar um ambiente global com várias bases de usuários espalhados pelo globo, servidores distribuídos da mesma maneira. A interação pode ocorrer entre todos, de maneira a cumprir a maior % de SLA e otimizando ao máximo os recursos de cada servidor. Esse paradigma que diz respeito a utilizar o maior recurso possível da estrutura e otimizar os custos, promove a ideia de requerer menos recursos no futuro, que vai de encontro a um dos novos desafios da computação em nuvem: A chamada computação verde, onde os recursos utilizados, a quantidade de energia gasta, e o lixo tecnológico devem ser o menor possível.

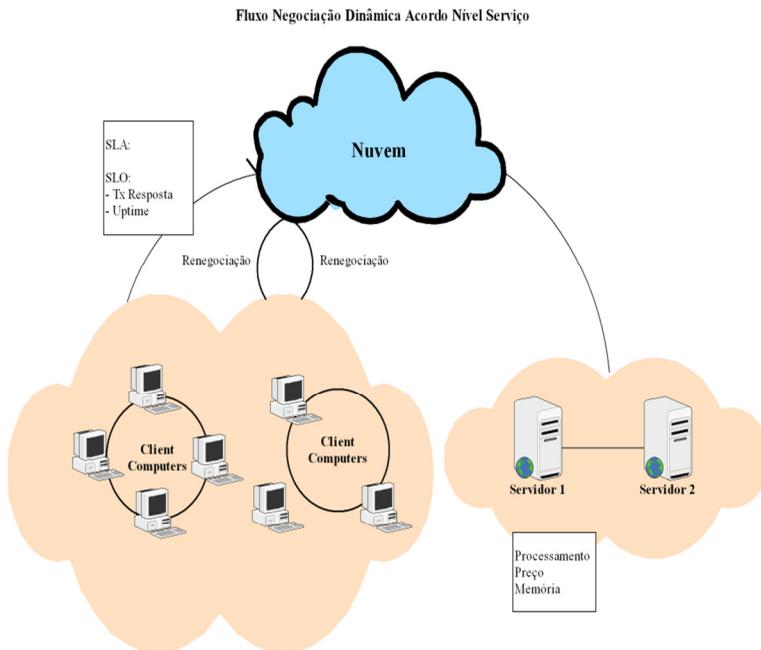


Figura 3.12a Modelo de negociação dinâmica simulada.

O passo seguinte foi a definição da base de usuários, cada base pertence a uma das 5 regiões do globo, possui SLO composto por Tempo de Resposta medido em *ms*, medição da % de *Uptime* para atender a esse tempo de resposta. Os servidores são compostos dos seguintes atributos: processamento, preço da CPU, preço da memória e também são divididos em regiões.

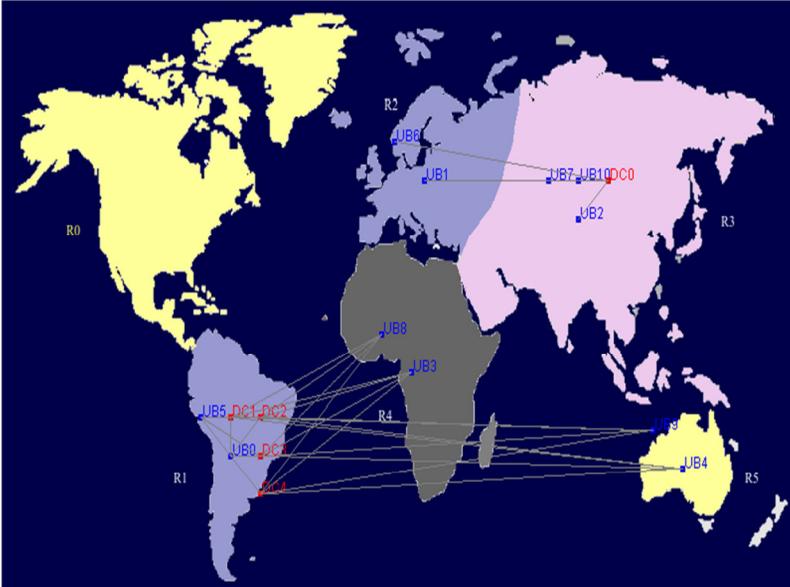


Figura 3.12b Exemplo de execução: Comunicação entre diversas bases de usuários (UB) e servidores (DC) alocados no globo.

Fonte: WICKREMASINGHE (2009).

Abaixo, têm-se as tabelas com as configurações dos clientes e dos servidores:

Tabela 3.12a Configuração dos clientes no ambiente

Nome	Base de Usuários	Tempo Resposta (ms)	Uptime(%)	Região
Cliente 0	1	60	97	1
Cliente 1	2	60	98	2
Cliente 2	3	60	94	3
Cliente 3	4	60	99	4
Cliente 5	5	60	94	5

Tabela 3.12b. Configuração dos servidores no ambiente

Nome	Processamento	Preço CPU	Preço Memória	Região
Servidor 0	10000	0,10	0,05	1
Servidor 1	12000	0,12	0,05	2
Servidor 2	11000	0,11	0,05	3
Servidor 3	14000	0,14	0,05	4
Servidor 4	14000	0,14	0,05	5

A proposta do trabalho é uma abordagem dinâmica de SLA. Para provar a eficiência dela em relação ao método tradicional, ou do estático, precisamos entender de que forma cada um deles se comporta no sistema:

O método estático descreve uma interação entre os clientes com os servidores de nuvem sem qualquer tipo de negociação/renegociação dinâmica de SLA. Esse método busca simular um usuário decidindo pelo melhor serviço sem ter acesso a nenhum tipo de ajuda automática. O usuário seleciona o melhor servidor para suas necessidades, ele envia a SLA como parâmetro e não sabe se o servidor irá conseguir cumprir suas exigências com o passar do tempo. Por vezes o algoritmo escolhe o servidor mais barato (visando simular um perfil de usuário que escolheria o servidor mais barato), porém muitas vezes este servidor não consegue cumprir a SLA acordada, fazendo com que o percentual de tempo da SLA do cliente seja devidamente atendida baixando drasticamente. O descumprimento de uma única SLO já é o bastante para invalidar o acordo inteiro (SLA), levando em consideração que há um tempo de resposta que deve ser atendido ou um % de atendimento.

O método dinâmico descreve uma interação entre os clientes com os servidores de nuvem com a ajuda de um registro central onde as SLA's podem ser negociadas de maneira dinâmica. Cada servidor tem publicado nesse registro seus preços e quais SLA's consegue cumprir devidamente. O usuário simulado neste cenário busca sempre escolher o servidor que atenderá melhor sua SLA com o melhor preço. Sempre que necessário o usuário pode renegociar a SLA com o servidor e migrar de servidor caso mesmo não atenda sua necessidade, ou o preço esteja fora dos padrões do usuário. O resultado esperado utilizando este método é um cumprimento maior às SLA's uma vez que podemos fugir de algum servidor que esteja sobrecarregado, ou mesmo aproveitar melhor a ociosidade de outros servidores.

No Algoritmo Dinâmico cada servidor calcula sua carga interna (quantidade de usuários sendo atendidos), aloca seus preços dentro da disponibilidade de memória e processamento. Depois disso cada servidor publica as SLA's, que pode atender e a que preços no servidor de serviços.

Cada cliente pede ao registro a lista de servidores, SLA's e preços. Procura na lista por todos servidores que conseguem cumprir sua SLA. Este usuário escolhe o que melhor satisfaz sua necessidade (relação custo/benefício) através de um arquivo SLO, onde requisita alguns parâmetros que devem ser atendidos pelo servidor.

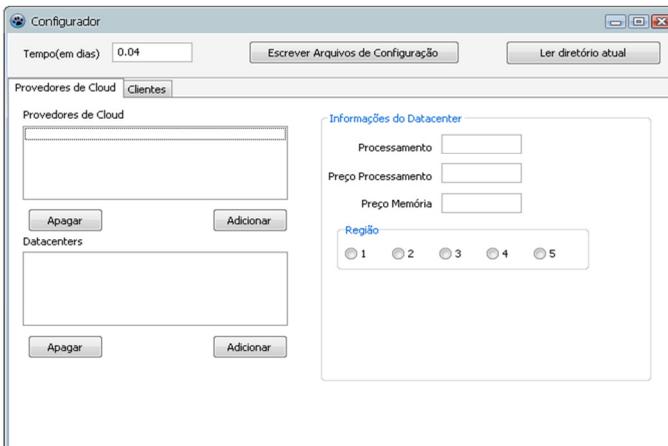


Figura 3.12c Configurador criado para facilitar o ajuste de parâmetros.

A simulação foi dividida em duas partes, uma amostra semanal, visando simular o curto prazo, e semestral visando simular o longo prazo. Em cada simulação foram usadas as mesmas bases de usuários, servidores e configurações de rede das regiões.

Abaixo temos os resultados para duas interações, uma dinâmica e outra estática, simulando uma semana de processamento nesse ambiente. Podemos notar que o custo global na simulação dinâmica foi menor em apenas 2% do que o estático, porém o cliente 4 teve um custo de R\$ 30,58, quase o dobro da média global.

Tabela 3.12c Custo em uma simulação Estática Semanal.

Custo Estático		
Cliente		Médio
0	R\$	16.41
1	R\$	14.83
2	R\$	14.64
3	R\$	15.12
4	R\$	30.58
5	R\$	14.83
<b>Total</b>	<b>R\$</b>	<b>17.74</b>

Tabela 3.12d Custo em uma simulação Dinâmica Semanal.

Custo Dinâmico		
Cliente		Médio
0	R\$	16.41
1	R\$	15.39
2	R\$	16.67
3	R\$	17.81
4	R\$	19.56
5	R\$	18.86
<b>Total</b>	<b>R\$</b>	<b>17.45</b>

Na sequência tem-se a taxa de sucesso na mesma simulação semanal. Nota-se um ganho considerável da aplicação dinâmica em relação à estática, cerca de 7%. Analisando o cliente 4, nota-se que o mesmo apresentou um custo elevadíssimo e a pior taxa de sucesso global. Isto aconteceu porque ele não conseguiu renegociar e procurar outro servidor com menor carga.

Tabela 3.12e Taxa de Sucesso em uma simulação Dinâmica Semanal.

Tx Sucesso Dinâmico (%)	
Cliente	Médio
0	85.43
1	94.01
2	96.31
3	91.43
4	75.05
5	94.12
<b>Total</b>	<b>89.39</b>

Tabela 3.12f Taxa de Sucesso em uma simulação Estática Semanal.

Tx Sucesso Estático (%)	
Cliente	Médio
0	85.31
1	85.95
2	89.15
3	87.70
4	46.10
5	99.11
<b>Total</b>	<b>82.22</b>

Visando comprovar os resultados obtidos, e mostrar uma evolução da negociação dinâmica em relação à estática, simulou-se por 6 meses. Com um maior tempo de simulação procurou-se provar o ganho total ao ambiente que a renegociação gera. Indo de encontro aos anseios atuais, como computação verde, otimização de ambientes, e menor custo possível aos clientes.

Tabela 3.12g Custo em uma simulação Estática Semestral.

Custo Estático	
Cliente	Médio
0	R\$ 674.64
1	R\$ 632.85
2	R\$ 1,917.74
3	R\$ 11,330.08
4	R\$ 613.64
5	R\$ 602.97
<b>Total</b>	<b>R\$ 2,628.65</b>

Tabela 3.12h Custo em uma simulação Dinâmica Semestral.

Custo Dinâmico	
Cliente	Médio
0	R\$ 674.64
1	R\$ 643.66
2	R\$ 663.91
3	R\$ 721.76
4	R\$ 774.60
5	R\$ 763.70
<b>Total</b>	<b>R\$ 707.04</b>

Tabela 3.12i Taxa de Sucesso em uma simulação Estática Semestral.

Tx Sucesso Estático (%)	
Cliente	Médio
0	9.19
1	9.46
2	3.74
3	0.05
4	93.75
5	9.57
<b>Total</b>	<b>20.96</b>

Tabela 3.12j Taxa de Sucesso em uma simulação Dinâmica Semestral.

Tx Sucesso Dinâmico (%)	
Cliente	Médio
0	9.18
1	59.30
2	74.71
3	56.50
4	7.65
5	70.56
<b>Total</b>	<b>46.31</b>

Os resultados apresentados acima podem ser melhor visualizados através do comparativo entre abordagem x período de simulação:

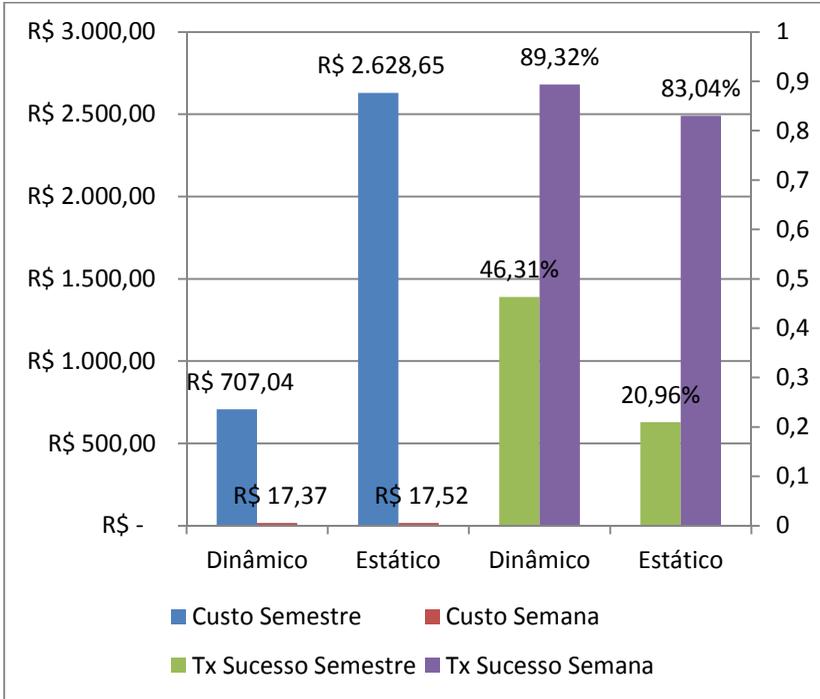


Figura 3.12d Resumo executivo comparando-se: Estática x Dinâmica por período:

Observa-se um ganho maior da abordagem dinâmica em relação à estática, tanto em custo como em taxa de sucesso. Que deixa claro, partindo do princípio que a dinâmica consegue renegociar e encontrar o melhor servidor daquele momento, evitando picos, e tendo um melhor consumo global. O resultado da taxa de sucesso ficou cerca de 120% melhor.

### 3.13 Observações sobre o Capítulo

Apresentou-se uma aproximação para a negociação de SLAs de processos de negócio e a aplicação correspondente de QoS a nível de infraestrutura de TI. Que consiste em três tarefas principais. A formalização dos NFPs permite que sejam comparados os níveis de serviço existentes. Usando um SLA dinâmico podem-se encontrar níveis de serviço automaticamente, reconfigurando réplicas do serviço em ambientes de computação em nuvem. A avaliação experimental semanal

demonstrou-se que seria possível melhorar níveis de serviço em 7% sob cargas semanais e 120% em um período semestral.

Isto nos reserva que os QoS individuais ou SLO suprem às exigências específicas de cada processo de negócio e suporta conceitos como custo x benefício em uma empresa. Argumenta-se também que a abordagem fechou os GAPS da literatura, onde os trabalhos apresentados não conseguiam prover uma solução de SLA que pudessem ao longo do tempo ser renegociadas, melhorando o desempenho global e consequentemente o ganho individual de cada usuário na nuvem.

Essa abordagem se mostrou mais eficiente que a estática em atender os anseios da nova computação, pois consegue gerar um economia de recursos aos servidores, uma vez que reduzimos a ociosidade e sobrecarga de recursos, o usuário por sua vez tem um melhor resultado individual pagando menos por isso. A longo prazo temos a possibilidade de reduzir o parque total de computadores, uma vez que os usuários tendem a contratar serviços de nuvem e pagar por eles somente o necessário, ou pagar pelo uso.

A abordagem provou ir de encontro às demandas do usuário de pagar somente pelo uso, e dar garantias de que os seus requisitos estão sendo atendidos de maneira satisfatória.

## 4. ESTUDO DE CASO

### 4.1 Estudo de Caso – Open Utility Cloud Computing

Um dos atuais modelos de *Cloud Computing* com mais evidência no ambiente corporativo visa à disponibilização de serviços de infraestrutura computacional, envolvendo elementos como CPU, Memória e espaço de armazenamento. Enfim, toda a infraestrutura computacional necessária para a execução de aplicações. Esse cenário é bastante favorável para atender demandas de curta duração de empresas que não desejam investir capital para aquisição de equipamentos.

Os consumidores, ao optarem por utilizar esse tipo de serviço, desejam obter uma infraestrutura computacional em nuvem que seja de fácil integração e utilização, sem que haja a necessidade de grandes modificações no seu próprio ambiente de TI.

O nível de abrangência Open Cloud corresponde à total usabilidade dos recursos por parte dos consumidores. Ou seja, não deverá haver quaisquer restrições ao uso dos recursos disponibilizados em forma de serviço. Essa característica é fundamental para o sucesso desse modelo, pois consumidores que buscam o consumo de Cloud Computing como ambiente de infraestrutura desejam executar suas aplicações sem ter que se preocupar com certas imposições ou limitações tecnológicas da plataforma.

Esse modelo é uma grande oportunidade para o mercado corporativo, uma vez que as empresas buscam alternativas para contornar os altos investimentos em infraestrutura computacional. Os equipamentos adquiridos muitas vezes apresentam capacidade excedente, ou seja, o proprietário não utiliza a total capacidade do produto mesmo tendo pago por isso. Outro grande desafio é a rápida desvalorização dos equipamentos eletrônicos com base na depreciação.

Todas essas características são influências positivas para adoção de um ambiente computacional baseado no modelo de Open Utility Cloud Computing, onde os consumidores não precisam se preocupar com a operação da infraestrutura e pagam apenas pelos serviços utilizados, modelo extremamente positivo para a diminuição do custo total de propriedade (TCO).

Um dos grandes exemplos deste modelo de serviço disponível, baseado em Cloud Computing, é o *Amazon Elastic Compute Cloud* (Amazon EC2), que tem como objetivo fornecer recursos

computacionais como serviços para clientes finais com um rápido e fácil provisionamento.

O objetivo desse estudo de caso é levantar informações sobre o modelo de negócios do Amazon EC2, suas vantagens competitivas, bem como as tendências de adoção deste serviço por inúmeras organizações no mercado corporativo.

#### **4.1.1 Coleta de dados**

Todos os dados foram coletados a partir do portal do sistema do Amazon Elastic Compute Cloud (AMAZON 2009), também denominado Amazon EC2.

#### **4.1.2 Visão Geral**

O serviço da Amazon, denominado Amazon EC2, provê um ambiente computacional virtual para que o consumidor possa executar aplicações baseadas nos sistemas operacionais Linux e Windows. Esse serviço permite ao consumidor total controle sobre os recursos computacionais, possibilitando que o usuário inicie e pare suas instâncias sempre que desejar. O Amazon EC2 também reduz para minutos o tempo necessário para obter novos servidores e iniciar suas instâncias, permitindo assim que os ambientes sejam escaláveis de acordo com mudanças nos requerimentos computacionais (AMAZON 2009).

O formato de cobrança dos serviços baseado no conceito *pay-as-you-go* é uma das características fundamentais para o sucesso do Amazon EC2. Nesse modelo, o usuário é cobrado somente pela capacidade computacional realmente utilizada, ou seja, o consumidor não paga pela capacidade excedente ou pela capacidade ociosa dos seus recursos computacionais.

O Amazon EC2 é atualmente um dos serviços de Open Utility Cloud mais aceito no mercado de Tecnologia da Informação, principalmente por oferecer um ambiente escalável, seguro e de fácil habilitação, características fundamentais para um ambiente de Computação em Nuvem. Devido ao crescente sucesso do EC2, a Amazon lançou uma família completa de serviços para integrar seu ambiente de Cloud Computing, como o *Amazon Simple Storage Service* e *Amazon Simple DB*.

Para utilizar o Amazon EC2, os usuários podem administrar os seus sistemas operacionais, customizar suas aplicações e gerenciar as conexões de redes a partir de uma interface web. Basta apenas criar uma imagem de máquina da Amazon (AMI) contendo suas aplicações, bibliotecas e dados, fazer o upload da mesma para o ambiente do EC2. Os usuários ainda têm a possibilidade de escolher os tipos de sistemas operacionais que queiram utilizar, podem iniciar, monitorar e terminar quantas instâncias de sua AMI forem necessárias utilizando APIs web ou uma variedade de ferramentas disponibilizadas pela própria Amazon.

### 4.1.3 Serviços de destaque

O Amazon EC2 fornece alguns serviços que apresentam uma maior vantagem competitiva e facilidades para seus usuários.

- Elasticidade: permite que o usuário possa aumentar ou diminuir a capacidade de processamento do seu ambiente em apenas alguns minutos. Também é possível comissionar uma, centenas e até mesmo milhares de instâncias simultaneamente.

- Flexibilidade: o usuário pode escolher por múltiplos tipos de instância (configuração de hardware), sistemas operacionais, softwares e pacotes. O Amazon EC2 permite selecionar a configuração de memória, CPU e armazenamento mais adequado para um determinado tipo de sistema operacional e aplicação.

- Integração com outros Web Services da Amazon: trabalha em conjunto com o Amazon Simple Storage Service (S3), Amazon SimpleDB e Amazon Simple Queue Service (SQS) para prover um solução completa de computação.

- Confiança: oferece um ambiente extremamente confiável onde instâncias substitutas podem rapidamente serem realocadas. Buscando um indicador de alta- disponibilidade o serviço especificado no SLA estima 99.95% de disponibilidade.

- Segurança: provê serviços para configurar opções de firewall que controla o acesso de rede entre os grupos de instância.

- Baixo custo: o serviço EC2 fornece os benefícios do ambiente de larga escala de computação da Amazon. O usuário paga uma taxa menor para consumir a mesma capacidade se tivesse que adquirir o próprio equipamento.

#### 4.1.4 Tipos de instâncias

O Amazon EC2 também fornece dois tipos principais de instâncias: Standard e High-CPU. A instância padrão (Standard) está de acordo com a maioria das aplicações existentes, e corresponde a três principais configurações:

- Small: 1.7 GB de memória RAM, 1 unidade de computação EC2 (1 core virtual com 1 unidade de computação EC2), 160 GB de espaço para armazenamento e plataforma de 32-bit.

- Large: 7.5 GB de memória RAM, 4 unidades de computação EC2 (2 cores virtuais com 2 unidades de computação EC2 cada), 850 GB de espaço para armazenamento e plataforma de 64-bit.

- Extra-Large: 15 GB de memória RAM, 8 unidades de computação EC2 (4 cores virtuais com 2 unidades de computação EC2 cada), 1690 GB de espaço para armazenamento e plataforma de 64-bit. Já as instâncias de High-CPU têm mais recursos de CPU do que memória RAM, sendo indicada para uso de aplicações que requerem intensiva computação. Estão disponíveis em duas configurações principais:

- Medium: 1.7 GB de memória RAM, 5 unidades de computação EC2 (2 cores virtuais com 2.5 unidades de computação EC2 cada), 350 GB de espaço para armazenamento e plataforma de 32-bit.

- Extra-Large: 7GB de memória RAM, 20 unidades de computação EC2 (8 cores virtuais com 2.5 unidades de computação EC2 cada), 1690 GB de espaço para armazenamento e plataforma de 64-bit.

#### 4.1.5 Sistemas operacionais e softwares

As imagens de máquinas da Amazon (AMIs) são configuradas com uma lista dos sistemas operacionais que mais cresceram em utilização nos últimos anos. A Tabela 4.1.5a traz os sistemas operacionais disponíveis no Amazon Elastic Compute Cloud:

Tabela 4.1.5a Sistemas Operacionais disponíveis no Amazon

---

Sistema Operacional
Red Hat Enterprise Linux
OpenSolaris
Fedora
Windows Server 2003
OpenSuse Linux
Gentoo Linux
Oracle Enterpris e Linux
Ubuntu Linux
Debian

---

FONTE: AMAZON EC2 (2009).

A Amazon EC2 também disponibiliza alguns softwares que podem ser aplicados à AMIs, tanto pagas quanto livres. Uma amostra dos softwares disponíveis no Amazon EC2 é listado na Tabela 4.1.5b.

Tabela 4.1.5b – Softwares disponíveis

BASE DE DADOS	BATCH PROCESSING	WEB HOSTING	DESENVOLVIMENTO
IBM DB2	Hadoop	Apache HTTP	Java Application Server
IBM Informix D	Condor	IIS/Asp.Net	JBoss Enterprise Application Platform
MySQL	Open MPI	IBMLotus Web Content Management	IBMsMash
Oracle 11g		IBMWebSphere Portal Server	Ruby on Rails
Microsoft SQL			Oracle WebLogic Server

FONTE: AMAZON EC2 (2009).

### 4.1.6 Preços

Os preços da Tabela 4.1.6a referem-se às instâncias on-demand, ou seja, uma configuração que possibilita que o usuário pague, por hora, pela capacidade de processamento, sem a necessidade de acordos de longo prazo. Essas características são positivas para promover a mudança de um cenário de altos custos, com planejamentos complexos, aquisições e manutenção de máquinas, por custos variáveis extremamente menores. Os preços abaixo são referentes à AMIs privadas e públicas para os sistemas operacionais Linux e Windows.

Tabela 4.1.6a – Preços de instâncias On-demand do Amazon

Instâncias		Preços (\$/hora)	
		Linux/UNIX	Windows
Padrão	Small (default)	0,10	0,125
	Large	0,20	0,50
	Extra Large	0,80	1,00
High	Medium	0,20	0,30
CPU	Extra large	0,80	1,20

FONTE: AMAZON EC2 (2009).

Os preços especificados são cobrados por cada hora de consumo de cada instância, desde o momento que é iniciada até o seu término. O consumo referente ao período menor do que uma hora (por exemplo, 45 minutos) é cobrado como uma hora cheia.

A Tabela 4.1.6b está composta com os preços referentes à transferência de dados na Internet.

Tabela 4.1.6b – Preço de transferência de dados na Internet através do Amazon

Tipo de dados	Quantidade	Preço (\$/GB)
Entrada	Todos os dados	0,10
	Primeiros 10 TB por mês	0,17
	Próximos 40 TB por mês	0,13
Saída	Próximos 100 TB por mês	0,11
	Mais de 150 TB por mês	0,10

FONTE: AMAZON EC2 (2009).

#### 4.1.7 Análise

Os benefícios de *Cloud Computing* tornam-se notórios principalmente para as pequenas e médias empresas, as quais muitas vezes não dispõem de recursos para investimentos em tecnologias da informação, como aquisição de hardware e software. Com o conceito do *Open Utility Cloud Computing*, essas organizações podem optar por ter um parque tecnológico hospedado em um provedor de serviço, como a Amazon EC2, sendo necessário pagar somente o utilizado, sem a necessidade de grandes projetos de implementação e sem custos de operação.

A vantagem competitiva de *Cloud Computing* é o seu modelo pay-as-you-go, ou seja, os clientes pagam somente pelos recursos consumidos, situação que se mostra bastante interessante, uma vez que a utilização da capacidade de servidores geralmente circula em torno dos 20%.

Outra característica bastante importante é a facilidade para o provisionamento de um novo serviço. Uma organização quando precisar de um ambiente computacional para o processamento de uma determinada aplicação, basta apenas contratar o serviço via Web (como é o caso do Amazon EC2) e em alguns minutos a capacidade computacional já estará disponível para a utilização.

A questão de operação de uma infraestrutura computacional é algo que também deve ser levado em consideração, principalmente em pequenas e médias empresas, as quais geralmente não têm uma organização de TI para cuidar de tarefas desse propósito. Pode-se exemplificar esse cenário citando uma organização de pequeno porte especializada em prestação de serviços de recursos humanos e que não tem uma estrutura para administrar e operar o seu ambiente de tecnologia da informação no padrão 24/7, ou seja, 24 horas durante todos os dias da semana.

Mesmo que haja organização de TI na empresa, para garantir uma operação 24/7 dos seus serviços tecnológicos, seria necessário um maior investimento em ferramentas e pessoal, o que conseqüentemente acarretaria um maior custo. Entretanto, no modelo de *Open Utility Cloud Computing* o responsável pela operação do ambiente tecnológico e por garantir a alta disponibilidade é do próprio provedor do serviço.

Nesse cenário, a empresa cliente é responsável apenas por utilizar e administrar suas próprias aplicações, mas não é necessário preocupar-se com os recursos físicos, problemas de hardware e também configurações

de redes. Perante todas essas características do *Open Utility Cloud Computing*, nota-se que esse modelo é também atraente para atender demandas sazonais de empresas de todos os tamanhos. As organizações podem administrar o aumento da demanda em um determinado período, como festa de fim de ano, sem a necessidade de aquisição de novos equipamentos e ferramentas. É possível apenas contratar um serviço especializado no modelo *Cloud Computing* durante o tempo da demanda sazonal, pagar apenas pelo utilizado e finalizar o contrato após esse período.

Outro ponto de destaque é o fato que, no caso de aquisição de dispositivos computacionais, as organizações precisam arcar com a taxa de depreciação, a qual varia em torno de 20% ao ano em equipamentos de informática. Já com o modelo de serviços baseado em Nuvem, o consumidor não é proprietário de nenhum equipamento, apenas utiliza a capacidade computacional como um serviço, sem a necessidade de se preocupar com a depreciação dos equipamentos, instalações físicas e suporte de hardware.

#### **4.1.8 Resultados e contribuições**

O crescimento do modelo de *Cloud Computing* tem acarretado benefícios no contexto econômico em dois principais modos – geração de novos negócios para empresas de tecnologias e também formas mais atraentes de investimentos em tecnologia por parte dos clientes.

Conforme foi abordado no estudo de caso, a Amazon é uma empresa de comércio eletrônico que tem um ambiente computacional de larga escala, o qual não estava sendo utilizado por completo para suprir a sua demanda. A empresa então identificou uma oportunidade de negócio através de *Cloud Computing*, como vender a capacidade excedente de seus data centers em forma de serviços na Internet. Hoje em dia esse serviço está ganhando evidência e a Amazon tem investido para ampliar seu parque tecnológico.

É importante citar que outras empresas também estão seguindo a mesma linha e iniciando serviços baseados em todos os tipos de *Cloud Computing*. A Salesforce.com é uma organização que já tem uma marca consolidada em tecnologia de CRM, sendo que os seus principais serviços são providos através da Internet.

Essa alternativa de disponibilização de software torna-se uma excelente oportunidade para empresas que necessitam de um ambiente

de CRM de forma simples e ágil, sem a necessidade de comprar equipamentos, licenças e gastar com os projetos de implementação. Em alguns minutos, uma empresa pode ter acesso a uma ferramenta de CRM completa com o Salesforce.com através de um portal Web.

É importante também ressaltar a tendência de *Cloud Computing* alinhada com a depreciação; que corresponde à diminuição do valor dos bens devido ao desgaste pelo uso, ação da natureza ou obsolescência tecnológica; e a Globalização, facilitando ainda mais a quebra de barreiras geográfica. Os seus serviços são, na maioria das vezes, orientados à Internet, o que permite que os seus clientes não estejam apenas em sua abrangência local, podendo estar em qualquer país.

Uma empresa do Brasil pode facilmente contratar os serviços da Amazon EC2 e alocar parte de sua infraestrutura de TI nos domínios da empresa Americana, que por sinal poderia disponibilizar os seus serviços a partir de outros locais, como países de baixo custo, muito fáceis de encontrar na prática de *offshoring*. Já do ponto de vista dos usuários de *Cloud Computing*, foi visto até o momento diversos benefícios da tecnologia em termos de redução de investimentos, tanto em relação à aquisição de equipamentos, licença, contratação de mão-de-obra e suporte.

Esse cenário é mais evidente em países emergentes, como o Brasil, onde a tecnologia para ser adquirida e implementada pode ter a necessidade de importação de produtos. É esperada uma mudança na economia das tecnologias computacionais com o avanço de *Cloud Computing*, principalmente por esse modelo permitir a migração da entrega de recursos computacionais através de serviços, e não mais produtos.

#### 4. 2 Sistema Operacional em Cloud: Azure

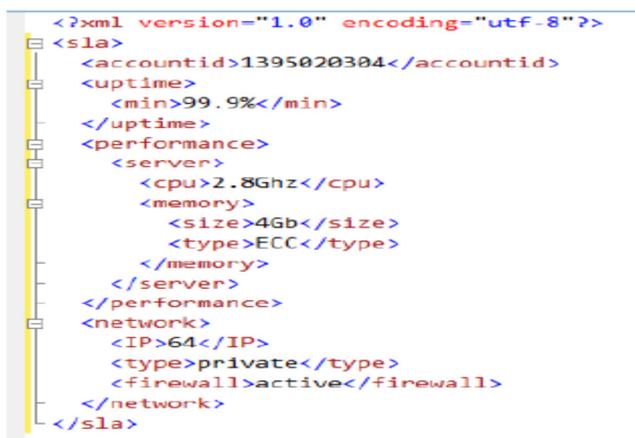
A SLA é um componente crítico para provedores de *Cloud Computing*, pois seus consumidores corporativos estão rodando negócios de missão crítica e é de extrema importância que a disponibilidade e qualidade seja mantida durante o curso do contrato.

Para o provedor do serviço em *Cloud*, há três objetivos principais que esses provedores precisam cumprir para garantir as SLAs:

- SLA deve ser flexível para que os consumidores que demandem diferentes níveis de qualidade possam conseguir esses níveis para sua conta. Um exemplo seria um consumidor corporativo que necessita de 99,99% de *uptime* enquanto outro possa precisar apenas de 97% de *uptime*.

- A provisão de serviços e aplicação de SLA's selecionadas deve automatizar para que a escalabilidade seja melhorada.
- As métricas de SLA devem ser monitoradas e verificadas para que os consumidores possam confiar no resultado dos sistemas de monitoração.

Atualmente o *Windows Azure Cloud Services* (WACS) identificou várias metodologias e tecnologias que atendem especificamente esses requisitos. O WACS foi desenvolvido do início para suportar o consumo público de serviços e diferente do Amazon Web Services(AWS) que foi inicialmente desenvolvido para uso interno apenas. O WACS adota várias maneiras para prover e monitorar as SLAs de um vasto conjunto de contas sem nenhuma intervenção humana (automação).



```
<?xml version="1.0" encoding="utf-8"?>
<sla>
  <accountid>1395020304</accountid>
  <uptime>
    <min>99.9%</min>
  </uptime>
  <performance>
    <server>
      <cpu>2.8Ghz</cpu>
      <memory>
        <size>4Gb</size>
        <type>ECC</type>
      </memory>
    </server>
  </performance>
  <network>
    <IP>64</IP>
    <type>private</type>
    <firewall>active</firewall>
  </network>
</sla>
```

Figura 4.2a - Arquivo com SLA simples.

FONTE: Kankanhalli (2009).

Para prover uma flexibilidade na seleção de SLA, o WACS identificou vários *templates* pré-definidos que podem ser selecionados pelos consumidores durante o processo de inscrição que será aplicado instantaneamente. Como o WACS e o AWS anunciam publicamente seu serviço de *cloud*, a negociação de SLA é normalmente feita online, onde os consumidores utilizam um arquivo XML para especificar a qualidade de serviço requisitada baseado em um conjunto de parâmetros. O arquivo de configuração é anexado à conta se puder ser aceito de acordo com a capacidade do provedor, e será aplicado na sessão de serviços ativa. Depois de aprovado, o serviço de monitoramento de SLA utilizará o arquivo que foi aprovado na negociação como base para rastrear os

padrões de qualidade e verificar o cumprimento da SLA. Para orquestração, dependendo da SLA selecionada, são alocados recursos para os consumidores que foram agrupados em diferentes níveis de disponibilidade, segurança e performance. O processo de automação é feito pelo serviço de provisão que está no controlador de estrutura. O controlador de estrutura é uma camada de serviço que controla todos os serviços virtualizados na nuvem Azure.

Quando um pedido de um consumidor é feito para o WACS, o controlador de estrutura primeiramente verifica a SLA que foi associada com a conta e baseado nas regras que foram definidas nela irá alocar os serviços e recursos para o consumidor. Por exemplo, um consumidor corporativo que requisitou 200% de redundância, com a máxima segurança e a máxima performance será alocado nos recursos que possuem todos seus serviços principais replicados fisicamente entre duas máquinas virtuais diferentes e pré-configuradas para prover um nível de performance superior com a rede configurada com prevenções de ataques externos.

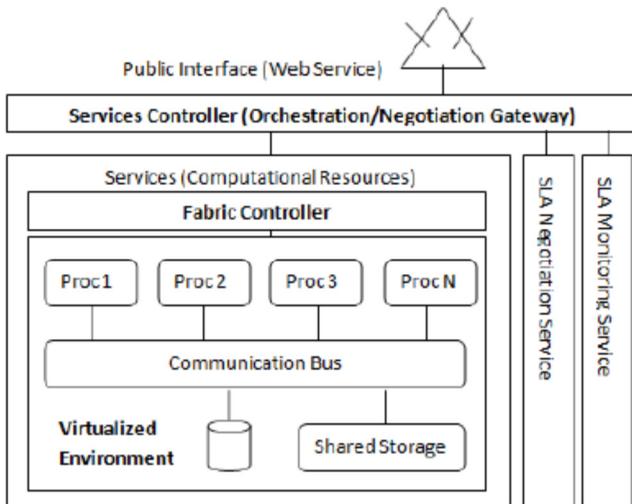


Figura 4.2b. Services Layers no Azure.

FONTE: Kankanhalli (2009).

Para o WACS poder oferecer escalabilidade a automatização das funções principais necessárias para que as contas possam ser providas rapidamente. O controlador de estrutura faz a maior parte da automação como provisão de contas, delegação, monitoramento e resolução de SLAs. O controlador de estrutura automaticamente dá para uma conta

um hardware desejado, software e serviços de rede para o cliente baseado no seu requisito de projeto e SLA. Monitoramento também é feito pelo controlador de estrutura em conjunto com um serviço externo confiável de auditoria para assegurar que as SLA são cumpridas durante todo o tempo. Devido à natureza sensível do monitoramento de SLA, uma auditoria terceirizada é necessária e deve prover monitoramento externo de todos os serviços oferecidos pela WASC e as informações são tornadas públicas para que a compensação devida seja aplicada caso uma SLA não esteja sendo cumprida.

Como falhas de serviço são inevitáveis na escala na qual WASC está operando, um monitoramento ativo e resolução de serviço também é provido pelo controlador de estrutura. O controlador de estrutura adota uma estratégia inovadora na detecção de falhas e protocolo de resolução. Rodando um conjunto de servidores de monitoramento, os servidores irão simular tráfego do mundo real para os servidores que estão rodando no WASC. Se alguma falha é detectada, o servidor em particular será tirado do grid ativo e, dependendo da SLA, um serviço será colocado no lugar dele e ativado para repor o serviço que falhou. O tipo de monitoramento automatizado dá uma forma de capacidade de "auto cura" onde os serviços que estão com falhas são identificados e substituídos antes de uma falha em produção ocorrer, o que pode fazer com que uma SLA seja descumprida. Todo sistema de serviço que é criado pelo controlador de estrutura terá um estado base durante a fase de provisão, baseado na SLA sendo associada com ele. Esse estado base é ativamente monitorado para assegurar que as métricas de performance e Qualidade de Serviço(QoS) sejam mantidas e que no evento onde uma falha é detectada na SLA, o consumidor será alertado e uma compensação será dada a ele. Essa estratégia proativa de monitorar o cumprimento da SLA junto com resolução de falhas faz com que se possa manter um alto nível de QoS sem comprometer serviço de entrega e performance. Como sistemas públicos de cloud estão rapidamente se tornando populares como forma de serviço e entrega de plataformas, as inovações em negociação de SLA, orquestração e automação vão ser levantadas enquanto os consumidores demandarem maior customização e tempo de *deploy* mais rápido.

## 5. CONCLUSÕES

A velocidade da pesquisa de negociação dinâmica e automatizada de SLA teve uma aceleração nos últimos anos, enquanto o mundo corporativo vê que serviços terceirizados são chave para se obter uma vantagem competitiva que possibilita cortar custos e focar apenas nas competências primárias da empresa. Provedores de sistemas de cloud em larga escala, como o Windows Azure e Amazon Web Service começaram uma nova onda tecnológica que faz com que qualquer um no mundo possa se inscrever no seu serviço e obter qualidade corporativa pagando apenas pelo uso num modelo de serviços sob demanda.

Devido à substancial alta na adoção e demanda de sistemas com qualidade, os SLAs se tornaram a ferramenta legal para os negócios terem garantia de sucesso quando eles terceirizam operações de missão crítica para esses provedores. Mesmo que sistemas legados tenham protocolos e procedimentos bem estabelecidos na área de gerência de SLA, eles estão lentamente se tornando obsoletos com a necessidade de SLAs dinâmicas, escaláveis e instantâneas onde monitoramento constante é requerido.

Apresentou-se uma aproximação para a negociação de SLAs de processos de negócio e a aplicação correspondente de QoS a nível de infraestrutura de TI. Que consiste em três tarefas principais. A formalização dos NFPs permite que sejam comparados os níveis de serviço existentes. Usando um SLA dinâmico podem-se encontrar níveis de serviço automaticamente, reconfigurando réplicas do serviço em ambientes de computação em nuvem. Essa abordagem propicia ao usuário saber se está pagando apenas pelo uso necessário e dando garantias em de que suas demandas estão sendo atendidas em tempo real. Em um pico mais recursos são renegociados com o servidor, e em uma baixa são dispensados recursos ociosos.

Foi simulado um cenário dentro do CloudAnalyst, comparando-se uma execução semanal e semestral, de uma base de usuários espalhados pelo globo. As comparações foram entre abordagem tradicional ou estática x dinâmica. Medindo-se tempo de resposta e % *uptime*. A avaliação experimental semanal demonstrou-se que seria possível melhorar níveis de serviço em 7% sob cargas semanais e 120% em um período semestral.

Provou-se que QoS individuais ou SLO suprem às exigências específicas de cada processo de negócio e suporta conceitos como custo

x benefício em uma empresa. Argumenta-se também que a abordagem fechou os GAPS da literatura, onde os trabalhos apresentados não conseguiam prover uma solução de SLA que pudessem ao longo do tempo ser renegociadas, melhorando o desempenho global e consequentemente o ganho individual de cada usuário na nuvem.

Essa abordagem se mostrou mais eficiente que a estática em atender os anseios da nova computação, pois consegue gerar um economia de recursos aos servidores, uma vez que reduzimos a ociosidade e sobrecarga de recursos, o usuário por sua vez tem um melhor resultado individual pagando menos por isso. A longo prazo temos a possibilidade de reduzir o parque total de computadores, uma vez que os usuários tendem a contratar serviços de nuvem e pagar por eles somente o necessário, ou pagar pelo uso.

### 5.1 Trabalhos Futuros

Trabalha-se atualmente em uma interface com o usuário que permitirá aos donos dos processos de configurar suas preferências esperadas considerando NFPs de uma forma fácil e conveniente.

Desse modo, um conjunto pré-definido de nível de serviços por tempo de resposta, disponibilidade e outros NFPs irão corresponder a uma simples descrição (e.g. gold, standard, otimizado por custo). Usuários poderão selecionar uma configuração usando uma interface simples como um "slider" e a infraestrutura se adaptará automaticamente a configuração.

Além disso, atualmente investigam-se meios de derivar tais preferências de processos existentes e incorporá-los no repositório de modelos de processo como SLOs que resultarão por fim na provisão automática de composição de serviços que melhor atendem os requisitos funcionais e não-funcionais do processo de negócio. Planeja-se resolver limitações de replicação distribuída de serviços com respeito à disponibilidade e particularmente a trade-off comunicação versus disponibilidade.

## REFERÊNCIAS

ABDENNADHER, N., Boesch, R.: Deploying phylip phylogenetic package on a large scale distributed system. In: IEEE International Symposium on Cluster Computing and the Grid, pp. 673–678 (2007).

ABDENNADHER, N., Boesch, R.: A scheduling algorithm for high performance peer-to-peer platform. In: Lehner, W., Meyer, N., Streit, A., Stewart, C. (eds.) Euro-Par Workshops 2006. LNCS, vol. 4375, pp. 126–137. Springer, Heidelberg (2007).

AMAZON Elastic Compute Cloud (EC2), <<http://www.amazon.com/ec2/>>. Acessado em 2009.

ANTONIO, S. D. et al. Managing service level Agreements in Premium IP Networks a business-oriented approach. Computer Networks. v46, n. 6, 20 Dec. 2004.

ASSUNÇÃO, Marcos (2004). “Implementação e análise de uma arquitetura de grids de agentes para a gerência de redes e sistemas”. Dissertação de Mestrado. PPGCC-UFSC, Florianópolis, 2004.

BALEN, Douglas; FRANKE, Hans; HARTMANN, Eduardo; WESTPHALL, Carlos. Gerenciamento de Redes em Grids de Computadores: Uma abordagem usando Redes de Sensores e Contexto no Monitoramento de Redes. Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC 2008). XIII Workshop de Gerência e Operação de Redes e Serviços (WGRS 2008). Rio de Janeiro (RJ), de 26 a 30 de maio de 2008.

BECHTOLSHEIM, A. Cloud Computing and Cloud Networking. UC Berkeley, Dezembro 2008.

BEST, I. Implementing Service Level Management. Committed Technologies, 2002.

VISIBLE MEASURES GROUP. Acesso em Setembro 2009. Disponível em <<http://www.visiblemeasures.com>>

BHATTI, R.; SHAFIQ, B.; SHEHAB, M.; GHAFOR, A. Distributed

access management in multimedia IDCs. IEEE Computer Volume 38, Issue 9, Sept. 2005 Page(s):60 - 69 Digital Object Identifier 10.1109/MC.2005.296.

BUYYA, R., Chee, S. Y., Srikumar, V., Broberg, J., and Brandic, I. (2008). Cloud Computing and Emerging IT Platforms: Vision, Hype, and Reality for Delivering Computing as the 5th Utility. Retrieved February 10, 2009. Disponível em <<http://www.gridbus.org/reports/CloudITPlatforms2008.pdf>>

CAMPBELL, Andrew T; COULSON, Geof; KOUNAVIS, Michael. "Managing Complexity: Middleware Explained," IT Professional, vol. 01, no. 5, pp. 22-28, September/October, 1999.

CANFORA, G., Di Penta, M., Esposito, R., Villani, M.L.: An approach for QoS aware service composition based on genetic algorithms. In: Proceedings of the 2005 conference on Genetic and evolutionary computation, pp. 1069–1075 (2005).

Chu, X, Nadiminti, K., Jin, C. Venugopal, S. and Buyya, R. Aneka: Next-Generation Enterprise Grid Platform for e-Science and e-Business Applications. In Proceedings of the 3th IEEE International Conference on e-Science and Grid Computing (e-Science 2007), Bangalore, India, Dec. 2007.

DAS, S.; DAS, S.K. 5th International Workshop on Distributed Computing, volume 2918 of Lecture Notes in Computer Science, Kolkata, India, December 2003.

FOSTER, I.; KESSELMAN, C. The Grid: Blueprint for a New Computing Infrastructure. San Francisco, CA, USA: Morgan Kaufmann Publishers, 1999.677p.

FOSTER, I; KESSELMAN, C.; TUECKE S. The Anatomy of the Grid: Enabling Scalable Virtual Organizations. International Journal of Supercomputer Applications, v.15 n.3, 2001.

FOSTER, I. "What is the Grid? A Three Point Checklist". GRIDToday, July 20, 2002.

FOSTER, C. KESSELMAN, J. NICK, S. Tuecke. The Physiology of the

Grid: An Open Grid Services Architecture for Distributed Systems Integration. Open Grid Service Infrastructure WG, Global Grid Forum, June 22, 2002

FOSTER, I. & KESSELMAN, C. Foster, I. & Kesselman, C. (ed.) The Grid 2: Blueprint for a New Computing Infrastructure Elsevier, 2004.

FOSTER, I; TUECKE, S. The Different Faces of IT as Service Ian Foster and Steven Tuecke's article from the Enterprise Distributed Computing issue of ACM Queue (Vol. 3, No. 6 - July/August 2005)

FOSTER, I. Globus toolkit version 4: Software for service-oriented systems. IFIP International Conference on Network and parallel computing, Springer-Verlag LNCS 3779: p. 2-13, 2005.

FRANKE, Hans Alberto; WESTPHALL, Carlos Becker; ROLIM, Carlos Oberdan; NAVARRO; Fabio. Mobilidade em Ambiente de Grades Computacionais. XXIV Simpósio Brasileiro de Redes de Computadores. IV Workshop on Computational Grids and Applications - WCGA. Curitiba - PR, de 29 de maio a 2 de junho de 2006 (Resumo).

FRANKE, Hans A.; ROLIM, Carlos O.; WESTPHALL, Carlos B.; KOCH, Fernando; BALEN, Douglas O. Grid-M: Middleware to Integrate Mobile Devices, Sensors and Grid Computing. In The Third International Conference on Wireless and Mobile Communications – ICWMC 2007, Guadeloupe, French Caribbean, March 4-9, 2007.

FRANKE, H.; Balen, D.; KOCH, F.; Vieira, K.; ROLIM, O.; WESTPHALL, C.. Grid-M : Middleware para Integrar Dispositivos Móveis, Sensores e Grids. XII Workshop de Gerência e Operação de Redes e Serviços (WGRS 2007), 28 de maio de 2007 junto ao SBRC, em Belém-PA.

GAI, S. “Internetworking IPv6 with Cisco Routers”. Acessado em 28 de dezembro de 2005 , disponível em: <http://www.ip6.com>.

GAYNOR, M. MOULTON, S.L. WELSH, M. LaCombe, E. Rowan, A. Wynne, J. Integrating wireless sensor networks with the grid. Internet Computing, IEEE Volume 8, Issue 4, July-Aug. 2004 Page(s):32 – 39 Digital Object Identifier 10.1109/MIC.2004.18  
GOOGLE App Engine, <http://appengine.google.com>

GUIGUERE, E. Java 2 Micro edition: The ultimate guide on programming handheld and embedded devices. John Wiley and Sons, Inc., USA, 2001.

GRID-M website, <http://grids.lrg.ufsc.br/>

GRIDBUS Project web-site, Grid Computing and Distributed Systems Laboratory, <http://www.gridbus.org/>

GRIDFORUM Website "Understanding Grids" Acessado em 08 de dezembro de 2005, disponível em: [http://www.gridforum.org/UnderstandingGrids/ggf\\_grid\\_understand.php](http://www.gridforum.org/UnderstandingGrids/ggf_grid_understand.php).

HAMILTON, D.. 'Cloud computing' seen as next wave for technology investors. Financial Post, 04 June 2008. <http://www.financialpost.com/money/story.html?id=562877>

HILES, A. The complete guide to I.T service level agreements. [S.I.] Rothstein, 2000.

IBACH, P., Stantchev, V., Keller, C.: Daedalus a peer-to-peer shared memory system for ubiquitous computing. In: Nagel, W.E., Walter, W.V., Lehner, W. (eds.) Euro-Par 2006. LNCS, vol. 4128, pp. 961–970. Springer, Heidelberg (2006).

J2ME. Java 2 Micro Edition web-site, Sun Corporation, <http://java.sun.com/j2me>.

KAMINSKI, H., Perry, M. (2006), SLA Automated Negotiation Manager for Computing Services, IEEE, Computer Society.

Kankanhalli, A. Service level agreement (SLA) negotiation, automation and orchestration for Cloud computing, 2009.

KEAHEY, K. I. Foster, T. Freeman, and X. Zhang. Virtual workspaces: Achieving quality of service and quality of life in the Grid. Scientific Programming, 13(4):265-275, October 2005.

KOCK, F; MEYER, J.-J. C; DIGNUM, F and RAHWAN. I. Programming deliberative agents for mobile services: the 3apl-m platform. In Proceedings of AAMAS'05 Workshop on Programming

Multi-Agent Systems (ProMAS'2005)., 2005.

YAU, S.S., Wang, Y., Huang, D., Hoh, P.: Situation-aware contract specification language for middleware for ubiquitous computing. In: The Ninth IEEE Workshop on Future Trends of Distributed Computing Systems, 2003. FTDCS 2003. Proceedings, May 28-30, 2003, pp. 93–99 (2003).

LEGION, World Wide Virtual Computer, <http://legion.virginia.edu/>

LDAP, openLDAP <http://www.openldap.org/>

LITKE, A.; SKOUTAS, D and VARVARIGOU, T. “Mobile grid computing: Changes and challenges of resource management in a mobile grid environment,” in Proc. of Practical Aspects of Knowledge Management (PAKM 2004), Austria, December 2004.

LLORENTE, I., OpenNebula Project. <http://www.opennebula.org/>

MAKRIPOULIAS, Y., Makris, C.,Panagis, Y.,Sakkopoulos, E.,Adamopoulou, P.,Pontikaki, M., Tsakalidis, A.: Towards Ubiquitous Computing with Quality of Web Service Support. Upgrade, The European Journal for the Informatics Professional VI(5),29–34 (2005).

MICROSOFT. Comparing Web Service Performance: WS Test 1.1 Benchmark Results for .NET 2.0, NET 1.1, Sun One/ JWS DP 1.5 and IBM WebSphere 6.0 (2006), <http://www.theserverside.net/tt/articles/content/NET2Benchmarks>.

MICROSOFT Live Mesh, <http://www.mesh.com>

MORGAN Stanley. Technology Trends. 12 June 2008. <http://www.morganstanley.com/institutional/techresearch/pdfs/TechTrends062008.pdf> [18 July 2008]

OBJECT WEB, Open Source Middleware. Acessado em 28 de dezembro de 2005, disponível em: <http://middleware.objectweb.org/>

POLZE, A., Schwarz, J., Malek, M.: Automatic generation of fault-tolerant corbaservices Tools, 205 (2000).

PARKHILL, D. The Challenge of the Computer Utility. Addison-Wesley Educational Publishers Inc., US, 1966.

ROLIM, Carlos Oberdan; WESTPHALL, Carlos Becker; KOCH, Fernando; ASSUNÇÃO, Marcos. Towards a Grid of Sensors for Telemedicine. CBMS 2006 - 19th IEEE International Symposium on Computer-Based Medical Systems. Salt Lake City, USA, 22-23 June, 2006.

SAS70. <http://www.sas70.com/about.htm>. Acessado em: 2009.

SATYANARAYANAN, M. Pervasive computing: vision and challenges. IEEE Personal Communications, 8(4):10-17, 2001.

SCHIKUTA, E., Mach, W.: Optimized workflow orchestration of parallel database aggregate operations on a heterogenous grid. In: The 37th International Conference on Parallel Processing (ICPP 2008), Portland, Ohio, USA. IEEE Computer Society, Los Alamitos (2008).

SOLBERG, A., Amundsen, S., Aagedal, J.O., Eliassen, F.: A Framework for QoS-aware Service Composition. In: Proceedings of 2nd ACM International Conference on Service Oriented Computing (2004).

STANCHEV, V., Malek, M.: Architectural Translucency in Service-oriented Architectures. IEE Proceedings - Software 153(1), 31–37 (2006)

STANCHEV, V.: Effects of Replication on Web Service Performance in WebSphere. Icsi tech report 2008-03, International Computer Science Institute, Berkeley, California 94704, USA (February 2008).

STANCHEV, V., Malek, M.: Addressing Web Service Performance by Replication at the Operating System Level. In: ICIW 2008: Proceedings of the 2008 Third International Conference on Internet and Web Applications and Services, pp. 696– 701. IEEE Computer Society, Los Alamitos (2008).

STANCHEV, V. Schröpfer, C.: Negotiating and Enforcing QoS and SLAs in Grid and Cloud Computing, FOM Fachhochschule fuer Oekonomie und Management, Berlin, Germany (2009)

SUN (SunGrid), <http://www.network.com>. Acessado em 2008.

THAM, Chen-Khong; BUYYA, Rajkumar. SensorGrid: Integrating Sensor Networks and Grid Computing. CSI Communications, pages 24-29, Vol.29, No.1, Computer Society of India (CSI) Publication, July 2005.

TOKAIRIN, Y., Yamanaka, K., Takahashi, H., Sukanuma, T., Shiratori, N.: An effective qos control scheme for ubiquitous services based on context information management. cec-eee, 619–625 (2007).

USKELA, S. Key concepts for evolution toward beyond 3g networks. IEEE Wireless Communications, 10(1):43-48, 2003.

UNICORE, UNiform Interface to COmputer Resources, <http://www.unicore.org/>

VOGELS, W. A Head in the Clouds – The power of Infrastructure as a Service. In First Workshop on Cloud Computing and Applications (CCA '08), Outubro 2008.

XML, W3C Consortium <http://www.w3.org/XML/>

WANEK, H., Schikuta, E., Haq, I.U.: Grid workflow optimization regarding dynamically changing resources and conditions. Concurrency and Computation: Practice and Experience (2008).

ZENG, L., Benatallah, B., Ngu, A.H.H., Dumas, M., Kalagnanam, J., Chang, H.: QoS-aware middleware for Web services composition. IEEE Transactions on Software Engineering 30(5), 311–327 (2004).

WICKREMASINGHE, Bhatiya. Calheiros, Rodrigo N. Buyya, Rajkumar. CloudAnalyst: A CloudSim-based Visual Modeller for Analysing Cloud Computing Environments and Applications. Technical Report, CLOUDS-TR-2009-12, Cloud Computing and Distributed Systems Laboratory, The University of Melbourne, Australia, Oct. 23, 2009.

## **ANEXO A – Código Fonte - CloudAnalyst**

O código-fonte do CloudAnalyst, pode ser acesso através do site:  
[www.cloudbus.org/cloudsim/](http://www.cloudbus.org/cloudsim/).

## ANEXO B – Resultados Simulação

Segue abaixo alguns exemplos dos resultados das simulações apresentadas no capítulo 3.12:

Tabela B.1 Resultado Custo Dinâmico Semestre

Client	Rodada	Data Center	VM Cost \$	Data Transfer Cost \$	Total \$
0	0	DC0	564,97	109,67	674,64
0	1	DC0	564,97	109,67	674,64
0	2	DC0	564,97	109,67	674,64
0	3	DC0	564,97	109,67	674,64
0	4	DC0	564,97	109,67	674,64
1	0	DC0	563,27	69,6	632,86
1	1	DC0	565,77	69,6	635,37
1	2	DC4	112,68	13,76	126,45
1	2	DC3	112,68	13,99	126,67
1	2	DC2	113	13,88	126,88
1	2	DC1	112,83	14,1	126,93
1	2	DC0	112,71	13,87	126,58
1	3	DC4	115,65	13,81	129,46
1	3	DC3	116,1	13,96	130,06
1	3	DC2	115,88	13,98	129,86
1	3	DC1	116,23	13,91	130,14
1	3	DC0	116,12	13,94	130,06
1	4	DC4	119,53	14,14	133,67
1	4	DC3	119,43	14,04	133,47
1	4	DC2	119,77	13,76	133,53
1	4	DC1	119,51	13,88	133,4
1	4	DC0	119,12	13,77	132,89
2	0	DC4	123	5,94	128,94
2	0	DC3	123,51	5,94	129,45
2	0	DC2	123,15	5,94	129,09

2	0	DC1	123,48	5,98	129,45
2	0	DC0	123,32	5,93	129,25
2	1	DC4	124,97	5,9	130,87
2	1	DC3	124,86	5,95	130,8
2	1	DC2	125,01	5,9	130,91
2	1	DC1	124,6	5,98	130,58
2	1	DC0	124,7	6	130,7
2	2	DC4	126,19	5,9	132,09
2	2	DC3	126,64	5,89	132,53
2	2	DC2	126,85	6,04	132,89
2	2	DC1	126,45	5,94	132,38
2	2	DC0	126,46	5,97	132,43
2	3	DC4	128,55	5,9	134,45
2	3	DC3	128,05	5,93	133,98
2	3	DC2	128,38	6,05	134,43
2	3	DC1	128,34	5,97	134,31
2	3	DC0	128,55	5,88	134,43
2	4	DC0	655,84	29,74	685,58
3	0	DC0	656,05	49,75	705,8
3	1	DC4	129,69	9,85	139,54
3	1	DC3	130,51	9,98	140,49
3	1	DC2	130,32	10,06	140,38
3	1	DC1	130,05	9,87	139,92
3	1	DC0	129,92	9,99	139,91
3	2	DC4	133,25	9,93	143,17
3	2	DC3	132,93	10,01	142,94
3	2	DC2	132,81	9,93	142,74
3	2	DC1	132,61	9,97	142,58
3	2	DC0	133,04	9,92	142,96
3	3	DC4	135,88	10,03	145,91
3	3	DC3	136,52	9,88	146,4

3	3	DC2	136,38	9,97	146,36
3	3	DC1	136,8	9,95	146,76
3	3	DC0	136,16	9,91	146,07
3	4	DC0	707,09	49,75	756,85
4	0	DC0	723,31	49,75	773,06
4	1	DC0	725,22	49,75	774,98
4	2	DC0	725,22	49,75	774,98
4	3	DC0	725,22	49,75	774,98
4	4	DC0	725,22	49,75	774,98
5	0	DC4	139,3	7,86	147,17
5	0	DC3	139,75	7,95	147,7
5	0	DC2	139,68	8,04	147,73
5	0	DC1	139,68	7,88	147,55
5	0	DC0	139,67	7,97	147,64
5	1	DC4	142,3	7,94	150,24
5	1	DC3	141,46	8,02	149,48
5	1	DC2	142,27	7,94	150,21
5	1	DC1	142,57	7,85	150,42
5	1	DC0	142,47	7,94	150,41
5	2	DC0	743,16	39,7	782,86
5	3	DC4	144,97	7,96	152,93
5	3	DC3	145,5	8,02	153,51
5	3	DC2	145	7,91	152,91
5	3	DC1	145,58	7,95	153,53
5	3	DC0	145,43	7,87	153,29
5	4	DC4	148,47	7,97	156,44
5	4	DC3	148,12	8,01	156,13
5	4	DC2	148,34	7,89	156,22
5	4	DC1	148,05	8,04	156,09
5	4	DC0	148,22	7,8	156,02

Tabela B.2 Taxa de Sucesso Dinâmico Semestre

Client	Rodada	Userbase	Success Rate(%)
0	0	UB0	9,55
0	0	UB10	9
0	0	UB1	8,97
0	0	UB2	9,23
0	0	UB3	9,12
0	0	UB4	9,46
0	0	UB5	9,31
0	0	UB6	9,5
0	0	UB7	8,76
0	0	UB8	9,18
0	0	UB9	8,84
0	1	UB0	9,51
0	1	UB10	9,11
0	1	UB1	8,91
0	1	UB2	9,21
0	1	UB3	9,21
0	1	UB4	9,53
0	1	UB5	9,35
0	1	UB6	9,4
0	1	UB7	8,79
0	1	UB8	9,19
0	1	UB9	8,8
0	2	UB0	9,51
0	2	UB10	9,11
0	2	UB1	8,89
0	2	UB2	9,29
0	2	UB3	9,09

0	2	UB4	9,52
0	2	UB5	9,25
0	2	UB6	9,51
0	2	UB7	8,8
0	2	UB8	9,19
0	2	UB9	8,81
0	3	UB0	9,54
0	3	UB10	9,06
0	3	UB1	8,87
0	3	UB2	9,26
0	3	UB3	9,07
0	3	UB4	9,6
0	3	UB5	9,4
0	3	UB6	9,52
0	3	UB7	8,76
0	3	UB8	9,08
0	3	UB9	8,82
0	4	UB0	9,54
0	4	UB10	9,01
0	4	UB1	8,84
0	4	UB2	9,28
0	4	UB3	9,12
0	4	UB4	9,51
0	4	UB5	9,29
0	4	UB6	9,51
0	4	UB7	8,84
0	4	UB8	9,11
0	4	UB9	8,85
1	0	UB0	9,17

1	0	UB1	9,79
1	0	UB2	9,87
1	0	UB3	9,13
1	0	UB4	9,1
1	0	UB5	9,58
1	0	UB6	9,54
1	1	UB0	9,63
1	1	UB1	9,34
1	1	UB2	9,23
1	1	UB3	9,62
1	1	UB4	9,43
1	1	UB5	9,31
1	1	UB6	9,31
1	2	UB0	92,8
1	2	UB1	92,58
1	2	UB2	93,09
1	2	UB3	93,97
1	2	UB4	93,35
1	2	UB5	92,99
1	2	UB6	93,67
1	3	UB0	92,31
1	3	UB1	92,4
1	3	UB2	92,25
1	3	UB3	93,02
1	3	UB4	92,55
1	3	UB5	92,55
1	3	UB6	92,69
1	4	UB0	91,77
1	4	UB1	91,29

1	4	UB2	91,91
1	4	UB3	92,11
1	4	UB4	92
1	4	UB5	91,87
1	4	UB6	92,19
2	0	UB0	91,72
2	0	UB1	91,96
2	0	UB2	91,89
2	1	UB0	91,38
2	1	UB1	91,41
2	1	UB2	91,96
2	2	UB0	91,11
2	2	UB1	91,19
2	2	UB2	90,67
2	3	UB0	90,95
2	3	UB1	90,6
2	3	UB2	90,3
2	4	UB0	8,62
2	4	UB1	8,21
2	4	UB2	8,64
3	0	UB0	8,51
3	0	UB1	7,99
3	0	UB2	8,25
3	0	UB3	8,61
3	0	UB4	8,61
3	1	UB0	89,92
3	1	UB1	89,26
3	1	UB2	89,41
3	1	UB3	89,62

3	1	UB4	89,75
3	2	UB0	88,92
3	2	UB1	88,72
3	2	UB2	87,84
3	2	UB3	89,3
3	2	UB4	89,15
3	3	UB0	87,58
3	3	UB1	87,61
3	3	UB2	87,99
3	3	UB3	88,23
3	3	UB4	88,21
3	4	UB0	7,96
3	4	UB1	7,26
3	4	UB2	7,99
3	4	UB3	8,14
3	4	UB4	7,61
4	0	UB0	7,94
4	0	UB1	7,63
4	0	UB2	7,74
4	0	UB3	7,27
4	0	UB4	7,59
4	1	UB0	7,92
4	1	UB1	7,12
4	1	UB2	7,76
4	1	UB3	7,99
4	1	UB4	7,48
4	2	UB0	7,94
4	2	UB1	7,13
4	2	UB2	7,73

4	2	UB3	7,98
4	2	UB4	7,49
4	3	UB0	7,9
4	3	UB1	7,18
4	3	UB2	7,76
4	3	UB3	7,91
4	3	UB4	7,48
4	4	UB0	7,93
4	4	UB1	7,12
4	4	UB2	7,8
4	4	UB3	7,97
4	4	UB4	7,45
5	0	UB0	87,91
5	0	UB1	86,32
5	0	UB2	87,72
5	0	UB3	88,45
5	1	UB0	87,07
5	1	UB1	86,03
5	1	UB2	85,96
5	1	UB3	87,18
5	2	UB0	7,51
5	2	UB1	7,74
5	2	UB2	7,51
5	2	UB3	7,7
5	3	UB0	85,72
5	3	UB1	85,62
5	3	UB2	85,6
5	3	UB3	87,21
5	4	UB0	84,87

5	4	UB1	84,22
5	4	UB2	84,55
5	4	UB3	86,24