MODELO DE CONFIANÇA PARA A TROCA DE ARQUIVOS EM UMA NUVEM PRIVADA

EDNA DIAS CANEDO

TESE DE DOUTORADO EM ENGENHARIA ELÉTRICA

DEPARTAMENTO DE ENGENHARIA ELÉTRICA

FACULDADE DE TECNOLOGIA

UNIVERSIDADE DE BRASÍLIA

UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

MODELO DE CONFIANÇA PARA A TROCA DE ARQUIVOS EM UMA NUVEM PRIVADA

EDNA DIAS CANEDO

ORIENTADOR: RAFAEL TIMÓTEO DE SOUSA JÚNIOR

TESE DE DOUTORADO EM ENGENHARIA ELÉTRICA

PUBLICAÇÃO: PPGENE.TD – 061/2012

BRASÍLIA, DF: 15 DE AGOSTO DE 2012.

UNIVERSIDADE DE BRASÍLIA

FACULDADE DE TECNOLOGIA

DEPARTAMENTO DE ENGENHARIA ELÉTRICA

MODELO DE CONFIANÇA PARA A TROCA DE ARQUIVOS EM UMA NUVEM PRIVADA

EDNA DIAS CANEDO

TESE DE DOUTORADO SUBMETIDA AO DEPARTAMENTO DE ENGENHARIA ELÉTRICA DA FACULDADE DE TECNOLOGIA DA UNIVERSIDADE DE BRASÍLIA, COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR EM ENGENHARIA ELÉTRICA.

APROVADO POR:

_____
RAFAEL TIMÓTEO DE SOUSA JÚNIOR
DOUTOR, UNB/ENE (ORIENTADOR)

_____
ROBSON DE OLIVEIRA ALBUQUERQUE
DOUTOR, ABIN (EXAMINADOR EXTERNO)

_____
MARISTELA TERTO DE HOLANDA
DOUTORA, UNB/CIC (EXAMINADORA EXTERNA)

_____
GEORGES DANIEL AMVAME-NZE
DOUTOR, UNB/FGA (EXAMINADOR EXTERNO)

_____
FLAVIO ELIAS GOMES DE DEUS
DOUTOR, UNB/ENE (EXAMINADOR INTERNO)

BRASÍLIA, DF, 15 DE AGOSTO DE 2012.

FICHA CATALOGRÁFICA

Canedo, Edna Dias.

Modelo de Confiança para a Troca de Arquivos em uma Nuvem Privada [Distrito Federal], 2012.

Xiii, 93p., 210 x 297mm (ENE/FT/UnB, Doutor, Engenharia Elétrica, 2012).

Tese de Doutorado – Universidade de Brasília, Faculdade de Tecnologia.

Departamento de Engenharia Elétrica.

1. Sistemas Distribuídos	2. Computação em Nuvem
3. Disponibilidade	4. Troca de Arquivos
5. Modelo de Confiança	
I. ENE/FT/UnB	II. Título (série)

REFERÊNCIA BIBLIOGRÁFICA

Canedo, Edna Dias. (2012). Modelo de Confiança para a Troca de Arquivos em uma Nuvem Privada. Tese de Doutorado, Publicação PPGENE.TD-061/2012, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 93p.

CESSÃO DE DIREITOS

AUTOR: Edna Dias Canedo

TITULO DA TESE: Modelo de Confiança para a Troca de Arquivos em uma Nuvem Privada.

GRAU / ANO: Doutor / 2012

_____

Edna Dias Canedo
EQRSW 02/03 Lote 03 Apartamento 211
Espaço Caravela – Setor Sudoeste
CEP: 70.675-260 - Brasília - DF
Tel. 55 – 61 – 81140478 / ednacanedo@unb.br

AGRADECIMENTOS

Gostaria de agradecer, em primeiro lugar, ao meu filho Breiner Gabriel pelo completo apoio e compreensão de estarmos e ficarmos distantes durante o período em que estou dedicando-me a este trabalho. O seu carinho e amor são fundamentais para a realização deste trabalho.

A toda equipe do LabRedes, onde tenho o prazer de conviver com pessoas maravilhosas, em especial a Adriana e Beatriz.

Agradeço a Robson pelo incentivo e discussões sobre o tema.

Em especial, agradeço ao Professor Rafael Timóteo, pelo carinho, confiança e compartilhamento do grande conhecimento que possui.

Agradeço ao Ministério do Planejamento, Orçamento e Gestão por apoiar minha pesquisa.

Por último e muito importante, meu eterno agradecimento ao grande amigo Laerte Peotta, seu incentivo é fundamental para que eu possa continuar. Obrigada por tudo.

Meus sinceros agradecimentos.

RESUMO

MODELO DE CONFIANÇA PARA A TROCA DE ARQUIVOS EM UMA NUVEM PRIVADA

Autor: Edna Dias Canedo
Orientador: Professor Dr. Rafael Timóteo de Sousa Junior
Programa de Pós-graduação em Engenharia Elétrica
Brasília, 15 de Agosto de 2012.

Os recentes avanços na tecnologia de computação em nuvem demonstram um aumento nos problemas relacionados à segurança, privacidade e confiança em diferentes aspectos, os quais não haviam sido previstos na concepção dos ambientes de computação em nuvem. Entre estes, o problema da confiança entre os usuários e a garantia do acesso seguro aos recursos destes ambientes tem merecido uma atenção especial. Neste trabalho, é apresentada uma revisão dos conceitos da confiança e reputação, da computação em nuvem e são abordadas algumas questões em aberto relacionadas à confiança e segurança em ambientes de computação em nuvem. A representação da confiança e da reputação em sistemas computacionais tem sido amplamente discutida e aplicada em diversos cenários da tecnologia da informação, se tornando objeto de pesquisas científicas tanto do ponto de vista teórico quanto prático. Como resultado, diversos trabalhos estão sendo apresentados na tentativa de minimizar e solucionar os problemas decorrentes da segurança e confiabilidade nestes ambientes. Esta pesquisa propõe a criação de um modelo de confiança de alto nível para garantir a troca confiável de arquivos entre os usuários de uma nuvem privada, bem como o cálculo da confiança entre os respectivos usuários, de acordo com as métricas estabelecidas. Para validar o modelo proposto foi utilizado um ambiente de simulação com a ferramenta CloudSim. A sua utilização para executar as simulações dos cenários adotados permitiu: calcular a tabela de confiança dos nós (máquinas virtuais) e selecionar os considerados mais confiáveis; identificar que as métricas adotadas influenciam diretamente no cálculo da confiança em um nó; avaliar a adequação das métricas utilizadas, possibilitando identificar e selecionar as mais adequadas em relação ao histórico do comportamento dos nós pertencentes ao ambiente analisado; verificar que o modelo de confiança proposto permite efetivamente a escolha da máquina mais adequada para efetuar a troca de arquivos.

ABSTRACT

TRUST MODEL FOR THE EXCHANGE OF FILES IN A PRIVATE CLOUD
Author: Edna Dias Canedo
Supervisor: Professor Dr. Rafael Timóteo de Sousa Junior
Programa de Pós-graduação em Engenharia Elétrica
Brasília, 15 August 2012

Recent advances in cloud computing have shown an increase in problems related to security, privacy and trust in different aspects, which had not been anticipated in the design of cloud computing environments. Among these, the lack of trust between users and ensuring secure access to the resources in these environments has deserved special attention. In this work, it is presented a review of concepts of cloud computing, trust and reputation and some open issues related to trust and security in cloud computing environments are addressed. The representation of trust and reputation in computer systems has been widely discussed and applied in various Information Technology scenarios, becoming the subject of scientific research, both from the theoretical and practical point of view. As a result, several studies are being presented in an attempt to minimize and solve problems relating to security and reliability in these environments. This research proposes the creation of a high level trust model to ensure the reliable files exchange between the users of a private cloud as well as the measurement of their trust, according to the metrics established. To validate the proposed model, a simulation environment with the tool CloudSim was used. Its use to run the simulations of the adopted scenarios allowed us to calculate the nodes (virtual machines) trust table and select those considered more reliable; identify that the metrics adopted by us directly influenced the measurement of trust in a node; assess the adequacy of metrics used, allowing to identify and select the most appropriate in relation to the historical behavior of the nodes belonging to the analyzed environment; verify that the trust model proposed effectively allows the selection of the most suitable machine to perform the exchange of files.

SUMÁRIO

LISTA DE TABELAS

LISTA DE FIGURAS

LISTA DE ACRÔNIMOS

API	*Application Programming Interface*
CDRM	*Cost-effective Dynamic Replication Management*
CRM	*Customer Relationship Management*
CSP	Provedor de Serviço em Nuvem
DaaS	*Data as a service*
DAÍ	Inteligência Artificial Distribuída
EC2	*Elastic Compute Cloud*
HDFS	*Hadoop Distributed File System*
GFS	*Google File System*
IBM	*International Business Machines*
IaaS	*Infrastructure as a Service*
LAN	Rede Local de Computadores
MD5	*Message-Digest algorithm 5*
MTCEM	*Multi-Tenancy Trusted Computing Environment Model*
NP	Nuvem Privada
NTFS	*New Technology File System*
PaaS	*Platform as a Service*
QoS	Qualidade de Serviço
RTA	*Reputation Trust Agent*
SaaS	*Software as a Service*
SGBD	Sistema de Gerenciamento de Banco de Dados
SLA	Acordo de Nível de Serviço
S3	*Simple Storage Service*
SQL	*Data Definition Language*
TC	*Trusted Coordinator*
TCCP	*Trusted Cloud Computing Platform*
TCP	*Trusted Computing Platform*
TI	Tecnologia da Informação
VM	Máquina Virtual

1 - INTRODUÇÃO

Com a evolução das diversas tecnologias computacionais, como a computação distribuída, computação pervasiva, computação em grade, internet e linguagens de programação, tornou-se possível o surgimento de novas áreas da computação. A venda de recursos computacionais sob demanda, de acordo com a necessidade do cliente é um desejo antigo que era limitado às grandes instituições capazes de adquirir recursos computacionais das grandes detentoras de poder computacional. Em consequência do desenvolvimento de novas tecnologias nos mais abrangentes campos da computação e a massificação do uso da internet, tem-se disponibilizados aplicativos web que podem ser acessados independentes de sua localização.

O desenvolvimento de tecnologias de virtualização viabiliza a venda sob demanda, de forma escalável, de recursos computacionais e infraestrutura, os quais são capazes de sustentar os aplicativos web. Surge assim a computação em nuvem, gerando uma tendência cada vez maior de aplicativos que podem ser acessados de forma eficiente, independente de sua localização. Com este surgimento cria-se a necessidade de repensar o modo como as aplicações são desenvolvidas e disponibilizadas aos usuários, ao mesmo tempo em que motiva o desenvolvimento de tecnologias capazes de dar suporte ao seu aprimoramento. A computação em nuvem vem ganhando cada vez mais força com a adesão de grandes empresas do ramo de Tecnologia da Informação, que vem realizando cada vez mais esforços no desenvolvimento de tecnologias para esse ambiente.

Desde que a IBM Corporation anunciou o seu programa de computação em nuvem no final do ano de 2007, outras grandes empresas de Tecnologia da Informação (TI) vem adotando nuvens progressivamente, por exemplo: O Google App Engine, que permite a criação e hospedagem de aplicativos da web com os mesmos sistemas que acionam os aplicativos do Google; o Amazon Web Services (AWS) da Amazon, que foi uma das primeiras empresas a proporcionar serviços em nuvem ao público; o Elastic Compute Cloud (EC2) da Amazon, que permite aos usuários alugar computadores virtuais nos quais podem executar suas próprias aplicações fornecendo um controle completo sobre seus recursos computacionais e permitindo a execução no ambiente computacional; o Simple Storage Service (S3) da Amazon, que permite o armazenamento de arquivos no serviço de armazenamento virtual; o iCloud da Apple e o Azure Services Platform da Microsoft, que

introduziram produtos de computação em nuvem (Zhang et al, 2010). No entanto, a computação em nuvem ainda apresenta riscos relacionados à segurança de dados nos seus diferentes aspectos, tais como, confidencialidade, integridade e autenticidade (Wang et al, 2010) (Uppoor et al, 2010) (Popovic et al, 2010).

A tecnologia de computação em nuvem tem como objetivo proporcionar serviços de TI sob demanda, sendo tarifada ou não conforme o uso, assim como os demais serviços básicos. Tendências anteriores à computação em nuvem foram limitadas a uma determinada classe de usuários ou focadas em tornar disponível uma demanda específica de recursos de TI, principalmente de informática (Miller, 2008). Esta tecnologia pretende atender mundialmente, sendo não só utilizada por grandes empresas que terceirizariam todos seus serviços de TI a outra empresa, como também pelo usuário que deseja hospedar seus documentos pessoais na internet. Este modelo de tecnologia permite não só a utilização de recursos de armazenamento e processamento, mas de todos os serviços computacionais.

Na computação em nuvem os recursos de TI são fornecidos como um serviço, permitindo que os usuários o acessem sem a necessidade de conhecimento da tecnologia utilizada. Assim, os usuários e as empresas passaram a acessar os serviços sob demanda e independente de localização, o que aumentou a quantidade de serviços disponíveis (Minqi, 2010). Com isso, os usuários estão movendo seus dados e aplicações para a nuvem, podendo acessá-los de forma simples e de qualquer local. A computação em nuvem surge da necessidade de construir infraestruturas de TI menos complexas em relação às tradicionais, onde os usuários têm que realizar instalação, configuração e atualização de sistemas de software. Além disso, recursos de infraestrutura são propensos a ficarem obsoletos rapidamente. Assim, a utilização de plataformas computacionais de terceiros é uma solução inteligente para os usuários lidarem com a infraestrutura de TI.

Computação em nuvem é um modelo de computação distribuída que deriva características da computação em grades, no que diz respeito à provisão de informação sob demanda para múltiplos usuários concorrentes (Minqi, 2010). Um provedor de serviço em nuvem oferece aplicações na nuvem sem que o usuário tenha que se preocupar com o local onde os serviços estão sediados ou como eles são oferecidos. Fatias do poder computacional dos nós da rede são oferecidas, reduzindo os custos para fornecer uma infraestrutura própria para prover os serviços. Os recursos são cedidos apenas durante o período de uso, reduzindo o consumo de energia quando a utilização não for mais necessária. A virtualização fornece a tecnologia base para muitas soluções em nuvem.

Além disso, em muitas soluções são oferecidos ambientes onde os usuários são capazes de escolher seus recursos virtualizados tais como linguagem de programação, sistema operacional e outros serviços personalizados. Os principais benefícios são: a redução dos custos de investimento em infraestrutura, dos custos operacionais e a escalabilidade para a provisão de serviços sob demanda.

Embora apresente vários benefícios, a computação em nuvem possui ainda alguns riscos relacionados com a integridade de dados, segurança, recuperação de dados, privacidade da informação, dentre outros, que não foram ainda solucionados, conforme discutido em (Wang et al, 2010) (Uppoor et al, 2010) (Popovic et al, 2010).

Neste trabalho são apresentados os conceitos da arquitetura de computação em nuvem e as questões relacionadas à segurança, privacidade e disponibilidade. Além disso, os conceitos de confiança e reputação computacional serão abordados. É proposto também um modelo de confiança para estabelecer um *ranking* entre os nós e permitir a troca confiável de arquivos entre seus pares em uma nuvem privada.

Em uma nuvem privada, a infraestrutura de nuvem é utilizada exclusivamente para atender as necessidades de uma organização, sendo esta nuvem local ou remota e administrada pela própria organização ou por terceiros. Neste modelo de implantação são empregadas políticas de acesso aos serviços. As técnicas utilizadas para prover tais características podem ser em nível de gerenciamento de redes, configurações dos provedores de serviços e a utilização de tecnologias de autenticação e autorização. O objetivo principal de uma nuvem privada é fornecer aos usuários locais uma infraestrutura ágil e flexível para suportar cargas de trabalho de serviços dentro de seu próprio domínio administrativo.

Implantar uma nuvem privada necessita mais do que tecnologia para que a mesma funcione. É necessário considerar aspectos como objetivos corporativos, planejamento de serviços, políticas, processos e papel das organizações.

O ambiente de computação em nuvem privada permite que seja trabalhado com um contexto específico de distribuição de arquivos, de modo que os arquivos tenham uma distribuição e disponibilidade desejada, sendo possível ter garantias do administrador da nuvem que seu acesso é restringido, bem como a identificação dos nós seja única e controlada.

No modelo proposto, a escolha do nó mais confiável é efetuada levando em consideração a sua disponibilidade. A seleção dos nós e a avaliação do seu valor de confiança determinarão se o nó é confiável ou não, a qual será realizada de acordo com o

espaço de armazenamento, sistema operacional, capacidade de processamento e enlace do nó. A confiança é estabelecida com base nas consultas e requisições realizadas entre os nós da nuvem privada.

1.1 - MOTIVAÇÃO

Apesar dos benefícios do uso de serviços sob demanda, ainda existem problemas relacionados com a segurança e confiança na troca de arquivos entre as máquinas em uma nuvem. Os trabalhos atuais abordando modelos de confiança (estudados e revisados) não tratam a representação e utilização da confiança na computação em nuvem para efetuar a troca confiável de arquivos entre os pares, sendo necessário um estudo e proposta de um modelo de confiança em nuvem para a troca de arquivos entre as mesmas em uma nuvem privada.

A elaboração de uma pesquisa detalhada, sua aplicação e a criação de um modelo que trate da confiança em ambientes de computação em nuvens é um aspecto motivacional importante porque abre um leque de discussões sobre confiança e a possibilidade de resolução de problemas atuais sobre o tema. Além disso, possibilita uma contribuição real e prática através de uma revisão atualizada sobre o tema, de uma implementação do modelo proposto e das análises decorrentes da implementação e dos resultados obtidos.

1.2 - OBJETIVOS DO TRABALHO

A computação em nuvem é uma área promissora a ser explorada através de pesquisas e análises experimentais. Além de manter o foco no uso da confiança em ambientes de computação em nuvem, os objetivos deste trabalho podem ser resumidos em:

a. Desenvolvimento de uma proposta de um modelo de confiança que possa representar a troca confiável de arquivos entre os pares em ambientes de computação em nuvem privada;
b. Prover a disponibilidade dos arquivos através do modelo de confiança proposto;
c. Simular o modelo proposto em uma nuvem privada utilizando uma plataforma aberta;
d. Abordar aspectos relacionados à segurança, confiança computacional e reputação para a garantia da troca de informações confiáveis em uma nuvem

privada, evitando a possibilidade de o arquivo não ser transmitido e ou alteração de informações na troca de arquivos entre os nós.

Assim, o principal objetivo deste trabalho, observando o estado da arte da confiança em computação em nuvem, está voltado para a elaboração/proposta de um modelo de confiança computacional na troca de arquivos entre pares em nuvens privadas, envolvendo métricas que sejam capazes de representar ou mapear o grau de confiança em um nó para realizar a troca de arquivos em uma nuvem.

1.3 - METODOLOGIA DE PESQUISA

A metodologia de pesquisa proposta foi dividida em fases, 1,2 e 3, para facilitar o entendimento do trabalho, conforme apresentado a seguir. A divisão em fases busca aprofundar o estudo relacionado ao tema e problema proposto neste trabalho, identificando os assuntos abordados pela comunidade acadêmica atualmente e as métricas a serem identificadas para o fornecimento e busca da confiança em nuvens privadas.

Fase 1: Realizar pesquisa bibliográfica, a qual é importante por ser um mecanismo que permite a identificação, leitura e análise de artigos relevantes ao desenvolvimento do projeto. Os artigos e documentos relacionados nesta fase foram analisados e registrados, para que o trabalho seja produtivo e dinâmico.

Fase 2: Obter informações sobre os problemas relacionados à segurança, reputação e confiança nas nuvens privadas, identificando as soluções conceituais exploradas e aplicadas, bem como a construção e proposta de um modelo de confiança para o ambiente de nuvem privada.

Fase 3: Simular e desenvolver o modelo proposto, analisando os resultados. Nesta fase serão efetuadas as conclusões e identificação das contribuições.

1.4 - CONTRIBUIÇÕES DO TRABALHO

Buscam-se com este trabalho as seguintes contribuições:

- Apresentação do estado da arte da computação em nuvem, levando em consideração aspectos que são importantes para definições do modelo de confiança;
- Apresentação dos trabalhos recentes empregando segurança, confiança e reputação em computação em nuvem;

- Apresentação de uma proposta de um modelo de confiança, permitindo que seja possível efetuar a troca confiável de arquivos entre os pares de um ambiente de computação em nuvem privada;
- Simulação do modelo proposto, aplicando à solução proposta em um cenário de uma nuvem privada, identificando os nós que possuem maior confiabilidade para a troca de arquivos.

1.5 - ORGANIZAÇÃO DO TRABALHO

Para um melhor entendimento deste trabalho, a sua organização é descrita a seguir.

O Capítulo 2 oferece uma revisão dos principais conceitos abordados, incluindo principalmente a computação em nuvem, confiança e a reputação, realizando um mapeamento da segurança, confiança e da reputação em um ambiente de computação em nuvem. Além disso, são apresentados os trabalhos correlatos, bem como os sistemas de distribuição de arquivos em nuvem e alguns dos problemas em aberto.

No Capítulo 3 é apresentado algumas considerações sobre confiança, segurança, privacidade, disponibilidade, confidencialidade e integridade. Além disso, tem-se a proposta do modelo de confiança para a troca confiável de arquivos entre os pares em um ambiente de computação em nuvem privada.

O Capítulo 4 apresenta as simulações realizadas e os resultados obtidos.

O Capítulo 5 conclui este trabalho, sinalizando algumas perspectivas possíveis, o fechamento dos resultados obtidos e os caminhos futuros que foram seguidos para a sequência deste trabalho.

2 – ESTADO DA ARTE E REVISÃO BIBLIOGRÁFICA

Este capítulo tem como foco a revisão dos principais conceitos de computação em nuvem, confiança, reputação e suas aplicações. Com o intuito de abranger o tema em um cenário amplo e ao mesmo tempo, ser possível a separação dos conceitos, assuntos correlatos e de assuntos similares, foi realizado uma divisão dos assuntos em tópicos específicos. Na seção 2.1 é abordado o tema de computação em nuvem. Na seção 2.2 é abordado os conceitos relacionados à confiança. Na seção 2.3 é abordado o tema reputação. Na seção 2.4 são apresentados os trabalhos relacionados. Na seção 2.5 são apresentados os principais sistemas de distribuição de arquivos em nuvem. Na seção 2.6 serão abordados alguns aspectos de segurança, tais como: confidencialidade, integridade, disponibilidade e privacidade para que um modelo de confiança seja capaz de trocar informações confiáveis entre os usuários de uma nuvem privada e por fim na seção 2.7 são apresentados os problemas em aberto relacionados à computação em nuvem.

2.1 - COMPUTAÇÃO EM NUVEM

Computação em nuvem refere-se ao uso, independente de plataforma e localização, das mais variadas aplicações por meio da internet com a mesma facilidade de tê-las instaladas nos computadores do usuário. Muitas definições formais têm sido propostas pela academia e indústria para computação em nuvem. Neste trabalho é adotado a seguinte definição: "Computação em nuvem é um modelo que permite acesso à rede sob demanda para um conjunto compartilhado de recursos computacionais configuráveis (por exemplo, redes, servidores, armazenamento, aplicações e serviços) que podem ser rapidamente configurados e liberados com um mínimo esforço de gerenciamento ou de interação com o prestador de serviços" (Mell, 2009). Esta definição inclui arquiteturas em nuvem, segurança e estratégias de implantação, uma vez que engloba todos os recursos necessários e oferecidos pela computação em nuvem (Mell, 2009).

A computação em nuvem, pela sua característica de agrupar diversas tecnologias (virtualização¸ computação distribuída, computação em grade, arquitetura orientada a serviços, entre outros) e não tratar apenas de um paradigma computacional, mas também de um modelo de negócios, requer a necessidade de atividades gerenciais de mais alto nível. Tais atividades de gerenciamento incluem:

a) Qual provedor de serviços selecionar;
b) Que tecnologia de virtualização utilizar;
c) Como disponibilizar os recursos virtuais;
d) Como monitorar seu adequado funcionamento e conformidade com o Acordo de Nível de Serviço (SLA) que tenha sido definido entre as partes envolvidas.

Estabelecer uma arquitetura e um sistema de monitoramento, que englobe todas estas necessidades e permita exercer as atividades usuais de planejamento, provisionamento, escalabilidade, segurança e confiança é uma agregação importante ao desenvolvimento e utilização da computação em nuvem. Chang *et al.* (2010) apresenta alguns desafios relacionados à segurança, privacidade, performance e disponibilidade em nuvem, tais como: perda e alteração de dados, *phishing*, *botnet* (executados remotamente em uma coleção de máquinas) que representam graves ameaças aos dados e software organizacionais.

2.1.1 - Características da Computação em Nuvem

Uma solução de computação em nuvem é composta de vários elementos, conforme apresentado na Figura 2.1. Estes elementos formam as três partes de uma solução de computação em nuvem: cliente, *datacenter* e servidores distribuídos (Velve *et al*., 2011). Cada elemento tem uma finalidade e possui um papel específico em oferecer um aplicativo funcional baseado em nuvem, como representado na Figura 2.1.

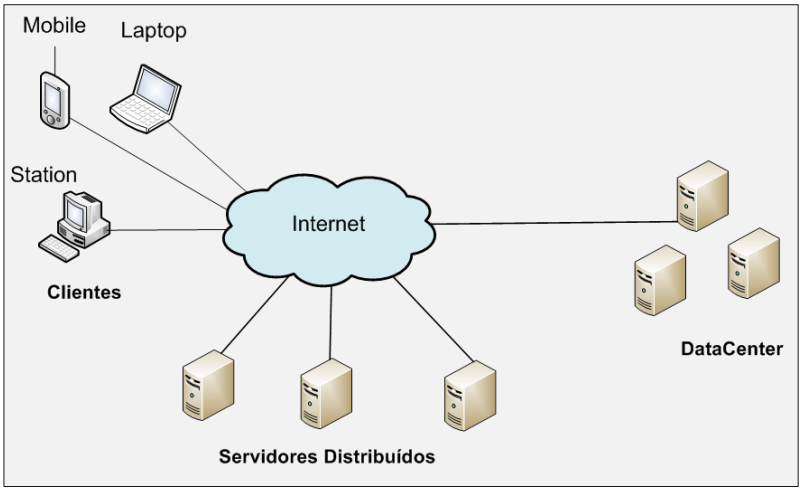

Figura 2.1 - Três Elementos da Solução de Computação em Nuvem (Velve *et al*., 2011).

A Tabela 2.1 apresenta a descrição dos elementos de uma solução de computação em nuvem.

Tabela 2.1 – Elementos de uma Solução de Computação em Nuvem (Velve *et al*., 2011).

Elemento	Descrição
Clientes	Os clientes são em uma arquitetura de computação em nuvem, exatamente o que eles são em uma rede local (LAN). São os dispositivos com os quais os usuários finais interagem para gerenciar sua informação na nuvem.
Data Center	É um conjunto de servidores onde o aplicativo (CRM, ERP, financeiro, etc) é armazenado. Uma tendência crescente no mundo da TI é a virtualização de servidores, isto é, o software pode ser instalado permitindo que vários servidores virtuais sejam usados.
Servidores Distribuídos	Os servidores não têm que estar alocados em um mesmo local. Normalmente, os servidores estão em diferentes posições geográficas, o que permite ao provedor de serviços maior flexibilidade nas opções e na segurança, por exemplo, a Amazon possui uma solução de nuvem no mundo inteiro. Se algo acontecer em um local, causando uma falha, o serviço poderá ser acessado através de outro local. Novos hardwares podem ser adicionados à nuvem, conforme a necessidade.

Uma das vantagens da computação em nuvem é a possibilidade de utilizar aplicações diretamente da Internet, sem que estas estejam instaladas no computador do usuário. Mas, há outras significativas vantagens e desvantagens (Miller, 2008), conforme Tabela 2.2.

Tabela 2.2 Vantagens e Desvantagens de Computação em Nuvem

Vantagens	Desvantagens
Baixo custo de infraestrutura de TI.	Requer uma conexão de rede constante.
Aumento do poder de computação.	Dependente da largura de banda da rede.
Capacidade de armazenamento ilimitada.	Recursos podem ser limitados.
Maior compatibilidade entre Sistemas Operacionais.	Os dados armazenados podem não estar seguros.
Facilidade na colaboração em grupo.	Se a nuvem perder os seus dados, não terá acesso às suas informações, não sendo característica somente da nuvem e sim inerente a qualquer sistema.
Acesso universal aos documentos.	

A computação em nuvem possui características e atributos que ainda estão sendo elaborados. As seguintes características próprias podem ser enumeradas (Badger, 2011):

1) **Oferta de serviços sob demanda**: alocação dinâmica dos serviços requisitados. Um consumidor pode unilateralmente prover capacidades computacionais, tais

como tempo de servidor e armazenamento em rede, conforme necessário, sem interação humana com o provedor dos serviços. O hardware e o software dentro de uma nuvem podem ser automaticamente reconfigurados, orquestrados e estas modificações são apresentadas de forma transparente para os usuários, que possuem perfis diferentes e, assim, podem personalizar os seus ambientes computacionais, por exemplo, instalação de software e configuração de rede para a definição de determinados privilégios.

2) **Amplo acesso aos recursos computacionais**: acesso por meio de diversos protocolos padronizados, para uma grande variedade de dispositivos como PCs, laptops, dispositivos móveis, dentre outros. O acesso é feito simplesmente via um navegador de internet. A interface de acesso a nuvem não obriga os usuários a mudar suas condições e ambientes de trabalho, como por exemplo, linguagens de programação e sistema operacional. O software cliente instalado localmente para o acesso a nuvem é leve, como um navegador de internet.
3) ***Pooling* de recursos:** os provedores de recursos computacionais são agrupados para atender vários consumidores através do modelo multi-inquilino (*multi-tenant*), com diferentes recursos físicos e virtuais atribuídos dinamicamente e novamente de acordo com a demanda do consumidor. Há um sentido de independência local em que o assinante geralmente não tem controle ou conhecimento sobre a localização exata dos recursos disponibilizados, mas pode ser capaz de especificar o local em um nível maior de abstração (por exemplo, estado, país, ou *Datacenter*). Exemplos de recursos incluem o armazenamento, processamento, memória, largura de banda e máquinas virtuais.
4) **Transparência**: o usuário não precisa conhecer a localização física dos recursos computacionais oferecidos.
5) **Escalabilidade**: os serviços devem ser alocados e desalocados rapidamente, apenas no decorrer da requisição do usuário. Em alguns casos, automaticamente, escalar e rapidamente liberar. Para o consumidor, as capacidades disponíveis para provisionamento, muitas vezes parecem ser ilimitadas e podem ser compradas em qualquer quantidade e a qualquer momento.
6) **Gerência**: a infraestrutura deve oferecer mecanismos para a gerência de recursos, de armazenagem, de processamento e largura de banda, dentre outros.
7) **Serviço medido:** sistemas de nuvem automaticamente controlam e otimizam o uso dos recursos, aproveitando uma capacidade de medição em algum nível de

abstração apropriado ao tipo de serviço (por exemplo, armazenamento, processamento, largura de banda e conta de usuário ativo). O uso de recursos pode ser monitorado, controlado e relatado, proporcionando transparência para o provedor e consumidor do serviço utilizado.

Independente da aplicação, com a computação em nuvem o usuário não necessita conhecer toda a estrutura, ou seja, ele não precisa saber quantos servidores executam determinada ferramenta, bem como as configurações de hardware e software utilizadas, como o escalonamento é feito, onde está a localização física do *Datacenter*, etc. O que importa ao usuário é saber que a aplicação está disponível na nuvem, não importa de que forma.

Segundo Chen (2009) a computação em nuvem combina o modelo de dados compartilhado e o modelo de serviço estatístico, possuindo três características básicas:

1) Arquitetura da Infraestrutura de Hardware: baseado em *clusters*, em larga escala e baixo custo. A infraestrutura de computação em nuvem é composta por um grande número de servidores de baixo custo, ou mesmo a arquitetura de servidores X86;
2) Desenvolvimento colaborativo de serviços básicos e aplicativos para alcançar a utilização máxima dos recursos. Desta forma, a construção do aplicativo é aperfeiçoada. No modelo de computação tradicional, as aplicações tornam-se completamente dependente do serviço básico;
3) O problema da redundância entre vários servidores de baixo custo é resolvido através do software. Por causa do uso de um grande número de servidores de baixo custo, falhas entre os nós não podem ser ignoradas, assim, a questão da tolerância a falhas entre os nós deve ser levada em consideração, ao projetar o software.

2.1.2 - Arquitetura da Computação em Nuvem

A arquitetura de computação em nuvem é baseada em camadas, sendo que cada uma trata de uma particularidade na disponibilização de recursos para as aplicações. Em geral é dividida em duas camadas, camada de recursos inferior e a camada de serviços superior. A camada inferior é a de infraestrutura física, responsável pela virtualização dos recursos sob a forma de armazenamento e computação. A camada superior provê serviços específicos. Estas camadas podem ter seu gerenciamento ou monitoramento de forma independente

uma da outra, melhorando a flexibilidade, reuso e escalabilidade. A Figura 2.2 apresenta as camadas da arquitetura da computação em nuvem (Zhang *et al*., 2010).

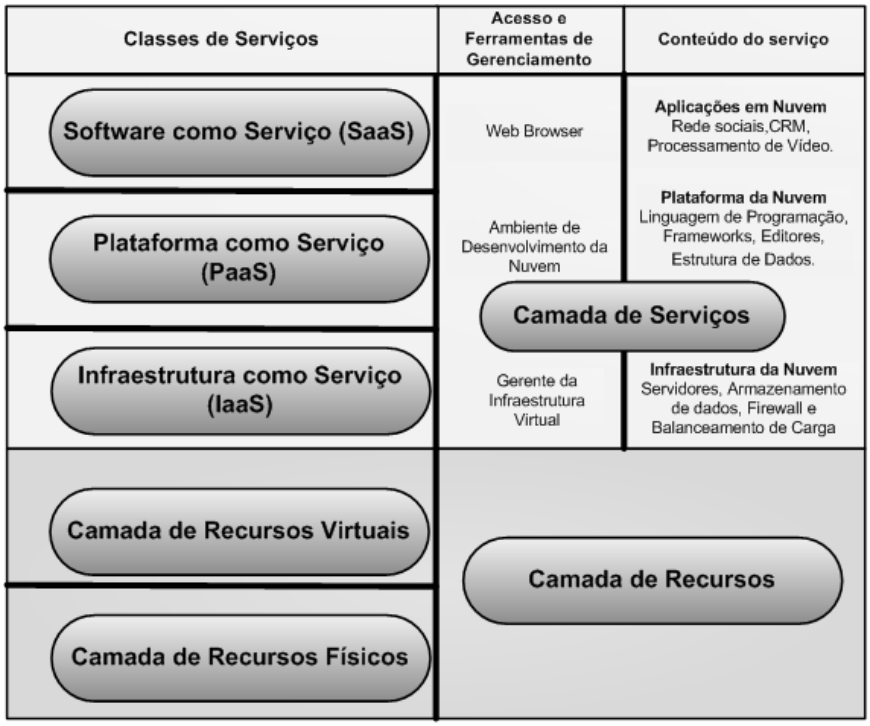

Figura 2.2 - Arquitetura da Computação em Nuvem (Zhang *et al*., 2010)

A seguir é explicado cada camada de serviço.

2.1.2.1 - Software Como Serviço

O Software como um Serviço (*Software as a Service* - SaaS) provê todas as funções de uma aplicação tradicional, mas por acesso via internet. O modelo SaaS elimina preocupações com servidores de aplicativos, sistemas operacionais, armazenamento, desenvolvimento de aplicações, etc. Assim, os desenvolvedores concentram-se em inovação e não na infraestrutura, permitindo o desenvolvimento rápido de sistemas de software dos quais participam.

O SaaS reduz os custos, pois é dispensada a aquisição de licenças de sistemas de software, os usuários usam o serviço sob demanda. Como o software é acessado via Web pelos usuários de qualquer lugar e a qualquer momento, permite maior integração entre unidades de uma mesma organização ou outros serviços de software. Assim, novos recursos podem ser incorporados automaticamente aos sistemas de software sem que os usuários percebam estas ações, tornando transparente a evolução e atualização dos

sistemas. Exemplo de SaaS (Minqi, 2010): Google Docs, Serviços de *Customer Relationship Management* (CRM), o *Photoshop.com* da *Adobe*, o jogo *Quake Live* e conversores como o *Free PDF Converter* e *Cellsea*, dentre outros.

2.1.2.2 - Plataforma Como Serviço

A Plataforma como um Serviço (*Platform as a Service* - PaaS) é a camada do meio da camada de serviços em nuvem, ou seja, entre o SaaS e o IaaS. A PaaS oferece aos usuários um melhor desempenho, serviços de software e hardware personalizados sem necessidade de downloads ou instalações. A PaaS oferece uma infraestrutura com alto nível de integração para implementar e testar aplicações em nuvem. O usuário não administra ou controla a infraestrutura, incluindo rede, servidores, sistemas operacionais ou armazenamento, mas tem controle sobre as aplicações implantadas e, possivelmente, as configurações das aplicações disponibilizadas nesta infraestrutura (Mell, 2011).

A PaaS fornece um sistema operacional, linguagens de programação e ambientes de desenvolvimento para as aplicações, auxiliando a implementação de sistemas de software, uma vez que contém ferramentas de desenvolvimento e colaboração entre desenvolvedores. Do ponto de vista do negócio, a PaaS permite aos usuários utilizar serviços de terceiros, aumentando o uso do modelo de suporte no qual os usuários se inscrevem para solicitações de serviços de TI ou para resoluções de problemas pela Web. Com isso, pode-se melhorar o gerenciamento do trabalho e as responsabilidades das equipes de TI das empresas. Exemplo de PaaS (Minqi, 2010): *Azure Services Platform* (*Azure*), *Force.com*, *EngineYard* e *Google App Engine*.

2.1.2.3 - Infraestrutura Como Serviço

A Infraestrutura como um Serviço (*Infrastructure as a Service* - IaaS) é a parte responsável por prover toda a infraestrutura necessária para a PaaS e o SaaS. Seu principal objetivo é tornar mais fácil e acessível o fornecimento de recursos, tais como servidores, rede, armazenamento e outros recursos de computação fundamentais para construir um ambiente sob demanda, que podem incluir sistemas operacionais e aplicativos.

A IaaS possui uma interface única para administração da infraestrutura, Interface de Programação de Aplicativos (*Application Programming Interface* - API) para interação com hosts, switches, roteadores e o suporte para a adição de novos equipamentos de forma simples e transparente. Em geral, o usuário não administra ou controla a infraestrutura da

nuvem, mas tem controle sobre os sistemas operacionais, armazenamento e aplicativos implantados, e, eventualmente, seleciona componentes de rede, tais como *firewalls*.

O termo IaaS refere-se a uma infraestrutura computacional baseada em técnicas de virtualização de recursos de computação. Esta infraestrutura pode escalar dinamicamente, aumentando ou diminuindo os recursos de acordo com as necessidades das aplicações. O principal benefício da IaaS é o esquema de pagamento baseado no uso (Mell, 2011). Exemplo de IaaS (Minqi, 2010): *Amazon Elastic Cloud Computing* (EC2) e *Elastic Utility Computing Architecture Linking Your Programs To Useful Systems (Eucalyptus*).

2.1.3 - Papéis na Computação em Nuvem

Os papéis definem as responsabilidades, acesso e perfil para os diferentes usuários que fazem parte e estão envolvidos em uma solução de computação em nuvem. São classificados como atores dos modelos de serviços, de acordo com os papéis desempenhados por eles. A Figura 2.3 apresenta os papéis definidos na camada de serviços (Marinos, 2009).

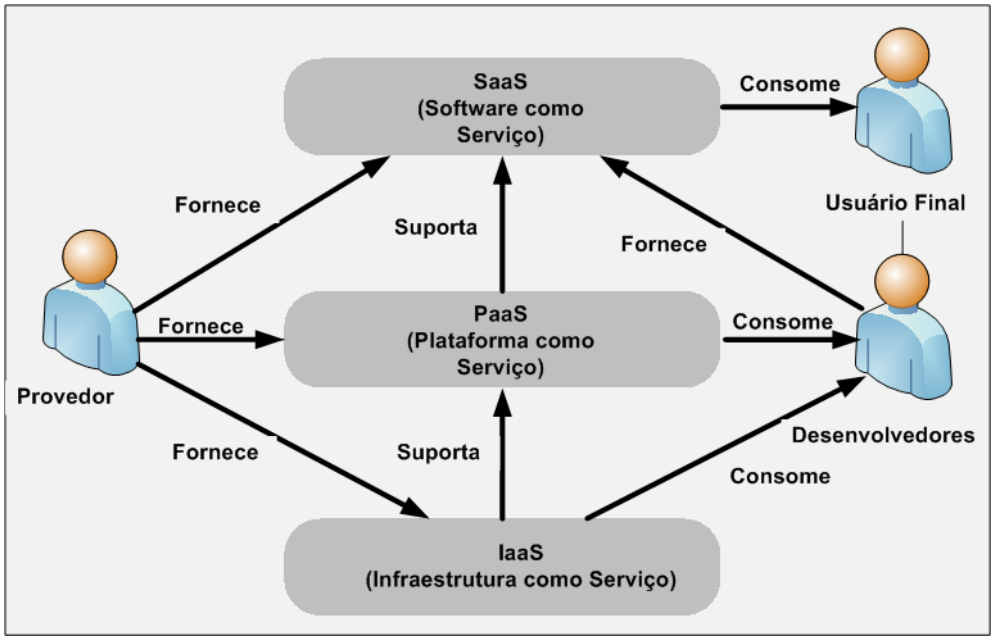

Figura 2.3 - Papéis na Computação em Nuvem (Marinos, 2009)

O provedor é responsável por disponibilizar, gerenciar e monitorar toda a estrutura da solução de computação em nuvem, deixando o desenvolvedor e o usuário final sem esse tipo de responsabilidade e fornecendo serviços nos três modelos de serviços.

Os desenvolvedores utilizam os recursos fornecidos pelo IaaS e PaaS e disponibilizam serviços para os usuários finais. A organização em papéis ajuda a definir os atores e os seus diferentes interesses. Os atores podem assumir vários papéis ao mesmo tempo de acordo com os interesses, sendo que apenas o provedor fornece suporte a todos os modelos de serviços.

2.1.4 - Implantação da Computação em Nuvem

Em relação ao acesso e disponibilidade de ambientes de computação em nuvem, têm-se diferentes tipos de modelos de implantação (Mell, 2011). A restrição ou liberação de acesso depende do processo de negócio, do tipo de informação e do nível de visão organizacional. Em algumas organizações é necessário um ambiente mais restrito, onde somente usuários devidamente autorizados possam acessar e utilizar determinados recursos dos serviços providos no seu ambiente de computação em nuvem. Os modelos de implantação da computação em nuvem podem ser divididos em nuvem pública, privada, comunidade e híbrida (Mell, 2011), que estão resumidos na Tabela 2.3.

Tabela 2.3 - Modelos de Implantação de Serviços em Nuvem (Mell, 2011)

Modelo da Nuvem	Descrição
Privada	Neste modelo, a infraestrutura de nuvem é utilizada exclusivamente para uma organização, sendo esta nuvem local ou remota e administrada pela própria empresa ou por terceiros. São empregadas políticas de acesso aos serviços. As técnicas utilizadas para prover as características podem ser em nível de gerenciamento de redes, configurações dos provedores de serviços e a utilização de tecnologias de autenticação e autorização.
Pública	No modelo de implantação de nuvem pública, a infraestrutura de nuvem é disponibilizada para o público em geral, sendo acessado por qualquer usuário que conheça a localização do serviço. Neste modelo não podem ser aplicadas restrições de acesso quanto ao gerenciamento de redes, bem como utilizar técnicas para autenticação e autorização.
Comunidade	No modelo de implantação de nuvem comunidade ocorre o compartilhamento por diversas organizações de uma nuvem, sendo esta suportada por uma comunidade específica que compartilha os mesmos interesses, tais como a missão, requisitos e política de segurança e considerações sobre flexibilidade. Este modelo pode existir localmente ou remotamente, sendo geralmente administrado por alguma organização da comunidade ou por terceiros.
Híbrida	No modelo de implantação de nuvem híbrida, existe uma

	composição de duas ou mais nuvens, que podem ser privadas, comunidade ou pública e que permanecem como entidades únicas, ligadas por uma tecnologia padronizada ou proprietária que permite a portabilidade de dados e aplicações.

A computação em nuvem pode apresentar alguns problemas relacionados com a proteção, a confiança, a privacidade e a segurança de dados dos usuários. Atualmente um dos fatores mais importantes para o sucesso da computação em nuvem é a confiança e segurança (Qiang, 2011).

2.2 - CONFIANÇA

Os conceitos de confiança, modelo de confiança e gerenciamento da confiança têm sido objeto de vários projetos de pesquisa recentes. A confiança é reconhecida como um aspecto importante para a tomada de decisão em aplicações distribuídas e auto-organizada (Marsh, 1994) (Beth *et al*., 1994). Além disso, é uma área de pesquisa que também está voltada para a solução de problemas que envolvem a segurança da informação em diversos ambientes computacionais.

O conceito de confiança define, sob o ponto de vista de um indivíduo, o quanto ele confia em outro indivíduo. Para que um indivíduo seja confiável, é necessário que ele tenha atitudes positivas (honestas e colaborativas) com relação às entidades que dele dependem. Sendo assim, a confiabilidade é a capacidade de um indivíduo ser confiável e a confiança é uma consequência da confiabilidade.

As principais definições de confiança voltadas para o aspecto humano se baseiam nas relações entre indivíduos, transparecendo de forma clara a relação entre confiança e o sentimento de segurança (Gambetta, 2000) (Lamsal, 2006). Desta forma, a confiança no aspecto humano está relacionada com o sentimento de segurança voltada para um determinado contexto, para a satisfação de uma expectativa de uma solução que seja provável de ser resolvida (Gambetta, 2000) (Lamsal, 2006). O processo de confiar em um indivíduo é um resultado de várias análises que em conjunto gera a definição de confiança.

A confiança (ou, simetricamente, a desconfiança) é um nível particular de probabilidade subjetiva, na qual um agente acredita que outro agente ou grupo de agentes irão realizar uma determinada ação, que está sujeita a um monitoramento (ou independente

da sua capacidade de monitorá-la) e em um contexto em que ela afeta a sua própria ação (Gambetta, 2000).

A confiança ainda é definida em Gambetta (2000) como o conceito social mais importante que auxiliam seres humanos a cooperar em seu ambiente social e está presente em todas as iterações humanas. De uma forma geral, sem a confiança (em outros seres humanos, agentes, organizações, etc.) não existe cooperação e conseqüentemente não existe uma sociedade. De uma maneira análoga, a confiança pode ser tratada como uma probabilidade de comportamento de que um agente realize uma determinada ação esperada por outro agente.

Um agente pode fazer a verificação da execução de uma ação solicitada (se estiver dentro de sua capacidade), e dentro de um contexto que a realização da ação esperada afetará a própria ação do agente em si (envolvendo uma tomada de decisão). Assim, se alguém é confiável, significa que existe uma probabilidade alta o suficiente de que esta pessoa irá executar uma ação, considerada benéfica de alguma forma, para que seja considerada a cooperação com ela. Em uma situação contrária, simplesmente acredita-se que a probabilidade é baixa o suficiente para que a cooperação seja evitada (Albuquerque, 2008).

Em Gambetta (2000) é proposto que a confiança tenha uma relação com a cooperação, de forma que a cooperação tenha importância com a aquisição da confiança. Se a confiança é unilateral a cooperação não pode obter sucesso. Por exemplo, se só existe a desconfiança entre dois agentes, então não pode existir nenhuma cooperação entre eles, logo não podem realizar uma operação em conjunto para uma solução de um problema. Então de maneira análoga, se existe um alto nível de confiança, é muito provável que exista uma alta cooperação entre os agentes para a solução de um problema particular.

Josang et al (2007) definem que a confiança é a probabilidade subjetiva pela qual um indivíduo, A, espera que outro indivíduo, B execute uma determinada ação da qual o bem-estar de A dependa. Esta definição inclui o conceito de dependência e da confiabilidade (probabilidade) da parte confiável, como vista pela parte confiante.

Na utilização da confiança, existe a perspectiva de que uma entidade P solicite informação de uma entidade Q a uma entidade R. Suponha que a entidade P precise de uma informação sobre uma entidade com quem ela ainda não se relacionou (entidade S), P pode perguntar para as entidades com que possuem relacionamentos se alguma delas conhece a entidade S, e qual a opinião delas sobre S (experiências/relacionamentos já

realizadas com a entidade S), fornecendo uma idéia da reputação da entidade S em relação à entidade consultada.

Em um cenário onde uma entidade conhece várias outras entidades, mas existe uma entidade que não conhece uma determinada entidade (entidade R não conhece a entidade Z), ela pode enviar uma pergunta sobre a referida entidade desconhecida às entidades que possuem relações com ela e aguardar as suas respostas. Se uma das entidades conhecerem a entidade investigada, retornarão a resposta para a entidade requisitante relatando a sua opinião em relação à mesma, conforme demonstrado na Figura 2.4. A partir das opiniões sobre o comportamento de uma entidade pode-se realizar o cálculo de confiança, baseado em um modelo, e a partir do resultado obtido, tomar uma decisão de relacionamento, que define se uma entidade irá se relacionar ou não com uma determinada entidade, dentro de um determinado contexto.

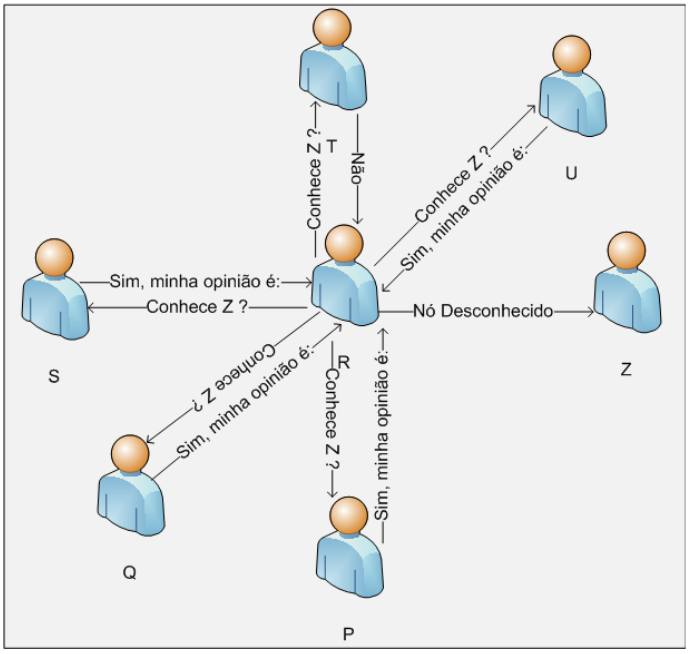

Figura 2.4 - Relação de Confiança

Patel (2007) define que a visão de confiança é modular, em virtude das suas diversas características. A confiança é calculada seguindo uma série de observações e implicações que versam desde a confiança adquirida unicamente por um agente, como a confiança calculada baseada em situações passadas e obtidas em um contexto social através de organizações e relacionamentos.

Como pode ser observada na Tabela 2.4 a confiança apresenta características básicas.

Tabela 2.4 - Características Básicas da Confiança (Albuquerque, 2008)

Característica	Exemplo
A confiança é relativa a um determinado contexto ou situação.	**A** pode confiar em **B** para lhe oferecer uma carona. Entretanto, **A** não confia em **B** o suficiente para dirigir seu carro.
A confiança tem um aspecto direcional.	**A** pode confiar em **B**, entretanto **B** pode não confiar em **A**.
A confiança é passível de ser mensurada.	A pode confiar mais em **B** do que confia em **C**.
A confiança possui aspecto temporal e evolutivo.	A confiança que **A** tem em **B** pode aumentar ou diminuir conforme **A** e **B** interagem.
A confiança pode ser influenciada por uma recomendação.	**A** que já confia em **B**, passa a ter certa confiança em **C**, que por sua vez lhe foi apresentado por **B**.
A confiança não é transitiva.	Se **A** confia em **B** e **B** confia em **A** e **C**, não significa dizer que **A** confia em **C**.

Na literatura, Marsh está entre os primeiros a estudar a confiança computacional. Marsh (1994) apresenta os conceitos de confiança, um formalismo para a implementação da confiança, e aplica um modelo de confiança para um sistema de Inteligência Artificial Distribuída (DAI), a fim de permitir que os agentes tomem decisões baseadas em confiança.

Segundo Abdul et al (1998) a maioria dos estudiosos chegaram a um consenso de que a confiança deve ter três características importantes:

1) **Subjetividade** – diferentes entidades, com uma mesma visão das coisas, serão afetadas pela variação de alguns fatores tais como suas preferências individuais;

2) **A probabilidade esperada** – o grau de confiança pode ser extraído e formalizado como a probabilidade estimada de um determinado evento;

3) **Relevância** – confiança é um aspecto de coisas, de conteúdo específico.

2.2.1 - Tipos de Confiança

Nesta seção é apresentado os tipos de confiança referenciados pelos diversos autores pesquisados durante a elaboração deste trabalho.

2.2.1.1 - Confiança Direta

Confiar em uma entidade diretamente significa acreditar em suas capacidades no que diz respeito a uma dada classe de confiança. A recomendação de confiança expressa a crença na capacidade de uma entidade decidir se uma outra entidade é de confiança em uma dada classe de confiança e na sua honestidade quando recomendada por uma terceira entidade. Beth *et al*., (1994) propõe um modelo de confiança direta, a qual é definida pela equação 2.1:

$$P\ confia_x^{seq}\ Q\ valor\ V \tag{2.1}$$

Uma relação de confiança direta existe se todas as experiências com Q com relação à classe de confiança X que P conhece são experiências positivas. O *seq* é a seqüência de entidades que mediaram as experiências (o caminho recomendado) com exceção de P e Q. *V* é o valor de uma relação de confiança, que é uma estimativa da probabilidade de que Q se comporte bem, sendo confiável. Este valor é baseado no número de experiências positivas com Q que P conhece.

Dado que *p* seja o número de experiências positivas com Q, que P conhece, na classe de confiança x, então o valor (V_z) dessas experiências é calculado pela equação 2.2:

$$V_{z(p)} = 1 - \alpha^p \tag{2.2}$$

Esse valor é a probabilidade de que Q seja mais confiável que o valor α, baseando nas informações que P possui sobre Q. Esta relação de confiança reflete a expectativa de que Q seja confiável para P dentro de uma perspectiva da execução de uma única tarefa, isto é, uma tarefa de valor 1. α deve ser bem mensurado para que as estimativas sejam seguras o suficiente.

O modelo de confiança direta proposto por Patel (Albuquerque, 2008), trabalha com a idéia de apresentação de valores específicos utilizando uma visão probabilística. Neste modelo a confiança está diretamente ligada com uma alta probabilidade de que uma entidade executará uma ação em particular que está relacionada com o cumprimento de determinadas ações durante uma iteração. A probabilidade neste caso é extremamente subjetiva porque só pode ser assumida por um ponto de vista individual da entidade, fundamentado em suas experiências pessoais.

Patel (2007) sinaliza pontos importantes e que são necessários na formação de um modelo de confiança. Os seus requisitos envolvem basicamente funcionalidades ou aspectos que devem ser tratados em um modelo de confiança. Estes aspectos são resumidos na Tabela 2.5.

Tabela 2.5 - Requisitos Gerais de um Modelo de Confiança (Patel, 2007)

#	Requisito	Descrição
1	Modelo escalável	O modelo deve ser escalável, o seu desempenho não pode ser afetada pelo número de entidades adicionadas a ele.
2	Modelo descentralizado	O modelo deve ser robusto e continuar seu funcionamento, mesmo em situações de falhas na rede.
3	Distinguir entre entidade e ambiente.	Deve ser capaz de fazer a distinção entre o papel de uma entidade e o papel do ambiente percebido pela entidade, principalmente pelas características não determinísticas de uma rede que normalmente deverá ser inserido.
4	Calcular confiança direta	O modelo deve permitir que um agente realize o cálculo de um nível de confiança para um agente de iteração potencial, baseado em experiências passadas que já teve com esta entidade potencial.
5	Calcular a reputação	Caso não tenha experiência prévia, o modelo deve ser capaz de permitir que a entidade faça o cálculo do nível de confiança para uma entidade de iteração potencial baseado nas opiniões de outras entidades.
6	Encontrar reputação	Oferecer mecanismos que permitam a identificação de fontes de opiniões em potencial, além dos protocolos eficazes para obter as mesmas.
7	Incentivar provimento de opiniões	Para a realização do requisito #5, a entidade deve prover razões para que outras entidades dêem sua opinião sobre outra.
8	Ajustar opiniões	Prover mecanismos para ajustar as opiniões de outras entidades (confiabilidade) de tal forma que não seja enganado por opiniões falsas.
9	Guardar histórico de iterações	De acordo com a necessidade de novos cálculos de níveis de confiança, a entidade deve manter uma base com o histórico de situações passadas com outras entidades e respectiva opinião sobre o fato ocorrido.
10	Usar informações sociais	Em determinadas organizações, o uso de informações sociais se torna fonte de informações que pode auxiliar no cálculo da confiança.
11	Valor de confiança dinâmico	O valor da confiança deve ser ajustável de acordo com a opinião formada da entidade sobre outra. A medida que a opinião muda, o valor de confiança também deverá ser mudado, dependendo da situação em si.
12	Consenso no nível de confiança	Deve permitir que um grupo de entidades alcance um consenso sobre o nível de confiança em um determinado indivíduo interno ou externo ao grupo.
13	Nível de confiança do grupo	Em alguns casos é necessário avaliar ou ter o nível de confiança de um grupo ou de seus membros dentro do grupo.
14	Troca efetiva de opiniões	Deve prover uma forma que permita as entidades trocarem informações de maneira eficiente e rápida.

#	Requisito	Descrição
15	Distribuir informações sobre confiança	Dado o requerimento #2, as informações sobre confiança não podem ficar guardadas unicamente em um ponto central.
16	Dependente de contexto	Permitir que as entidades troquem informações sobre um contexto em si e os níveis de confiança obtidos de forma que possam avaliar o contexto e o nível de confiança que foi obtido.

2.2.1.2 - Recomendação de Confiança

A recomendação de confiança segundo Beth *et al.*, (1994) é definida pela equação:

$$\text{P trusts. rec}_{\text{x}}^{\text{seq}} \text{ Q when. path } S_p \text{ when. target } S_t \text{ value V} \quad (2.3)$$

Uma relação de recomendação de confiança existe se P está disposto a aceitar relatos de Q sobre experiências com entidades terceiras com relação à classe de confiança x. Essa confiança é restrita as experiências com entidades em S_t (conjunto de restrições alvo) mediada pelas entidades em S_p (o caminho do conjunto de restrições). seq é a seqüência de entidades que mediaram a recomendação de confiança. V é o valor da relação de confiança. Representa a porção de experiências oferecidas que P está disposto a aceitar de Q e é baseado nas experiências que P teve com as entidades recomendadas por Q.

Dado um número de experiências positivas e negativas P e N, respectivamente com a entidade recomendada, o valor de confiança recomendado V_r é calculado de acordo com a equação 2.4:

$$v_\tau(p,n) = \begin{cases} 1 - \alpha^{p-n} & if \;\; p > n \\ 0 & else \end{cases} \quad (2.4)$$

Este valor pode ser considerado como o grau de similaridade entre P e Q, levando em conta que entidades diferentes podem ter experiências diferentes com entidades terceiras. Uma entidade que normalmente envia mensagens não classificadas será frequentemente menos confrontada com deslealdade que uma entidade que transmite informações secretas. Dada tal dissimilaridade, a última entidade não levará muito em consideração as experiências relatadas pela entidade anterior em relação as informações descritas sobre as outras entidades. As experiências com a entidade recomendada são formadas pela experiência com a entidade que recomenda. Se uma entidade recomendada se comporta bem, temos a experiência de uma recomendação valiosa. Igualmente, se a

recomendação é questionável, não poderá ser deduzido que a entidade que recomendou tenha mentido sobre o seu ponto de vista.

2.2.1.3 - Confiança Indireta

Não existe um consenso sobre a definição da confiança indireta, diversos autores a define, fazendo uma relação com outros conceitos (Patel, 2007), podendo ser tratada como o conhecimento das capacidades e confiabilidade de uma entidade por meio de entidades terceiras.

Patel (2007) descreve a reputação como um mecanismo para alcançar a confiança indireta, onde a reputação pode ser definida como um conceito multifacetado onde uma única entidade pode ter diferentes valores de reputação, cada uma para um contexto específico.

2.2.1.4 - Confiança Situacional

Marsh (1994) dividiu a confiança em categorias. Nesta definição, a confiança situacional é a mais importante, quando consideramos a confiança em situações cooperativistas. Para estimar a confiança situacional X, será necessário considerar vários aspectos da situação. Para Marsh (1994) a confiança possui os seguintes tipos, conforme a Tabela 2.6.

Tabela 2.6 - Tipos de Confiança Marsh (1994)

Tipo de Confiança	Descrição
Básica	É um valor que indica a disposição de uma entidade a confiar. Este valor é calculado com base no histórico das interações da entidade, representada por T_x, para a confiança básica de uma entidade X. O valor de está no intervalo [-1,1]. Assim, -1 ≤ T_x < 1.
Geral	É o valor que indica o quanto uma entidade X confia em uma entidade Y no tempo t, representada por $T_x(y)^t$, com valores entre [-1,1], onde o valor 0 é um valor neutro. O valor -1 representa uma confiança negativa, ou desconfiança total. $-1 \leq T_x(y)^t < +1$.
Cega	Considera-se o valor -1 para total desconfiança. Confiança cega implica em confiar em uma entidade sem uma observação ou análise do contexto. Para existir confiança é necessário que existam considerações e avaliações no contexto em que se vai confiar. O valor -1 é questionável, pois quando o valor -1 é alcançado pode ser entendido que ninguém é confiável. Não existindo confiança ou desconfiança total.
Não confiar e desconfiar	Os dois conceitos: confiança zero (não confiar) e desconfiar, não são a mesma coisa. São representados pelos valores 0 e -1. Zero pode significar que a entidade X não conhece a entidade Y

Tipo de Confiança	Descrição
	ou que X confiava em Y, mas o valor da confiança foi reduzido devido ao seu comportamento.
Situacional	Dado dois agentes X, Y ∈ A, e uma situação α, a confiança situacional $T_x(y,\alpha)^t$ representa a confiança que X tem em Y em uma situação α no tempo t. Os valores variam entre [−1, +1). Considera-se que a confiança está nas pessoas e não nas suas ações.

Marsh (1994) define a confiança situacional, sem considerar o tempo como a equação:

$$T_x(y,\alpha) = U_x(\alpha) \times I_x(\alpha) \times \widehat{T_x(y)} \quad (2.5)$$

Onde $T_x(y,\alpha)$ é a confiança que X tem em Y em uma situação α. $U_x(\alpha)$ é a utilidade da situação α para X, $I_x(\alpha)$ é a importância para X da situação e $\widehat{T_x(y)}$ é a confiança média que X tem em Y.

Marsh (1994) apresenta uma extensão da confiança situacional dada por uma estimativa do quanto X é confiável por Y, denotada por $(\widehat{T_y(x)})^x$, descrita na equação 2.6.

$$T_x(y,\alpha) = \left(\left(U_x(\alpha) + \widehat{T_x}(y)\right) \times \left(I_x(\alpha)\right) \times \widehat{T_x(y)}\right)^x \quad (2.6)$$

$(\widehat{T_y(x)})^x$ Representa o quanto X é confiável do ponto de vista de Y. A estimativa de X sobre a opinião de Y é importante, assim como a estimativa de X sobre a estimativa de Y em relação a X também tem a sua importância, levando a uma recursividade infinita, não possuindo sentido em uma aplicação real.

2.3 - REPUTAÇÃO

A reputação pode ser definida em um cenário onde não se possui informação o suficiente para realizar a inferência de que uma entidade é ou não confiável (Patel, 2007), e para alcançar este valor de inferência, uma entidade questiona sobre a opinião de outras entidades. A partir das informações das entidades questionadas, a entidade requisitante realiza o cálculo da reputação a partir de suas próprias informações, que é baseada em seus valores de confiança e nas informações obtidas das entidades terceiras (seu grau de confiança nas mesmas). Com as informações necessárias, a entidade avalia o contexto da situação em si, sendo capaz de chegar a um valor de reputação. O cálculo da reputação é obtido através da análise do comportamento de uma entidade ao longo do tempo.

A reputação no cenário computacional, segundo as revisões de trabalhos correlatos sobre confiança, indica que a mesma pode ter uma forte influência no cálculo da confiança (Patel, 2007) e (Gambetta, 2000), permitindo que a confiança esteja interligada com a reputação na geração dos valores de confiança e que estes valores sejam objeto não só da percepção do comportamento de uma entidade, mas também de avaliação própria por parte de quem tem interesse em algum tipo de iteração em um determinado contexto.

Segundo o modelo de Patel (2007), a maneira mais confiável de prever o comportamento de uma entidade é a partir da avaliação do histórico da interação direta com esta entidade. O histórico de interações de α_3 com α_2 no tempo t pode ser representado pela equação 2.7:

$$R^t{}_{a3,a2} = (m^t_{a3,a2}, n^t_{a3,a2}) \tag{2.7}$$

De maneira análoga, a opinião de a_3 com relação à a_2 é dada pela equação 2.8:

$$\hat{R}^t_{a3,a2=} (\hat{m}^t_{a3,a2}, \hat{n}^t_{a3,a2)} \tag{2.8}$$

É importante notar que em geral $R^t_{a3,a2} \neq \hat{R}^t_{a3,a2}$ no modelo, pois a opinião provida não é imparcial (possui a tendência positiva ou negativa de alterar o que seria de fato resultado da interação), e caso o opinante seja honesto, $R^t_{a3,a2} = R^t_{a3,a2}$.

A entidade α_1 deve calcular o valor de reputação $T^r_{\alpha 1,\alpha 2}$ em relação à α_2 pelas opiniões coletadas de outras entidades. As interações bem sucedidas e as mal sucedidas precisam ser enumeradas e somadas resultando nos valores de $N_{\alpha 1,\alpha 2}$ e $M_{\alpha 1,\alpha 2}$, os quais serão utilizados para calcular os parâmetros da distribuição beta utilizado no modelo. Esses parâmetros servirão para o cálculo da reputação, conforme demonstrado nas equações 2.9, 2.10 e 2.11:

$$N_{a1,a2} = \sum_{k=0}^{p} \hat{n}_{ak,a2}\,, M_{a1,a2} = \sum_{k=0}^{p} \hat{m}_{ak,a2}\,, onde\ p\ \text{é}\ o\ número\ de\ relatos \tag{2.9}$$

$$\hat{\alpha} = M_{a1,a2} + 1\ e\ \hat{\beta} = N_{a1,a2} + 1 \tag{2.10}$$

$$T^r_{a1,a2} = \frac{\hat{\alpha}}{\hat{\alpha} + \hat{\beta}} \tag{2.11}$$

Sabendo como calcular a reputação, é possível criar o agente de reputação (RTA – *Reputation Trust Agent*), que avalia a confiabilidade das informações coletadas de terceiros. Assim, para esse caso, toda a confiança de α_1 em α_2 é igual à reputação de α_2, representada na equação 2.12:

$$T_{\alpha 1,\alpha 2=} T^r_{\alpha 1,\alpha 2} \tag{2.12}$$

2.4 - TRABALHOS RELACIONADOS

Esta seção apresenta os trabalhos estudados e revisados, relacionados a segurança e confiança em ambientes de computação em nuvem.

2.4.1 - Segurança na Nuvem

Uma série de tecnologias computacionais têm sido empregadas, a fim de fornecer segurança para ambientes de computação em nuvem. A criação e proteção de certificados de segurança não é normalmente suficiente para garantir os níveis de segurança necessários na nuvem (Popovic, 2010). Algoritmos criptográficos utilizados com aplicações em nuvem geralmente reduzem o desempenho, o que deve ser limitado a níveis aceitáveis (Popovic, 2010) (Takabi, 2010).

A computação em nuvem oferece aos usuários uma maneira conveniente de compartilhar uma grande quantidade de recursos distribuídos pertencentes a diferentes organizações. Se por outro lado, a própria natureza do paradigma da computação em nuvem torna os aspectos de segurança e privacidade muito mais complexo. A confiança é a principal preocupação dos consumidores e prestadores de serviços em um ambiente de computação em nuvem (Zhidong , 2010).

A computação em nuvem é composta de ambientes completamente diferentes trazendo desafios especiais à segurança da computação em nuvem. Por um lado, os mecanismos de segurança devem oferecer aos usuários um nível suficiente de garantias. Por outro lado, os mecanismos não devem ser demasiadamente complexos tornando difícil para os usuários o uso do sistema. A abertura e flexibilidade computacional de sistemas operacionais populares disponíveis no mercado foram fatores importantes para apoiar a adoção da computação em nuvem. No entanto, esses mesmos fatores aumentam a complexidade do sistema, reduzem o grau de confiança e introduz buracos, transformando-se em ameaças à segurança (Zhidong , 2010).

Huan *et al.*, (2010) investigam os diferentes métodos de avaliação de vulnerabilidades da segurança para ambientes em nuvem. Experimentos mostram que as vulnerabilidades são mais detectadas se as ferramentas e os servidores estão em uma mesma Rede Local de Computadores (LAN). Em outras palavras, os *hackers* podem encontrar uma maneira mais fácil de obter as informações se elas estiverem na mesma rede

local dos sistemas comprometidos. Os resultados dos experimentos podem ser usados para analisar o risco na terceira parte da nuvem.

Popovic *et al*., (2010) discutem questões de segurança, requisitos e desafios dos Provedores de Serviços em Nuvem (CSP) durante a engenharia da nuvem. Padrões de segurança e modelos de gerenciamento são recomendados para enfrentar estes desafios, tanto para a comunidade técnica como para a empresarial.

2.4.2 - Segurança no Sistema de Arquivos

Como o número de dispositivos gerenciados pelos usuários tem aumentado continuamente, surge a necessidade de sincronizar vários sistemas de arquivos distribuídos hierarquicamente, através de dispositivos com conectividade *ad-hoc*. Uppoor *et al.*, (2010) apresentam uma nova abordagem para a sincronização de sistemas de arquivos distribuídos hierárquicos. A abordagem mantém as vantagens da sincronização *peer-to-peer* com a abordagem baseada em nuvem que armazena uma réplica *master online*. O sistema proposto provê a sincronização de dados em uma rede *peer-to-peer*, eliminando os custos e os requisitos de largura de banda normalmente presentes na abordagem réplica-master da computação em nuvem.

Em (Qingsong, 2010) é apresentado o *Cost-effective Dynamic Replication Management* (CDRM), um esquema para distribuição dinâmica de réplicas de arquivos em um cluster de armazenamento em nuvem. Esta abordagem atualiza periodicamente o número e a localização de réplicas dos blocos de um arquivo no cluster. O número de réplicas é atualizado de acordo com a disponibilidade real dos nós do cluster e da disponibilidade do arquivo esperado. O algoritmo de distribuição dinâmica para a colocação de réplicas leva em consideração a capacidade computacional e de armazenamento, bem como a largura de banda da rede de comunicação dos nós do cluster. Uma implementação do esquema proposto utilizando um sistema aberto de sistema de arquivos distribuído chamado HDFS (*Hadoop Distributed File System*) é discutido. Medições experimentais apontam que o esquema dinâmico supera os algoritmos de distribuição de arquivos estáticos existentes.

2.4.3 - Confiança na Nuvem

Confiança e segurança tornaram-se cruciais para garantir o desenvolvimento saudável das plataformas em nuvem, oferecendo soluções para problemas como a falta de privacidade e proteção, a garantia de segurança e direitos autorais.

Privacidade e segurança são dois importantes obstáculos na adoção da computação em nuvem. A fim de resolver este problema na camada de serviço IaaS, um modelo confiável de computação em nuvem foi proposto para disponibilizar um ambiente de execução fechado que garanta a execução confidencial das máquinas virtuais (Wang *et al*., 2010). Neste trabalho é mostrado como o problema pode ser resolvido usando o *Trusted Platform Module*. O modelo proposto, chamado *Trusted Cloud Computing Platform* (TCCP) foi projetado para prover maior disponibilidade, confiabilidade e segurança. Na solução apresentada, existe um nó cluster que atua como um *Trusted Coordinator* (TC). Outros nós no cluster precisam se registrar com o TC para certificar e autenticar sua chave e lista de medição. O TC mantém uma lista dos nós confiáveis. Quando uma máquina virtual é iniciada ou ocorre uma migração, o TC verifica se o nó é confiável para que o usuário da Máquina Virtual esteja sempre em uma plataforma confiável. Uma chave e assinatura são usadas para a identificação do nó. No modelo TCCP, a autoridade de certificação privativa é envolvida em cada transação junto com o TC (Wang et al, 2010).

Em (Zhidong *et al*., 2010) é apresentado um método para construir um ambiente de computação em nuvem confiável, integrando o *Trusted Computing Platform* (TCP) ao sistema de computação em nuvem. O TCP é usado na autenticação, confidencialidade e integridade do ambiente de computação em nuvem. Esse esquema apresenta resultados positivos com autenticação, acesso baseado em regras e proteção de dados no ambiente de computação em nuvem.

Os Provedores de Serviços em Nuvem (CSP) devem garantir que os serviços que oferecem, não violando a privacidade e confidencialidade dos usuários. Em (Xiao *et al*., 2010) é proposto um Modelo de Ambiente Computacional Confiável Multi-Inquilinos (*Multi-Tenancy Trusted Computing Environment Model* (MTCEM)). Esse modelo foi projetado para a camada IaaS, com o objetivo de assegurar uma infraestrutura confiável de computação em nuvem para os usuários. O MTCEM tem dois níveis hierárquicos: um modelo de confiança transitiva que suporta a separação dos conceitos entre funcionalidade e segurança. O modelo possui 03 entidades: a) os clientes, que alugam os serviços da nuvem de um CSP; b) o CSP, que proveem os serviços de IaaS; c) o auditor, que é

opcional, mas recomendado, que é responsável por verificar se a infraestrutura fornecida pelo CSP são confiáveis em nome dos usuários. No MTCEM, o CSP e os usuários podem colaborar entre si para construir e manter um ambiente de computação em nuvem confiável.

Em (Zhimin *et al*., 2010) é apresentado um modelo de confiança colaborativa para o *firewall* em um ambiente de computação em nuvem. O modelo apresenta três vantagens: a) utiliza diferentes políticas de segurança para diferentes domínios; b) o modelo considera o contexto das transações, os dados históricos das entidades e suas influências na medição do valor de confiança dinamicamente; c) o modelo de confiança é compatível com o *firewall* e não quebra as políticas de controle local do *firewall*. É utilizado o modelo de confiança de domínio. A confiança é medida pelo valor de confiança que depende do contexto e do comportamento histórico da entidade, que não é um valor fixo. A nuvem é dividida em um número de domínios autônomos e a relação de confiança entre os nós é dividida em relações de confiança intra e inter-domínio. As relações de confiança intra-domínio baseiam-se nas transações realizadas dentro do mesmo domínio. Cada nó mantém duas tabelas: tabela de confiança direta e uma lista de recomendações. Se um nó necessita calcular o valor de confiança de outro nó, ele primeiro verifica a tabela de confiança direta e usa o valor, caso o valor referente ao nó exista, se não existir um valor agregado ao nó, a lista de recomendação é verificada para encontrar um nó que tenha uma relação de confiança direta com os nós da lista de recomendação e usa-se o valor recomendado. Os valores de confiança Inter-domínio são calculados com base nas transações entre os nós do inter-domínio. O valor de confiança inter-domínio é um valor global dos valores de confiança direta dos nós e valor de confiança recomendado em outros domínios. São mantidas duas tabelas de confiança implantadas em cada domínio: de relações de confiança inter-domínio e o peso do valor da tabela deste nó no domínio.

Em (Santos *et al*., 2009) uma plataforma de computação em nuvem confiável (TCCP), que permite aos provedores de IaaS oferecer um ambiente de execução caixa fechada que garante a execução confidencial dos clientes das máquinas virtuais (VMs) é apresentado. Este sistema permite ao cliente verificar se seu cálculo será executado de forma segura, antes de requisitar o serviço para iniciar uma VM. TCCP assume que existe um coordenador confiável hospedado em uma entidade externa de confiança. O TCCP garante a confidencialidade e a integridade do usuário da VM, e permite que um usuário determine antecipadamente ou não se o IaaS aplica essas propriedades.

Em (Hwang *et al*., 2009) é apresentado um modelo de confiança usando reputação para sistemas distribuídos em nuvens e redes *peer-to-peer*. É proposta uma arquitetura em nuvem confiável (incluindo a delegação de confiança e sistemas de reputação para recursos e *datacenters* em nuvem) com recursos garantidos, incluindo datasets para serviços sob demanda.

2.5 - DISTRIBUIÇÃO DE ARQUIVOS NA NUVEM

A computação em nuvem oferece grande flexibilidade para os usuários, uma vez que estes não precisam se preocupar com a complexidade de gerenciamento inerente a cada sistema, por exemplo, os bancos de dados podem ser transferidos para *datacenters* de grandes empresas especializadas, embora o gerenciamento dos dados em ambientes terceirizados nem sempre sejam confiável. Os usuários acabam ficando dependentes da disponibilidade e integridade fornecida pelos provedores de serviço de armazenamento. Assim, é necessária a utilização de modelos de armazenamento de dados seguros visando garantir a integridade dos dados dos usuários da nuvem (Wang *et al.*, 2009).

Um dos problemas que a computação em nuvem ajuda a resolver é o armazenamento de arquivos e a sua distribuição com alta taxa de disponibilidade. Existem diversas abordagens para gerenciar dados em nuvem e cada sistema utiliza uma abordagem específica para persistir os dados. Dentre estas abordagens, podemos destacar novos sistemas de arquivos, *frameworks* e propostas para o armazenamento e processamento de dados.

Nas seções seguintes é apresentado uma visão geral dos sistemas de arquivos, *Google File System*, *Amazon S3*, *Microsoft Azure* e *Hadoop*.

2.5.1 - Google File System

O *Google File System* (GFS) é um sistema de arquivos distribuído proprietário desenvolvido pelo *Google* e projetado especialmente para fornecer acesso eficiente e confiável aos dados usando grandes *clusters* de servidores (Ghemawat *et al*., 2003). O desenvolvimento de um novo sistema de arquivos distribuído surgiu em 2003 a partir da ideia de armazenar dados de forma escalável, confiável, com alto desempenho e disponibilidade mesmo em máquinas não confiáveis, devendo atender às necessidades de uso do Google de gerar e manter grandes quantidades de dados. Em comparação com os

sistemas de arquivos tradicionais, o GFS foi projetado e otimizado para ser executado em centros de dados e fornecer elevada vazão, baixa latência e tolerância a falhas individual de servidores (Ghemawat *et al*., 2003).

O tamanho dos arquivos gerenciados pelo GFS varia em torno de 100 megabytes, podendo alcançar vários gigabytes. Para gerenciar o espaço em disco de forma eficiente o GFS organiza os dados em pedaços/blocos (*chunks* – blocos de 64 megabytes) – os quais raramente são sobrescritos ou comprimidos.

O GFS foi projetado para rodar sobre os *clusters* do *Google*, onde cada nó do sistema consiste de computadores comuns semelhantes aos computadores domésticos, o que significa que precauções devem ser tomadas a fim de lidar com a grande incidência de falhas e subsequente perda de dados que podem acarretar grandes danos. Para contornar este problema os nós são divididos em dois tipos: um servidor mestre e centenas ou milhares de servidores menores chamados de *chunkserver*, responsáveis por armazenar os dados. A arquitetura do GFS consiste de três elementos: Clientes, *Master* e *chunkservers*, conforme mostrado na Figura 2.5.

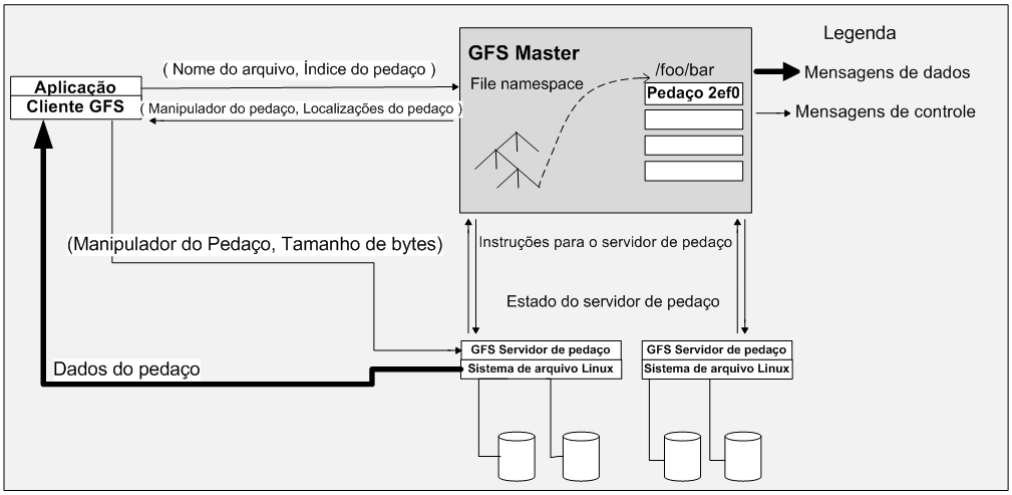

Figura 2.5 - Arquitetura do GFS (Ghemawat *et al*., 2003)

Os *Masters* mantêm os metadados, incluindo o mapeamento da árvore de diretórios para os *chunks*. O servidor mestre é único para todo o sistema de arquivos não armazenando blocos de dados em si, ele contém todos os metadados dos arquivos: espaço de nomes, informações para controle de acesso e o mapeamento dos seus blocos, bem como a localização de cada *chunk* nos *chunkservers*. Todos os metadados são atualizados pelo servidor mestre, o qual se comunica periodicamente com cada *chunkserver* através de

troca de mensagens, chamadas de *Heartbeat messages*, para dar-lhe instruções e coletar o seu estado.

2.5.2 - Amazon S3

O *Amazon Simple Storage Service* (S3) é um sistema de armazenamento distribuído desenvolvido com base no Dynamo (DeCandia, 2007). O Dynamo utiliza o modelo chave-valor armazenado em uma Tabela *Hash* Distribuída (DHT) e não possui suporte a associações ou esquemas. Para garantir um nível de escalabilidade e disponibilidade, os dados são particionados e replicados em várias máquinas, usando um *hashing* consistente e a consistência é facilitada pelo uso de versões de objetos. A consistência entre réplicas durante as atualizações são mantidas por uma técnica chamada de *quorum-like* e um protocolo de sincronização de réplica descentralizado.

No Dynamo, as operações de leitura e escrita são unicamente identificadas por uma chave. O estado é armazenado como objetos binários, identificados por chaves únicas, não existindo suporte às operações sobre múltiplos objetos. As propriedades das transações da base de dados apresentam as seguintes características: atomicidade e isolamento são garantidos pela escrita de um único objeto e a durabilidade é obtida por meio de escrita replicada. A API do Dynamo possui duas operações *get*() e *put*(). A operação *get*(chave) localiza as réplicas do objeto associado com a chave no sistema de armazenamento e retorna um objeto único ou uma lista de objetos com versões conflitantes, ao longo de um contexto. A operação *put* (chave, contexto, objeto) determina onde as réplicas do objeto devem ser colocadas com base na chave associada, e escreve a réplica no disco. As informações de contexto são armazenadas junto com o objeto para que o sistema possa verificar a validade do contexto do objeto fornecido na requisição.

A escalabilidade, balanceamento de carga e suporte à heterogeneidade estão presentes no Dynamo e torna relativamente simples adicionar nós, distribuir as requisições e suportar nós com características diferentes. As propriedades de simetria e descentralização são utilizadas de forma que todos os pontos são igualmente responsáveis e com o objetivo de evitar pontos únicos de falhas. O Dynamo usa a técnica de *hashing* consistente para prover a fragmentação e a replicação. Para tratar o problema de balanceamento de carga, decorrente da utilização de poucos nós ou nós heterogêneos, o Dynamo utiliza a estratégia de nós virtuais, onde cada nó físico pode ter vários nós virtuais associados a ele. Dessa forma, máquinas com maior capacidade de processamento e

armazenamento podem ter uma maior quantidade de nós, sendo estes distribuídos aleatoriamente.

O Amazon S3 utiliza o conceito de objeto, o qual consiste de dados, metadados e um identificador único. Os objetos são organizados em *buckets* e estes servem para agrupar objetos ou espaço de nomes. Para tratar a ocorrência de falhas, os dados são propagados entre os centros de dados de forma postergada e o usuário pode especificar a localização geográfica de um *buckets*. No S3, as operações de escrita são atômicas, mas podem ser executadas sobre múltiplas chaves. Entretanto, este sistema não fornece mecanismos de bloqueio. A Tabela 2.7 apresenta um resumo das técnicas utilizadas no Dynamo e suas respectivas vantagens.

Tabela 2.7 - Problemas, Técnicas e Vantagens do Dynamo (DeCandia, 2007)

Problema	Técnica	Vantagens
Particionamento	*Hashing* consistente.	Aumenta a escalabilidade.
Alta disponibilidade da escrita.	Relógios de vetor (nó, contador), de forma a capturar a causalidade entre as diferentes versões do mesmo objeto. Reconciliação durante a leitura.	Tamanho da versão é desacoplado da taxa de atualização.
Manipulação de falhas temporárias.	*Sloppy Quorum* e *hinted handoff*.	Fornece alta disponibilidade e garantia de durabilidade quando uma das réplicas não está disponível.
Recuperação de falhas permanentes.	Anti-entropia usando árvores de *Merkle*.	Sincroniza as réplicas divergentes em segundo plano.
Membros e detecção de falhas.	Detecção de falha e protocolo baseado em membros – *Gossip*.	Preserva a simetria e evita ter um registro centralizado para armazenar os membros e as informações do nó.

2.5.3 - Microsoft Azure

Microsoft SQL Azure é composto por um conjunto de serviços para o armazenamento e processamento de dados em nuvem (Azure, 2011). O SQL Azure juntamente com o Windows Azure Storage compõem a solução de gerenciamento de dados em nuvem da Microsoft. O objetivo do Windows Azure Storage é oferecer um armazenamento escalável, durável, altamente disponível e proporcionar ao usuário o pagamento sob demanda.

Permite fácil acesso aos dados, oferecendo uma interface simples, disponíveis remotamente e em *datacenters*. Os serviços de armazenamento do Windows Azure Storage são oferecidos em quatro níveis de abstração:

1) BLOBs – Fornece uma interface simples para armazenamento de grandes itens de dados. Um BLOB é um par (nome, objeto) que permite armazenar objetos com tamanho de até 50 GB;
2) Tabelas – Fornecem um conjunto de entidades, que contêm um conjunto de propriedades. Um aplicativo pode manipular as entidades e consultar qualquer uma das propriedades armazenadas em uma tabela. São diferentes das tabelas relacionais e são compostas de entidades. Elas não são acessadas usando a linguagem SQL, mas por meio de serviços de dados;
3) Filas – Fornece armazenamento confiável para a entrega de mensagens, propiciando expedição assíncrona de trabalhos para habilitar a comunicação entre os serviços de diferentes partes (papéis) de sua aplicação. Sua função principal é fornecer um serviço de troca de mensagens persistentes e confiável;
4) *Drives* – Fornece volumes NTFS (Sistema de Arquivos de Nova Tecnologia) duráveis para aplicações.

O Windows Azure Storage inclui armazenamento persistente por meio dos BLOBs, tabelas e filas. O acesso ao armazenamento e o balanceamento de carga é realizado automaticamente através de um conjunto de nós responsáveis pelo armazenamento físico, proporcionando escalabilidade e disponibilidade.

No armazenamento de dados do Azure, a Microsoft promete efetuar a confidencialidade dos dados dos usuários. O procedimento mostrado na Figura 2.6 oferece segurança no acesso aos dados para garantir que os dados não sejam perdidos.

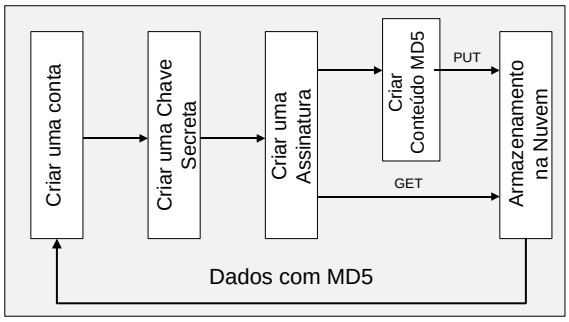

Figura 2.6 - Procedimento de Segurança no Acesso aos Dados (Rajkumar *et al*., 2011)

Para usar o serviço de armazenamento do Windows Azure, o usuário precisa criar uma conta de armazenamento, que pode ser obtida a partir da interface web do portal do Windows Azure. Depois de criar uma conta, o usuário receberá uma chave de 256 bits. Toda vez que o usuário desejar enviar ou buscar os dados a partir da nuvem, ele tem que usar sua chave secreta para criar uma assinatura para cada pedido requisitado individualmente. O usuário usa sua assinatura para autenticar a requisição no servidor. A assinatura é passada com cada requisição para autenticar o pedido do usuário através da verificação da assinatura.

O *Windows Azure Storage* utiliza o algoritmo *Message-Digest* 5 (MD5) (Rajkumar *et al.*, 2011) desenvolvido pela empresa RSA *Data Security* para verificar a integridade dos arquivos, com uma entrada de dados variável e saída fixa (unidirecional), sem possibilidades de através do *hash* obter a informação original. O algoritmo permite verificar se cada bit do arquivo que foi obtido na origem está integralmente idêntico ao destino. Este procedimento aplicado nos serviços de armazenamento em nuvem não pode garantir a integridade dos dados (Rajkumar *et al.*, 2011).

2.5.4 - Hadoop/HDFS

O Hadoop é um *framework* de código livre desenvolvido em Java para rodar aplicações que manipulem dados em larga escala em ambientes distribuídos. O Hadoop é composto pelo sistema de arquivos Hadoop *Distributed File System* (HDFS) e um ambiente de execução paralela (Borthakur, 2007).

O HDFS - Hadoop *Distributed File System* é o sistema de arquivos distribuído do Hadoop, e tem como objetivo o armazenamento de grandes quantidades de dados através de múltiplos nós. O HDFS realiza o tratamento a possíveis falhas de hardware ao replicar os dados em diferentes nós do *cluster*. Os dados são divididos em blocos de 64 MB, e são replicados em três nós – dois nós no mesmo *rack* e outro em um *rack* diferente. Esta estratégia é chamada *de rack-awareness*, onde o *framework* tenta maximizar o tráfego de dados entre nós do mesmo *rack*.

A arquitetura do HDFS consiste em um nó mestre e múltiplos nós escravos. O nó mestre consiste de um único *NameNode* e diversos *DataNodes* (normalmente, um por nó do *cluster*). O *NameNode* é o servidor principal responsável pelo gerenciamento do espaço de nomes do sistema de arquivos e por regular o acesso dos clientes aos arquivos (executa operações no sistema de arquivos, como abrir, fechar e renomear arquivos e diretórios).

Além disso, gerencia a localização dos diversos blocos de dados. Os *DataNodes* tratam do armazenamento dos blocos, sendo responsáveis pela leitura e escrita das requisições feitas pelos clientes ao sistema de arquivos. Eles também executam a criação, exclusão e replicação de blocos sob instruções do *NameNode*. Esta arquitetura possibilita a descentralização das operações de escrita e leitura visto que o *NameNode* apenas informa a qual *DataNode* o cliente deve solicitar a operação. A disponibilidade do dado é garantida pela replicação do mesmo nos *DataNodes*. Como só existe um único *NameNode* em um *cluster* Hadoop, este se torna um ponto único de falha. Caso o *NameNode* falhe, o sistema de arquivos do Hadoop ficará *off-line*. A Figura 2.7 representa a arquitetura do HDFS.

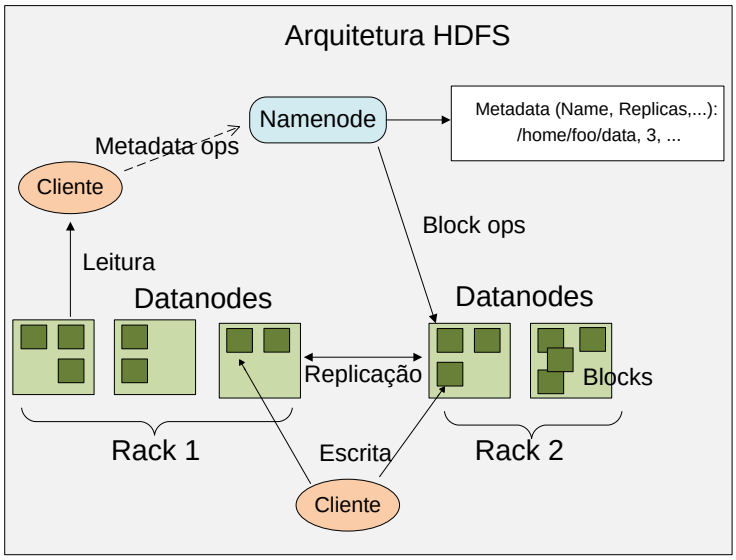

Figura 2.7 - Arquitetura do Sistema de Arquivo Distribuído HDFS (Borthakur, 2007)

O HDFS oferece um espaço de nomes no sistema de arquivos e permite que os dados do cliente sejam armazenados em arquivos. Internamente, um arquivo é dividido em um ou mais blocos e esses blocos são armazenados em um conjunto de *DataNodes*. No HDFS não é permitido que um arquivo seja gravado por mais de um cliente ao mesmo tempo. Existe um mecanismo eficiente de consistência, que auxilia no controle dos dados no servidor de metadados ou *NameNode*. É priorizada a execução de aplicações próximas à máquina de origem ou naquela em que os dados residem. Quando os nós de execução se encontram distantes dos *DataNodes*, os dados devem ser transferidos pela rede.

Os desenvolvedores têm acesso ao HDFS através de interfaces Java que permitem não apenas utilizar o sistema de arquivos do Hadoop, mas também alternar para sistemas de arquivos distintos, como por exemplo, Amazon S3 e Google File System.

O Hadoop armazena arquivos grandes em vários servidores e obtém a confiabilidade por meio da replicação de dados. Similar ao GFS, os dados são armazenados em nós geograficamente distribuídos. O Hadoop possui como principais características: sistemas de armazenamento distribuído, arquivos particionados em grandes blocos e distribuídos nos nós do sistema, blocos replicados para lidar com falha de hardware e um local para dados temporários. Diferentemente de outras abordagens de sistemas de arquivos distribuídos, o armazenamento e processamento do HDFS são feitos em cada nó do sistema.

O Hadoop foi implementado utilizando o *MapReduce*, onde cada operação é composta por duas funções. A primeira chamada de função de Mapeamento, que recebe uma porção do arquivo de entrada e de acordo com a especificação do usuário emite um conjunto de tuplas intermediárias no formato chave-valor. A segunda função, chamada Redução, recebe um conjunto de valores associados a cada chave, chamados de blocos. O processamento, definido pelo usuário, é realizado sobre cada bloco. Cada função de redução emite um conjunto de tuplas que são armazenadas em arquivos de saída. O sistema MapReduce gerencia o processamento através de um processo *master*, cuja função é orquestrar o processamento, gerenciar o processo de agrupamento de registros e distribuir os blocos de forma equilibrada.

2.6 - DISCUSSÕES SOBRE SEGURANÇA DA INFORMAÇÃO

A informação é um ativo que, como qualquer outro ativo importante, é essencial para os negócios de uma organização e consequentemente necessita ser adequadamente protegida. A proteção é extremamente importante no ambiente dos negócios, cada vez mais interconectado e exposto a um crescente número e a uma variedade de ameaças e vulnerabilidades. A segurança da informação é a proteção da informação de vários tipos de ameaças para garantir a continuidade do negócio, minimizar o risco ao negócio, maximizar o retorno sobre os investimentos e as oportunidades de negócio (ABNT, 2005).

Nakamura (2007) define que a segurança se refere à proteção existente sobre as informações de uma determinada organização ou indivíduo, isto é, aplica-se tanto as informações corporativas quanto às pessoais. Assim, a segurança significa permitir que as organizações busquem seus lucros, os quais são conseguidos por meio de novas oportunidades de negócios, que são resultados da flexibilidade, facilidade e disponibilidade dos recursos.

A segurança da informação é obtida a partir da implementação de um conjunto de controles adequados, incluindo políticas, processos, procedimentos, estruturas organizacionais e funções de hardware e software. Estes controles precisam ser estabelecidos, implementados, monitorados, analisados criticamente e melhorados, para garantir que os objetivos do negócio e de segurança da organização sejam atendidos.

Um serviço de segurança é definido como um serviço fornecido por uma camada de protocolo de comunicação de sistemas abertos, que garante a segurança adequada dos sistemas ou das transferências de dados. Os serviços de segurança implementam políticas ou diretrizes de segurança, as quais podem ser físicas ou lógicas (Nakamura, 2007):

1) **Físicas**: são barreiras que limitam o contato ou acesso direto a informação ou a infraestrutura (que garante a existência da informação) que a suporta, tais como: portas, trancas, paredes, blindagem, guardas, etc.;
2) **Lógicas**: são barreiras que impedem ou limitam o acesso à informação, que está em ambiente controlado, geralmente eletrônico, e que, de algum modo, ficaria exposta a alteração não autorizada por indivíduo mal intencionado. Os controles lógicos podem ser implementados por mecanismos de criptografia, assinatura digital, mecanismos de garantia da integridade da informação, mecanismos de controle de acesso, mecanismos de certificação, confiança, etc.

As medidas de segurança são necessárias para proteger os dados durante sua transmissão e evitar violações de segurança, como por exemplo, um usuário A transmite um arquivo ao usuário B. O arquivo contém informações confidenciais (por exemplo, registros de folha de pagamento) que devem ser protegidas contra divulgação. O usuário C, que não está autorizado a ler o arquivo, é capaz de monitorar a transmissão e obter uma cópia do arquivo durante a sua transmissão (Stallings, 2007).

São características básicas da segurança da informação os atributos de confidencialidade, integridade, disponibilidade e privacidade, não estando esta segurança restrita somente a sistemas computacionais, informações eletrônicas ou sistemas de armazenamento. O conceito se aplica a todos os aspectos de proteção de informações e dados, incluindo não apenas a segurança dos dados/informação, mas também a dos sistemas em si.

A confidencialidade é a proteção dos dados e informações transmitidas entre um emissor e um ou mais destinatários contra divulgação não autorizada. Em outras palavras, confidencialidade é a garantia do resguardo das informações dadas pessoalmente em confiança e proteção contra a sua revelação não autorizada, devendo ser feita

independentemente da segurança do sistema de comunicação utilizado. Em um sistema que garante a confidencialidade, um terceiro que toma posse de informações trocadas entre o emissor e o destinatário não é capaz de extrair qualquer conteúdo inteligível. A confidencialidade é garantida utilizando-se mecanismos de criptografia (Stallings, 2007).

A integridade é a garantia de que os dados recebidos estão exatamente como foram enviados pelo emissor, ou seja, não constam modificação, inserção, exclusão ou repetição, seja ela acidental ou proposital. Assim, a integridade significa ter a disponibilidade de informações confiáveis, corretas e dispostas em formato compatível com o de utilização, ou seja, informações íntegras (Stallings, 2007).

A disponibilidade é definida como sendo a propriedade de um sistema ou de um recurso do sistema ser acessível e utilizável sob demanda por uma entidade autorizada do sistema, de acordo com especificações de desempenho (ou seja, um sistema estará disponível se oferecer os serviços de acordo com o projeto do sistema, sempre que os usuários solicitarem). Logo, o princípio da disponibilidade é garantido quando se tem a prestação contínua do serviço, sem interrupções no fornecimento de informações para pessoas autorizadas, sempre que necessário.

De acordo com (Shirey, 2000), a privacidade pode ser definida como o direito de uma determinada entidade (normalmente um indivíduo), agindo em seu próprio nome, de determinar o grau de interação de suas informações com o contexto onde se encontra inserida, incluindo o grau de comprometimento/disposição em divulgar essas informações para outras entidades. A privacidade é uma habilidade do indivíduo em controlar a exposição e a disponibilidade de informações acerca de si.

Três elementos estão presentes na privacidade: o sigilo, o anonimato e o isolamento (o direito de ficar sozinho) (Wright, 2004). O sigilo está fortemente ligado à confidencialidade, o anonimato está relacionado à proteção da identidade do indivíduo (isto é, sem identificação ou autenticação) e o isolamento ao direito de ficar indisponível para outros indivíduos. A privacidade também pode ser vista como a capacidade de um usuário realizar ações em um sistema sem ser identificado. A privacidade consiste nos direitos e obrigações dos indivíduos e organizações com relação à coleta, uso, conservação e divulgação de informações pessoais (Mather *et al*., 2009).

A privacidade pode ser considerada como um aspecto da confidencialidade. A confidencialidade define que uma informação não deve estar disponível ou divulgada a indivíduos, entidades ou processos não autorizados pela política de acesso (Shirey, 2000). Por sua vez, a privacidade é a proteção contra a exposição indevida de informações

pessoais ou o desejo de controlar o nível de exposição e uso dessas informações. Um mecanismo de segurança que provê privacidade garante que uma mensagem enviada somente será lida pelo destinatário escolhido, ou seja, uma mensagem enviada ao destinatário X, somente será lida por ele. Caso seja lida por um destinatário não autorizado, estaremos diante de uma falha de privacidade.

Em uma solução de computação em nuvem, os conceitos de confiança, segurança, privacidade, confidencialidade, integridade e disponibilidade são importantes e devem estar presentes (Wang *et al*., 2010) (Uppoor *et al*., 2010) (Popovic *et al*., 2010). A Figura 2.8 apresenta a relação destes conceitos, com o conceito de confiança adotado para o modelo proposto neste trabalho.

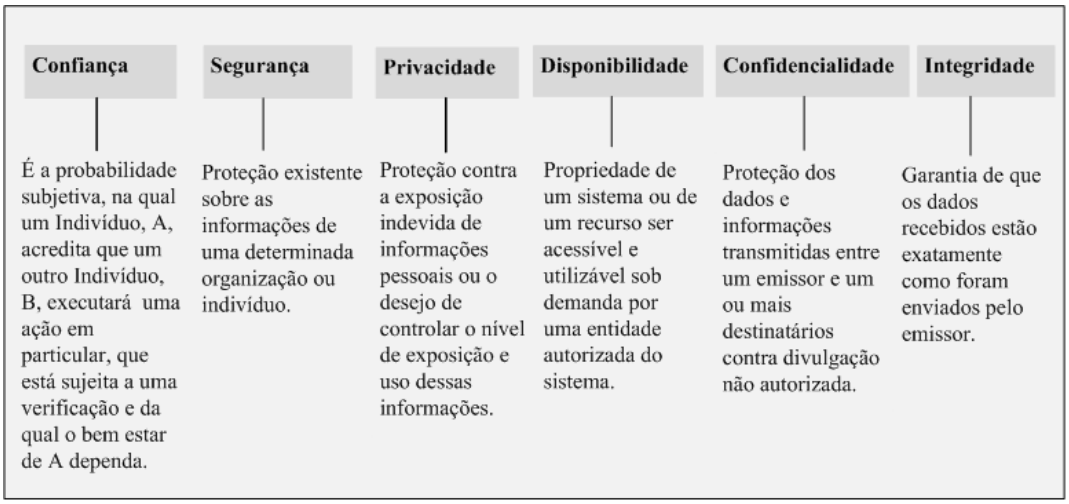

Figura 2.8 - Confiança versus Segurança na Computação em Nuvem

Neste trabalho, a confiança será considerada como a probabilidade na qual um nó A, acredita que outro nó, B, executará uma atividade em particular, sujeita a uma verificação, dentro de um determinado contexto, e que a ação realizada terá influencia nas ações decorrentes.

No modelo de confiança proposto a escolha do nó confiável dar-se-á pela sua disponibilidade em transmitir o arquivo solicitado. Será considerado que os trabalhos propostos (seção 2.4.3) de segurança e privacidade para ambientes de computação em nuvem funcionam corretamente.

Em ambientes distribuídos, a aquisição da confiança entre os nós envolve a análise das ações realizadas anteriormente ou o histórico do seu comportamento, podendo influenciar na tomada de decisão de relacionamento e no cálculo do valor de confiança

para a execução de uma determinada ação de maneira colaborativa. Caso não haja nenhum tipo de iteração anterior e ou histórico com um determinado nó, o valor será calculado a partir das informações fornecidas pelos outros nós.

2.7 - PROBLEMAS EM ABERTO

Um dos principais desafios que impedem os usuários finais de adotarem os serviços de armazenamento em nuvem é o medo de perder seus dados ou a violação deles. Assim, a integridade dos dados e o armazenamento não confiável é um grande desafio para os provedores de serviços de armazenamento em nuvem (Rajkumar *et al*., 2011). É fundamental para minimizar o medo dos usuários, prover tecnologias que permita aos usuários verificar a integridade dos seus dados.

Os provedores de serviços de armazenamento de arquivos em nuvem apresentados não tratam dos problemas relacionados à troca confiável de arquivos entre os pares. A preocupação dos mesmos está relacionada à distribuição e disponibilidade dos arquivos, solucionando este problema através da réplica de arquivos. Além disso, adotam algoritmos robustos para prover a confidencialidade dos dados armazenados, dependendo dos protocolos de segurança da rede de comunicação, para proteger os dados na transmissão na rede.

Técnicas de criptografia foram propostas e utilizadas para garantir a privacidade dos dados na nuvem. No entanto, estas técnicas têm implicações significativas de desempenho de consultas em SGBDs (Rajkumar *et al*., 2011). Dessa forma, alternativas para a integração de técnicas de criptografia com SGBDs devem ser avaliadas e propostas, já que a complexidade computacional da criptografia de dados aumenta o tempo de resposta da consulta. Além disso, também se faz necessário a proposta de um modelo para troca confiável de arquivos no ambiente de computação em nuvem, pois este problema ainda não foi tratado.

2.8 – SÍNTESE DO CAPÍTULO

O objetivo deste capítulo foi fazer uma revisão dos principais conceitos de computação em nuvem, confiança, modelos de confiança e reputação. Além disso, foi realizado um mapeamento das questões de segurança, privacidade e confiança em ambientes de computação em nuvem, bem como apresentado os trabalhos correlatos, permitindo citar

autores e modelos referenciados em trabalhos que tratam do problema da segurança e confiança em computação em nuvem. Os principais sistemas de distribuição de arquivos em nuvem foram discutidos. Os conceitos de segurança, confidencialidade, privacidade e disponibilidade foram discutidos e apresentados através de representações específicas. Por fim, foram apresentados alguns problemas em aberto.

3 – PROPOSTA DE MODELO DE CONFIANÇA PARA TROCA DE ARQUIVOS

Este capítulo apresenta o modelo de confiança proposto para garantir a troca confiável de arquivos entre os usuários de uma nuvem privada, bem como o cálculo da confiança entre os respectivos usuários, de acordo com as métricas estabelecidas.

3.1 - MODELO PROPOSTO

De acordo com a revisão e as pesquisas relacionadas (Wang *et al*., 2010) (Uppoor *et al.*, 2010) (Zhidong *et al.*, 2010) (Xiao *et al.*, 2010) (Hwang *et al.*, 2009) (Zhimin *et al.*, 2010) (Santos *et al.*, 2009) (Qiang *et al.*, 2011) (Udhayakumar *et al.*, 2011), a computação em nuvem necessita de um modelo de confiança para garantir a troca confiável de arquivos entre os usuários da nuvem.

Neste trabalho, propõe-se um modelo de confiança para estabelecer um nível de confiança entre os nós e permitir a troca confiável de arquivos entre seus pares em uma nuvem privada. O ambiente de computação em nuvem privada permite que seja trabalhado com um contexto específico de distribuição de arquivos, de modo que os arquivos tenham uma distribuição e disponibilidade desejada, sendo possível ter garantias do gerenciador da nuvem que seu acesso é restringido, bem como a identificação dos nós é única e controlada.

Uma das técnicas utilizadas para trocar arquivos na nuvem é dividir o arquivo em vários blocos de arquivos (Qingsong *et al.*, 2009). A recuperação total do arquivo é realizada juntando os blocos, com o ID obtém-se o número do pedaço do bloco transmitido, aplicando uma regra de criptografia mais *hash*, tem-se o arquivo criptografado e íntegro, podendo-se através dele verificar o *hash* para saber se o mesmo foi modificado. Pode-se usar uma extensão desta regra para resolver o problema da disponibilidade, uma vez que esta regra resolve o problema da confidencialidade e integridade, mas não resolve a disponibilidade. Aplicando uma regra de Confiança para distribuir o arquivo, é possível aumentar a sua disponibilidade. A escolha do nó mais confiável será efetuada em termos da disponibilidade. Na aplicação do modelo de confiança proposto será adotada a premissa que o arquivo transmitido é apenas um bloco (Qingsong *et al.*, 2009).

Inicialmente, é proposto um modelo de confiança em que a seleção dos nós e a avaliação do valor de confiança do nó que determinará se o nó é confiável ou não, será realizada com base no espaço de armazenamento, sistema operacional, capacidade de processamento e enlace do nó. Por exemplo, um determinado nó tem um espaço de armazenamento na nuvem privada, mas não possui nenhum critério de seleção para determinar a qual nó irá enviar um arquivo em particular. Quando um nó deseja compartilhar arquivos com outros nós da nuvem, ele fará a seleção dos nós confiáveis para enviar os arquivos, usando, por exemplo, os seguintes resultados: uma máquina com processamento de 100% (A carga de trabalho que o nó é submetido constantemente é muito importante, por exemplo, se a máquina tiver uma utilização de 100% da capacidade de processamento a maior parte do tempo terá um grau maior de dificuldade em atender qualquer demanda que for solicitada a ela), sistema operacional (sistema operacional que possui um histórico menor de vulnerabilidade será menos suscetível a falhas), capacidade de armazenamento e enlace (quanto melhor for o enlace, maior será a confiança em enviar alguma informação e menor o tempo para a recuperação da informação). A confiança será estabelecida com base nas consultas/requisições enviadas para os nós e as respostas recebidas dos nós na nuvem, considerando as métricas adotadas, como mostrado na Figura 3.1.

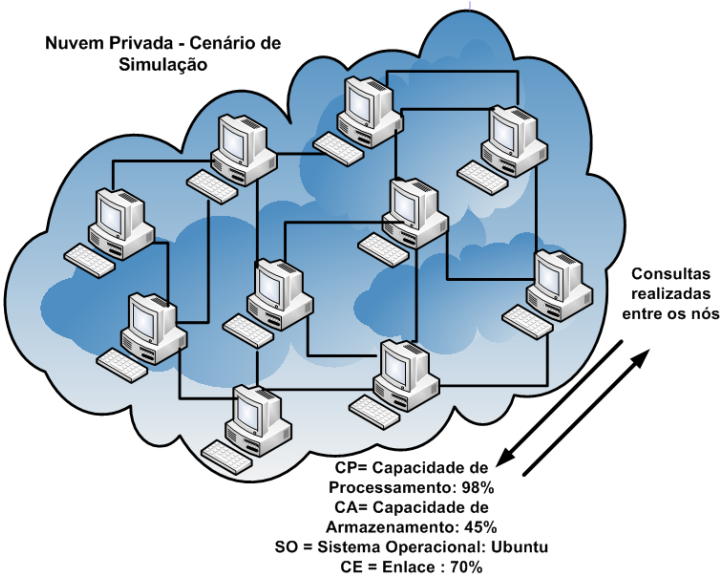

Figura 3.1 - Métricas Adotadas para o Cálculo dos Valores de Confiança

Cada nó mantém duas tabelas, uma de confiança e uma de recomendações, sendo calculadas da seguinte maneira:

a) Uma tabela de confiança direta e uma de recomendações. Se um nó necessita calcular o valor de confiança de outro nó, ele inicialmente verifica a sua tabela de confiança e usa o valor de confiança caso o valor referente ao nó exista, se não existir um valor agregado ao nó, em seguida, a lista de recomendação é verificada para descobrir um nó que tenha uma relação de confiança direta com os nós da nuvem nas listas recomendadas e usa-se o valor recomendado. Caso não exista estas informações é enviado uma consulta (*query*) aos nós pares solicitando informações sobre sua capacidade de armazenamento, sistema operacional, capacidade de processamento e enlace. O valor fornecido pelo nó será comparado com o valor médio fornecido por todos os nós da nuvem. Os valores de confiança são calculados com base nas consultas realizadas entre os nós.

b) O nó irá atribuir um valor de confiança ao nó que possuir maior capacidade de armazenamento e/ou processamento e melhor enlace. Além disso, o sistema operacional também será considerado como critério de confiança.

No modelo proposto assume-se que o nó possui uma identidade única na rede. Como a confiança é evolutiva, quando um nó entrar na rede, o nó requisitante ainda não o conhece, logo, será solicitado informações sobre a sua reputação aos outros nós da rede. Caso nenhum nó possua informações sobre o respectivo nó (não tenha tido nenhuma experiência com o mesmo), o nó requisitante decidirá se irá relacionar com o nó requisitado, solicitando inicialmente alguma atividade/demanda para que ele execute. A partir das suas repostas será construída a confiança com o respectivo nó. A tabela de confiança do nó conterá um temporizador (guardando o comportamento/acontecimentos que aumentam e diminuem a confiança de um determinado nó) e será atualizada em determinados momentos.

A confiança entre os nós será calculada de acordo com as seguintes premissas: a partir das informações existentes sobre o determinado nó; informações solicitadas na rede sobre o nó analisado ou informações solicitadas diretamente ao nó.

A Figura 3.2 apresenta uma visão de alto nível do fluxograma representando as ações realizadas pelos nós quando efetuam consultas aos seus pares para obter as informações necessárias para construir a sua tabela de confiança local.

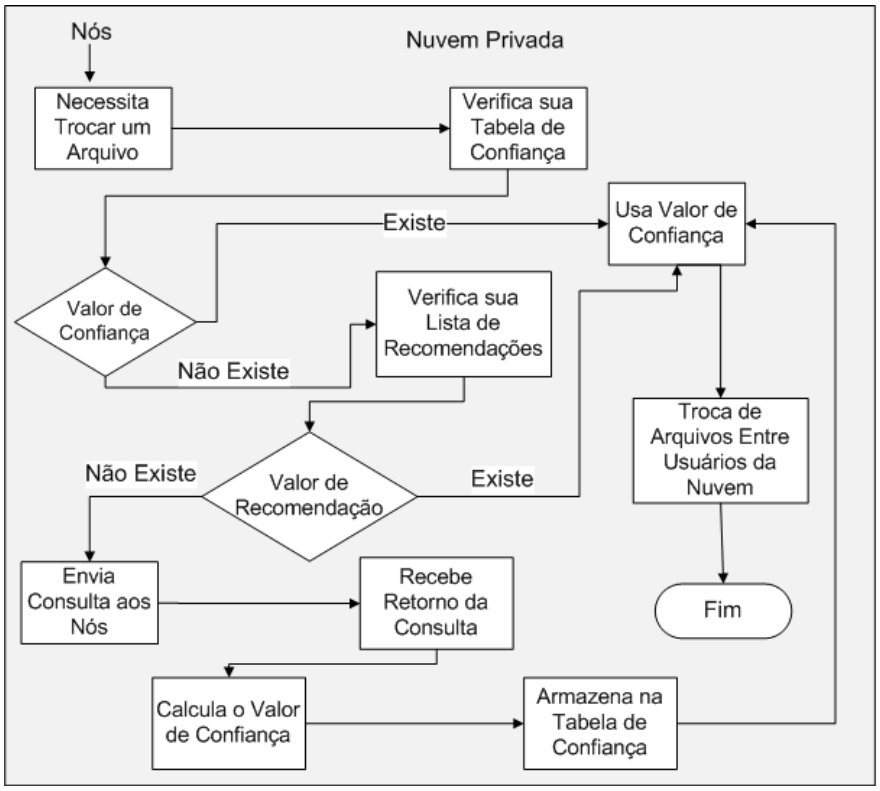

Figura 3.2 – Fluxograma do Modelo de Confiança Proposto

O modelo de confiança proposto para estabelecer um nível de confiança entre os nós, permitindo a troca confiável de arquivos entre os nós da nuvem privada, busca responder como o nó A pode confiar em um nó B, para enviar um arquivo através da nuvem. Para obter estas respostas o nó A necessita obter algumas informações a respeito do comportamento/funcionamento do nó B.

O cenário de solicitação das informações para uma troca confiável de arquivos entre os nós é apresentado na Figura 3.3.

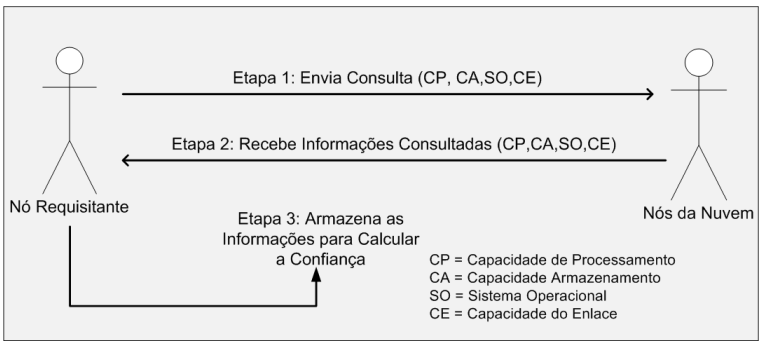

Figura 3.3 - Cenário de Requisições de Informações

Quando o nó A necessita trocar um arquivo na nuvem e deseja saber se o nó B é confiável para enviar o arquivo ao mesmo, ele irá utilizar o Protocolo do Modelo de Confiança proposto, que pode ser descrito com o seguinte cenário:

1) Na etapa 1, o nó A envia uma requisição aos nós da nuvem, incluindo o nó B, perguntando sobre sua Capacidade de Armazenamento, Sistema Operacional, Capacidade de Processamento e Enlace;
2) Na etapa 2, os nós, incluindo o nó B, enviam uma resposta a requisição de A, fornecendo as informações solicitadas;
3) Na etapa 3, o nó A avalia as informações recebidas de B e dos demais nós. Caso as informações fornecidas por B, estejam de acordo com o esperado, ou seja, com o valor médio das informações dos demais nós, os valores serão armazenados na tabela local de recomendações do nó A, para posteriormente efetuar o cálculo de confiança e armazenar na sua tabela de confiança local.

O valor de confiança de um nó indica a sua disposição/adequação para realizar as operações entre os pares da nuvem. Este valor é calculado com base no histórico das interações/consultas feitas entre os nós, podendo ter o seu valor variando entre [0,1].

De uma maneira geral, a confiança de um nó A em um nó B em uma nuvem privada NP pode ser representada por um valor V que mensura a expectativa de que um determinado nó irá se comportar bem na nuvem privada, logo a confiança pode ser expressa pela equação:

$$T_{(a,b)}^{np} = V_{np}^{(a,b)} \tag{3.1}$$

$T_{(a,b)}^{np}$ Representa a confiança de A em B na nuvem privada NP e $V_{np}^{(a,b)}$ representa o valor de confiança de B na nuvem privada NP analisado por A.

Segundo a definição de confiança, $V_{np}^{(a,b)}$ equivale às consultas enviadas e recebidas (interação) por parte de A em relação a B na nuvem NP. À medida que são realizadas as interações entre os nós da nuvem privada, as informações são usadas para o cálculo da confiança.

Os nós de uma nuvem privada devem ser capazes de analisar se um valor de confiança é ou não aceitável, gerando o limiar de confiança. Se o nó ultrapassar o limiar dentro de um conjunto de valores analisados, ele deve ser capaz de julgar o referido nó sobre um determinado grau de confiança. O grau de confiança pode variar de acordo com uma avaliação quantitativa: um nó tem uma confiança muito alta em outro, um nó tem baixa confiança em outro, um nó não tem critérios suficientes para opinar, um nó confia o

suficiente para opinar, etc. A Tabela 3.1 representa os valores que foram estabelecidos para determinar a avaliação quantitativa da confiança e reputação em um determinado nó.

Tabela 3.1 - Valores de Referência para a Confiança (Marsh, 1994)

Valor	Descrição	Decisão
0	Nenhuma Confiança nos nós da nuvem privada	Sem opinião
[0, 0.39]	Baixa Confiança nos nós da nuvem privada	Não confia
[0.4, 0.59[	Média Confiança nos nós da nuvem privada	Não confia
[0.6, 0.89[	Alta Confiança nos nós da nuvem privada	Confia
[0.9, 0.99[	Confiança Muito Alta nos nós da nuvem privada	Confia

De acordo com as informações de referência da Tabela 3.1, um nó confia em outro a partir do valor de confiança $T \geq 0{,}6$. Os valores de confiança serão calculados a partir das consultas realizadas entre os nós da NP, possibilitando obter as informações necessárias para o cálculo final da confiança.

As informações de confiança serão armazenadas através de registros individuais de interação com o respectivo nó, mantendo-se em uma base de dados local as informações sobre o comportamento de cada nó da nuvem que deseja se trocar um arquivo (tabela de confiança local e tabela de recomendação local).

A Tabela 3.2 apresenta os nós e as métricas que serão utilizadas para o cálculo da confiança. *n* representa os nós da nuvem privada e o *m* representa as métricas adotadas, mencionadas anteriormente.

Tabela 3.2 - Matriz dos Nós x Métricas

Nó	Métricas		
n_1	m_1	m_2	m_x
n_2	...	...	...
n_x	...	...	...

A partir da matriz apresentada na Tabela 3.2 o cálculo da confiança de um nó A sobre o nó B na nuvem NP será representado pela equação:

$$T_{(a,b)}^{fnp} = \frac{\sum_{np=1}^{j} V_{np}^{b}((b,m_1) + (b,m_2) + (b,m_3) + \cdots + (b,m_x))}{j} \quad (3.2)$$

$T_{(a,b)}^{fnp}$ Representa a confiança final de A em B na nuvem privada NP. O valor de confiança de B é definido através do somatório dos valores das métricas que o nó B possui

(m) na nuvem NP, *j* representa à quantidade de interações de confiança do nó A com o nó B na nuvem NP.

Para calcular a reputação de um determinado nó em relação aos demais nós, ou seja, o cálculo a partir das informações que os nós da nuvem possuem, é necessário calcular o valor de reputação de cada nó em relação à nuvem NP. Inicialmente é calculado as interações satisfatórias e não satisfatórias de um nó, conforme definido em Patel (2007) na equação 3.3:

$$N_{(N1,N2)} = \sum_{k=0}^{p} N_{Nk,N2} \ , \ M_{(N1,N2)} = \sum_{k=0}^{p} M_{Nk,N2} \quad (3.3)$$

N representa o número de interações satisfatórias entre os nós N_1 e N_2 e M representa o número de interações não satisfatórias entre os nós N_1 e N_2, p representa o total de interações realizadas entre os nós. O valor obtido é usado para calcular o parâmetro na forma da distribuição beta (Patel, 2007), dada na equação 3.4 e estendida na equação 3.5:

$$\alpha = M_{(N1,N2)} + 1 \ e \ \beta = N_{(N1,N2)} + 1 \quad (3.4)$$

Logo, o cálculo da reputação será dado pela equação 3.5 (Patel, 2007):

$$R_{(n1,n2)} = \frac{\alpha}{\alpha + \beta} \quad (3.5)$$

Assim, para calcular a reputação de um determinado nó em relação aos demais nós da nuvem é preciso analisar as experiências satisfatórias e não satisfatórias realizadas com este nó pelos demais, conforme demonstrado a seguir:

$\beta = \{ i \cap j \}$, representando todos os valores que coincidem com os valores informados pelos nós consultados. Assim, β representa os valores igualmente presentes nos dois grupos.

$\alpha = \{ i + j - (i \cap j)\}$, representando todos os valores que não coincidem com os valores informados pelos nós consultados. Assim, α representa os valores que estão presentes em apenas um dos grupos.

Com os valores da reputação do nó é possível calcular a reputação média de cada nó. Logo, o valor da reputação de um nó em relação aos demais pode ser representado pela equação 3.6:

$$R = T_{(N1,N2)} = \frac{\alpha}{\alpha + \beta} \quad (3.6)$$

3.1.1 - Cálculo da Confiança

Para a proposta deste trabalho, quatro aspectos podem ter impacto sobre o cálculo da confiança de um nó, como mostrado na Tabela 3.3.

Tabela 3.3 - Aspectos que Influenciam no Cálculo da Confiança do nó (Xiao *et al*., 2010) (Neisse *et al*., 2011) (Garg *et al*., 2012) (Manuel *et al*., 2009)

Capacidade de Armazenamento (CA)	Capacidade de Processamento (CP)	Capacidade de Enlace (CE)	Sistema Operacional (SO)	Confiança no Nó
Alta	Alta	Alta	Alta	Alta
Alta	Alta	Baixa	Baixa	Alta
Alta	Baixa	Alta	Baixa	Média (Depende dos Valores de Armazenamento e Processamento)
Alta	Baixa	Baixa	Baixa	Baixa
Baixa	Alta	Alta	Baixa	Média (Depende dos Valores de Armazenamento e Processamento)
Baixa	Alta	Baixa	Baixa	Baixa
Baixa	Baixa	Alta	Baixa	Baixa
Baixa	Baixa	Baixa	Baixa	Baixa

Uma maior capacidade de armazenamento e processamento possui peso maior na escolha de nós mais confiáveis, devido a estas características, garantir a integridade e o armazenamento de arquivos. Assim, para calcular a confiança do nó, atribui-se pelo administrador da nuvem privada a capacidade de armazenamento e processamento com pesos de 35%, ao enlace 15% e sistema operacional com os 15% restantes. Os pesos são variáveis e serão atribuídos pelo administrador da nuvem privada de maneira que as métricas capacidade de armazenamento e processamento tenham sempre um valor significativo em relação às métricas enlace e sistema operacional.

Os pesos foram escolhidos arbitrariamente neste trabalho para fixar os pesos das variáveis, de modo a observar somente o valor de confiança, no sentido restrito da definição apresentada na seção 2.2. Entretanto, apesar de arbitrário, procurou se atribuir valores com critérios razoáveis. Por essa razão, observou-se nos trabalhos (Xiao *et al*., 2010) (Neisse *et al*., 2011) (Garg *et al*., 2012) (Manuel *et al*., 2009) que a capacidade de armazenamento e processamento de dados tem grande relação com o desempenho obtido nas trocas de arquivos na nuvem. A capacidade de enlace parece ter atingido um valor

adequado para aplicações em nuvem, assim como os sistemas operacionais tem conseguido se equiparar quanto aos serviços e desempenho oferecidos.

Sabe-se que um nó qualquer, pode ter o valor da confiança variando entre [0,1] e que esses valores são variáveis ao longo do tempo, por exemplo, um determinado nó pode ter sua capacidade de armazenamento aumentada ou diminuída, se faz necessário que a confiança reflita o comportamento do nó em um determinado período de tempo. Os nós com características mais constantes devem, portanto, ser mais confiáveis, pois possuem menor oscilação em suas características básicas.

De acordo com os pesos atribuídos as quatro métricas propostas, é possível calcular a confiança do nó a partir da equação 3.7:

$$T_{(a,b)}^{fnp} = \frac{\sum_{np=1}^{j} V_{np}^{b} \left(((b,m_1) * 0{,}35) + ((b,m_2) * 0{,}35) + ((b,m_3) * 0{,}15) + ((b,m_4) * 0{,}15)\right)}{j} \quad (3.7)$$

3.2 – SÍNTESE DO CAPÍTULO

O objetivo deste capítulo foi apresentar o modelo de confiança proposto e as equações envolvidas para efetuar o cálculo dos valores de confiança para um determinado nó de uma nuvem privada, utilizando métricas pré-estabelecidas e seus respectivos pesos.

4 - SIMULAÇÕES E RESULTADOS

Este capítulo trata dos resultados obtidos com a simulação experimental do modelo de confiança, demonstrando a viabilidade do modelo proposto para a troca confiável de arquivos entre os nós de uma nuvem privada.

Os resultados obtidos foram coletados utilizando a ferramenta automatizada CloudSim (Rajkumar *et al*., 2009) através de saídas de comandos implementados ou analisados. A sua representação, de maneira a simplificar a demonstração dos resultados, será feita através de tabelas e gráficos e a sua correspondente análise, bem como a descrição dos cenários adotados na simulação.

4.1 - FERRAMENTAS UTILIZADAS

O framework CloudSim é uma ferramenta que oferece os recursos necessários para a simulação de ambientes computacionais em nuvem, ou seja, permite a modelagem e a simulação da infraestrutura de uma nuvem para que os prestadores de serviços possam realizar testes em um ambiente no qual não há cobrança de taxas (Calheiros *et al*., 2011). Desenvolvido em linguagem Java e licenciado pelo General Public License (GPL), o *framework* é extensível, facilmente adaptável e permite a criação de simulações generalizadas em grande escala com alto grau de customização (Rajkumar *et al*., 2009).

O CloudSim oferece uma estrutura de simulação que permite a experimentação, modelagem e simulação de infraestruturas e serviços de aplicações de computação em nuvem. Uma das principais vantagens do CloudSim é permitir aos desenvolvedores e pesquisadores testar o desempenho do ambiente de computação em nuvem, o qual pode ser controlado e configurado com facilidade. As principais características oferecidas pelo *framework* são (Calheiros *et al*., 2011):

1) Suporte para modelagem e simulação em grande escala de ambientes de computação em nuvem, incluindo *Datacenters*, na computação em um único nó físico;
2) Possui plataformas independentes de modelagem de nuvens, provedores de serviços, fornecimento e atribuições de políticas de alocação de recursos;
3) Suporte para simulação de conexões de rede entre os elementos do sistema simulado;

4) Eficácia: exige-se menos tempo e esforço para a implementação do ambiente de simulação;
5) Flexibilidade e Aplicabilidade: os desenvolvedores podem testar os seus modelos com pouco esforço de programação;
6) Facilidades para a simulação de ambientes de nuvens federadas e recursos inter-rede de domínios privados e públicos;
7) Mecanismo de virtualização que auxilia na criação e gerenciamento de múltiplos e independentes serviços virtualizados no nó do *Datacenter*;
8) Flexibilidade para alternar entre as políticas de alocação de espaço compartilhado (*space-shared*) e tempo compartilhado (*timeshared*) de processamento de núcleos de serviços virtualizados.

O principal benefício deste *framework* é a possibilidade de testar os serviços propostos e desenvolvidos sem que eles estejam atrelados a uma determinada nuvem. Além disso, por ser uma simulação, não há um modelo econômico de tarifação que cobre pelo seu uso. Desta forma, os testes podem ser realizados/simulados quantas vezes forem necessárias, sem que haja preocupação com custos monetários.

A Figura 4.1 apresenta as camadas que compõem a arquitetura do *framework* CloudSim e os seus componentes (Calheiros *et al*., 2011).

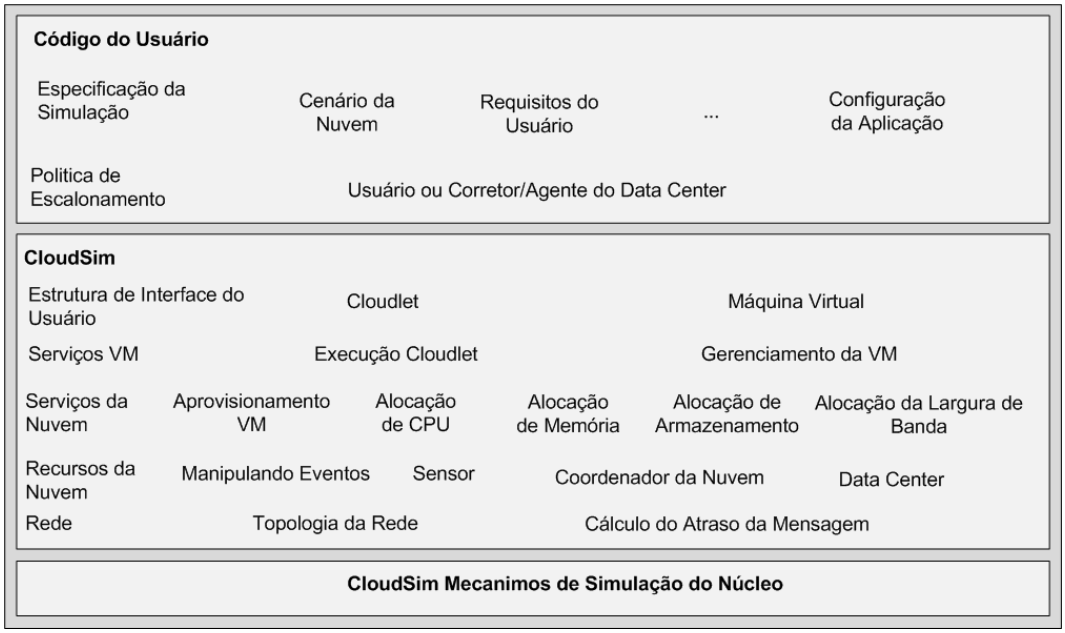

Figura 4.1 - Camadas da Arquitetura CloudSim (Calheiros *et al*., 2011)

Na camada de nível inferior, encontra-se o núcleo de simulação, responsável pelas operações de criação, gerenciamento e exclusão das entidades simuladas (serviços, *host*, *datacenter*, *broker* e máquina virtual). A camada do meio representa as principais classes que compõem o *framework*, sendo composta por diferentes módulos. No módulo de rede, é realizado o mapeamento de enlaces entre *datacenters* e clientes e o cálculo de atraso das mensagens trocadas entre os mesmos.

O módulo de recursos da nuvem realiza a manipulação e coordenação dos eventos da simulação, além de gerenciar os dados relativos à infraestrutura oferecida por meio dos *datacenters* simulados. Em seguida, o módulo de serviços da nuvem representa as ações de provimento de máquinas virtuais e alocação de recursos como memória de sistema, processamento, armazenamento de dados e largura de banda de comunicação. Em seguida, temos o módulo de serviços das máquinas virtuais, onde é realizado o gerenciamento das máquinas virtuais e a execução das tarefas enviadas pelos clientes, denominadas *cloudlets*. Por fim, a comunicação das entidades que compõem a nuvem com os clientes que utilizam seus recursos é feita por meio do módulo, estrutura de interface do usuário, no qual máquinas virtuais e *cloudlets* podem ser manipuladas.

A camada superior representa o código que o usuário do *framework* deve implementar para a criação dos ambientes de simulação. O módulo de política de escalonamento permite a criação das políticas de decisão e escalonadores que nortearão os processos de simulação. Além dos mecanismos de decisão denominados políticas de *broker*, o CloudSim permite a implementação de políticas de alocação de máquinas virtuais entre os *hosts* de um mesmo *datacenter*, escalonadores de máquinas virtuais em *hosts* e escalonadores de *cloudlets* em máquinas virtuais. Estes escalonadores podem lançar mão do compartilhamento de recursos no domínio do espaço ou do tempo.

4.2 - DESCRIÇÃO DO AMBIENTE SIMULADO

Com o intuito de demonstrar os objetivos propostos neste trabalho, faz se necessário definir um ambiente de simulação capaz de mensurar/validar as métricas utilizadas, na expectativa de atingir resultados de acordo com os parâmetros e critérios de informações de confiança adotados neste trabalho. Além disso, o ambiente de simulação serve como base para discussões futuras, bem como a evolução desta proposta através de novos ambientes de computação em nuvem.

Através da implementação do ambiente de simulação é possível discutir e analisar os parâmetros necessários para um modelo de confiança em uma nuvem privada, avaliar a geração da tabela de confiança local dos nós, bem como a efetividade das métricas adotadas, e por fim gerar resultados que sirvam para a discussão do problema da troca confiável de arquivos entre os pares de uma nuvem privada.

O ambiente de simulação do CloudSim reproduz a interação entre um provedor de Infraestrutura como Serviço (IaaS) e seus clientes (Calheiros *et al*., 2011). A Figura 4.2 demonstra o cenário do ambiente de simulação do CloudSim. O provedor de IaaS possui um ou muitos *datacenters*, que podem ser configurados e modelados de acordo com seu sistema operacional, arquitetura dos processadores, monitor de máquina virtual ou *hypervisor* empregado (camada de software posicionada entre o hardware da máquina e o sistema operacional), largura de banda disponível, custos de utilização, política de alocação de máquinas virtuais e opções de consumo de energia.

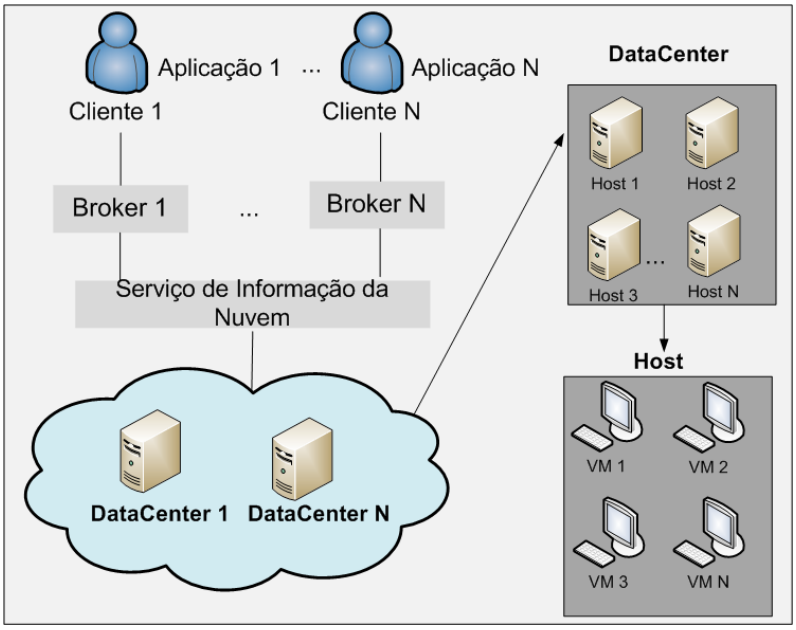

Figura 4.2 - Cenário do Ambiente de Simulação do CloudSim

Os *hosts* que compõem um *datacenter* podem ser configurados atribuindo-se valores de memória de sistema, largura de banda, capacidade de armazenamento, capacidade de processamento e opções de escalonadores de máquinas virtuais e modelos de utilização de energia.

A criação de *datacenters* heterogêneos é efetuada através da adição de diferentes perfis de configuração dos *hosts*. Os clientes são modelados por meio de um perfil de

utilização de recursos e configurações relativas às máquinas virtuais que serão executadas através da infraestrutura oferecida pelo provedor. O perfil de utilização consiste em uma descrição das tarefas, denominadas *cloudlets*, que serão processadas pelas máquinas virtuais e pela indicação de uma política de escolha de *datacenters* representada pela entidade *broker*.

As *cloudlets* são descritas através de características como quantidade de núcleos necessários para o processamento, comprimento em milhões de instruções por segundo (MIPS) e tamanho dos arquivos de entrada e saída que serão transferidos entre os clientes e *datacenters*. Existem também modelos distintos de utilização de CPU (unidade central de processamento/processador), memória e largura de banda. O tipo de *broker* determina de que forma serão escolhidos os *datacenters* que alocarão as máquinas virtuais e, por conseguinte, executarão as *cloudlets*. Os *brokers* selecionam os *datacenters* de acordo com os critérios de custo, capacidade de processamento e latência de rede.

Os clientes normalmente alocam suas máquinas virtuais utilizando os recursos disponíveis nos *datacenters* do provedor e seguindo as políticas de seus respectivos *brokers.* A etapa de alocação de máquina virtual pode ocorrer durante todo o período de simulação através da prática de migração de máquinas virtuais. Os clientes enviam suas *cloudlets* para execução e aguardam o retorno das mesmas. Os dados gerados durante todo o processo de simulação são exibidos para o usuário no *console* do CloudSim.

4.2.1 - Cenário da Simulação

O cenário das simulações realizadas neste trabalho através do framework CloudSim compreendem a um provedor de IaaS, que possui três *datacenters* e um cliente que dispõe desse serviço. O cliente utiliza os recursos oferecidos pelo provedor para o envio e alocação de máquinas virtuais que executam um conjunto de tarefas, denominadas *cloudlets.* A dinâmica da escolha dos *datacenters* para o envio e alocação de máquinas virtuais e a execução de *cloudlets* é definida pelo perfil de utilização do cliente e pelos recursos oferecidos pelo provedor. O cenário simulado neste trabalho é constituído por um provedor de IaaS que dispõe de três *datacenters* distribuídos em localidades distintas, Goiânia-GO, Anápolis – GO e Brasília-DF, um cliente com um perfil de utilização, 04 *hosts*, 30 máquinas virtuais e 100 *cloudlets*.

A arquitetura de processamento, o sistema operacional, o *hipervisor* utilizado e os custos de utilização são atributos únicos de um *datacenter*, sendo automaticamente

aplicados a todos os seus nós. Por outro lado, a capacidade de processamento, a capacidade de memória de sistema e a capacidade de armazenamento são características dos *hosts*, que podem ser customizadas individualmente, criando-se *datacenters* com recursos heterogêneos.

A Tabela 4.1 apresenta as características dos *datacenters* modelados. Os valores de custo de utilização de Processamento, Memória, Armazenamento e Largura de Banda são fornecidos em unidades monetárias (u.m.). Os três *datacenters* simulados são compostos por nós homogêneos, ou seja, todos os *hosts* de um mesmo *datacenter* possuem a mesma configuração/característica.

Tabela 4.1 - Configurações dos Datacenters Modelados

Características do Datacenter			
Datacenter	**Goiânia**	**Anápolis**	**Brasília**
Arquitetura	x86	x86	x86
Sistema Operacional	Linux	Linux	Linux
Hipervisor	gyn	aps	bsb
Número de Host	1	1	2
Custo de Processamento (por segundo)	3.0 u.m.	3.0 u.m.	2.0 u.m.
Custo de Memória (por MB)	0.05 u.m.	0.05 u.m.	0.06 u.m.
Custo de Armazenamento (por MB)	0.001 u.m.	0.001 u.m.	0.002 u.m.
Custo de Largura Banda (por Bit)	0.03 u.m.	0.03 u.m.	0.05 u.m.

A Tabela 4.2 apresenta a configuração individual dos *hosts*. Estes atributos exercem forte influência na execução do processo de simulação, para que uma máquina virtual seja efetivamente alocada, é necessário que ao menos um *host* do *datacenter* seja capaz de atender simultaneamente a seus requisitos de recursos e ao perfil de utilização do cliente.

Tabela 4.2 - Configurações dos *Hosts* Modelados

Características dos Hosts				
Host	Goiânia	Anápolis	Brasília	
Memória RAM	14089 Mbyte	14089 Mbyte	6400 Mbyte	7680 Mbyte
Capacidade de Armazenamento / HD	100 Gbyte	100 Gbyte	50 Gbyte	50 Gbyte
CPUs/Core	One-core	One-core	Quad-core	Dual-core
MIPS por processador	33624	33624	12712	20912

A composição do cenário simulado é composta por um cliente que inicialmente cria 100 tarefas (*cloudlets)*. O cliente envia as *cloudlets* para o *broker* (intermediador) que gerencia para quem serão enviadas as tarefas. O *broker* tem a sua disposição três *datacenters*, Anápolis, Goiânia e Brasília, os quais possuem as máquinas virtuais que executarão as *cloudlets* e as enviarão de volta após a execução. As máquinas virtuais devem ser alocadas em *hosts* que atendam aos requisitos mínimos de recursos necessários. Os *hosts* escolhidos devem pertencer a um *datacenter* que atenda ao perfil de utilização do cliente.

A Tabela 4.3 apresenta as características das máquinas virtuais. Assim como os atributos dos *hosts*, os recursos exigidos por cada máquina virtual desempenham um papel importante no fluxo de execução da simulação, pois o *host* tem que possuir capacidade correspondente às características das máquinas virtuais. Caso alguma das características das máquinas virtuais somadas ou individuais ultrapassem as características do *host* ela não será alocada e como para a execução da aplicação é necessário ao menos uma máquina virtual para a execução pode ocorrer a inviabilidade da execução da mesma.

Tabela 4.3 - Características das Máquinas Virtuais

Características das Máquinas Virtuais						
IDVM	**HD**	**RAM**	**MIPS**	**Largura de Banda**	**CPUs**	**Name**
0	10 G	768 MB	1128	1.024	1	gyn
1	10 G	1 G	1256	1.024	2	gyn
2	10 G	1.280 MB	1384	1.024	2	gyn
3	10 G	1,5 G	1512	1.024	2	gyn
4	10 G	1.792MB	1640	1.024	2	gyn
5	10 G	2 G	1768	1.024	2	gyn
6	10 G	1.792MB	1640	1.024	3	gyn
7	10 G	1,5 G	1512	1.024	3	gyn
8	10 G	1.280MB	1384	1.024	3	gyn
9	10 G	1 G	1256	1.024	3	gyn
10	10 G	768 MB	1128	1.024	1	aps
11	10 G	1 G	1256	1.024	2	aps
12	10 G	1.280MB	1384	1.024	2	aps
13	10 G	1,5 G	1512	1.024	2	aps
14	10 G	1.792MB	1640	1.024	2	aps
15	10 G	2 G	1768	1.024	2	aps
16	10 G	1.792 MB	1640	1.024	3	aps
17	10 G	1,5 G	1512	1.024	3	aps
18	10 G	1.280MB	1384	1.024	3	aps
19	10 G	1 G	1256	1.024	3	aps
20	10 G	768 MB	1128	1.024	1	bsb
21	10 G	1 G	1256	1.024	2	bsb
22	10 G	1.280MB	1384	1.024	2	bsb

23	10 G	1,5 G	1512	1.024	2	bsb
24	10 G	1.792 MB	1640	1.024	2	bsb
25	10 G	2 G	1768	1.024	2	bsb
26	10 G	1.792 MB	1640	1.024	3	bsb
27	10 G	1,5G	1512	1.024	3	bsb
28	10 G	1.280 MB	1384	1.024	3	bsb
29	10G	1 G	1256	1.024	3	bsb

A Tabela 4.4 apresenta as características das *cloudlets*. Todas as *cloudlets* executadas possuem o mesmo tamanho de entrada e saída, 300 bytes. Para definir o total de instruções (milhões de instruções - MI) foi adotado que se o identificador for múltiplo de 2, é acrescentado 300 MI, se o identificador da *cloudlet* for múltiplo de 04 é diminuído 100 MI e se for múltiplo de 03 é diminuído 250 MI. A simulação considera tamanhos diferentes de instruções em função do tempo de execução para cada tarefa. Assim, o MI varia para não ter sempre o mesmo tempo de saída, ou seja, uma tarefa irá demorar mais que a outra para ser executada, dependendo do seu tamanho.

Tabela 4.4 - Características das *Cloudlets*

Características das *Cloudlets*			
ID	MIPS	Tamanho Entrada	Tamanho Saída
0	500	300	300
1	450	300	300
2	450	300	300
3	750	300	300
4	500	300	300
5	700	300	300
6	700	300	300
7	750	300	300
8	750	300	300
9	950	300	300
...	...	...	...
99	4750	300	300

A Figura 4.3 apresenta o fluxo de execução do cenário de simulação do *framework* CloudSim implementado para esta proposta. Primeiramente, a biblioteca do Cloudsim é iniciada. Os *datacenters* são criados, que no caso deste modelo são três: Anápolis, Brasília e Goiânia. Em seguida é criado o *broker*, as 30 máquinas virtuais e as 100 *cloudlets* a serem executadas durante a simulação. Após a criação dos itens necessários para a execução, a simulação é iniciada. A classe DataCenterMD é responsável por verificar se

existe *tarefas/cloudlets* a serem executadas e submete-las ao processo de execução. Existindo *cloudlets* submetidas é efetuado o cálculo da confiança da máquina virtual, de acordo com o modelo proposto e métricas definidas. O cálculo da confiança da máquina virtual é efetuado pela classe Trust. Nesta classe tem-se implementado todas as regras de cálculo apresentadas na seção 3.1. Em seguida, é selecionada a máquina virtual com melhor limiar de confiança em relação à máquina *baseline* que executará a *cloudlet.* Caso exista mais *cloudlets* a serem executados, volta-se ao processo executado pela classe DataCenterMD. Caso não exista mais *cloudlets* a serem executadas, é criada uma lista com o resultado final das *cloudlets* executadas. A simulação é finalizada, o resultado é apresentado no *console* da ferramenta, cria-se um gráfico geral com todas as *cloudlets* executadas e confiança das máquinas virtuais que a executaram. Em seguida é criado um gráfico por cada máquina virtual, com seu respectivo limiar de confiança em relação a cada interação realizada e o processo é encerrado.

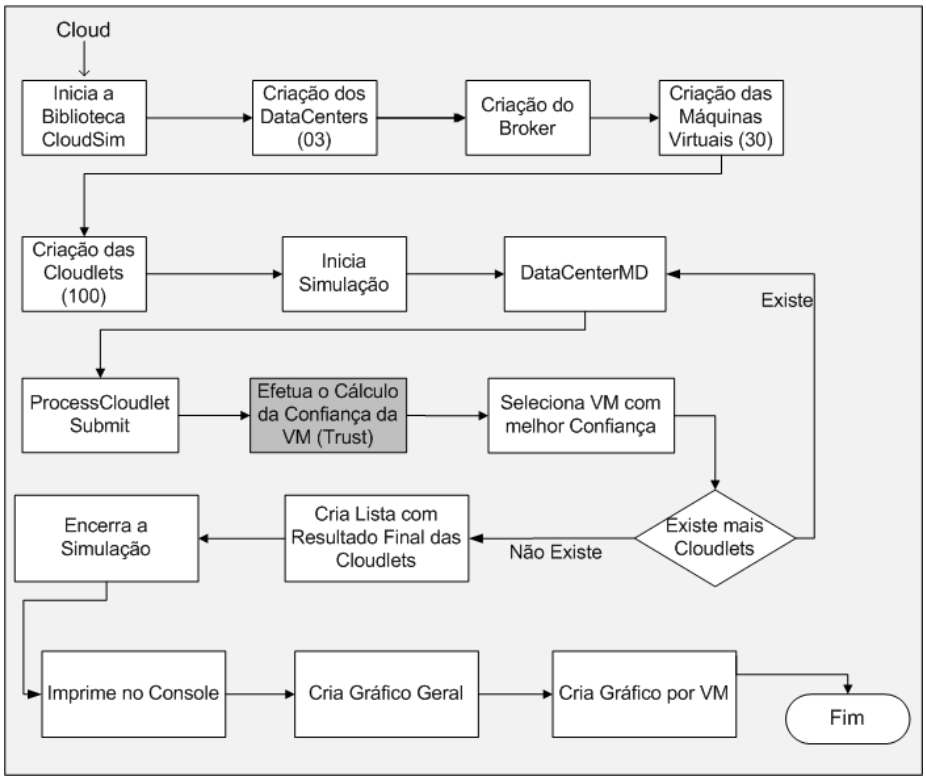

Figura 4.3 - Cenário de Execução da Proposta no CloudSim

4.3 - RESULTADOS E ANÁLISES

Com o ambiente de simulação do CloudSim definido e configurado, conforme apresentado nas Tabelas 4.1, 4.2, 4.3 e 4.4 e Figura 4.3, e uma vez atribuído os pesos as métricas, pode-se efetuar o cálculo da confiança de um nó através da execução dos cenários implementados no *framework*. Os resultados são divididos em três tópicos. O primeiro trata da apresentação dos resultados em que todas as tarefas/*cloudlets* foram executadas com sucesso. O segundo tópico trata dos resultados e análise em que as tarefas/*cloudlets* foram executadas com sucesso e insucesso. O terceiro trata dos resultados e análise do ambiente simulado com mudanças em uma das métricas.

4.3.1 - Cenário de Simulação com Todas as Tarefas Executadas com Sucesso

Para realizar a simulação do ambiente proposto inicialmente é necessário definir as configurações que serão consideradas ideais para que uma máquina seja confiável, logo, precisa-se definir a configuração da máquina *baseline* para poder comparar com os valores das demais máquinas virtuais do ambiente de simulação. Como no contexto desta aplicação existem tarefas pequenas e de baixa complexidade, é utilizado a configuração de *baseline* conforme padrão definido pela Amazon (Amazon, 2012), tentando assim se aproximar o mais possível da relação custo beneficio existentes nas nuvens reais, onde as configurações das máquinas são compatíveis com as cargas e serviços oferecidos. A configuração adotada neste trabalho é apresentada na Tabela 4.5.

Tabela 4.5 - Configuração da Máquina Baseline (Amazon, 2012)

Valores	
Tamanho Ideal de HD	163840 MB
Tamanho Ideal de Memória RAM	1740 MB
Tamanho Ideal de MIPS	5000
Tamanho Ideal da Largura de Banda	1024 Kbytes

Com o propósito de efetuar comparações e análise dos resultados em diversos cenários, foram executadas diversas simulações durante o trabalho proposto. Será apresentado nesta seção uma das simulações realizadas, conforme cenário descrito na

seção 4.2.1. Os resultados obtidos após a execução são apresentados na Tabela 4.6 (A Tabela 4.6 compreende a um dos resultados obtidos, de acordo com o cenário modelado).

Tabela 4.6 - Execução de Todas as Tarefas Com Sucesso

ID Cloudlet	Status Cloudlet	ID Datacenter	ID VM	Tempo Inicial	Tempo Execução	Tempo Final
0	SUCCESS	2	6	1,32	0,3	1,62
1	SUCCESS	2	16	0,67	0,3	0,97
2	SUCCESS	2	26	1,06	0,3	1,36
3	SUCCESS	2	5	2,76	0,3	3,06
4	SUCCESS	2	15	1,21	0,3	1,51
5	SUCCESS	2	25	2,25	0,3	2,55
6	SUCCESS	2	7	1,70	0,3	2,0
7	SUCCESS	2	17	1,32	0,3	1,62
8	SUCCESS	2	6	1,92	0,3	2,22
9	SUCCESS	2	16	1,89	0,3	2,19
10	SUCCESS	3	6	1,81	0,3	2,11
...	...	...	...	...	...	...
50	SUCESS	4	16	4,67	0,3	4,97
...	...	...	...	...	...	...
87	SUCESS	2	15	8,33	0,3	8,63
...	...	...	...	...	...	...
99	SUCCESS	3	7	8,08	0,3	8,38

A Figura 4.4 apresenta a confiança de cada máquina virtual após a execução das 100 *cloudlets*. Algumas máquinas não executaram *cloudlets* por não satisfazerem as condições de avaliação/verificação de uma máquina confiável para a execução de uma tarefa, em comparação a máquina de *baseline*. A Tabela 4.7 apresenta as Máquinas Virtuais que executaram *cloudlets* com sucesso.

Tabela 4.7 - Cloudlets/Tarefas Executadas Pelas Máquinas Virtuais

Máquina Virtual	Tarefas Executadas com sucesso
VM 05	13
VM 06	13
VM 07	12

VM 15	12
VM 16	13
VM 17	12
VM 25	12
VM 26	13

As demais máquinas virtuais do cenário simulado não executaram nenhuma tarefa/*cloudlet* por não satisfazerem o limiar de confiança desejável.

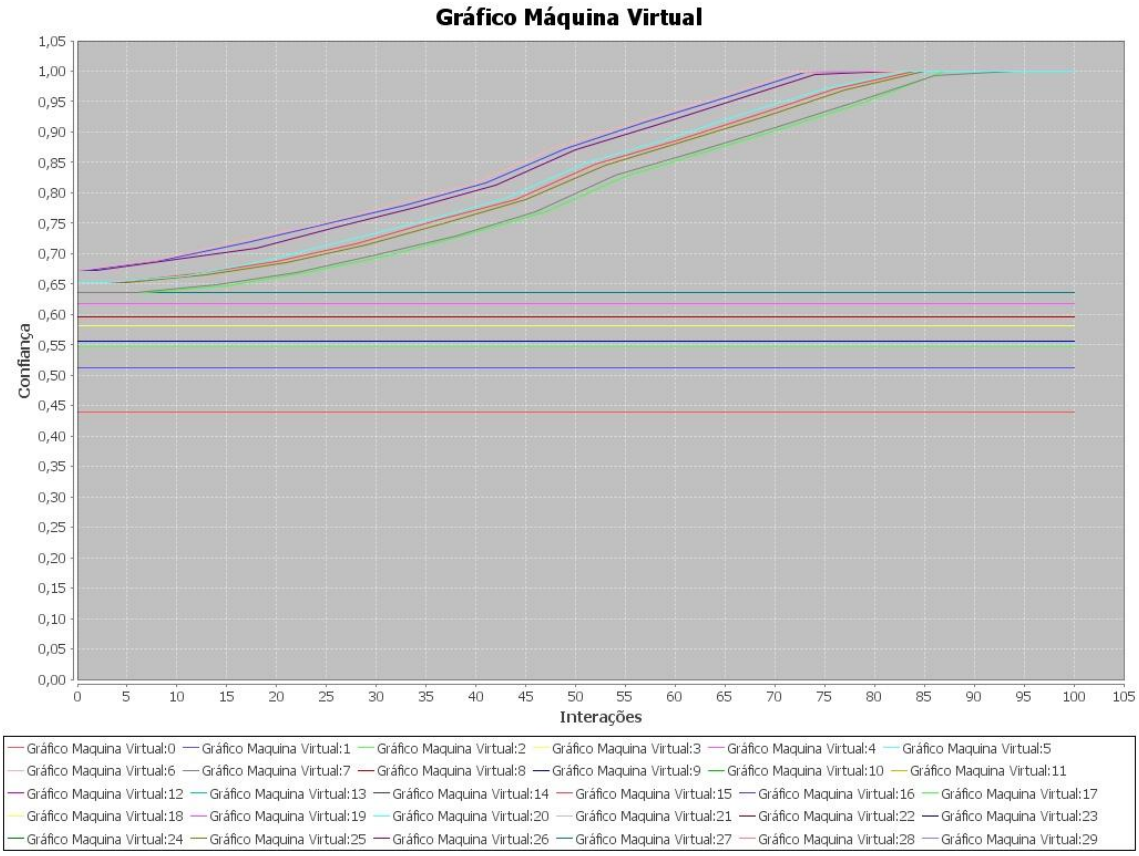

Figura 4.4 - Confiança das Máquinas Virtuais após a Execução

A Figura 4.5 apresenta o limiar de confiança da máquina virtual 25 que executou o menor número de *cloudlets* durante a simulação, 12 tarefas. O valor inicial de confiança da máquina foi 0.6516120114942529, calculado conforme equação 3.7, definida na seção 3.3.1. O valor final da confiança após a execução da décima segunda tarefa foi 1.0.

A Tabela 4.8 apresenta as cloudlets/tarefas executas e os valores de confiança da máquina virtual 25.

Tabela 4.8 - Limiar de Confiança da Máquina Virtual 25 ao Executar 12 Cloudlets com Sucesso.

Número da Tarefa	Limiar de Confiança da Máquina Virtual 25 a cada Interação
5	0.6516120114942529
13	0.6643405394359269
21	0.6860072061025935
29	0.7143197061025935
37	0.7508197061025935
45	0.7881113727692602
53	0.8447185156264032
61	0.8851872656264033
69	0.9249650434041812
77	0.969090043404181
85	1.0
93	1.0

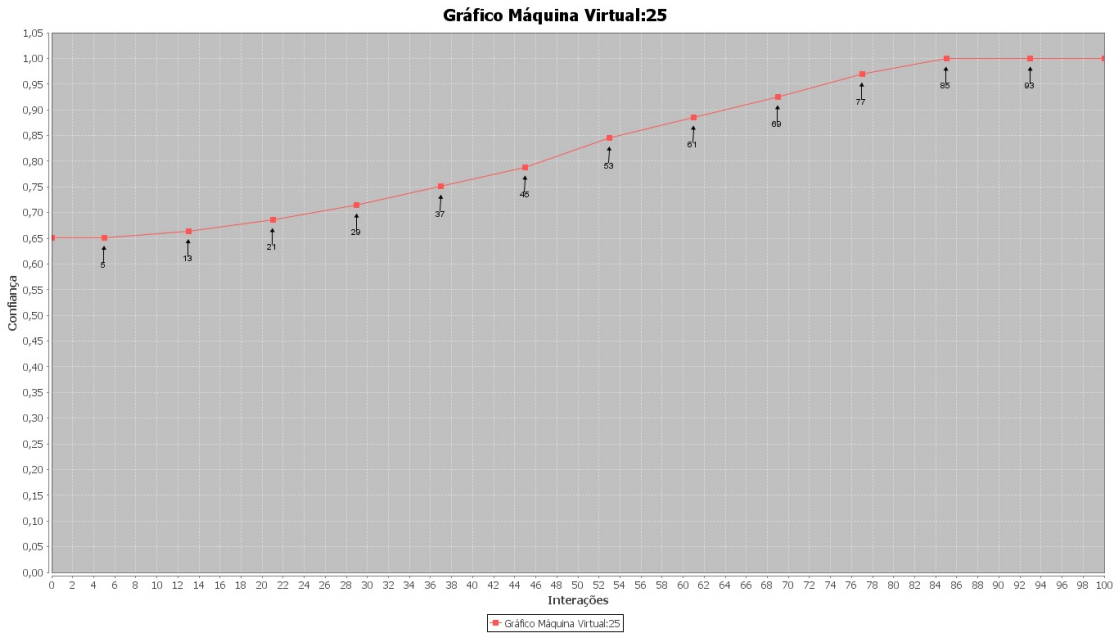

Figura 4.5 - Confiança da Máquina Virtual 25 ao Executar 12 Cloudlets

A Figura 4.6 apresenta o limiar de confiança da máquina virtual 16 que executou o maior número de cloudlets durante a simulação, 13 tarefas. O valor inicial de confiança da máquina foi 0.671875 e o valor final 1.0. Analisando o gráfico é possível identificar que ao executar uma tarefa com sucesso a confiança aumenta gradativamente.

A Tabela 4.9 apresenta as cloudlets/tarefas executas e os valores de confiança da máquina virtual 16.

Tabela 4.9 - Limiar de Confiança da Máquina Virtual 16 ao Executar 13 Cloudlets

Número da Tarefa	Limiar de Confiança da Máquina Virtual 16 a cada Interação
1	0.671875
9	0.6901934704704096
17	0.7176934704704097
25	0.7486309704704096
33	0.7795309704704098
41	0.815364303803743
49	0.873221446660886
57	0.915877696660886
65	0.9564332522164416
73	0.9972332522164417
81	1.0
89	1.0
97	1.0

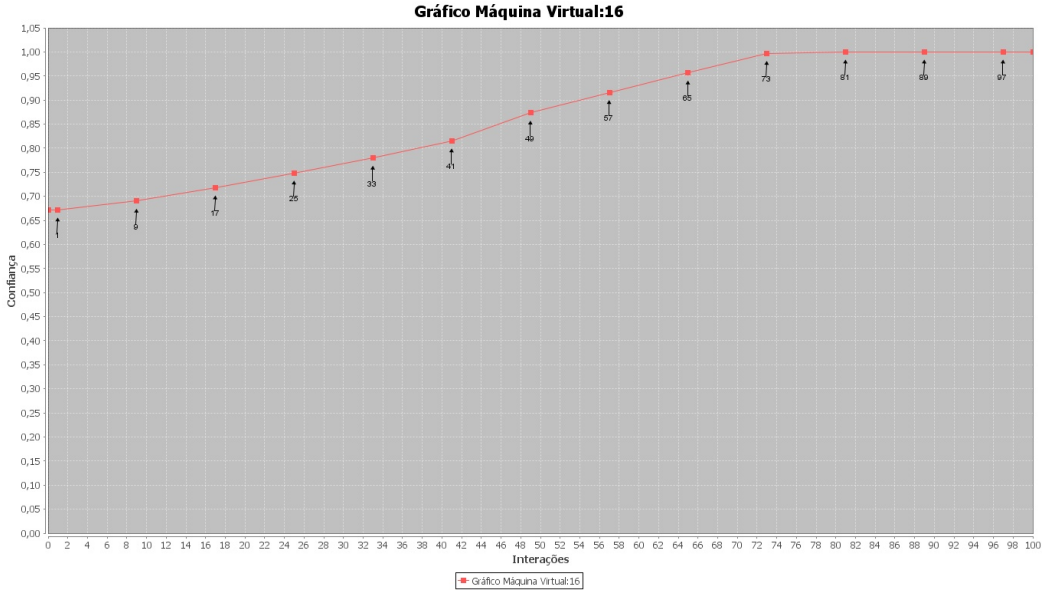

Figura 4.6 - Confiança da Máquina Virtual 16 ao Executar 13 Cloudlets

A Figura 4.7 apresenta o limiar de confiança da máquina virtual 12 que não executou nenhuma *cloudlet* durante a simulação, logo não há variação no gráfico. Todas as máquinas que não executaram nenhuma tarefa possuem o gráfico similar.

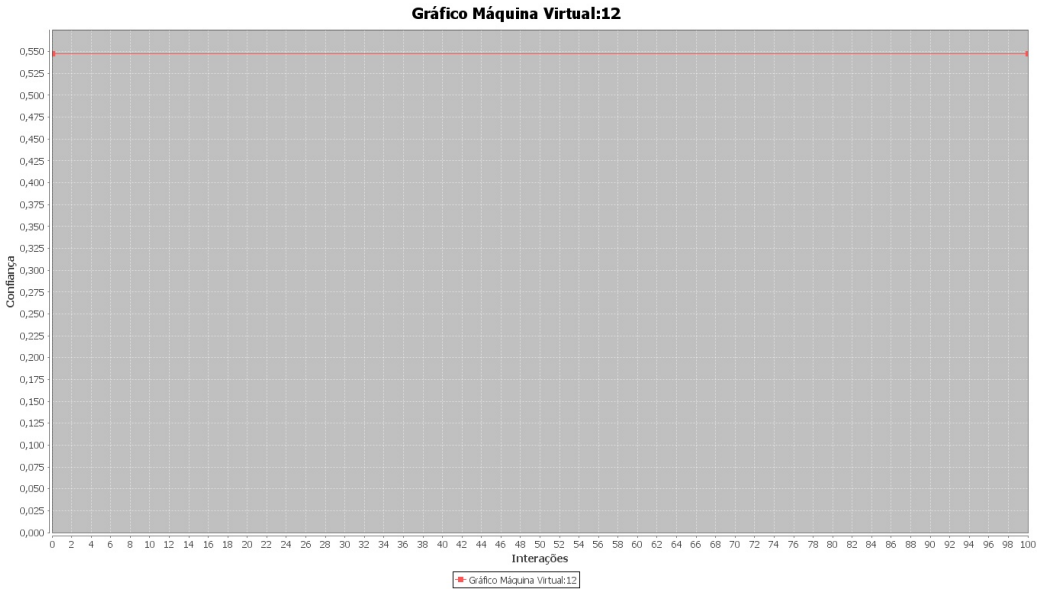

Figura 4.7- Confiança da Máquina Virtual 12 ao Executar 0 Cloudlets

A confiança de uma máquina virtual no modelo simulado aumenta em uma proporção, conforme a do ser humano, ou seja, quando um indivíduo executa uma atividade ou resolve um determinado problema para nós com sucesso, aumentamos nossa confiança no mesmo gradualmente. Assim, a cada *cloudlet* executada com sucesso, o valor de confiança de uma VM será aumentado em 2,5%, até o limiar de confiança igual a 0,85. Acima de 0,85 a confiança aumentará em 5% até alcançar o limiar máximo de 1.0.

Caso uma máquina não execute uma determinada tarefa com sucesso, ou seja, não resolve o nosso problema, perdemos a confiança. O peso da desconfiança normalmente é maior do que o peso da confiança. Assim, em nosso modelo simulado o limiar de desconfiança é de 5% a cada tarefa executada sem sucesso.

Observando os resultados, a partir das simulações realizadas no ambiente CloudSim em que todas as tarefas foram executadas com sucesso, é possível identificar o limiar de confiança das máquinas virtuais que executaram as *cloudlets.*

Conforme as informações de referência, um nó confia em outro a partir do valor de confiança $T \geq 0.6$. Como a confiança reflete o funcionamento das relações de confiança entre os nós da nuvem privada e o cliente e todas as máquinas virtuais estão acima do valor de referência, logo, são consideradas confiáveis. Assim a confiança do nó tende a ir aumentando à medida que as interações com sucesso ocorrem, conforme pode ser observado nos resultados obtidos.

4.3.2 - Cenário de Simulação com as Tarefas Executadas com Sucesso e sem Sucesso

Na tentativa de simular um ambiente mais próximo da realidade, foi conduzido um cenário de simulação em que as *cloudlets* não são totalmente executadas, permitindo que as máquinas virtuais mudem o seu comportamento no decorrer do tempo, refletindo um fato mais condizente com um ambiente real de computação em nuvem privada.

Durante o processo de configuração e programação deste cenário foi definido que uma tarefa sem sucesso será escolhida randomicamente e que irá ocorrer quando o número randomico for maior que 0.8, ou seja, a possibilidade de tarefa com sucesso neste cenário será de 80%. Desta forma, pode-se alterar o cenário de simulação, conforme o desejado.

Na simulação do cenário proposto, algumas máquinas também não executaram *cloudlets* por não satisfazerem as condições de verificação de uma máquina confiável para a execução de uma tarefa, em comparação a máquina de *baseline*.

A Figura 4.8 apresenta a confiança de cada máquina virtual após a execução das 100 *cloudlets* no cenário em que se tem máquinas executando tarefas com sucesso e sem sucesso.

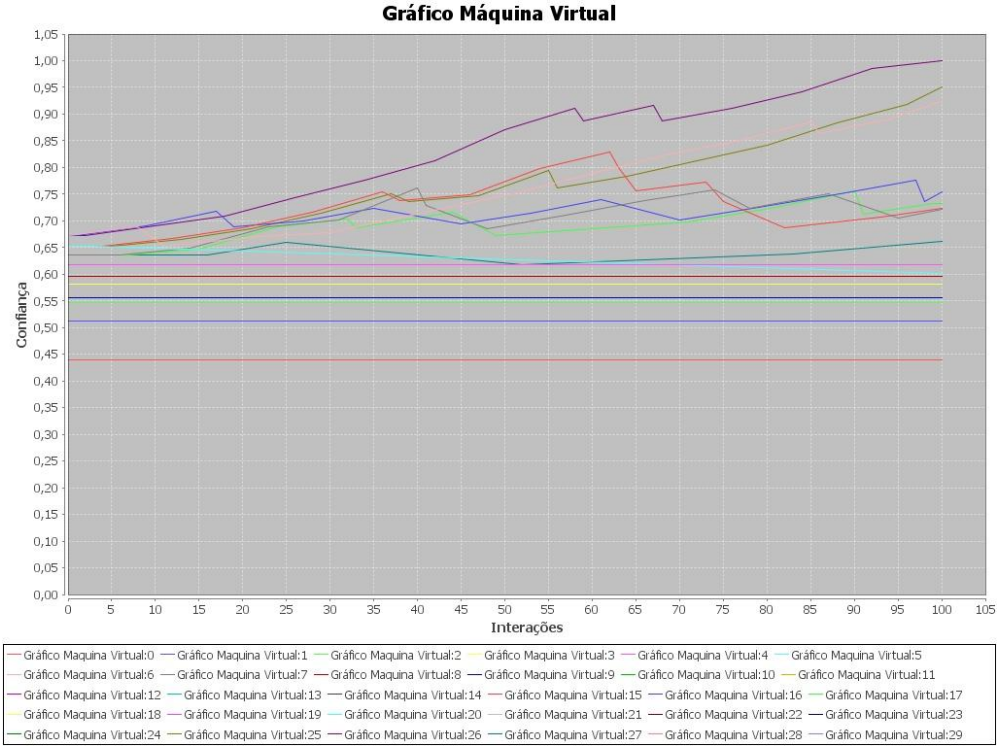

Figura 4.8- Confiança das Máquinas Virtuais após a Execução de Tarefas com Sucesso e sem Sucesso

A Tabela 4.10 apresenta as máquinas virtuais que executaram *cloudlets* neste cenário. As demais máquinas virtuais não executaram nenhuma tarefa/cloudlet por não satisfazerem o limiar de confiança desejável (máquina de *baseline*), conforme é apresentado na Figura 4.8.

Tabela 4.10 - Cloudlets/Tarefas Executadas Pelas Máquinas Virtuais com Sucesso e sem Sucesso

Máquina Virtual	Tarefas Executadas com sucesso	Tarefas Executadas sem sucesso	Total de Tarefas Executadas
VM 05	0	01	01
VM 06	11	02	13
VM 07	09	04	13
VM 15	10	05	15
VM 16	10	04	14
VM 17	09	03	12
VM 25	12	02	14
VM 26	12	02	14
VM 27	03	01	04

A Figura 4.9 apresenta o limiar de confiança da máquina virtual 15 que executou o maior número de *cloudlets* durante a simulação, 10 tarefas com sucesso e 05 sem sucesso. O valor inicial de confiança da máquina foi 0.6516120114942529 e o valor final 0.7078565131671932, conforme Tabela 4.11.

Tabela 4.11 – Limiar de Confiança da Máquina Virtual 15 ao Executar 15 Cloudlets

Número da Tarefa	Limiar de Confiança da Máquina Virtual 15 a cada Interação
4	0.6516120114942529
12	0.6678405394359269
20	0.6871738727692602
28	0.7159238727692602
36	0.7541738727692604
38	0.7380560301220356
46	0.7491927523910854
54	0.7975677523910855
62	0.8290121968355298
63	0.799216491247195
65	0.7560346730653769
73	0.7730519277223226
75	0.7361898465005265

82	0.6861898465005264
93	0.7078565131671932

A variação do limiar de confiança da VM 15 foi calculada de acordo com as interações com sucesso e sem sucesso. A cada interação realizada com sucesso o valor de confiança é aumentado em 2,5% e a cada interação realizada sem sucesso, o valor é decrementado em 5% do valor do limiar, conforme peso estabelecido.

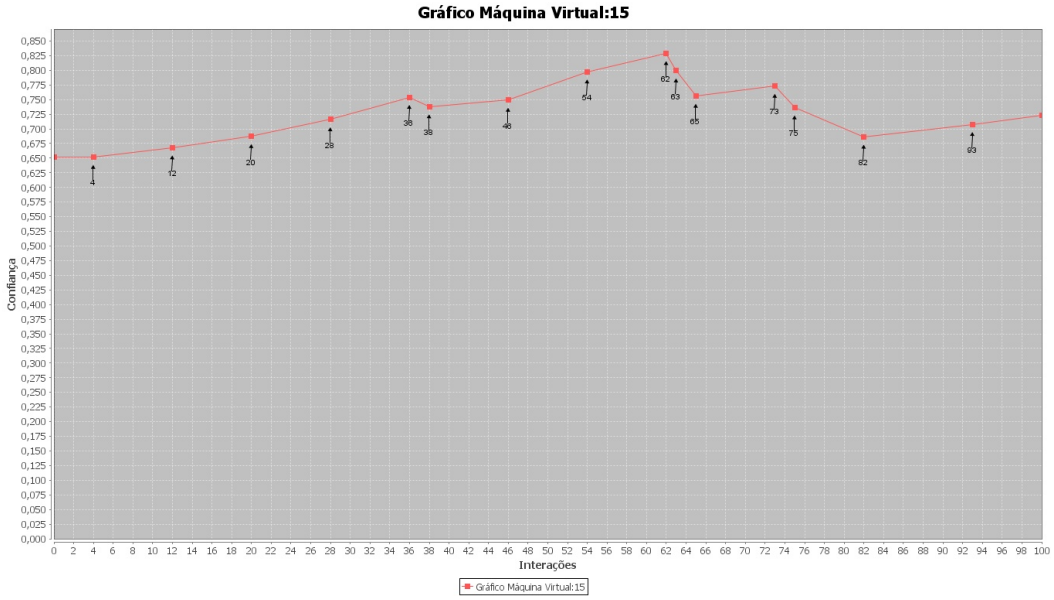

Figura 4.9 - Confiança da Máquina Virtual 15 ao Executar 10 Cloudlets com Sucesso e 05 sem Sucesso

A Figura 4.10 apresenta o limiar de confiança da máquina virtual 27 que executou o menor número de *cloudlets* durante a simulação, 03 tarefas com sucesso e 01 sem sucesso. O valor inicial de confiança da máquina foi 0.6351177586206896 e o valor final 0.637201091954023.

A Tabela 4.12 apresenta o limiar de confiança da máquina virtual 27 ao executar as 04 cloudlets/tarefas.

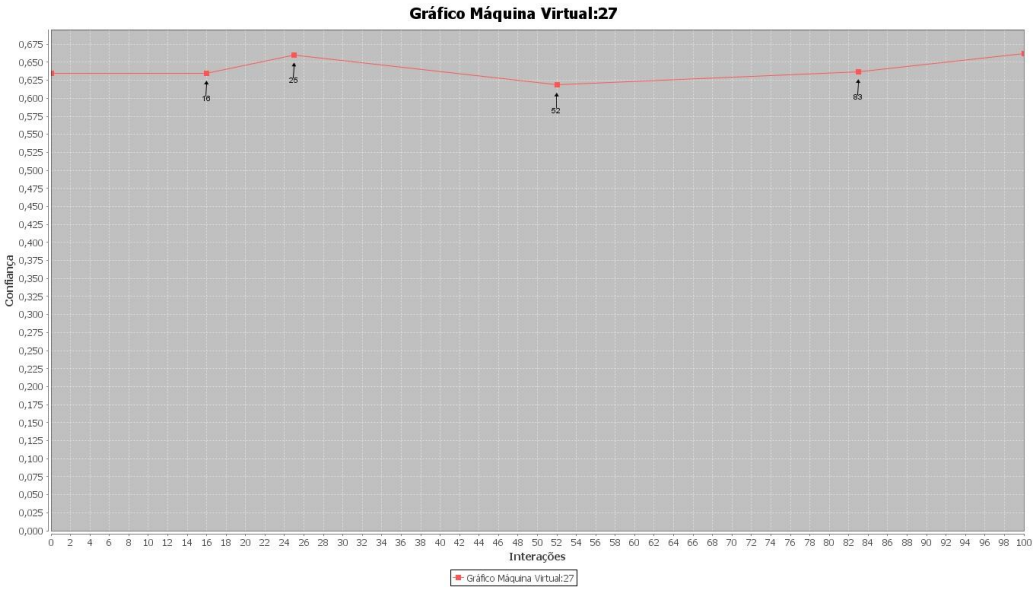

Figura 4.10 - Confiança da Máquina Virtual 27 ao Executar 03 Cloudlets com Sucesso e 01 sem Sucesso

Tabela 4.12 - Limiar de Confiança da Máquina Virtual 27 ao Executar 04 Cloudlets

Número da Tarefa	Limiar de Confiança da Máquina Virtual 15 a cada Interação
16	0.6351177586206896
25	0.6601177586206897
52	0.6184510919540229
83	0.637201091954023

Assim como na simulação em que todas as máquinas executaram as *cloudlets* com sucesso, nesta simulação as máquinas que não executaram nenhuma *cloudlet*, permaneceram com o gráfico estático, ou seja, não há nenhuma variação no gráfico, conforme apresentado na Figura 4.7, logo todas as máquinas virtuais que não executaram nenhuma tarefa, possuem o gráfico simular.

4.3.3 - Cenário de Simulação com Modificações nas Configurações das Máquinas Virtuais

Nesta etapa foi simulada a execução das *cloudlets* em que o ambiente configurado foi alterado. Escolheu-se a máquina virtual 15 para efetuar a alteração, uma vez que a mesma foi a VM que executou o maior número de *cloudlets* na execução/simulação apresentação na seção 4.3.2.

A configuração da VM 15 nas simulações realizadas e apresentadas nas seções 4.3.1 e 4.3.2, foi de acordo com a Tabela 4.13.

Tabela 4.13- Características da Máquina Virtual 15

Característica da Máquina Virtual 15						
IDVM	**HD**	**RAM**	**MIPS**	**Largura de Banda**	**CPUs**	**Name**
15	10 G	2 GB = 2048MB	1768	1.024	2	aps

No primeiro cenário de simulação foi alterado o valor de configuração de um dos parâmetros, capacidade de processamento, que é compreendido pela Memória RAM, MIPS e CPUs.

4.3.3.1 - Cenário de Simulação com Modificações na Capacidade de Processamento

Neste cenário simulado a configuração/características da máquina virtual 15 foi definida conforme Tabela 4.14, onde alteramos a memória RAM para 1536 MB.

Tabela 4.14 - Características Modificadas da Máquina Virtual 15

Característica da Máquina Virtual 15						
IDVM	**HD**	**RAM**	**MIPS**	**Largura de Banda**	**CPUs**	**Name**
15	10 G	1.536 MB	1768	1.024	2	aps

A Tabela 4.15 apresenta as máquinas virtuais que executaram *cloudlets* neste cenário. As demais máquinas virtuais não executaram nenhuma tarefa/*cloudlet* por não satisfazerem o limiar de confiança desejável, conforme é mostrado na Figura 4.11.

Tabela 4.15 - Cloudlets/Tarefas Executadas Pelas Máquinas Virtuais com Sucesso e sem Sucesso com Modificações na Capacidade de Processamento

Máquina Virtual	Tarefas Executadas com sucesso	Tarefas Executadas sem sucesso	Total de Tarefas Executadas
VM 04	10	01	11
VM 05	02	02	04
VM 06	12	03	15
VM 07	00	01	01
VM 14	10	03	13
VM 15	06	01	07
VM 16	11	03	14

VM 24	00	01	01
VM 25	03	01	04
VM 26	12	03	15
VM 27	12	03	15

Conforme Figura 4.11 é possível verificar que ao alterar a configuração de uma única VM, todo o cenário de simulação foi alterado, não impactando somente na VM 15 e sim em várias outras VMs.

A Figura 4.12 apresenta o limiar de confiança da máquina virtual 15 após a alteração da sua capacidade de processamento (RAM). Durante a simulação a VM 15 executou 06 tarefas com sucesso e 01 sem sucesso.

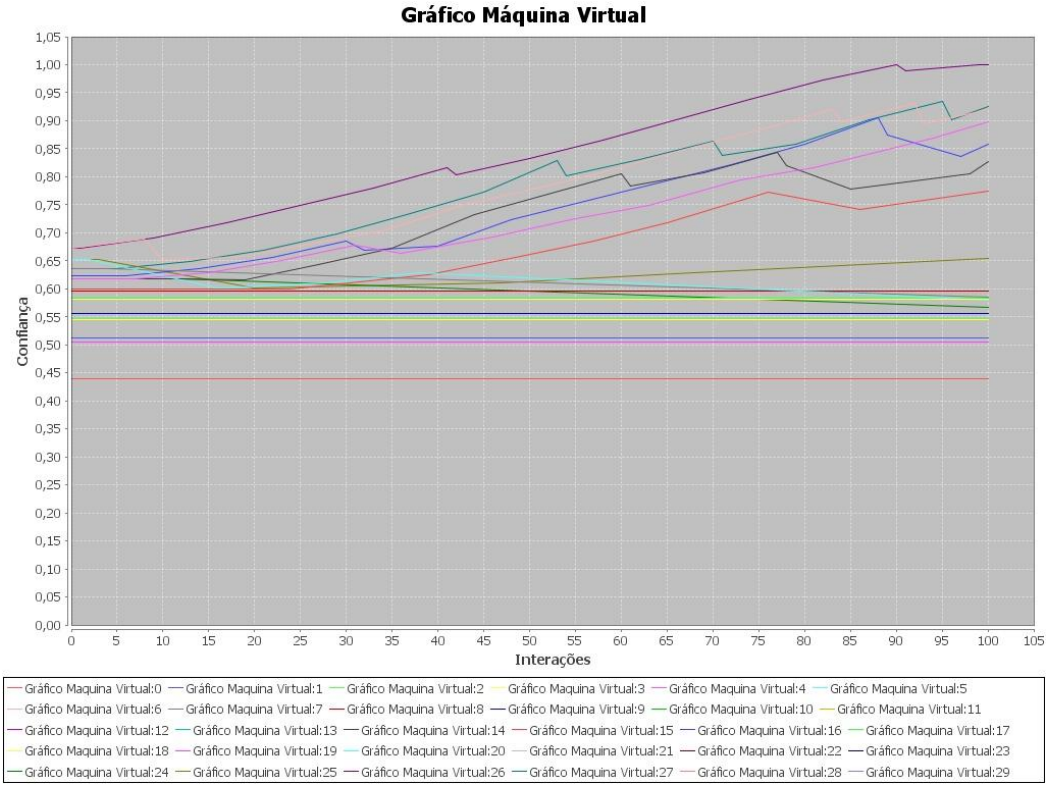

Figura 4.11- Confiança das Máquinas Virtuais após a Execução de Tarefas com a Configuração da VM 15 Modificada.

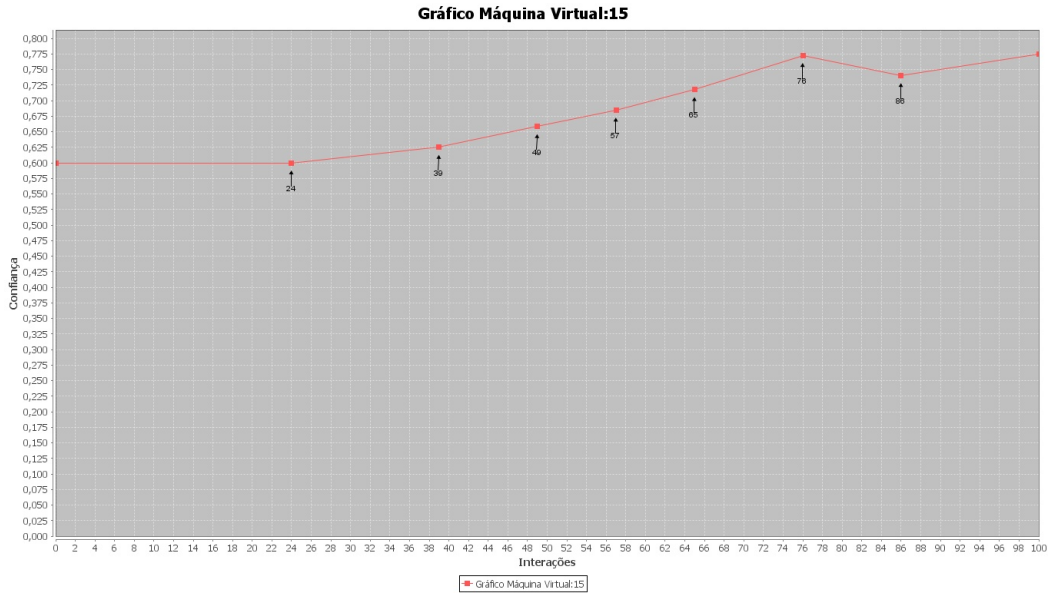

Figura 4.12 - Confiança da Máquina Virtual 15 ao Executar 06 Cloudlets com Sucesso e 01 sem Sucesso

O valor inicial do limiar de confiança da máquina virtual 15 foi 0.6001177586206896 e o valor final 0.740658822801445, conforme Tabela 4.16.

Tabela 4.16 - Limiar de Confiança da Máquina Virtual 15 ao Executar 07 Cloudlets

Número da Tarefa	Limiar de Confiança da Máquina Virtual 15 a cada Interação
24	0.6001177586206896
39	0.6251177586206896
49	0.6584510919540231
57	0.68500285592486
65	0.71800285592486
76	0.7728016799443022
86	0.740658822801445

4.3.3.2 - Cenário de Simulação com Modificações na Capacidade de Armazenamento

Neste cenário simulado a configuração/características da máquina virtual 15 foi definida conforme Tabela 4.17. A capacidade de processamento retornou aos seus valores de origem, ou seja, o valor da memória RAM retornou para 2048 MB e o valor da capacidade de Armazenamento, HD, foi alterado para 7 GB.

Tabela 4.17- Características Modificadas da Máquina Virtual 15

Característica da Máquina Virtual 15						
IDVM	**HD**	**RAM**	**MIPS**	**Largura de Banda**	**CPUs**	**Name**
15	7 G	2.048MB	1768	1.024	2	aps

A Tabela 4.18 apresenta as máquinas virtuais que executaram *cloudlets* neste cenário. As demais máquinas virtuais não executaram nenhuma tarefa/*cloudlet* por não satisfazerem o limiar de confiança desejável, conforme é mostrado na Figura 4.13.

Tabela 4.18 - Cloudlets/Tarefas Executadas Pelas Máquinas Virtuais com Sucesso e sem Sucesso com Modificações na Capacidade de Armazenamento

Máquina Virtual	Tarefas Executadas com sucesso	Tarefas Executadas sem sucesso	Total de Tarefas Executadas
VM 05	12	01	13
VM 06	06	04	10
VM 07	11	01	12
VM 15	10	01	11
VM 16	12	02	14
VM 17	09	01	10
VM 25	11	03	14
VM 26	13	02	15
VM 27	00	01	01

Conforme Figura 4.13 é possível verificar que ao alterar a configuração de uma única VM, novamente todo o cenário de simulação é alterado, não impactando somente na VM 15 e sim em várias outras VMs.

A Figura 4.14 apresenta o limiar de confiança da máquina virtual 15 após a alteração da sua capacidade de armazenamento (HD). Durante a simulação a VM 15 executou 10 tarefas com sucesso e 01 sem sucesso.

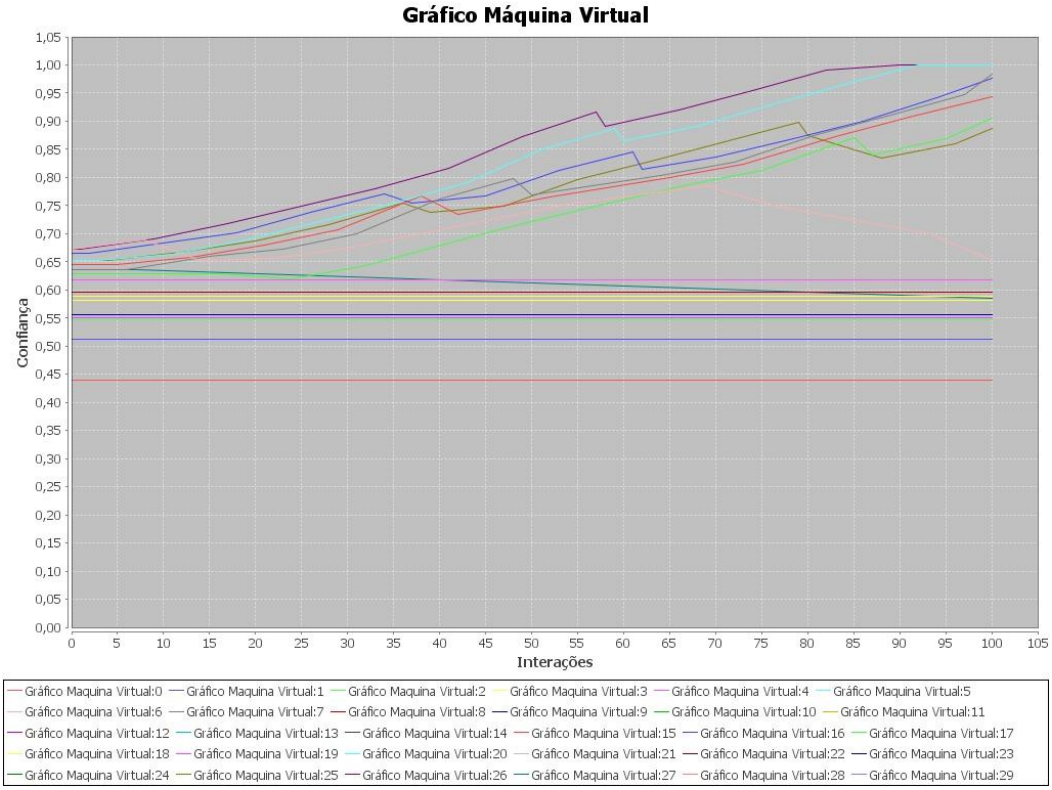

Figura 4.13- Confiança das Máquinas Virtuais após a Execução de Tarefas com a Configuração da VM 15 Modificada na Capacidade de Armazenamento.

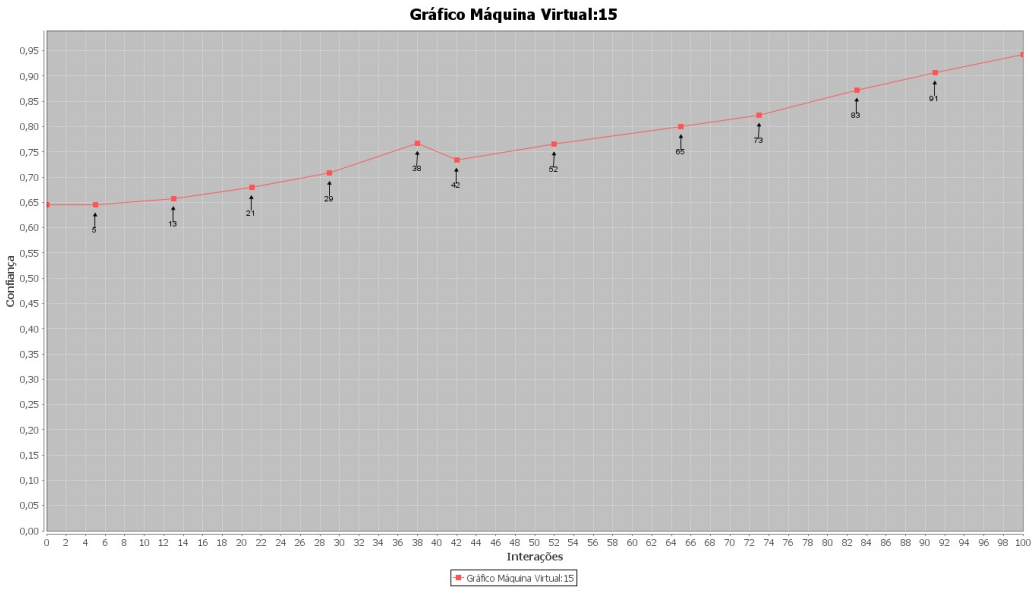

Figura 4.14 - Confiança da Máquina Virtual 15 ao Executar 10 Cloudlets com Sucesso e 01 sem Sucesso

O valor inicial do limiar de confiança da máquina virtual 15 foi 0.6450495114942529 e o valor final 0.9063290183760725, conforme Tabela 4.19.

Tabela 4.19 -Limiar de Confiança da Máquina Virtual 15 ao Executar 11 Cloudlets

Número da Tarefa	Limiar de Confiança da Máquina Virtual 15 a cada Interação
5	0.6450495114942529
13	0.6577780394359268
21	0.6794447061025936
29	0.7077572061025936
38	0.7670157949259241
42	0.7336824615925907
52	0.7658253187354479
65	0.800200318735448
73	0.8228899916113755
83	0.8724692860230407
91	0.9063290183760725

As simulações realizadas efetuando alterações em dois cenários permite observar que na primeira execução com o cenário inalterado foi executado um número de 15 tarefas/*cloudlets*, com 10 tarefas com sucesso e 05 sem sucesso na máquina virtual 15 (seção 4.3.2). É possível identificar que a capacidade de processamento tem um impacto maior na execução das tarefas, uma vez que ao alterar o valor da RAM no cenário inicial, seção 4.3.3.1, foram executadas 07 tarefas, 06 tarefas com sucesso e 01 sem sucesso. No cenário em que foi alterada a capacidade de armazenamento, HD, seção 4.3.3.2, teve se a execução de 10 tarefas com sucesso e 01 sem sucesso, totalizando 11 tarefas executas, muito próximo ao cenário de origem (seção 4.3.2).

Na simulação em que foi alterado o valor da capacidade de armazenamento, o valor da memória RAM foi retornado ao seu valor de origem. Assim, pode-se concluir que a capacidade de processamento influencia de maneira significativa na execução das *cloudlets*, em relação a capacidade de armazenamento, HD. A Tabela 4.20 apresenta o comparativo realizado nas execuções com os referidos cenários.

Tabela 4.20 - Comparativo de Cloudlets/Tarefas Executadas nos Cenários Simulados pela Máquina Virtual 15

Ambientes Simulados	Tarefas Executadas com sucesso	Tarefas Executadas sem Sucesso	Total de Tarefas Executadas
Cenário Inicial	10	05	15 Tarefas

Cenário RAM Alterada	06	01	07 Tarefas
Cenário HD Alterado	10	01	11 Tarefas

4.3.3.3. - Cenário de Simulação com Modificações na Capacidade de Processamento e Capacidade de Armazenamento

Neste cenário simulado a configuração/características da máquina virtual 15 foi definida conforme Tabela 4.21, onde alteramos o HD para 7 G e memória RAM para 1536 MB.

Tabela 4.21 - Características Modificadas da Máquina Virtual 15 (HD e RAM)

Característica da Máquina Virtual 15						
IDVM	**HD**	**RAM**	**MIPS**	**Largura de Banda**	**CPUs**	**Name**
15	7 G	1.536MB	1768	1.024	2	aps

A Tabela 4.22 apresenta as máquinas virtuais que executaram *cloudlets* neste cenário. As demais máquinas virtuais não executaram nenhuma *cloudlet* por não satisfazerem o limiar de confiança desejável. O resultado da simulação é apresentado na Figura 4.15.

Tabela 4.22 - Cloudlets/Tarefas Executadas Pelas Máquinas Virtuais com Sucesso e sem Sucesso com Modificações na Capacidade de Processamento e Capacidade de Armazenamento

Máquina Virtual	Tarefas Executadas com sucesso	Tarefas Executadas sem sucesso	Total de Tarefas Executadas
VM 03	00	01	01
VM 04	12	02	14
VM 05	08	02	10
VM 06	09	06	15
VM 07	01	02	03
VM 08	08	02	10
VM 13	03	01	04
VM 14	00	01	01
VM 15	07	01	08
VM 16	00	01	01
VM 24	00	01	01
VM 25	12	02	14

VM 26	13	01	14
VM 27	01	02	03
VM 28	00	01	01

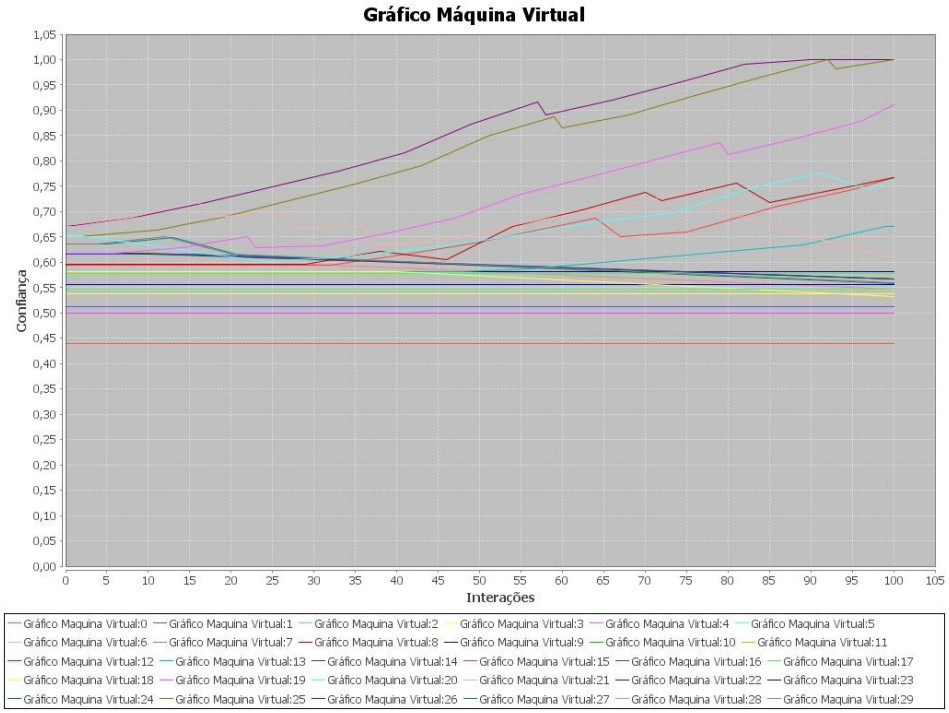

Figura 4.15 - Confiança das Máquinas Virtuais após a Execução de Tarefas com a Configuração da VM 15 Modificada na sua Capacidade de Processamento e Armazenamento.

A Figura 4.16 apresenta o limiar de confiança da máquina virtual 15 após a alteração da sua capacidade de processamento (HD e RAM). Durante a simulação a VM 15 executou 07 tarefas com sucesso e 01 sem sucesso. Avaliando os resultados obtidos com a alteração dos dois parâmetros de configuração da VM 15, também é possível identificar que todo o cenário simulado foi alterado, não impactando somente na máquina modificada, e sim nas demais máquinas virtuais. Além disso, o número de tarefas/*cloudlets* executadas com a alteração dos dois cenários foi bem próxima ao resultado obtido com a alteração efetuada na seção 4.3.3.1, onde é possível identificar novamente que a capacidade de processamento possui impacto maior nos resultados da simulação.

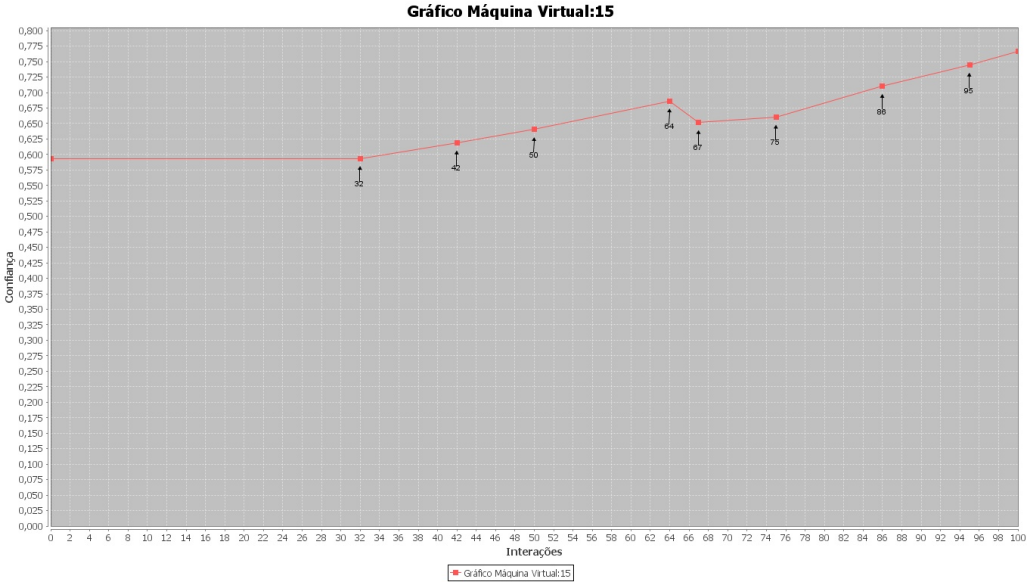

Figura 4.16 - Confiança da Máquina Virtual 15 ao Executar 07 Cloudlets com Sucesso e 01 sem Sucesso

O valor inicial do limiar de confiança da máquina virtual 15 foi 0.5935552586206897 e o valor final 0.7442351812748581, conforme Tabela 4.23.

Tabela 4.23 - Limiar de Confiança da Máquina Virtual 15 ao Executar 08 Cloudlets

Número da Tarefa	Limiar de Confiança da Máquina Virtual 15 a cada Interação
32	0.5935552586206897
42	0.6185552586206897
50	0.6402076105818058
54	0.6864683466109688
67	0.6514683466109688
75	0.6602111892581934
86	0.7098601812748581
95	0.7442351812748581

4.4 – SÍNTESE DO CAPÍTULO

O objetivo deste capítulo foi implementar e analisar a proposta deste trabalho na perspectiva da troca confiável de arquivos em um ambiente de computação em nuvem privada, utilizando o framework CloudSim (Rajkumar et al, 2009) .

No primeiro cenário simulado, conforme definido na seção 4.3.1, todas as tarefas/*cloudlets* foram executadas com sucesso e as máquinas virtuais com maior valor de

confiança tiveram seus valores acrescidos gradualmente a cada tarefa executada com sucesso. No cenário de simulação definido na seção 4.3.2 as máquinas virtuais executaram tarefas com sucesso e insucesso e tiveram seus valores de confiança acrescidos e decrementados conforme o esperado. Neste cenário, o número de *cloudlets* executadas pelas máquinas virtuais variou, pois a escolha da tarefa a ser executada sem sucesso foi selecionada aleatoriamente. Neste cenário, mais uma máquina virtual, a VM 27, foi adicionada na execução das tarefas, diferentemente do cenário 4.3.1, em que a VM 27 não executou nenhuma *cloudlet*, comportamento este esperado, uma vez que ao executar uma tarefa sem sucesso, o valor de confiança da máquina virtual é decrementado, surgindo assim máquinas virtuais com valores de confiança mais adequados em relação à máquina de baseline.

No cenário definido na seção 4.3.3 é possível avaliar o impacto das métricas adotadas no cálculo da confiança das máquinas virtuais. De acordo com o cenário simulado nas seções 4.3.3.1 e 4.3.3.2 é identificado que a capacidade de processamento possui um impacto maior no cálculo da confiança. Foram efetuadas várias alterações nos valores destas métricas durante as simulações e em todos os resultados obtidos a capacidade de processamento resultou em um número menor de *cloudlets* executadas.

Dentro de uma perspectiva geral, os resultados obtidos com a simulação utilizando a ferramenta CloudSim, demonstram a viabilidade da aplicação da confiança em ambientes de computação em nuvem, segundo a proposta do modelo elaborado. Nos resultados foi possível perceber como as métricas adotadas influenciam na confiança individual de cada máquina virtual como um todo porque a confiança é calculada segundo os valores das mesmas.

5 - CONCLUSÕES

A computação em nuvem tem sido foco de pesquisa em diversos trabalhos recentes, os quais demonstram a importância e necessidade de um modelo de confiança que garanta a troca confiável e segura de arquivos. É uma área promissora a ser explorada através de pesquisas e análises experimentais, utilizando a confiança computacional para amenizar os problemas existentes nos aspectos relacionados à segurança, confiança e reputação para a garantia da troca de informações íntegras em ambientes de nuvem privada, reduzindo a possibilidade de falha e ou alteração de informações na troca de arquivos, envolvendo métricas que sejam capazes de representar ou mapear o grau de confiança de um nó da rede a fim de realizar a troca de arquivos em uma nuvem privada.

A metodologia utilizada neste trabalho, dividida em fases, demonstrou-se eficaz para o direcionamento da construção do modelo proposto. A pesquisa bibliográfica realizada identifica o estado da arte sobre segurança, confiança e reputação envolvendo a computação em nuvem, as quais foram relevantes e imprescindíveis ao desenvolvimento final deste trabalho e do modelo proposto.

A proposta discutida neste trabalho, de desenvolver um novo modelo de confiança para a troca confiável de arquivos em um ambiente de computação em nuvem privada, utilizando os conceitos de confiança e reputação, tem se mostrado promissora, devido à identificação dos problemas e vulnerabilidades relacionados a segurança, privacidade e confiança que um ambiente de computação em nuvem apresenta.

As simulações e resultados apresentados permitem identificar que as métricas adotadas influenciam diretamente no cálculo da confiança em um nó. As simulações futuras utilizando um ambiente real permitirão avaliar o comportamento dos nós em um ambiente de computação em nuvem privada, bem como o histórico de suas iterações e valores assumidos durante toda a execução das máquinas.

A utilização da plataforma aberta, *CloudSim* (Rajkumar et al, 2009), para executar as simulações dos cenários adotados, permitiu calcular a tabela de confiança dos nós (máquinas virtuais) e selecionar os considerados mais confiáveis. Além disso, avaliou-se a adequação das métricas utilizadas no modelo de confiança proposto, possibilitando identificar e selecionar as mais adequadas em relação ao histórico do comportamento dos nós pertencentes ao ambiente analisado. Verificou-se que a confiança permite efetivamente a escolha da máquina mais adequada para efetuar a troca de arquivos.

5.1 - TRABALHOS FUTUROS

Como proposta de trabalhos futuros são indicadas alguns pontos que podem ser evoluídos.

Existe a necessidade de desenvolver um mecanismo para detectar a mudança do status de confiável a não confiável de um nó em um ambiente de computação em nuvem, porque é complexo averiguar se um nó é realmente confiável e quais os parâmetros que fizeram que o seu índice de confiança diminuísse e quais são os critérios que o tornaram confiável novamente. Por exemplo, considerando critérios de confiança e reputação relacionados com a disponibilidade, um nó com má reputação poderá assumir a identidade de um com boa reputação, burlando a confiança e por fim aumentando suas chances de efetuar a execução de mais *cloudlets*.

Outro ponto em aberto são as considerações sobre o consenso da confiança e da reputação. Os nós de alguma maneira devem concordar sobre confiar ou não confiar além de limiares previamente definidos. O consenso, assim como a confiança e a reputação, deve ser dinâmico e evoluir com o tempo. Além disso, estabelecer novas métricas, protocolos e modelos de confiança para ambientes de computação em nuvem.

A simulação em um ambiente de computação em nuvem real não foi realizada. Isto indica que existe a possibilidade de avaliar o modelo de confiança proposto para troca confiável de arquivos tratando a segurança, confidencialidade, disponibilidade, confiança e reputação em ambientes reais, utilizando um dos sistemas de distribuição de arquivos em nuvem. Esta simulação permitirá a avaliação e adequação das métricas propostas para o cálculo da confiança de um nó em um ambiente real, bem como a identificação de novas métricas e o seu real impacto em ambientes reais.

5.2 - PUBLICAÇÕES RELACIONADAS A ESTE TRABALHO

1) CANEDO, Edna Dias; Rafael Timóteo de Sousa Júnior; Robson de Oliveira Albuquerque and Fábio Lúcio Lopes de Mendonça. **File Exchange in a Private Cloud supported by a Trust Model**. In: CyberC 2012 – International Conference on Cyber-enabled distributed computing and knowledge discovery, 2012, October 10-12, Sanya, China. 2012 (Trabalho aceito).
2) CANEDO, Edna Dias; Rafael Timóteo de Sousa Júnior ; Rhandy Rafhael de Carvalho; Robson de Oliveira Albuquerque . **Trust Model For Private Cloud**. In: The International Conference on Cyber Security, CyberWarfare and Digital Forensic-CyberSec2012, 2012, Kuala Lumpur, Malasia. The International Conference on Cyber Security, CyberWarfare and Digital Forensic, 2012. p. 128-132. ISBN: 978-1-4673-1425-1. Kuala Lumpur, Malaysia.

3) CANEDO, Edna Dias; Rafael Timóteo de Sousa Júnior; Robson de Oliveira Albuquerque ; Flávio Elias Gomes de Deus . **File Exchange in Cloud**. In: 11th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (IEEE TrustCom-12), 2012, Liverpool, UK. 2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications TrustCom 2012, 2012. p. 1217-1222.
4) Canedo, Edna Dias; Rafael Timóteo de Sousa Júnior, Robson de Oliveira Albuquerque. **Trust Model for Reliable File Exchange In Cloud Computing**. International Journal of Computer Science and Information Technology, v.4, p.139 - 18, 2012. Home page: [http://airccse.org/journal/jcsit/0212csit01.pdf].
5) Canedo, Edna Dias; Robson de Oliveira Albuquerque, Rafael Timóteo de Sousa Júnior. **Review of Trust-based File Sharing in Cloud Computing**. In: The Fourth International Conference on Advances in Mesh Networks - MESH 2011 - August 21-27, 2011 - French Riviera, 2011, Nice/St Laurent du Var - France. IARIA Conference - MESH 2011, The Fourth International Conference on Advances in Mesh Networks. , 2011. p.44 – 50.
6) Canedo, Edna Dias; Robson de Oliveira Albuquerque, Rafael Timóteo de Sousa Júnior. **Trust Model for File Sharing in Cloud Computing**. In: CLOUD COMPUTING 2011: The Second International Conference on Cloud Computing, GRIDs, and Virtualization, 2011, Roma - Italia. IARIA Conference - CLOUD COMPUTING, International Conference on Cloud Computing, GRIDs, and Virtualization. , 2011. p.66 – 73.
7) Laerte Peotta de Melo; Edna Dias Canedo, Robson de Oliveira Albuquerque, Rafael Timóteo de Sousa Júnior. **Social Networks: Security and Privacy.** In: Procedings of the First International Conference on Forensic Computer Science Investigation (ICoFCS 2010), 2010, Brasília. Procedings of the First International Conference on Forensic Computer Science Investigation (ICoFCS 2010). Editorial Board, 2010. v.5. p.67 – 72.

REFERÊNCIAS BIBLIOGRÁFICAS

Abdul, A.. et al; A. Abdul-Rahman and S. Hailes. (1998). “A distributed trust model,” In Proceedings of the 1997 New Security Paradigms Workshop, 48-60.

ABNT NBR ISO/IEC 27002:2005. (2005). Tecnologia da Informação - Técnicas de Segurança - Código de Prática para a Gestão da Segurança da Informação. ISBN 978-85-07-00668-0.

Agrawal, D.. et al; Agrawal, D., Das, S., and Abbadi, A. E. (2010). Big Data and Cloud Computing: New Wine or just New Bottles? 36th International Conference on Very Large Data Bases, September 1317, Singapore. Proceedings of the VLDB Endowment, Vol. 3, No. 2, 1647–1648.

Albuquerque; Albuquerque, Robson de O. (2008). Uma proposta de um modelo de confiança computacional para grupos em ambientes distribuídos. Tese de Doutorado, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF.

Amazon; Amazon (2012). Amazon Web Services. Acessado em 01/06/2012. Disponível em: http://aws.amazon.com/pt/ec2/instance-types/.

Azure; Azure. (2011). Microsoft Azure. 2011. Acessado em 15/11/2011. Disponível em: http://www.microsoft.com/azure/.

Badger, L.. et al; Lee Badger, Robert Patt-Corner, Jeff Voas and T. Grance. (2011). “DRAFT Cloud Computing Synopsis and Recommendations". Special Publication 800-146. Recommendations of the National Institute of Standards and Technology. U.S May.

Beth T.. et al; Beth, T., Borcherding, M. & Klien, B. (1994). “Valuation of Trust in Open Networks”. Proceedings of the European Symposium on Research in Computer Security, Brighton, UK.

Borthakur; D. Borthakur. (2007). The Hadoop Distributed File System: Architecture and Design, The Apache Software Foundation, Acessado em 06/12/2011. Disponível em: http://hadoop.apache.org.

Calheiros, R.. et al; Calheiros, Rodrigo, N.; Rajiv Ranjan; Anton Beloglazov; De Rose, Cesar, A. F.; Buyya, Rajkumar. (2011). CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms, Software: Practice and Experience (SPE), Volume 41, Number 1, 23-50, ISSN: 0038-0644, Wiley Press, New York, USA, January.

Chang, E.. et al; T. Dillon, Chen Wu, and E. Chang. (2010) “Cloud Computing: Issues and Challenges,” 24th IEEE International Conference on Advanced Information Networking and Applications (AINA), 27-33. Australia.

Chen K.. et al; Chen Kang and Zen WeiMing. (2009) “Cloud computing: system instance and current research,” Journal of Software, 20-25:1337-1347.

DeCandia, G.. et al; DeCandia, G., Hastorun, D., Jampani, M., Kakulapati, G., Lakshman, A., Pilchin, A., Sivasubramanian, S., Vosshall, P., and Vogels. (2007). Dynamo: amazon’s highly available key-value store. Proceedings of twenty-first ACM SIGOPS symposium on Operating systems principles. ACM. New York, NY, USA.

Gambetta; Gambetta, Diego. (2000). “Can We Trust Trust?”, in Gambetta, Diego (ed.) Trust: Making and Breaking Cooperative Relations, electronic edition, Department of Sociology, University of Oxford, chapter 13, 213-237.

Garg, S.. et al; Garg, Saurabh Kumar; Versteeg, Steve; Buyya, Rajkumar. (2012). “SMICloud: A Framework for Comparing and Ranking Cloud Services”. 2011 Fourth IEEE International Conference on Utility and Cloud Computing. Melbourne, VIC, Australia.

Ghemawat, S.. et al; Ghemawat, S., Gobioff, H., and Leung, Shun-Tak. (2003). The google file system. Proceedings of the nineteenth ACM symposium on Operating systems principles ACM. New York, Volume 37 Issue 5, Dezembro. NY, USA.

Hwang, K.. et al; Hwang Kai, Sameer Kulkareni, and Yue Hu. (2009). “Cloud Security with Virtualized Defense and Reputation-Based Trust Mangement,” Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing (DASC), Chengdu, 717-722. China.

Huan, L.. et al; Li Huan-Chung Li, Po-Huei Liang, Jiann-Min Yang, and Shiang-Jiun Chen. (2010). “Analysis on Cloud-Based Security Vulnerability Assessment”. IEEE 7th International Conference on e-Business Engineering (ICEBE), pp. 490-494.

Josang, et al; Josang Audun, Roslan Ismail, Colin Boyd. (2007). A Survey of Trust and Reputation Systems for Online Service Provision. Decision Support Systems. Volume 43 Issue 2, March. Elsevier Science Publishers B. V. Amsterdam, The Netherlands, The Netherlands.

Lamsal; Lamsal, Pradip. (2006). “Understanding Trust and Security”. Department of Computer Science University of Helsiki, Finland, October 2001. Acessado em 13/02/2006. Disponível em: http://www.cs.helsinki.fi/u/lamsal/asgn/trust/UnderstandingTrustAndSecurity.pdf

Manuel, P.. et al; Manuel, P.D.; Thamarai Selvi, S.; Barr, Mostafa Ibrahim Abd-EI; (2009). “Trust Management System for Grid and Cloud Resources”. First International Conference on Advanced Computing. ICAC, 2009. Kuwait.

Marinos, A.. et al; A. Marinos and G. Briscoe. (2009). “Community cloud computing,” in First International Conference Cloud Computing, CloudCom, volume 5931 of Lecture Notes in Computer Science, 472–484. Springer.

Marsh, Stephen Paul. (1994). “Formalizing Trust as a Computational Concept”. Department of Computing Science and Mathematics, University of Stirling. Doctorate Thesis. April.

Mather. T.. et al; Mather, T., Kumaraswamy, S., e Latif, S. (2009). Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance. O'REILLY MEDIA, INC, USA.

Mell, P.. et al; P. Mell and T. Grance. (2009). "Draft NIST working definition of cloud computing”, v15, 21 de agosto.

Mell, P..et al; P. Mell and T. Grance. (2011). The NIST Definition of Cloud Computing (Draft). National Institute of Standards and Technology. http://csrc.nist.gov/groups/SNS/cloud-computing. 2009. 30 may 2011.

Miller; M. Miller. (2008). Cloud Computing – Web-Based Applications That Change the Way You Work and Collaborate Online, Que Publishing, Pearson Education, Canada.

Minqi, Z.. et al; Minqi Zhou, Rong Zhang, Dadan Zeng, and Weining Qian. (2010). “Services in the cloud computing era: a survey,” Software Engineering Institute. Universal Communication. Symposium (IUCS), 4th International. IEEE Shanghai, pp. 40-46. China. 978-1-4244-7821-7.

Nakamura, Emilio Tissato e Geus, Paulo Lício de (2007). Segurança de Redes em Ambientes Cooperativos.Editora Novatec. São Paulo - SP.

Neisse et al; Neisse, Ricardo; Holling, Dominik; Pretschner, Alexander. (2011). “Implementing Trust in Cloud Infrastructures”. Cluster, Cloud and Grid Computing (CCGrid), on 11th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing. Fraunhofer IESE, Kaiserslautern, Germany.

Patel, Jigar. (2007). “A Trust and Reputation Model for Agent-Based Virtual Organizations”. Thesis of Doctor of Philosophy. Faculty of Engineering and Applied Science. School of Electronics and Computer Science. University of Southampton. January.

Popovic, K.. et al; K. Popovic and Z. Hocenski. (2010). “Cloud computing security issues and challenges,” MIPRO, 2010 Proceedings of the 33rd International Convention, 344-349, 24-28 May.

Press, W.. et al; William H. Press, Saul A. Teukolsky, William T. Vetterling and Brian P. Flannery. (2007). Numerical Recipes: The Art of Scientific Computing, Third Edition, published by Cambridge University Press (ISBN-10: 0521880688, or ISBN-13: 978-0521880688).

Qiang, G.. et al; Qiang Guo; Dawei Sun; Guiran Chang; Lina Sun; Xingwei Wang. (2011). Modeling and Evaluation of Trust in Cloud Computing Environments. 3rd International Conference on Advanced Computer Control (ICACC 2011) Shenyang, China.

Qingsong, W.. et al; Qingsong Wei, Bharadwaj Veeravalli, Bozhao Gong, Lingfang Zeng, and Dan Feng. (2010). “CDRM: A Cost-Effective Dynamic Replication Management Scheme for Cloud Storage Cluster,” 2009 IEEE International Conference on Cluster Computing (CLUSTER), 188-196.

Rajkumar, B.. et al; Rajkumar Buyya; James Broberg; Andrzej Gościński. (2011). Cloud computing: principles and paradigms. Hoboken, N.J. : Wiley.

Rajkumar, B.. et al; Rajkumar Buyya, Rajiv Ranjan and Rodrigo N. Calheiros. (2009). “Modeling and Simulation of Scalable Cloud Computing Environments and the CloudSim Toolkit: Challenges and Opportunities”, Proceedings of the 7th High Performance Computing and Simulation Conference (HPCS 2009, ISBN: 978-1-4244-4907-1, IEEE Press, New York, USA), Leipzig, Germany, June 21-24.

Santos, N.. et al; N. Santos, K. Gummadi, and R. Rodrigues. (2009). “Towards Trusted Cloud Computing,” Proc. HotCloud. June.

Shirey, R (2000). RFC 2828. Internet Security Glossary. The Internet Society. Disponível em http://www.ietf.org/rfc/rfc2828.txt, acessado em 29 de fevereiro de 2012.

Stallings, William (2007). Criptografia e Segurança de Redes: Princípios e Práticas. Editora: PRENTICE HALL, Edição: 4ª. ISBN: 8576051192. São Paulo - SP.

Takabi, H.. et al; Takabi H., J. B. D. Joshi, and G. Ahn. (2010). “Security and Privacy Challenges in Cloud Computing Environments,” IEEE Security and Privacy, vol. 8, no. 6, 24-31, Nov./Dec, doi:10.1109/MSP.2010.186.

Udhayakumar, S..et al; Udhayakumar, S. and Chandrasekaran, S. and Tamilselvan, Latha and Ahmed, Fareez. (2011). An Adaptive Trust Model for Software Services in Hybrid Cloud Environment. Proceedings of the 15th WSEAS International

conference on Computers. Corfu Island, Greece, 497-502.

Uppoor, S.. et al; S. Uppoor, M. Flouris, and A. Bilas. (2010). "Cloud-based synchronization of distributed file system hierarchies," Cluster Computing Workshops and Posters (CLUSTER WORKSHOPS), IEEE International Conference, 1-4.

Velve, T.. et al; Anthony T. Velve, Toby J. Elsenpeter, Robert. (2011). Cloud Computing - Computação em Nuvem - Uma Abordagem Prática. Tradutor: Mei, Gabriela Edel. Editora: Alta Books. Edição: 1ª, 352.

Wang, H.. et al; Wang Han-zhang and Huang Liu-sheng. (2010). "An improved trusted cloud computing platform model based on DAA and Privacy CA scheme," IEEE International Conference on Computer Application and System Modeling (ICCASM 2010). 978-1-4244-7235-2.

Wang, J.. et al; Wang, J., Shao, Y., Jiang, S., e Le, J. (2009). Providing privacy preserving in cloud computing. Em International Conference on Test and Measurement, páginas 213–216. IEEE Computer Society. Hong Kong.

Wright, Thomas. (2004). Security, privacy, and anonymity. Volume 11. Issue 2, December 2004 ACM New York, NY, USA

Xiao; L.. et al; Xiao-Yong Li, Li-Tao Zhou, Yong Shi, and Yu Guo. (2010) "A Trusted Computing Environment Model in Cloud Architecture". Proceedings of the Ninth International Conference on Machine Learning and Cybernetics. 978-1-4244-6526-2. Qingdao, 11-14. Beijing Jiaotong Univ., Beijing, China.

Zhang, J.. et al; Xue Jing and Zhang Jian-jun. (2010). "A Brief Survey on the Security odel of Cloud Computing," 2010 Ninth International Symposium on Distributed Computing and Applications to Business, Engineering and Science (DCABES), Hong Kong IEEE, 475 – 478.

Zhidong, S.. et al; Zhidong Shen, Li Li, Fei Yan, and Xiaoping Wu. (2010) "Cloud Computing System Based on Trusted Computing Platform," Intelligent Computation Technology and Automation (ICICTA), IEEE International Conference on Volume: 1, 942-945. China.

Zhimin, Y.. et al; Zhimin Yang, Lixiang Qiao, Chang Liu, Chi Yang, and Guangming Wan. (2010). "A collaborative trust model of firewall-through based on Cloud Computing," Proceedings of the 14th International Conference on Computer Supported Cooperative Work in Design. Shanghai, China. 329-334, 14-16.

APÊNDICES

APÊNDICE A – CÓDIGO DA CLASSE TRUST

A classe Trust implementa o modelo de confiança proposto nesta tese. Esta classe foi totalmente codificada durante o desenvolvimento deste trabalho e será incorporada ao framework CloudSim.

```
package modelo;
import org.cloudbus.cloudsim.Cloudlet;
import org.cloudbus.cloudsim.Vm;
import controler.Cloud;
public class Trust {
	public static Vm bestIdVm(Cloudlet cloudlet) {
		// Método responsável por realizar no começo de cada interação a definição dos valores correntes de cada máquina virtual.
		join();
		// Para cada máquina virtual calcula-se a confiança da máquina conforme a equação definida, e posteriormente calcula-se a confiança em relação a máquina baseline.
		// Realiza o incremento e decremento do valor de confiança conforme o número de tarefas executadas com sucesso e sem sucesso
		for (VmDm vm : Cloud.vmList) {
		// O cálculo das capacidades de armazenamento e processamento são realizados a partir da razão entre a característica da máquina em comparação com a característica ideal definida na máquina baseline.
		double ca = vm.getCurrentHd() / Cloud.idealSize;
		double cp = ((vm.getCurrentRam()/Cloud.idealRam) + (vm.getCurrentTotalMips() / Cloud.idealMips)) / 2;
		double ce = vm.getCurrentBw() / Cloud.idealbw;
		double so = 1; // se for linux = 1 se for windows = 0,5
		// Garantir que a razão não ultrapasse o limiar = 1 que ocorre se a característica for superior ao definido com o ideal.
			if (ca > 1)ca = 1;
			if (cp > 1)cp = 1;
			if (ce > 1)ce = 1;
```

```
double confiancaVm = 0.0;

// Caso seja o primeiro valor de confiança da máquina virtual, executa-se diretamente o calculo da confiança.
if (vm.getHistoryTarefas().size() < 1) {
    confiancaVm = ((ca*0.35)+(cp*0.35)+(ce*0.15)+(so*0.15));
//Senão realiza o somatório dos históricos dos valores de confiança e soma mais o calculo da confiança atual e divide pelo número de interações já realizadas mais um.
}else {
double somatorioConfianca = ((ca*0.35)+(cp*0.35)+(ce*0.15)+(so*0.15));
// Percorre o histórico de valores de confiança da máquina
for (Integer integer : vm.getHistoryTarefas()) {
somatorioConfianca += vm.getDesenvolvimentoConfianca().get(integer);
        }
confiancaVm = somatorioConfianca / (vm.getHistoryTarefas().size() + 1);
    }
// A cada tarefa executada sem sucesso o valor de confiança é decrementado em 0.05.
for (int i = 0; i < vm.getCloudletsNaoExecultadas(); i++) {
confiancaVm -= 0.05;
}
// A cada tarefa executada com sucesso a máquina virtual tem adicionado ao seu valor de confiança 0.25. Se o valor de confiança estiver acima de 0.85 a cada tarefa executada o valor adicionado é de 0.05.
for (int i = 0; i < vm.getCloudletsExecultadas(); i++) {
    confiancaVm = confiancaVm + 0.025;
}
// Para garantir que o valor de confiança não ultrapasse o limiar de 1, caso a soma do valor de confiança da máquina virtual mais o valor de confiança acumulado seja maior que 1 reduzimos o valor da confiança para 1.
if (confiancaVm > 1) {
    confiancaVm = 1;
```

```
}
vm.getDesenvolvimentoConfianca().add(confiancaVm);
}
double melhorConfianca =
Cloud.vmList.get(0).getDesenvolvimentoConfianca().get(Cloud.execution);
int idBestVmDm = Cloud.vmList.get(0).getId();
// Identifica o id da máquina virtual com maior confiança e disponível
for (VmDm vmDm : Cloud.vmList) {
if (!(vmDm.getExecTime() > Cloud.execution) &&
vmDm.getDesenvolvimentoConfianca().get(Cloud.execution) > melhorConfianca) {
melhorConfianca =
vmDm.getDesenvolvimentoConfianca().get(Cloud.execution);
idBestVmDm = vmDm.getId();
}
}
// Condicional responsável por selecionar as tarefas que não serão
executadas com sucesso.
double rodon = Math.random();
// String responsável pelo status da execução da tarefa
String status;
// Se "randon" for maior que 0.8 a tarefa será executada sem sucesso
if (randon > 0.8) {
status = "semSucesso";
Cloud.vmList.get(idBestVmDm).setCloudletsNaoExecultadas();
}else {
status = "comSucesso";
Cloud.vmList.get(idBestVmDm).setCloudletsExecultadas();
// Envia a máquina virtual para ser calculado e aplicado o desgaste
que ocorrera com a execução da tarefa.
ConsumoRecurso.desgaste(cloudlet,
Cloud.vmList.get(idBestVmDm));
}
Cloud.vmList.get(idBestVmDm).getSituacoesExecucoes().add(status);
// Adiciona o valor de confiança ao histórico de cada máquina virtual.
```

```
                Cloud.vmList.get(idBestVmDm).getHistoryTarefas().add(Cloud.execution);
                // Atribui- se a variável bestVmDm a máquina definida com a melhor
máquina, ou seja, a que possui o maior valor de confiança.
        Cloud.relatorio.add(new DadosRelatorioTXT(cloudlet.getCloudletId(),
        Cloud.vmList.get(idBestVmDm).getId(),
        Cloud.vmList.get(idBestVmDm).getDesenvolvimentoConfianca().get(Cloud.execu
        tion), status));
                Cloud.execution++;
                // Retorna a máquina virtual livre com melhor valor de confiança no
momento para executar a cloudlet.
                return Cloud.vmList.get(idBestVmDm);
        }
                // Método responsável por realizar no começo de cada interação a definição
dos valores correntes de cada máquina virtual.
                private static void join() {
                for (VmDm vmDm : Cloud.vmList) {
                // Verifica se a máquina virtual está ociosa e se tiver seta o valor estático
como valor corrente
                if (vmDm.getExecTime() < Cloud.execution || vmDm.getExecTime() == 0)
{
                vmDm.setCurrentBw(vmDm.getBw());
                vmDm.setCurrentRam(vmDm.getRam());
                vmDm.setCurrentHd(vmDm.getSize());
                vmDm.setCurrentTotalMips(vmDm.getMips()*vmDm.getNumberOfPes());
                        }
                }
        }
}
```