

# Neblina - Uma aplicação web para nuvens computacionais

Felipe Juliani Fernandes, Bruno Schulze, Antônio Roberto Mury  
Laboratório Nacional de Computação Científica  
Grupo ComCiDis - Computação Científica Distribuída  
Av. Getúlio Vargas, 333 - Quitandinha  
Petrópolis, RJ  
{felipejf,schulze,aroberto}@lncc.br

## ABSTRACT

*This work proposes a web environment based on cloud computing paradigm. The created environment has the purpose to provide and manage a large amount of virtualized resources among its users, with specific characteristics, optimizing computing power, sharing files and information to meet the demand. Through customizable dedicated and virtualized platforms, the application provides workspaces remotely accessible, with secure connection, through the browser.*

## RESUMO

*O trabalho apresenta uma proposta de ambiente web baseado no paradigma da computação em nuvem. Este ambiente tem o propósito de gerenciar e disponibilizar uma grande quantidade de recursos virtualizados entre seus usuários, particularizando características, otimizando a capacidade computacional e compartilhando arquivos e informações para atender a demanda. Por meio de plataformas virtualizadas dedicadas e customizáveis, a aplicação fornece espaços de trabalhos remotamente acessíveis, com conexão segura, através de um browser.*

## Keywords

Computação em Nuvem, Virtualização, Espaços de Trabalho Virtualizados

## 1. INTRODUÇÃO

Com a evolução das tecnologias de processadores, incorporando tecnologias de virtualização [1] e o aumento na capacidade de comunicação de dados, o modelo de computação em nuvem [2] surgiu, fornecendo aos seus usuários soluções, que flexibilizam e facilitam o uso de recursos, outrora disponíveis somente em grandes centros de processamento. Esse conceito, até pouco tempo abstrato, dispõe hoje em dia de uma grande quantidade de ferramentas e serviços, disponibilizados para diversos tipos de usuários.

O presente projeto insere-se dentro de um conjunto de pes-

quisas relacionadas à utilização da infraestrutura de nuvens computacionais, em proveito de aplicações científicas em ambientes de alto desempenho. As nuvens computacionais contribuem como fonte de recursos adicionais ao agregar rapidamente novos recursos sob demanda (escalabilidade). Permitem que seus usuários sejam capazes de montar e alterar a sua infraestrutura em função do tipo de problema a ser tratado (flexibilidade), acessando aplicações remotamente, dissociando o espaço geográfico onde os recursos e pessoas que os utilizam se encontram (portabilidade), além do compartilhamento destes, contribuindo assim para otimização do uso dos mesmos.

## 2. OBJETIVO

O projeto Neblina surgiu no laboratório do ComCiDis/LNCC (Computação Científica Distribuída/Laboratório Nacional de Computação Científica) com o objetivo de flexibilizar o uso dos seus recursos atendendo a demanda de diferentes usuários, fornecendo plataformas dedicadas as suas necessidades específicas. Esses usuários passariam a ter seus próprios espaços de trabalho virtualizados, isolados [3], e com os recursos necessários para executar suas tarefas, utilizando a infraestrutura computacional do laboratório, por meio de um navegador *web*, ou diretamente por meio de um cliente de *desktop* remoto (Figura 1). Os administradores da nuvem também poderiam controlar todo o ambiente computacional através dessa aplicação, obtendo informações de estados, informações de recursos e gerenciando e disponibilizando as plataformas para cada um desses usuários.

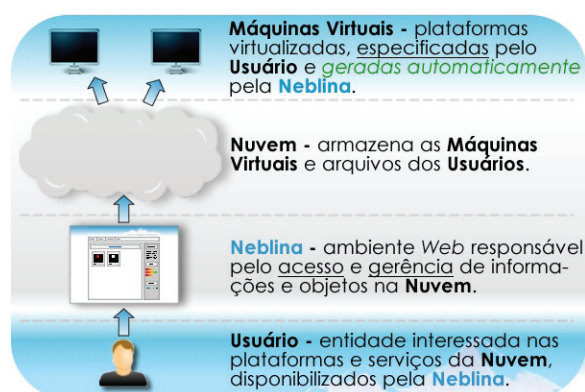


Figura 1: Ambiente Neblina

### 3. ARQUITETURA

O usuário acessa a interface da Neblina através de um *browser*, pela internet ou intranet, em seu computador. A interface visual da Neblina fornece informações de estado da nuvem, arquivos para *upload* ou *download* e espaços de trabalhos virtualizados dedicados e previamente criados, para uso. A Neblina interage diretamente com os *hosts* e seus componentes de virtualização, extraindo informações e executando comandos, a partir de requisições feitas por seus usuários. O repositório guarda os discos virtuais que são usados como modelos para a criação das estações de trabalho virtualizadas, de cada usuário. O servidor de arquivos guarda os arquivos enviados pelos usuários e compartilha-os com as estações de trabalho virtualizadas para utilização.

A solução proposta compreende tanto a infraestrutura computacional (*hardware*, gerenciador de máquinas virtuais, sistema operacional), quanto a aplicação em si. A seguir serão apresentadas os elementos da infraestrutura e arquitetura da aplicação *web*, com suas especificações e componentes.

#### 3.1 Infraestrutura

A infraestrutura básica da nuvem é composta por um servidor *web*, um repositório de discos virtuais, um servidor de arquivos e de *hosts* para o consumo de recursos. Todas as máquinas possuem: 2 processadores Xenon e5520 de 4 núcleos, 12 GB de RAM e 1 TB de disco rígido. O sistema operacional utilizado é o Ubuntu Linux 10.04 server. O servidor *web* é o Apache Tomcat 6.0.29, o SGDB MySQL Database 5.0 e a Java Virtual Machine 1.6.20. O gerenciador de máquinas virtuais é o VirtualBox 3.2.12. O Diagrama de Implantação UML [4] na Figura 2 apresenta a disposição desses recursos:

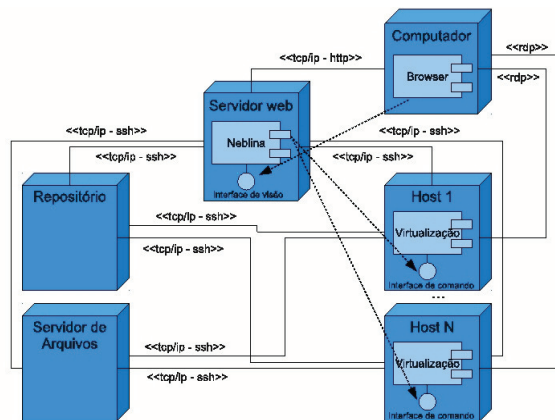


Figura 2: Diagrama UML de Implantação.

#### 3.2 Aplicação Web

A Neblina, foi construída baseando-se no padrão de projeto [5] *Model-View-Controller* (MVC), com o *framework* de desenvolvimento ágil Grails [6]. O Grails usa o Java e incorpora tecnologias *open source* como o *Hibernate*[6], o *Spring*[6] e o *SiteMesh*[6], este último implementa outro padrão de projeto chamado *Decorator*. A *Integrated Development Environment* (IDE) de desenvolvimento utilizada foi o *Springsource Tools Suite* (STS) na versão 2.3.2.

Na Figura 3 é apresentada a comunicação entre os componentes da aplicação *web* Neblina:

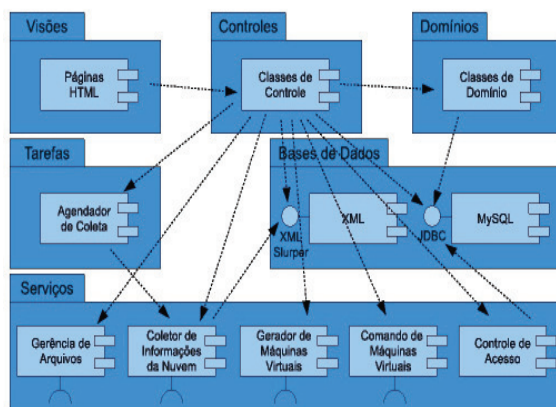


Figura 3: Diagrama UML de Componentes

##### 3.2.1 MVC e Persistência de Dados

Os três pacotes do topo representam o padrão MVC, por meio do qual o usuário interage com o sistema. Estes pacotes são responsáveis pelo controle, persistência de dados e apresentação das informações da nuvem, das máquinas virtuais disponíveis como espaços de trabalho e dos arquivos de trabalho. São esses componentes, também, que restringem o acesso aos módulos da aplicação, de acordo com o tipo de usuário, habilitando um acesso personalizado ao sistema e as plataformas virtualizadas.

O pacote "Bases de Dados" agrupa as bibliotecas do *Grails Object-Relational Mapping* (GORM)[6] / *Hibernate* que é a camada de persistência de dados vindos dos componentes que retratam o MVC. O componente "GORM" é responsável por popular o banco de dados com as informações parametrizadas, vindas tanto dos formulários *web* quanto dos serviços. Este pacote também possui um componente "XML" que armazena as informações da nuvem como, estado dos *hosts*, estado das máquinas virtuais, carga de processamento e memória, otimizando a consulta dessas informações.

##### 3.2.2 Tarefas e Serviços

O pacote "Tarefas" consiste de pequenos serviços executados automaticamente, destinados a verificação periódica do estado dos recursos e plataformas virtualizadas.

O pacote de "Serviços" é responsável pela execução de procedimentos como, a criação, armazenamento e execução das máquinas virtuais e arquivos, e também de funções como, coleta de dados referentes a capacidade real dos recursos e autenticação de usuários. Isto é feito com base nos parâmetros estipulados pelas informações inseridas na base de dados e oriundos da infraestrutura.

Alguns componentes do pacote, tem o papel de interagir com as interfaces de comando dos virtualizadores e do próprio sistema operacional dos *hosts*. Esta interação é feita através de um conjunto de protocolos padronizados, executados por comandos vindos dos componentes deste pacote, disponibilizando assim os recursos e informações para o usuário final.

No atual estágio de desenvolvimento, há um forte acoplamento entre a aplicação *web* e os componentes de comandos e coletas de informações da infraestrutura. Isto se dá, pois estes componentes se comportam como um elemento único. No futuro para que seja possível o uso da aplicação *web* dissociado da infraestrutura planeja-se implementar, dentro do pacote "Serviços", um conjunto de interfaces com protocolos compatíveis com as interfaces da infraestrutura desejada.

## 4. FUNCIONALIDADES

O projeto Neblina surgiu como possibilidade de flexibilizar a oferta de recursos computacionais, para usuários que necessitem de plataformas dedicadas específicas. Algumas funcionalidades entretanto, precisam ser ressaltadas, pois são pontos de destaque deste trabalho.

### 4.1 Gerência de Usuários

Atualmente a aplicação suporta dois tipos de usuários, administradores e utilizadores. Os administradores têm o papel de gerenciar o ambiente da nuvem, consultando informações, criando e disponibilizando estações de trabalho virtualizadas, gerenciando seu cadastro, *hosts* e plataformas, e quando necessário as máquinas virtuais. Os utilizadores podem manipular seus arquivos compartilhados, acessar e controlar suas estações de trabalho virtualizadas.

### 4.2 Gerência de Hosts

Os *hosts*, depois de manualmente configurado para fazer parte da nuvem, tem suas informações inseridas na base de dados da Neblina. Nesse momento, o administrador cadastra seu endereço na rede e seu papel na nuvem, dizendo se o *host* é um *host* propriamente dito, um repositório e/ou um servidor de arquivos. Feito isso, a aplicação inicia o seu monitoramento, disponibilizando-o como um recurso da nuvem para seus usuários.

### 4.3 Gerência de Plataformas

As plataforma são discos virtuais com sistemas operacionais e aplicativos instalados previamente, localizadas no repositório e chamadas também de *appliances*[1]. As referências de nome e caminho são inseridas na aplicação *web* para que, quando solicitada a criação de uma máquina virtual, uma clonagem desta *appliance* seja executada, sendo disponibilizada posteriormente para o usuário solicitante do recurso.

### 4.4 Gerência de Arquivos

A necessidade do compartilhamento de arquivos com a nuvem é outro ponto importante. Os usuários precisam migrá-los para a nuvem, para que possam utilizá-los em suas estações de trabalho virtualizadas. Então, foi criada uma representação de um sistema de arquivos básico, onde cada usuário possui seu diretório no ambiente para: navegação, *upload*, *download*, compactação e extração. Desta forma, pode-se fazer o *upload* de arquivos trabalhados em outras estações para a estação de trabalho virtualizada, podendo-se então manipulá-los e quando necessário, fazer o *download*.

### 4.5 Geração de Máquinas Virtuais

As máquinas virtuais são geradas pelos administradores e passadas para os seus utilizadores. Um utilizador especifica a plataforma da estação de trabalho que será virtualizada e

suas características (Modelo de Máquina Virtual) para um administrador. Insere na base de dados da aplicação o modelo de máquina virtual baseado nesta especificação. Desta forma, gera a estação de trabalho virtualizada e a disponibiliza para o usuário requisitante. Este conceito de modelo, tornou possível o reaproveitamento de especificações para gerar outras máquinas virtuais semelhantes.

## 4.6 Controle das Estações de Trabalho

O controle de uma estação de trabalho virtualizada pode ser feito através da própria aplicação *web*, que disponibiliza um *applet*, que implementa uma interface do *Remote Desktop Protocol* (RDP); ou através de qualquer aplicativo de conexão remota, que aceite o RDP. Nesta segunda forma, o usuário não depende da aplicação *web* para acessar a estação de trabalho virtualizada, possibilitando sua utilização na nuvem, mesmo que o site fique fora do ar.

## 5. CONCLUSÃO E TRABALHOS FUTUROS

O presente trabalho apresentou a aplicação *web* Neblina para ambientes de nuvens computacionais, sua arquitetura e funcionalidades. Acredita-se que com este ambiente implantado, os recursos de um centro de processamento possam ser tanto disponibilizados quanto manipulados de forma mais flexível e segura. A administração dos recursos é facilitada, visto que a Neblina possui um monitoramento gráfico, em tempo real, do estado dos *hosts* e das máquinas virtuais na nuvem. No aspecto de atendimento de demanda, usuários com necessidades específicas podem fazer uso dos recursos, sem a necessidade de conhecimentos específicos para a configuração e gerenciamento do seu espaço de trabalho.

O trabalho apresentado encontra-se em uso por projetos pertencentes ao LNCC (aplicações médicas, processamento de imagens e testes de aplicações distribuídas) e insere-se dentro da proposta de plano de trabalho de Iniciação Científica relacionado ao uso de nuvens computacionais.

Como trabalhos futuros, pretende-se incorporar mecanismos de balanceamento de carga, agendamento, importação e exportação de *appliances*, além do gerenciamento de *clusters* virtuais, controle de perfis de usuário e mecanismos que implementem o consumo racional de energia.

## 6. REFERÊNCIAS

- [1] Danielle Ruest and Nelson Ruest. *Virtualization: A Beginner's Guide*. The Mc Graw Hill Company, 2009.
- [2] Borko Furht and Armando Escalante, editors. *Handbook of Cloud Computing*. Springer, 2010.
- [3] Ian Foster, Yong Zhao, Ioan Raicu, and Shiyong Lu. Cloud computing and grid computing 360-degree compared. *Grid Computing Environments Workshop - GCE, 2008*, 2008.
- [4] Martin Fowler. *UML distilled: A brief guide to the standard object modeling language*. Pearson Education, Inc, 2004.
- [5] Erick Freeman and Elisabeth Freeman. *Use a Cabeça! Padrões de Projetos*. Alta Books, Segunda Edição Revisada, 2008.
- [6] Bashar Abdul-Jawad. *Groovy and Grails Recipes*. Apress; 1st edition, 2009.