Quinta Conferencia de Directores de Tecnología de Información, TICAL 2015 Gestión de las TICs para la Investigación y la Colaboración, Viña del Mar, del 6 al 8 de junio de 2015

GT-PID: Uma Nuvem IaaS Universitária Geograficamente Distribuída

Rodrigo de Souza Couto [a], Tatiana Sciammarella [b], Hugo de Freitas Siqueira Sadok Menna Barreto [b], Miguel Elias Mitre Campista [b], Marcelo Gonçalves Rubinstein [a], Luís Henrique Maciel Kosmalski Costa [b], Fausto Vetter [c], André Marins [c]

[a] Universidade do Estado do Rio de Janeiro
FEN/DETEL
rodrigo.couto@uerj.br, rubi@uerj.br

[b] Universidade Federal do Rio de Janeiro,
PEE/COPPE/GTA – DEL/POLI
Rio de Janeiro, Brasil
tatiana@gta.ufrj.br, sadok@gta.ufrj.br, miguel@gta.ufrj.br, luish@gta.ufrj.br

[c] Rede Nacional de Ensino e Pesquisa
Diretoria de Pesquisa e Desenvolvimento
fausto.vetter@rnp.br, amarins@rnp.br

Resumo. O objetivo do projeto GT-PID é promover o compartilhamento de recursos computacionais entre centros de pesquisa a partir da disponibilização na nuvem de todos os recursos existentes. Para isso, utiliza-se o conceito de máquinas virtuais que são oferecidas a cada solicitação de utilização da infraestrutura. Assim, o usuário possui acesso em nível de administrador a essas máquinas virtuais e pode instalar suas aplicações. O usuário, então, possui total flexibilidade na escolha dessas aplicações, caracterizando um serviço em nuvem IaaS (*Infrastructure as a Service*). Por exemplo, o usuário pode utilizar as máquinas virtuais para realizar suas simulações, instalando ferramentas específicas para sua pesquisa. A arquitetura básica adotada pelo GT-PID baseia-se na plataforma OpenStack e oferece uma interface web aos usuários, permitindo o gerenciamento do ciclo de vida de suas máquinas virtuais; por exemplo, a criação e a destruição de VMs. Além disso, a interface web permite ao usuário a visualização do console das VMs. Para atender aos objetivos do projeto, realizaram-se modificações no OpenStack para o cenário geodistribuído. Foram modificados os processos de escalonamento e a hierarquia de usuários do OpenStack para permitir que as instituições participantes possam gerenciar seus próprios sítios. Além disso, a interface web de gerenciamento do OpenStack foi modificada para refletir tais mudanças.

Palavras Chave: computação em nuvem; IaaS; OpenStack.

1 Introdução

Há um consenso entre pesquisadores trabalhando em instituições de pesquisa de que muitas vezes os recursos computacionais de um determinado laboratório permanecem ociosos por longos períodos, enquanto em outros momentos esses recursos não são suficientes. Por exemplo, na proximidade do prazo final de submissão de uma conferência, muitos usuários do mesmo laboratório tendem a compartilhar os recursos computacionais disponíveis, muitas vezes de forma pouco coordenada. Tendo isso em vista, a utilização do poder de processamento das

máquinas ocorre, de alguma forma, em rajadas, o que dificulta o planejamento prévio dos recursos, que pode ser feito tanto pelo pico quanto pela média em um determinado período. Partindo dessa premissa, é de extrema utilidade uma plataforma que dê suporte à utilização de recursos computacionais de forma compartilhada. Dessa forma, o planejamento leva a uma utilização mais eficiente, já que evita tanto períodos longos de ociosidade quanto períodos de saturação. A ideia do projeto GT-PID é promover o compartilhamento de recursos computacionais entre centros de pesquisa a partir do compartilhamento em nuvem de todos os recursos existentes. Assim, durante períodos de ociosidade, os recursos estariam disponibilizados para outros laboratórios, enquanto que, em períodos de necessidade crítica, eles poderiam ser usados de maneira distribuída pelos diferentes usuários dos laboratórios participantes [1]. Por um lado, provê-se uma capacidade global do sistema superior à oferecida localmente. Por outro lado, diminui-se a ociosidade dos recursos computacionais, aumentando a eficiência e o retorno do investimento financeiro aportado na pesquisa.

A infraestrutura computacional do GT-PID é organizada em torno de uma nuvem distribuída, que interliga todas as universidades brasileiras e estrangeiras participantes. Dessa forma, o GT-PID visa construir uma plataforma computacional distribuída, agregando capacidade de processamento e de armazenamento. Este aspecto é importante, pois frequentemente em atividades de pesquisa os processos executados requerem alto poder de processamento e de armazenamento [2], tome-se por exemplo a simulação de protocolos de redes sem-fio ou a análise de dados experimentais em física. No Brasil, as universidades e demais instituições de pesquisa poderão utilizar a infraestrutura existente da rede Ipê (http://www.rnp.br/ipe/) para se interligarem, aproveitando um serviço já prestado pela RNP. Já a infraestrutura física da plataforma computacional distribuída é formada pelos recursos computacionais cedidos por cada laboratório participante. Sendo assim, um determinado laboratório disponibiliza uma certa quantidade de recursos computacionais e, em troca, pode utilizar os serviços da plataforma distribuída.

Um dos principais papéis da plataforma do GT-PID é permitir a execução de simulações e serviços de pesquisadores dos laboratórios envolvidos. Para tanto, a infraestrutura provida pode ser utilizada para experimentos e simulações das mais diversas áreas de pesquisa e não somente da área de computação. Para isso, é utilizado o conceito de máquinas virtuais (VMs – *Virtual Machines*), que são oferecidas a cada solicitação de utilização da infraestrutura [3, 4]. As VMs são disponibilizadas em um conjunto responsável por executar a simulação ou serviço desejado. Assim, o pesquisador possui acesso em nível de administrador a essas VMs e pode instalar as ferramentas necessárias sem a interferência de um administrador da infraestrutura física. O pesquisador, então, possuirá total flexibilidade na escolha de suas ferramentas. Ao término da utilização da plataforma pelo pesquisador, as máquinas virtuais criadas podem ser simplesmente removidas da plataforma [5].

O piloto IaaS (*Infrastructure as a Service*) construído no projeto permite que usuários criem e utilizem VMs, que são hospedadas nos Servidores de VMs espalhados pelas universidades. Os Servidores de VMs consistem em PCs comuns executando uma plataforma de virtualização (hipervisor) que, no contexto do GT-PID, é o KVM. Para a gerência da infraestrutura há dois tipos de administradores no GT-PID: global e local. O administrador global é responsável pela gerência do piloto

IaaS como um todo, enquanto os administradores locais são responsáveis por operações de gerenciamento dentro de um único sítio. Um sítio é definido no GT-PID como o conjunto de Servidores de VMs e de Discos disponibilizado por uma universidade ou laboratório de pesquisa, ou seja, uma unidade administrativa autônoma. As operações de gerenciamento incluem a migração local e migração global, detalhadas mais adiante neste trabalho.

Este trabalho está organizado da seguinte forma. A Seção 2 detalha a arquitetura adotada pelo piloto do GT-PID. A Seção 3 detalha o OpenStack, que é o orquestrador de nuvem utilizado como base pelo piloto, e mostra as modificações realizadas no contexto do GT-PID. A Seção 4 detalha as modificações realizadas no OpenStack e mostra as decisões de projeto, enquanto a Seção 5 conclui o trabalho e aponta direções futuras.

2 Arquitetura

Esta seção descreve a arquitetura do piloto, bem como suas principais funções.

2.1 Visão Geral

A arquitetura básica adotada pelo GT-PID é apresentada na Fig. 1. O Controlador é responsável pelo gerenciamento dos sítios, sendo capaz, por exemplo, de alocar recursos destinados às VMs nos diversos Servidores de VMs. Os Servidores de VMs de cada sítio se comunicam com o Controlador através de túneis VPN (*Virtual Private Network*) estabelecidos pela Internet. Os usuários, por sua vez, acessam o Controlador através de uma interface web que permite o gerenciamento do ciclo de vida de suas máquinas virtuais como, por exemplo, a criação e a destruição de VMs. Além disso, a interface web permite ao usuário a visualização do console das VMs.

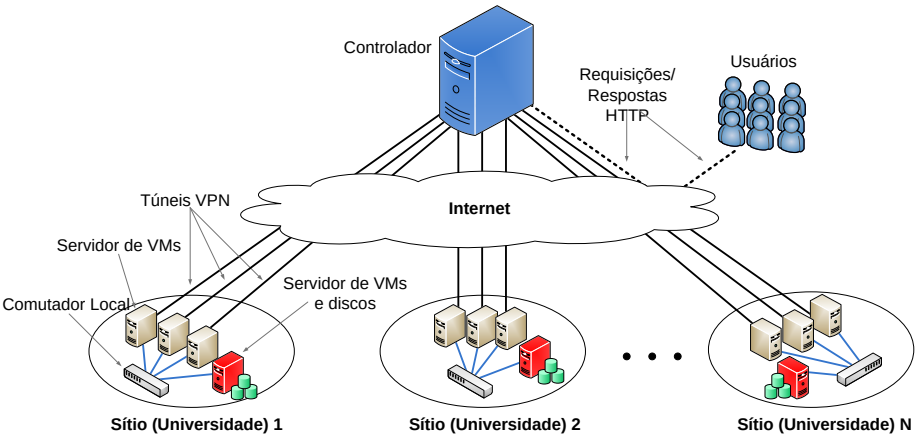

Fig. 1. Arquitetura básica do piloto IaaS do GT-PID.

Além dos Servidores de VMs, cada sítio possui um Servidor de VMs e Discos, que é responsável por hospedar os discos virtuais das VMs. A existência dessa máquina é importante para permitir migração ao vivo no sítio, ou seja, as VMs hospedadas em um Servidor de VMs podem ser transferidas em tempo de execução para outro Servidor de VMs do sítio, sem que haja suspensão de seus serviços. Como os discos virtuais estão localizados no Servidor de VM e Discos, não é necessário realizar cópia de discos entre os servidores envolvidos. Vale notar que esse servidor também pode hospedar VMs. Por fim, cada sítio possui um Comutador local interligando todas as suas máquinas, possibilitando as comunicações de VMs hospedadas em diferentes servidores e também operações de disco através de NFS (*Network File System*).

2.2 Características das VMs

As VMs disponibilizadas para o usuário podem executar diversos sistemas operacionais. O Controlador fornece um conjunto de imagens com sistemas operacionais pré-instalados, e o usuário escolhe qual dessas imagens utilizará em sua VM. Além disso, o usuário pode fornecer ao Controlador sua própria imagem de VM, o que possibilita uma maior flexibilidade na escolha do sistema operacional. Uma VM básica possui um endereço IP privado (ou seja, não acessível pela Internet) e acessa a Internet através de NAT (*Network Address Translation*). Ao criar uma nova VM, o usuário pode escolher duas formas possíveis de instanciação: baseada em *imagem* e baseada em *volume*. A primeira consiste na cópia de uma imagem fornecida pelo Controlador, que é destruída após o desligamento da VM. A instanciação baseada em volumes, por sua vez, copia a imagem escolhida para um volume, que é uma unidade lógica de armazenamento persistente. Assim, o usuário continua com seus dados gravados mesmo após o desligamento das VMs. A quantidade de volumes disponíveis para cada usuário é limitada para que usuários diferentes tenham oportunidade de armazenar seus respectivos volumes.

2.3 Funções principais do piloto

A utilização do piloto desenvolvido no GT-PID ocorre através de uma interface web. Essa interface consiste em uma versão modificada do módulo Horizon do OpenStack, detalhado adiante. A interface é acessível a partir de qualquer PC conectado à Internet através da utilização da URL referente ao Controlador do piloto. Ao entrar na URL, o usuário encontra a tela de autenticação, como mostra a Fig. 2. Um exemplo da interface web pode ser visto na Fig. 3.

Fig. 2. Tela de login.

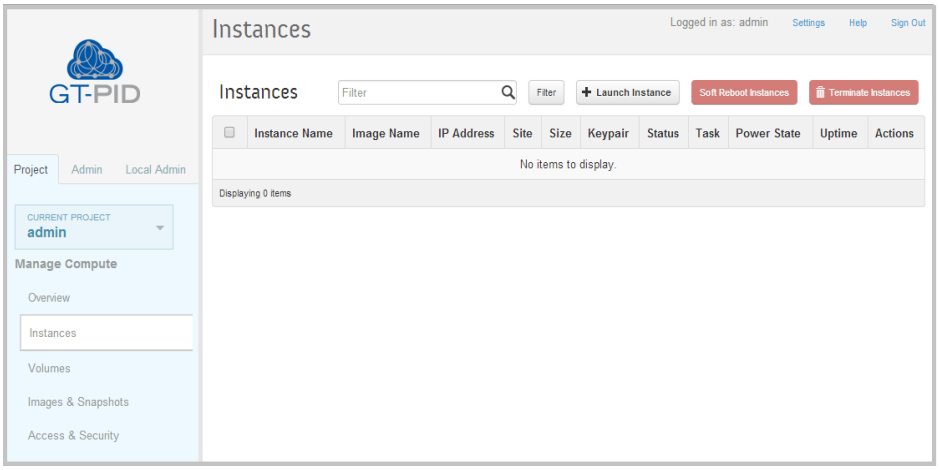

Fig. 3. Exemplo da tela da interface web. Neste exemplo, é mostrada a tela de Administrador Global, que possui todas as funções possíveis.

Além do usuário final, no piloto do GT-PID existem dois tipos de administrador: o local e o global, como discutido anteriormente. Logo, a interface do piloto possui funções específicas para cada tipo de usuário. As funções principais oferecidas pelo piloto podem ser divididas em Autenticação, Funções de Usuário, Funções de Administrador Local e Funções de Administrador Global, detalhadas a seguir. Vale ressaltar que todas as funções abaixo descritas podem ser acessadas via interface web.

Autenticação - Para ter acesso a qualquer serviço, usuários e administradores (locais e globais) devem se autenticar na interface web que, por sua vez, se comunica com o módulo de autenticação. No desenvolvimento do piloto, integra-se esse módulo com uma plataforma brasileira de identidade federada, denominada CAFe [6]. A CAFe é uma federação de identidade que reúne diversas instituições de ensino e pesquisa brasileiras. Cada instituição mantém uma base local de seus usuários e estabelece uma relação de confiança com as outras instituições participantes. Assim, um determinado usuário pode acessar os serviços providos por todas as instituições da federação utilizando apenas suas credenciais fornecidas por sua instituição de origem. Com a integração com a CAFe, a plataforma do GT-PID poderá ser utilizada por qualquer usuário que possua alguma credencial da federação.

Funções de Usuário - A piloto do GT-PID permite as seguintes funções aos usuários:

- Criação de VMs: Nesse serviço VMs são criadas através da escolha de diversas configurações como, por exemplo, total de memória RAM e número de CPUs virtuais alocadas para a VM. Além disso, o usuário poderá escolher em qual sítio a VM será alocada ou deixar essa decisão para o Controlador;

- Gerenciamento de VMs: Nessa função o usuário pode desligar, ligar, reiniciar e pausar suas VMs. Além disso, pode atribuir IPs públicos para as VMs;

- Acesso ao console: Pela interface web o usuário é capaz de acessar o console de cada uma de suas VMs que, dependendo da imagem escolhida, consiste em um terminal gráfico ou uma interface de linha de comando;

- Upload de imagens: Usuários podem carregar novas imagens de VMs no Controlador para uso pessoal ou para disponibilizá-las a outros usuários.

Funções de Administrador Local - Atualmente a única função de Administrador Local disponível é a migração local de VMs. Nessa função, o Administrador Local pode migrar VMs entre dois Servidores de VMs de seu sítio, através de solicitações ao Controlador. Essa migração é ao vivo, o que significa que a VM não fica indisponível durante a migração. A migração é útil quando um Administrador Local deseja desligar um dos servidores do seu sítio, por exemplo, por questões de manutenção. Dessa forma, o administrador deve migrar todas VMs operacionais naquele servidor para outra máquina do sítio. Outras funções podem ser adicionadas ao piloto utilizando a hierarquia de administradores criada pelo GT-PID, como detalhado mais adiante.

Funções de Administrador Global - Além de todas as funções do Administrador Local, o Administrador Global pode executar as seguintes funções:

- Criação de usuários: Criação de usuários, grupos de usuários ou administradores locais que utilizam a infraestrutura;
- Definição de limites: Define limites máximos de utilização de recursos para usuários ou grupos, como memória RAM e núcleo de CPUs;
- Migração global de VMs: Consiste em migrar VMs entre dois Servidores de VMs localizados em sítios diferentes. Para tal, as VMs são pausadas.

3. OpenStack

O OpenStack é um orquestrador de nuvem de código aberto, utilizado para gerenciar recursos em um centro de dados. Em linhas gerais, o OpenStack gerencia três tipos de recursos: computação (p.ex. processamento e memória), rede e armazenamento. Para tal, fornece um conjunto de serviços e APIs que permitem a manipulação da nuvem. Além disso, o OpenStack fornece uma interface gráfica para acessar essas APIs (*Application Programming Interfaces*), disponibilizada na forma de interface Web. Os serviços do OpenStack, descritos adiante, são organizados de forma modular, permitindo fácil modificação da plataforma de acordo com as necessidades do GT-PID. Além do OpenStack, existem diversas plataformas de gerenciamento de nuvem em código aberto como, por exemplo, CloudStack [7] e Eucalyptus [8]. Escolheu-se o Openstack pois, além de atender a todos os requisitos do projeto, essa plataforma possui uma comunidade de desenvolvedores em constante crescimento, o que tende a tornar sua documentação mais completa e possibilita maior facilidade na resolução de problemas. Um trabalho em andamento no contexto do GT-PID verifica a adequação do CloudStack aos requisitos do projeto. Essa análise está sendo realizada devido à crescente importância do CloudStack no mercado de nuvem.

3.1 Serviços do OpenStack utilizados

O OpenStack é dividido em diferentes serviços, denominados *projetos* na sua terminologia. Os projetos OpenStack utilizados no GT-PID são descritos a seguir:

Horizon - Fornece a interface web do OpenStack, se comunicando através de APIs com todos os outros serviços utilizados;

Nova - Gerencia o ciclo de vida completo das VMs. Em seu ciclo de vida, uma VM pode estar no estado *definido* (a máquina está configurada, mas não está em execução e não possui memória não-volátil associada, não consumindo recursos da infraestrutura), *ativo* (a máquina está executando na infraestrutura), *pausado* (a máquina não está executando, porém os recursos da infraestrutura continuam alocados

para ela) e *suspenso* (a máquina não está executando, pode-se liberar o uso dos recursos, à exceção de recursos de armazenamento necessários para seu disco virtual). Assim, o serviço Nova é responsável por realizar a transição entre os diferentes estados. Além disso, esse serviço escalona as máquinas virtuais na infraestrutura, escolhendo qual Servidor de VMs irá hospedar cada uma. Outra função do serviço Nova é realizar a migração de VMs;

Cinder - Fornece armazenamento persistente às máquinas virtuais. Esse armazenamento persistente, chamado de volume na terminologia do OpenStack, pode ser visto como um disco rígido virtual;

Keystone - Realiza o gerenciamento de identidades, sendo responsável, por exemplo, pela autenticação de usuários da infraestrutura. No GT-PID integra-se esse serviço à federação CAFe da RNP;

Glance - Fornece imagens de máquinas virtuais. A imagem de uma máquina virtual, da mesma forma que um volume, pode ser vista como um disco rígido virtual. Entretanto, a instanciação de máquinas virtuais através de imagem não permite armazenamento persistente. Ou seja, todo conteúdo gravado no disco da máquina virtual será apagado após seu desligamento. Assim, caso o usuário necessite de persistência, o OpenStack copia a imagem para um volume e inicia a máquina virtual a partir desse volume, que é oferecido pelo serviço Cinder. Desta forma as imagens são úteis para fornecer aos usuários sistemas pré-instalados. Além disso, o próprio usuário pode fornecer novas imagens para o repositório gerenciado pelo Glance;

Ceilometer - Serviço de medição e monitoramento dos componentes da nuvem. Esse serviço se comunica com os demais serviços do OpenStack para obter as medidas desejadas como, por exemplo, o consumo de processamento por um usuário. Além disso, ele fornece alarmes que são enviados após a utilização de um determinado recurso atingir algum limiar. A partir do alarme ações podem ser tomadas como, por exemplo, migrar as máquinas do usuário para servidores com maior capacidade de processamento.

O OpenStack fornece funcionalidades de rede através do serviço denominado Neutron. Entretanto, o piloto atual do GT-PID utiliza as funcionalidades de redes mais simples fornecidas pelo serviço Nova. Futuras etapas do GT-PID planejam utilizar o Neutron para, por exemplo, possibilitar o estabelecimento de túneis seguros entre VMs de diferentes sítios.

3.2 Localização dos Serviços do OpenStack na Arquitetura do GT-PID

As Fig. 4, Fig. 5 e Fig. 6 mostram quais serviços cada máquina da arquitetura da GT-PID emprega. Note que alguns dos serviços listados na Seção 3.1 são divididos

em vários módulos, representados por elipses nas figuras. Os serviços que não possuem divisão em módulos são representados nas figuras por retângulos.

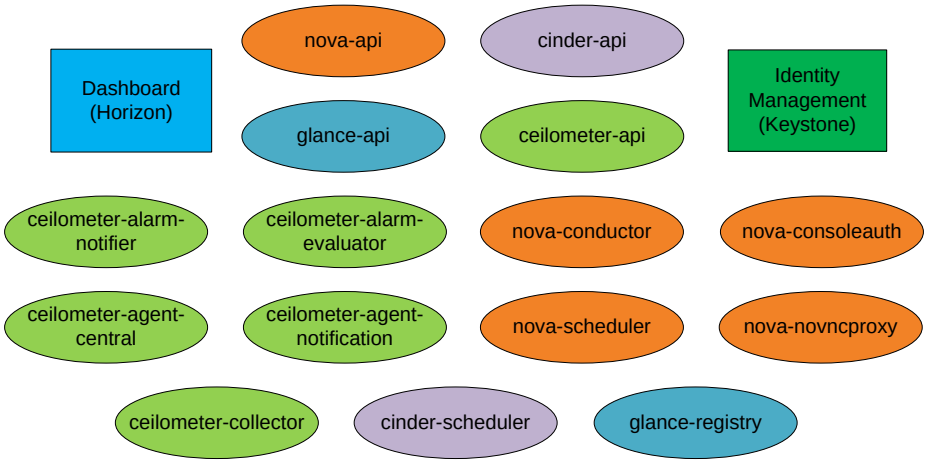

Fig. 4. Módulos utilizados no Controlador.

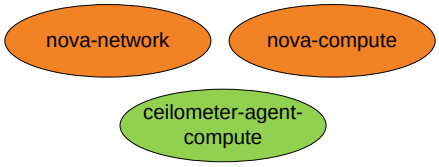

Fig. 5. Módulos utilizados no Servidor de VMs.

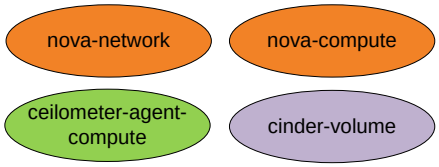

Fig. 6. Módulos utilizados no Servidor de VMs e discos.

Conforme mostrado na Fig. 4, o nó Controlador é responsável por hospedar todos os módulos de APIs e os módulos escalonadores (*schedulers*). As APIs são utilizadas para acessar os serviços de cada projeto, enquanto os escalonadores escolhem qual Servidor irá ser responsável por atender uma requisição. Por exemplo, para criar um volume, realiza-se uma requisição ao módulo cinder-api. Esse módulo, por sua vez, consulta o cinder-scheduler para indicar qual Servidor de VMs e de

Discos será utilizado no tratamento da requisição. Após a escolha do Servidor de VMs e de Discos, o Controlador notifica o módulo cinder-volume (Fig. 6) da máquina escolhida. Um processo semelhante ocorre nas interações entre os módulos nova-api, nova-scheduler e nova-compute (Fig. 5 e Fig. 6). A Fig. 7 mostra a comunicação entre todos os módulos utilizados. Os módulos do Nova, assim como outros projetos do OpenStack, se comunicam entre si através de uma fila fornecida pelo programa RabbitMQ, que é uma implementação do protocolo aberto AMQP (*Advanced Message Queueing Protocol*) [9]. Em linhas gerais, nesse protocolo a fila possui diversos canais de comunicação e cada módulo indica de quais canais quer receber mensagens. Ao enviar uma determinada mensagem, um módulo a coloca em um canal específico e, assim, todos os módulos que já se inscreveram para aquele canal recebem a mensagem. Por exemplo, um canal pode ser dedicado a eventos de criação de máquina virtual, e todos os módulos envolvidos nesse serviço devem escutar esse canal. O sistema de filas facilita a implementação dos serviços, pois um módulo não necessita delegar tarefas aos outros módulos, visto que cada um está ciente de seu papel no sistema e se inscreve nos canais correspondentes. O Hipervisor, por sua vez, é responsável por fornecer a abstração de hardware virtual às VMs, atuando como uma camada entre o hardware físico e as VMs. Por fim, o banco de dados do Nova, implementado em MySQL, permite o armazenamento de informações do estado do sistema, como instâncias de VMs criadas, tipos de instâncias suportados, etc.

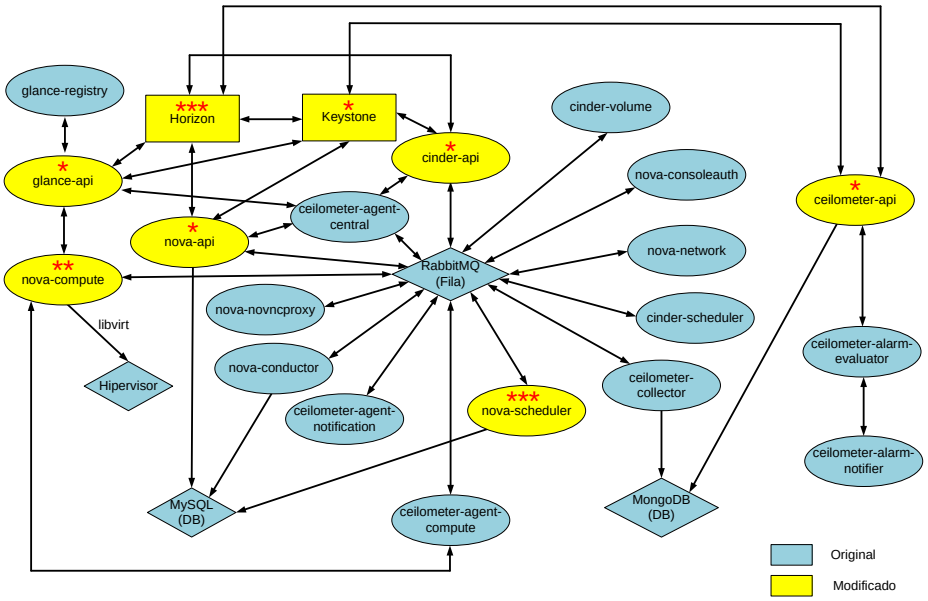

Fig. 7. Módulos do OpenStack utilizados e modificados.

Como mostra a Fig. 7, os módulos do OpenStack foram modificados para o contexto do GT-PID ou empregados na sua forma original. Essas modificações são detalhadas na próxima seção. As descrições dos módulos utilizados são apresentadas a seguir:

nova-api - é a porta de entrada do Nova para as requisições realizadas por usuários e administrador. Para tal, fornece APIs para todos os serviços do Nova. Note que as ações requisitadas pelos usuários através da interface web (Horizon) são encaminhadas para o nova-api de forma a serem atendidas;

nova-compute - se comunica com as APIs do Hipervisor para executar ações como criação e término de VMs. No GT-PID, as APIs do Hipervisor são fornecidas através da Libvirt [10];

nova-novncproxy - fornece um proxy para acessar as VMs por um console VNC (*Virtual Network Computing*) [11], que é um protocolo utilizado para manipular interfaces gráficas (p.ex. o Gnome no Linux ou até uma linha de comando) de forma remota;

nova-consoleauth - autentica os usuários ao nova-novncproxy, fornecendo *tokens* para acesso ao proxy;

nova-conductor - é um mediador das interações do nova-compute com a base de dados MySQL, possibilitando o isolamento entre esses dois componentes;

nova-scheduler - determina em qual Servidor de VMs uma determinada instância de VM será executada. Foi modificado pelo GT-PID para alocar as VMs de acordo com o ambiente geodistribuído fornecido;

nova-network - realiza tarefas de manipulação de rede, configurando interfaces de rede virtuais nas VMs e implementando regras de firewall no *iptables*;

cinder-api - API que recebe requisições para o Cinder e as encaminha para o módulo cinder-volume em um determinado servidor de VMs e de Discos;

cinder-volume - interage com os discos lógicos instalados no seu Servidor de VMs e de Discos correspondente;

cinder-scheduler - determina em qual Servidor de VMs e de Discos o volume da VM será instanciado. No piloto do GT-PID esse módulo sempre escolhe o Servidor de VMs e de Discos que se encontra no mesmo sítio escolhido pelo nova-scheduler;

glance-api - API que recebe requisições para o Glance;

glance-registry - Armazena e processa metadados da imagens, além de responder requisições de pedido desses metadados;

ceilometer-agent-compute - Solicita ao nova-compute estatísticas de utilização de recursos;

ceilometer-agent-central - Solicita estatísticas de utilização de recursos aos módulos presentes no Controlador, não relacionados a instâncias específicas e aos módulos nova-compute;

ceilometer-agent-notification - Inicia ações de alarmes;

ceilometer-collector - Monitora e envia para o banco de dados as mensagens de notificação e estatísticas vindas dos agentes;

ceilometer-alarm-evaluator - Controla o disparo dos alarmes, medindo estatísticas e comparando-as com os limiares estabelecidos;

ceilometer-alarm-notifier - Permite a configuração de alarmes baseados em limiares;

ceilometer-api - Recebe solicitações de aplicações externas e as responde a partir de informações requisitadas ao banco de dados.

4. Modificações do OpenStack e Decisões de Projeto no GT-PID

Esta seção detalha os conceitos utilizados do OpenStack e as decisões tomadas para adequar esse orquestrador ao contexto do GT-PID.

4.1 Zona de Disponibilidade

Para lidar com regiões geograficamente distintas, o GT-PID utiliza o conceito de Zonas de Disponibilidade implementado pelo OpenStack. As Zonas de Disponibilidade permitem organizar os nós físicos da nuvem em grupos lógicos. Com essa separação, um usuário, ao criar sua máquina virtual, pode escolher em qual zona esta será criada. Além disso, é possível desenvolver recursos que tirem proveito dessas zonas provendo maior confiabilidade, ao separar serviços em zonas distintas, ou menor latência entre as máquinas de um usuário, agrupando todas as suas VMs em uma mesma zona. No caso do GT-PID, cada universidade ou centro de pesquisa participante é uma zona de disponibilidade, chamada de sítio no contexto do projeto. Dessa forma, cada usuário pode escolher em qual universidade sua VM irá executar. Além disso, o usuário pode deixar a decisão de escolha para o próprio controlador.

Um exemplo corresponde a alocar as VMs de forma a distribuí-las geograficamente pela infraestrutura.

4.2 Módulo nova-scheduler

Como apresentado anteriormente, o OpenStack possui o módulo nova-scheduler, que decide onde criar uma VM, a partir de requisições dos usuários. Antes de criar uma máquina virtual, o nova-scheduler cria uma lista com todos os Servidores de VMs disponíveis. Essa lista passa então por duas etapas: Filtro e Peso. Na etapa de Filtro, diversas máquinas são descartadas a partir de parâmetros definidos pelo usuário na criação, como por exemplo, a Zona de Disponibilidade. Além disso, o filtro descarta os servidores que não possuem capacidade para hospedar uma determinada VM. A etapa de Peso, que ocorre logo após o Filtro, ordena as máquinas físicas de acordo com a quantidade de recursos disponíveis e as demais preferências definidas pelo administrador. Por exemplo, o nova-scheduler pode dar preferência em alocar as VMs em servidores com mais recursos disponíveis. A Fig. 8 mostra um exemplo de funcionamento do nova-scheduler. Suponha que o usuário solicitou a criação de uma VM na UERJ. Primeiramente, o nova-scheduler lista todos os Servidores de VMs junto com as suas respectivas informações de utilização de recursos como, por exemplo, a quantidade de memória RAM disponível. Nesse exemplo, os filtros são a Zona de Disponibilidade e a capacidade de hospedar uma VM. Assim, esse módulo passa a lista pelos filtros que, por exemplo, já descartam todas as máquinas que não são da UERJ. Além disso, é descartada a máquina UERJ_A, que não possui recursos disponíveis. A seguir, a lista passa por uma etapa de pesos que classifica as máquinas de acordo com uma política pré-definida. No exemplo, a preferência é dada para o Servidor de VMs com menor utilização e, assim, o servidor UERJ_B se encontra em primeiro lugar na lista e será então escolhido para instanciar a VM.

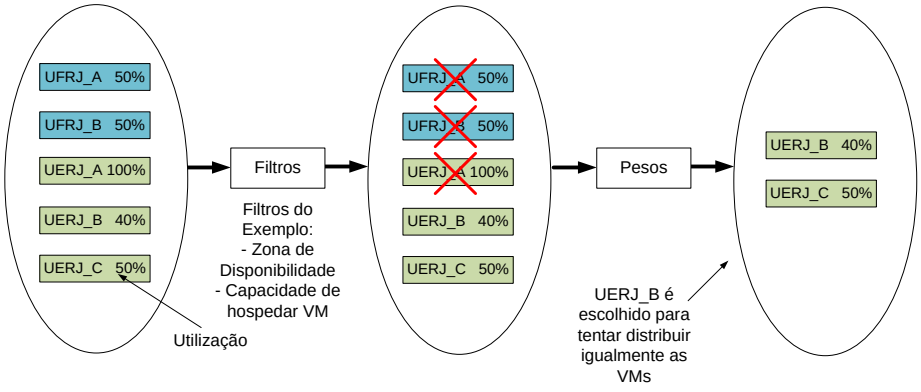

Fig. 8. Funcionamento do nova-scheduler.

Além dos filtros padrão disponíveis no OpenStack, como os que selecionam Servidores de VMs em uma Zona de Disponibilidade, é possível criar filtros personalizados, que possam atender mais especificamente aos objetivos do GT-PID. Da mesma forma, novos mecanismos de pesos podem ser definidos.

4.3 Escalonador de sítios

Como detalhado anteriormente, o OpenStack possui o escalonador nova-scheduler, que escolhe quais Servidores de VMs serão utilizados para hospedar as VMs de uma requisição. Para essa decisão, o OpenStack utiliza as informações de recursos disponíveis nos Servidores de VMs. Apesar de realizar o escalonamento no nível de Servidores de VMs, o OpenStack padrão não possui um mecanismo apropriado para a escolha de Zonas de Disponibilidade. As Zonas de Disponibilidade são escolhidas aletoriamente, sem verificação dos recursos disponíveis, ou são definidas explicitamente pelos usuários. Como o GT-PID utiliza o conceito de Zona de Disponibilidade para a definição dos sítios, foram realizadas modificações no nova-scheduler para tornar o escalonador mais apropriado aos requisitos do piloto. As contribuições do GT-PID nisso foram modificações realizadas no nova-scheduler para permitir que o Controlador decida, com base nas informações de recursos disponíveis nos sítios, em qual sítio uma requisição será atendida, se o usuário não especificar explicitamente qual sítio será utilizado. Além disso, é possível distribuir as VMs em diversos sítios de forma a aumentar a resiliência de um determinado serviço, eliminando pontos únicos de falha. No caso de distribuir as VMs entre os sítios, aplica-se um esquema de alocação *round-robin* com todos os sítios capazes de suportar uma determinada requisição de VMs. Já na criação de VMs de forma centralizada (isto é, com todas as VMs em um único sítio), os filtros verificam quais sítios são capazes de atender todas as requisições e ordenam os sítios pelos pesos. Assim, o sítio com o maior peso será escolhido para hospedar todas as VMs. O Escalonador de Sítios atua em conjunto com os filtros já existentes no OpenStack, que selecionam quais Servidores de VMs serão utilizados em cada sítio escolhido.

4.4 Controle de Acesso

No GT-PID adicionou-se outro tipo de usuário à plataforma OpenStack, denominado Administrador Local. Esse usuário é responsável pela administração de apenas um sítio, podendo realizar ações de migração das máquinas no sítio sob seu controle. Dessa forma, o Administrador Local encontra-se em um nível hierárquico intermediário entre o Administrador Global (isto é, o administrador da nuvem já existente no OpenStack, chamado de Admin) e o Usuário Final. A Fig. 9 mostra a hierarquia dos usuários. O Administrador Local (Admin Local) possui todas as funções do Usuário Final, como criação de VMs, mas possui funções próprias como migração de VMs com origem e destino em máquinas de seu sítio (isto é, migração local). O Administrador Global (Admin) possui a função de Admin Local de todos os

sítios, além de possuir funções próprias, como a criação de novos usuários e a migração de VMs entre sítios diferentes (isto é, migração global).

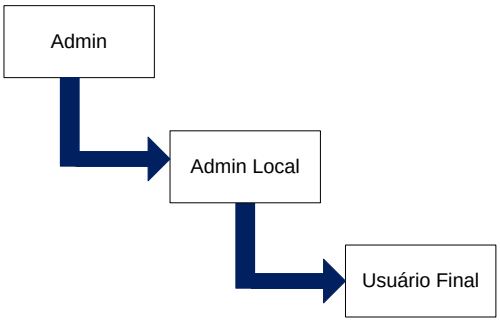

Fig. 9. Hierarquia dos usuários no GT-PID.

Para a criação do Admin Local, utilizou-se o tipo de controle de acesso denominado RBAC (*Role Based Access Control* – Controle de Acesso Baseado em Papéis) no qual cada usuário possui um ou mais papéis (*roles*) na utilização do sistema. A Fig. 10 apresenta os papéis necessários ao Administrador Local tomando-se como exemplo o sítio da UERJ. Todo Administrador Local possui o papel de membro, que é o papel dado pelo OpenStack aos Usuários Finais. Vale notar que, na arquitetura do GT-PID, os Usuários Finais possuem apenas o papel de membro, que permite a criação e visualização de VMs. Além do papel de membro, o Admin da UERJ possuirá também o papel adm_local, que permite o acesso à interface web de administração local, e o papel adm_uerj, que permite a realização de tarefas de gerenciamento em máquinas do sítio da UERJ. Da mesma forma, o Administrador Local da UFRJ possuirá todos os papéis da Fig. 10, à exceção do adm_uerj, que será substituído pelo adm_ufrj. A Fig. 11 apresenta um exemplo da interface de gerenciamento do Administrador Local da UERJ.

Fig. 10. Exemplo de papéis para o Administrador Local da UERJ.

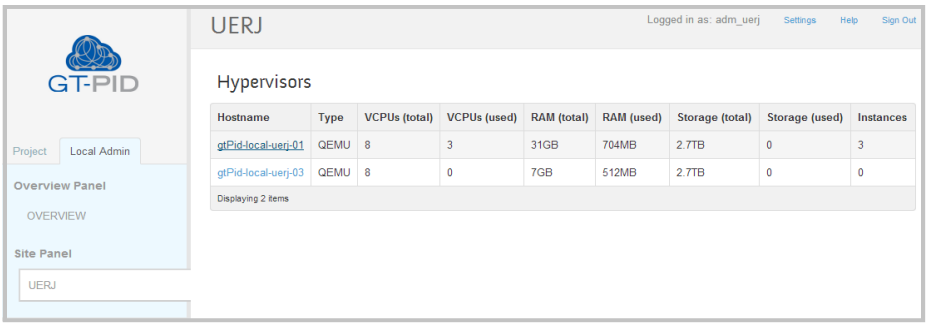

Fig. 11. Exemplo da interface de gerenciamento do Administrador Local para o sítio da UERJ.

O Administrador Global possui, além de seus papéis específicos, todos os papéis de Administradores Locais e de Usuários Finais, como mostrado na Fig. 12. Nessa figura, considera-se que a infraestrutura possui dois sítios: UFRJ e UERJ. Caso um novo sítio seja adicionado, o Administrador Global necessitará receber o papel de administrador do novo sítio como, por exemplo, adm_uff no caso da UFF entrar na nuvem.

Fig. 12. Exemplo de papéis para o Administrador Global.

A arquitetura do GT-PID permite a migração de máquinas virtuais entre Servidores de VMs. A migração é útil, por exemplo, no caso de manutenção de Servidores de VMs. Dessa forma, é possível migrar todas as máquinas de um servidor para outro e efetuar as operações de manutenção necessárias.

4.5 Migração

A migração disponível no piloto pode ser local ou global. A migração local, a qual é permitida aos Administradores Locais, consiste na migração de máquinas virtuais entre Servidores de VMs de um mesmo sítio. Na migração global, por sua vez, é possível realizar migrações entre Servidores de VMs de sítios diferentes. No piloto atual, esse tipo de migração é permitido apenas ao Administrador Global.

Os dois tipos de migração disponíveis são realizados de forma ao vivo, isto é, sem a necessidade de desligar as VMs. Vale notar que durante a migração pode haver um período de indisponibilidade das VMs, dependendo da quantidade de dados a serem transferidos entre os dois Servidores de VMs envolvidos. Na migração local, esse período tende a ser pequeno visto que as VMs em um mesmo sítio compartilham o mesmo disco físico a partir de um servidor central. Dessa forma, são necessárias apenas cópias da RAM virtual, do estado da CPU virtual, etc, o que representa tipicamente uma menor quantidade de dados. Na migração global é necessária também a migração do disco da VM para o novo sítio, o que acarreta em períodos mais longos de indisponibilidade.

Um requisito para o funcionamento da migração ao vivo é que os dois Servidores de VMs executem o mesmo conjunto de instruções. Por exemplo, a princípio, não é possível migrar uma máquina de um Servidor de VMs com processador Xeon para um Servidor de VMs com processador i7, visto que esses processadores utilizam conjuntos de instruções diferentes entre si. Essa limitação ocorre, pois o Sistema Operacional (SO) de uma máquina virtual seleciona um conjunto de instruções na sua inicialização. No caso da migração ao vivo, no qual a VM não é reiniciada, a VM continua utilizando o mesmo conjunto de instruções disponível quando foi iniciada. Assim, o SO da VM poderia tentar usar uma instrução que não está mais presente na CPU da máquina de destino, caso a migração envolvesse servidores com conjuntos de instruções diferentes. Tendo isso em vista, o OpenStack bloqueia a migração ao vivo entre CPUs com conjuntos de instrução diferentes. Essa exigência poderia limitar a inserção de novos nós no piloto do GT-PID, pois todos os Servidores de VMs teriam que utilizar uma arquitetura de CPU homogênea. A Libvirt, no entanto, possui formas de evitar que certas instruções muito específicas sejam usadas pelo SO, de modo a aumentar a compatibilidade na hora da migração. A solução da Libvirt é forçar as VMs a utilizar um conjunto de instruções genérico, suportado por todas as CPUs. Dessa forma, no GT-PID utilizou-se essa opção da Libvirt e modificou-se o OpenStack para aceitar esse novo comportamento.

4.6 Modificações no código do OpenStack

Como mencionado anteriormente, a Fig. 7 ilustra os módulos do OpenStack utilizados, mostrando os que foram modificados e os que foram adotados em sua forma original. Nessa figura, os módulos são classificados de um a três asteriscos, dependendo da quantidade de modificações realizadas. Um asterisco indica pequenas modificações, enquanto três asteriscos representam grandes modificações.

A lista a seguir apresenta uma breve descrição das principais modificações realizadas nos módulos do OpenStack:

Keystone:

- Inserção de políticas específicas do Administrador Local.

glance-api:

- Inserção de políticas específicas do Administrador Local.

cinder-api:

- Inserção de políticas específicas do Administrador Local.

ceilometer-api:

- Inserção de políticas específicas do Administrador Local.

nova-api:

- Inserção de políticas específicas do Administrador Local;
- Passagem de parâmetros necessários ao Escalonador de Sítios.

nova-scheduler:

- Implementação de esquemas de filtros e pesos para o Escalonador de Sítios.

nova-compute:

- Modificação do código para forçar que volumes sejam alocados nos mesmos sítios nos quais as VMs foram alocadas;
- Modificação do código para permitir migração de VMs entre Servidores de VMs com CPUs diferentes.

Horizon

- Modificação do formulário de criação de VMs para considerar as opções de criação do GT-PID (p.ex. criação distribuída ou centralizada);
- Inserção de botão e formulário para migração Global;
- Inserção de botão e formulário para migração Local;
- Inserção de interface completa de gerenciamento para Administrador Local.

5. Conclusões e Trabalhos Futuros

O piloto desenvolvido emprega o paradigma de Computação em Nuvem para compartilhamento de infraestrutura computacional entre universidade e centros de pesquisa. Como a infraestrutura desse tipo de instituição é geralmente utilizada em rajadas (ou seja, alta utilização durante curtos períodos e com ociosidade em boa parte do tempo), o compartilhamento de infraestrutura é uma alternativa viável para melhorar o aprovisionamento de recursos computacionais nessas instituições. Além disso, serviços de Computação em Nuvem tornam-se cada vez mais essenciais a

diversas instituições. Entretanto, devido à escassez de infraestruturas de nuvem acadêmicas, as universidades geralmente adotam soluções oriundas da iniciativa privada, gerando maior oneração. O GT-PID propôs uma infraestrutura em nuvem geodistribuída entre universidade e centros de pesquisa. Os resultados do projeto mostraram que é viável implementar uma nuvem geodistribuída baseada na plataforma de nuvem OpenStack. Atualmente, o piloto está instalado em três universidade brasileiras, UFRJ, UERJ e UFF.

Um trabalho futuro, e já em andamento, consiste em integrar a plataforma do piloto a um esquema de autenticação federada provido pela RNP, denominado CAFe, possibilitando o acesso ao serviço por usuários de todas as instituições da federação. Assim, o usuário da infraestrutura poderá utilizar suas credenciais fornecidas por suas instituições de origem, ao invés de ser obrigado a realizar um novo cadastro para utilização do serviço. Além disso, adicionam-se novas funcionalidades de rede ao piloto de forma a permitir que VMs de diferentes sítios comuniquem-se entre si de forma segura.

Referências

1. Bari, M. F., Boutaba, R., Esteves, R., Granville, L. Z., Podlesny, M., Rabbani, M. G., Qi, Zhang, Zhani, M. F.: Data Center Network Virtualization: A Survey. IEEE Communications Surveys & Tutorials 15(2), 909 a 928 (2013)
2. Costa, L. H. M. K., Amorim, M. D., Campista, M. E. M., Rubinstein, M. G., Florissi, P., Duarte, O. C. M. B.: Grandes Massas de Dados na Nuvem: Desafios e Técnicas para Inovação. Em: Minicursos do Simpósio Brasileiro de Redes de Computadores (SBRC'2012), pp. 1 a 58. Sociedade Brasileira de Computação, Porto Alegre (2012)
3. Khan, A., Zugenmaier, A., Jurca, D., Kellerer, W.: Network Virtualization: A Hypervisor for the Internet?. IEEE Communications Magazine 50(1), 136 a 143 (2012)
4. Duan, Q., Yan, Y., Vasilakos, A. V.: A Survey on Service-Oriented Network Virtualization Toward Convergence of Networking and Cloud Computing. IEEE Transactions on Network and Service Management 9(4), 373 a 392 (2012)
5. Alves, R. S., Campista, M. E. M., Costa, L. H. M. K., Duarte, O.C. M. B.: Towards a Pluralist Internet Using a Virtual Machine Server for Network Customization. Em: Asian Internet Engineering Conference (AINTEC'2012), pp. 9 a 16. ACM Press, Nova Iorque (2012)
6. CAFe – Comunidade Acadêmica Federada, http://portal.rnp.br/web/servicos/cafe
7. Apache CloudStack – Open Source Cloud Computing, http://cloudstack.apache.org/
8. Eucalyptus - Open Source AWS Compatible Private Cloud, http://www.eucalyptus.com/
9. AMPQ - Advanced Message Queuing Protocol, http://www.amqp.org/
10. Libvirt – The Virtualization API, http://libvirt.org/
11. Richardson, T., Stafford-Fraser Q., Wood, K.R., Hopper, A.: Virtual Network Computing. IEEE Internet Computing 2(1), 33 a 38 (1998)