

Uma implementação de um nó computacional de baixo consumo de energia em uma nuvem OpenStack utilizando Raspberry Pi

João Vítor Valente Tarouco de Oliveira¹; Vitor Alano Ataídes²; Maurício Lima Pilla³; Laércio Lima Pilla⁴

¹Universidade Federal de Pelotas – jvvtoliveira@inf.ufpel.edu.br

²Universidade Federal de Pelotas – vaataides@inf.ufpel.edu.br

³Universidade Federal de Pelotas – pilla@inf.ufpel.edu.br

⁴Universidade Federal de Santa Catarina – laercio.pilla@ufsc.br

1. INTRODUÇÃO

A computação em nuvem pode ser considerada um modelo de negócio que procura disponibilizar recursos computacionais, tais como poder de processamento e armazenamento, como serviços (Erl et al., 2013). Dentro desse paradigma, existem soluções de código aberto e fechado para a criação e manutenção de nuvens computacionais definindo especificações da arquitetura do sistema e exigindo que provedores de computação em nuvem sigam diretivas para implementação correta de soluções. Dentre essas soluções, o OpenStack se destaca pela sua modularidade que provê múltiplas funcionalidades à nuvem utilizando diferentes módulos interconectados que enriquecem seu gerenciamento e funcionamento.

Devido ao consumo elevado de energia por *data centers*, a necessidade de se prover diferentes tecnologias e recursos que se desenvolvam nas premissas da computação verde (*Green Computing*) cresceu na última década (Song et al., 2015). Com esse objetivo, uma das soluções para promover tal mudança é a utilização de nós com baixo consumo de energia como substitutos dos nós computacionais convencionais.

Os *softwares* de simulação de ambientes de nuvem expressam apenas parte das características de um *data center* real e a construção de um *data center* para testes acadêmicos é inviável em razão do custo (Tso et al., 2013). Portanto, a solução proposta por Tso et al. (2013) é implementar uma nuvem utilizando dispositivos Raspberry Pi, pois estes expressam as características de *data centers* com grande fidelidade mantendo o custo acessível.

Dos Reis et al. (2016) fizeram uma análise de consumo de energia de um *cluster* de dispositivos com arquitetura ARM e processadores de baixo consumo de energia, utilizando cargas de trabalho padronizadas. Foi concluído que um aumento no número de nós do *cluster* ARM tem baixo impacto no consumo de energia. Porém, o acesso lento à memória prejudica a escalabilidade do sistema quando a carga de trabalho não é projetada especificamente para a computação distribuída.

Neste trabalho é desenvolvido um nó computacional de baixo consumo usando um dispositivo Raspberry Pi como base, o acoplando a nuvem já existente no LUPS (*Laboratory of Ubiquitous and Parallel Systems*), situado na Universidade Federal de Pelotas. Esta nuvem está descrita na Seção 2 junto de uma descrição detalhada do Raspberry Pi utilizado. A Seção 3 apresenta descrição das dificuldades e particularidades do acoplamento à nuvem. Por fim a Seção 4 conclui o artigo e discute trabalhos futuros.

2. METODOLOGIA

Uma implementação básica de uma nuvem OpenStack utilizando os módulos indicados na Tabela 1 está organizada em um nó de controle que funciona tanto como controle quanto gerenciador de rede, seguido por dez nós computacionais.

Os primeiros nove nós são compostos por hardware convencional utilizando o agente virtualizador KVM enquanto o décimo nó é composto de um dispositivo de baixo consumo de energia com o agente virtualizador Docker.

Tipicamente existe apenas um serviço de rede servindo as máquinas virtuais, sendo executado em um nó específico do serviço. Porém, pelo uso do serviço de rede Nova-Networking, o nó de controle serve como gerenciador de rede também. Isso apresenta um problema: caso o nó pare de funcionar, não é possível acessar as instâncias gerenciadas por ele. Também existe o risco desse nó se tornar um gargalo na nuvem caso o tráfego de rede que passa por ele seja excessivo.

Tabela 1. Relação entre os componentes instalados na nuvem e suas características

Componente	Detalhes
Armazenamento de bloco	Cinder
Gerenciador de imagens dos sistemas	Glance
Agente Virtualizador	KVM
Serviço de Rede	Nova-Networking
Banco de dados para gerenciamento	Postgres
Provedor de Identidade	Keystone

O hardware de baixo consumo utilizado neste trabalho foi um Raspberry Pi modelo B com processador de arquitetura ARMv6 e seu sistema operacional Arch Linux.

Por ser um dispositivo da primeira geração dos Raspberry Pi, muitas funcionalidades mais atuais, tais como suporte a virtualização no kernel, não existem portanto sendo necessário adaptar os componentes de um nó computacional OpenStack para o dispositivo. Isto também decorre do fato da Canonical, empresa que é uma das principais desenvolvedoras do OpenStack, não provê suporte dos componentes a distribuições de Linux ARM.

Adaptar os componentes de um nó computacional para este dispositivo envolve obter o código fonte de cada componente individualmente e executar uma compilação diretamente no dispositivo ou usar uma cadeia de *cross-compiling* em um processador convencional. A primeira opção apresenta uma limitação de tempo pois o hardware presente no dispositivo é limitado em termos de processamento, o que torna o processo muito lento. A segunda opção apresenta um modo mais elegante de compilar o código fonte dos módulos porém há a necessidade de transferir os executáveis compilados para

o dispositivo, junto dos arquivos de configuração necessários, através de algum meio sendo ele presencial ou remoto.

3. RESULTADOS

Como não há suporte a uma virtualização utilizando o módulo KVM, que possui grande adesão pela comunidade, a alternativa LXC (*Linux Containers*) através da ferramenta Docker foi utilizada como módulo de virtualização do sistema. Este método provê um ambiente muito próximo do encontrado em uma máquina virtual tradicional porém dispensando o sobrecurso de virtualizar o kernel do sistema e seu hardware. De um modo geral, pode-se caracterizar *containers* como um grupo de processos executando em cima do kernel da máquina porém isolados dos demais *containers* e processos de outros usuários. Técnicas de *namespace* proveem ao usuário uma visão única do sistema e grupos de controle gerenciam os recursos utilizados por determinado usuário.

A virtualização de uma instância nesse nó tem de ser feita utilizando uma imagem de sistema específica do Docker selecionado na *Dashboard* do Opentack quando um usuário deseja criar a instância de máquina. Esta instância possui uma arquitetura condizente com o hardware de baixo consumo, portanto, é necessário que os programas sejam compilados tomando essa arquitetura em mente. Migrações de instâncias podem ser feitas apenas entre nós com arquiteturas iguais o que leva a conclusão de que, em uma nuvem com mais de um nó de baixo consumo, é possível executar as mesmas trocas de instância que aconteceriam em um nó convencional portanto os mesmos algoritmos de escalonamento são viáveis nestes nós.

Para mostrar que o acoplamento foi completo com sucesso, a Figura 1 apresenta um teste de conexão entre o nó computacional 10, composto pelo dispositivo de baixo consumo, e o nó de controle.

```
[alarm@alarmpi ~]$ ping controller
PING controller (192.168.0.20) 56(84) bytes of data.
64 bytes from controller (192.168.0.20): icmp_seq=1 ttl=64 time=0.573 ms
64 bytes from controller (192.168.0.20): icmp_seq=2 ttl=64 time=0.348 ms
lups@controller:~$ ping compute10
PING compute10 (192.168.0.10) 56(84) bytes of data.
64 bytes from compute10 (192.168.0.10): icmp_seq=1 ttl=64 time=0.497 ms
64 bytes from compute10 (192.168.0.10): icmp_seq=2 ttl=64 time=0.405 ms
```

Figura 1. Teste de conexão entre o nó de controle e o nó computacional

4. CONCLUSÕES E TRABALHOS FUTUROS

Neste artigo foi apresentada a possibilidade de criação de um nó computacional em uma nuvem OpenStack utilizando dispositivos com processadores de baixo consumo de energia, descrevendo o seu fator limitante, a adaptação dos componentes. Por esses não possuírem suporte por padrão na solução escolhida, foi necessário compilar e instalar cada módulo a partir de seu código fonte, tarefa que se torna computacionalmente demorada se a compilação for executada nos próprios dispositivos.

Como trabalho futuro pretende-se realizar o acoplamento do nó utilizando o módulo Neutron e na sequência medir e comparar o consumo e poder computacional do nó de baixo consumo frente a um nó convencional.

5. REFERÊNCIAS BIBLIOGRÁFICAS

ERL, T., PUTTINI, R., MAHMOOD, Z. **Cloud computing: concepts, technology & architecture**. New York: Pearson Education, 2013.

TSO, F. P., WHITE, D. R., JOUET, S., SINGER, J., PEZAROS, D. P.. The glasgow Raspberry Pi cloud: A scale model for cloud computing infrastructures. **2013 IEEE 33rd International Conference on Distributed Computing Systems Workshops**. p. 108-112, 2013

DOS REIS, E. S., ROSTIROLLA, G., e DA ROSA RIGHI, R.. Análise do Consumo de Energia de um Cluster ARM Utilizando Benchmark NAS. **Escola Regional de Alto Desempenho**, p. 227-230, 2016

SONG, Z., ZHANG, X., ERIKSSON, C.. Data Center Energy and Cost Saving Evalutation. **Energy Procedia**. p. 1255-1260. 2015