

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

João Vítor V.T. de Oliveira<sup>1</sup>, Vitor A. Ataídes<sup>1</sup>, Maurício L. Pilla<sup>1</sup>, Laércio L. Pilla<sup>2</sup>

<sup>1</sup> Universidade Federal de Pelotas – (UFPEL)  
Pelotas – RS – Brasil

<sup>2</sup> Universidade Federal de Santa Catarina – (UFSC)  
Florianópolis – SC – Brasil.

**Abstract.** *Data centers have a big impact on energy consumption. The need for new technologies that mitigate this impact has grown in the past ten years. One of the solutions for reducing the energy impact is the exchange of conventional computational nodes for their low power counterparts. This article presents the implementation of a low power compute node on an OpenStack cloud using a Raspberry Pi.*

**Resumo.** *Como data centers apresentam um consumo elevado de energia, a necessidade de se prover diferentes tecnologias que tenham um impacto menor no consumo cresceu na última década. Uma das abordagens para a solução desse problema é a troca de nós computacionais convencionais por nós computacionais de baixo consumo utilizando hardware específico para isto. Este artigo apresenta o acoplamento de um nó computacional de baixo consumo Raspberry Pi a uma nuvem OpenStack.*

## 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 a serem contratados [Erl et al. 2013]. Dentro desse paradigma existem soluções de código aberto e de código fechado para a criação e manutenção de nuvens computacionais definindo especificações da arquitetura do sistema e exigindo que provedores de computação em nuvem sigam diretrizes para implementação correta de soluções. Dentre estas soluções o OpenStack [OpenStack 2016b] se destaca pela sua modularidade que provê múltiplas funcionalidades à nuvem, enriquecendo 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 (*compute nodes*) 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

**Tabela 1. 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

acadêmicos é inviável em razão do custo [Tso et al. 2013]. Portanto, a solução proposta é criar uma nuvem utilizando dispositivos Raspberry Pi, pois estes expressam as características de *data centers* com grande fidelidade mantendo o custo acessível.

Em (Dois Reis et al. 2016) foi feita uma análise de consumo de energia de um cluster de dispositivos com processadores de baixo consumo, de arquitetura ARM, utilizando cargas de trabalho padronizadas. Foi concluído que um aumento no número de nós de um 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 utilizando um dispositivo Raspberry Pi como base. O acoplando a nuvem já existente no LUPS (*Laboratory of Ubiquitous and Parallel Systems*) localizado na Universidade Federal de Pelotas. Esta nuvem está descrita na Seção 2.1. Na sequência a seção 2.2 descreve o dispositivo de baixo consumo utilizado. A Seção 2.3 descreve as dificuldades e particularidades do acoplamento à nuvem. Por fim a Seção 3 conclui o artigo e discute trabalhos futuros.

## 2. Implementação

Nesta seção é apresentada a implementação de um nó computacional OpenStack em um dispositivo de baixo consumo. Primeiro será apresentada a nuvem já instalada que foi usada como base para o acoplamento do dispositivo. Na sequência é detalhado o dispositivo de baixo consumo escolhido, onde foram instalados o módulos computacionais do OpenStack apresentando sua arquitetura, algumas especificações e as dificuldades de instalação. A Seção seguinte apresenta particularidades do *hardware* de baixo consumo frente a um *hardware* convencional.

### 2.1. Estrutura

Está presente no laboratório uma implementação básica de uma nuvem OpenStack utilizando os módulos indicados na Tabela 1

A nuvem está organizada em um nó de controle 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 ativo e gerenciando as máquinas virtuais, sendo executado em um nó específico para este fim. 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. [OpenStack 2016a].

## 2.2. Dispositivo de baixo consumo

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

Por ser um dispositivo da primeira geração dos Raspberry Pi, algumas funcionalidades presentes em versões mais atuais, como suporte a virtualização no *kernel*, não existem neste dispositivo portanto tanto o agente virtualizador quando componentes do nó computacional possivelmente estão indisponíveis neste dispositivo, isto gera a necessidade de utilização de alternativas que suprem a necessidade do módulo ausente. A indisponibilidade de pacotes prontos dos componentes decorre, também, do fato da Canonical, empresa que é uma das principais desenvolvedoras do OpenStack, não prover suporte a distribuições de Linux ARM.

Adaptar os componentes de um nó computacional para este dispositivo envolve obter o código fonte do componente, disponível devido a solução ser *open source*, executar uma compilação diretamente no Raspberry Pi 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 inferior, em termos de processamento, ao processador convencional o que potencialmente inviabiliza a completa instalação dos módulos em tempo hábil. Por outro lado, 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, seja ele presencial ou remoto.

A virtualização de uma instância nesse nó tem de ser feita utilizando uma imagem de sistema específica do Docker que é selecionada no momento da criação da instância na *Dashboard* do OpenStack, isto indica a solução aonde esta máquina virtual deve residir dentre os nós computacionais. 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 entre os nós computacionais podem ser feitas apenas entre nós com arquiteturas iguais o que leva a conclusão que, em uma nuvem com mais de um nó computacional de baixo consumo, é possível aplicar os algoritmos de escalonamento já existente nestes nós.

Para mostrar que o acoplamento foi completo com sucesso, a Figura 1 apresenta um teste de conexão entre o nó computacional 10, aqui representado por [alarm@alarmpi], e o nó de controle, representado por [lups@controller].

## 2.3. Particularidades

Os módulos necessários para o nó computacional são Nova-Networking, Nova-Compute e os componentes do módulo virtualizador.

Como não há suporte a uma virtualização utilizando o módulo KVM, que possui grande adesão pela comunidade, a alternativa LXC (*Linux Containers*) através do agente

```
[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

virtualizador Docker foi utilizada como módulo de virtualização do sistema. Este método de virtualização provê um ambiente muito próximo do encontrado em uma máquina virtual tradicional porém dispensando o sobrecusto de virtualizar o *kernel* o 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 do resto dos *containers* e processos de outros usuários [Linux Containers 2016], técnicas de *namespace* proveem ao usuário uma visão única do sistema e grupos de controle gerenciam os recursos utilizados por determinado usuário.

### 3. Conclusão e Trabalhos Futuros

Este artigo apresenta que é possível criar um nó *compute* em uma nuvem OpenStack utilizando dispositivos com processador de baixo consumo de energia descrevendo o seu fator limitante, a adaptação dos componentes. Por estes não possuírem suporte por padrão da 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.

### Referências

- Erl, T., Puttini, R., and Mahmood, Z. (2013). *Cloud computing: concepts, technology, & architecture*. Pearson Education.
- Linux Containers (2016). Linux Containers. <https://www.linuxcontainers.org/>.
- OpenStack (2016a). Documentation, Networking with nova-network. <http://docs.openstack.org/admin-guide/compute-networking-nova.html>. Acessado Em: 01-08-2016.
- OpenStack (2016b). OpenStack. [www.openstack.org](http://www.openstack.org). Acessado Em: 01-08-2016.
- Song, Z., Zhang, X., and Eriksson, C. (2015). Data Center Energy and Cost Saving Evaluation. *Energy Procedia*, 75:1255–1260.
- Tso, F. P., White, D. R., Jouet, S., Singer, J., and Pezaros, D. P. (2013). The glasgow Raspberry Pi cloud: A scale model for cloud computing infrastructures. In *2013 IEEE 33rd International Conference on Distributed Computing Systems Workshops*, pages 108–112. IEEE.