CENTRO PAULA SOUZA
FACULDADE DE TECNOLOGIA DE FRANCA
“Dr. THOMAZ NOVELINO”

TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

CARLOS HENRIQUE RIBEIRO GOMES

CEPH COMO OBJECT STORAGE

Estudo de caso de implantação de hiperconvergência

FRANCA/SP
2020

CARLOS HENRIQUE RIBEIRO GOMES

CEPH COMO OBJECT STORAGE

Estudo de caso de implantação de hiperconvergência

Trabalho de Graduação apresentado à Faculdade de Tecnologia de Franca - "Dr. Thomaz Novelino", como parte dos requisitos obrigatórios para obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas

Orientadora: Profª. Drª. Jaqueline Brigladori Pugliesi

FRANCA/SP

2020

Ficha catalográfica

G633c Gomes, Carlos Henrique Ribeiro
CEPH COMO OBJECT STORAGE: estudo de caso de implantação de hiperconvergência / Carlos Henrique Ribeiro Gomes, [s.n], 2020

55f.; 30 cm; il

Trabalho de Graduação (Curso Superior de Análise e Desenvolvimento de Sistemas) Fatec - Faculdade de Tecnologia “Dr. Thomaz Novelino”.
Orientadora: Profa.Dra. Jaqueline Brigladori Pugliesi

1. Ceph. 2.Storage. 3 Proxmox. 4.Kvm. 5.Qemu.
I. Autor. II. Título.

CDD – 004

CARLOS HENRIQUE RIBEIRO GOMES

CEPH COMO OBJECT STORAGE

Trabalho de Graduação apresentado à Faculdade de Tecnologia de Franca – "Dr. Thomaz Novelino", como parte dos requisitos obrigatórios para obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas

Trabalho avaliado e aprovado pela seguinte Banca Examinadora:

Orientador(a).....: _____
Nome.....: Profª. Drª. Jaqueline Brigladori Pugliesi
Instituição.....: Faculdade de Tecnologia de Franca – "Dr. Thomaz Novelino"

Examinador(a) 1 : _____
Nome.....: Profª. Drª. Érica Aparecida Araújo
Instituição.....: Faculdade de Tecnologia de Franca – "Dr. Thomaz Novelino"

Examinador(a) 2.: _____
Nome.....: Prof. Me. Carlos Eduardo de França Roland
Instituição.....: Faculdade de Tecnologia de Franca – "Dr. Thomaz Novelino"

Franca, 01 de dezembro de 2020.

AGRADECIMENTO

Agradeço primeiramente a meus pais Ismar e Rosa e minha irmã Cristiani, que mesmo distantes são um apoio aos meus projetos.

Agradeço aos meus amigos *nerds* de longa data do IRC, alguns deles eternamente sem nome ou uma face pra associar, e aos gênios do Area31 Hackerspace, sempre alfinetando para que eu continuasse meus estudos.

Agradeço ao amigo Frederico Siena da UFTPr Londrina, que com um projeto similar em paralelo acabou indo muito mais longe e serviu de incentivo para que eu me aprofundasse no conhecimento da ferramenta, fizesse laboratórios e produzisse uma documentação à altura.

Agradeço aos Mestres Bruno Emanuel e Tomas Waldow, sempre disponíveis para iluminar o grupo do ProxMox no Telegram com suas experiências, e a grande atenção que deram as minhas dúvidas no início do projeto.

Agradeço à minha companheira Mina, que supervisionou e auditou parte desta documentação e me atura focado no computador por dias a fio.

Agradeço aos grandes amigos do Entebate, responsáveis por grande parte da carga filosófica e distorção semiótica para continuar procurando desafios.

Agradeço à minha orientadora Jaqueline, que aceitou a tarefa problemática de me apoiar no universo acadêmico e enfim concluir a graduação da FATEC.

Agradeço, por fim, aos servidores da Unesp Franca, que nunca pouparam esforços em incentivar minha qualificação profissional.

Dedico o presente Trabalho de Graduação ao meu pai Professor (de tudo menos matemática) Ismar e a todos os companheiros que ao longo dos anos me ensinaram mais que qualquer curso.

Intelligence is the ability to avoid doing work, yet getting the work done.

Linus Torvalds

RESUMO

O tema desta monografia surgiu a partir do contexto de um laboratório na Diretoria Técnica de Informática no campus de Franca da Unesp. O laboratório que antes fazia fazia uso da ferramenta proprietária VMware sobre equipamentos relativamente antigos apresentava problemas de performance, gerenciamento e altos custos. Devido às limitações da gestão estatal quanto à novas aquisições, mostrou-se necessário pesquisar alternativas similares, acessíveis, open source e sem custo de licenciamento para utilização. Assim, o presente documento tem por objetivo apresentar as melhorias realizadas nesse laboratório, como a implantação de hiperconvergência utilizando-se da combinação de virtualização baseada em QEMU + KVM sobre Proxmox e armazenamento sobre Ceph, além de algumas aplicações e funcionalidades dessas ferramentas livres e as dificuldades encontradas nesse processo. O uso das ferramentas descritas revelou-se uma solução sem custos financeiros extras, dependente principalmente de estudo e tempo de trabalho dos funcionários envolvidos. Os resultados obtidos com essas mudanças foram amplamente satisfatórios na entrega de um cluster de alta disponibilidade e fácil gerenciamento. Além disso, servem como incentivo para que sejam feitas considerações e análises de aderência em ferramentas de código aberto antes de se optar exclusivamente por serviços proprietários, considerando a competência e resiliência de projetos da magnitude do Ceph e Proxmox abordados nesse documento.

Palavras-chave: Ceph. Storage. Proxmox. Kvm. Qemu.

ABSTRACT

The theme of this monograph arose from the context of a laboratory at the Unesp Franca TI department. The laboratory used to work over the proprietary tool VMware with relatively old equipment and presented problems of performance, management and high costs. Due to the limitations of the state management in terms of new acquisitions, it proved necessary to search for similar, accessible, open source alternatives and without the cost of licensing for use. Thus, this document aims to present the improvements made in this laboratory, such as the implementation of hyperconvergence using the combination of virtualization based on QEMU + KVM over Proxmox and storage over Ceph, in addition to some applications and features of these free tools and the difficulties encountered in this process. The application of the described tools proved to be a solution without extra financial costs, mainly dependent on study and working hours of the employees involved. The results obtained with these changes were largely satisfactory in the delivery of a highly available and easily managed cluster. In addition, they work as an incentive regarding the usage and consideration of open source tools before opting exclusively for proprietary services, considering the competence and resilience of projects of the magnitude of Ceph and Proxmox addressed in this document.

Keywords: Ceph. Storage. Proxmox. Kvm. Qemu.

LISTA DE FIGURAS

LISTA DE QUADROS

LISTA DE SIGLAS

API – *Application Programming Interface*

AGPL – *Affero General Public License*

AMD – *Advanced Micro Devices*

ARM – *Advanced RISC Machine*

BSD – *Berkeley Software Distribution*

CERN – *Conseil Européen pour la Recherche Nucléaire*

CIFS – *Common Internet File system*

CPU – *Central Processing Unit*

CRUSH – *Controlled Replication Under Scalable Hashing*

DMA – *Direct Memory Access*

DTI – Diretoria Técnica de Informática

FCHS – Faculdade de Ciências Humanas e Sociais

FFS – *Fast File System*

FS – *File System*

FUSE – *Filesystem in Userspace*

GNU – *GNU's not Unix*

GPL – *General Public License*

IOMMU – *Input-Output Memory Management Unit*

KVM – *Kernel-based Virtual Machine*

LACP – *Link Aggregation Control Protocol*

LAGG – *Link Aggregation*

LAN – *Local Area Network*

LGPL – *Lesser General Public License*

LXC – *Linux Containers*

MDS – *Metadata Server*

MMU – *Memory Management Unit*

NAS – *Network Attached Storage*

NFS – *Network File System*

OSD – *Object Storage Device*

PG – *Placement Group*

PPC – *Power PC*

POSIX – *Portable Operating System Interface*

QEMU – *Quick EMUlator*

RADOS – *Reliable Autonomous Distributed Object Store*

SAN – *Storage Area Network*

SDN – *Software-defined Network*

SDS – *Software-defined Storage*

SSRC – *Storage Systems Research Center*

UNESP – Universidade Estadual Paulista

VE – *Virtual Environment*

VFS – *Virtual File System*

VLAN – *Virtual Local Area Network*

VM – *Virtual Machine*

VXLAN – *Virtual Extensible LAN*

ZFS – *Zettabyte File System*

SUMÁRIO

1 INTRODUÇÃO

Ao longo dos anos, os sistemas de armazenamento tiveram seus custos aumentados e reduzidos. Contudo, várias arquiteturas de sistemas de armazenamento não atingiram níveis de confiabilidade e escalabilidade, ou ainda dependem de muitos equipamentos específicos (WEIL, 2007).

Atualmente, grandes sistemas de armazenamento distribuído baseados em OSDs (*Object storage devices*) utilizam tecnologias de décadas anteriores. Esses sistemas foram projetados sobre computadores de grande porte que continuavam sendo uma coleção de processadores, memórias e discos executando operações de baixo nível, mas limitando sua escalabilidade.

Weil (2007) apresentou o protótipo da tecnologia Ceph com um conceito diferenciado na utilização de um sistema de arquivos distribuído, assumindo que sistemas na escala de petabytes são invariavelmente dinâmicos, grandes e construídos de forma incremental, nos quais falhas em *nodes* são a norma ao invés da exceção e a carga de utilização muda constantemente ao longo do tempo.

Com estas considerações, foi escolhido aprofundar-se nos recursos e aplicações dessa ferramenta de código aberto em detrimento das outras ferramentas líderes de mercado utilizadas em parques de armazenamento de grande porte, gerando uma abordagem inicial sobre o projeto e talvez criando uma referência básica sobre seu uso e aplicação, aproveitando a necessidade de evitar custos de licenciamento de software no ambiente da Unesp Franca.

Além dos sistemas de armazenamento abordados, é necessário passar pelo universo da virtualização, pois um dos grandes motivadores para que os servidores físicos se tornassem um ambiente homogêneo e gerenciável foi a facilidade de administração dos serviços, majoritariamente entregues por máquinas especializadas, gradativamente virtualizadas ao longo dos anos.

A preocupação desta pesquisa deu-se também pelas dificuldades de aquisição de novos equipamentos no ambiente da Unesp. Diante desse limitador, era preciso descobrir como se fazer o melhor uso possível com os equipamentos já existentes, evitando depender de novas aquisições, reparos ou atualizações das máquinas disponíveis.

Como diretriz organizacional da Unesp na criação de projetos de TI, a

priorização de ferramentas livres e softwares com licenciamento permissivo é importante, já que o custo adicional gasto com o software em situações passíveis de solução com ferramentas criadas pela própria comunidade não se mostra viável aos olhos do Estado. Além disso, destaca-se que a vasta maioria dos servidores e ferramentas responsáveis pelo funcionamento de serviços na universidade são baseados em ferramentas de código aberto, hospedadas em servidores sobre sistemas operacionais livres, rodando softwares construídos em linguagens de programação não proprietárias.

Este documento está dividido em: uma breve introdução sobre conceitos e as ferramentas utilizadas, uma descrição de alguns recursos da ferramenta Ceph e, finalmente, um estudo de caso realizado na FCHS (Faculdade de Ciências Humanas e Sociais) durante a migração dos servidores virtualizados para outra plataforma utilizando o Ceph como *storage*. Ao final, será apresentado um comparativo acerca do investimento necessário para obter-se resultados similares com ferramentas proprietárias em detrimento de ferramentas livres (Softwares Livres ou *Open Source*).

2 COMPOSIÇÃO

No citado projeto de migração utilizamos as seguintes ferramentas na construção do *cluster* hiperconvergente de alta disponibilidade: ProxMox VE(*Virtual Environment*), KVM (*Kernel-based Virtual Machine*), QEMU (*Quick EMUlator*), LXC (*Linux Containers*) e armazenamento Ceph, descritos de forma mais detalhada a seguir.

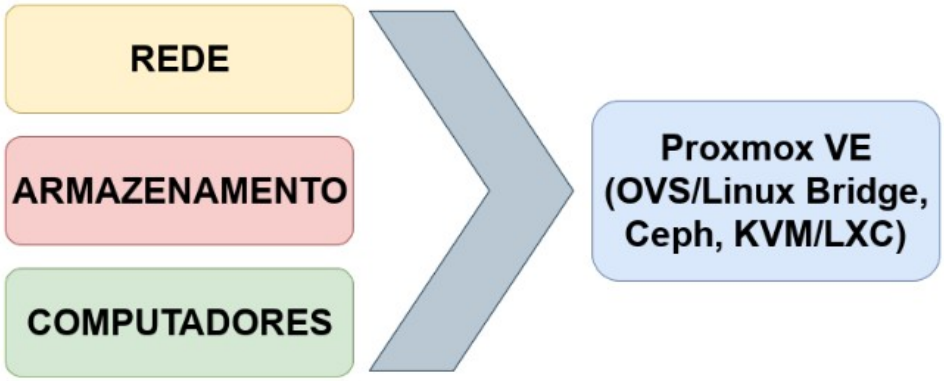
Segundo Seo (2009), o conceito de virtualização se apresenta como "uma metodologia ou arcabouço que possibilita a divisão dos recursos de um computador entre múltiplos ambientes de execução". Esse conceito macro será importante para compreender o objetivo de entregar os serviços majoritariamente em ambiente virtualizado, em contraposição à utilização de hipervisores no *bare-metal*.

2.1 HIPERCONVERGÊNCIA

2.1.1 Definição

A hiperconvergência é uma infraestrutura definida por software que desmembra as operações de infraestrutura do hardware do sistema e as converge em um único bloco no nível do hipervisor. Como exemplificado no modelo da Figura 1, os sistemas hiperconvergentes aproveitam a inteligência definida por software para eliminar as dificuldades de armazenamento e computação e permitir que esses recursos sejam executados e gerenciados na mesma plataforma de servidor.

Figura 1 - Conceito macro de hiperconvergência



Fonte: Do autor

2.1.2 VANTAGENS

A hiperconvergência elimina os tradicionais problemas de gerenciamento de TI, agrupando serviços de data center, como servidores, armazenamento e rede, em diferentes pacotes, permitindo assim que sejam gerenciados por uma única aplicação, eliminando ineficiências e acelerando a entrega de serviços computacionais.(HPE, 2020)

Algumas vantagens notáveis:

- Facilidade de implantação: os sistemas hiperconvergentes costumam já virem pré-configurados ou com o ambiente de processamento já integrado ao de armazenamento, sendo mais rápida sua entrega inicial.
- Baixo Custo: a possibilidade de utilização de equipamento heterogêneo permite dar novos destinos a equipamentos antigos ou aquisição de lotes de equipamentos mais baratos que uma nova aquisição.
- Agilidade: com um ambiente ideal para virtualização, a criação de novos serviços e realocação de recursos é feita quase que instantaneamente quando necessário.
- Escalabilidade: com os recursos distribuídos entre vários *nodes*, agregar mais recursos ao arranjo é uma tarefa simples e com quase nenhuma indisponibilidade dos serviços já disponibilizados.
- Facilidade de Recrutamento: devido às facilidades de manutenção e implantação dos serviços, não é necessária uma equipe imensa para gerir os recursos, concentrando as tecnologias que os administradores devem dominar e facilitando a especialização na entrega de melhorias contínuas. (HARVEY, 2016)

2.1.3 DESVANTAGENS

Como qualquer outra tecnologia, a hiperconvergência pode não ser adequada para todo tipo de situação, o que deve ser avaliado de acordo com cada projeto e aplicação. Algumas desvantagens possíveis:

- Performance: ambientes hiperconvergentes nem sempre estão construídos com hardware topo de linha e, por abarcarem uma quantidade vasta de possibilidades em ambientes virtualizados, os períodos de testes e

homologação de alguns fabricantes podem ser mais longos do que soluções instaladas isoladas em servidores convencionais.

- Alto Custo: com a possibilidade de um ambiente com vários equipamentos distintos, para os fabricantes de soluções de software proprietário é necessário validar e garantir a integridade do orquestrador que pode apresentar problemas de software, processamento ou armazenamento e dificultar o diagnóstico, tornando o suporte e licenciamento destas soluções relativamente caros.
- Escalonamento Inflexível: a escalabilidade que torna a hiperconvergência ideal para algumas cargas de trabalho é a mesma que pode atrapalhar em algumas situações, como por exemplo uma aplicação que usa mais armazenamento do que capacidade computacional pode levar a necessidade de se agregar um *node* completo para aumentar esta capacidade, não apenas novos discos dependendo da arquitetura do orquestrador.
- Dependência de Fornecedor: em alguns ambientes hiperconvergentes é muito mais simples adicionar um novo *node* ou *storage* se for idêntico aos outros já agregados ao cluster, o que pode aumentar custos ou criar janelas de manutenção forçadas caso não seja possível escalar o ambiente com o mesmo hardware utilizado no início do projeto. (HARVEY, 2016)

2.2 PROXMOX VE (GNU AGPL, V3)

O ProxMox VE é uma plataforma de gerenciamento de servidores construída em código aberto para virtualização de sistemas. Integradas ao hipervisor, as tecnologias KVM e LXC, armazenamentos definidos por software e gerenciamento de rede, permitem que a plataforma entregue em um único ambiente um gerenciamento facilitado de máquinas virtuais completas (KVM) e contêineres (LXC). O ProxMox oferece recursos de alta disponibilidade e integração com ferramentas de recuperação de desastres, além da facilidade de acessá-lo apenas abrindo um site no navegador (PROXMOX PROJECT, 2020).

Com recursos completamente baseados em software, é uma ferramenta completa para virtualização, otimização de recursos e aumento de eficiência com um custo mínimo.

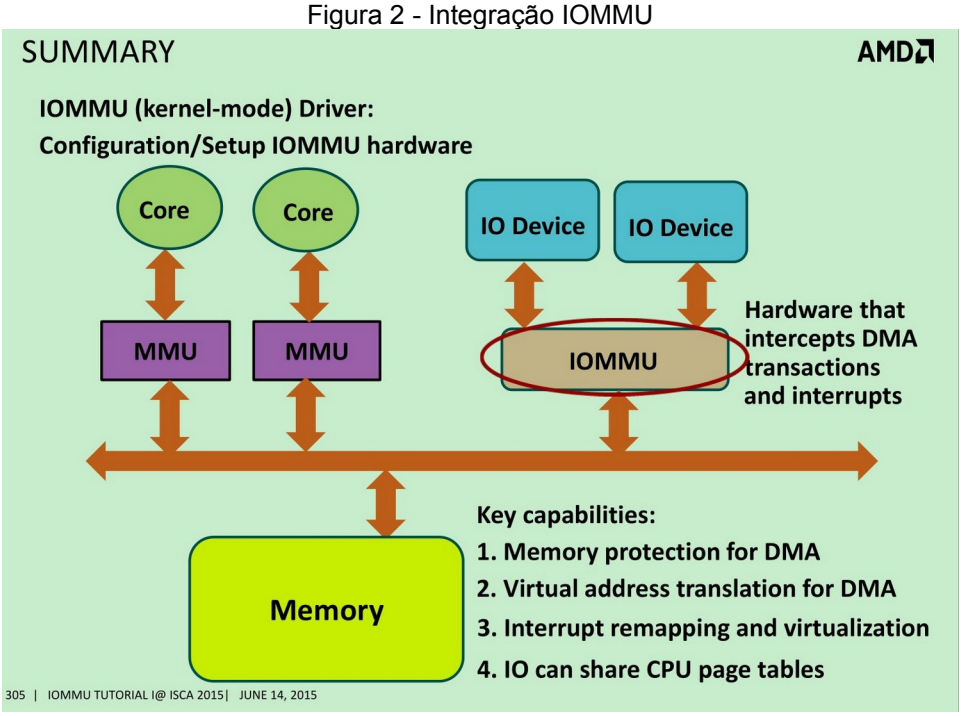
2.3 KVM/QEMU (GPL,V2)

O QEMU é um emulador e virtualizador de código aberto. Quando utilizado como emulador, é capaz de rodar sistemas operacionais completos e programas feitos especificamente para alguma arquitetura em uma máquina diferente (como uma placa ARM em um PC convencional), obtendo performance considerável. Quando utilizado como virtualizador, é capaz de entregar um desempenho quase nativo, executando o código convidado diretamente na CPU (*Central Processing Unit*) do hospedeiro.

KVM (*Kernel Virtual Machine*) é um módulo do *Kernel* Linux que permite que um programa em *userspace* utilize recursos de hardware do sistema hospedeiro. Hoje o KVM suporta entregar recursos dos processadores Intel e AMD, PPC 440, PPC 970, S/390, ARM e MIPS32 (KVM PROJECT, 2020).

Em conjunto, é possível entregar serviços com baixíssima latência através da pilha QEMU/KVM em um grande número de sistemas operacionais utilizando os recursos disponíveis no hardware com acesso direto. Fazendo uso da tecnologia IOMMU, é possível entregar o processamento de uma placa de vídeo a uma máquina virtual como se a mesma estivesse instalada diretamente neste sistema virtualizado, exemplificado na Figura 2.

Figura 2 - Integração IOMMU



Fonte: KEGEL et al. (2016, p. 305)

Traduzindo de forma simples o diagrama da figura 2, temos dois serviços de gerenciamento de unidades de memória, um deles acessando um barramento direto ao processador (core), e o outro gerenciando a entrada e saída de dispositivos, tendo na outra extremidade um barramento DMA (*Direct Memory Access*) com acesso direto a memória do sistema. Em um ambiente virtualizado, o serviço de IOMMU permite que um dispositivo ou barramento do sistema seja entregue exclusivamente a um host virtual.

2.4 LXC (GNU LGPLV2.1+)

O projeto LXC (*Linux Containers*) oferece uma interface em *userspace* para recursos de contenção do *Kernel* Linux. Por meio de uma poderosa API (*Application Programming Interface*) e ferramentas simples, é possível ao usuário criar e gerenciar facilmente contêineres de sistemas ou aplicações (LXC, 2020).

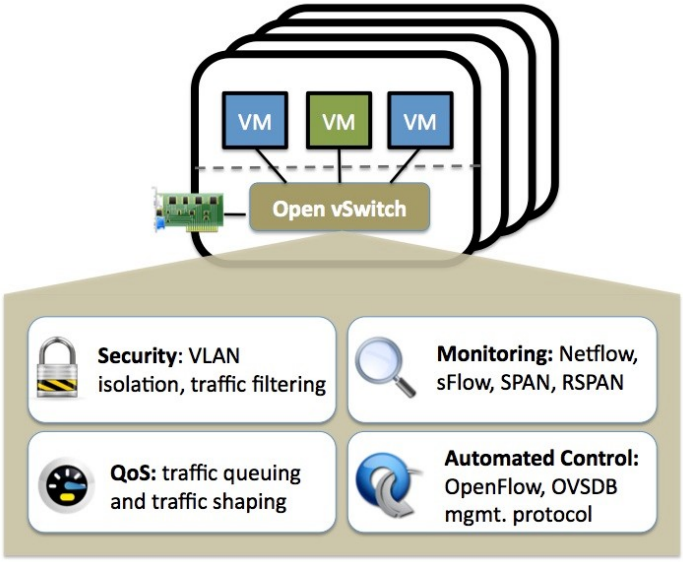
Podemos considerar o LXC como intermediário entre um chroot e uma máquina virtual completa, sendo este o objetivo do projeto: entregar um ambiente que seja o mais próximo possível de uma instalação completa padrão, sem que haja a necessidade de um *kernel* separado.

Destacamos que algumas ferramentas reunidas pelo projeto já existem no *kernel* linux, mas a demanda de uma implantação rápida gerou a necessidade de compilar tais ferramentas, como *namespaces* do *Kernel* (ipc, uts, mount, pid, rede e usuário), perfis de Apparmor e SELinux, políticas Seccomp, chroots (utilizando pivot_root) e CGroups (grupos de controle / limitadores de recursos)

2.5 OPEN VSWITCH (APACHE LICENSE 2.0)

O switch virtual multicamadas Open vSwitch foi projetado para permitir uma massiva automatização de rede através de extensões programáveis, suportando vários protocolos de gerenciamento (por exemplo NetFlow, sFlow, IPFIX, RSPAN, CLI, LACP, e 802.1ag). Sua definição de arquitetura na Figura 3 contempla os serviços citados, além de ter sido projetado para suportar a distribuição entre múltiplos servidores físicos de forma similar a projetos proprietários como da VMware e Cisco. (OPEN VSWITCH, 2020)

Figura 3 - Arquitetura Open vSwitch

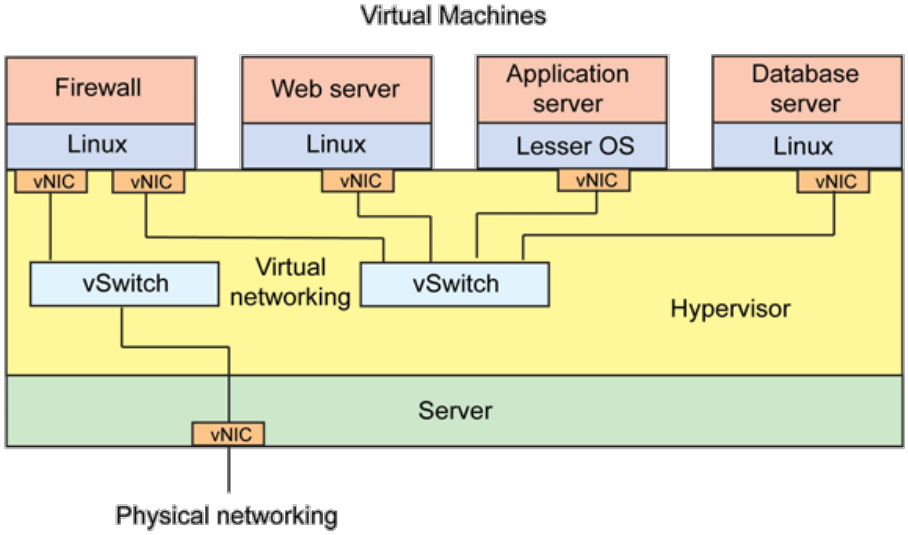


Fonte: OPEN VSWITCH (2020, p. 1)

Em qualquer orquestrador de virtualização do mercado, um dos principais recursos é a habilidade de fazer uma ponte de tráfego entre as VMs e contêineres com o mundo exterior. Em orquestradores baseados em Linux, era necessário se utilizar o switch de camada 2 já disponível no *Kernel* (*Linux Bridge*), que é rápido e confiável. Porém, com os recursos de camada 3 do Open vSwitch, novas possibilidades foram entregues para atender à realidade contemporânea de ambientes de virtualização, como no exemplo da Figura 4.

Hoje, na implantação de um *cluster* de vários servidores, utilizar essa pilha de camada 2 não é adequado, pois com um número vasto de máquinas virtuais e contêineres as mudanças no ambiente são frequentes: VMs sendo criadas e removidas, avançando e retrocedendo no tempo, alterações em seus ambientes de rede lógicos e em seu tamanho e localização de pontos de armazenamento.

Figura 4 - Roteamento com Switches Virtuais



Fonte: OPEN VSWITCH (2020, p. 4)

O Open vSwitch oferece suporte a vários recursos de camada 3 que permitem que um sistema de gerenciamento de rede responda e se adapte conforme o ambiente muda, utilizando protocolos como NetFlow, IPFIX e sFLOW e acompanhando a dinâmica dos sistemas conforme suas mutações.

Considerando a necessidade de se entregar alta disponibilidade nos serviços, essa tecnologia foi muito importante, pois anteriormente nas máquinas com VMware a segurança, separação de tráfego e monitoramento eram feitos diretamente nos switches e roteadores físicos, gerando uma camada adicional na manutenção e manipulação dos serviços virtualizados.

Em adição, o projeto de migração objetivava entregar um serviço seguro de alta disponibilidade, possibilitando deixar transparente aos equipamentos de rede o gerenciamento do tráfego.

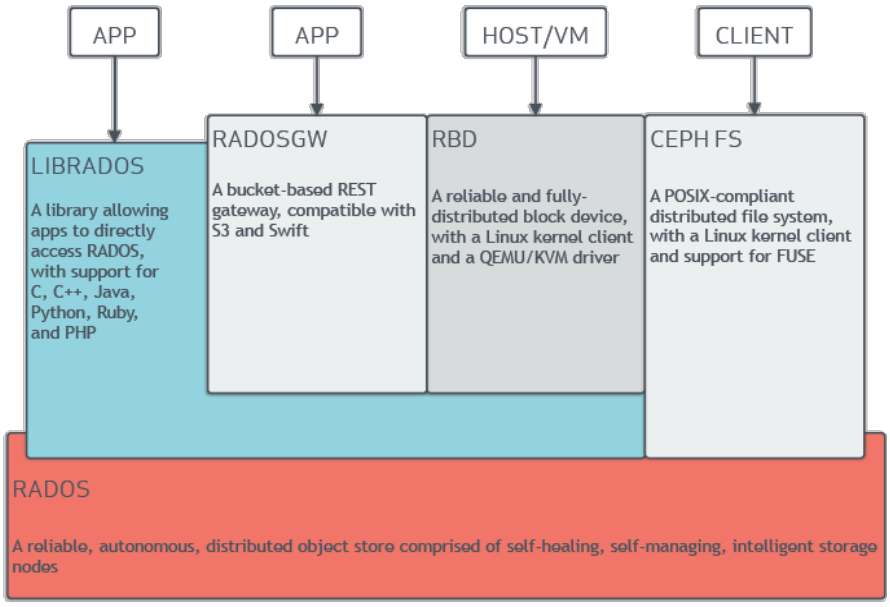
Foi possível utilizar os serviços diretamente no switch de maior capacidade do campus, com mais banda disponível tanto para tráfego quanto para gerência. A separação lógica por vlans ofereceu mais uma camada de separação de tráfego entre os serviços públicos e de gerência.

2.6 CEPH (GNU LGPL, V2.1)

Complementando as descrições da ferramenta contidas na introdução desta pesquisa, o projeto Ceph é um sistema de armazenamento definido por software (SDS) de código aberto, distribuído e escalável, oferecendo armazenamento por

blocos, objetos ou arquivos. Por meio do uso do algoritmo de replicação controlada sob um *hash* escalável (CRUSH), o Ceph elimina a necessidade de metadados centralizados, distribuindo a carga por todos os nós do *cluster*. A arquitetura de funcionamento do Ceph está exemplificado na Figura 5.

Figura 5 - Arquitetura do Ceph



Fonte: CEPH (2020b, p. 1)

O Ceph é uma solução SDS pura, permitindo que sua utilização seja livre para execução em hardware comum sem dependência de fabricantes, desde que dimensionado corretamente para oferecer as garantias sobre consistência dos dados. Este conceito é um grande salto para as tecnologias de armazenamento, que normalmente sofrem de dependências restritas a fornecedores e fabricantes (FISK, 2017)

Embora vários outros projetos de código aberto ofereçam serviços de armazenamento, o Ceph foi um dos poucos capazes de oferecer tal escalabilidade e resiliência sem a exigência de hardwares ou softwares proprietários.

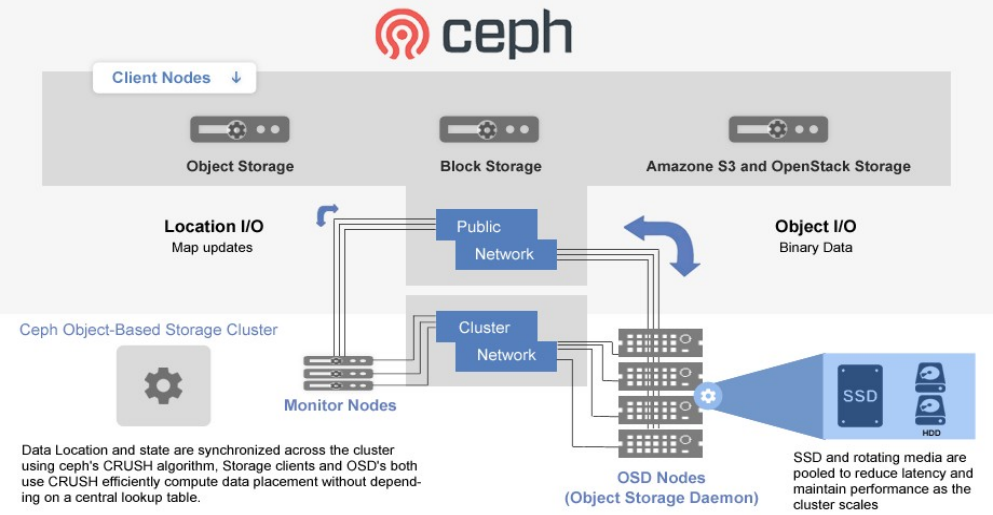
2.6.1 Recursos de Armazenamento

O Ceph fornece três tipos de armazenamento, sendo o bloco via RADOS Block Devices (RDB), o arquivo via Ceph Filesystem (CephFS ou BlueStore), e o objeto via o *gateway Reliable Autonomous Distributed Object Store* (RADOS), que

oferece compatibilidade com serviços simples de armazenamento (*Simple Storage Service* (S3)) e Switf (CEPH, 2020a).

Um diagrama sobre a arquitetura de armazenamento e sua topologia de rede básica está representado na Figura 6.

Figura 6 - Arquitetura do cluster e diagrama de rede resumido



Fonte: INNOSTORE (2020, p. 2)

Nesta figura 6 temos exemplos dos três tipos de armazenamento citados acima, além de uma exemplificação de como a localização e o estado dos dados são sincronizados em todo o cluster usando o algoritmo CRUSH do Ceph, em que os clientes e OSDs usam o CRUSH para calcular a localização dos dados de maneira eficiente sem depender de uma tabela de pesquisa central. Também temos a direita um exemplo em que SSDs e discos rígidos são agrupados para reduzir a latência e manter o desempenho conforme o cluster aumenta.

2.6.2 Object Storage

O *object storage* é uma arquitetura de armazenamento de dados que gerencia dados como objetos. Por sua vez, um sistema de arquivos gerencia os dados como uma hierarquia de arquivos, enquanto *block storages* gerenciam os dados divididos em blocos separados em trilhas e setores.

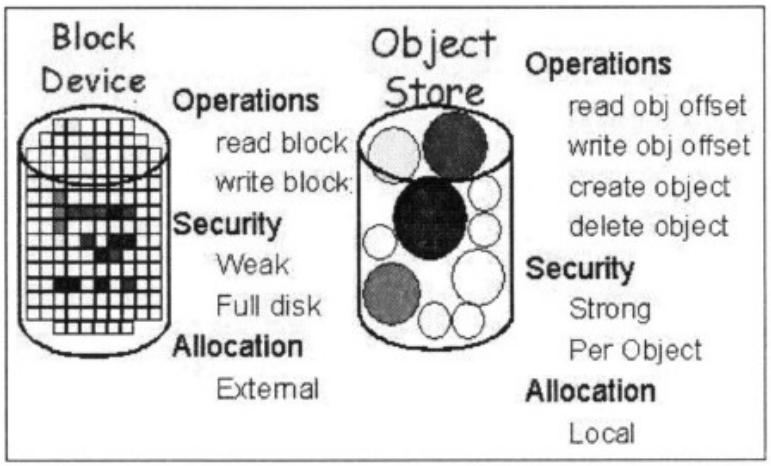
Cada objeto inclui tipicamente os próprios dados, uma quantidade variável de metadados e um identificador exclusivo, movendo algumas funcionalidades de baixo nível como gerenciamento de espaço para o dispositivo de armazenamento em si,

acessando-o através de uma interface padrão de objetos e não por camadas superiores de software, como em um sistema de arquivos (FACTOR et al., 2005).

O armazenamento de objetos pode ser implementado em vários níveis, incluindo o nível do dispositivo (dispositivo de armazenamento do objeto), o nível do sistema e o nível da interface. Assim, usuários acessam e manipulam os dados executando operações como criação de objetos, leitura / escrita em uma posição lógica no objeto, e remoção dos objetos criados. Somando a isso todas estas transações levam uma credencial, o que garante a possibilidade do *object store* verificar se as requisições possuem uma credencial válida, assegurando diferentes níveis de acesso com segurança e granularidade.

A Figura 7 mostra um comparativo entre um dispositivo em bloco e um OSD, demonstrando uma dinâmica diferente do modelo tradicional com alguns novos recursos de alocação e segurança.

Figura 7 - Block Device vs. um Object Storage Device (OSD)



Fonte: FACTOR et al. (2005, p. 1)

No dispositivo em bloco (*block device*), podemos ver que suas operações são limitadas a leitura e escrita nos blocos, sua segurança é fraca e apenas relativa ao disco todo, e sua alocação feita externamente; Enquanto que no armazenamento em objetos (*object store*) vemos que além das operações de leitura e escrita, a criação e remoção de objetos é possível, sua segurança é forte e granularizada por objeto, e a alocação feita localmente.

Com esses conceitos de objetos, é importante lembrar que o Ceph maximiza a separação entre o gerenciamento dos metadados de arquivos e o armazenamento dos dados dos arquivos. Operações de metadados (abrir, renomear, etc.) são

coletivamente gerenciadas por um *cluster* de servidores de metadados, enquanto os clientes interagem diretamente com os OSDs na leitura / escrita das informações (WEIL et al., 2006a).

2.6.3 Sistema de Arquivos

O planejamento de sistemas de arquivos nas últimas décadas foi altamente influenciado pelo sistema de arquivos Unix e o FFS (*Fast File System*) do BSD (*Berkeley Software Distribution*) Unix. A interface e comportamento desses sistemas de arquivos formaram a base para o padrão POSIX (*Portable Operating system Interface*), atualmente seguido por grande parte dos sistemas operacionais e aplicações.

Com a necessidade de entregar dados alocados em discos físicos locais, esses sistemas foram projetados para trabalhar alinhados com um número fixo de setores, blocos e cilindros, e os discos à época eram lentos na busca de posições por si só. Em contrapartida, quando na posição correta a leitura dos dados era relativamente rápida, forçando os desenvolvedores a considerarem formas de alocação e armazenamento que não causassem latência ou lentidão ao sistema, limitando a alocação dos dados em posições conhecidas ou próximas, como no mesmo cilindro ou região, inclusive mantendo metadados o mais próximo possível (WEIL et al., 2004).

Atualmente, vários sistemas de arquivos ainda apresentam limitações de projeto por conta da arquitetura dos discos, principalmente adaptando a estrutura de metadados, pois apesar dos tamanhos relativamente maiores, as limitações físicas ainda forçam a utilização de blocos de tamanho reduzido para a eficiente alocação de arquivos pequenos (por exemplo 4Kb). Esses mesmos sistemas, como XFS e ext4, mantêm um *log* de alterações de metadados chamado de *journal*, entregando uma camada adicional de segurança e possível recuperação de dados em uma situação crítica.

2.6.3.1 Sistemas de Arquivos Cliente – Servidor

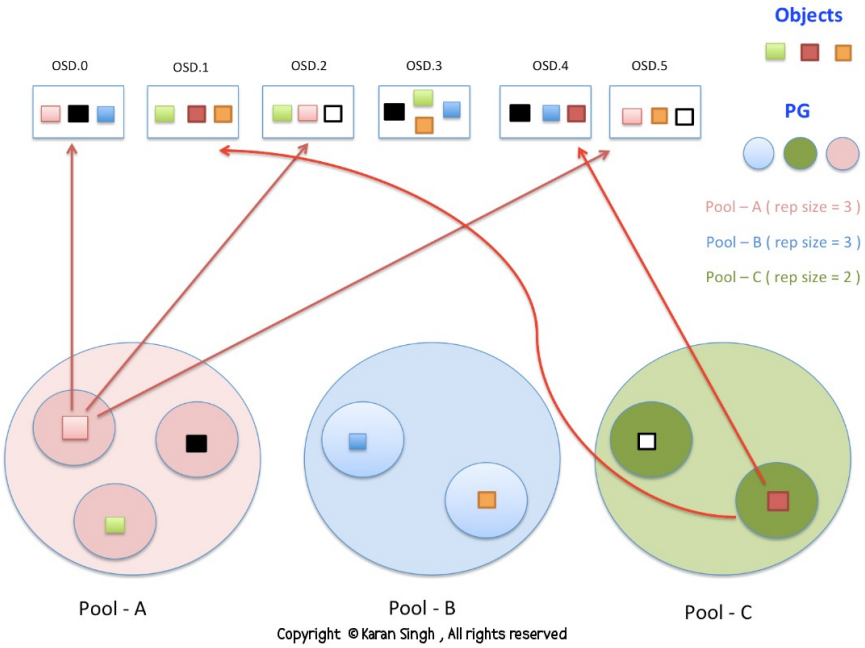
Com a presença da informática no mundo corporativo, a necessidade de sistemas de arquivos compartilhados teve um salto considerável, levando várias soluções proprietárias a criarem seus próprios sistemas. Porém, os mais utilizados e ainda em larga adoção são os sistemas NFS (*Network File System*) e CIFS (*Common Internet File System*), (ALMEIDA, 2006) que permitem que um servidor central entregue pontos de montagem locais a sistemas operacionais remotos, os quais mapeiam esses recursos localmente em seu próprio *namespace*.

Sistemas de arquivo centralizados facilitaram o surgimento de equipamentos e sistemas especializados, popularmente conhecidos como NAS (*Network Attached Storage*) que estavam limitados ao modelo cliente-servidor no qual a escalabilidade se torna um fator limitante, pois todos os pontos de montagem geralmente se concentram em um único servidor. Ao mesmo tempo, segregar dados em diferentes pontos de montagem ou implantando incorretamente uma SAN (*Storage Area Network*) pode trazer mais problemas aos *sysadmins* quando os dados começarem a expandir ou se duplicarem nos arranjos (WEIL, 2007).

2.6.3.2 Armazenamento de dados no Ceph

Com esses conceitos podemos resumir como é feito o armazenamento dos dados no *cluster* Ceph. Como exemplificado na Figura 8, devemos considerar os objetos a menor porção de dados armazenada, contida em um *Placement Group* (PG) que, por sua vez, está contido em uma *pool* que tem sua taxa de replicação distribuída nos OSDs definidos.

Figura 8 - Armazenamento de dados no cluster Ceph



Fonte: CEPH (2014, p. 1)

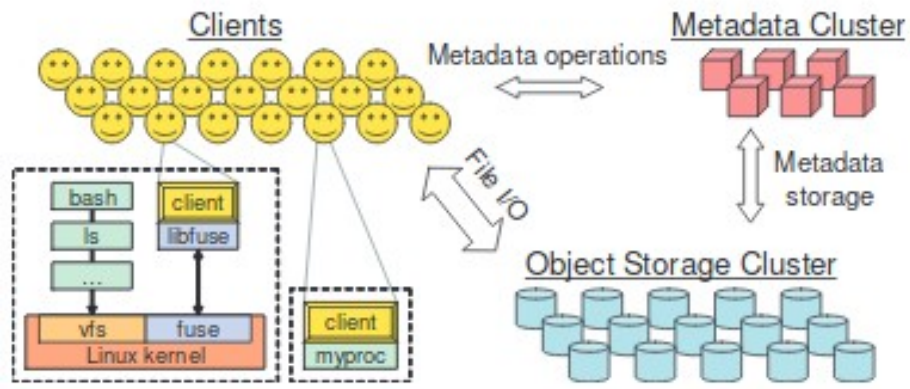
Para um melhor dimensionamento do arranjo, é necessário fazer um cálculo de relacionamento prévio de quantos PGs serão adequados à taxa de replicação escolhida e os OSDs dos monitores. A fórmula sugerida é PGs = (OSDs * 100) / Réplicas, porém há ferramentas online disponíveis para auxiliar no dimensionamento inicial, como a *ceph-calculator* (FLORIAN, 2018).

2.6.4 RADOS

O utilitário RADOS (*Reliable Autonomic distributed Object Storage*) fornece um serviço de armazenamento de objetos escalável e confiável sem comprometimentos com desempenho. A arquitetura do RADOS se utiliza da manipulação de um mapa disponibilizado pelos monitores do *cluster* (*cluster map*), que especifica quais OSDs estão inclusos no arranjo e de forma inteligente distribui todos os dados entre os dispositivos, de acordo com o dimensionamento feito na implantação

Abaixo na figura 9 é possível ver a arquitetura do sistema em que os clientes realizam leitura e escrita comunicando-se diretamente com o cluster de armazenamento (OSDs), que pode estar ligado diretamente através de um ponto de montagem em *userspace*, ou em uma instância cliente.

Figura 9 - Diagrama operacional do RADOS



Fonte: WEIL (2007, p. 18)

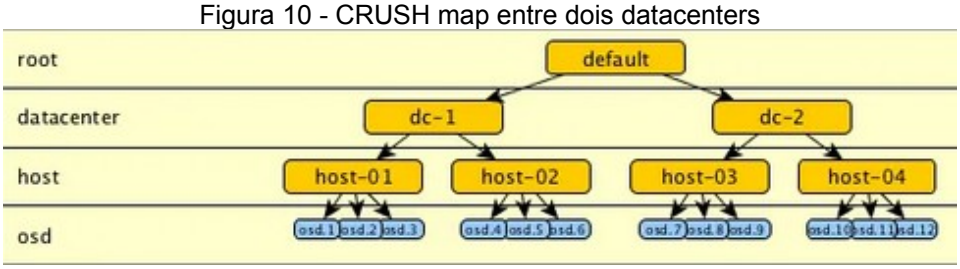
2.6.5 CRUSH

Como o CRUSH é um algoritmo, o posicionamento de dados é calculado e pode ser dimensionado para centenas de petabytes sem o risco de gargalos e pontos únicos de falha, ao contrário do que ocorre quando o posicionamento de dados é baseado em pesquisas de tabela. Os clientes também formam conexões diretas com o servidor que armazena os dados solicitados e, portanto, não há gargalo centralizado no caminho dos dados. (WEIL et al., 2006b)

A distribuição pseudoaleatória dos dados através do algoritmo CRUSH (*Controlled Replication Under Scalable Hashing*) permite distribuir de forma eficiente e robusta réplicas dos objetos através de um *cluster* heterogêneo e estruturado, trazendo duas vantagens chave: é completamente distribuído de modo que qualquer parte alocada em um sistema possa ter sua posição calculada rapidamente; e os poucos metadados necessários são principalmente estáticos, mudando apenas quando os dispositivos são adicionados ou removidos (WEIL et al., 2006b).

Existem várias possibilidades de organização dos dados, gerando os mais diversos *CRUSH maps*. A Figura 10 ilustra a hierarquia de um *CRUSH map* simples entre dois *datacenters,* sendo o dc-1 e dc-2 mapas contidos no mapa padrão principal, mas com informações replicadas entre outros mapas permitindo o cálculo de posicionamento das informações por qualquer origem de solicitação feita ao cluster.

Figura 10 - CRUSH map entre dois datacenters



Fonte: CEPH (2015, p. 1)

2.6.6 Escalabilidade

Diferente dos sistemas cliente-servidor citados anteriormente, uma das grandes vantagens na utilização do Ceph como *object storage* está em sua escalabilidade. O conhecimento da distribuição dos dados encapsulada no *cluster map* permite que o RADOS distribua a gerência da redundância dos dados, faça a detecção e recuperação de falhas aos OSDs que venham a comprometer o *cluster*, e ofereçam um sistema de balanceamento e replicação dos dados de forma que, na falha ou na adição de um *node*, os dados estejam armazenados e replicados de forma segura, com baixíssima indisponibilidade ao sistema que estiver consumindo esse *storage*. (WEIL et al., 2007)

Chamados de monitores, os *nodes* membros do *cluster* são coletivamente responsáveis por gerenciar o sistema de armazenamento, guardando cópias primárias do *cluster map* e anotando as mudanças periódicas nos estados dos OSDs (por ex. falha ou recuperação dos dispositivos)

Comparado com outros sistemas de arquivos baseados em *object storage* que apenas substituem longas listas de blocos por arquivo alocado por listas menores de objetos, o Ceph elimina o uso de listas de alocação completamente. Em vez disso, os dados são distribuídos em objetos com nomenclatura previsível e a alocação dos mesmos é feita pelo algorítimo CRUSH. Isso permite que a posição do objeto que possui o conteúdo desejado seja prevista pelo utilizador de forma matemática sem a necessidade do armazenamento de uma lista com tais posições (WEIL et al., 2006ª).

Todas essas funcionalidades e recursos permitem que, com o planejamento adequado, toda alteração de soma ou redução de *nodes* esteja prevista no *cluster*, de forma que esses procedimentos se tornem simples e sem indisponibilidade do *storage*.

No projeto do ProxMox, para se fazer uso das funcionalidades de alta disponibilidade, recomenda-se o mínimo de três *nodes* para atender aos conceitos de quorum, já que com apenas dois *nodes* um sempre elegeria o outro como mestre, causando um *loop* na relação de peso entre eles deixando o sistema indisponível.

2.6.7 Automação de Implantação

É possível fazer uma implantação do Ceph de forma automatizada utilizando-se da ferramenta ceph-deploy em um ambiente mínimo, em que apenas o acesso SSH entre os servidores é necessário. Toda a configuração inicial feita no primeiro *node* é carregada para o segundo, criando um novo monitor e já disponibilizando ao arranjo para a adição de OSDs (FISK, 2017).

Além do próprio ceph-deploy, há vários *scripts* criados pela comunidade, utilizando-se de várias aplicações como docker, vagrant, ansible e orquestradores para ferramentas específicas, como o projeto rook.io de gerenciamento de *storage* para Kubernetes capaz de gerenciar *storages* de várias tecnologias além do Ceph como Cassandra, NFS, EdgeFS, YugabyteDB e CockroachDB (ROOK, 2020).

2.6.8 Deduplicação

A técnica de deduplicação em sistemas de armazenamento serve para eliminar cópias de dados repetidos, objetivando eliminar informações desnecessárias para poupar recursos de armazenamento e garantir a integridade dos dados.

Sobre a implementação do Ceph, há algumas implicações, pois os metadados dos objetos são tratados de forma diferente em relação a outros sistemas de arquivos. Como um dos recursos do Ceph é justamente a replicação dos dados, a necessidade de aplicar deduplicação nos dados armazenados seria específica de cada projeto. Exemplificando: um serviço sobre S3 no qual a maioria dos acessos aos dados é apenas para leitura, como em um e-commerce no qual após a criação (escrita) de um produto, a grande maioria dos acessos será apenas para leitura dessas informações pelos usuários navegando na plataforma.

2.6.9 Otimização de Armazenamento

A arquitetura do Ceph garante que o conhecimento dos *nodes* sobre a topologia do *cluster* esteja disponível através do *cluster map*, formado por cinco mapas distintos que contêm informações sobre as alterações ocorridas nos monitores, replicadas integralmente a todos os membros do *cluster*, a saber: (CEPH, 2020a)

1. *Monitor Map*: contém o id do *cluster*, a posição, endereço, nome e a porta de cada monitor. Também indica a época atual, quando o mapa foi criado e a última vez em que foi alterado.
2. *OSD map*: contém o id do *cluster*, quando o mapa foi criado e modificado pela última vez, uma lista de *pools*, tamanhos de réplicas, números dos PG, uma lista de OSDs e seu status (por exemplo, *up, in*)
3. *PG map*: contém a versão do PG, seu carimbo de data / hora, a última época do OSD map, as proporções completas e detalhes de cada PG, como o id do PG, o estado do PG (por exemplo , *active* + *clean*) e estatísticas de uso de dados para cada *pool*.
4. *CRUSH map*: contém uma lista de dispositivos de armazenamento, a hierarquia de domínio de falha (por exemplo, dispositivo, *host*, *rack*, posição, sala, etc.) e regras para atravessar a hierarquia ao armazenar dados.
5. *MDS map*: contém a época do mapa MDS (*Metadata Server*) atual, quando o mapa foi criado e a última vez em que foi alterado. Ele também contém o *pool* para armazenar metadados, uma lista de servidores de metadados e quais servidores de metadados estão *up* e *in*.

Com essas informações, o serviço nativo de balanceamento é capaz de otimizar a disposição dos PGs através dos OSDs de forma que a carga do sistema fique transparente aos usuários do sistema de arquivos, podendo ser customizada de acordo com pesos e prioridades das zonas e *pools* disponíveis no *cluster*.

2.6.10 Integrações da Plataforma

Com a popularidade da ferramenta surgiram várias possibilidades de integração com o Ceph, tanto para monitoria quanto para implantação do serviço. Abaixo serão citadas duas integrações com ferramentas de grande presença no

mercado atual.

2.6.10.1 Kubernetes / rook.io

Kubernetes é um orquestrador de contêineres de código aberto para automatizar a implantação, escalonamento e gerenciamento de aplicativos em contêineres, hospedado pela Cloud Native Computing Foundation (CNCF) (KUBERNETES, 2020).

Originalmente projeto do Google, o Kubernetes teve seu código aberto em 2014, trazendo à comunidade quinze anos de experiência da empresa rodando aplicações em escalas consideráveis. Hoje referência para aplicações com desenvolvimento dinâmico em várias camadas de implementação e testes, tornou-se popular por sua velocidade e praticidade, com a ferramenta integradora rook.io sendo utilizada para o gerenciamento de *clusters* Ceph na implementação de contêineres (ROOK, 2020).

2.6.10.2 OpenStack

OpenStack é um sistema operacional em nuvem que controla grandes conjuntos de recursos de computação, armazenamento e rede em um *datacenter*, todos gerenciados e provisionados por meio de APIs com mecanismos de autenticação comuns. Além da funcionalidade padrão de infraestrutura como serviço, componentes adicionais fornecem orquestração, gerenciamento de falhas e gerenciamento de serviços para garantir alta disponibilidade dos aplicativos do usuário (OPENSTACK, 2020).

O Ceph seria apropriado para integração ao OpenStack, segundo o estudo feito por (KTENZER, 2016), com as ferramentas integradas Cinder (*block storage*), Glance (imagens) e Nova (discos virtuais para VMs), além das outras ferramentas completas Switf e Manila, citando casos de sucesso como o CERN com mais de 10.000 VMs implantadas (DANIEL, 2009).

2.6.11 SDSs Similares

O CEPH não é o único storage definido por software, e no mercado existem alternativas com diversas abordagens. Para critério comparativo algumas ferramentas para uso em larga escala serão abordadas neste capítulo para referência.

2.6.11.1 Gluster

O Gluster é um sistema de arquivos capaz de agregar recursos de armazenamento de disco de múltiplos servidores em um único *namespace*, ou seja, entregar um ponto de montagem transparente ao usuário sendo gerenciado virtualmente sobre uma organização “clusterizada” em hardware de armazenamento (GLUSTER, 2020).

Para não depender de módulos no *kernel*, o GlusterFS utiliza um sistema de arquivos em *userspace*, fazendo uso da tecnologia FUSE (*File System in Userspace*). Essa tecnologia foi desenvolvida devido às dificuldades de implementação de sistemas de arquivos em *userspace*. Apesar do FUSE em si ser um módulo do *kernel*, ele suporta a interação entre VFS (*Virtual File Systems*) e aplicações não privilegiadas, servindo uma API que pode ser consumida e acessada em *userspace*.

Em estudo comparativo entre as ferramentas, considerando a diferença entre o Ceph basear-se em *object storage* e o Gluster em *block storage*, a IOP Science mostrou que é possível que um arranjo sobre Gluster tenha superado a performance do Ceph, porém apresentou algumas instabilidades que resultaram em perda parcial ou total dos dados, o que deve ser levado em conta em um projeto com escalabilidade em tempo mais curto que a capacidade de cura e replicação do sistema de arquivos (DONVITO; MARZULLI; DIACONO, 2014).

2.6.11.2 Lizard

Outro sistema de armazenamento definido por software bastante competente é o LizardFS. Com várias similaridades ao Ceph, a ferramenta é escalável, tolerante a falhas e focado em alta disponibilidade, permitindo que o usuário combine espaços

em disco distribuídos fisicamente em diversos servidores em um *namespace* centralizado, capaz de atender sistemas *Unix-like* e Windows como qualquer outro sistema de arquivos (LIZARD, 2020).

As maiores similaridades ao Ceph são as replicações dos dados através dos servidores e discos e a capacidade de agregar novos ativos sem nenhum *downtime*, tendo sua escalabilidade e confiabilidade garantidas pelas replicações e autogerenciamento do sistema.

É possível implantar o Lizard em diversas combinações de hardware sem dependência de fornecedores, pois o auto-balanceamento de recursos é feito de forma inteligente e contínua, facilitando também a remoção de equipamentos do arranjo.

2.6.11.3 ZFS

O sistema de arquivos ZFS, originalmente desenvolvido pela Sun (atualmente Oracle) foi considerado um sistema incomum à época de seu lançamento, pois diferentemente dos sistemas de arquivos até então, unificava as atribuições do gerenciador de volumes e do sistema de arquivos, sendo capaz de garantir que anomalias ocorridas como falhas físicas, de sistema operacional ou eventos de corrupção de dados (OPENZFS, 2020).

É também possível montar arquiteturas e arranjos de hardware para replicação em alta disponibilidade sobre ZFS, mas somente com muitos recursos de rede e memória RAM disponível, o que gera um custo maior para utilizar a ferramenta com todas as suas vantagens.

Separado em três camadas (*Storage Pool Alocator, Data Management Unit e Dataset Layer*), o ZFS é capaz de gerenciar *pools* nas quais os sistemas de arquivos entregues ao usuário são verificados em tempo real, garantindo a integridade e a disponibilidade dos dados distribuídos no arranjo.

Por ser um sistema utilizado em larga escala e um projeto adulto capaz de replicar dados entre sistemas de arquivos, o ZFS pode servir como base para constatarmos a evolução desse sistema, desenvolvido anos antes de várias tecnologias disponíveis atualmente cujas cargas de processamento podem ser diluídas em outras camadas.

2.6.12 Complicações

Apesar de qualquer sistema de produção estar sujeito à ocorrência de incidentes, (KRAL, 2011), o Ceph, por trabalhar focado em alta disponibilidade e replicação dos dados, raramente apresentará uma falha capaz de derrubar completamente o ambiente, dadas condições normais de segurança e proteção em um *datacenter*.

Além de problemas inerentes de hardware ou fenômenos computacionais, disponibilizar um cluster demanda de um projeto conciso que leve em consideração todos os recursos do ambiente e as características dos serviços que serão hospedados, questionando inclusive se este tipo de arquitetura será realmente o mais adequado. (CEPH, 2020)

2.6.12.1 Auto-monitoramento e cura

Quando na console do Ceph, é possível ver a situação de todos os PGs através de comandos como `ceph -w` ou `ceph -s`. Como citado no item 2.6.9 os PGs podem ter um ou mais estados, dentre eles *repair, recovering, forced_recovery e scrubbing*, por exemplo, são estados em que o próprio Ceph detectou alguma necessidade de cura e já está trabalhando para que todos os PGs mantenham a integridade dos dados (CEPH, 2020c).

2.6.12.2 Split Brain

Um dos fenômenos possíveis foi citado no item 2.6.6 na relação de quorum, na qual é necessário um número sempre ímpar de *nodes* para evitar o *split-brain*, termo baseado em uma síndrome neurológica humana, que em termos computacionais, traria inconsistência nos dados, pois dois conjuntos de informações separados se sobrepõem por alguma indisponibilidade de recursos ou dissintonia do sistema.

Alguns usuários do ProxMox já experimentaram em laboratório forçar configurações de quorum para montar um arranjo entre dois servidores, mas os resultados obtidos são apenas a replicação dos dados, pois caso um dos servidores caia é preciso uma ação manual para reaver acesso ao sistema, impraticável em

ambiente de produção.

2.6.12.3 Rede e latência

Uma arquitetura de cluster é baseada na agregação de servidores e equipamentos, portanto é crucial que a comunicação entre eles esteja dimensionada e disponível de acordo com a necessidade do arranjo.

Muitos dos problemas de alinhamento e desempenho do cluster Ceph são originadas em configurações de rede inadequadas ou tráfego misto. É importante que toda a comunicação dos monitores Ceph sejam feitas por uma rede exclusiva, resiliente, redundante e sem latência para que o alinhamento dos discos (OSDs) seja rápido e constante.

Como o custo da solução de problemas de desempenho em um pequeno cluster provavelmente excede o custo das novas unidades de disco, é necessário otimizar o planejamento do projeto evitando a tentação de sobrecarregar as unidades de armazenamento, adquirindo discos em maior quantidade do que exclusivamente de alta densidade.

Dependendo dos serviços, por exemplo, pode ser mais vantajoso se utilizar de SSDs para o armazenamento de metadados e agilizar a entrega das informações dos OSDs do que armazenar os dados em SSDs contando apenas com seu alto desempenho individual de leitura e escrita.

2.6.13 Comunidade

As pesquisas e implementações iniciais do projeto do Ceph foram realizadas no Centro de Pesquisas em Sistemas de Armazenamento (*Storage Systems Research Center*) da Universidade da Califórnia em Santa Cruz pelo pesquisador Sage Weil. Defendida em 2007, a tese em Ciência da Computação foi o resultado de vários artigos publicados nos anos anteriores com coautoria de outros grandes especialistas em armazenamento. (WEIL, 2007)

Para garantir que a tecnologia continuasse sem dependência de fabricantes e grandes empresas, Sage alinhou o projeto com a Linux Foundation e transformou o projeto Ceph também em uma fundação, garantindo a continuidade das pesquisas e

a manutenção de sua disponibilidade a toda a comunidade, hoje atuante com conferências e apresentações praticamente semanais sobre a ferramenta e suas novas funcionalidades e versões.

Sage também fez várias doações, algumas milionárias, ao centro de pesquisas onde estudou e desenvolveu o projeto, mantendo a SSRC como centro de referência em pesquisas na área de armazenamento.

Além de um projeto desta dimensão e impacto na comunidade *open source*, serve como um ótimo caso de sucesso de uma ferramenta desenvolvida com ouvidos voltados para as necessidades da comunidade, e que tomou um caminho diferente, não sendo fagocitada por uma grande empresa mas se tornando uma fundação de pesquisa ativa, com eventos frequentes e melhorias sucessivas.

3 DESENVOLVIMENTO

O capítulo três aborda a experiência do estudo e implantação das ferramentas discutidas, levando em consideração o ambiente da Unesp Franca e suas limitações relativas ao parque de TI.

3.1 SOBRE A FCHS

Segundo informações contidas no website da Unesp Franca - Faculdade de Ciências Humanas e Sociais (FCHS, 2020), a unidade oferece cursos de graduação em História, Serviço Social, Direito e Relações Internacionais. Além dos cursos de graduação, conta com programas de mestrado e doutorado, grupos de extensão e outros projetos que atendem alunos e comunidade.

Fundada em 1962 como “Faculdade de Filosofia, Ciências e Letras de Franca”, foi incorporada pela UNESP em 1976. Atualmente, a unidade conta com aproximadamente 150 servidores, 90 professores e 1900 alunos de graduação e pós-graduação.

3.2 LABORATÓRIO DE IMPLANTAÇÃO E AMBIENTE ANTERIOR

No quadro organizacional da FCHS, a DTI (Diretoria Técnica de Informática) é responsável por um parque de 400 computadores, 191 telefones VoIP, 138 ativos de rede entre switches, câmeras, servidores, *access points* e periféricos diversos, além de oferecer serviços digitais com 23 sites de acesso público, portais e ferramentas de gestão corporativa (UNESP, 2020b).

Com sérias restrições orçamentárias, toda e qualquer atualização do parque tecnológico da FCHS passa por vários processos burocráticos de aquisição, justificativas, custos, licitações e outros entraves que acabam consumando o *backbone* de infraestrutura com equipamentos que obrigatoriamente devem rodar por muitos anos sem melhorias, ou são apenas atualizados quando atingem situações extremas de indisponibilidade.

Além disso, a universidade é signatária do protocolo de Brasília (UNESP, 2011), que prevê que ferramentas ou serviços disponibilizados no parque de informática sejam preferencialmente baseados em software de código aberto, salvas

restrições onde não seja possível obter resultados semelhantes aos das ferramentas proprietárias.

No momento inicial de implantação dos servidores virtualizados no campus, a instância escolhida foi a VMware ESXi. Entretanto, as dificuldades e custos adicionais vivenciados durante o início da implantação motivaram a busca por alternativas mais alinhadas à realidade da universidade, resultando no arranjo de hiperconvergência descrito anteriormente.

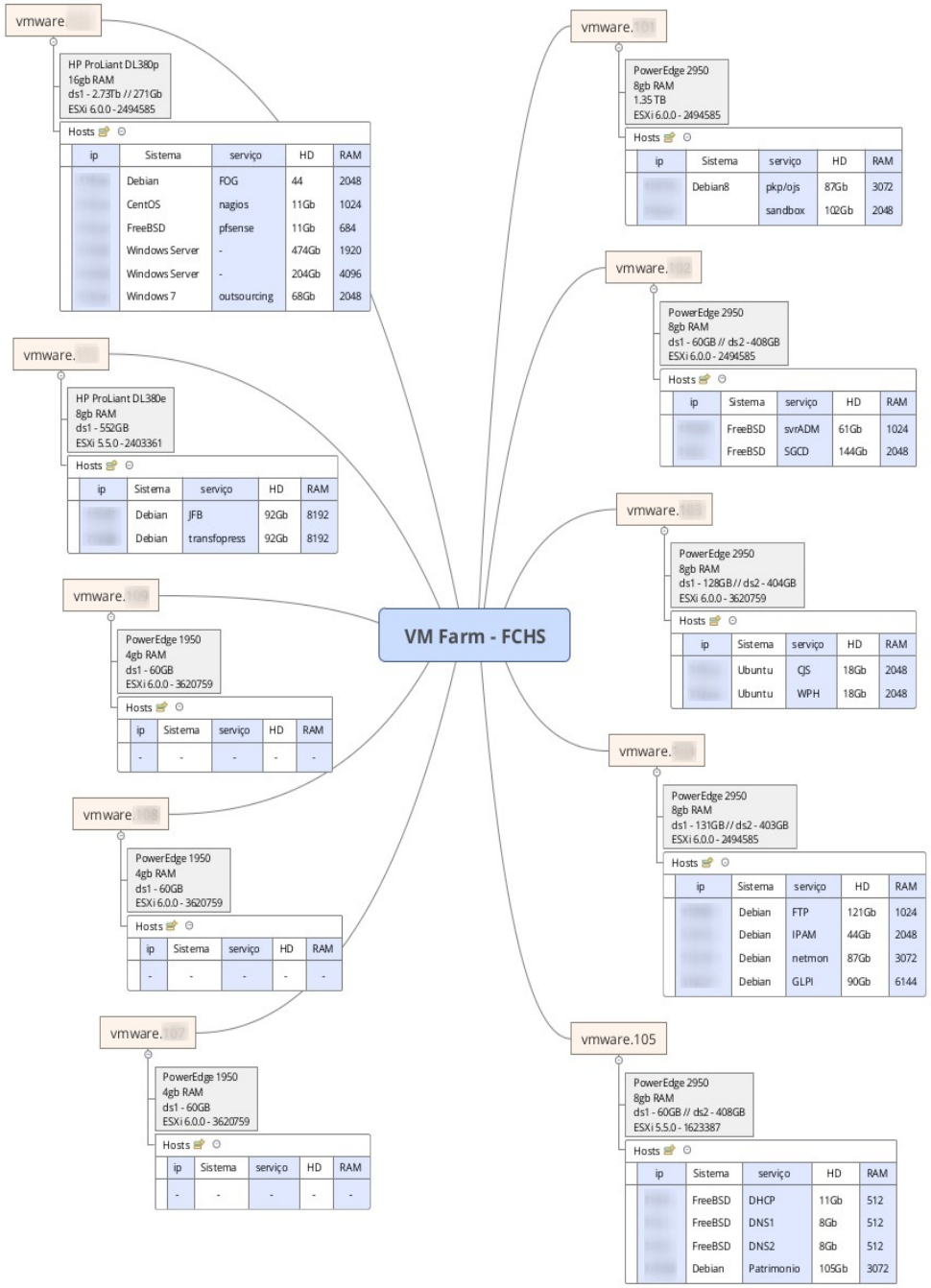
Antes de caminhar para a hiperconvergência, o parque da FCHS mantinha serviços instalados diretamente em máquinas físicas, algumas superdimensionadas para os serviços que ofereciam. A primeira tentativa de virtualização foi feita utilizando o sistema proprietário VMware ESXi, que para o uso em servidores isolados possui uma licença gratuita, porém vários serviços de monitoramento ou entrega de recursos físicos como LACP (*Link Aggregation Control Protocol*) demandam licenciamento. Sendo assim, não era possível fazer uso completo dos recursos, além de ser necessário lidar com a dependência dos clientes de gerenciamento instaláveis apenas em algumas versões e sistemas operacionais.

3.3 ESTADO INICIAL

Inicialmente, o ambiente virtualizado contava com servidores isolados, cada um com seu arranjo de VMs e sem compartilhamento de recursos ou alta disponibilidade, exemplificados na Figura 11, da qual algumas informações foram ocultadas em cumprimento das políticas de segurança da informação da FCHS.

Mostrou-se necessário organizar serviços similares ou buscar alternativas para redução de danos em caso de falha dos servidores físicos, alguns deles com atribuições que, se indisponíveis, deixariam serviços críticos inacessíveis por tempo considerável até sua efetiva migração manual para outro *node*.

Figura 11 - Parque de Virtualização Inicial na FCHS – Vmware



Fonte: O autor

3.4 MIGRAÇÃO DOS SERVIÇOS

Com o início dos estudos de ferramentas capazes de suprir as limitações do VMware, foram necessários alguns testes para homologação de migração das máquinas tanto virtualizadas no ESXi quanto instaladas nos servidores físicos.

Uma das técnicas mais simples para as máquinas Linux foi a utilização de

migração via rede, criando uma máquina com configurações similares no ProxMox e clonando via Clonezilla (CLONEZILLA, 2020), uma ferramenta popular no universo de manutenção de sistemas criada como alternativa ao Norton Ghost.

Em máquinas xBSD foi necessário migrar as partições uma a uma, também feitas via processo similar, utilizando principalmente a ferramenta Partclone inclusa no pacote do Clonezilla.

Com relação a máquinas Windows, outros procedimentos foram necessários. Todos os *drivers* precisaram ser removidos antes de importar a máquina e, em seguida, reinstalados com os *drivers* adequados do KVM / VirtIO. A seguir, foi necessário habilitar o serviço de *balooning*, recurso capaz de entregar dinamicamente a alocação de memória RAM.

Há diversas formas de se trazer os discos e dados dos servidores em tempo real ou em situação de recuperação nos sistemas com problemas de disco ou falta de espaço. Alguns exemplos úteis a saber:

Método 1: *Stream* direto via *pipe*, através de conexão ssh migrando os dados de forma mais direta com acesso ao disco da máquina origem > destino:

```
#dd_rescue /dev/sda - | ssh user@remote.host "cat - > /dev/pve/vm-ID-disk-1"
```

Metodo 2: Utilizando um arquivo de *dump*:

```
# ssh 192.168.100.254 'cat /storage/dump/vzdump-qemu-104-2020_08_06-21_07_16.vma.lzo|lzop -d' | qmrestore - 104 --storage local-lvm
```

Metodo 2.1: Fazendo *dump* e importando simultaneamente:

```
# ssh 192.168.100.2 'vzdump --stdout --storage snapshots 104' | qmrestore - 104 --storage local-lvm
```

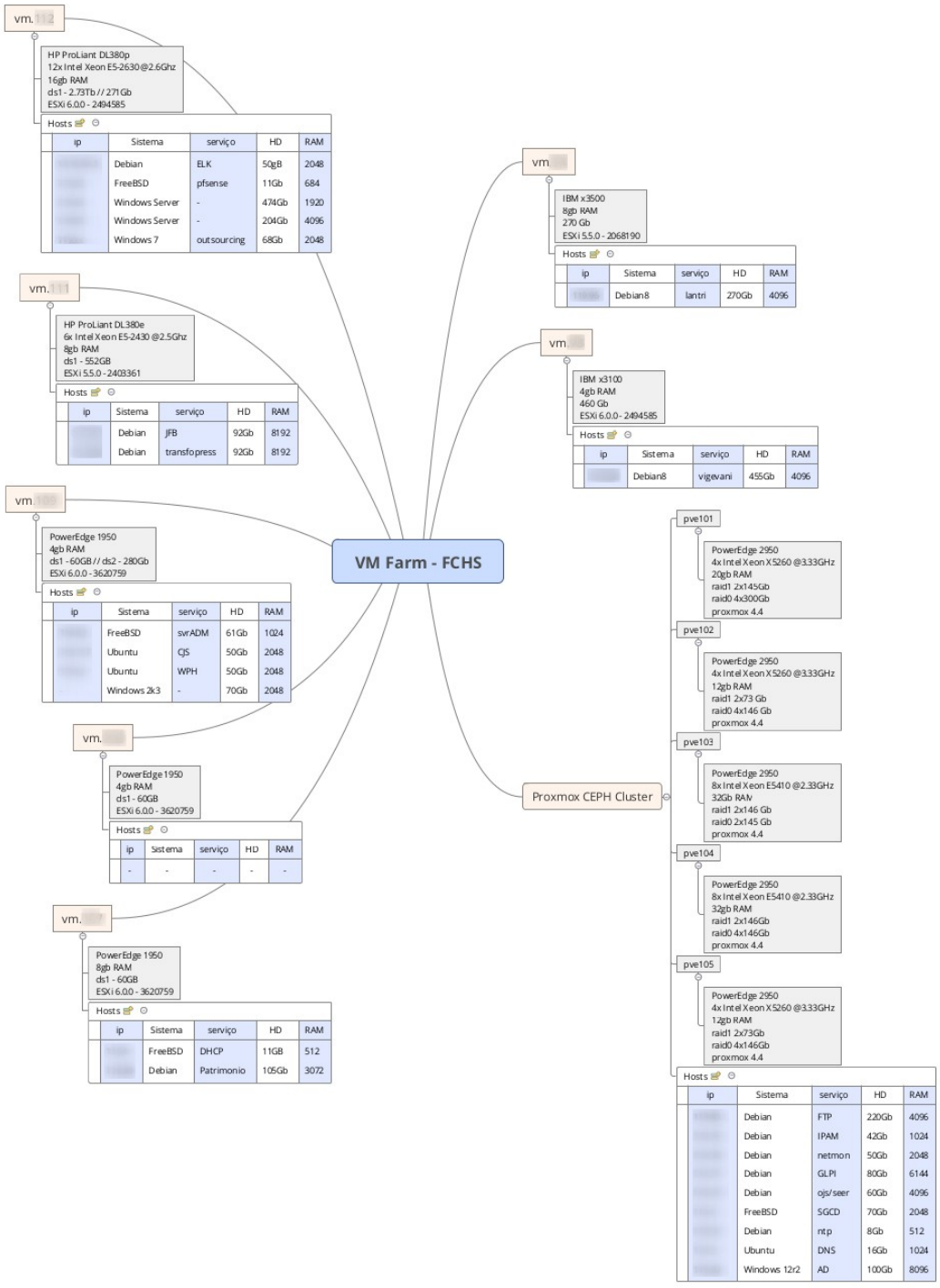
Método 3: em especial para máquinas no ESXi, é possível migrar diretamente o volume do disco em formato proprietário .vmdk diretamente para a *pool* do Ceph com a ferramenta de gerenciamento qm do QEMU:

```
# sshfs -o ro root@IP_ESXI:/vmfs/volumes/{UU_ID}/{VM}/ /mnt/tmp/
# qm importdisk {VM_ID} /mnt/tmp/{DISCO}.vmdk POOL_CEPH --format raw
```

3.4.1 Estado Atual

Dessa forma, foi possível migrar grande parte das máquinas do parque, como demonstrado na Figura 12, chegando ao seguinte estado atual de organização no hipervisor.

Figura 12 - Parque de Virtualização da FCHS em migração para QEMU/KVM



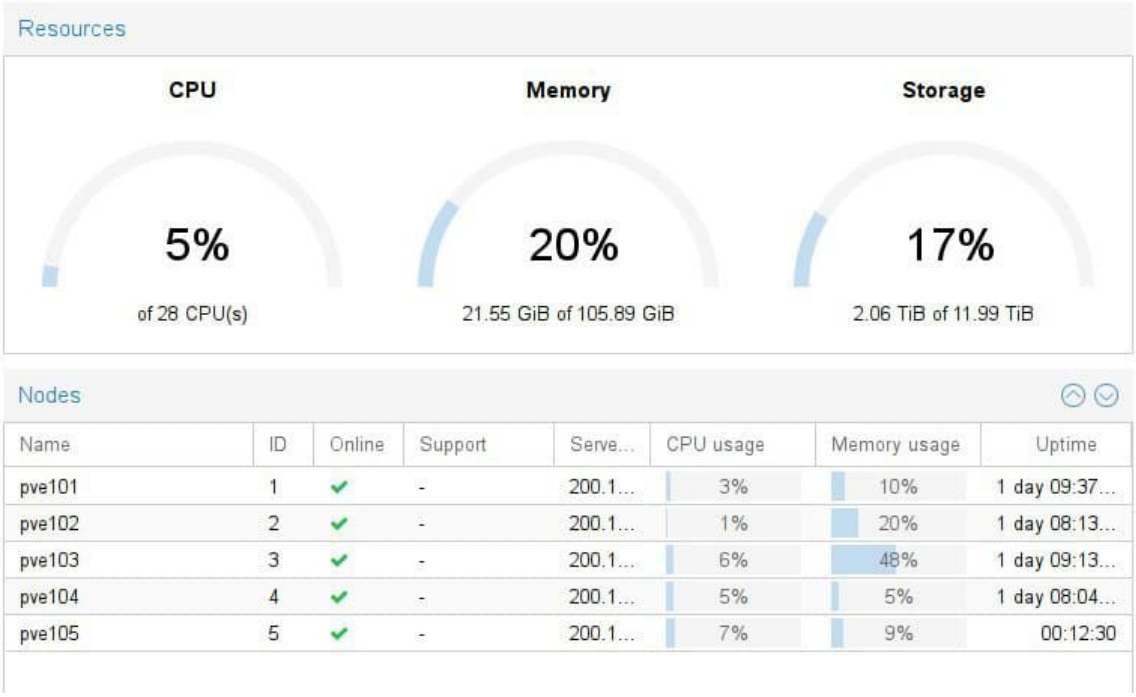
Fonte: O autor

Atualmente, grande parte das Máquinas Virtuais e serviços já foram migrados para o *cluster* Ceph e, por estarem alinhados cinco dos doze servidores (Figura 13) com recursos compartilhados, a carga de recursos disponíveis no arranjo para rodar todos os serviços mostra-se suficiente.

No total, há doze servidores físicos no *datacenter* da FCHS, que agora com

parte dos serviços realinhados neste *cluster* de cinco *nodes* será possível fazer um laboratório de homologação e versionamento, e em um futuro próximo todas as máquinas e serviços virtualizados estarão distribuídos em um *cluster* heterogêneo.

Figura 13 - Consumo de Recursos com o Cluster de 5 nodes, replicação x3



Fonte: O autor

Na figura 13 é possível ver uma captura dos recursos disponíveis nos cinco *nodes* alinhados com grande parte das VMs rodando, totalizando entre eles 28 CPUs, 106Gb de RAM, e 12 Terabytes de armazenamento. O gráfico superior é a soma dos recursos, sendo também possível ver a alocação de uso de CPU e memória de cada node individualmente.

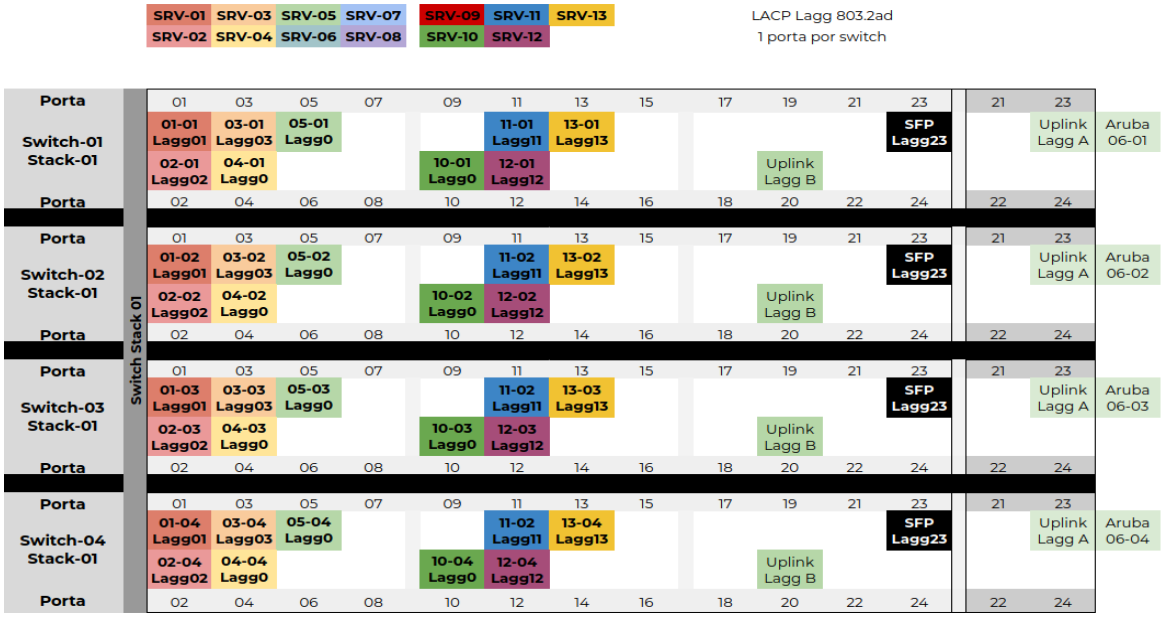
3.4.2 Próximos Passos

Com a homologação e testes do *cluster*, já foi possível adquirir mais discos e memórias, agora não apenas como sobressalentes, mas como valor agregado no parque da FCHS.

Faltam vários passos para a melhora do arranjo dos servidores e para que todos os equipamentos se tornem nossa própria nuvem local de hiperconvergência, em que todos os doze servidores estarão alinhados permitindo uma replicação de perto de sete vezes todos os dados sendo utilizados pelos servidores virtualizados.

Um deles será segregar a conexão dos servidores ao *switch-core*, migrando as conexões para quatro switches alinhados usando ainda mais recursos do Open vSwitch, onde cada servidor terá uma interface de rede conectada em cada um dos switches (Figura 14) em modo de agregação de links (LAGG), sendo um barramento de 4Gb/s com tolerância que até três switches possam cair sem indisponibilidade dos serviços, além de 40Gb/s de *uplink* via fibra óptica para o *switch-core*:

Figura 14 - Agregação das 4 switches via LACP (803.2ad)



Fonte: O autor

Três servidores de menor porte já foram destinados à montagem de uma zona separada do ambiente de produção na qual seja possível simular um *cluster* mínimo e todos os testes de homologação para atualizações do *cluster* principal sejam realizados previamente em um ambiente seguro. Além disso, há a possibilidade de se alinhar esse arranjo paralelo como contenção em situações críticas do *cluster* principal.

Outro objetivo do presente projeto é disponibilizar uma documentação capaz de servir como sugestão ao ambiente da Unesp em relação à melhor gestão de custos e o uso de ferramentas livres, principalmente oriundas de estudos acadêmicos, como o Ceph.

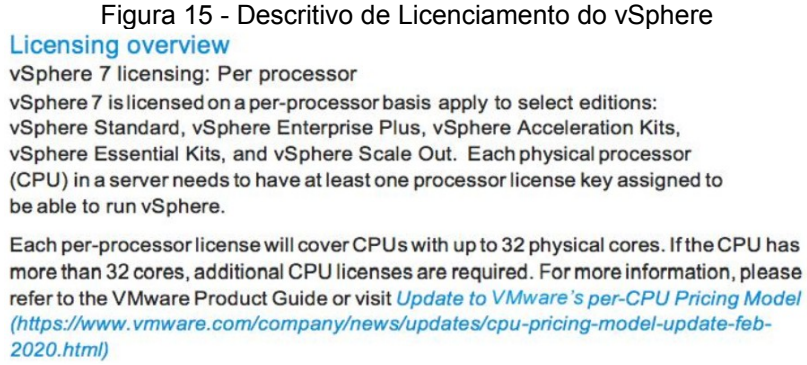
No contexto contemporâneo de poucos investimentos ao ensino superior público, desvio ou mau aproveitamento de recursos e de um verdadeiro sucateamento da educação pública nacional, operar com iniciativas gratuitas e colaborativas apresenta-se como ótima saída aos dilemas cotidianos de instituições

como a Unesp.

3.4.2.1 Comparativo de custos

Para caráter comparativo e para a conclusão dos objetivos deste laboratório, analisaremos a seguir os custos do projeto caso fossem licenciados os serviços da VmWare. Para tanto, serão considerados apenas os 5 servidores iniciais, equipamentos Dell 2950 compostos por dois processadores Intel(R) Xeon(R) CPU X5260 @ 3.33GHz 1x 2018,00 MHz + 1x 2010,00 MHz + 1x 1995,00 MHz + 1x 1994,00 MHz (INTEL, 2020), totalizando 2 processadores físicos e 4 *cores* por servidor. Levamos em consideração, também, os critérios de licenciamento da VmWare, dimensionados pela composição dos servidores, (Figura 15), em que cada unidade de chave de licença é aplicada para cada processador físico, cobrindo até 32 *cores*.

Figura 15 - Descritivo de Licenciamento do vSphere



Licensing overview

vSphere 7 licensing: Per processor

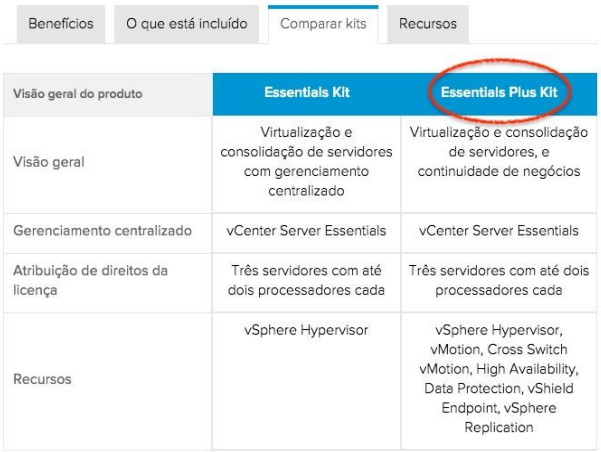
vSphere 7 is licensed on a per-processor basis apply to select editions: vSphere Standard, vSphere Enterprise Plus, vSphere Acceleration Kits, vSphere Essential Kits, and vSphere Scale Out. Each physical processor (CPU) in a server needs to have at least one processor license key assigned to be able to run vSphere.

Each per-processor license will cover CPUs with up to 32 physical cores. If the CPU has more than 32 cores, additional CPU licenses are required. For more information, please refer to the VMware Product Guide or visit *Update to VMware's per-CPU Pricing Model (https://www.vmware.com/company/news/updates/cpu-pricing-model-update-feb-2020.html)*

Fonte: VMWARE (2020c, p. 1)

Considerando contratar os serviços da VMware de forma que fosse possível obter resultados similares aos entregues pelo ProxMox, principalmente os recursos de alta disponibilidade, LACP e storage compartilhado, não foi possível considerar as licenças de entrada descritas na Figura 16, como o Essentials Plus Kit, que é limitado a 3 servidores, de forma não cumulativa:

Figura 16 - Descritivo do kit Essentials Plus



Benefícios | O que está incluído | Comparar kits | Recursos

Visão geral do produto	Essentials Kit	Essentials Plus Kit
Visão geral	Virtualização e consolidação de servidores com gerenciamento centralizado	Virtualização e consolidação de servidores, e continuidade de negócios
Gerenciamento centralizado	vCenter Server Essentials	vCenter Server Essentials
Atribuição de direitos da licença	Três servidores com até dois processadores cada	Três servidores com até dois processadores cada
Recursos	vSphere Hypervisor	vSphere Hypervisor, vMotion, Cross Switch vMotion, High Availability, Data Protection, vShield Endpoint, vSphere Replication

Fonte:VMWARE (2020a, p. 1)

Dessa forma, é necessário pensar na licença *Enterprise*, na qual todos os servidores fossem gerenciados pelo mesmo orquestrador, prevendo o projeto inicial com cinco servidores e eventualmente agregando todos os outros no arranjo de hiperconvergência. Os custos deste licenciamento estão descritos no quadro 1.

Quadro 1 - Custos de Licenciamento VMware Enterprise

	Servidor	CPUs	Custo Anual	Custo Triênio
01	Dell 2950	2	R$ 25.166,40	R$ 33.422,82
02	Dell 2950	2	R$ 25.166,40	R$ 33.422,82
03	Dell 2950	2	R$ 25.166,40	R$ 33.422,82
04	Dell 2950	2	R$ 25.166,40	R$ 33.422,82
05	Dell 2950	2	R$ 25.166,40	R$ 33.422,82
06	Dell 2950	2	R$ 25.166,40	R$ 33.422,82
07	Dell 1950	2	R$ 25.166,40	R$ 33.422,82
08	Dell 1950	2	R$ 25.166,40	R$ 33.422,82
09	Dell 1950	2	R$ 25.166,40	R$ 33.422,82
10	HP DL380	2	R$ 25.166,40	R$ 33.422,82
11	HP DL380	2	R$ 25.166,40	R$ 33.422,82
12	HP DL380	2	R$ 25.166,40	R$ 33.422,82
		Custo 5 Nodes	R$ 125.832,00	R$ 167.114,10
		Custo Total	R$ 301.996,80	R$ 401.073,84

Fonte: VMWARE (2020b)

No quadro 1 algumas linhas estão coloridas de outra forma, notadamente da linha 1 a 5 exemplificando o custo de cinco servidores licenciados, descritos no final

como "Custo 5 Nodes" sendo um projeto mínimo como descrito anteriormente. As sugestões mínimas de um arranjo de servidores para alta disponibilidade é de tres nodes 2.6.6, e cinco seria um ótimo começo para validar a ferramenta além do comparativo de custos de licenciamento. O custo total é calculado caso todos os 12 servidores estivessem licenciados com a assinatura básica para gerenciamento centralizado da VMware.

Para uma economia trienal dessa magnitude com licenciamento de software, as possibilidades de aplicação destes recursos em outras demandas são expressivas e, em um momento futuro, a FCHS poderia inclusive de forma colaborativa com o projeto do ProxMox assinar suas licenças de suporte básico, projetando na data deste documento o custo aproximado de R$2.500,00 por *socket*, algo em torno de R$60.000,00 por ano para todos os 12 servidores e mesmo assim seria um investimento consideravelmente mais acessível (PROXMOX, 2020).

3.4.2.2 Legado e Manutenibilidade

Uma das preocupações de projetos deste porte na máquina pública é sua manutenibilidade, considerando a heterogeneidade da equipe e a morosidade nos processos de compartilhamento e produção de documentos, além da possibilidade da ausência do servidor com maior intimidade com as ferramentas em uma situação crítica que comprometa a disponibilidade dos serviços.

Além de todas as ferramentas utilizadas terem sua documentação acessível por se tratar de projetos de código aberto, toda sua base e estrutura partem da utilização do sistema operacional Linux, utilizado em larga escala em grande parte dos parques computacionais, principalmente em universidades públicas e federais.

Em paralelo a este documento foi criado um guia de referência para operações recorrentes como backup e restauração, inclusão e remoção de um node no arranjo, configuração de switches e equipamentos de rede, manipulação de discos defeituosos e procedimentos para desligamento total em janelas de manutenção ou indisponibilidade de energia elétrica no campus. Este guia foi gerado executando-se todas as operações em conjunto com um dos *sysadmin* da FCHS, de forma que a documentação fosse validada contendo todas as informações necessárias para a execução dos serviços.

Como este documento será apresentado como trabalho de conclusão do curso de Análise e Desenvolvimento de Sistemas da Fatec Franca, eventualmente terá seu acesso aberto a comunidade e não estão inclusas informações mais detalhadas de configuração de rede, topologia e segurança, mas foram documentadas devidamente para os outros *sysadmins* da FCHS como referência a eventuais manutenções.

Não somente no contexto da FCHS, é importante frisar a necessidade de se criar bases de conhecimento locais, nas quais uma documentação sobre todas as experiências e referências necessárias para manter os serviços em produção esteja acessível para a equipe, principalmente se tratando de um serviço hiperconvergente em que um problema pode escalar ao ponto de deixar toda a infraestrutura da instituição indisponível.

CONSIDERAÇÕES FINAIS

Apesar de a primeira vantagem apresentada à instituição ser a redução dos custos, os impactos sociais da validação e implantação de ferramentas livres mostra-se, também, considerável. Através destas, é possível obter resultados satisfatórios sem que haja a necessidade de gastos extras em pagamentos de licenças ou compra de novos equipamentos.

Na universidade pública, toda aquisição de equipamentos e serviços deveria ser feita de forma consciente, levando-se em consideração não apenas o escopo do projeto, mas todos os entraves relacionados à sua manutenibilidade, possíveis restrições orçamentárias ou dificuldades futuras de contratação de servidores e serviços de terceiros. Não depender de fornecedores e tecnologias específicas permite, assim, um remanejamento de recursos de forma inteligente, com impacto direto na qualidade dos serviços prestados e presença da universidade diante da comunidade global.

Como demonstrado ao longo da pesquisa, as ferramentas livres possibilitaram a implantação do projeto na universidade sem custo adicional de software utilizando um orquestrador gerenciável de qualidade e fazendo uso apenas dos recursos já existentes no parque de informática, algumas horas de estudo e trabalho. A economia gerada pela adoção dessas ferramentas poderá, no futuro, possibilitar a ampliação de projetos de TI existentes e fomentar a criação de novos, aumentando as oportunidades que a Unesp têm de contribuir com a cidade de Franca e a comunidade universitária.

Alguns resultados obtidos com este projeto incluem a melhoria na gerência dos servidores físicos e máquinas virtuais, gestão e dimensionamento de recursos, alta disponibilidade, resiliência, e maior facilidade nos processos de backup e restauração de aplicações.

Apesar dos resultados positivos, nem todo o processo foi fluido do início ao fim. Para validação do ambiente, foi necessário estudar várias tecnologias, em especial a possibilidade de “clonar” a máquinas físicas e virtuais a quente para o qemu, considerando um ambiente com máquinas legadas tanto de hardware quanto software.

Também foi preciso validar configurações de rede em situações nas quais muitas regras de segurança restritivas não permitiam um laboratório realista a não ser em horários em que os usuários não estivessem consumindo os recursos. Essas limitações podem gerar horas de trabalho além do expediente padrão ou agendamentos arriscados que simulam erros críticos em ambientes de produção.

Assim, mesmo com todos os testes e laboratórios, apenas foi possível a virada definitiva em um período de férias no campus da universidade, fator que nem sempre será possível em uma empresa sem um calendário pendular como no ambiente educacional.

Além da questão da economia de custos, outra crítica pertinente ao tema é o fato de, mesmo sendo uma das maiores universidades da América Latina (UNESP, 2020a) e fazendo uso frequente de ferramentas livres, a Unesp e suas DTIs contribuem pouco à comunidade internacional de software livre, apesar das infinitas possibilidades que essa relação apresenta. O alcance e experiência da universidade e de seus funcionários poderiam contribuir amplamente para a divulgação de ferramentas gratuitas entre os docentes, os estudantes e a própria comunidade onde o campus está inserido, amplificando o seu uso e popularizando-as.

Entretanto, a observação da universidade revela um grupo de alunos que, em maioria, não se incomoda em fazer uso de aplicações piratas ou não tem recursos para a aquisição dos softwares utilizados ou sugeridos pelos professores. Os professores, por sua vez, nem sempre estão conscientes dos impactos causados pela utilização de formatos proprietários para a pesquisa acadêmica, indexação e compartilhamento de documentos.

Mesmo que a adoção de softwares livres na Unesp seja discutida há anos, os esforços pulverizados demonstraram resultados morosos e uma decepcionante restrição desse uso à DTI, não havendo incorporação efetiva dessas ferramentas no ambiente universitário.

Por fim, gostaríamos de acrescentar que o aprendizado e o conhecimento agregados à equipe de administração de rede durante o processo de implantação das tecnologias aqui descritas teve também um valor muito significativo, trazendo ao grupo mais autonomia quanto à busca de soluções que atendam às exigências dos sistemas utilizados.

REFERÊNCIAS

ALMEIDA, A. V. DE. **Arquiteturas de redes de armazenamento de dados**. Dissertação—Campinas, SP: Universidade Estadual de Campinas, Instituto de Computação, 2006.

CEPH. **How Data Is Stored In CEPH ClusterCeph**, 24 jan. 2014. Disponível em: <https://ceph.io/geen-categorie/how-data-is-stored-in-ceph-cluster/>. Acesso em: 22 nov. 2020

CEPH. **CRUSHMAP : Example of a Hierarchical Cluster MapCeph**, 2 fev. 2015. Disponível em: <https://ceph.io/geen-categorie/crushmap-example-of-a-hierarchical-cluster-map/>. Acesso em: 5 nov. 2020

CEPH. **Ceph Documentation**. Disponível em: <https://docs.ceph.com/en/latest/>. Acesso em: 8 nov. 2020a.

CEPH. **Hardware Recommendations — Ceph Documentation**. Disponível em: <https://docs.ceph.com/en/latest/start/hardware-recommendations/>. Acesso em: 18 nov. 2020.

CEPH. **Architecture — Ceph Documentation**. Disponível em: <https://docs.ceph.com/en/latest/architecture/>. Acesso em: 28 nov. 2020b.

CEPH. **Placement Group States — Ceph Documentation**. Disponível em: <https://docs.ceph.com/en/latest/rados/operations/pg-states/?highlight=heal>. Acesso em: 22 nov. 2020c.

CLONEZILLA. **Clonezilla**. Disponível em: <https://clonezilla.org/>. Acesso em: 23 nov. 2020.

DANIEL, V. DER S. **Ceph at CERN: A Year in the Life of a Petabyte-Scale Block Storage Service**. Disponível em: <https://www.openstack.org/videos/vancouver-2015/ceph-at-cern-a-year-in-the-life-of-a-petabyte-scale-block-storage-service>. Acesso em: 22 nov. 2020.

DONVITO, G.; MARZULLI, G.; DIACONO, D. Testing of several distributed file-systems (HDFS, Ceph and GlusterFS) for supporting the HEP experiments analysis. **Journal of Physics: Conference Series**, v. 513, n. 4, p. 042014, jun. 2014.

FACTOR, M. et al. Object Storage: The Future Building Block for Storage Systems. **IBM Haifa Research Laboratories**, 2005.

FCHS. **Unesp**. Disponível em: <https://franca.unesp.br/sobre/>. Acesso em: 8 nov. 2020.

FISK, N. **Mastering Ceph: Redefine your storage system**. 2. ed. Birmingham: Packt Publishing Ltd, 2017.

FLORIAN, T. **Ceph Calculator**. Disponível em: <https://florian.ca/ceph-calculator/>. Acesso em: 5 nov. 2020.

GLUSTER. **Gluster Docs**. Disponível em: <https://docs.gluster.org/en/latest/>. Acesso em: 22 nov. 2020.

HARVEY, C. **Hyperconvergence: Pros and Cons**. Disponível em: <https://www.datamation.com/data-center/hyperconvergence-pros-and-cons/>. Acesso em: 18 nov. 2020.

HPE. **O que é hiperconvergência? – Definições de TI empresarial**. Disponível em: <https://www.hpe.com/br/pt/what-is/hyper-converged.html>. Acesso em: 18 nov. 2020.

INNOSTORE. **Innostore - Scale-Out Storage Solution - Ceph**. Disponível em: <http://www.innotta.com.au/products/innostore/>. Acesso em: 28 nov. 2020.

INTEL. **Intel® Xeon® X5260 (Cache de 6 M, 3,33 GHz, barramento frontal de 1333 MHz)**. Disponível em: <https://ark.intel.com/content/www/br/pt/ark/products/33907/intel-xeon-processor-x5260-6m-cache-3-33-ghz-1333-mhz-fsb.html>. Acesso em: 18 nov. 2020.

KEGEL, A. et al. Virtualizing IO through THE IO Memory Management Unit (IOMMU). p. 323, 3 abr. 2016.

KRAL, P. **The Incident Handlers Handbook**SANS Institute, , 5 dez. 2011. . Acesso em: 20 out. 2018

KTENZER. **OpenStack: Integrating Ceph as Storage BackendKeith Tenzer**, 12 set. 2016. Disponível em: <https://keithtenzer.com/2016/09/12/openstack-integrating-ceph-as-storage-backend/>. Acesso em: 22 nov. 2020

KUBERNETES. **Kubernetes Documentation**. Disponível em: <https://kubernetes.io/docs/home/>. Acesso em: 5 nov. 2020.

KVM PROJECT. **KVM**. Disponível em: <https://www.linux-kvm.org/page/Main_Page>. Acesso em: 8 nov. 2020.

LIZARD. **LizardFS**, 2020. Disponível em: <https://lizardfs.com/>. Acesso em: 8 nov. 2020

LXC. **Linux Containers**. Disponível em: <https://linuxcontainers.org/>. Acesso em: 8 nov. 2020.

OPEN VSWITCH. **Open vSwitch**. Disponível em: <https://www.openvswitch.org/>. Acesso em: 8 nov. 2020.
OPENSTACK. **Open Source Cloud Computing Infrastructure**. Disponível em: <https://www.openstack.org/>. Acesso em: 22 nov. 2020.

OPENZFS. **OpenZFS**. Disponível em: <https://openzfs.org/wiki/Main_Page>. Acesso em: 8 nov. 2020.

PROXMOX. **Proxmox VE Enterprise Support Subscriptions**. Disponível em: <https://www.proxmox.com/en/proxmox-ve/pricing>. Acesso em: 18 nov. 2020.

PROXMOX PROJECT. **About Proxmox**. Disponível em: <https://www.proxmox.com/en/about>. Acesso em: 8 nov. 2020.

ROOK. **Rook**. Disponível em: <https://rook.io/>. Acesso em: 5 nov. 2020.

SEO, C. E. Virtualização - Problemas e desafios. p. 7, 2009.

UNESP. **Protocolo de Brasília**, 2 set. 2011. Disponível em: <https://www.unesp.br/aci_ses/unespinforma/acervo/22/protocolo-de-brasilia>. Acesso em: 2 set. 2011

UNESP. **Unesp nos Rankings**. Disponível em: <https://www2.unesp.br/portal#!/rankings>. Acesso em: 24 nov. 2020a.

UNESP, D. **DTI FCHS**. Disponível em: <https://www.franca.unesp.br/#!/dti>. Acesso em: 18 nov. 2020b.

VMWARE. **VMware vSphere Essentials Kits**. Disponível em: <https://store.vmware.com/store/vmwbr/pt_BR/cat/categoryID.67818600&src=eBIZ_StoreHome_Featured_EssentialsPlus_BR>. Acesso em: 18 nov. 2020a.

VMWARE. **VMware vSphere Enterprise Plus**. Disponível em: <https://store-us.vmware.com/vmware-vsphere-enterprise-plus-284281000.html>. Acesso em: 18 nov. 2020b.

VMWARE. **VMware vSphere Pricing**. Disponível em: <https://www.vmware.com/reusable_content/vsphere_pricing.html>. Acesso em: 12 nov. 2020c.

WEIL, S. A. et al. Dynamic Metadata Management for Petabyte-scale File Systems. p. 12, 2004.

WEIL, S. A. et al. Ceph: A Scalable, High-Performance Distributed File system. **7th USENIX Symposium on Operating systems Design and Implementation**, p. 14, 2006a.

WEIL, S. A. et al. **CRUSH: Controlled, Scalable, Decentralized Placement of Replicated Data**. Disponível em: <https://www.ssrc.ucsc.edu/pub/weil-sc06.html>. Acesso em: 19 nov. 2020b.

WEIL, S. A. **Ceph: Reliable, Scalable, and High-Performance Distributed Storage**. Dissertação—Santa Cruz, CA: University of California, 2007.

WEIL, S. A. et al. RADOS: A Scalable, Reliable Storage Service for Petabyte-scale Storage Clusters. p. 10, 2007.