CEPH – Dados Geo-Distribuídos de Baixo Custo e Alta Disponibilidade

Guilherme A. Geronimo [1] **e Fabricio Hipolito** [1]

[1]Superintendência de Governança Eletrônica e T.I. e Comunicação – SeTIC
Universidade Federal de Santa Catarina – UFSC
Florianopolis – SC – Brasil

{guilherme.geronimo,fabricio.hipolito}@ufsc.br

Resumo. *Atualmente soluções comerciais de armazenamento apresentam um valor por Terabyte líquidos demasiadamente elevado para ambientes de (i) alta disponibilidade e (ii) com dados geograficamente distribuídos. Assim, recorremos ao projeto de código aberto CEPH para sanar esta demanda. Este artigo aborda o contexto, requisitos, modelagem e implementação da solução de armazenamento de dados adotado pela Universidade Federal de Santa Catarina – UFSC. Modelamos e implementamos um cenário, com 3 centros de dados, no qual pode-se desativar um centro de dados e ainda garantimos o acesso de leitura e escrita as aplicações, sem a necessidade de duplicá-los, reduzindo o custo do Terabyte armazenado em 60% a 83% aproximadamente.*

1. Introdução

O armazenamento de dados é uma demanda com crescimento constante e que, para as IFES, se acentuou nos últimos anos com a digitalização e centralização de serviços analógicos (i.e. telefonia voip, processos digitais, cameras IP etc). Para sanar estas demandas, as instituições vem adotando soluções comerciais que, apesar de oferecerem suporte e outras garantias, apresentam uma série de desvantagem que impactam majoritariamente nos órgãos governamentais, como:

Componentes Exclusivos são requisitos forçados pelos equipamentos adquiridos, e.g discos com *firmwares* exclusivos. Isto impossibilita adquirir componentes alternativos, a fim de torná-los *commodity*. Forçando os órgãos a (i) comprar peças que chegam a custar até 10 vezes mais ou (ii) expandir a garantia de trocas do equipamento, que eleva o valor da garantia.

Expansão gradual dos equipamentos existentes é inviável, o que força a expansão em grandes "saltos", mesmo que estes não sejam utilizados por um bom tempo. Isto dificulta a compra, pois exige uma alta dotação orçamentária. Devido as regras de tributação brasileira, é mais barato comprar um equipamento repleto de discos que comprar um equipamento seco e, aos poucos, ir adicionando-os discos sob demanda. Além de elevar investimento inicial, inibe a utilização de verbas menores que surgem esporadicamente.

Depreciação Forçada de equipamentos é um fato que ocorre devido a falta (ou valor exacerbado) de garantia e suporte a equipamentos com mais de 5 anos de uso. Acarretando no desperdício de dinheiro público.

Tentando sanar estas e outras demandas, selecionamos o projeto (código aberto) CEPH [1] para utilizar em nossa infraestrutura. Iniciado em 2007, este projeto é agnóstico aos equipamento tornando estes uma *commodity*, i.e. independe da marca ou modelo. Internamente, ele trata os dados como objetos e conta com módulos complementares para implementar diferentes interfaces de acesso, e.g. Sistema de Arquivos (CephFS), Serviço de Objetos (RGW), Serviço de Blocos via rede, e.g. NBD [2] e iSCSI [3]. No quesito de alta disponibilidade, ele provê copia de dados e o uso de algoritmos similares ao RAID (Redundant Array of Independent Disks), que possibilita aumentar a resiliência do serviço sem aumentar as despesas (*overhead*) de armazenamento.

Com esta solução, implementamos um cenário no qual nossos dados estão distribuídos entre 3 Centros de Dados, possibilitando que a queda de um deles não interfira no acesso aos dados. Neste, 6 servidores provêm 384 Terabyte de armazenamento bruto, que resulta em aproximadamente 250 Terabyte de espaço líquido.

Assim, as contribuições deste trabalho estão em:

- Descrever os requisitos do nosso cenário (Seção 2.1), para que o leitor possa entender e, possivelmente, comparar nosso ambiente com o seu.
- Explicar a arquitetura do CEPH e seu funcionamento (Seção 2.2), criando uma base de conhecimento superficial para o leitor entender nossas escolhas.
- Apresentar nossa proposta de solução (2.3), baseado nos requisitos e arquitetura apresentados previamente.
- Descrever os detalhes técnicos de implementação e configuração da infraestrutura, desenhando a ponte entre o planejado e o implementar. E, apresentar os testes de performance obtidos e observados na implementação (Seção 3).
- Correlacionar as demandas com a solução e apresentar os próximos passos da pesquisa (Seção 4).

2. Método

A fim de desenharmos uma solução para a nossa demanda, iniciamos definindo o que realmente necessitamos, em outras palavras, os requisitos funcionais e não funcionais de nosso cenário. Então, a fim de introduzir os conceitos ao leitor, apresentamos um breve histórico do CEPH e sua arquitetura. Por fim, unimos as demandas aos conceitos e apresentamos nossa proposta de solução.

2.1. Requisitos

Sobre os requisitos não funcionais, que não estão diretamente ligados a ferramenta em si, listamos:

Replicação geograficamente distribuída, em uma instituição com histórico de perda de dados decorrido de incêndios e enchentes, o armazenamento descentralizado é considerado imprescindível para a solução.

Heterogeniedade de equipamentos é uma situação ativa em nosso ambiente. Devemos utilizar servidores de marcas variadas, discos de diferentes modelos e com diferentes tamanhos.

Alta Disponibilidade dos dados de forma que um centro de dados possa ficar fora do ar, sem interromper as operações de leitura e escrita de dados.

Sobre o requisitos funcionais, que o sistema tem que implementar, listamos:

Métodos de Acesso que atendam a pluralidade do nosso ambiente. Da camada de aplicação até a camada de virtualização, identificamos que necessitaríamos das seguintes interfaces de acesso: (i) Serviço de Objetos, para que aplicações externalizem o tráfego de *uploads/downloads* e o armazenamento de seus dados, possibilitando um melhor uso de *containers*; (ii) Sistema de Arquivos remoto, como um serviço NFS, que possibilite o compartilhamento de arquivos entre vários sistemas operacionais simultaneamente; e como (iii) Dispositivo de Bloco (Block Device) na rede, como os protocolos iSCSI [3] e NBD [2], que possibilitam entregar um bloco de disco aos sistemas operacionais ou para hipervisores de virtualização.

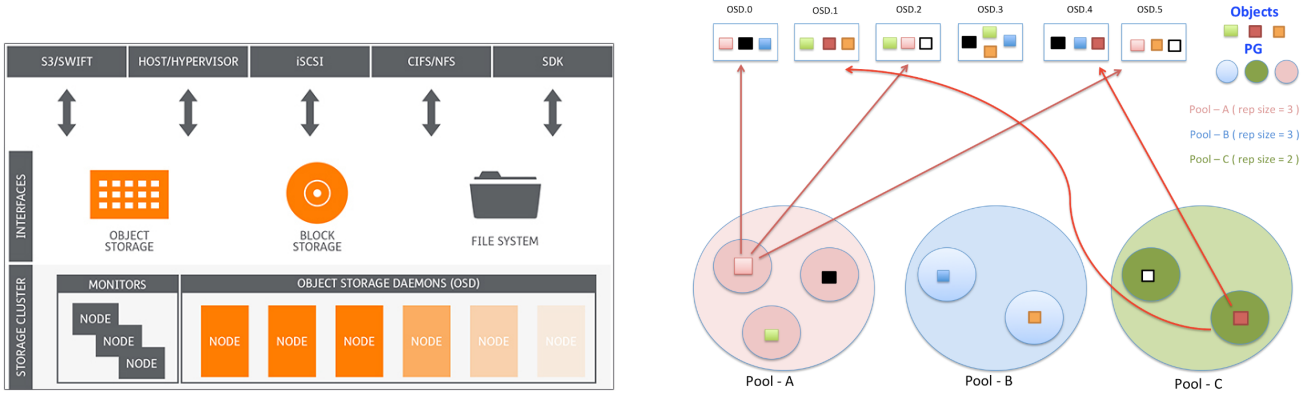
Snapshots é indispensável não somente para máquinas virtuais, mas também diretamente no uso de Sistemas de Arquivos. Em nosso cenário, utilizamos para fazer pontos de restauração de bancos de dados, possibilitando retorná-los de forma rápida e fácil em um determinado ponto do tempo passado.

2.2. CEPH

A ideia do CEPH surgiu em 2007 na tese de doutorado de Sage Weil [4]. Apesar de ser Co-fundador da empresa DreamHost [5], esta iniciativa foi patrocinada por diversos orgãos públicos e privados[1] e tem como meta principal prover uma solução de armazenamento com (i) operações completamente distribuídas, (ii) sem um ponto único de falha, (iii) escalável ao nível de exabytes e (iv) disponível para a comunidade.

No ambiente CEPH, todos os arquivos (ou agrupamento de dados) são denominados objetos, compostos pelo dado em si, uma identificação única (ID) e múltiplos metadados, configurados pela solução e pelos usuários da ferramenta (e.g. data de criação, expiração, DOI, aplicação relacionada etc).

[1] United States Department of Energy (DOE), Oak Ridge National Laboratory (ORNL), Lawrence Livermore National Laboratory (LLNL), Los Alamos National Laboratory (LANL), Intel Corporation, Microsoft Corporation, SAP Laboratories e outros [6]



(a) Diagrama da arquitetura CEPH **(b) Diagrama de Pools, PGs e OSDs.**

Figura 1: Arquitetura e Logica do CEPH

O núcleo da solução é a estratégia RADOS – Armazenamento Confiável Autônomo Distribuído de Objetos (*Reliable Autonomic Distributed Object Store*), que conta com 5 tipos de agentes (3 essenciais e 2 opcionais) para orquestrar o ambiente, como demonstra a Figura 1a:

OSD Armazenadores de Objetos (Object Storage Daemons) lidam com a entrada e saída dos objetos nos discos, sua integridade (*scrub*) e replicação em casos de falha. No ambiente, recomenda-se não mais de um OSD para cada disco. Paralelamente, aconselha-se reservar um mínimo de 2GB de memória para cada OSD, no entanto, em momento de recuperação após sinistros, eles ocupem mais memória.

MON Monitores gerenciam o status do ambiente (i.e. serviços disponíveis e indisponíveis), dividindo-os temporalmente em "eras", possibilitando que os serviços que voltem a funcionar identifiquem o que "perderam" enquanto estavam indisponíveis. No ambiente, recomenda-se estarem sempre em números ímpares, devido ao algoritmo de eleição utilizado;

MGR Processo de Gerenciamento, rodando junto aos monitores, adotam responsabilidades auxiliares aos MONs na forma de plugins, e.g. Interface web, integração com ferramentas externas (zabbix), API (REST) etc. No ambiente, sempre deve haver um rodando para garantir a integração com outros serviços externos.

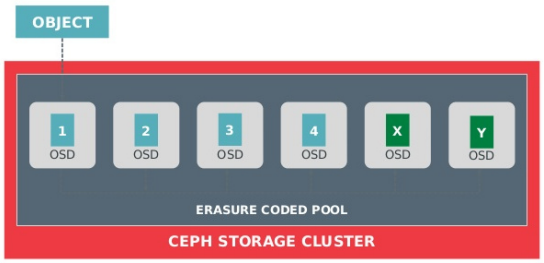
MDS Servidor de Metadados, exclusivo no provimento do CephHFS, este faz a tradução entre objetos e arquivos, tendo a representação da arvore de arquivos e provendo ao usuário a localização dos objetos que o compõe. No ambiente, recomenda-se haver sempre 2 ou mais, agindo de forma passiva entrando em ação em caso de falha do principal (*failover*).

Considere a Figura 1b. Conceitualmente, os objetos são distribuídos em Grupos de Alocação (PGs) que compõe Tanques (*pools*) utilizados por diferentes aplicações (e.g. CephFS, RGW e RBD). Cada Pool possui sua própria política de replicação e é constituído por múltiplos PGs – estes não são compartilhados com outros Pools. Cada PG enumera os OSDs onde os dados devem ser escritos e replicados.

A Figura 1b apresenta um exemplo de 3 Pools, 7 PGs e 6 OSDs. Pela legenda sabemos que o Pool A e B possuem uma política de replicação igual a 3 e o Pool C igual a 2. Pela setas, identificamos que um dos PGs do Pool A irá armazenar seus dados nos OSDs 0, 2 e 5; e um dos PGs do Pool C, irá armazenar no OSD 1 e 4. A ordem de escrita não é explicita, porém numa operação de escrita, o cliente escreve apenas no OSD principal do PG (o primeiro OSD) e este OSD fará as replicas para os outros OSDs. Note que, a ordem dos OSDs, a quantidade de PGs do Pool e o número de replicas são configurados na criação do Pool.

A replicação dos dados é distribuída por domínios, e.g. OSD, host, rack, sala etc. Isto significa que ao distribuir os dados o CEPH colocará uma replica em diferentes elementos do domínio. Caso o domínio seja "salas", ele colocará os dados em servidores que fiquem em salas diferentes, podendo perder até $N-1$ "salas". Caso não hajam elementos o bastante (i.e. mais réplicas que "salas") o sistema fica degradado.

Utilizar réplicas traz consigo um grave problema de *overhead* de dados, pois quanto mais replicas são necessárias, mais espaço de armazenamento bruto é necessário. Por exemplo, em uma política de 3 réplicas é necessário disponibilizar 200% a mais de espaço bruto para armazenamento, e.g. para 1 Petabyte



(a) Estrategia de Erasure Code.

Recurso	OSDs	MON	MDS
Processadores	2	2	2
Núcleos	16	2	2
Rede	10GB	1GB	1GB
Memória	128 GB	8GB	8GB
Discos	8 x 8 TB	50GB	50GB
SSDs	2 x 120 GB	-	-

(b) Especificação dos Servidores

Figura 2

líquido, são necessários 3 Petabyte de armazenamento bruto. Visando sanar esta demanda, o projeto provê a estratégia *Erasure Code*, baseada no algoritmo de correção de erros [7], quebra o dado em K blocos e gera mais M blocos de paridade de mesmo tamanho, para corrigir possiveis falhas. Resultando em $K + M$ blocos armazenados e uma tolerância a M falhas. Na Figura 2a temos um dado quebrado em 4 partes ($K = 4$) e 2 blocos de paridade ($M = 2$), resultando em 6 blocos armazenados – *overhead* de 50% – , e.g. Para 1 Petabyte líquido, são necessários 1.5 Petabyte de armazenamento bruto. Apesar de parecer milagroso, podendo-se aumentar indefinidamente a resiliência do pool, quando mais blocos são necessários, mais processamento é necessário e a latência de escrita e leitura do dado acabe se elevando.

2.3. O Design

Dados os requisitos da Seção 2.1 e as funcionalidades da Seção 2.2, propomos a seguinte solução:

Serviços: Como almejamos armazenar backups, implementou-se os serviços: (i) CephFS para backups de maquinas virtuais (integrado a solução VEEAM [8]) e de arquivos em geral. O VEEAM gere suas alterações sem intervenções, para os arquivos armazenamos no CephFS, fazemos snapshots e um RSYNC atualiza os mesmos; (ii) RBD para oferecer blocos as VMs que utilizam ZFS e *snapshots*.

Pools: Apesar do CephFS prover uma única árvore de arquivos, cada serviço de backup fica em uma pasta própria e estas ficam em pools distintos. Assim, temos liberdade flexibilidade na politica de replicação. Ainda, visando otimizar os serviços, dedicamos discos de baixa latência (SSD) para os pools de metadados do CephFS e de RBD.

Replicação: Todo Pool de dados (CephFS e RBD) utiliza uma política de *erasure code* $4 + 2$ e "Rack" como domínio de replicação, pois em cada Centro de Dados existem 2 Racks (virtuais) com os servidores. Considerou-se também uma política de *erasure code* $2 + 1$ com "Centro de Dados" como domínio, porém isto abre uma nova fragilidade: a queda simultânea de 2 ou mais discos em diferentes Centros de Dados. Com a política $4 + 2$ garantimos: (i) um baixo *overhead* – 50%, (ii) uma resiliência contra queda de um Centro de Dados, e (iii) uma resiliência contra queda de dois discos aleatórios.

3. Resultados

Para testar a solução utilizou-se 6 servidores físicos (especificação na Tabela 2b) interligados com uma rede 10GB em camada 2 com MTU 9000 (*jumboframe*).Em cada centro de dados há um servidor físico de menor porte que exerce o papel de MON. No centro de dados principal, virtualizado, fica o servidor de MDS.

Testou-se a performance da solução com a ferramenta CrystalDiskMark [9] em um servidor virtual ligado ao CEPH por uma rede 10GB. Para extrapolar qualquer camada de cache de leitura ou escrita, optou-se por executar testes com 32GB de escrita e leitura de dados. Para obter um valor médio de medição, cada teste foi executado 5 vezes. Assim os seguintes testes foram processados:

Seq Q32T1: Fila de 32 operações de Leitura e Escrita de arquivos grandes.
4K Q32T1: Fila de 32 operações de Leitura e Escrita de arquivos com 4KB.
Seq: Sucessivas operações de Leitura e Escrita de arquivos grandes.
4K: Sucessivas operações de Leitura e Escrita de arquivos com 4KB.

Considere a Figura 3c, apesar de ter apresentado uma performance satisfatória, no dia a dia as ferramentas de monitoração mostram números um pouco menores. A taxa de escrita media é de 430 MB/s com picos de 700 MB/s. A taxa de leitura média é de 430 MB/s com picos de 1.3 GB/s, como mostram as Figuras 3b e 3a No entanto, durante procedimentos internos de recuperação de dados presenciou-se fluxos de 1,6 GB/s de escrita. Estes dados foram observados durante a execução das ferramentas de utilizada, logo as métricas podem variar.

4. Conclusão

Neste artigo apresentamos a solução de armazenamento de dados distribuído implantado na Universidade Federal de Santa Catarina. Este distribui os dados entre 3 centros de dados e garante o acesso de leitura e escrita caso um destes fique indisponível. A solução demonstrou uma performance de leitura e escritas superiores a 600 MB/s, que, para a utilização dada, é considerada suficiente. Monetariamente, comparando com 3 orçamentos de soluções proprietárias o custo da solução saiu entre 60% a 83% mais barato.

No entanto alguns problemas ainda persistem na solução: (i) para expandir o ambiente deve-se adquirir N servidores novos, onde N é o numero de domínios utilizado nas replicações. Ou seja, em nosso caso, temos que adicionar sempre 6 discos ou servidores novos, Porém esta ainda é uma questão politico-estratégica a ser tomada, não técnica.

Os próximos passos desta solução é: (i) implantar o serviço de objetos (RGW), tipo Swift e Amazon S3 e; (ii) migrar a área de buffer dos discos mecânicos para dispositivos de baixa latência, i.e. NVMe.

5. Notas de Reconhecimento

Agradecemos a todos os amigos e colegas que acreditaram nesta ideia, investindo seu tempo, suor e confiança para o sucesso do mesmo. Em especial ao Diretor de T.I. Bruno Carlo Celegrim Amattos, ao Técnicos Ramon Dutra Miranda e a equipe de EaD do curso de Libras e Administração.

Referências

[1] CEPH. Project documentation. URL {http://ceph.com}.
[2] Wikipedia. Network block device, . URL {https://nbd.sourceforge.io/}.
[3] Wikipedia. Internet small computer system interface, . URL {https://pt.wikipedia.org/wiki/ISCSI}.
[4] Sage A Weil, Scott A Brandt, Ethan L Miller, Darrell DE Long, and Carlos Maltzahn. Ceph: A scalable, high-performance distributed file system. In *Proceedings of the 7th symposium on Operating systems design and implementation*, pages 307–320. USENIX Association, 2006.
[5] Dreamhost. Dreamhost co. URL {https://dreamhost.com/}.
[6] System Research Lab. Srl sponsors. URL {https://systems.soe.ucsc.edu/sponsors}.
[7] Irving S Reed and Gustave Solomon. Polynomial codes over certain finite fields. *Journal of the society for industrial and applied mathematics*, 8(2):300–304, 1960.
[8] VEEAM. Software company. URL {https://www.veeam.com}.
[9] Crystal Dew World. Crystal disk mark. URL {https://crystalmark.info/en/software/crystaldiskmark/}.

(a) Teste de Leitura e Escrita. **(b) Medições de Leitura.** **(c) Medições Escrita.**

Figura 3: Resultados