Analyzing the Performance of the S3 Object Storage API for HPC Workloads

... S3 ...

- Um dos primeiros serviços da AWS, lançado em 2006.

... S3 ...

- Um dos primeiros serviços da AWS, lançado em 2006.
- Armazenamento de arquivos (objetos), sem noção de hierarquias

... S3 ...

- Um dos primeiros serviços da AWS, lançado em 2006.
- Armazenamento de arquivos (objetos), sem noção de hierarquias
- Unidade lógica de coleção de objetos é o Bucket

... S3 ...

- Um dos primeiros serviços da AWS, lançado em 2006.
- Armazenamento de arquivos (objetos), sem noção de hierarquias
- Unidade lógica de coleção de objetos é o Bucket
- Cada objeto possui um nome, chamado de chave, único no seu bucket.

... S3 ...

- Um dos primeiros serviços da AWS, lançado em 2006.
- Armazenamento de arquivos (objetos), sem noção de hierarquias
- Unidade lógica de coleção de objetos é o Bucket
- Cada objeto possui um nome, chamado de chave, único no seu bucket.
- Um dos serviços mais utilizados para armazenar documentos na AWS, liberando espaço de serviços mais caros como RDS.

... Object Storage API ...

- API REST estável (i.e última atualização da API data de 1/3/2006)

... Object Storage API ...

- API REST estável (i.e última atualização da API data de 1/3/2006)
- Capacidade de receber arquivos de até 5TB

... Object Storage API ...

- API REST estável (i.e última atualização da API data de 1/3/2006)
- Capacidade de receber arquivos de até 5TB
- Controle majoritário via cabeçalhos customizados e parâmetros de consulta na url

... Object Storage API ...

- API REST estável (i.e última atualização da API data de 1/3/2006)
- Capacidade de receber arquivos de até 5TB
- Controle majoritário via cabeçalhos customizados e parâmetros de consulta na url
- Poucos endpoints

... Object Storage API ...

- API REST estável (i.e última atualização da API data de 1/3/2006)
- Capacidade de receber arquivos de até 5TB
- Controle majoritário via cabeçalhos customizados e parâmetros de consulta na url
- Poucos endpoints
- Pelas idiossincrasias da api, sugere-se acessa-la via bibliotecas

... Object Storage API ...

- API REST estável (i.e última atualização da API data de 1/3/2006)
- Capacidade de receber arquivos de até 5TB
- Controle majoritário via cabeçalhos customizados e parâmetros de consulta na url
- Poucos endpoints
- Pelas idiossincrasias da api, sugere-se acessa-la via bibliotecas
- Tendo se tornado padrão de fato no armazenamento de objetos, múltiplas implementações fora da aws surgiram
  - On-premises
    - Swift
    - MinIO
  - Cloud
    - Google
    - IBM

... HPC ...

- Computação para problemas complexos (i.e fisica, quimica, engenharia) e custosos computacionalmente

... HPC ...

- Computação para problemas complexos (i.e fisica, quimica, engenharia) e custosos computacionalmente
- Criação de programas para ganho de performance

... HPC ...

- Computação para problemas complexos (i.e fisica, quimica, engenharia) e custosos computacionalmente
- Criação de programas para ganho de performance
- Dispostos em clusters com conexão de alta velocidade

... HPC ...

- Computação para problemas complexos (i.e fisica, quimica, engenharia) e custosos computacionalmente
- Criação de programas para ganho de performance
- Dispostos em clusters com conexão de alta velocidade
- Usualmente realizado em on-premises
- Sistemas distribuídos sem memória compartilhada, comunicando-se através de mensagens

... HPC ...

- Computação para problemas complexos (i.e fisica, quimica, engenharia) e custosos computacionalmente
- Criação de programas para ganho de performance
- Dispostos em clusters com conexão de alta velocidade
- Usualmente realizado em on-premises
- Sistemas distribuídos sem memória compartilhada, comunicando-se através de mensagens
- Pela dificuldade em coordenar manualmente o cluster e por padrões de interação emergirem, surgiu um padrão de comunicação chamado MPI

... HPC ...

- Computação para problemas complexos (i.e fisica, quimica, engenharia) e custosos computacionalmente
- Criação de programas para ganho de performance
- Dispostos em clusters com conexão de alta velocidade
- Usualmente realizado em on-premises
- Sistemas distribuídos sem memória compartilhada, comunicando-se através de mensagens
- Pela dificuldade em coordenar manualmente o cluster e por padrões de interação emergirem, surgiu um padrão de comunicação chamado MPI
- O MPI possui varias implementações sendo a mais famosa o OpenMPI

... Workloads

- Tarefas de HPC consistem em operações de computação interna aos nós, IO na forma de mensagens entre os nós, IO no consumo de dados em memória secundaria e IO na produção de dados em memória secundaria, alguns finais, outros que serão consumidos por outros nós.

... Workloads

- Tarefas de HPC consistem em operações de computação interna aos nós, IO na forma de mensagens entre os nós, IO no consumo de dados em memória secundaria e IO na produção de dados em memória secundaria, alguns finais, outros que serão consumidos por outros nós.
- A performance da leitura e a escrita desses dados depende do tamanho da informação a ser lida ou escrita, da topologia da rede e da localidade das informações lidas ou escritas.

... Workloads

- Tarefas de HPC consistem em operações de computação interna aos nós, IO na forma de mensagens entre os nós, IO no consumo de dados em memória secundaria e IO na produção de dados em memória secundaria, alguns finais, outros que serão consumidos por outros nós.
- A performance da leitura e a escrita desses dados depende do tamanho da informação a ser lida ou escrita, da topologia da rede e da localidade das informações lidas ou escritas.
- Tradicionalmente, Data centers utilizam sistemas de arquivos paralelos como Lustre ou GPFS para obter a maior performance.

... Workloads

- Tarefas de HPC consistem em operações de computação interna aos nós, IO na forma de mensagens entre os nós, IO no consumo de dados em memória secundaria e IO na produção de dados em memória secundaria, alguns finais, outros que serão consumidos por outros nós.
- A performance da leitura e a escrita desses dados depende do tamanho da informação a ser lida ou escrita, da topologia da rede e da localidade das informações lidas ou escritas.
- Tradicionalmente, Data centers utilizam sistemas de arquivos paralelos como Lustre ou GPFS para obter a maior performance.
- Lustre oferece uma API posix para acesso aos arquivos.

... Workloads

- Tarefas de HPC consistem em operações de computação interna aos nós, IO na forma de mensagens entre os nós, IO no consumo de dados em memória secundaria e IO na produção de dados em memória secundaria, alguns finais, outros que serão consumidos por outros nós.
- A performance da leitura e a escrita desses dados depende do tamanho da informação a ser lida ou escrita, da topologia da rede e da localidade das informações lidas ou escritas.
- Tradicionalmente, Data centers utilizam sistemas de arquivos paralelos como Lustre ou GPFS para obter a maior performance.
- Lustre oferece uma API posix para acesso aos arquivos.
- 25% a 90% dos arquivos gerados por um sistema HPC residem na faixa de 0 a 64KB, o que pode gerar pior performance em um workload do que um arquivo único maior.

... Workloads

- Tarefas de HPC consistem em operações de computação interna aos nós, IO na forma de mensagens entre os nós, IO no consumo de dados em memória secundaria e IO na produção de dados em memória secundaria, alguns finais, outros que serão consumidos por outros nós.
- A performance da leitura e a escrita desses dados depende do tamanho da informação a ser lida ou escrita, da topologia da rede e da localidade das informações lidas ou escritas.
- Tradicionalmente, Data centers utilizam sistemas de arquivos paralelos como Lustre ou GPFS para obter a maior performance.
- Lustre oferece uma API posix para acesso aos arquivos.
- 25% a 90% dos arquivos gerados por um sistema HPC residem na faixa de 0 a 64KB, o que pode gerar pior performance em um workload do que um arquivo único maior.
- Obtenção de metadados (i.e relação de nomes e localização, estrutura de diretorios, permissões) também é uma operação tipica de workloads em HPC.

Analyzing the Performance ...

- Para se analisar algo, precisamos de métricas e testes padronizados

Analyzing the Performance ...

- Para se analisar algo, precisamos de métricas e testes padronizados
- Benchmarks
  - IO500
  - MD-Workbench

´

IO500

- Estrutura testes em dois setups : easy e hard
- Setup easy feito para mostrar o melhor caso do sistema de armazenamento avaliado, com configurações do usuário do benchmark
- Setup hard feito para mostrar o pior caso do sistema de armazenamento avaliado, com configurações pré-configuradas.
- O artigo utilizou dois benchmarks do IO500, a saber, IOR e MDTest nos dois setups, para colher as metricas
  - IOEasy simulating applications with well-optimized I/O patterns.
  - IOHard simulating applications that utilize segmented input to a shared file.
  - MDEasy simulating metadata access patterns on small objects.
  - MDHard accessing small files (3901 bytes) in a shared bucket.

IO500 - metricas

Metricas avaliadas :

- ior-easy-write
- mdtest-easy-write
- ior-hard-write
- mdtest-hard-write
- ior-easy-read
- mdtest-easy-stat
- ior-hard-read
- mdtest-hard-stat
- mdtest-easy-delete
- mdtest-hard-read
- mdtest-hard-delete

IO500 - modificações s3

For IO500, an optimistic S3 interface backend using the libS3 client library is implemented for IOR in the sense that it stores each fragment as one independent object, and as such, it is expected to generate the best performance for many workloads. For identifying bottlenecks, it supports two modes:

- Single bucket mode: created files and directories result in one empty dummy object (indicating that a file exists), every read/write access happens with exactly one object (file name contains the object name + size/offset tuple); deletion traverses the prefix and removes all the objects with the same prefix recursively.
- One bucket per file mode: for each file, a bucket is created. Every read/write access happens with exactly one object (object name contains the filename + size/offset tuple); deletion removes the bucket with all contained objects.

MD-Workbench

- Possui dois modos
  - One bucket, the D datasets are prefixed by the process rank.
  - One bucket per dataset.
- MD-Workbench casa muito melhor com a API do S3

MD-Workbench - metricas

- rate
- throughput

MD-Workbench - modificações s3

- O MD-Workbench funciona com um sistema de plugins, em que o arquivo principal possui um array de vtables com operações de IO
- A adição do suporte ao S3 foi apenas questão de adicionar um novo plugin, implementando as chamadas de IO como chamadas a libs3 e adicionar esse plugin ao array supracitado

sistema sobre teste

- Supercomputador Mistral alemão
- Performance de 2.54 PetaFLOPS (pico de 3.14 PetaFLOPS)
- #33 na lista top500

minIO

- MinIO is a high-performance, S3 compatible object store. It is built for large scale AI/ML, data lake and database workloads. It is software-defined and runs on any cloud or on-premises infrastructure.
- Pode utilizar multiplos backends para prover essa interface. O lustre é utilizado como backend do minIO a fim de rodar os testes sob uma mesma interface mas com uma baseline de um sistema de arquivos tipico.

minIO - modos

- gateway (gw)
  - in this mode, any a S3 request is converted to POSIX requests for the shared file system.
- Standalone (sa)
  - runs one MinIO server on one node with a single storage device. We test configurations from tmpfs (in-memory fs/shm) and the local ext4 file system
- Distributed servers (srv)
  - runs on multiple nodes, object data and parity are striped across all disks in all nodes. The data are protected using object-level erasure coding and bitrot. Objects are accessible from any MinIO server node. In our setup, each server uses the local ext4 file system

fase 1 - minIO - single cliente

- The first tests are performed using IOR directly
- the standalone mode shows the best performance
- Gateway mode is effective for reads, but slow for writes
- Write performance is 1/3rd of the read performance
- Adding a load balancer has minimal influences on throughput in this setting, except when activating the caching mechanism, which has a tremendous impact on the read throughput.
- Compared to the Infiniband network throughput (about 6 GiB/s), only 7.5% and 2.5% of this performance can be obtained for read and write, respectively.

fase 1 - minIO - parallel IO

- Assuming an ideal scenario, we start MinIO in standalone mode with RAM as backend (sa-shm) on one node.
- when increasing the number of tasks per node, we achieve the best throughput (about 6000 MiB/s).
- The performance per client node is 1.5 GiB/s
- For the single server, it is close to the available network bandwidth

fase 2 - Tests against S3 Compatible Systems

- The Benchmarks are performed on Mistral using IO500 and MD-Workbench.
- Tests are conducted against the OpenStack Swift system already available in DKRZ
- Swift version 2.19.2 is used, and the S3 interface is implemented using Swift3 version 1.12.1
- the tests were only conducted using MD-Workbench.
- o motivo dos testes usarem apenas o MD-Workbench é por uma incompatibilidade de assinaturas nas versões da libse.
- On four client nodes and with 20 PPN, a rate of 269.5 IOPs and a 0.5 MiB/s throughput are observed during the benchmark phase with MD-Workbench.
- MD-Workbench reports not only the throughput, but also the latency statistics for each I/O operation.
- A latencia observada com o swift era menos previsivel do que com o Lustre, levando a conclusão que os processos por trás da implementação do s3 eram a principal origem da latência.

fase 3 - S3Embedded

- Por conta dos problemas de escalabilidade reportados na fase 2, foi criada uma biblioteca, S3Embedded, baseada na libs3
- partes da libs3 foram substituidas, a fim de investigar se aquela parte removida era a causadora da latência
- a biblioteca é compativel a nivel de código e binario com a libs3, permitindo que aplicações linkadas dinamicamente com a libs3 possam usar a libEmbedded ao invés.
- a biblioteca contém sub-bibliotecas
  - libS3e.so: This is an embedded library wrapper that converts libs3 calls to POSIX calls inside the application address space.
  - ibS3r.so: This library converts the libs3 calls via a binary conversion to TCP calls to a local libS3-gw application that then executes these POSIX calls, bypassing the HTTP protocol.
  - The results delivered by S3Embedded are very close to the ones obtained for the Lustre direct access,mainly for files larger than 32 MB
  - Com mais nós, a biblioteca S3Embedded melhora a performance em relação ao lustre, porém um gap de 10x existe

Conclusão

- Bibliotecas s3 não estão prontas para hpc
- Com o IO-500 aumentado, a comunidade pode observar o s3
- integrações entre S3 e sistemas de arquivos permitiriam que clientes s3 acessassem IO de HPC