Object store characterisation for ATLAS distributed computing

Doug Benjamin[1], Alastair Dewhurst[2], Peter Love[3], Jaroslava Schovancova[4], on behalf of the ATLAS Experiment

1. Duke University 2. STFC 3. Lancaster University 4. CERN

Summary

In this work we developed a platform for running functional and stress tests against object storage instances. The approach taken was to closely mimic the use case seen in ATLAS distributed computing in terms of client libraries, object sizes, connection counts, and request rates. This was achieved by using Kubernetes clusters to scale the number of concurrent client processes at a level expected to be seen on Grid resources. Initial results show some interesting differences and bottleneck on both the storage side and client side. The decision to use Kubernetes as a platform has highlighted the need to understand the details of such a system compared to traditional running on local platforms. Differences in site behaviour are observed and feedback to site operators has proven useful for their deployments. Much larger scale and more distributed tests are planned and we expect these to provide a better understanding of object store performance and limits.

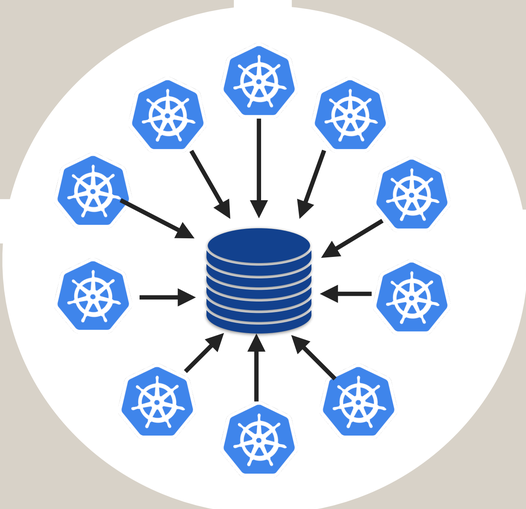
Introduction

Various workflows used by ATLAS Distributed Computing (ADC) are now using object stores as a convenient storage resource using the widespread S3 interface. The typical load produced by these use cases vary widely across the different workflows and for heavier cases it has been useful to understand the limits of the underlying object store implementation. This work describes the performance of object stores currently used by ADC and describes a tool which run periodic functional tests and on-demand stress testing. Initial measurements of transfer times have been made using object of similar size to the ATLAS Event Service use case.

These simple transfer tests are a good start to characterising the object store performance and raise a number of questions about the deployment details of Ceph.

Tooling

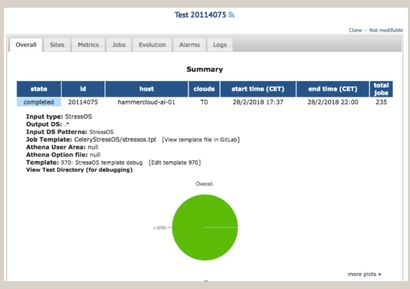
- We use a single threaded python script to mimic a client process which uploads data to the object store
- The client script is packaged in a Docker container to ensure a reproducible environment
- Client BOTO libraries were identical to those used in ATLAS production systems (boto-2.48)
- Scale is achieved by deploying this script on a Kubernetes cluster and replicating the Pod
- Scaling via Kubernetes has the benefit of being deployed in a distributed fashion rather than using multi-threaded client on a single machine or machines in one place.
- To date we have only deployed at Google
- Costs are managed by only running a Kubernetes cluster for the duration of the test, typically 20 minutes



Functional tests

Regular functional tests are managed through the well-established HammerCloud platform. All object stores used by ATLAS are instances of Ceph. There are currently four deployments at research institutions (AGLT2, CERN, MWT2, RAL) and 1 commercial deployment (LANCS).

Hammercloud functional tests provide a long term view of instance health.

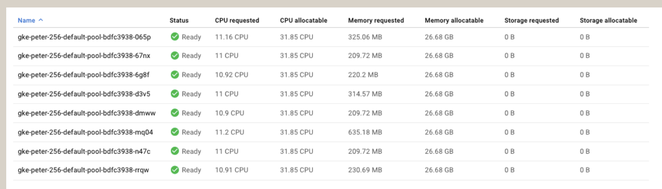


Stress tests

Spinning up an 8-node Kubernetes cluster on Google Compute Engine (GCE) takes only a few minutes. This would provide enough resource (vcpu) for ~1000 client Pods. We have encountered various limits within the GCE which need to be understood prior to scaling further.

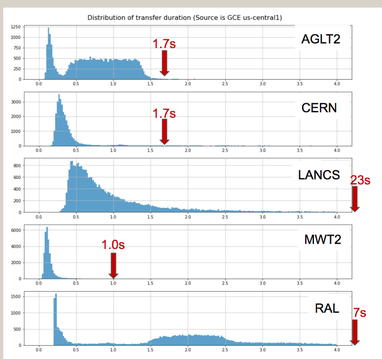
Scaling to 10k connections is planned by simply deploying a larger Kubernetes cluster and different issues are expected to be seen at this level.

Example of an 8-node Kubernetes cluster run on Google Compute Engine.

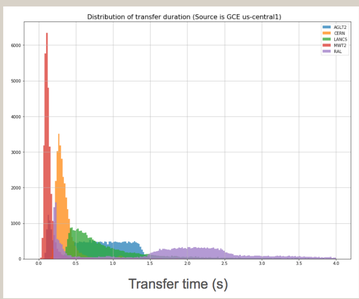


Results

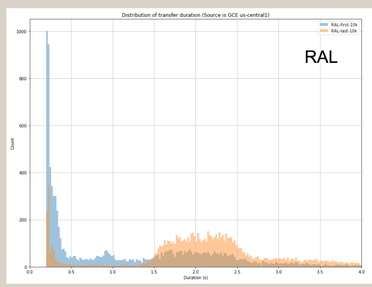
- 30k objects were transferred to each object store
- Client script was a single threaded process running within a Kubernetes Pod with 1000 replica
- Transfer times at the 99th percentile were noted and have large variation, consistent with the latency between client and object store geographical location
- Distribution of transfer times show distinct pattern with AGLT2 and RAL having an interesting plateau at longer times in addition to the spike of quicker transfers.



Site	99th percentile
AGLT2	1.7s
CERN	1.7s
LANCS	23s
MWT2	1.0s
RAL	7.0s

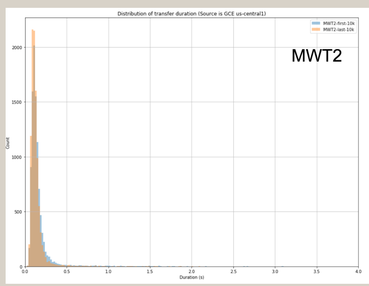


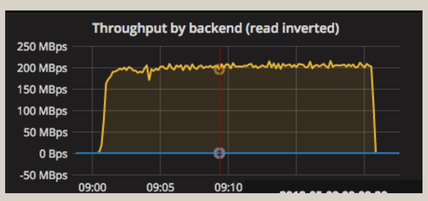
Results



Comparing the first 10k transfer with the last 10k transfers at RAL (left) show a different distribution of times

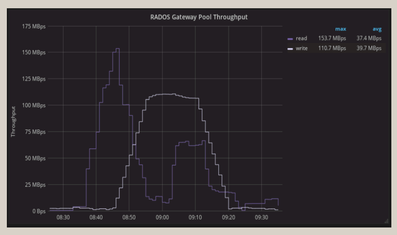
There is no particular reason to see this effect, as shown by the results for MWT2 (right)





Site-based monitoring shows typical throughput bandwidths of 100-200Mbps for a small-scale test of ~1000 simultaneous connections.

AGLT2 (left) MWT2 (right)



We acknowledge the collaboration with colleagues at the following Institutions.