Introduction

This project consisted of many constitute parts concerned with object storage, from integration with high-performance computing, visualisation, cloud technologies and the future of storage at Diamond. These posters will cover various aspects of these projects. This poster will serve as an introduction to the concepts, terms and the current storage at Diamond.

Storage at Diamond

There are different ways to deal with data, at Diamond we currently utilise a high-performance distributed filesystem(s), GPFS[1], to handle the 1GB/s to 40GB/s of unstructured data produced from the detectors at beamlines

What Is A Filesystem?

It is a method of storing data in a hierarchical structure; it works well with data that can easily be categorised. The benefits of this system are that it is human interpretable and works well with data that has an inherent structure to it. The data is stored in Files, Folders, Sub-directories and Directories.

How Object Storage Could Help

Objects stores are ideal for the type of data produced at Diamond. So may prove to be more efficient in a faster and simpler way, whilst being better equipped to cope at scale, due to their history in archival storage and be more capable at handling a broader range of data sizes withe less tuning. Object storage is also closely aligned with the cloud and has a restful interface allowing for integration with remote data analysis

What Is Ceph[2]

Ceph is a storage system that can act as a file, object and block storage. It is actively maintained by Redhat[3] and has a strong scientific community with organisations such as STFC, CERN, the University of Michigan, being associate members[4]. Throughout this project, we use it as an object store exclusively.

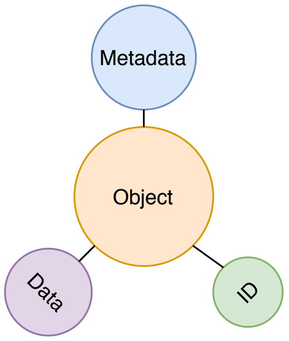
The Problem

Storing unstructured data in a filesystem has problems with complexity, difficulty dealing with large and small data sizes, causing significant input and output overhead and being hard to easily integrate with remote data analysis for the cloud.

What Is Object Storage?

It is a way to handle unstructured data, such as images, audio files, videos and data that that does not have a predefined way of storing it sensibly. The data is stored as an object with three parts the data, metadata (that describes information about it) and a unique identifier.

Example Object



References

[1] R. Jain, P. Sarkar, and D. Subhraveti. "GPFS-SNC: An Enterprise Cluster File System for Big Data". In: *IBM J. Res. Dev.* 57.3–4 (May 2013). ISSN: 0018-8646. DOI: 10.1147/JRD.2013.2243531. URL: https://doi.org/10.1147/JRD.2013.2243531.

[2] Sage A. Weil et al. "Ceph: A Scalable, High-Performance Distributed File System". In: *Proceedings of the 7th Symposium on Operating Systems Design and Implementation*. OSDI '06. Seattle, Washington: USENIX Association, 2006, 307–320. ISBN: 1931971471.

[3] Redhat. *Red Hat to Acquire Inktank, Provider of Ceph*. 2014. URL: https://www.redhat.com/en/about/press-releases/red-hat-acquire-inktank-provider-ceph (visited on 05/28/2020).

[4] Ceph Foundation. *The Ceph Foundation Ceph*. 2019. URL: https://ceph.io/foundation/ (visited on 06/01/2020).

Visualising Object Stores

The project aimed to create a way to visualise tomography data processing in real-time. Achieving this using object storage as the backend, with hopes to showcase the capabilities of the DosNa[1], object storage and to help prove a useful tool for users when running tomography data processing. It was a collaborative project done in part with the Science and Technology Facilities Council (STFC) who created a Raspberry Pi Ceph storage cluster.

Tomography

X-ray Tomography is a non-destructive imaging technique that takes 2-dimensional cross-sectional images at different angles of a sample reconstructing the cross-sections into a 3-dimensional representation of the sample. Applications of tomography are extensive ranging from reconstructing the internal structure of an orchid bee eye to assessing the size and shape of cracks in aircraft parts.

DosNa

DosNa allows for the distribution of N-dimensional arrays over an object storage backend, such as Ceph. It can also be used as a plugin with Savu as a replacement to HDF5[2].

Video Demo

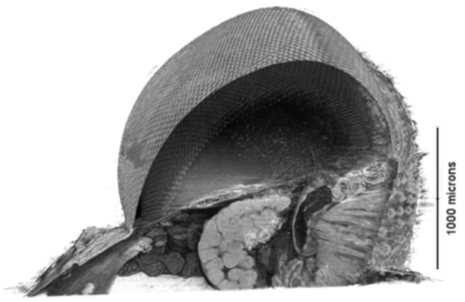
Result

As the video demo displays, this project was a success with results being exhibited in Denver at SuperComputing 2019. This tool will potentially be further developed to form a set of tools for Diamond when using object stores for tomography data processing in future.

Savu

There are many a variety of ways to process tomographic data to get a good reconstruction of the sample, at Diamond, we use an in house program called Savu[3] to do this.

Tomography of A Bee Eye



Web App

The web app allows users to browse objects inside a pool in real-time, viewing slices of the images stored. Due to the way DosNa handles the data by creating a dataset object, which stores the dataset metadata and links the chunks created, we are able to reconstruct the data inside the object store, by calling only the dataset object and taking slices.

References

[1] Scientific Software. *Distributed Object-Store Numpy Array (DosNa)*. https://github.com/DiamondLightSource/DosNa. 2019-NNNN.

[2] The HDF Group. *Hierarchical Data Format, version 5*. http://www.univa.com/products/. 1997-NNNN.

[3] Nicola Wadeson and Mark Basham. , *N-dimensional, Large Tomography Datasets*. 2016. arXiv: `1610.08015 [cs.DC]`.

Image Credit

Tomogram showing a cross-section of a bee's eye. Data collected on the Diamond Manchester Imaging Branchline (I13- 2). Courtesy of Gavin Taylor, Emily Baird, and Andrew J Bodey.

Transient Object Store

This project aims to create an object store that could be deployed on the high performance compute (HPC) cluster, using the RAM on the nodes for storage, as this would allow for fast data processing and quick access to an object-store as well helping show Ceph's ability to scale.

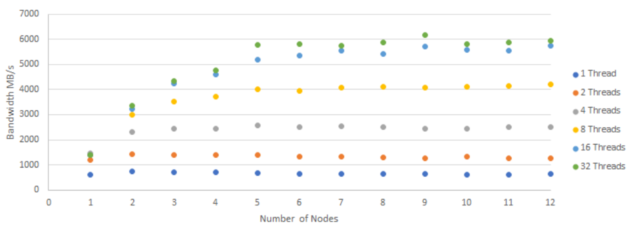
HPC at Diamond

Diamond currently has multiple HPC clusters, which are used for a variety of computing jobs for data processing, in total it consists presently of 8680 cores and 244 GPU's. To manage the job scheduling, we use univa grid engine (UGE)[1]

Submission Script

The submission script works, by utilising Ceph's deployment tool, ceph-deploy, job environment parameters from UGE and Message Passing Interface (MPI)[2] to speed up deployment time. Within the submission script, you can specify how much RAM you want on each host.

Performance



Conclusion

Here we demonstrate that Ceph can be deployed as a cluster job to deal with transient data and scales appropriately across our cluster while leaving room for other jobs to happen at the same time.

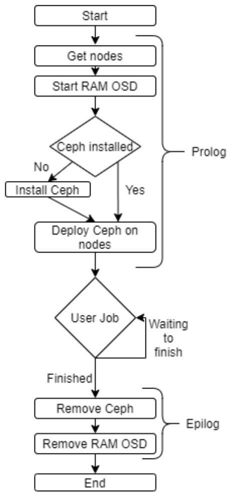
Future Work

Future work includes making the submission script more natural to use and require less knowledge of ceph as well as working closely with MX to see other potential uses for this technology.

Scaling

There are two types of scaling, vertical and horizontal. Vertical consists of improving the capacity of existing hardware to get improvements, whereas horizontal scaling is about adding more machines to work as a single entity. We are interested in horizontal scaling as this is how we use the HPC cluster.

Submission Script



Performance

In the performance graph we demonstrate Ceph's ability to scale to the network limit of $\sim$ 6000MB/s within 5 nodes, a linear rate of approx 1GB/s per node, while not optimised. In this graph, we also show to achieve this performance we only need 16 threads on each node, which means another job can be run on the node at the same time.

References

[1] Univa Engineering. *Grid Engine Users's Guide*. Version 8.5.4.

[2] Message P Forum. *MPI: A Message-Passing Interface Standard*. Tech. rep. USA, 1994.

The Future of Storage

Here we talk about whether object storage has shown to be a viable solution to the problem presented at the beginning and what the future of storage may look like at Diamond.

The Cloud

Currently at Diamond the users employ Diamond compute resource to process their jobs. However, we are starting to move toward remote data analysis, with this comes the issue of moving data around, here object storage shines through with the S3[1] interface.

Problems to Overcome

There are still problems we will need to overcome. Some detectors used at Diamond expect to see a filesystem as a backend, there is not currently a friendly user interface with object storage such as a file browser, and users will need to get used to dealing with objects instead of files.

Future of Storage At Diamond

We believe that due to the prevalence of cloud use and users wanting to transfer data around quickly that object storage will become the go-to storage mechanism within synchrotrons during the next five years.

Overall Conclusion

With the information presented within this series of posters we have shown that object storage can integrate with existing technologies such as Savu, it can act as a distributed RAMDisk, process data at the rates required and neatly integrate with the cloud technologies. Given the research conducted, we would strongly recommend the adoption of object storage, within Diamond and synchrotron facilities.

References

[1] Thomas J. Leeper. *aws.s3: AWS S3 Client Package*. R package version 0.3.21. 2020.

S3

S3 is a data transfer application programming interface (API) built by Amazon and has a restful interface to object stores; most object stores allow for communication via S3. Therefore, making moving data flexible and straightforward for users as there is a near to universal interface for data transfer and interaction across multiple sites.

Future Work

There is now work underway to implement a high-performance object store for the Eiger detectors with MX. However, more work needs to be done into optimising a Ceph, improving DosNa performance and visualisation component,

Closing Remarks and Acknowledgements

This project was a success, helping to demonstrate object storage as a viable solution for data processing at Diamond. However, this would not have been possible without the help of multiple people so we would like to give huge thanks to the scientific computing team, the Ceph user group at RAL especially Thomas Byrne at STFC, the ceph community group, Oliver Freyermuth at university of Bonn and my supervisors Dave Bond and Mark Basham without these people's help and insights this project would not have succeeded in its current form.

diamond