Global EOS: exploring the 300-ms-latency region

To cite this article: L Mascetti et al 2017 J. Phys.: Conf. Ser. 898 062029

View the article online for updates and enhancements.

Related content
- CERNBox + EOS: end-user storage for science
  L Mascetti, H Gonzalez Labrador, M Lamanna et al.
- EOS developments
  Elvin A Sindfiliaru, Andreas J Peters, Geoffray M Addé et al.
- An efficient, modular and simple tape archiving solution for LHC Run-3
  S Murray, V Babyl, G Cancio et al.
Global EOS: exploring the 300-ms-latency region

L Mascetti, D Jericho and C-Y Hsu
CERN European Organization for Nuclear Research, 1211 Geneva, CH
E-mail: luca.mascetti@cern.ch

Abstract. EOS, the CERN open-source distributed disk storage system, provides the high-performance storage solution for HEP analysis and the back-end for various work-flows. Recently EOS became the back-end of CERNBox, the cloud synchronisation service for CERN users.

EOS can be used to take advantage of wide-area distributed installations: for the last few years CERN EOS uses a common deployment across two computer centres (Geneva-Meyrin and Budapest-Wigner) about 1,000 km apart (~20-ms latency) with about 200 PB of disk (JBOD). In late 2015, the CERN IT Storage group and AARNET (Australia) set-up a challenging R&D project: a single EOS instance between CERN and AARNET with more than 300ms latency (16,500 km apart).

This paper will report about the success in deploy and run a distributed storage system between Europe (Geneva, Budapest), Australia (Melbourne) and later in Asia (ASGC Taipei), allowing different type of data placement and data access across these four sites.

1. Introduction
EOS, the CERN open-source distributed disk storage system, provides the high-performance storage solution for high energy physics (HEP) analysis and the back-end for various work-flows. Recently EOS became the back-end of CERNBox, the cloud synchronisation service for CERN users.

EOS can be used to take advantage of wide-area distributed installations: for the last few years CERN EOS uses a common deployment across two computer centres with about 200 PB of raw disks (JBOD), one located in the CERN Meyrin site in Geneva, Switzerland and the second one about 1,000 km apart located in the Wigner Research Centre in Budapest, Hungary. These two centres have a latency of 22-ms and they are connected via two dedicated network links of 100Gbit/s each, the installation of a third link is planned for beginning of 2017.

In late 2015, the CERN IT Storage group (IT-ST) and the Australia’s Academic and Research Network (AARNET) set-up a challenging R&D project: a single EOS instance between CERN and AARNET with more than 300ms latency and 16,500 km apart. Eventually also the Academia Sinica Grid Computing (ASGC) in Taipei joined the R&D providing a fourth site with additional storage capacity for prototype studies of data placement and data access across three continents.

In this report we provide a background description of this activity, its main objective and the success in deploying and running a distributed storage system between Europe (Geneva, Budapest), Australia (Melbourne) and Asia (Taipei).
2. Background
EOS is the main CERN disk storage system characterised by a low-latency hierarchical in-memory namespace. Its main role is to provide disk-only storage optimised for concurrent access[1]. EOS also offers a complete quota systems for users and groups with secure authentication and authorisation. Its development started in mid-2010 and it was deployed as a production service in early 2011.

In 2012 CERN tendered for a remote computer centre to accommodate its growing demand for storage and computing capacity for the Large Hadrons Collider (LHC) programme. The “Call for Tender” was won by the Wigner Research Center for Physics in Budapest, Hungary.

After the decision to use a second computer centre, EOS was optimised for efficiently managing data in different physical locations and providing a single site view to our users[2].

In February 2014 the first “remote” storage space in Wigner was made available inside our EOS production instances and today, with the latest hardware delivery, the EOS raw capacity installed in the two computer centres reached the 50:50 ratio (around 170 PB). The EOS location awareness and the GEO scheduling functionality are fundamental for operating the system across two different data centres with about 22ms round trip (Geneva-Budapest).

3. R&D Collaboration
The challenging and innovative R&D project aiming to federate in a single EOS instance multiple remote storage sites was born by the initial collaboration of CERN and AARNET in data services for science.

Notably both CERN and AARNET share the common interest in scaling ownCloud[1], an open-source collaboration software, to the petabytes range for their users community.

CERN IT in fact provides to its end-users a cloud synchronisation service called CERNBox[3], which allows syncing and sharing on all major mobile and desktop platforms (Linux, Windows, MacOSX, Android, iOS) and provides online and offline availability to any data stored in the CERN EOS infrastructure.

CERN IT storage engineers have been collaborating with their AARNET colleagues and with ownCloud to develop future storage APIs able to cope with truly large fast storage infrastructures, which will provide scientists with scalable access and sharing abilities for multi-petabytes of data on a global basis. These new capabilities will be integrated into AARNET’s CloudStor large file sharing and storage service which is founded on a local implementation of CERN’s EOS storage system.

Later also ASGC colleagues joined the working group for this R&D project, extending the size of the prototype providing storage space in Taiwan and allowing more complex data placement across three continents and four different geographic locations.

4. Main Objectives
EOS has been developed for efficiently managing data in different computer centres and providing a single site view to our users. To achieve this, each storage node is tagged with a “geocode” to identify its location and allow to serve to our clients the closest replica available (avoiding unnecessary remote access). CERN IT is running a “two site setup” in production since 2014 with six EOS instances, one for each of the major LHC experiments, a public instance for non-LHC experiments and an additional one called EOSUSER, used as backend to store user data from CERNBox. The current policy for data placement distributes the two replicas of each file across the two CERN data centres, one replica in Geneva and the second one 1000km away, in Budapest.

1 https://owncloud.org
2 https://cloudstor.aarnet.edu.au
One of the primary goals of this R&D was to test if the EOS software components were able to cope with latencies much higher than 30ms and how the entire software stack was affected by this. In particular, running the software as a single instance across 300ms latency was helping us to explore and discover possible flaws caused by heartbeats, retries and default timeouts in such environments.

An additional metric of the prototype is to measure how easy it is to deploy this global infrastructure and measure if and how the performance of the service is affected. Moreover it is also very important to understand the behaviour of the system and describe how it is possible to improve its performance (hiding network latencies).

The results gathered from this early prototype will be very useful for future strategies based on the aggregation of storage services from smaller sites into a single storage entity. The aim of this R&D is also to improve the performance of the storage, bringing data safely closer to researcher’s workflows instead of requiring the data to be transferred across the globe, providing extremely fast access, as if user’s data was always local.

5. Globally Distributed Architecture

The architecture of EOS has been designed to separate the IO path into metadata access and data access (see Figure 1).

Metadata access is served by the EOS namespace (NS) with the main daemon responsible for the metadata service called MGM. In order to guarantee minimal file access latencies, all metadata is kept in-memory on the server node and persisted on disk as well. To allow communication between the different components and to dispatch messages, a daemon responsible for the message queues (MQ) is also present on the namespace node.

![Figure 1. EOS Architecture.](image)
Figure 2 shows a world map where the components of the prototype were deployed. A server holding the main namespace in read-write mode (MGM master) was configured and deployed in Geneva (with geotag “GVA”), a second namespace in read-only mode and configured as a MGM slave was configured and deployed in Melbourne (with geotag “MEL”). EOS keeps constantly in sync the two namespaces located between 290 and 320 milliseconds away.

CERN provided data storage nodes in Geneva (geotag “GVA”) and Budapest (geotag “BUD”), respectively two data nodes in Meyrin and one in Wigner, with a capacity of 200TB each.

AARNET provided 5 additional data nodes for a total of 15TB of space in Melbourne and ASGC one storage node in Taipei (geotag “TPE”) with 15TB of disk capacity. The relative latencies between our storage nodes are shown in Figure 2 and were computed as averages over time, given the fact that the network underneath was not fully dedicated and that the routing between our endpoints was changing on a daily or weekly bases.

The data access is served via file IO services (FST), here files are generally stored on JBOD disk and the system by default is configured to replicate files in at least two different storage nodes. EOS supports as well additionally erasure-encoding of files with two or three redundancy stripes. All three services (MGM, FST and MQ) have been implemented using the XRootD\(^3\) client-server framework.

\(^3\) http://xrootd.org
6. Data Placement
After the set up of the software components running in all four sites, the storage available was registered in the EOS instance. The namespace was partitioned in a way to accommodate multiple storage pools configured with the desired quality of service, number of replicas and geolocation. Each storage pool was attached to its dedicated area in the namespace, allowing testing of the desired space without reconfiguration of the entire instance, selecting in this way only the relevant part of the namespace.

As shown in Figure 3 several dedicated storage pools were configured in the EOS prototype, some of them containing an homogeneous set of disks. One example is the partition of the namespace /eos/asia/taiwan where we scheduled files for write and read to a storage pool located only in Taipei while /eos/australia/melbourne had a storage pool with disk located only in Melbourne.

Additional pools configured with geolocation and with copies in two or three sites were also generated, one example is the namespace partition /eos/dualcopy/mel-gva, this part of the namespace was storing file in a storage pool containing disks installed in both Geneva and Melbourne. Each file written in this part of the tree was generating two file replicas, one in Australia and the second one in Europe. In this setup, whenever a file is written, EOS schedules the first replica to the “closest” storage (w.r.t. the client) and then triggers a replication of the file to the secondary location. For file reads, EOS will direct the client to the “closest” available replica present in the system.

To explore all the possibilities available in this prototype, we also generate storage pools containing three geolocations, this is the case for all the spaces created under /eos/triplecopy/. The most interesting setup is the one mapped under the namespace mel-gva-tpe, as its name suggests its behaviour is the following: for each file created, EOS generates one replica in Melbourne, one replica in Geneva and one in Taipei.

![Figure 3. Global EOS: namespace and storage mapping structure.](image-url)
7. Results

Thanks to the very good collaboration between CERN, AARNET and ASG the setup of the instance was easy and it took quite a short amount of time, basically just the time to procure the physical hardware. Moreover the creation of this global EOS setup helped us make the software even more robust in handling nodes installed in separate network domains.

One of the interesting things that we immediately noticed in this global setup was the network routing between all the nodes of the system: routing in the global instance was not symmetric, as you can see in Figure 4, and it was changing on a daily bases. In one of our first measurement, the latency between Geneva and Melbourne was around 296ms, the day after the same latency was measured to be 320ms and the network path used changed from passing through Asia to another network path passing through USA, crossing both Atlantic and Pacific Oceans.

On the storage operation side we confirmed the stability and the robustness of EOS in working with such latency, no adaptation of timeouts or other parameter was needed in order to set up the system on this very large geographical scale, the system worked immediately out of the box. Functional tests were run successfully from clients in all four locations against all available storage pools. Local transfers reached the expected performance of the protocol, an example are transfers from clients in Melbourne to the Melbourne pool. In this example the client contacted the read-write namespace located in Geneva and then the data transfers is scheduled to a Melbourne disk. At this point the client pushed data directly to storage nodes located in the same geographical area reaching in average speed around 550MB/s for 1GB files. For these types of file write cases we notice some latency effect on the total time of the transfer, in fact the authentication phase of the transfers was adding three times the latency from the client to the namespace to be authorised to perform the transfer, lowering the effective average transfer speed to 470MB/s. Three time the latency is due to the number of exchanges (three) necessary between the client and the server to collect their authentication, validate their identity and then proceed with the data transfer. The file read use case does not see the same latency effect since the authentication in this case is delegated to the read-only namespace located in the same site and is not affected by such a big round trip time.

Concerning transfers to remote pools we highlight the example of the write case from
Melbourne to Geneva. Here the client located in Melbourne contact the namespace in Geneva to store the file and it is then redirected to the storage pool in Geneva, transferring the data across 16500km. In this case, in addition to the latency effect in the authentication phase, we noticed as well the effect of the TCP windows. With a standard installation of RHEL6 and without any performance tuning (both on client and server nodes) the effective average speed for these types of data transfers was around 45MB/s. We estimate that for a production setup with network tuning the ultimate performance will be much higher.

The last transfer test we report in this article is the three replica write and read. For this test we tried to write and to read using clients in each of our sites to the three replica pools with storage located in Geneva, Melbourne and Taipei. The configured behaviour of this pool is to generate a replica for each file written in all three locations. What we saw from these test results was that the client was configured to schedule the first replica to the "closest" site while the second and third site where chosen using the standard scheduling policy and not using a geographical order, this resulted in having non homogeneous replica order (e.g. with a client in MEL the possible order for the replication were the following: MEL-TPE-GVA and MEL-GVA-TPE). This behaviour was not disruptive but was introducing some noise in the test. The average effective speed measured in this setup with the creation of three replicas was around 35-40MB/s. This behaviour has been reported to the development team and fixed, it will not be present in the future versions of EOS.

8. Summary and Outlook
For this R&D project we successfully investigated the EOS storage capabilities across distributed computer centres connected with very high latencies. EOS is a key service in the CERN IT Storage strategy to further innovate end-user services. CERN together with AARNET and ASGC engineers collaborate around the globe on the open development of this project for the benefit the research and education community.

This first prototype built the foundation of a new way to use shared file systems on a global scale, to improve productivity of researchers from Australia to Europe or Asia and vice versa. In the future, storage providers will be able to place users’ data closer to them, in order to follow them safely, providing an extremely fast access, as if the data was always local.

The R&D showed a fully working EOS setup out of the box without the need of any particular ad-hoc settings to copy with these high latencies. The resources provided by CERN, AARNET and ASGC were fully functional and registered correctly in the instance to accomplish this global scale test.

We investigate the behaviour of the configured storage pools with clients locate in all four sites and we measure how the file transfers are affected by the latency and which are the measure that can be used to fully or partially compensate for this effect. In future tests we plan to optimise the cluster network setting in order to better perform with high latency links, in particular we see an advantage to modify the TCP settings and for example in increasing the default TCP window size, since it has a negative effect during the start of the transfer and in the case of packet loss.

The next step for this fruitful collaboration is to extend the storage environment to include the United States and another Asian end-point. This environment will help us to test the new geo-scheduling fine-grained policy features available in the new version of EOS (codename citrine). In addition it will be used as a playground to improve the sync and share experience of our respective cloud storage services (CERNBox, CloudStor and ASGCBox).

In the future we also envisage to extend this type of R&D to the High Energy Physics community given the advantages it has in operating remote storage at lower man-power costs and federating smaller sites together in a single instance.
A huge thanks to all the member of this collaboration for their interest in this project, their help and for working with us from different time zones.

References