EPJ Web of Conferences **295**, 01009 (2024) https://doi.org/10.1051/epjconf/202429501009
CHEP 2023

Automated Network Services for Exascale Data Movement

Justas Balcas[1,*], *Harvey* Newman[1], *Preeti* P. Bhat[1], *Frank* Würthwein[2], *Jonathan* Guiang[2], *Aashay* Arora[2], *Diego* Davila[3], *John* Graham[3], *Thomas* Hutton[3], *Tom* Lehman[4], *Xi* Yang[4], *Chin* Guok[4], *David* Alexander Mason[5], *Oliver* Gutsche[5], *Phil* DeMar[5], *Chih-Hao* Huang[5], *Syed* Asif Shah[5], *Dmitry* Litvintsev[5], *Ryan* Heath[5], and *Andrew* Malone Melo[6]on behalf of the CMS Collaboration

[1]George W. Downs Laboratory of Physics and Charles C. Lauritsen Laboratory of High Energy Physics 1200 E California Blvd Pasadena, California, USA 91125
[2]Department of Physics, UC San Diego, 9500 Gilman Drive, La Jolla, California, USA 92093
[3]SDSC, UC San Diego, 9500 Gilman Drive, La Jolla, California, USA 92093
[4]Energy Sciences Network, Lawrence Berkeley National Laboratory, 1 Cyclotron Rd, Berkeley, California, USA 94720
[5]Fermi National Accelerator Laboratory, Kirk and Pine St, Batavia, Illinois, USA 60510
[6]Department of Physics & Astronomy, Vanderbilt University, Nashville, Tennessee, USA 37235

Abstract. The Large Hadron Collider (LHC) experiments distribute data by leveraging a diverse array of National Research and Education Networks (NRENs), where experiment data management systems treat networks as a "blackbox" resource. After the High Luminosity upgrade, the Compact Muon Solenoid (CMS) experiment alone will produce roughly 0.5 exabytes of data per year. NREN Networks are a critical part of the success of CMS and other LHC experiments. However, during data movement, NRENs are unaware of data priorities, importance, or need for quality of service, and this poses a challenge for operators to coordinate the movement of data and have predictable data flows across multi-domain networks. The overarching goal of SENSE (The Software-defined network for End-to-end Networked Science at Exascale) is to enable National Labs and universities to request and provision end-to-end intelligent network services for their application workflows leveraging SDN (Software-Defined Networking) capabilities. This work aims to allow LHC Experiments and Rucio, the data management software used by CMS Experiment, to allocate and prioritize certain data transfers over the wide area network. In this paper, we will present the current progress of the integration of SENSE, Multi-domain end-to-end SDN Orchestration with QoS (Quality of Service) capabilities, with Rucio, the data management software used by CMS Experiment.

1 Introduction

The overarching goal of SENSE [1] is to enable National Labs and Universities to request and provision end-to-end intelligent network services for their application workflows, leveraging SDN capabilities. Scientific collaborations often see the network as an opaque resource that

*e-mail: jbalcas@caltech.edu

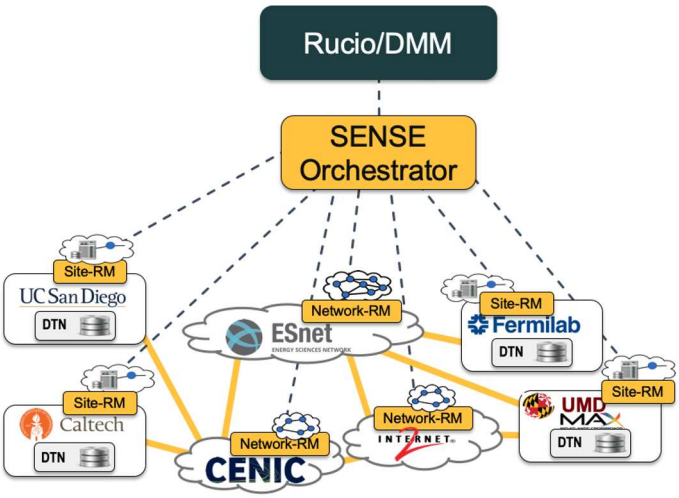

Figure 1. SENSE Architecture for US-CMS Deployment and current status of deployed Site, Network Resource Managers, and Orchestrator. It includes testbed resources at the following Sites: Tier 1 at Fermilab, Tier 2 at UCSD and Caltech, and Tier 3 at UMD. Additionally - network control over CENIC, ESnet, and Internet2.

helps them to move data from one place to another, always with the hope of receiving an acceptable quality of service, in other words, the Data Management Systems (like Rucio [2]) do not interact in any given form with the NRENs (National Research and Education Network). In turn, the Network does not know the priorities of the Data Management Systems, so it is forced to treat all data transfers equally regardless of their importance.

This work presents our approach to interfacing SENSE with the Rucio, the Data Management System of the CMS [3] collaboration. By connecting Rucio with SENSE, we can create network services for specific data workflows based on experiment priorities. The above allows for priority workflows to travel over the network a) completely isolated, b) bandwidth-guaranteed, and c) on predefined network segments, which translates into a) improved accountability, b) accurate time to completion, and c) better monitoring, respectively.

2 Infrastructure

Our testbed makes use of a debug instance of SENSE, our own instance of Rucio, an FTS debug instance at Fermilab, and two dedicated XRootD[4] instances deployed at two different institutions: Caltech and UCSD. These two storage systems are connected over a dedicated 400Gbps link. SENSE controls all components within this path, including the storage system's data nodes, the site's network switches, and the NREN routers along the path. As shown in Figure 1 - we built our prototype on top of the existing SENSE Architecture.

All the components in the testbed can be easily separated into three groups: 1. All SENSE components: Orchestrator, Network-RM, Site-RM, 2. The experiment or application side: Rucio, FTS, and XRootD, and 3. The Data Movement Manager (DMM) which connects the previous two groups. In the following sections, we briefly describe these components separately.

2.1 SENSE

SENSE operates between the science workflow agents/middleware and the automation Layer controlling the individual networks and end-site resources. Its core mission is to enable customized end-to-end service provisioning and management across multi-domain, multi-resource distributed infrastructures. Within the SENSE architecture, there are two distinct functional roles: Orchestrator and Resource Manager (RM). The interaction of Orchestrator(s) and RM(s) follows a hierarchical workflow structure whereby the Orchestrator accepts requests from users or user applications, determines the appropriate RMs to contact, and coordinates the end-to-end service request. The RMs are (administrative or technology) domain-specific and are responsible for configuring and managing local resources. The SENSE architecture is designed to be flexible and scalable, allowing for many-to-many relationships between Orchestrators and Resource Managers and supporting multi-level Orchestor hierarchies. More details about SENSE Architecture and its components can be found in this paper [1]

2.1.1 SENSE Modifications

To support this new development and network control, e.g., Layer 3 traffic re-routing at the Site Border, the SENSE Software suite was modified to provide a custom set of services to the Rucio/DMM system based on the following requirements:

- The Orchestrator obtains information from all the Site-RM about sites which include: sites available for service, Layer 3/BGP (Border Gateway Protocol) control, IPv6 subnets available for traffic engineering, site network connection speed (available, allocated, used), and site network topology.
- The Orchestrator processes Rucio/DMM requests for priority services which results in the following actions:
  - The Orchestrator interacts with all of the Network-RMs and Site-RMs in the path to instantiate a priority path for a specific data transfer and ensures that Layer 3/BGP is configured.
  - The Rucio/DMM system takes information provided by the Orchestrator to initiate a data transfer via the standard FTS mechanism which uses the priority path configured by SENSE.

The SENSE Network-RM service used for this use case is the standard Layer 2 Point-to-Point service, which creates a priority service between two sites. In this manner, a Layer2 network path is established which has a guaranteed bandwidth. This use case did not require any changes to the existing SENSE Network-RM-based services.

The SENSE Site-RM service did require changes to process service-specific information from the Orchestrator, to direct the specific traffic onto the priority network paths from the associated sites. This included Site-RM modifications to use Ansible and control network devices locally at the Site. This included the development of new Ansible modules for the following devices: Dell OS 9 [5], Arista EOS [6], and SONiC [7]; This service provides the ability for the Orchestrator to request traffic re-route via an initiated L2 path for a specific range of IP addresses. Also, Site-RM only supported Layer 2 QoS (Quality of Service) and it was extended to cover Layer 3 as well, using Linux Traffic Control (TC) [8]. Additionally, we implemented a monitoring feature in Site-RM to provide a complete picture of the site in terms of its performance and various (CPU, IO, storage, network links) load levels, updated in real-time via Prometheus, Node Exporter [9, 10].

2.2 Rucio, FTS, and XRootD

When it comes to data distribution, 3 main systems take care of this task within CMS:

- Rucio. The Data Management System keeps track of all of the datasets in the experiment, the replicas that exist, and where they are located i.e., which sites have them stored.
- FTS. This system manages the queue of Third Party Copy (TPC) transfers requested from Rucio. TPCs are file transfers between 2 sites that are orchestrated by a third party, in this case, FTS plays the role of that third party.
- Storage Systems. These systems reside in the Sites and are in charge of storing the experiment's data. They also provide an interface to support TPCs. Within CMS, there are several different implementations of these systems; XRootD is one of them.

2.3 Data Movement Manager

DMM is a homemade software component that handles the communication between Rucio and SENSE; it is responsible for obtaining the different endpoints of each RSE and keeps track of which of them are in use. Also, it calculates the priority path's bandwidth and adjusts it when needed. Finally, it takes care of negotiating the allocations with SENSE.

Figure 2 depicts the interaction between Rucio and DMM. The step marked as "i" in the diagram is a one-time interaction used by DMM to obtain information about the sites, like their available endpoints and network capacity. The rest of the graph describes the interactions triggered by the creation of a new rule in Rucio and it goes as follows:

1. A new rule is created in Rucio, and it reaches the Preparer phase;
2. DMM gets the information about the rule: number of files, total size, sites involved, and priority. If the rule has no priority, DMM sends back the default endpoints from each Site involved, and the data flow continues normally.
3. If the rule has a priority, DMM will pick one unused endpoint for each of the sites involved and calculate the rule's bandwidth based on its priority and those of other active paths between the same pair of sites;
4. DMM sends a request to SENSE to create a priority path between the two endpoints with the bandwidth allocation chosen in the previous step;
5. DMM provides Rucio with the endpoints selected in step 3;
6. Rucio requests the set of TPC transfers to FTS the same way as it is done in production;
7. SENSE orchestrates the required network services to create the priority path.

3 Current Status

At this time, all necessary features have been implemented to enable Rucio to trigger the creation of of automated network services to create priority paths between our test sites: UCSD and Caltech. This includes allowing Rucio to prioritize and re-route traffic based on priorities. As Shown in Figure 3 best-effort traffic (green) flow priority is decreased, while the priority path (purple) uses the full link capacity between 2 Sites. From a high-level view, we can say that Rucio requests are automatically translated into QoS parameters that are applied both by Network-RM and Site-RM on the Network and the Sites, respectively.

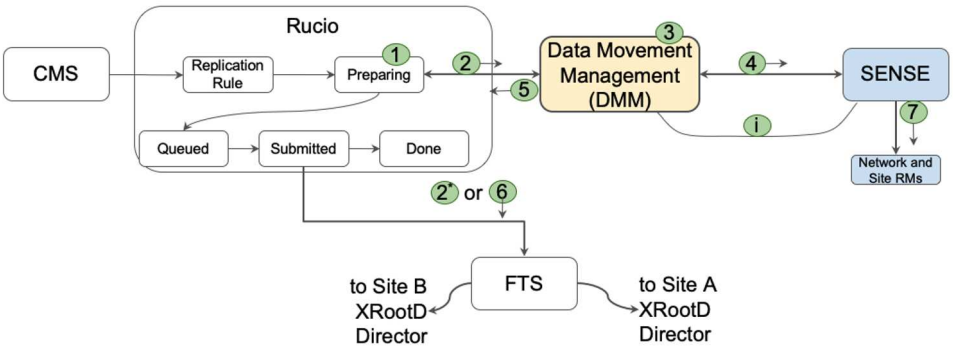

Figure 2. SENSE and Rucio interaction through DMM

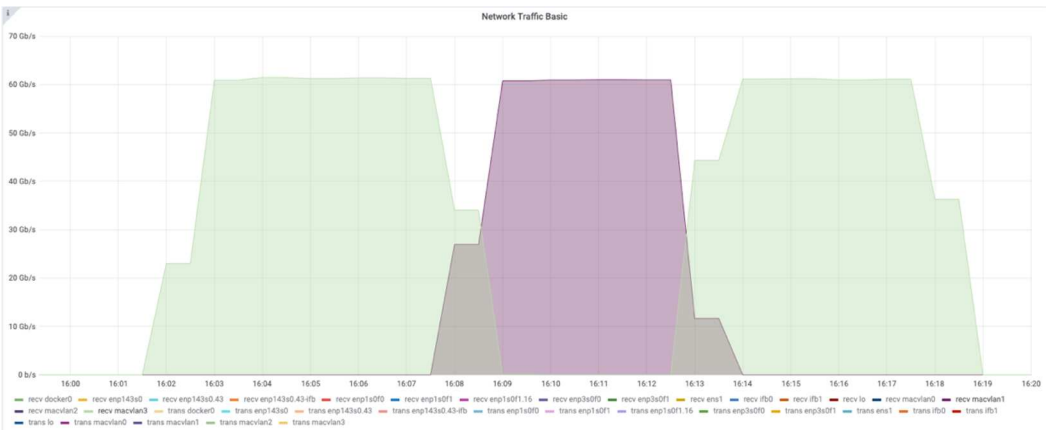

Figure 3. Measured and Prioritized Quality of Service traffic from UCSD to Caltech.

4 Issues and progress

This section will describe the progress and work done on the project's different components since the successful achievement of a PofC (Proof of Concept) published here [11].

4.1 The scale issue

One of our new objectives, which proved to be harder than we thought, was to replicate the PofC on a larger scale at 400Gbps. We rapidly ran into two different problems when trying to achieve the above: 1. The submission rate from Rucio to FTS and from FTS to XRootd was not fast enough to reach the amount of simultaneous active transfers needed to achieve 400Gbps usage of the network. 2. The QoS implementation in the Data Nodes consumed too much CPU, which lowered the sites' capacity to sustain the amount of active transfers needed to reach 400Gbps.

The first issue can be addressed by a combination of 1. creating and configuring a dedicated submitter daemon in Rucio to only look at priority requests and submit those at a faster rate, and 2. Implementation of a mechanism in DMM to modify, on demand, the limits that control the number of active transfers in FTS via its REST API. The second issue is noticeable once the expected throughput from a single server is 100Gbps or more. Due to QoS implementation, all traffic must go via kernel user space for traffic shaping, and queueing and high-clock CPU speed is needed. For the following issue, we envision requiring this to be done at SMART NICs (like Alveo U280) or using eBPF (extended Berkeley Packet Filter), which allows packet processing at the NIC (Network Interface Card) level. Additionally, Site RM implements QoS at the Server level but not on the local network devices at the Site. For the latter one, we envision modifying Site-RM and use rate policy features, which most of the Enterprise-grade network equipment already supports.

4.2 Multus

During the PofC tests, we ran into a problem where the XRootD instance at the destination side of a TPC would pick the default routable IP address on the host to establish the connection with the source site instead of using the IP that FTS used to communicate with it. Since the SENSE network services are based on the IPs, we needed a way to instruct XRootD to use the correct IP address. At the time, we could work around this problem by having Site-RM add a Routing rule in the host to use the desired local IP based on the TPC's destination IP. The above solution works in a Bare-metal/Docker-based deployment, while we noticed it to be complex to deal with Routing tables on Kubernetes-based systems. For Kubernetes-based clusters, a better solution is to use Multus [12] CNI (Container Network Interface). Multus is a Kubernetes [13] plugin that allows us to create k8s (short for Kubernetes) pods with a single IP address that is isolated from the rest of the addresses in the host. This is, in a node with N IP addresses, we can deploy N k8s pods, each of them running one XRootD instance with a single IP, which means it has its own routing table and rules. That solved an issue for Kubernetes-based deployments and for bare-metal/docker-based installations - we continue to support a Routing rule modification at the host level.

4.3 Monitoring

An important objective of this project is to increase accountability; to that end, significant effort has been put into implementing its monitoring mechanism. We decided to start using a public FTS instance, as opposed to using our own, in order to get access to the FTS monitoring records via CERN's MonIT infrastructure [14]. We use MonIT's grafana proxy [15] to pull FTS records and compare them with the monitoring data we obtain from Prometheus/Node exporter available via Site-RM. This way we can compare the network traffic in the host and local network interfaces with the throughput reported by FTS for the set of transfers going through a given priority path created by SENSE.

4.4 Rucio and DMM changes

One of the long-standing objectives of the project is to include all DMM capabilities either on Rucio or SENSE to this end, it is important for us to keep our Rucio instance as close to the the upstream version as possible. With this in mind, we put significant effort into adopting Rucio's official Kubernetes implementation based on Helm charts [16]. This will keep our project up to date with changes to the Rucio base code.

Following the changes on our Rucio deployment described above, DMM was modified to support having the different Rucio Daemons running in separate pods/containers. The previous version of DMM worked assuming that all the Rucio daemons were executed within the same pod/container as DMM.

4.5 Growing the testbed

Adding more sites to our testbed opens the door for testing more complicated scenarios, like creating multiple priority paths over a single network segment that is shared by more than 2 sites. For that reason, significant effort has been put into reaching out to more sites to collaborate with the project. Currently, we are in the process of including testbed resources at Vanderbilt, Nebraska - where testbed resources are as close to production deployment as possible.

5 Future plans

The following is the list of planned upgrades for SENSE that will be beneficial to this project:

- Adding a new Network Service: IP Traffic Steering

  This will allow establishing of site-to-site priority paths across the Wide Area Network (WAN) using Layer 3 traffic as opposed to Layer 2 VPN over WAN + Layer 3/BGP at the Site Level. This will provide an option for those sites that have reduced control of their network.

- Full Life-Cycle Service Monitoring and Troubleshooting

  The Orchestrator currently obtains information from all of the Resource Managers across the end-to-end path for service instantiation. This will be extended to include monitoring and troubleshooting data during the full life cycle of the service.

- Site-RM network device control

  Site-RM currently is limited to devices highlighted in 2.1.1 and will be extended to support more network device control, like Dell OS 10 [17], FreeRTR [18], and Juniper [19], as this is needed for deployment at new Sites.

- Policy Share and QoS As more and more bandwidth from the NRENs is controlled by SENSE an agreement among the different stakeholders is necessary to define the policies and implement the mechanisms required for a fair share of the network resources.

In the short term, we expect to get all of the US-CMS Tier2 Sites to form part of our testbed. This will help us gain experience and increase the system's overall stability. In the longer term, we see DMM integrated into CMS Rucio and interacting with a production SENSE instance that orchestrates priority paths amongst a significant percentage of all CMS sites.

6 Acknowledgments

This ongoing work is partially supported by the US National Science Foundation (NSF) Grants OAC-1836650, and PHY-2121686. In addition, the development of SENSE is supported by the US Department of Energy (DOE) Grants DE-AC02-05CH11231, DE-SC0015528, and Caltech ESnet agreement - LBNL-767694. Finally, this work would not be possible without the significant contributions of collaborators at ESnet, Caltech, SDSC, and Fermilab.

References

[1] I. Monga, C. Guok, J. MacAuley, A. Sim, H. Newman, J. Balcas, P. DeMar, L. Winkler, T. Lehman, X. Yang, Future Generation Computer Systems **110**, 181 (2020)
[2] M. Barisits, T. Beermann, F. Berghaus, B. Bockelman, J. Bogado, D. Cameron, D. Christidis, D. Ciangottini, G. Dimitrov, M. Elsing et al., Computing and Software for Big Science **3**, 1 (2019)
[3] *CMS Collaboration, The CMS experiment at the CERN LHC, JINST 3 (2008) S08004*, doi:10.1088/1748-0221/3/08/S08004
[4] A. Dorigo, P. Elmer, F. Furano, A. Hanushevsky, WSEAS Transactions on Computers **1**, 348 (2005)
[5] *Dell os 9*, https://www.dell.com/en-us/shop/ipovw/networking-os9
[6] *Arista eos*, https://www.arista.com/en/products/eos
[7] *Sonic github*, https://github.com/sonic-net/SONiC
[8] *tc - linux manual page*, https://man7.org/linux/man-pages/man8/tc.8.html
[9] *What is prometheus?*, https://prometheus.io/docs/introduction/overview/
[10] *Monitoring linux host metrics with the node exporter: Prometheus*, https://prometheus.io/docs/guides/node-exporter/
[11] F. Würthwein, J. Guiang, A. Arora, D. Davila, J. Graham, D. Mishin, T. Hutton, I. Sfiligoi, H. Newman, J. Balcas et al., *Managed Network Services for Exascale Data Movement Across Large Global Scientific Collaborations*, in *2022 4th Annual Workshop on Extreme-scale Experiment-in-the-Loop Computing (XLOOP)* (2022), pp. 16–19
[12] *Multus-cni*, https://github.com/k8snetworkplumbingwg/multus-cni
[13] *Kubernetes documentation*, https://kubernetes.io/docs/home/
[14] *Monit home page*, https://monit.web.cern.ch/
[15] *Monit docs: Grafana proxy*, https://monit-docs.web.cern.ch/overview/access/#grafana-proxy
[16] *Helm documentation*, https://github.com/rucio/helm-charts
[17] *Dell os 10*, https://www.dell.com/en-us/shop/ipovw/open-platform-software
[18] *Freertr project website*, http://www.freertr.org
[19] *Juniper - junos operating system*, https://www.juniper.net/us/en/products/network-operating-system/junos-os.html