IRIS-HEP Fellowship Proposal:

Add network measuring to data federation caches

Project Area: Data Organization, Management and Access (DOMA)
Ziyang Ye, UW-Madison
Mentor: Brian Lin, UW-Madison

The High-Luminosity Large Hadron Collider (HL-LHC) will bring a tremendous increase in data volume. Experiments have already begun to deploy distributed data federations consisting of caches and origins to efficiently transfer and reuse data. Research and data transfer is heavily dependent on network infrastructure and almost always across administrative domains. It is critical to ensure that the transfer and reuse of data works well in distributed data federations. Therefore, adding network performance tooling for data caching federations are needed.

The data federation is provided by Open Science Grid (OSG). Researchers store data on origin servers. Data is streamed from origin servers to distributed OSG cache servers all over the country. Researchers can achieve lower latency and higher data transfer throughput by using a cache server near their location rather than accessing the origin server directly. Each location can choose to build its own cache to reduce the amount of data transferred over the WAN, which is accomplished by building containers using docker. Docker containers are an open-source application container engine that allows developers to package their own applications in a unified form, then wrap them in a portable container and later publish them to any server where the Docker engine can be installed. Docker containers have little to no performance overhead, can easily run-on machines and data centers, and their best feature is that they do not rely on any system, framework or even language. As a result, researchers can build caches and access to data efficiently with docker.

perfSONAR will be used to build the network performance tooling. It is a network measurement toolkit for testing and sharing data on end-to-end network performance. Many problems arise when data is transferred between locations. When network performance is poor or errors occur, it can be difficult to identify and correct the root cause of the problem because the problem can occur in multiple networks. It is often impossible to find the crux by detecting local approaches alone. Therefore, perfSONAR will be used to create detection tools. perfSONAR will automatically test and monitor all nodes on the network path and look for areas of low performance. perfSONAR will send test configurations from the primary test site to other nodes around the world for testing after the tests are created. The test data will be stored on hosts everywhere. In this way, the performance differences between locations can be determined.

The first phase of this project is to deploy cache and perfSONAR containers. The second phase is to deploy the monitoring software using Kubernetes to orchestrate cache and perfSONAR containers. Kubernetes is a management system that provides more advanced and flexible management of Docker and containers. Kubernetes consists of the control node and worker nodes. The control node manages and controls the whole cluster, basically all the control commands of Kubernetes are sent to it, and it is responsible for the specific execution process. Each worker node is assigned some Docker containers by the control node. In this way, the cooperation between cache containers and perfSONAR containers can be accomplished.

Software deliverable:
A network measuring tool for data federation caches will be delivered by the end of August. It is developed using Kubernetes deployments that orchestrate cache and perfSONAR containers to monitor the data transfer between origins and caches.

Timeline:

- May 30th - June 6th: Familiarize with running a cache container
- June 6th - June 13th: Deploy a cache in containers
- June 14th - June 20th: Research on perfSONAR container deployments
- June 20th - June 28th: run a perfSONAR container locally
- June 28th - July 31st: Orchestrate cache and perfSONAR containers together using Kubernetes deployment
- August 1st - August 14th: Validation of the monitoring data in caches
- August 15th - August 30th: Complete the documentation and present the final work