FERMILAB-POSTER-19-089-CD

Unification of Monitoring in the Fermi File Transfer Service

B. Damman[1,2], R. Illingworth[2], M. Mengel[2], B. White[2]
[1]Des Moines Area Community College - CCI Program, [2]Fermilab

Program Background

The Fermi File Transfer Service (FTS) is a commonly used program by Fermilab to safely transfer files from experiments to an experiment's desired storage location. FTS observes a specified local directory for files to transfer safely to storage (as represented in Figure 1). These files are cataloged into the SAM Web Server, which preserves both file metadata and replica information.

The goal is to provide enhanced monitoring to compliment FTS's pre-existing monitoring which will enable new ways to visually graph and interpret logs in a non-local environment.

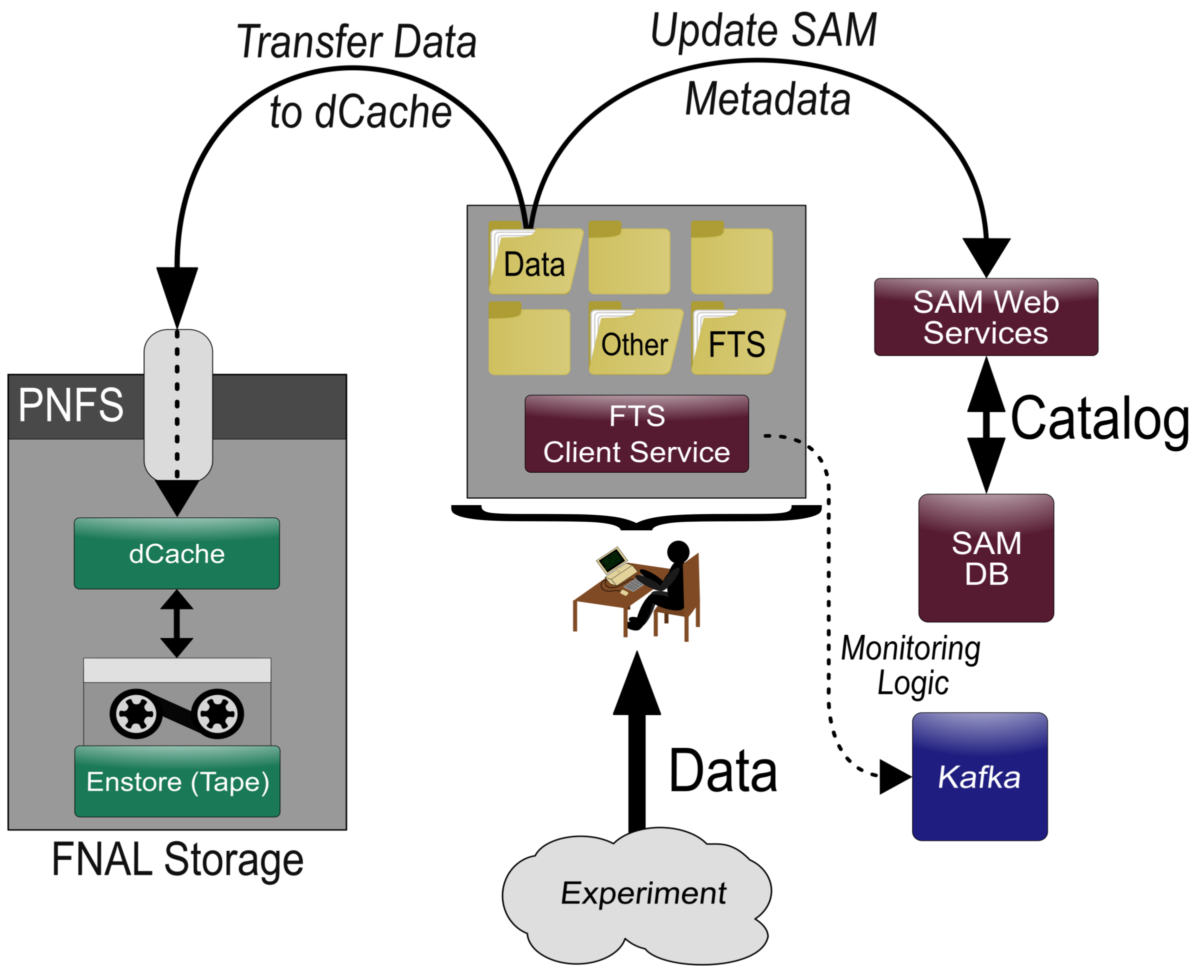

Figure 1. The overall program flow of the Fermi File Transfer Service (FTS) and Sequential Access via Metadata (SAM).

Data is collected and safely transferred to storage (dCache for instance) and metadata is updated to SAM. A Kafka interface was newly added, which receives information about each process.

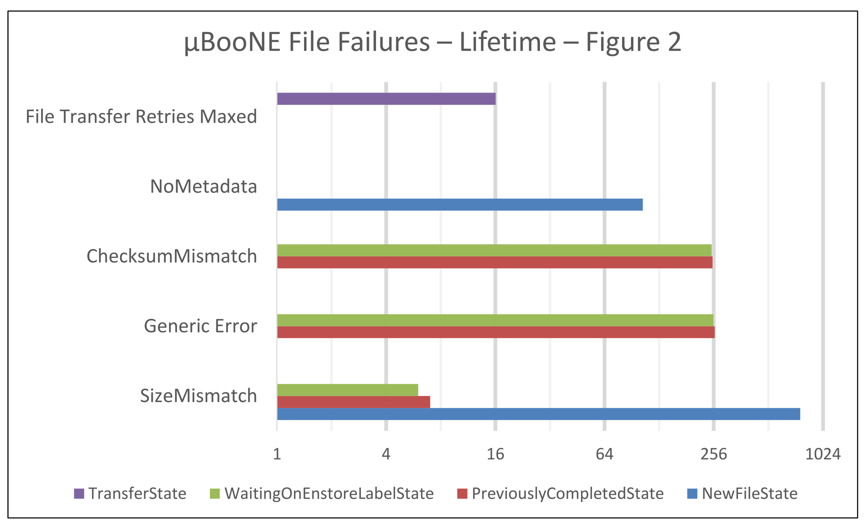

Figure 2. Graph generated from monitoring data pertaining to how many files have thrown errors in µBooNE's FTS Logging lifetime. This graph can be limited to minutes, hours, or days.

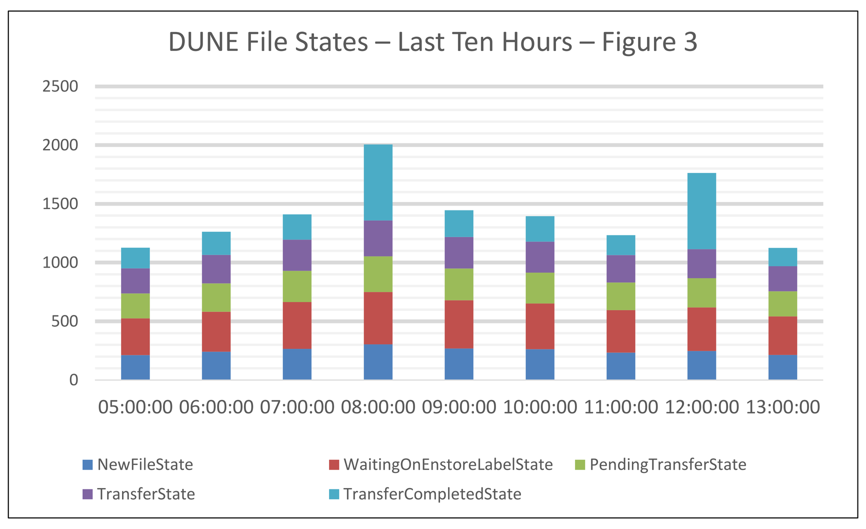

Figure 3. Graph generated from monitoring data pertaining to the number of files in each respective state in a ten-hour period. Based on all file types used by DUNE.

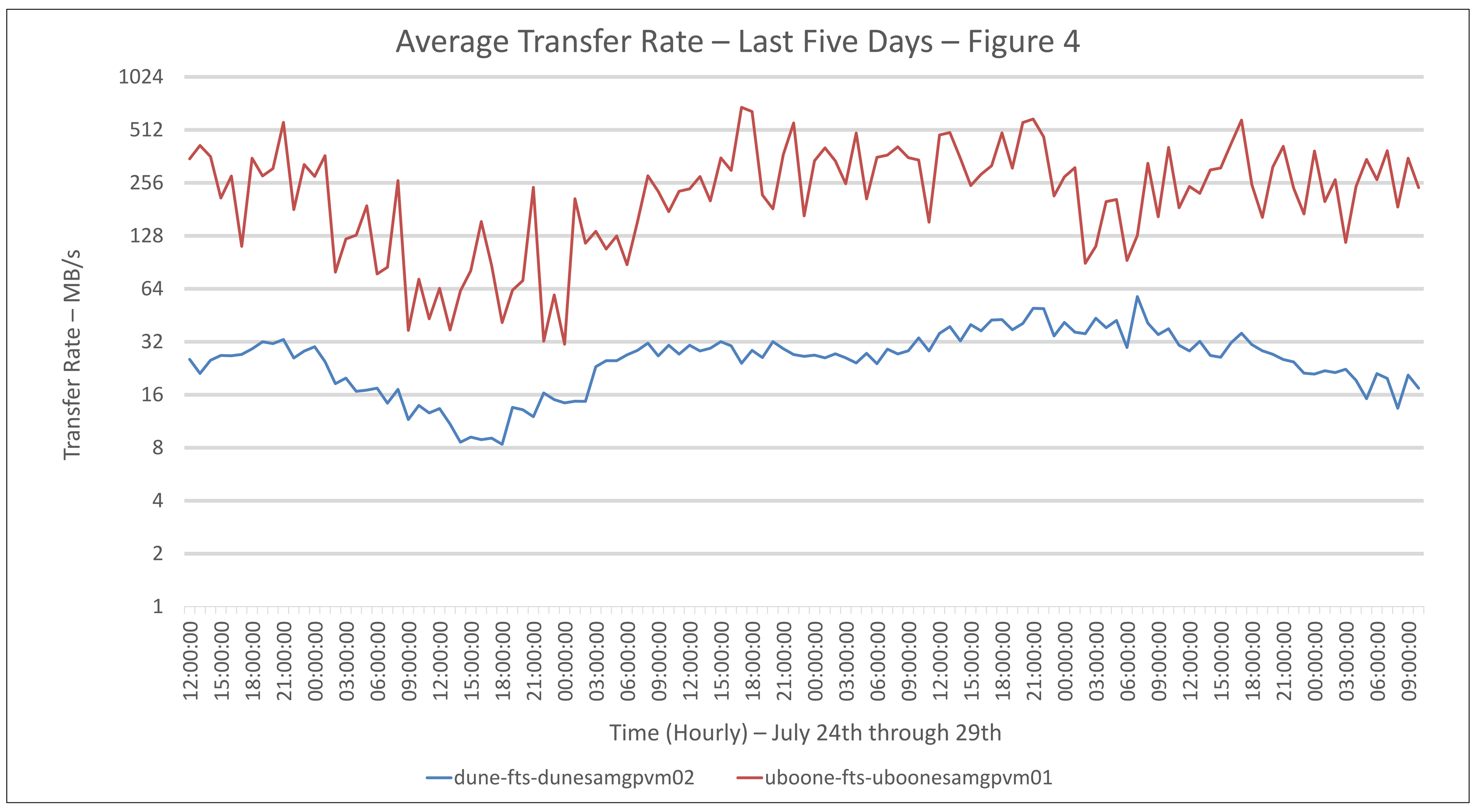

Figure 4. Graph generated from monitoring data detailing the average transfer rate (calculated as megabytes per second) for every file that successfully completed a transfer on an hourly basis. Both DUNE and µBooNE are represented on the graph.

Logging Centralization

To improve monitoring, a new component will be added to the source code of FTS. Monitoring can be done either locally or remotely. FTS will perform remote logging to avoid filling a host machine's hard drive with logs and to provide a centralized location for all recently updated services.

Message Delivery

As an extension from FTS's native source, the Kafka-Python library is currently used to send encoded JavaScript Object Notation (JSON) messages to a message broker. Each event that occurs in relation to a file's lifecycle is monitored. For events that are of interest (new files, metadata changes, transfer completion, etc.), messages are built with information regarding the file's name, state, size, and so forth. If an error is raised, the message will detail the error type and a brief description.

Messages sent to a specified message broker are consumed by Elasticsearch. Elasticsearch persists all consumed messages for parsing and visualization purposes.

Through Elasticsearch, persisted messages can be queried using NoSQL via the web interface on Landscape or graphically represented like Figure 2 through 4. Additionally, all query results can be exported for local data analysis.

Results

As shown by Figures 2 through 4, by adding the monitoring logic it is now possible to graphically represent the monitoring information consumed by Elasticsearch for provision of statistical information regarding any number of experiments. Additionally, systems running FTS are now able to provide detailed error messages for troubleshooting systems.

With the success of this project, the SAM Web Server will be updated with its own version of monitoring logic through a similar process to provide the same support as FTS now does.

DMACC DES MOINES AREA COMMUNITY COLLEGE Life's Calling™

This manuscript has been authored by Fermi Research Alliance, LLC under Contract No. DE-AC02-07CH11359 with the U.S. Department of Energy, Office of Science, Office of High Energy Physics.