GRACC: Grid Accounting Collector

Brian Bockelman, Derek Weitzel
CHEP 2018
GRACC: Accounting Service for the OSG

- GRACC is OSG’s flexible platform for data aggregation, storage, archival, and visualization.
  - Started as a redesign of the venerable Gratia collector, which was failing to scale and implemented as a monolith Java application.
  - Redesign included many lessons learned from the prior accounting service.
  - **Original use case**: store data on every job run on the OSG.
  - Emphasis on modularity has allowed the number of use cases flourish: transfer accounting, network data, log aggregator, etc. All reuse the same core infrastructure.
In the beginning: Individual Job Records

- Key concept in this redesign is that all data goes through the message broker!
- Allows us to route data according to policy (e.g., send to both online DB and archive).
- Any individual component can be broken out and replaced.

Let’s walk through the data flow!
Step 1: Probes

- Each supported batch system (HTCondor, PBS, SLURM, LSF) has a corresponding probe. The probe knows how to query the batch system for data and convert it to a common format. Examples:

  - **HTCondor** - Parses history files of completed jobs.
  
  - **SLURM** - Queries the SLURM database for completed jobs.

- The resulting common data records are uploaded to the GRACC collector via an HTTP POST.

  - **IMPORTANT**: all data is buffered locally until successfully POST’d.
Step 2: Data Collection

- The data collector is a simple Go-based stateless web service that:
  - Listens for incoming records.
  - Parses, does simple QA, and transforms the data into an internal format.
  - Pushes the data onto a message bus.
- As a simple stateless service, ideal for container-based deploy:
  - In fact, majority of GRACC is container-based and orchestrated via docker-compose.

https://github.com/opensciencegrid/gracc-collector
Step 3: Message Broker

- The message broker queues the incoming records and routes them to the correct data sinks according to policy.

- Message broker may seem superfluous but is essential:
  - Sinks can be added to address future needs.
  - Handles durability concerns (so we don’t have to):
    - Serves as a modest-sized buffer if a destination has a transient outage.
    - For longer outages, queue will fill and the broker will stop accepting incoming records. Records are buffered at the probe level.
  - All based on RabbitMQ.
Step 4: Data Sinks

- Two primary data sinks:
  - **ElasticSearch** (via LogStash). Primary online database for interacting with records.
  - **Archival script**. Writes all records into a tarball; nightly backups to disk.
Steps 5 - ????

• Sounds simple, right?

• Unsurprisingly, accounting has a lot of additional business logic (all part of the docker-compose):
  
  • Periodic summarization of records.
  
  • Daily / weekly / monthly email reports.
  
  • Process and upload WLCG-related records to APEL.

• In addition, there is visualization: **Grafana** and **Kibana**.
Records

• ElasticSearch documents for jobs are basically the interoperable Usage Record format converted to key-value pairs.

• Not particularly human friendly.

• Periodic summarization calculates more useful quantities and correlates with grid topology data.
Some Statistics

- Average job records a day: 1 million
- Total Raw Records: 1.5 billion
- Average incoming record (all types) processing rate: 110Hz.
- Average outgoing record rate: 300Hz.
- 11TB of data in ElasticSearch.
- 16GB weekly sent to tape.
Moving beyond Jobs

- GRACC is quickly moving beyond traditional CPU usage records:
  - GPU Accounting.
  - TCP transfer statistics from HTCondor submitters.
  - Network data from PerfSonar.
  - Transfer accounting from StashCache Data Federation (StashCache).
Data Federation

- GRACC also hosts per-file transfer records for the OSG’s data caching federation, StashCache.
- XRootD sends high-level and detailed transfer information to a central collector.
  - Instead of XML records in a HTTP POST, XRootD sends binary UDP packets.
  - These packets are decoded, correlated across user login sessions, and eventually transformed into JSON records.
  - The remaining steps are unchanged.
- Effectively new data types like this can be added by only writing a new collector and adapting configurations.
Data Federation

- Aggregate in per-directory metrics

Many smaller users

Astrophysics

Bioinformatics
Data Federation

- Also have per cache aggregations
Final Thoughts

- GRACC has turned into the swiss-army-knife of data collection, storage, archival, and visualization for the OSG.

- A few strategic architectural choices allowed us to easily expand its roles beyond the original “replace Gratia” task in 2016.

- The use of off-the-shelf components has meant far less development effort …

- … and the modular design allows us to plug in new components as needed.
Backup
GPU Accounting

- Many resources on the OSG are now providing access to GPUs
- Modified probes to report if the job ‘requested’ GPUs
- Integrated this GPU attribute with the job data, summarized as well
GPU Sites
TCP Transfer Statistics

• HTCondor can send TCP transfer statistics for job file transfers

• These are the transfers for input files, certificates, output files…

• Gives information such as packet loss, reordering…
Transfer Statistics

- General Statistics

Files Transferred
197,072,655

Bytes Transferred
302.726TB

- Retransmissions per GB