GRACC: New Generation of the OSG Accounting

K Retzke[1], D Weitzel[2], S Bhat[1], T Levshina[1], B Bockelman[2], B Jayatilaka[1], C Sehgal[1], R Quick[3] and F Wuerthwein[4]

[1] Fermilab, Batavia, IL, US
[2] University of Nebraska, Lincoln, Nebraska, US
[3] Indiana University, Bloomington, IN, US
[4] University of California, San Diego, CA, US

E-mail: kretzke@fnal.gov

Abstract. Throughout the last decade the Open Science Grid (OSG) has been fielding requests from user communities, resource owners, and funding agencies to provide information about utilization of OSG resources. Requested data include traditional accounting - core-hours utilized - as well as users certificate Distinguished Name, their affiliations, and field of science. The OSG accounting service, Gratia, developed in 2006, is able to provide this information and much more. However, with the rapid expansion and transformation of the OSG resources and access to them, we are faced with several challenges in adapting and maintaining the current accounting service. The newest changes include, but are not limited to, acceptance of users from numerous university campuses, whose jobs are flocking to OSG resources, expansion into new types of resources (public and private clouds, allocation-based HPC resources, and GPU farms), migration to pilot-based systems, and migration to multicore environments. In order to have a scalable, sustainable and expandable accounting service for the next few years, we are embarking on the development of the next-generation OSG accounting service, GRACC, that will be based on open-source technology and will be compatible with the existing system. It will consist of swappable, independent components, such as Logstash, Elasticsearch, Grafana, and RabbitMQ, that communicate through a data exchange. GRACC will continue to interface EGI and XSEDE accounting services and provide information in accordance with existing agreements. We will present the current architecture and working prototype.

1. Introduction

The Open Science Grid (OSG) facilitates distributed computing for researchers on resources belonging to many organizations, with much of the computing provided opportunistically to researchers not affiliated with the facility. Accounting who used how much, of what resources, and when, is an important service provided by the OSG for researchers and facilities alike. The Gratia [1] system was developed over a decade ago to collect, store, process, and present this accounting information. Over the years, the requirements of what types of usage should be tracked, and what information should be stored, have evolved as scientific computing has evolved. Gratia has struggled to keep up with these changes, due to inflexible record models and storage. Additionally, the Gratia service architecture has struggled to scale as the OSG usage and hence record rate has increased (see Figure 1). Finally, the user interface and visualization tools have fallen behind the state-of-the-art, due to large effort being required to build interfaces with the Gratia system.

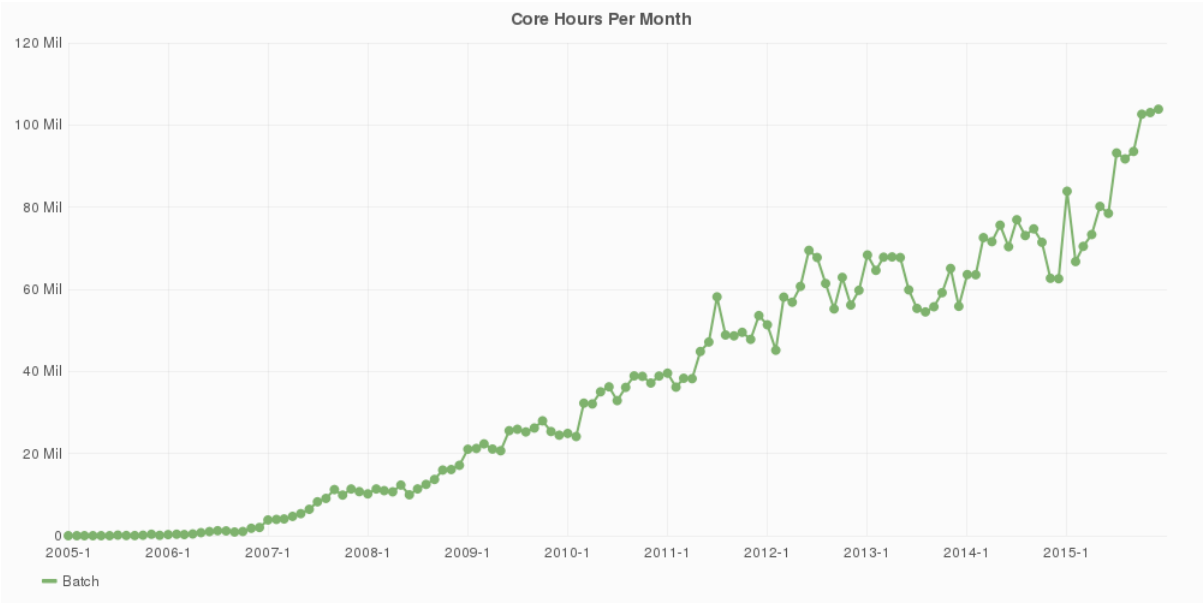

Figure 1. Open Science Grid total computing usage per month from 2005 through 2016.

For these reasons and more, in early 2016 the OSG and Fermilab decided to investigate re-designing Gratia, to provide a more flexible architecture, data storage format, and easier integration with open-source data exploration and visualization tools. The investigation settled on a microservice-based architecture, centered around *Elasticsearch*[2] for record storage and processing, and *RabbitMQ*[3] for communication between services.

2. Requirements

From the outset of the GRACC project, a number of key technical requirements were identified, based on experience with the limitations of Gratia and the existing and projected needs for the OSG accounting.

2.1. Preserve historical data

Gratia has summary usage data for batch computing jobs going back to 2005 (Figure 1), and it is necessary to be able to query and visualize the complete history. Additionally, it is necessary to maintain long-term archival of the detailed accounting records. The current archive is comprised of nearly two billion XML records compressed on redundant magnetic tape at Fermilab's Enstore facility [4]. Records are saved to tape daily by the Gratia collector. GRACC will need to have an equivalent service to ensure detailed history can be made available for backup and security audit purposes.

2.2. Maintain compatibility with Gratia

Gratia collects accounting data by means of distributed "probes" that run alongside the batch system (e.g. *HTCondor*, *PBS*, *SLURM*, etc.) at each facility and send to the central Gratia collector records detailing each job, or group of jobs, run by users from or on the OSG. Due to the large installed base, and multiple batch systems (and other related services) from which records are collected, it is necessary that GRACC have a service that can serve as a Gratia-compatible collector for these records, thus requiring no immediate changes to the probes. Additionally, Gratia collectors have a replication mechanism whereby records can be forwarded to another

Gratia collector; this will be the primary mechanism to transfer existing records from Gratia to GRACC during the transition.

2.3. Provide for fast multi-dimensional analytics
The main OSG Gratia collector receives on the order of one million job usage records a day (see Figure 2). GRACC needs to be able to provide responsive analytics and visualizations of these records over periods ranging from days to years, across many key dimensions, such as virtual organization (VO), facility, field of science, project, usage model ("dedicated" vs. "opportunistic"), and user. Gratia maintains daily summaries that collate usage metrics across these dimensions, which reduces the number of records by two to three orders of magnitude, as shown in Figure 2. This allows for responsive queries over extended periods of time, although it is typical for multi-year queries to require several minutes to complete.

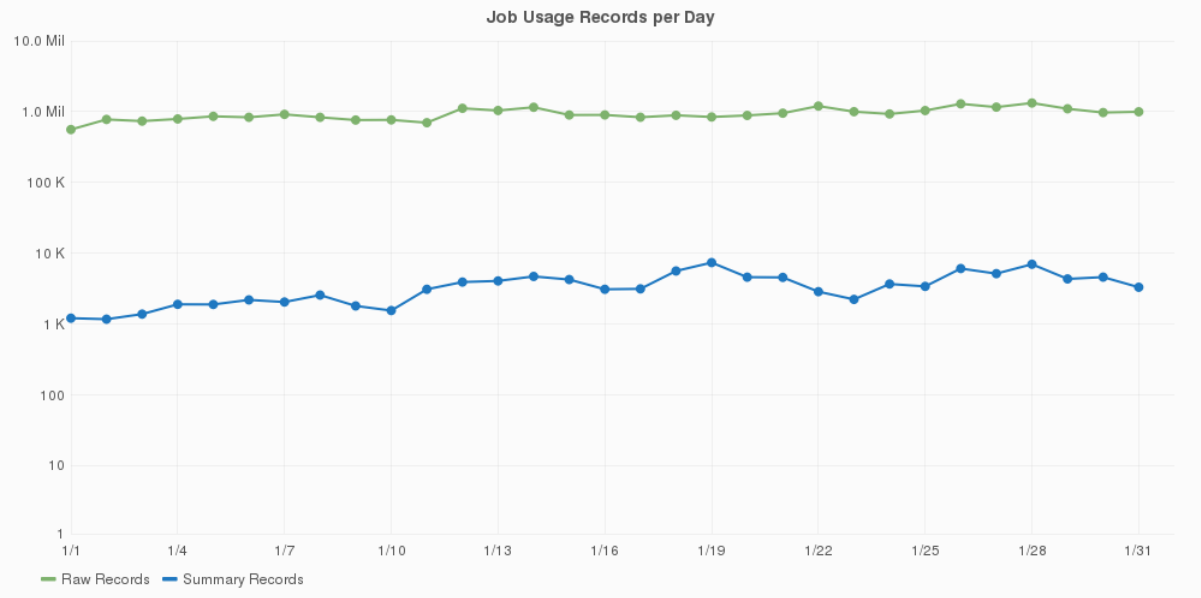

Figure 2. Count of Open Science Grid job usage records received each day in January 2016 and corresponding summary records

GRACC needs to support the same analytics use cases currently supported by Gratia, and given advances in data analytics technologies over the past several years should be able to provide even multi-year extended queries on the order of seconds rather than minutes. Furthermore, while queries against detailed job records in Gratia are very expensive, to the point of being impossible for all but the shortest time periods (i.e. several days), GRACC should be able to support interactive queries over time periods up to several months or even years.

In addition to Gratia providing the accounting service, a number of automatically-generated reports are sent to various stakeholders based on Gratia data. These include job success rate, user job efficiency, and opportunistic usage of OSG sites per VO, among others. GRACC needs to support a similar suite of reports that can quickly and reliably collect and aggregate data.

2.4. Support extensible data types and fields
The Gratia usage record interchange format is based on the Open Grid Forum's *Usage Record* format [5]. While this format is extensible, the storage in Gratia is limited to fixed set of fields, particularly in the daily summaries, which limits what dimensions are available for analytics,

as previously discussed. GRACC should support a more fluid record schema, allowing the individual needs of the users and facilities to dictate what fields are included and available for analytics.

In addition to the data included in the job usage records by the probes, it is necessary to enrich the records with data taken from external sources, notably the OSG Information Management System (OIM)[6]. This system provides additional dimensions for each VO, facility, and project. Gratia does this as part of the visualization interface, GratiaWeb. As GRACC is intended to provide a consistent view of the data across multiple visualization and analytics interfaces, it will be necessary to incorporate this information into the GRACC database.

2.5. Extensible architecture

A shortcoming identified in Gratia was the use of a monolithic architecture coupled with a rigid database schema, that made data migrations and code changes non-trivial productions. Based on this experience, it is desired that GRACC utilize a loosely-coupled architecture composed of smaller components connected through a message exchange with a flexible data schema and interchange format. This will allow individual components to evolve independently and at their own pace.

3. Architecture

The requirements laid out by the OSG technical committee and described in the previous section steered the GRACC investigation towards a small set of existing technologies. Furthermore, based on existing technical experience within the OSG and at Fermilab [7] and elsewhere, the choice was clear to utilize several key components for a proof-of-concept (POC): Elasticsearch as a data storage and query platform, RabbitMQ for data exchange, and Grafana for the primary user interface, supplemented by Kibana for ad-hoc analytics. This successful POC led to utilizing these technologies for full implementation.

An overview diagram of the components of this system is provided in Figure 3.

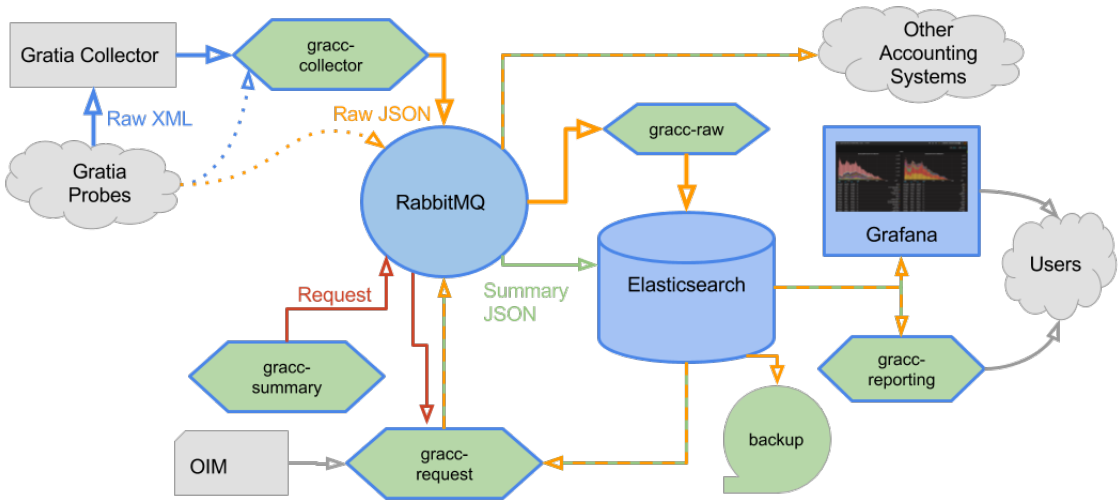

Figure 3. Diagram of the GRACC microservice architecture. Included are the existing Gratia probes and collectors, GRACC services, inter-process message passing, record storage, and visualization

3.1. Record Collection

Origin usage record generation for GRACC is handled by the existing Gratia probes. It is envisioned that in the future the probes will be updated to send records directly to the GRACC message exchange. For the near-term, the *gracc-collector* service [8] was developed as a transitory endpoint that is compatible with existing Gratia collectors and probes. The collector transforms the records from XML into a more flexible "flattened" JSON representation and publishes them to a standard RabbitMQ exchange for further processing and storage. No state is maintained with the collector, instead relying on RabbitMQ for buffering downstream, and/or backpressure to the probes upstream to handle downtimes and service interruptions.

3.2. Communication

All inter-process communication is handled through a RabbitMQ broker, which was chosen over the several major competing message systems that would meet the needs of GRACC primarily because the OSG had an existing deployment at its Grid Operations Center (GOC). Each stream[1] of usage records has several well-known exchanges for receiving records or communications: *raw* records, *summary* records, and *replay* requests. GRACC services create queues that are bound to these exchanges to process, store, or handle records and messages.

3.3. Record Processing & Summarization

Raw usage records that are sent to the *raw* RabbitMQ exchange have to be stored in the Elasticsearch database, after undergoing some processing (i.e. the addition or manipulation of fields) to facilitate storage and retrieval. Processing includes: 1) verifying records are complete and meet certain acceptance criteria, 2) pre-computing derived quantities that are commonly used in analytics, and 3) computing a unique checksum that will be used as the primary storage key and can indicate if two records are duplicates. *Summary* records undergo similar processing. *Logstash*[9] is used for both, as its pipeline architecture and large plugin library allows simple configuration of record processing as well as robust communication with both RabbitMQ and Elasticsearch in addition to any other components that may be included in GRACC in the future.

Record summarization is performed by a process that listens for *requests* (through RabbitMQ), that specify the time period to summarize and the RabbitMQ exchange to send the summarized records to. The summarization is done by aggregation query to Elasticsearch, formatting the results into summary records, performing name corrections/mappings as defined in a lookup table (stored in Elasticsearch), and further enriching the records by looking up corresponding data in OIM [6], such as field of science, cluster ownership, and site topology.

3.4. Record Storage

Elasticsearch is used for record storage, as it provides a fault-tolerant distributed architecture, flexible schema, partitionable indices, and powerful search and aggregations. Furthermore, there is a growing base of experience within the OSG and at Fermilab with using Elasticsearch for system and service monitoring and analytics. Each *stream* of *raw* records is stored in a time-based *index pattern*, where each index stores the records for one month. Likewise *summary* records are stored in yearly indices. The use of time-based indices allows for faster queries, as most will have a time-based component (e.g. "what were the total core hours used each day for the last month?"), and therefore the queries only have to be run against the indices that span the time range of interest. In addition, time-based indices allow for simple archival/removal of historical *raw* records.

[1] a *stream* is a co-hosted but otherwise independent series of usage records, e.g. grid jobs, file transfers, and virtual machine instances. These correspond to existing co-hosted Gratia collectors and databases.

3.5. Interactive Visualization

The primary visualization interface is Grafana[10], an open-source monitoring dashboard application, which supports a rich ecosystem [11] of data sources and graph plugins, and provides a powerful interface to create and share dashboards (e.g. Figure 4). Kibana[12] is also made available for in-depth analytics and data exploration.

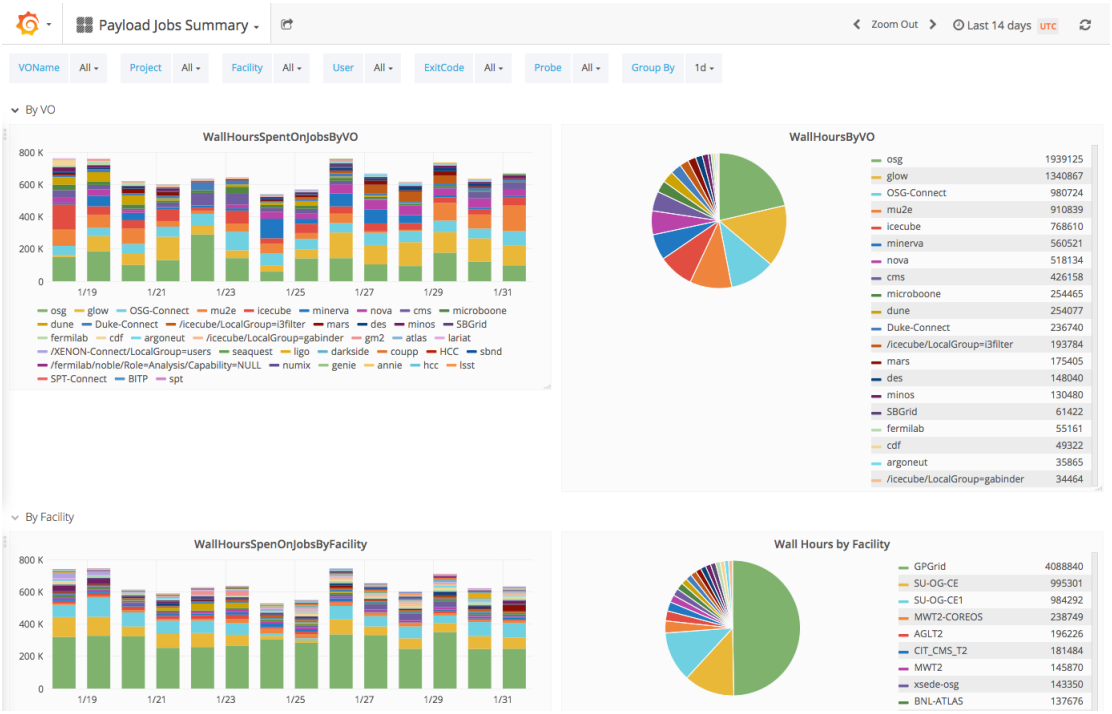

Figure 4. Example Grafana dashboard showing GRACC job accounting data

3.6. Reporting

As stated previously, the structure of GRACC differs greatly from Gratia, and thus the interfaces between the reports and the data layer along with the actual database queries had to be almost entirely rewritten. Furthermore, much of the new GRACC reporting infrastructure has been generalized, rather than residing in report-specific code. This will facilitate future development as well as reduce complexity and support volume. We have already seen substantial performance improvements: whereas in the past, the generation of reports had to be staggered over the period of 15-30 minutes to avoid overloading the database, we can now generate and send most daily reports in under a minute, and can send a standard daily set of reports within 2-3 minutes, even if the data has not been previously cached by Elasticsearch.

3.7. Monitoring

System and service monitoring is included in GRACC as a first-class citizen, as it is critical to understand the performance and limitations of the system, and to catch issues quickly. Prometheus[13] was chosen as the monitoring platform, as it is simple to operate, has a large selection of *exporters* and client libraries to collect telemetry, and integrates well with Grafana. The GRACC collector directly exports metrics through the Go client library, while *exporter* applications are used to collect performance metrics from RabbitMQ, Elasticsearch, Grafana, and the host nodes. An example monitoring dashboard is shown in Figure 5.

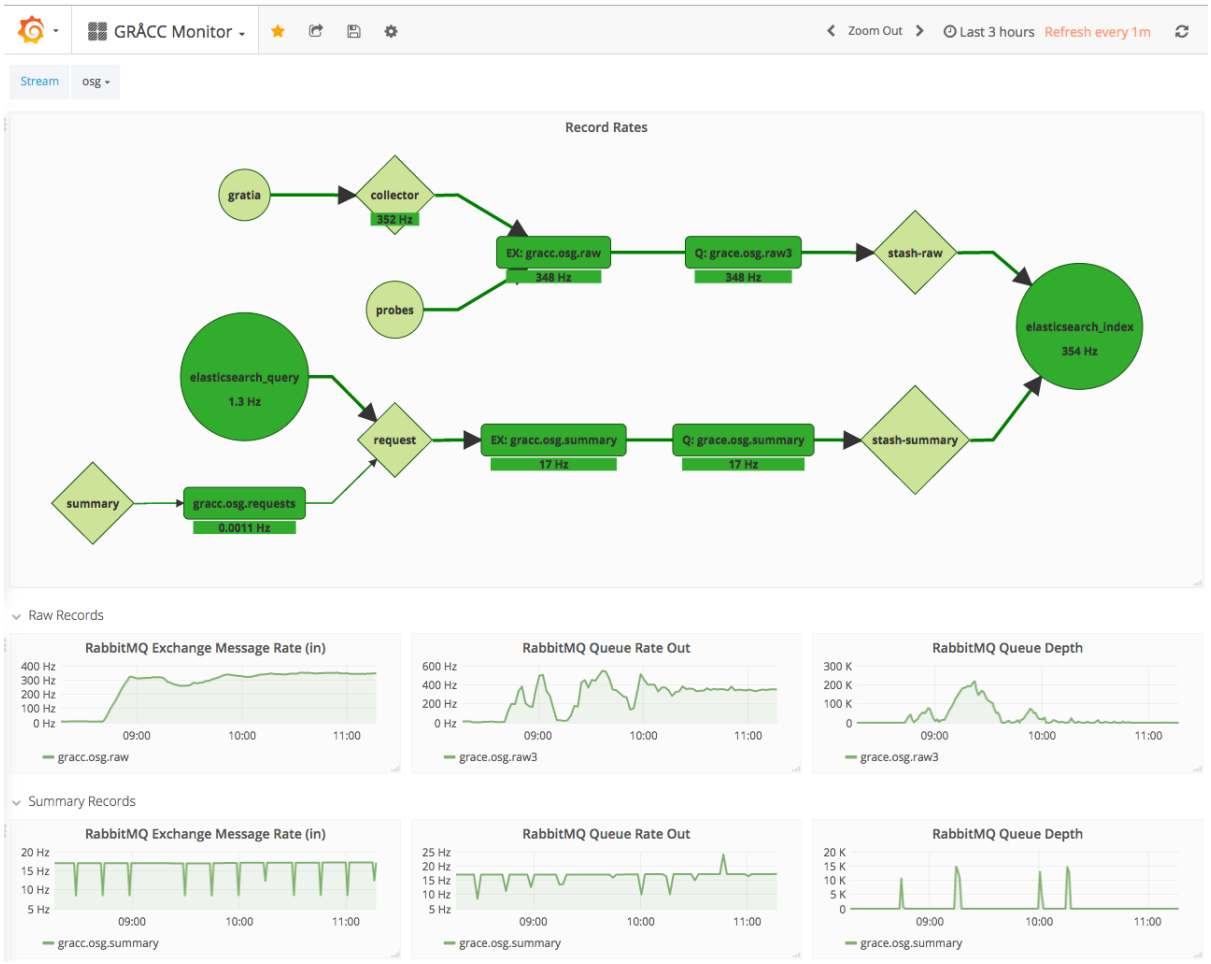

Figure 5. Example Grafana dashboard showing GRACC record flow paths, processing rates, and service metrics.

4. Conclusion

GRACC is a work in progress, in development as of this conference, with planned initial production deployment in early 2017. The GRACC development database was back-filled with raw records to early 2015, and is receiving new records continuously from the primary Gratia collector. The Elasticsearch cluster has over one billion records, utilizing 3.5 TiB of disk space, growing at approximately one million new records per day.

Throughout the development process we have observed significant gains in performance and user experience compared to Gratia. The interactivity of Grafana and Kibana provides for rich exploration of the accounting data that isn't possible with Gratia, and many users have started using GRACC even though it has not been declared production.

As GRACC approaches production deployment in Spring 2017, the focus is on reaching feature-parity with Gratia, while paving the way for expanding the capabilities beyond what was possible within the limits of Gratia. Example features include:

- Probes reporting directly to RabbitMQ and including site- and user-specific fields in the records.
- Adding new fields to records through entries in OIM.
- Support custom user analytics through direct queries to Elasticsearch and tools like Jupyter

Notebook and Zeppelin.

- Integrate display of accounting data with other sources, such as network statistics from perfSonar.

Acknowledgments

Fermilab is operated by Fermi Research Alliance, LLC under Contract number DE-AC02-07CH11359 with the United States Department of Energy.

The Open Science Grid is funded by the United States National Science Foundation and the Department of Energy.

References

[1] T Levshina *et al* 2015 *J. Phys. Conf. Ser.* **513** 032056.
[2] https://www.elastic.co/products/elasticsearch
[3] http://www.rabbitmq.com/
[4] G Oleynik *et al.* 2005 *22nd IEEE / 13th NASA Goddard Conf. on Mass Storage Systems and Technologies* April 2005, Monterey, CA, USA pp 73-80.
[5] R Mach *et al.* Usage Record - Format Recommendations, Open Grid Forum GFD-R-P.098.
[6] http://oim.grid.iu.edu/oim/home
[7] K Herner *et al.* 2017 *J. Phys. Conf. Ser.* Forthcoming.
[8] https://github.com/opensciencegrid/gracc-collector
[9] https://www.elastic.co/products/logstash
[10] http://grafana.org
[11] https://grafana.net
[12] https://www.elastic.co/products/kibana
[13] https://prometheus.io