Implementing the Belle II Conditions Database Using Industry-Standard Tools

Lynn Wood, Todd Elsethagen, Kevin Fox, Jeter Hall, Bibi Raju, Malachi Schram, Eric Stephan, Jovan Araiza

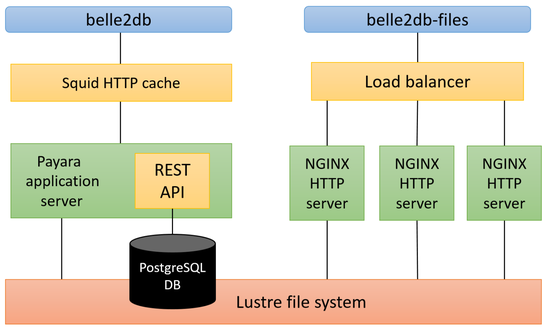
Pacific Northwest NATIONAL LABORATORY

Proudly Operated by Battelle Since 1965

The **Belle II Conditions Database** is operational and serving the current Belle II Grid-based Monte Carlo campaigns. The extensive use **of industry-standard applications and tools** has allowed it to be largely supported by IT staff, as opposed to dedicated scientists, reducing cost and effort. Performance has proven more than sufficient for current loads, and planned updates are underway to bring it up to full data production requirements.

Background

The Belle II Experiment at KEK is preparing for first collisions in early 2018. Processing the large amounts of data that will be produced will require conditions data to be readily available to systems worldwide in a fast and efficient manner that is straightforward to both the user and maintainer.



Current database back-end implementation

Description

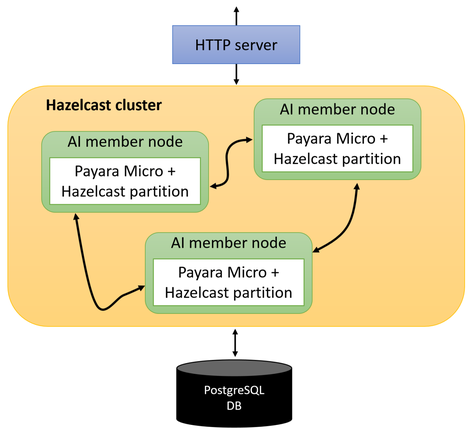
The Belle II Conditions Database was designed to make maintenance as easy as possible. To this end, a HTTP-based service interface was developed with industry-standard tools such as **Swagger** for the application interface development, Payara for the Java EE application server, and the Hazelcast in-memory data grid for support of scalable caching as well as transparent distribution of the service across multiple sites.

The left and right sides of the above figure represent two servers, one of which supports the REST API and database, while the other provides access to the payload data files. Each component in the figure above is implemented as a separate **Docker** "container", which reduces the resource and requirements and makes it significantly easier to start up new instances in different locations. This use of Docker containers also provides independent auto-restart functionality for each component.

Communication with the database uses standard HTTP using XML and JSON data. The choice of a standardized REST API makes the client coding independent of the actual database implementation details, and allows for easy caching support using a **Squid** cache proxy server to reduce load on the API and database.

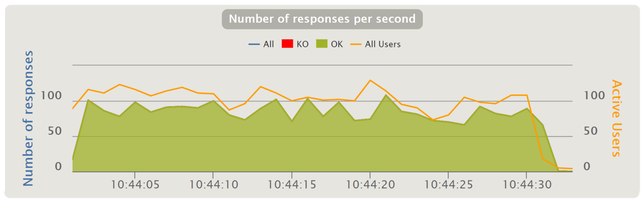
The database itself is **PostgreSQL**, but the schema and procedures have been kept generic so other database applications could be used if necessary. To keep the database small, the payloads only consist of references to files on a separate server. These "payload files" are currently assumed to be ROOT objects by the client, although there is nothing in the database design that limits the data type.

The REST API interface is built from two commercially-available applications. **Payara Micro** is a Java EE application server optimized for production operations in a containerized infrastructure. **Hazelcast** is a Java-based in-memory data grid. Data is distributed evenly among separate nodes of a computer cluster, which provides horizontal scaling in both available storage space and processing power. This provides multiple benefits, including caching of frequently-used data in memory, transparent scalability, and load-balancing to reduce the query load on the database.

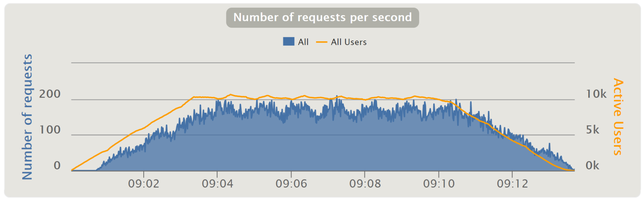


Application interface implementation using Payara and Hazelcast

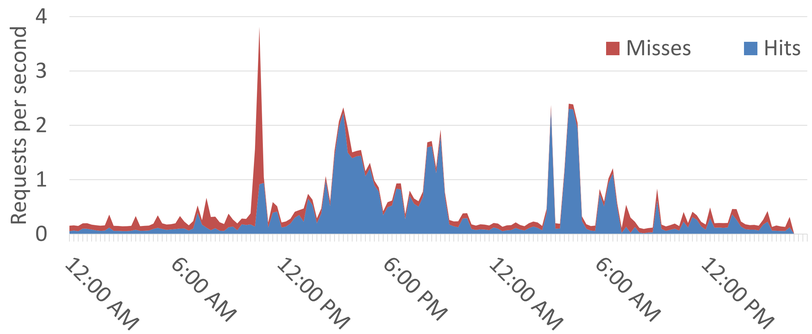
The file server currently consists of three **NGINX** HTTP servers handled by a load balancer to distribute traffic evenly across the servers. The back-end file system is based on **Lustre**, an open-source parallel file system for high-performance computing environments. The Lustre file system is shared between all back-end instances.



Gatling stress testing of the database server, showing a sustained rate of ~80 requests per second



Gatling stress testing of the payload file server, showing a sustained 180 requests per second and support of 10,000 simultaneous open connections



Squid HTTP cache hit/miss performance during 18 hours of the latest Monte Carlo campaign.

Performance

The back-end has been evaluated with both direct access by Grid-based Belle II Monte Carlo campaigns and directed testing by the database group. The directed testing was done using **Gatling**, an open-source load stress testing tool for HTTP servers which allows custom test design through scripting.

Testing of the database back-end was implemented by monitoring the usage patterns during Monte Carlo operations and then writing separate tests for the REST API and file server. The use of scripting allowed for much higher stress loads than was readily available from Grid-based testing. The directed stress testing has been successful; typical results are shown at left. The top graph shows the database server responding successfully to 80 requests per second, while the bottom graph shows the payload file server responding to180 file requests per second, with up to 10,000 open connections being handled simultaneously by the load-balanced HTTP servers. These tests correspond to nearly half of the payload bandwidth for the expected full-scale Belle II grid-based analysis cloud of 100,000 active nodes.

Future Improvements

Several improvements to the Belle II Conditions Database are in progress:

- Hazelcast supports cache clustering between remote sites, and will be evaluated as a means of supporting more localized database access worldwide. Placing portions of the cache at three or more sites will provide faster response times as well as backup capability for site or network outages.
- The PostgreSQL database is still a single-site instance and not currently scalable. Several options of supporting a distributed database are being investigated, including OpenStack **Trove** and **CockroachDB**. The replicated databases would be sited in tandem with the Hazelcast cache cluster sites.
- Authentication is currently not implemented, but is planned for services that would modify the database. The possibility of leveraging the X.509 authentication already present in the Belle II Grid computing interface is being investigated.

ABOUT Pacific Northwest National Laboratory

The Pacific Northwest National Laboratory, located in southeastern Washington State, is a U.S. Department of Energy Office of Science laboratory that solves complex problems in energy, national security, and the environment, and advances scientific frontiers in the chemical, biological, materials, environmental, and computational sciences. The Laboratory employs nearly 5,000 staff members, has an annual budget in excess of $1 billion, and has been managed by Ohio-based Battelle since 1965.

For more information on the science you see here, please contact:

Lynn Wood
Pacific Northwest National Laboratory
P.O. Box 999, MS-IN: J4-60
Richland, WA 99352
(509) 372-4583
lynn.wood@pnnl.gov

Acknowledgement *This work was carried out for the U.S. Department of Energy under Contract DE AC05 76RL01830 PNNL-SA-121968.*