Scalability Tests for the Conditions Database Stack of the Belle II Experiment

EDWARD MWANGI
Co-authors: David Joffe, Todd Elsethagen, Jeter Hall, Bryan Fulsom, Malachi Schram

Kennesaw State University

Kennesaw, GA

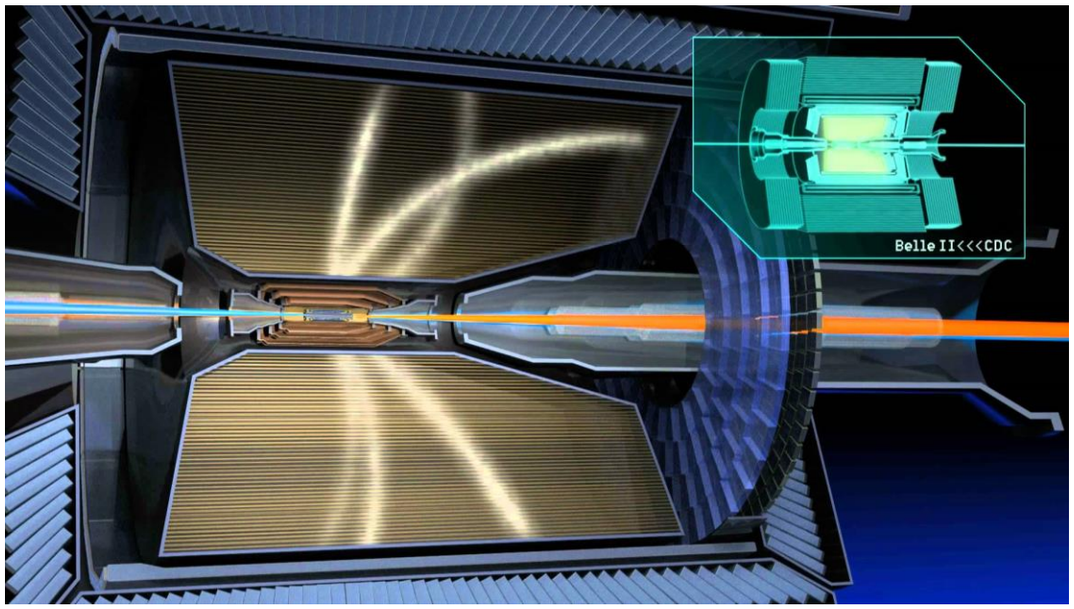
November 19, 2015

The Belle Experiment

- The Belle experiment is a high energy and high intensity particle physics experiment that investigates scientific phenomena.
- At the Belle detector, scientific searches made are CP-violations, rare decays and searches for exotic particles and precision measurements.
- Belle II is will be the second phase of the experiment at an even larger scale.
- This presentation is about the tests performed on the payload system of the Belle experiment.

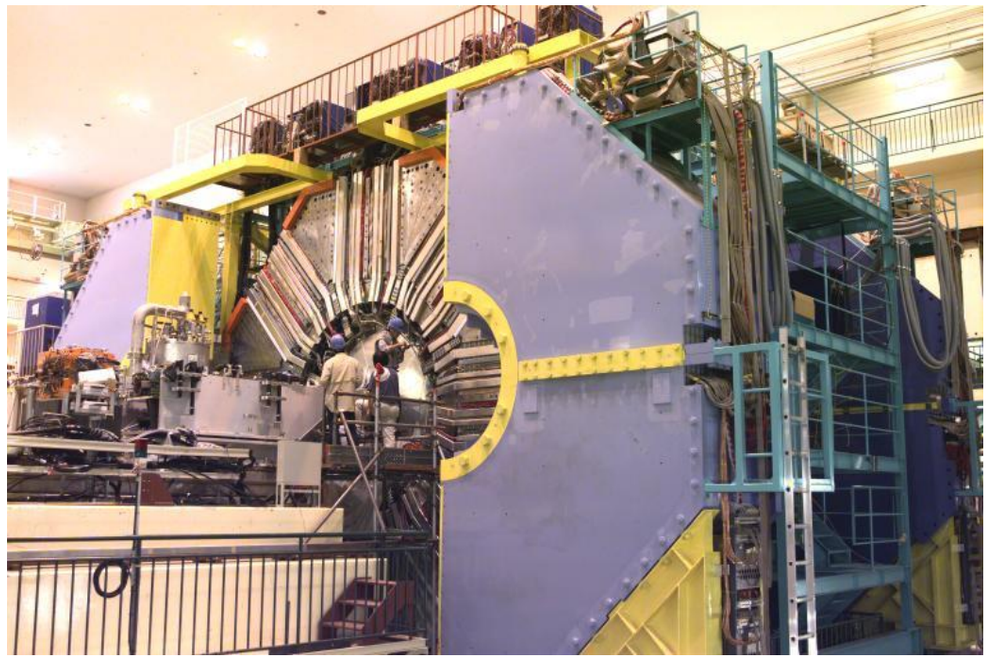
The Belle Experiment Overview

- The goal of the Belle experiment is to discover new physics phenomena by analyzing large amounts of data and searching for meaningful deviations.
- The experiment takes place in Tsukuba, Japan where the KEKB accelerator collides positron and electrons to yield events that can be quantified through the Belle detector.
- Raw data recorded from the detector is transferred to hard disks and magnetic tapes for further processing and analysis.



Scale of the Belle Experiment

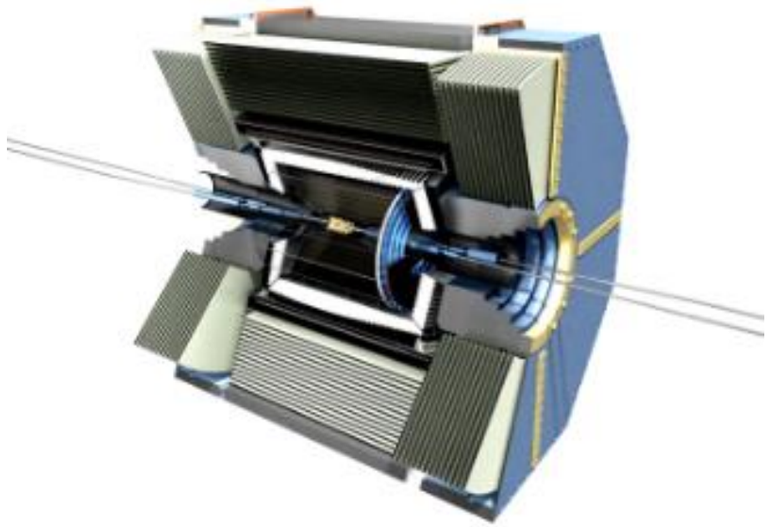
- The Belle experiment confirmed the Kobayashi-Maskawa theory of CP violation which led to the Nobel Prize for Professor Kobayashi and Professor Maskawa in 2008.
- The 3km KEKB accelerator and Belle detector yield and record about 200 events per second to a volume of about 1TB of data per day.
- Petabytes of data from the Belle detector have to be collected and analyzed to find only a few CP violations.
- Current and previous data is equally useful and hence needs to be stored and transferred for analysis among institutions around the world.
- Belle II “Super KEKB” accelerator will be 100x more powerful, the volume of data 50x more, and millions of channels to collect the data.



The Belle II Detector

Belle II Database Needs

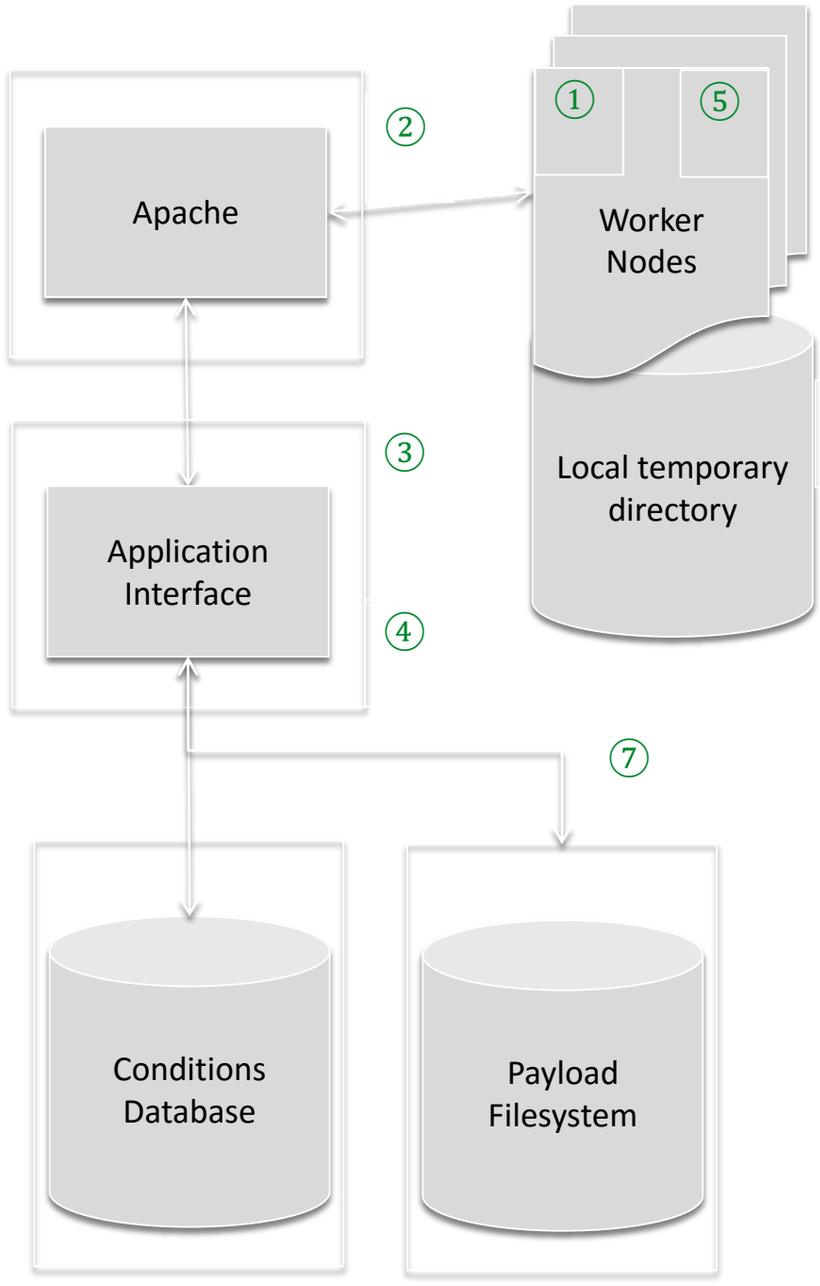
- Data from the Belle II detectors is distributed in raw and processed formats geographically across the world.
- Belle II relies on four critical databases:
  - Configuration DB – used to configure the subdetectors
  - Logger DB – used to record subdetector status
  - Conditions DB – used to store calibrations data for data processing
  - File metadata DB – used to identify and locate files
- The Conditions database is the focus of this study.
- The database needs to be scalable and robust for expected number of hits when Belle II kicks off.



The Belle II detector

Conditions DB

- The conditions database houses information about the conditions of the Belle II detectors. This information is needed during processing of data from the detectors.
- The payloads in the Conditions DB exist only as references. Actual payload files exist outside the database. This makes the Conditions DB smaller, faster and easier to back up.
- Conditions DB Terminology:
  - Payloads: A file that contains Conditions data
  - Interval of Validity (IOV): Period of time (experiment/runs) for which the payload is valid
  - Global Tag: A unique identifier used to define a collection of payloads with defined IOVs



Summer Project

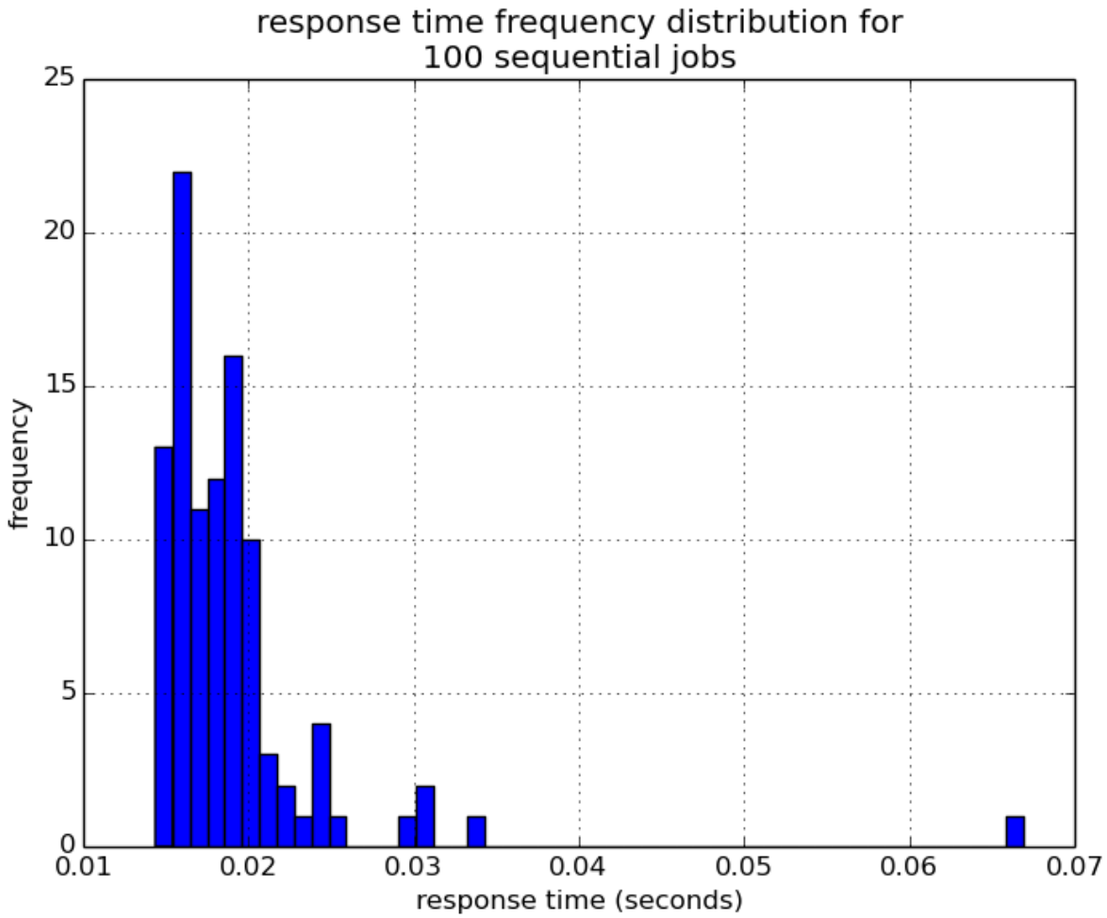
- Goals
  - Understand the Conditions Database software infrastructure.
  - Performance evaluations.
  - Develop test cases for Conditions DB load testing.
  - Acquire Conditions DB performance metrics.

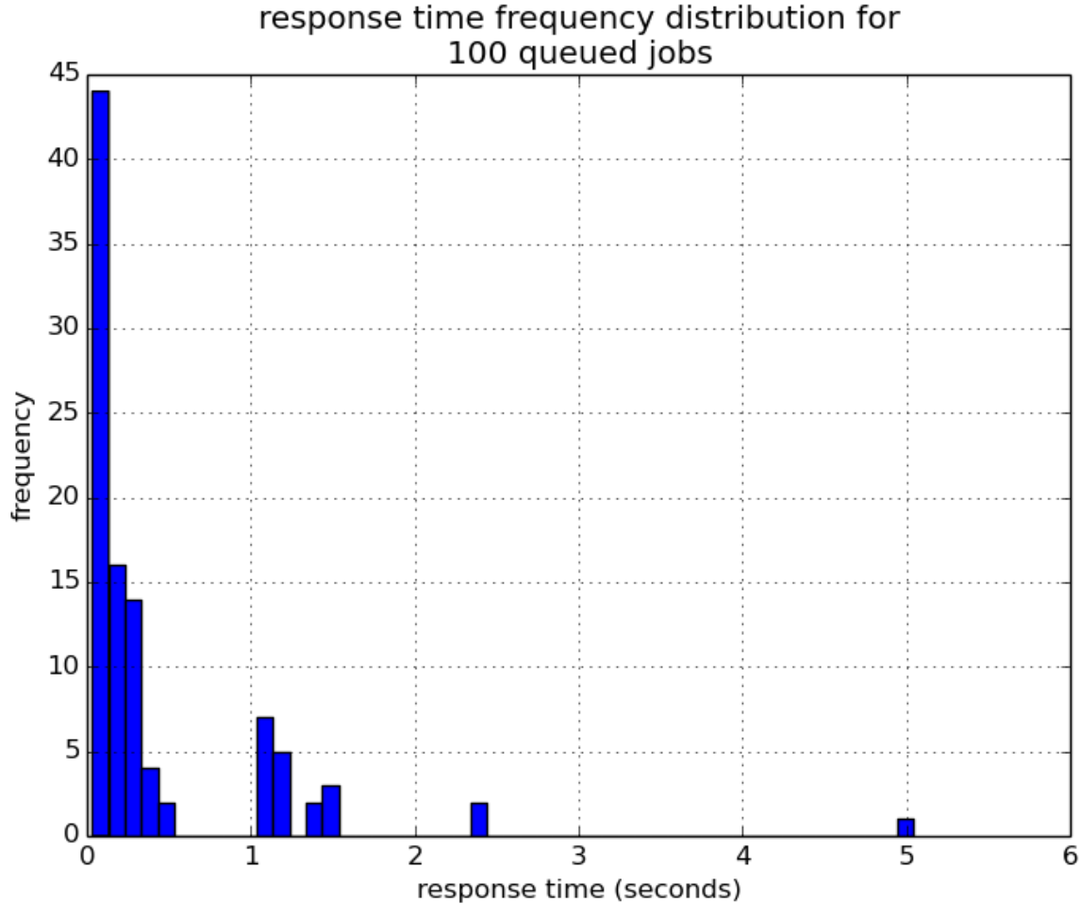
- Conditions Database Stack
  - PostgreSQL – this is the Database Management System (DBMS) for the Conditions Database.
  - Apache Server – The Conditions DB server runs an Apache server application.
  - DB Server Application Interface – A Java Application Container framework is used to expose the Conditions data via a REST API.

Load Testing Workflow

- A workflow was developed to run a set of defined test cases from the client to the server.
- The steps to the workflow are:
  - Connect to the DB and request a list of payloads.
  - Receive the list of payloads in JSON(JavaScript Object Notation) format.
  - Download the payloads from a payload host server.
- The workflow was submitted as a job to worker nodes (Via HTCondor or directly).
- Several jobs were run in parallel to load the Conditions database server.
- Metrics collected after running the test were:
  - Response time (Client level).
  - Query duration (DB level).
  - Number of active queries (DB level).

Client side Results: Retrieval Time for Payload List

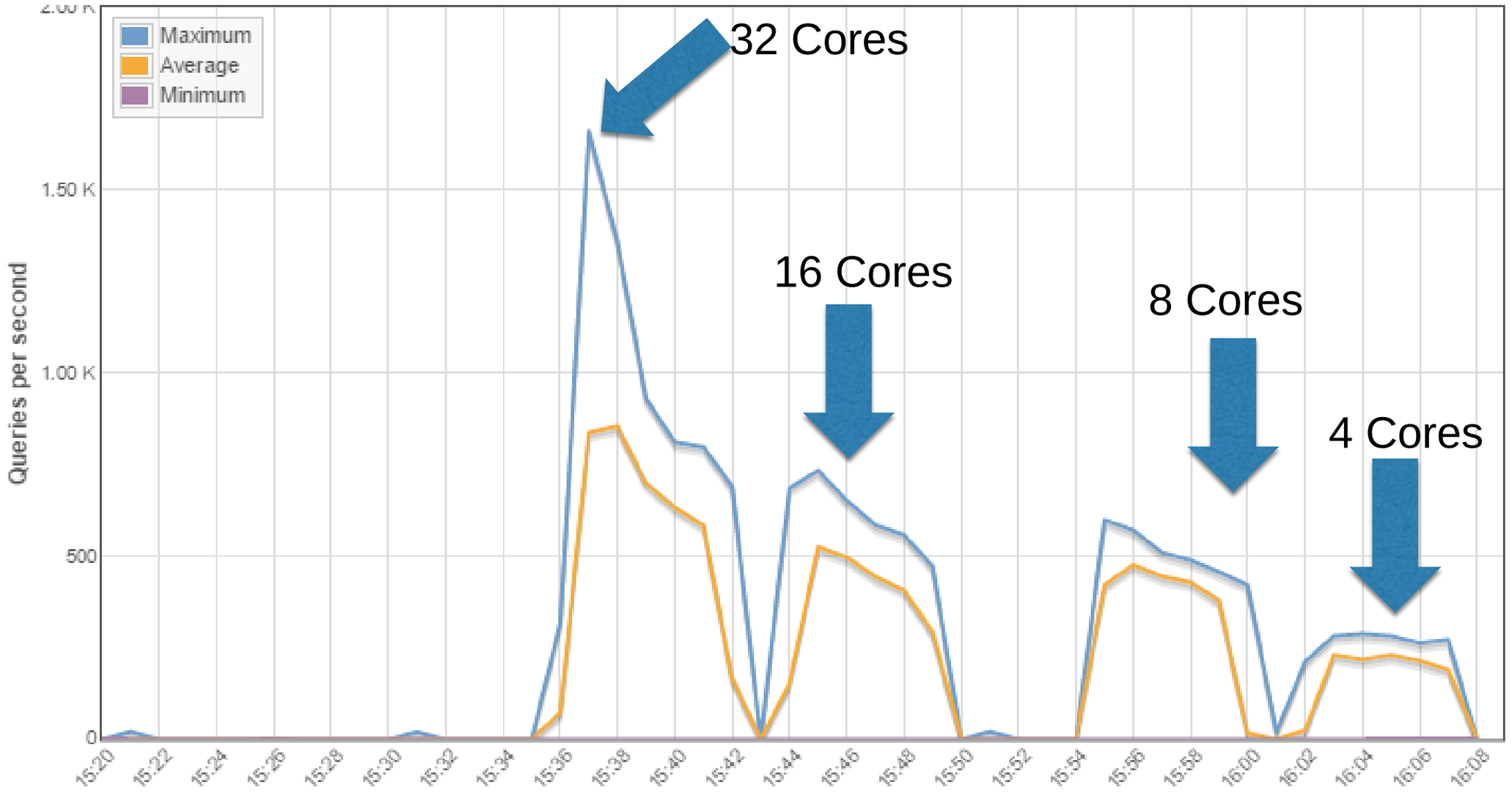




- Higher response time values were observed when jobs were queued on HTCondor (parallel) than when run sequentially.

Connections and Queries Plots

- The software tool pg_badger was deployed to monitor the PostgreSQL database.
- Metric collected:
  - Queries per second



Achievements and Moving Forward

- Goals Met
  - Learned the existing Conditions database infrastructure.
  - Developed software to use the existing REST API.
  - Created a workflow to study the DB performance
    - For single node jobs.
    - For HTCondor.
  - Deployed monitoring tools for the database server.
  - Developed tools to collect and visualize client side metrics

- Moving Forward
  - Acquire additional server profile details (CPU, RAM and Disk Usage).
  - Full scale local cluster tests.
  - Grid scale tests.
  - File System tests.