Conditions DB Scalable Access

Lee Lueking/Igor Mandrichenko

October 13, 2010

Overview

- Conditions Data is an umbrella term referring to information describing detector and beam conditions.
- It includes calibration, alignment, attenuation, pedestal, etc. for detector channels, as well as information about the intensity and characteristics of the beam.
- It is generally valid for detector data taken during specific periods of time, referred to as Intervals Of Validity (IOV)
- Some of this information is required for processing and analysis of detector data and thus access is required by many clients running simultaneously on interactive and GRID resources.
- Much of this data is stored in central databases or files, and approaches to scale the delivery to thousands of clients are needed.

Requirements

- Following are parameters that define the problem. Typical values need to be obtained from experiments and/or estimated.
- Expected request rate
  - Peak
  - Average
- Data unit size
- Latency requirements
- Accepted failure rate
- Some estimate of time correlation between requests
- Boundary conditions
  - hardware to be used
  - network bandwidth available
  - technologies to use and not to use

Example Solutions and Constraints

- Two classes of solution are possible,
  - 1) scaling up the central database service, and
  - 2) providing additional caching tiers between the database server and the client.
- Simply scaling up the central service is limited by server hardware and network constraints. In some cases scrutinizing the way each experiment stores and retrieves information can also provide performance improvements.
- By including additional caching layers, or tiers, lightweight, high performance components can be deployed. In many cases this can be done close to where the clients are running and significantly improve throughput while maintaining low central server and network loads.
- Examples: 1) SQLite files, 2) HTTP w/ proxy/caching servers.

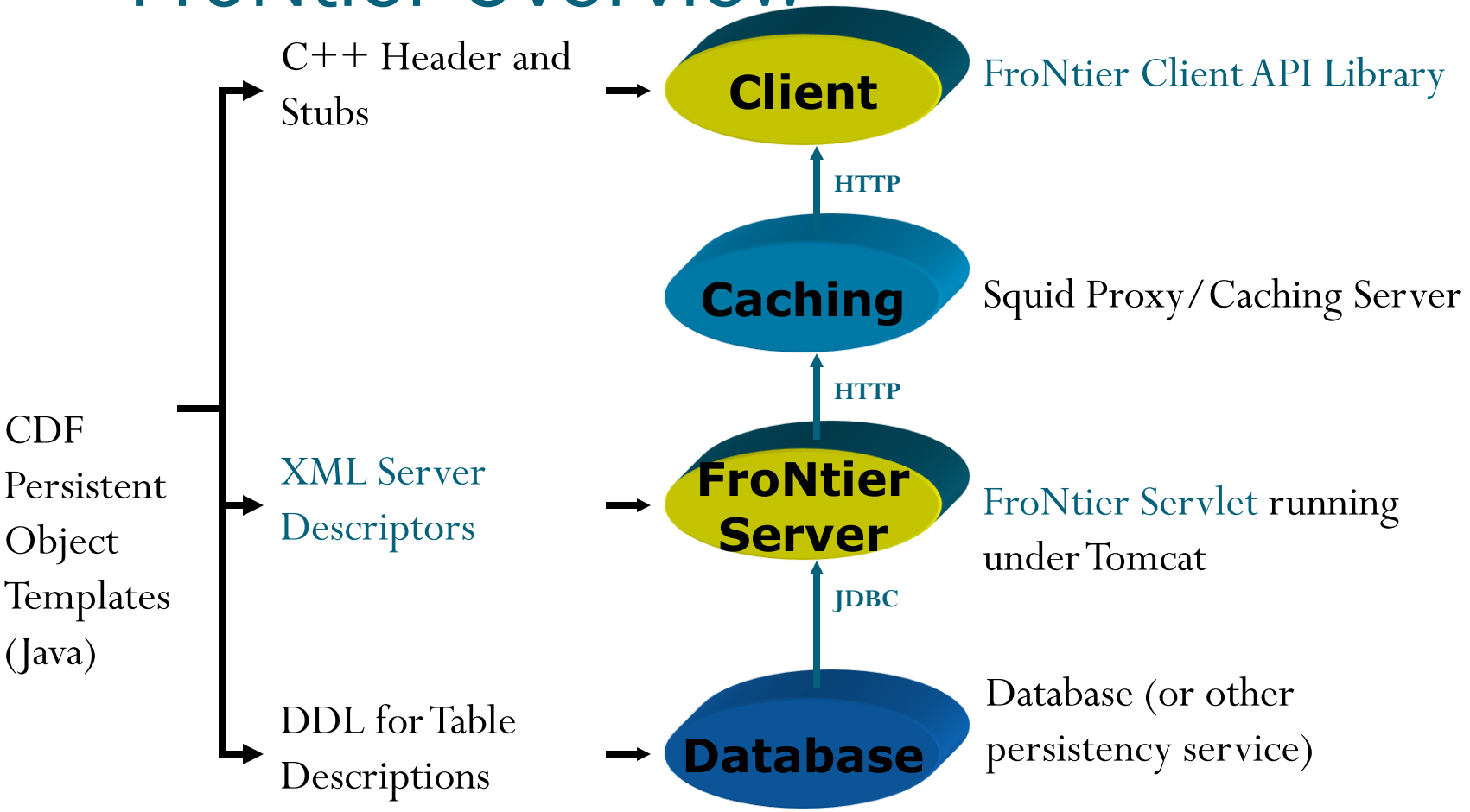
Constraints

- IOV table: the requests from the client must refer to an IOV "key" which is loaded into the client initially, before any requests for conditions data.
- API: Each experiments database access needs to be adapted to a single "standard" API that can be shared by everyone.
- Cache coherency policy: In a cached system, the cache can be stale and the refresh policy must be understood.
- Deployment: A central service that has connection pooling to the central database is required.

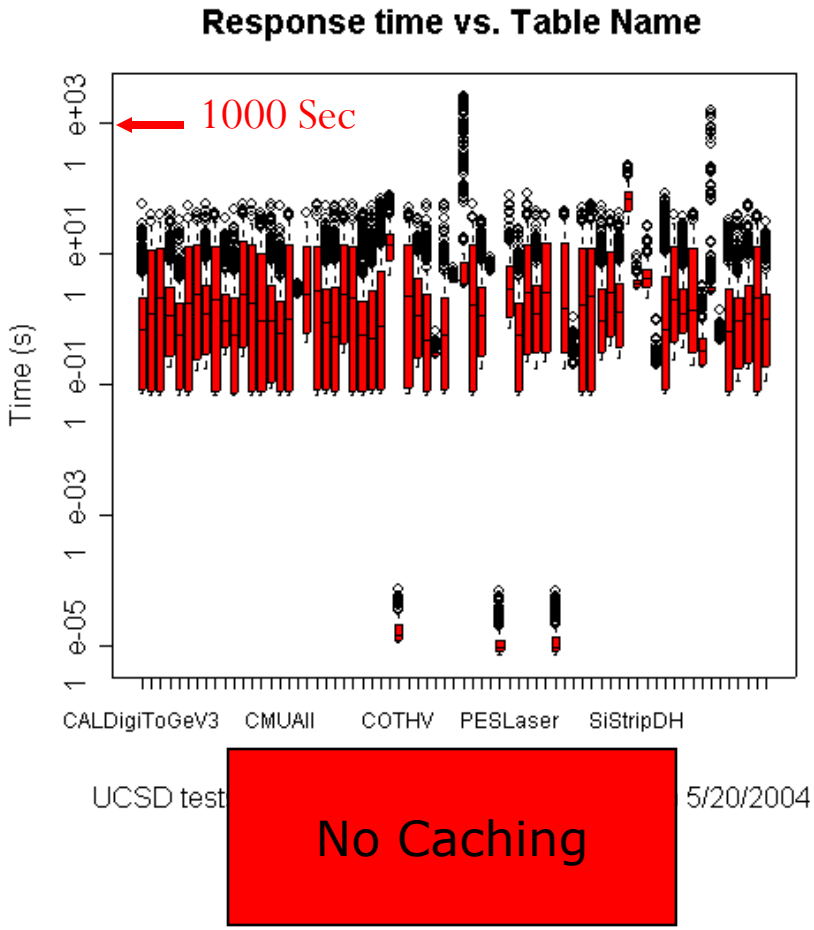
Examples of FroNTier Use

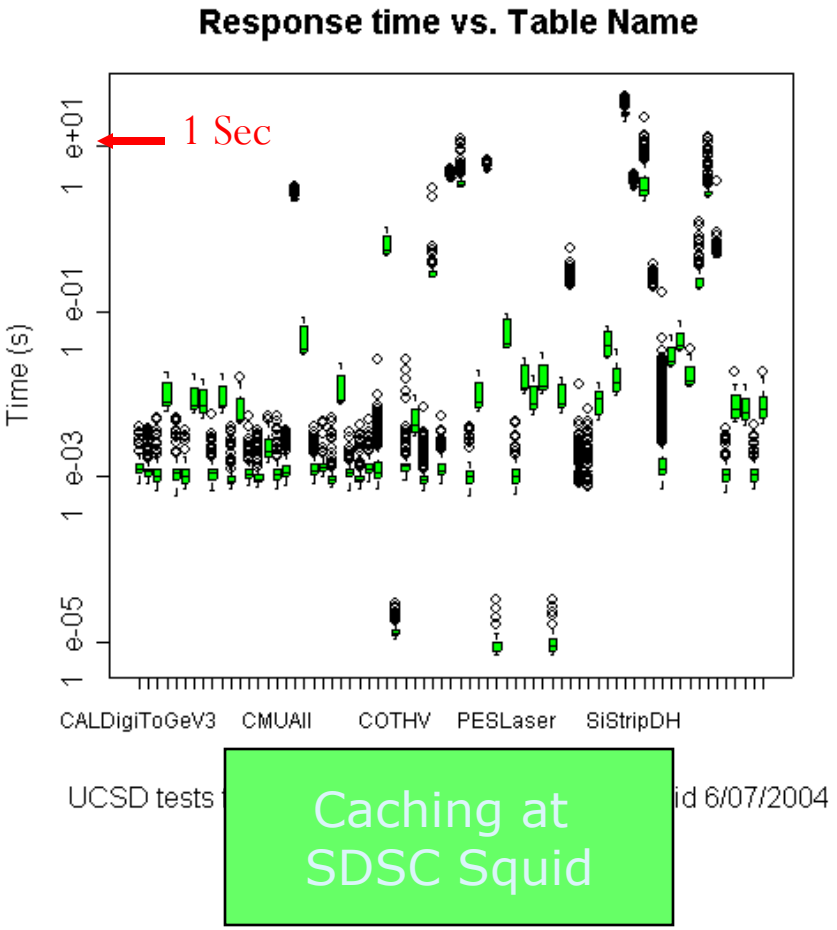
- CDF: http://www-cdf.fnal.gov/upgrades/computing/calibration_db/calib_primer.html
- ATLAS: http://atlas-computing.web.cern.ch/atlas-computing/links/nightlyDocDirectory/IOVDbSvc/html/index.html
- CMS: http://frontier.cern.ch/

FroNtier Overview



Client Side: FNAL/SDSC



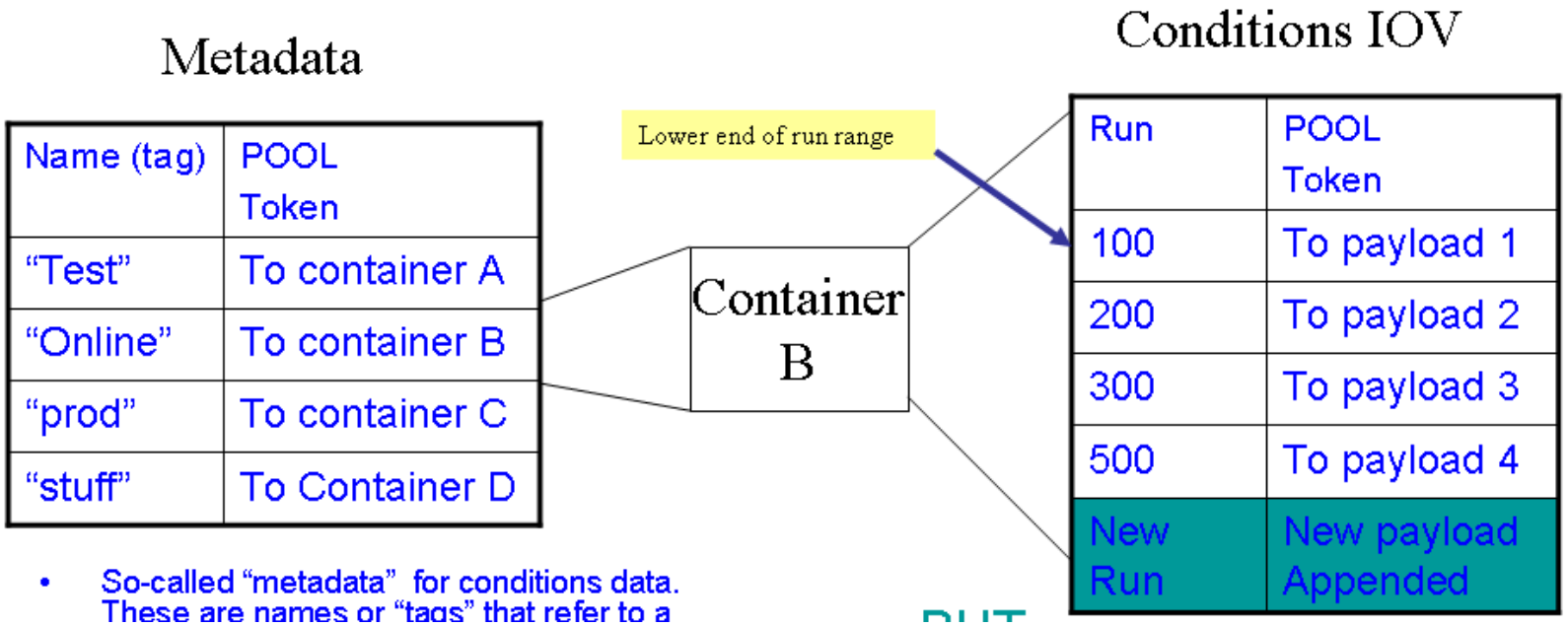


Conditions Terminology (1/2)

- CDF
  - CID: Calibration ID refers to a key, or unique identifier, for a calibration set. Each time a set of calibration constants is generated for a sub-detector a CID is assigned.
  - Valid-set: several sub detectors can be grouped in to a single valid-set.
  - Used-set: combines valid-set with the IOV.
- ATLAS
  - Uses COOL for their conditions DB
  - Cache alignment: an approach they use to "align" the queries so as to take advantage of the FroNTier cache.

Conditions Terminology (2/2)

- CMS



Monitoring

- CDF deployed a service at Fermilab which receives udp packets sent from each client. This info is stored in a central mySQL DB and can be mined to understand timing and access patterns.
- For FroNTier-like deployment, squid activity can be monitored with SNMP collectors. Squid logs provide detailed information for studying access patterns.

finis