

# Minerva Databases Work by FNAL CD

Minerva Collaboration Meeting

Igor Mandrichenko

December 3, 2010

# Who we are

- FNAL Computing Division
  - Scientific Programs Quadrant
    - Running Experiments Department (REX)
      - Scientific Database Applications Group (REXDB)
        - » 2 full time employees
        - » 2 part time
        - » 3.25 FTEs total, down to 2.75 FTE from January 2011

# Control Room Logbook

- MINERvA CRL is at <http://dbweb0.fnal.gov/minerva>
- One of ~20 instances of CRL we run
- dbweb0 is the front end to our redundant web servers cluster (currently of 2 computers, dbweb1 and dbweb2), helps us increase availability of CRL and other services

# Glaucus

- MINERvA Shifts Scheduling tool
- Developed and maintained by Nathaniel Tagg
- We deployed Glaucus at  
<http://dbweb1.fnal.gov:8080/glaucus/index.html>  
in January 2010
- To our knowledge, it is not being used by MINERvA

# MINERvA SAM Data Browser

- Migrated SAM Browser from MISWEB to QueryEngine
- Deployed MINERvA instance at:  
<http://dbweb0.fnal.gov/SAM/Minerva>  
Uses redundant web servers infrastructure

# Conditions Database

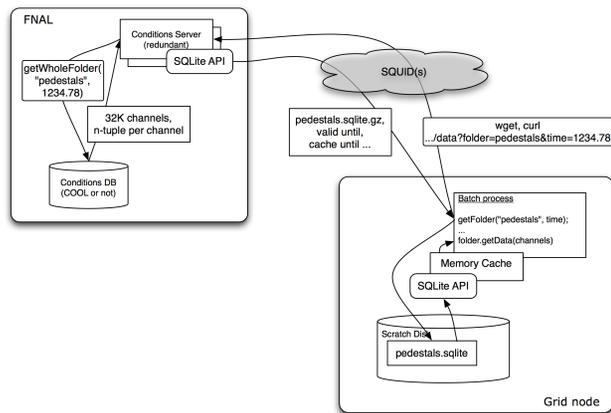
- Helping MINERvA since September 2010
- Investigated slow COOL performance
  - Margherita Vittone
  - Conclusion: the query used by Cool is not optimal for MINERvA case, can be significantly simplified
- Investigated what appears to be discrepancy between Oracle and MySQL representation
  - Federica Moscato
  - Found the reason for apparent data discrepancy
    - We believe that under some circumstances, while being written by COOL-based tools, data can get “corrupt” in the database
    - Replicating data from MySQL to Oracle “purifies” the data
  - Wrote a diagnostics tool which detects “corrupt” IOVs

# Conditions Data Delivery

- Project documents are at:

<https://cdcv.s.fnal.gov/redmine/projects/sci-db-app/wiki/Minerva>

MINERvA conditions data delivery



- Collected requirements
- Proposed web service - based data delivery architecture
- Designed simplified database schema specifically for MINERvA case

# Conditions Data Delivery

- Marc Mengel and I prototyped the system
  - Postgres database
    - ~600 pedestal calibration datasets
    - 32000 channels per dataset
    - 9 floating point numbers for each channel
    - Close to 1 year worth of data
  - Python API
  - Web server
  - Web client C++ API
  - Client running on “remote” node
- First performance test results:
  - 5 consecutive pedestals set fetches (2 of them cached) – 15.6 seconds
  - 150 non-cached fetches initiated simultaneously – last finished in 8 minutes
  - More at <https://cdcvs.fnal.gov/redmine/attachments/download/1617/MinervaConditionsDeliveryDemo.pdf>