



# CAF Monitoring

Hans Wenzel, Federica Moscato, Doug Benjamin

---



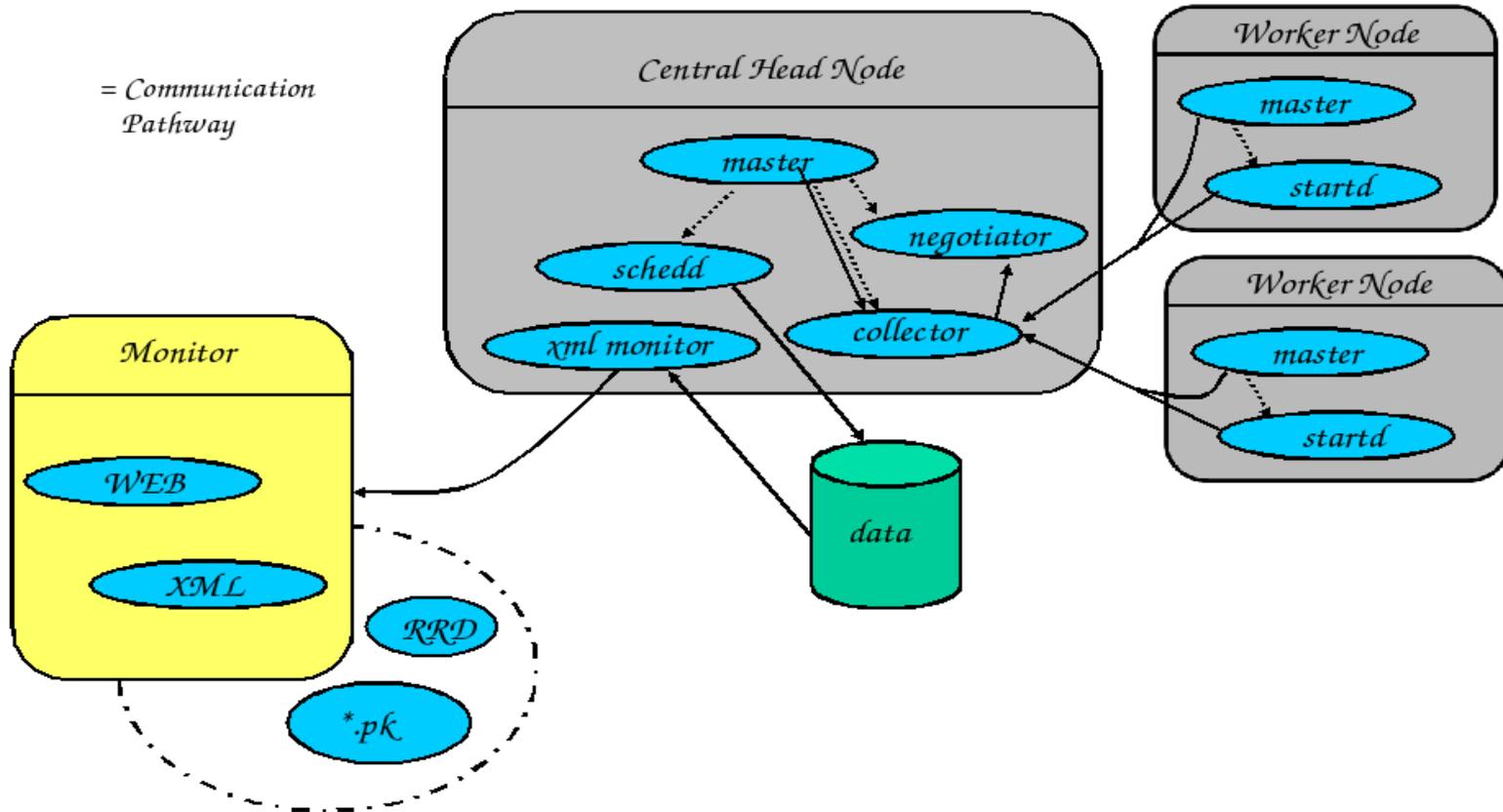
- What kind of Monitoring?
- The current monitoring System.
- Why update the system now?
- Requirements.
- Beyond a monitoring system.
- Path towards a better system, defining tasks
- Ask for your feed back

# What kind of Monitoring?

---

- Site/farm specific Admin/Operator view:
  - what's going on on the farm? who is doing what?
  - where do the jobs come from (grid)?
  - hardware/demons, services.
  - <http://cmsrv03.fnal.gov/condor>,
  - [zabbix](#), [nagios](#), [ganglia](#), [ngop](#)....
- User/Jobs/VO oriented view:
  - where on the world are my jobs running?
  - how are they doing?
  - .....

# The current monitoring system



# The current Monitoring system (cont.)

---

- **monitor:** provides the communication between the user and his jobs
- **xml\_monitor:** collects the data and provides the communications with the client system
- **state\_monitor:** produces the cache system.
- **client:** calls the monitor server and takes the data to store in python pk and RRD (round robin database) files.

# The current monitoring system (cont.)

---

- The data that is collected is well described (python dictionaries). There is a document in progress describing the current system.
- Creation of this data file is obtained via cron jobs in the background (comment: not very robust since cron jobs might fail to start when the system is very busy leading to gaps in the monitoring data)

# The current Monitoring system (cont.)

---

- To get appropriate key attributes into the .pk file, current model needs to parse many Condor and CAF log files or execute various condor commands. (not very robust as anyone knows who relied on parsing log files or the output of processes).
- Currently this causes many failures and it takes a lot of manpower fixing it.
- The user information can be accessed via the following web page.  
<http://cdfcaf.fnal.gov/groupcaf/user.html>

# Issues with current Monitoring system?

---

- sometimes Monitoring information is not recorded. Jobs run without leaving a trace in the Monitor.
- Very labor intensive
- Question: can better hardware (faster CPU more memory for the head node) remedy some of the problems?

# Why update the system now?

---

- move to the GRID makes more general solution necessary (must work for both LCG/OSG)
- use of new glideinWMS for OSG sites (need to keep track of glidein and VO classads). Considered for CMS so we might want to make use of tools developed for the LHC (SAM). Should see if some services can be provided by e.g. the group running the dashboard
- Question what's the submission mechanism on LCG sites?
- current system doesn't scale and is difficult to maintain. Need to be able to operate with less man power.

# Why update the system now?

---

- wish for more features currently not available, e.g. use of database will allow for specific queries
  - job failures/successes by user
  - job failures/successes by Error return code (Do we provide proper documented error codes?)
  - job failures/successes by site
  - glidein failures
  - user: what are “my” jobs doing
  - MC – data what is produced and where --> help in accounting, justify resources

# Requirements

---

- Robust.
- small impact on the System.
- simple configuration.
- use available and actively maintained frameworks and adhere to standards if possible. Reuse as much existing code as possible.
- provide at least the same 'useful' information as the current system.
- don't fix what ain't broken.
- allow for specific queries.
- webbased.
- allow for specific views (admin,user, summary) --> certificate based authentication.
- allow to monitor multiple sites (LCG and OSG).

# Beyond monitoring: Make operation easier

---

- allow for data mining, reports, summaries and discovery of e.g. Bottlenecks of the system.
- automatic detection of errors state, issue alerts and trigger 'automatic' action. Issue warnings that make sense and might include suggestions how to remedy the problem. (e.g. if jobs are held because a proxy expired, message should say that and point to instructions how to renew the proxy and release the jobs) (**ErrorLogger, ErrorReceiver, MessageSystem**)
- System should include a **StateManager** that checks that all demons and processes are running and reports the status to the web. (see online Monitoring)

# Path towards a better system, Tasks

---

- replacing pk files by db/xml?
  - could be a first step to get some experience.
- Document and review the current system.
  - xml description might be useful since it can be used on transport layer.
- A look at Quill (condor specific).
- A look at the ARDA Dashboard.
- A look at zabbix

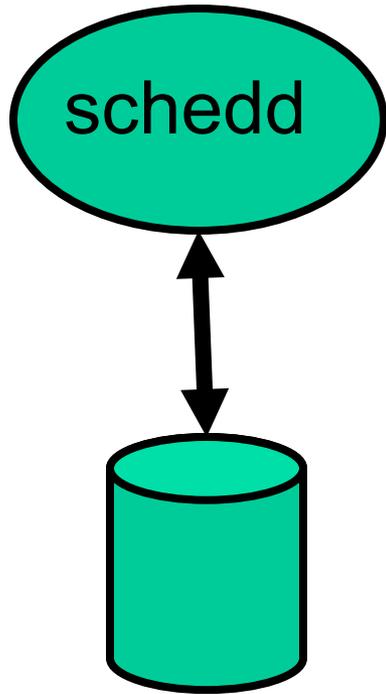
# What is Quill (actually Quill++)?

---

- A non-invasive method of storing a read only version of the job queue and job historical data in a relational database (POSTGRES). Now quill++ stores entire condor information.
- Presents the job queue information as a set of tables in a relational database. Querying the information should be more robust.
- Fault tolerance ??.
- Provides performance enhancements in very large and busy pools (True??). Distribute services.

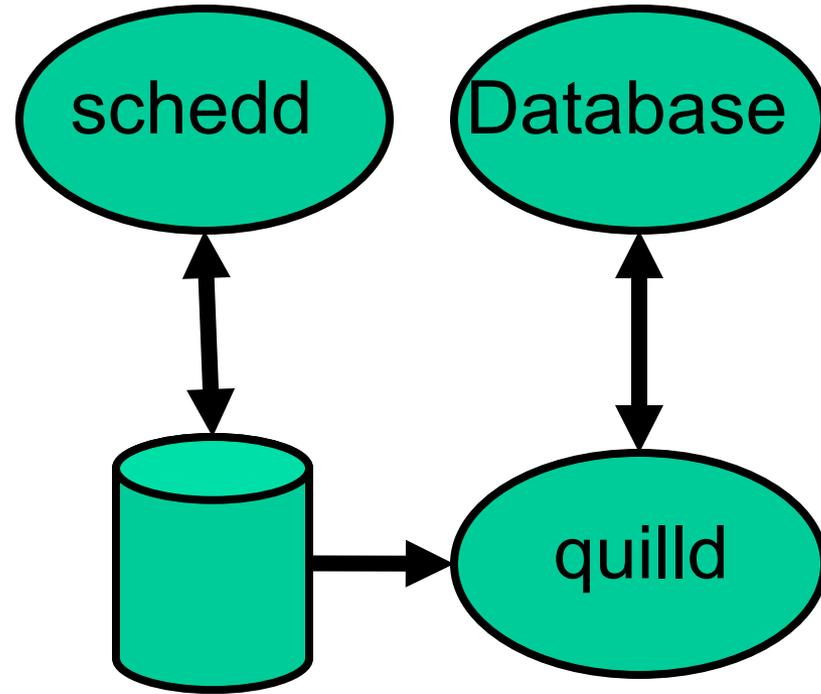
# Job Queue Management

Without Quill



Job Queue

With Quill



Job Queue

# A User Perspective:

---

## Example: condor\_q -name

```
Linux merlin > condor_q -name psilord_quilld@merlin.cs
```

```
-- DB: psilord_quilld@merlin.cs : <merlin.cs.wisc.edu:42999> : psilord_db
ID      OWNER      SUBMITTED      RUN_TIME ST PRI SIZE CMD
 92.0   psilord      4/21 09:21     0+00:00:00 I  0   9.8  foo
```

```
1 jobs; 1 idle, 0 running, 0 held
```

## Example: condor\_history -name

```
Linux merlin > condor_history -name psilord_quilld@merlin.cs
```

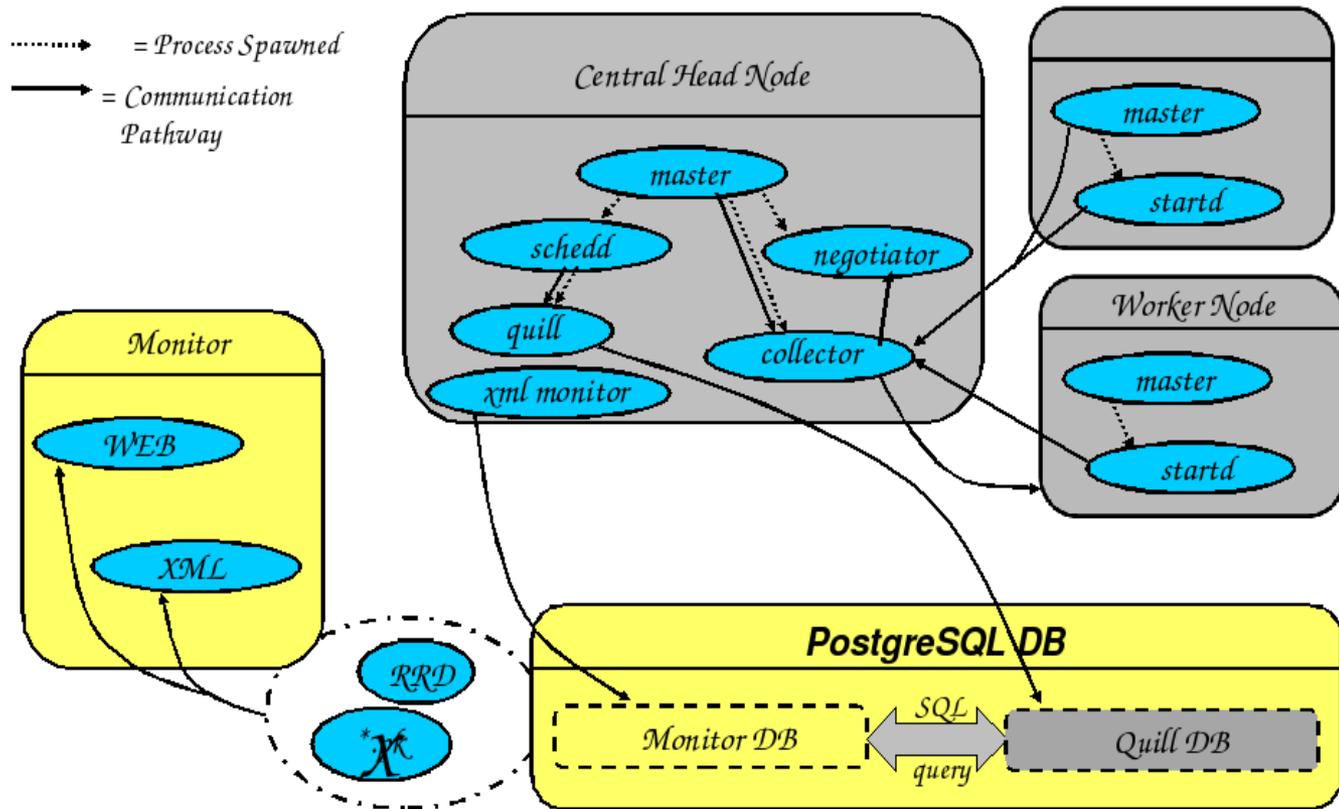
```
-- DB: psilord_quilld@merlin.cs : <merlin.cs.wisc.edu:42999> : psilord_db
ID      OWNER      SUBMITTED      RUN_TIME ST  COMPLETED CMD
 91.0   psilord      4/20 14:23     0+00:00:00 X   ???       /scratch/psilor
 92.0   psilord      4/21 09:21     0+00:00:00 X   ???       /scratch/psilor
 93.0   psilord      4/21 10:12     0+00:00:01 C   4/21 10:12 /scratch/psilor
```

# Evaluate Quill easy first step

---

- installation and data tables are well documented. Last Friday Federica and me installed it on the test system: fcdftest010.
- But:
  - still need to test (this week)
  - should also evaluate the hardware and see if upgrade necessary.
  - need to deal with db maintenance.

# One possible framework

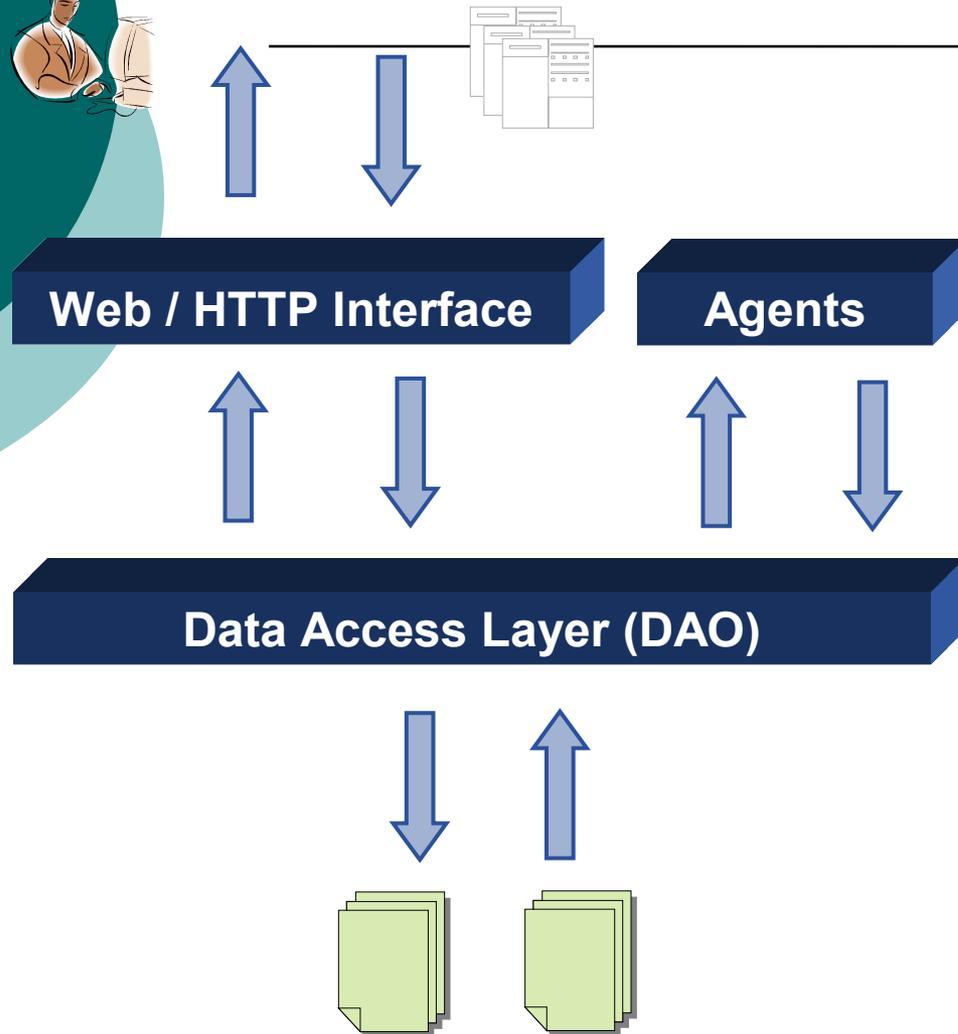


# ARDA Dashboard

---

- Used by all LHC experiments:  
<http://arda-dashboard.cern.ch>
- provides monitoring for both OSG and LCG sites
- supported by grid collaboration
- well documented
- did not take long to get the development version installed.
- nice environment eclipse/netbeans

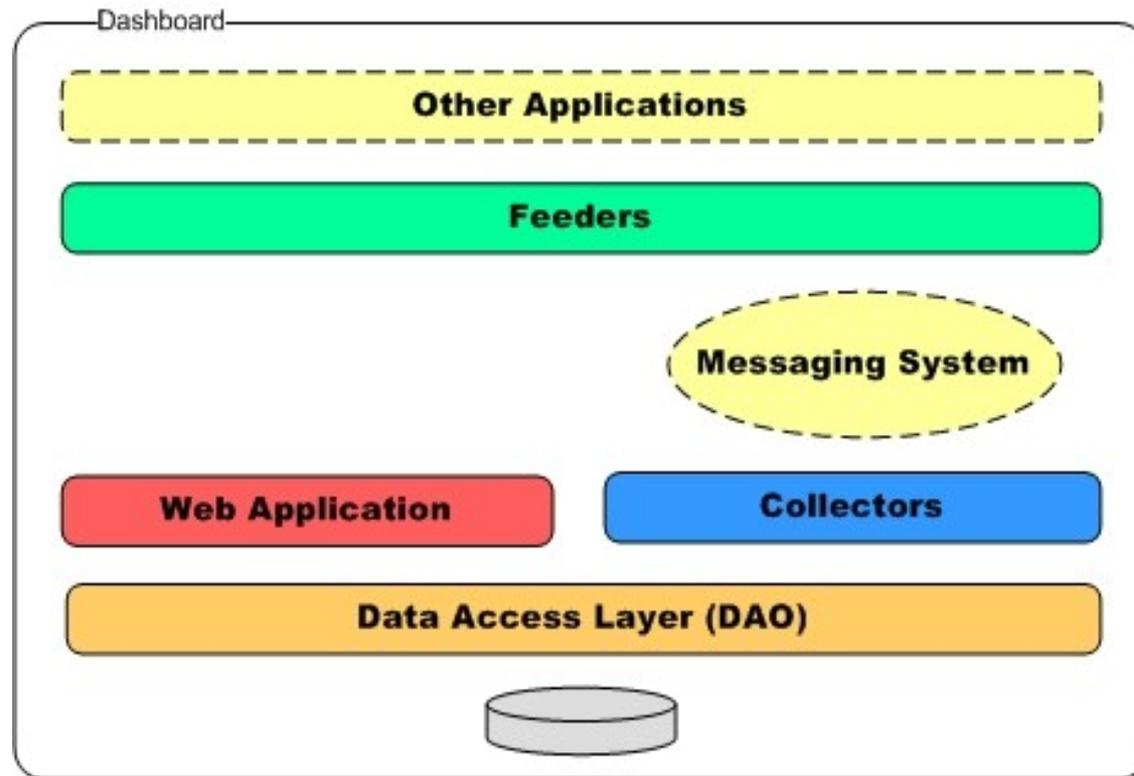
# Dashboard Framework



- Dashboard Clients
  - Scripts: pycurl, ...
  - Command line tools (optparser + pycurl)
  - Shell based: curl, ...
- Web Application
  - Apache + mod\_python
  - Model View Controller (MVC) pattern
  - multiple output formats: plain text, CSV, XML, XHTML
  - GSI support using gridsite
- Agents
  - collectors: RGMA, ICXML, BDII, ...
  - stats generation, alert managers, ...
  - *Service Configurator* pattern
  - common configuration (XML file) and management: *stop, start, status, list*
  - common monitoring mechanism
- Data Access Layer (DAO)
  - interfaces available to different backends (Oracle and PostgreSQL mainly, easy to add additional ones)
  - connection pooling

# Dashboard Components

---



- Build and development environment
  - based on python distutils (with several extensions)
  - covers code validation, binaries and documentation generation, unit testing and reports

---

  - automatic build for each of the release branches
  - packaging uses RPMs – APT repository available
- Release procedure
  - three main branches: nightly, unstable, stable
  - releases per component
  - enforced versioning scheme (no manual versioning or tagging, all done via distutils command extensions)
- Interesting links
  - Developers guide:
    - <http://dashb-build.cern.ch/build/nightly/doc/guides/common/html/dev/index.html>
  - Savannah Project
    - <http://savannah.cern.ch/groups/dashboard>



# Request for feed back

---

- Please send us feed back!