

Experiences with Grid Interfaces on CMS Farm

July 26, 2004



The CMS Farm

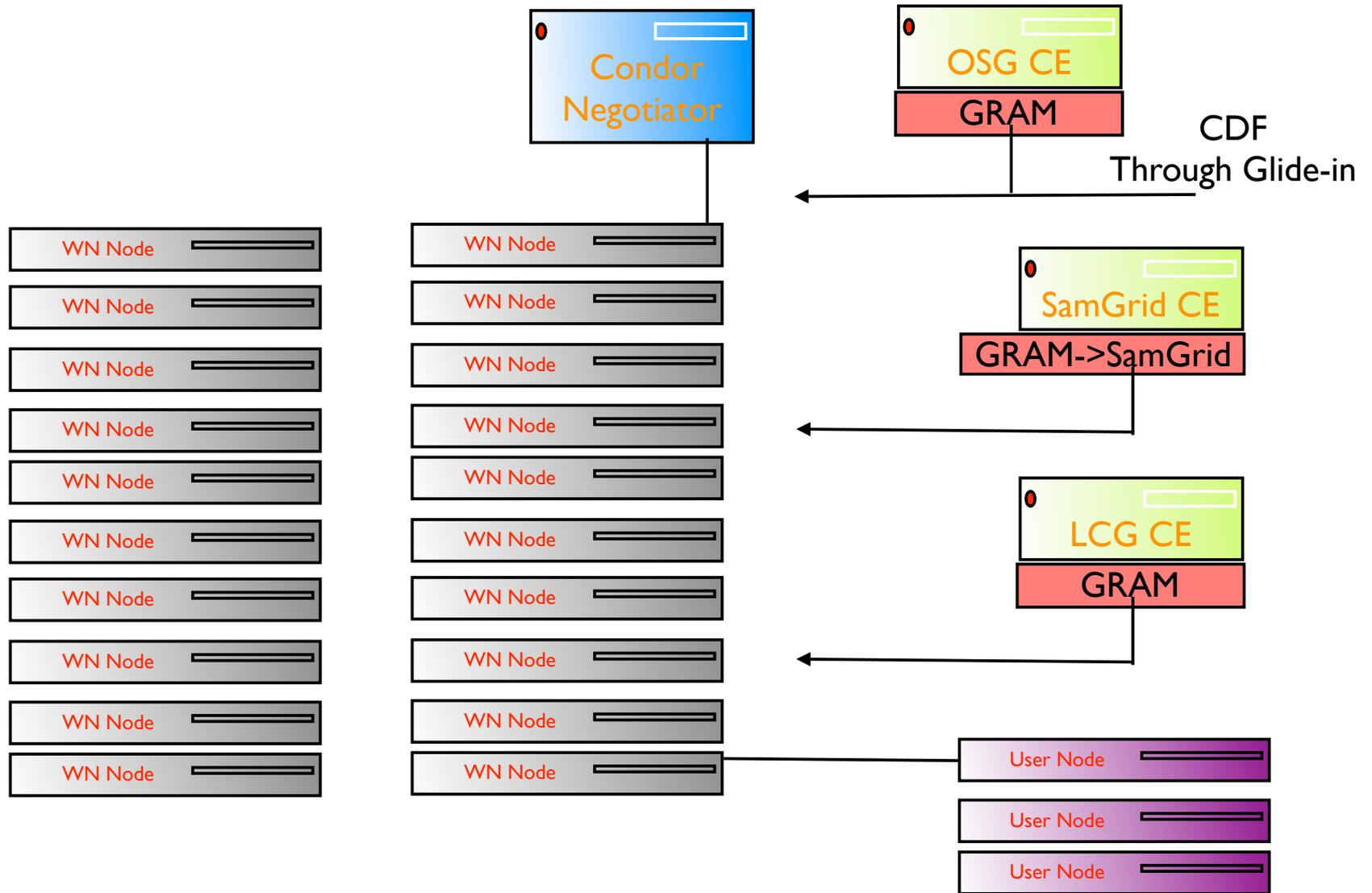


CMS has been consolidating on one batch farm for organized simulation and user analysis work

- ➔ Use batch queue policies and priorities instead of hard partitioning of resources to support multiple applications
- ➔ Should reduce the effort required to operate the cluster
- ➔ Still need some development to protect against misconfigured user jobs
 - Investigating submission wrapper. We are protected against long running jobs, but doesn't protect itself against taking too much of other resources (memory)

Goal is to make this farm accessible for local User Analysis Farm (UAF) submission, grid submission, submission, desktop submission, etc

- ➔ Farm is currently 530 batch slots. Majority of these are from Intel Xeon, though about 100 are Athlons
- ➔ First rack of Opteron systems will go into production next week
 - Farm capacity should double by the end of the fiscal year





Multiple Grid Interfaces



CMS has been able to support multiple Grids and VOs at FNAL

- ➔ LCG-2 Computing Element (CE)
 - Support International ATLAS and LCG testing
 - Heavily used only by CMS analysis applications through CMS Remote Analysis Builder (CRAB)
- ➔ OSG CE
 - Currently support all OSG VOs. Reasonable use by GRACEVO and ATLASVO. Heavy use by US-CMS central production
- ➔ Sam Grid to support DZero Reconstruction
 - Data Served from DZero SAM Stations
- ➔ CDF through the OSG Interface with Glide-in
 - Opportunistic use that takes advantage of proximity of CMS farm
- ➔ Deployed dCache SRM Storage Element (SE) for LCG and OSG

Reasonably successful demonstration of utilization by multiple VOs and sharing at the facility level. Fair bit of custom adjusting



LCG-2



LCG and OSG on the surface use many similar components

- ➔ VOMS for VO management
- ➔ GRAM for processing submission and SRM for transfer management

Unfortunately they vary at the details

- ➔ Globus version is older in LCG and probably prevents the use of OSG privilege callout
 - Fine grained authorization is handled in a different way
 - GRAM Job Manager in LCG is superior to what is used in OSG, but it's probably specific to Globus version.

They also vary in the installation and configuration approach

- ➔ LCG assumes the environment on the worker nodes and the availability of a number of client tools. (Mods to /etc/profile)
- ➔ Because OSG does not assume environment variables and only access to VDT tools, the two can be made to work together
 - As OSG deploys higher level services this needs to be checked



Issues Supporting LCG



A lot services have to succeed for a job to be successful

- ➔ Tight coupling of the success of the information provider and the resource broker submission
- ➔ Grid Monitor keeps track of job state, if this information system fails the user can't get the output back
- ➔ Both of these have failed at times. There is reasonable LCG site validation tests that run periodically throughout the day we usually receive notification from the angry e-mail monitoring system first

There is a general issue for both OSG and LCG that the users fly blind

- ➔ Applications memory leak and have to be killed, the difference between natural death by unstable services and intentional death by operator is difficult to determine from outside
 - Users often resubmit. Notification involves tracking the user down. Sometimes falling back to Google.
- ➔ There are a number of projects to aid in this, but nothing widely deployed or accepted.



OSG has a production release at the beginning of July

- ➔ At the time of the ribbon cutting there were 15k batch slots enabled for OSG

LCG relies heavily on submission services from the RB and the other infrastructure for sand boxing, job tracking, output registration etc.

OSG relies heavily on submission infrastructure developed by the VOs themselves.

- ➔ This has resulted in a surprising flexible and reasonably robust grid.
- ➔ It also increases the number of support and operations issues, because the number of ways of working increases
 - VOs specify how they want account mapping and roles handled
 - In the process of switching how output files are dealt with (-> SE)
 - VOs have implemented different ways of bulk submission

OSG is heterogeneous but the diversity comes at a price



SAM Grid



Installed SAM Grid JIM onto our original Grid3 interface

- ➔ SAM Grid appears as a second job manager beside our original GRAM pipe to Condor
 - A submission to this job manager turns into multiple submissions to Condor directly
- ➔ In addition to the job manager you need a local file stager
- ➔ We rely on having a local SAM Station run by DZero to deliver data.

Intrinsically this seems pretty simple to describe, but the installation and commissioning took a long time

- ➔ Instructions were reasonable, but we were installing on an existing grid interface.

After the installation the interface has been remarkably stable.

- ➔ The architecture provides for reasonable scaling
- ➔ Small issues with Condor's desire to resubmit

10 CPU Years delivered at of today.



CDF Glide-in



CDF Glide-in is the newest entry to the CMS farm

- ➔ Discovery jobs are sent in through the OSG interface
- ➔ When they receive a batch slot they register themselves back to the CAF and accept an application

Relies on a couple of assumptions

- ➔ Data is close by. For Analysis data is left on the CDF dCache and served to the worker nodes over high bandwidth low latency links
- ➔ We trust CDF. We only know that a Grid certificate from a trusted CDF member was used to submit the pilot. We trust the CAF kerberos infrastructure to verify the actual user
- ➔ CDF Software Server is mounted from the worker nodes. This should be solvable, but currently there is an expectation of software.

Purely opportunistic use, which should be transparent to users.

- ➔ Simple, for CMS, to setup



Interoperability and Sharing



The CMS farm is a successful example of providing access to a resource and sharing it using defined policy

- ➔ LCG, OSG, SamGrid, and CDF plus all the accompanying VOs have successfully used the farm

It's not a great example of Interoperability

- ➔ Essentially we have been able to weave everyones requirements together into a functional system
- ➔ We have one set of physical nodes but several interfaces
 - I'm don't know that this is a bad thing
 - Operationally the load increases
 - The functionality is maintained and we don't end up with lowest common denominator interface layers.