Fermilab

CMS FroNTier Deployment and Performance Update

Lee Lueking
CMS Offline Software and Computing

1 September 2007

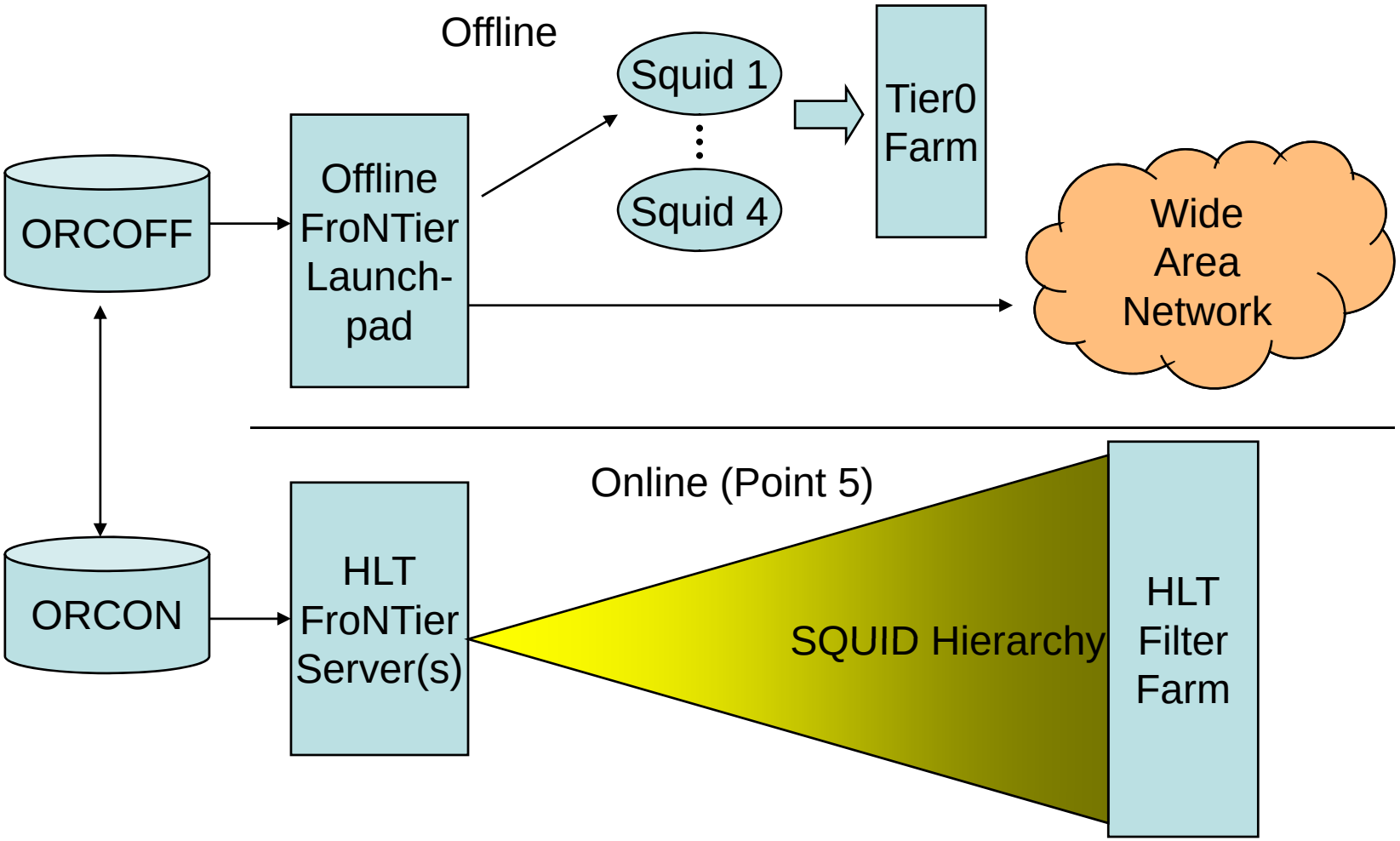
WLCG DB BoF 2007

Outline

- Deployment Overview
- Cache Coherency
- Performance for HLT and Tier0 farms
- Other performance improvements

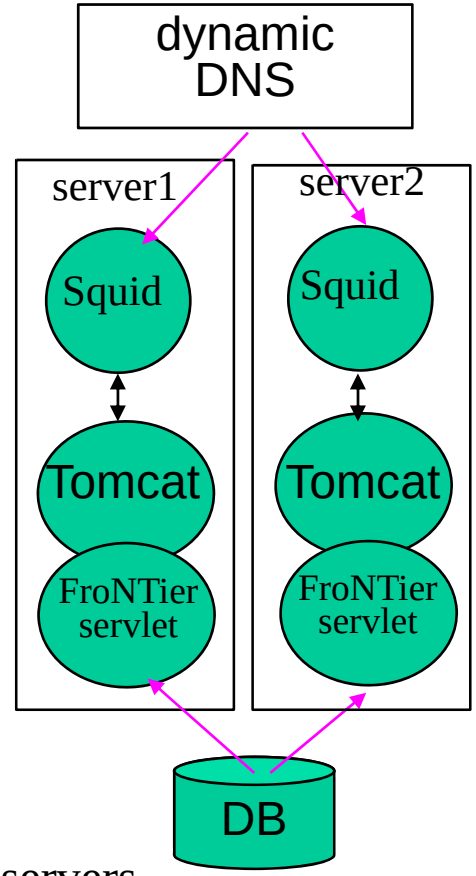
Acknowledgements
•The Frontier Team: Barry Blumenfeld (JHU), David Dykstra (FNAL), Eric Wicklund (FNAL)

Overview


Offline
Squid 1
Squid 4
Tier0
Farm
ORCOFF
Offline
FroNTier
Launch-
pad
Wide
Area
Network
Online (Point 5)
ORCON
HLT
FroNTier
Server(s)
SQUID Hierarchy
HLT
Filter
Farm

FroNTier "Launchpad"

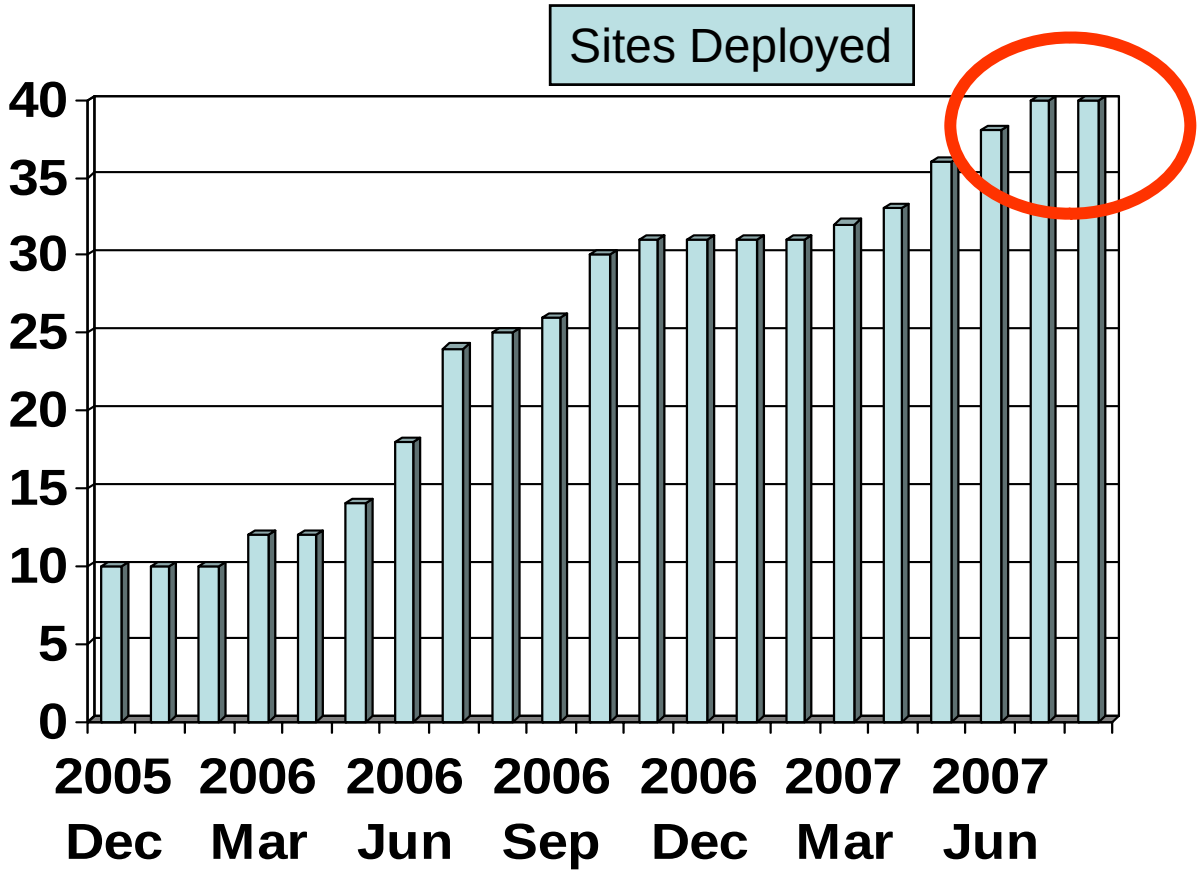
- Squid caching proxy
  - Load shared with Round-Robin DNS
  - Configured in "accelerator mode"
  - "Wide open frontier"*
  - Peer caching removed because it was incompatible with collapsed forwarding
- Tomcat - standard
- FroNTier servlet
  - Distributed as "war" file
    - Unpack in Tomcat webapps dir
    - Change 2 files if name is different
  - One xml file describes DB connection



•The squids in the launchpad ONLY talk to the Frontier Tomcat servers. No registration or ACL's required.

Squid Deployment Status

- Late 2005, 10 centers used for testing
- Additional installation May through Oct. 2006 used for CSA06
- Additional 20-30 sites for CSA07 possible
- Very few problems with the installation procedures CMS provides.



Cache Coherency (1/2)

Metadata

Name (tag)	POOL Token
“Test”	To container A
“Online”	To container B
“prod”	To container C
“stuff”	To Container D

Lower end of run range

Container B

Conditions IOV

Run	POOL Token
100	To payload 1
200	To payload 2
300	To payload 3
500	To payload 4
New Run	New payload Appended

- So-called “metadata” for conditions data. These are names or “tags” that refer to a specific set of IOV’s (Interval Of Validity) and associated payloads in the pool-ora repository.
- By decree, data in Conditions IOV can NOT change.
- Therefore, caching is OK.

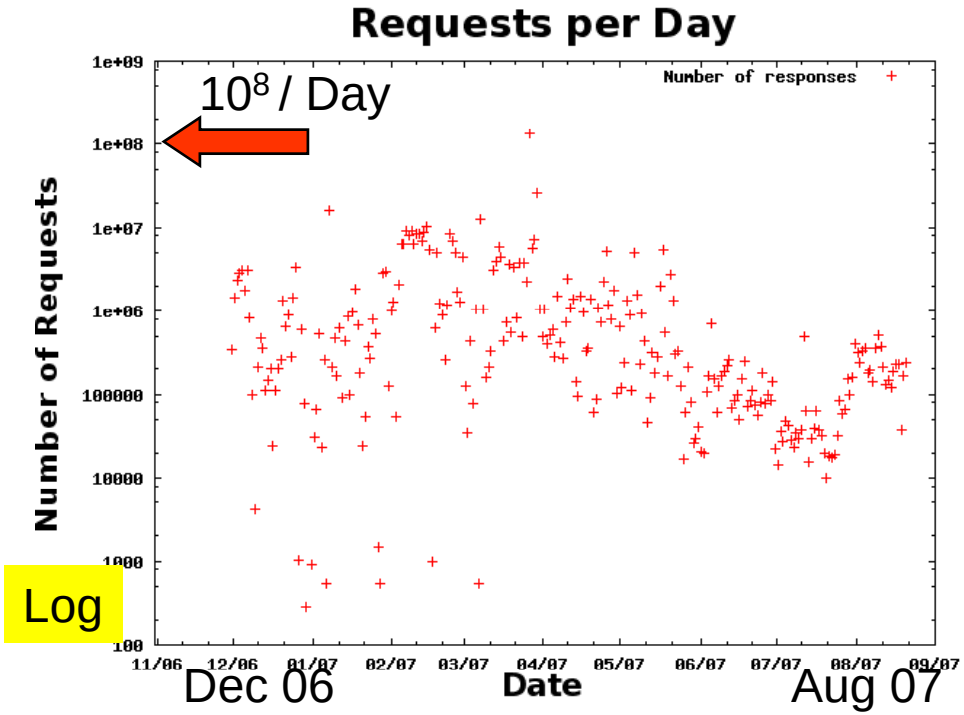
- BUT…
- New IOV’s and payloads can be appended to in order to extend the IOV range.
- This is NOT OK for caching…

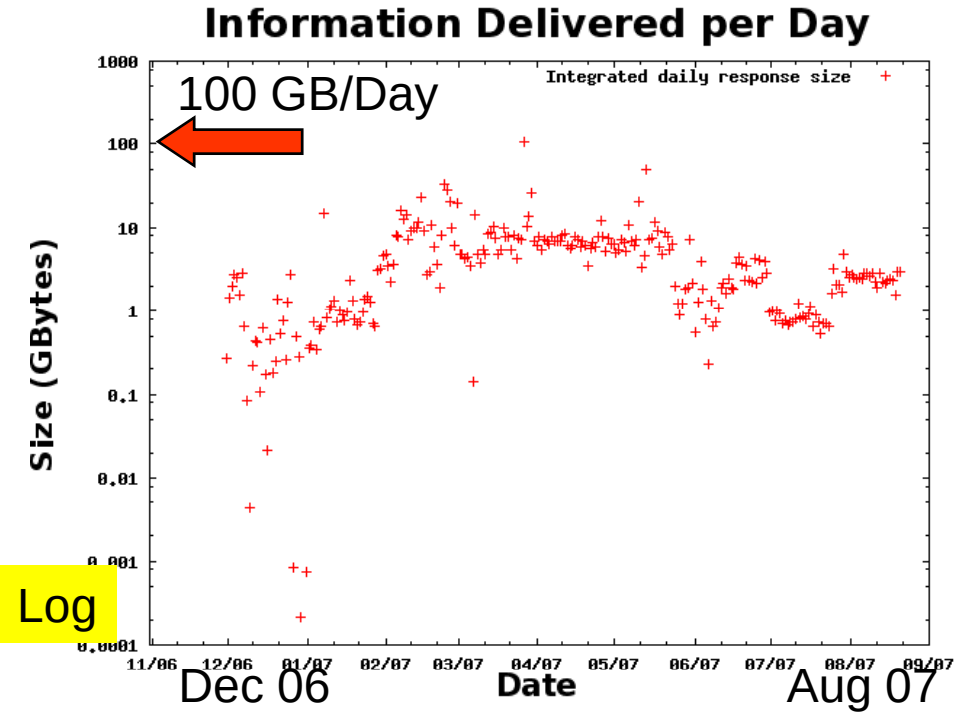
Cache Coherency (2/2)

- Adopted the following solution:
  - All objects that are cached have expiration times
  - Shortlived: Metadata objects, including the pointers to payload objects, expire on a short time period
  - Longlived: Payload objects have a long expiration time.
- The values of the short and long times varies depending on where the data is being used:
  - Online: the calibrations change quickly as new data is added for upcoming runs.
  - Tier 0: calibrations change on the order of a few hours as new runs appear for reconstruction.
  - Tier 1 +: Conditions data may be stable for weeks.
- The value of the short and long expirations is modified at the FroNTier server, so can easily be tuned as needed.

Launchpad Operation

All activity on Launchpad: Production, Development and Testing





- Number of daily objects varies widely
- Peak day (10**8) was fail-over from a local squid, good test of infrastructure.

- Object size depends on type of activity occurring

Example Calibration and Alignment Object Sizes (Monte Carlo)

- Table shows examples of the size of conditions data for a few sub-detectors
- Zipping of data can significantly reduce the size of network transfers, at the cost of some server and client performance.
  - Online and transfers local to CERN it does not help
  - For sites remote to CERN it is useful
- Object sizes and zipping factors for real detector data may be quite different than for MC

Detector Sub-System (not all systems included)	Data size (MB)	
	Non-compressed	Compressed (zipped)
HCAL	1	0.4
ECAL	7	3.2
Drift Tubes	12	?
Si Track	20	?
Pixel Track	130	?
Current Total in MC	280	?

Compression factors unrealistic for MC data

Specific Challenges

- HLT (High Level Trigger)
  - Startup time for Cal/Ali < 10 seconds.
  - Simultaneous
  - Uses hierarchy of squid caches
- Tier0 (Prompt Reconstruct)
  - Startup time for conditions load < 1% of total job time.
  - Usually staggered
  - DNS Round Robin should scale to 8 squids

Parameter	HLT	Tier0
# Nodes	2000	1000
# Processes	~16k	~3k
Startup	<10 sec all clients	<100 sec per client
Client Access	Simultaneous	Staggered
Cache Load	< 1 Min	N/A
Tot Obj Size	100 MB*	150 MB*
New Objects	100% / run*	100% / run*
# Squids	1 per node	Scalable (2-8)

* Worst case scenario

Starting Many Jobs Simultaneously

Online HLT Problem

- All nodes start same application at the same time
- Pre-loading data must be < 1 minute
- Loading data to jobs must be < 10 seconds
- Estimating 100MB of data, 2000 nodes, 8 jobs/node
  - 100 * 2000 * 8 = 1.6TB
- Asymmetrical network
  - Nodes organized in 50 racks of 40 nodes each
  - non-blocking gigabit intra-rack, gigabit inter-rack

Starting Many Jobs Simultaneously

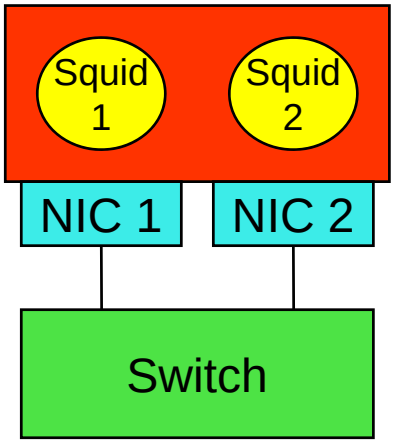
Online HLT Solution

- Each squid feeding 4 squids means 6 tiers for 2000 nodes
  - 50 racks reached in 3 tiers, 3 tiers inside each rack
  - bottleneck becomes the conversion from DB to http in FroNTier server.
  - Compression, in this case, makes overall performance worse.
- Measurements on test cluster indicate requirements can be met
  - Preloading takes 30 – 40 seconds.
  - 10-second loading always reads from pre-filled local squid.

Testing Squids w/ Multiple NICs

Potential Tier-0 and Tier-1 Augmentation

- On multi-processor/multi-core machines CPU resources under utilized. Using multiple network interfaces can resolve this.
- Two approaches tried
  - Multi-homed (2 ip addresses): machine looks like multiple nodes
  - Bonded interfaces (1 ip address): multiple NICs used together to increase throughput.
- Using Bonded approach at FNAL
  - Works best w/ specific load balancing approach
  - Two squids run on each server machine providing ~200 MBps throughput (Squid single threaded).
- Many sites, for now, prefer just adding more machines to make network management simpler. This may change.



Performance Improvements

- ✓ Reduced/eliminated forced cache refreshes.
- ✓ Number of queries for some objects reduced from 27k ➔ 3.
- ✓ Reduced number of HTTP connections by holding open TCP connection.
- ✓ Reduced client CPU by 90%
- ✓ Servlet compression time reduced from ~80s to ~4s.
- ✓ Optimized Squid memory cached object sizes (not intuitive).
- ✓ Simultaneous client startups now share queries.
- ✓ Looked at 64 bit SLC4 performance, see 5 to 10% improvement.
- ✓ Many details and other studies described at: https://twiki.cern.ch/twiki/bin/view/CMS/FrontierPerformanceImprovements

Summary

- Continuing to carefully monitor the FroNTier deployment at Tier 0, 1, and 2 centers.
- Looking for ways to meet the needs of the HLT conditions loading. Tests w/ squid on each node seems to meet requirements.
- Many other optimizations have been found and are being used in the system.

Finish

References