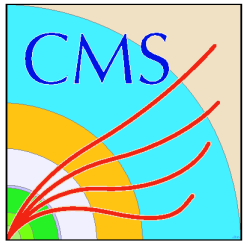
Frontier and HTTP Caching in the Future

Database Futures Workshop
Dave Dykstra, Fermilab
dwd@fnal.gov

Work supported by the U.S. Department of Energy under contract No. DE-AC02-07CH11359

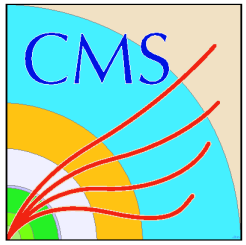
Outline

- What's good about Frontier & HTTP Caches
  - Also what their limitations are
- Current status & performance
- Recent improvements
- Planned improvements
- Expected impacts of increased usage



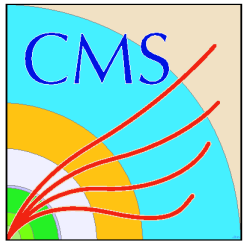
What's good about Frontier

- Frontier converts SQL queries to HTTP and back to native DB interface
  - Much better long distance performance than Oracle because single request/response
  - Makes good use of standard proxy caches
- Great for when there are many clients doing the same query close together
- CORAL interface gives same API for other DBs
- Protocol easily extensible beyond SQL



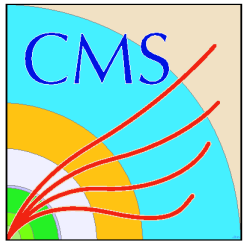
What's good about HTTP caches

- HTTP is designed for internet-sized scaling
  - Minimal overhead
  - Designed to be cached
  - Efficient, flexible, & elegant cache coherency
    - Server simply sets expiration time & last-modified time
- HTTP proxy caches can be easily inserted wherever repeated requests occur
  - Can chain as many as needed
- HTTP caches require little maintenance
- Multiple robust implementations to choose from



Frontier limitations

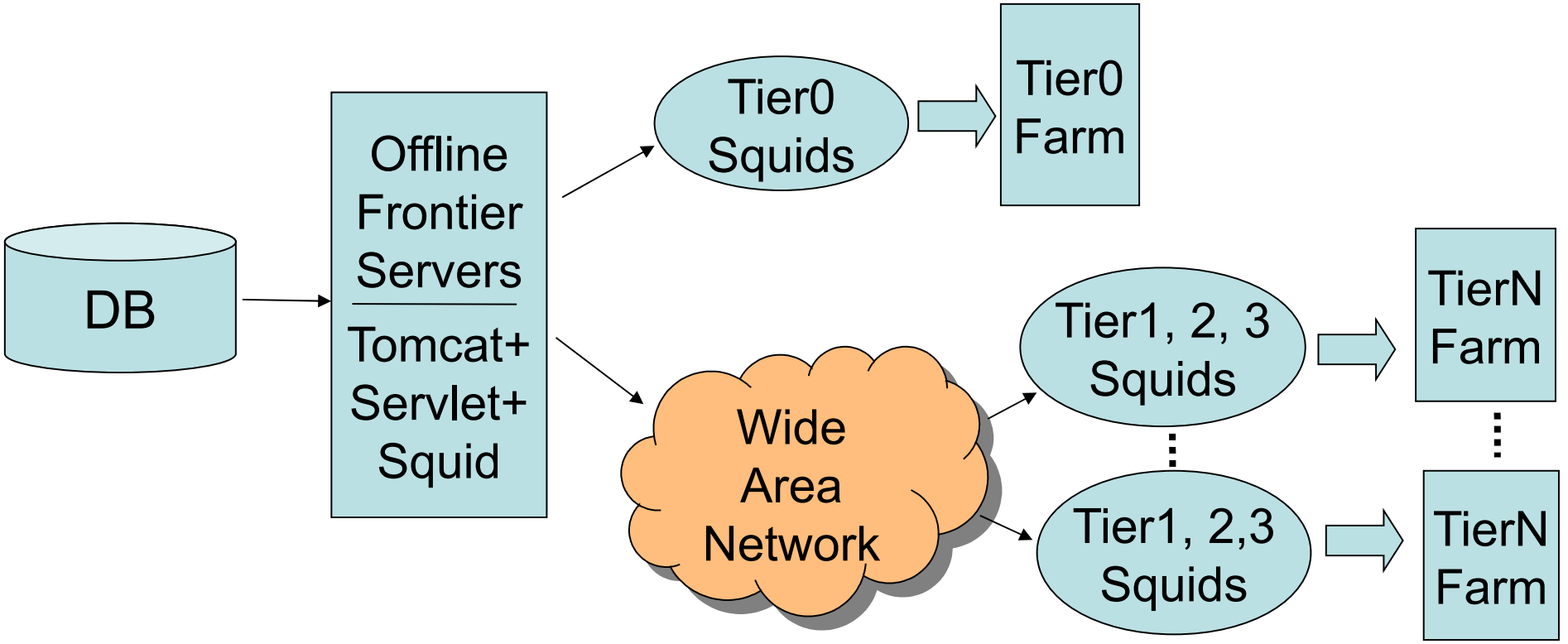
- Read-only
- Public data – no authorization
- Subset of SQL supported (e.g. no transactions, SELECT only)
- Not suited for large numbers of different queries at about the same time
- Works best with smallish responses (<~100MB)



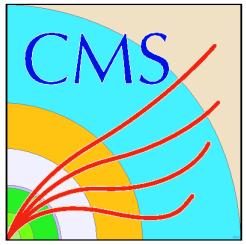
Current status

- Used for all conditions data reading in CMS
  - Offline & Online
- Used for conditions data in ATLAS for analysis work flows
- Also used by CMS for Luminosity data and recently for transferring & caching small files
- CMS has very effective monitoring of servers and worldwide squid caches
  - ATLAS working on it, new person just hired

CMS Offline Frontier



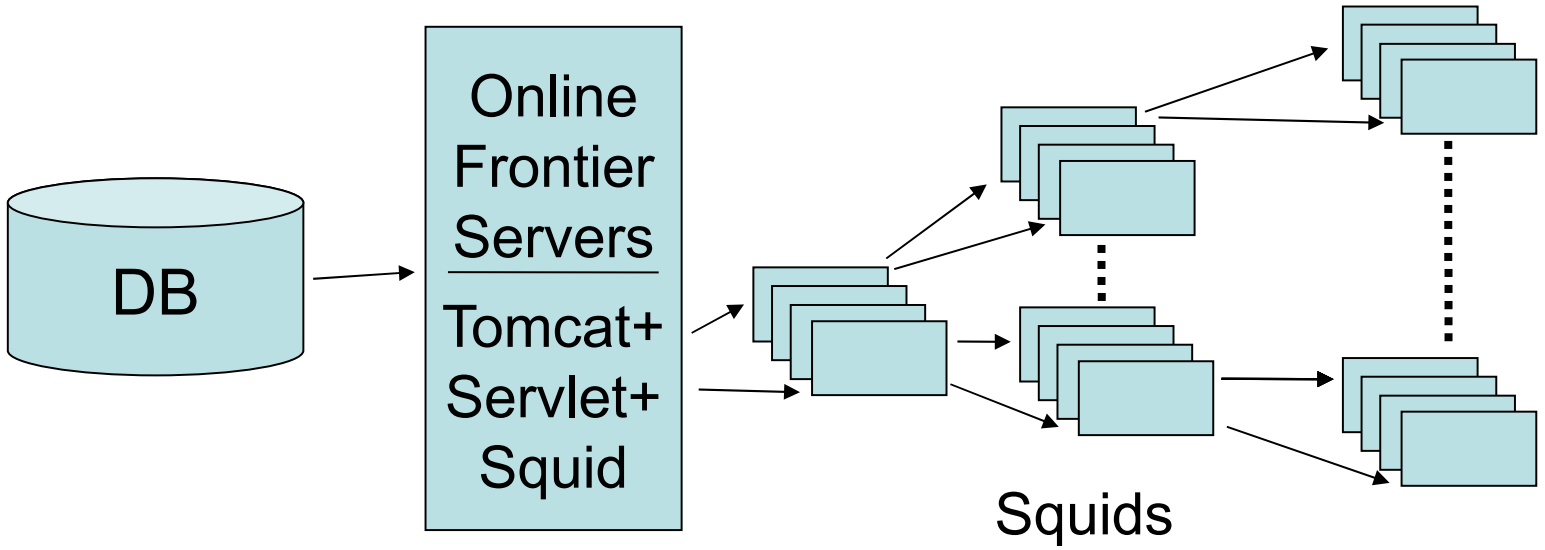
- Many copies of frontier_client in jobs on the worker node farms
- Jobs start around the world at many different times
- Cache expirations vary from 5 minutes to a year



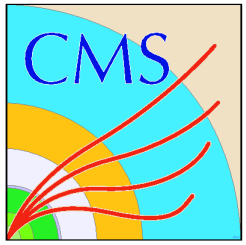
CMS Offline Frontier stats

- Average of 250 job starts per minute worldwide
- Average 500,000 total Frontier requests per minute, aggregate average total 500MB/s
- The 3 central squids at CERN only get 6,500 total requests per minute, and send 0.7MB/s
  - Factor of 77 improvement on requests and 715 on bandwidth
- Vast majority of jobs read very quickly because results already cached & valid in local squids

CMS Online Frontier



- Blasts data to all 1400 worker nodes in parallel
- Hierarchy of squids on worker nodes
- Each node feeds 4 others

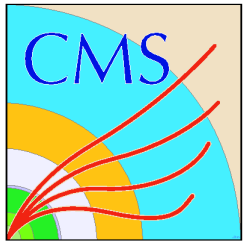


CMS Online Frontier stats

- Roughly 100MB of data loaded to all 1400 nodes in parallel in about 30 seconds, effectively an aggregate of almost 5GB/s
- Cache expires in 30 seconds so every run start verifies that every query is up to date
  - Most of the time, most of it is up to date so very little is actually transferred over the network

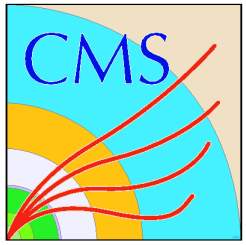
Recent improvements (1)

- New Frontier extension now used by CMS to transfer & cache small files (not SQL-based)
  - No change needed to client library (added small command-line program), small extension on server
  - Similar function to wget+apache but has all the advantages of existing robust infrastructure, retries
  - Not POSIX like CVMFS but convenient where Frontier already deployed
    - Potentially also more appropriate for frequently changing files



Recent improvements (2)

- CMS has new automated warnings to owner of a site cache when too many requests from a site fail over to read directly from central servers
- CMS has new graphing of server request queue lengths for each servlet, and warnings to operators when any queue reaches 75% of limit

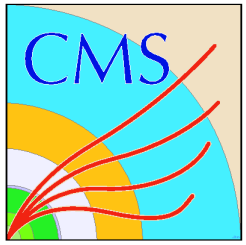


Planned improvements (1)

- Replication of CMS conditions/lumi database & Frontier servers at Fermilab
  - Wait for Oracle 11g's Active Data Guard after winter shutdown because far less DBA work than streams
  - Will allow Frontier operations to continue when CERN's Oracle or WAN connection down
  - ATLAS currently streaming conditions ~5 Tier 1s
    - Some have shut down, still too many in my opinion

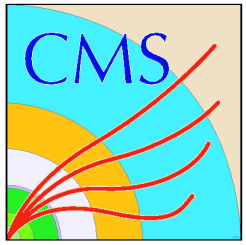
Planned improvements (2)

- Deploy additional backup proxy squid servers for CMS
  - Co-located with Frontier “launchpad” (tomcat+squid) servers
  - Coupled with disabling fail-overs to server, they keep the launchpad servers free from fail-overs
    - Better for serving the squids of normally functioning sites
    - Only failing sites will be harmed if too many site's fail-over at the same time
  - The fail-over monitoring that's now on the launchpad squids will be moved to these



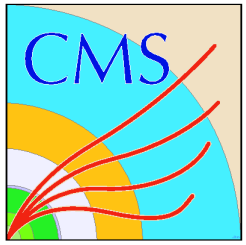
Planned improvements (3)

- Automate the configuration of MRTG monitoring of worldwide squids
  - From AGIS configuration database for ATLAS
  - From CVS copies of site-local-config for CMS
    - Currently CMS MRTG squid configuration maintained by hand but recently added automated audit to compare it to CVS
  - Eventually probably from BDII (more about that later)



Planned improvements (4)

- Authentication of sources
  - Currently vulnerable to man-in-the-middle + buffer-overflow attack
    - Obscure, but potentially highly valuable
  - Overcome the threat by adding to the response a digital signature of the request+most of response
- Use squid3 when it is ready
  - Total rewrite of squid2 in C++, multithreaded
    - Should handle higher bandwidth on multiple cores
  - Some important functionality still missing



Increased usage expected

- CVMFS using HTTP squids
- LHCb making plans to use Frontier
- Increased applications of both Frontier & HTTP caches by LHC experiments likely
  - Also other experiments sharing the same grids
- Natural growth of bandwidth demands for existing applications

Results of increased usage (1)

- Increased bandwidth will be needed
  - At a minimum add to bandwidth or replicate existing site squids
  - May eventually need to have heirarchy of squids at sites, such as a squid per rack fed from site squid

Results of increased usage (2)

- Need a standard method for automated discovery of HTTP proxies
  - BDII most likely
  - Proxies should be shared for all production, approved applications
  - Also should be separate, opportunistic proxy caches to avoid interference with production

Summary

- Increased usage of both Frontier & HTTP proxy caches expected
- Need a standard method for discovering proxies
- ATLAS monitoring being brought up to the level of CMS and a bit beyond it
- http://frontier.cern.ch