Fermilab

Frontier and Squid for same data access by many jobs

Dave Dykstra
dwd@fnal.gov

OSG Users' meeting
27 July 2007

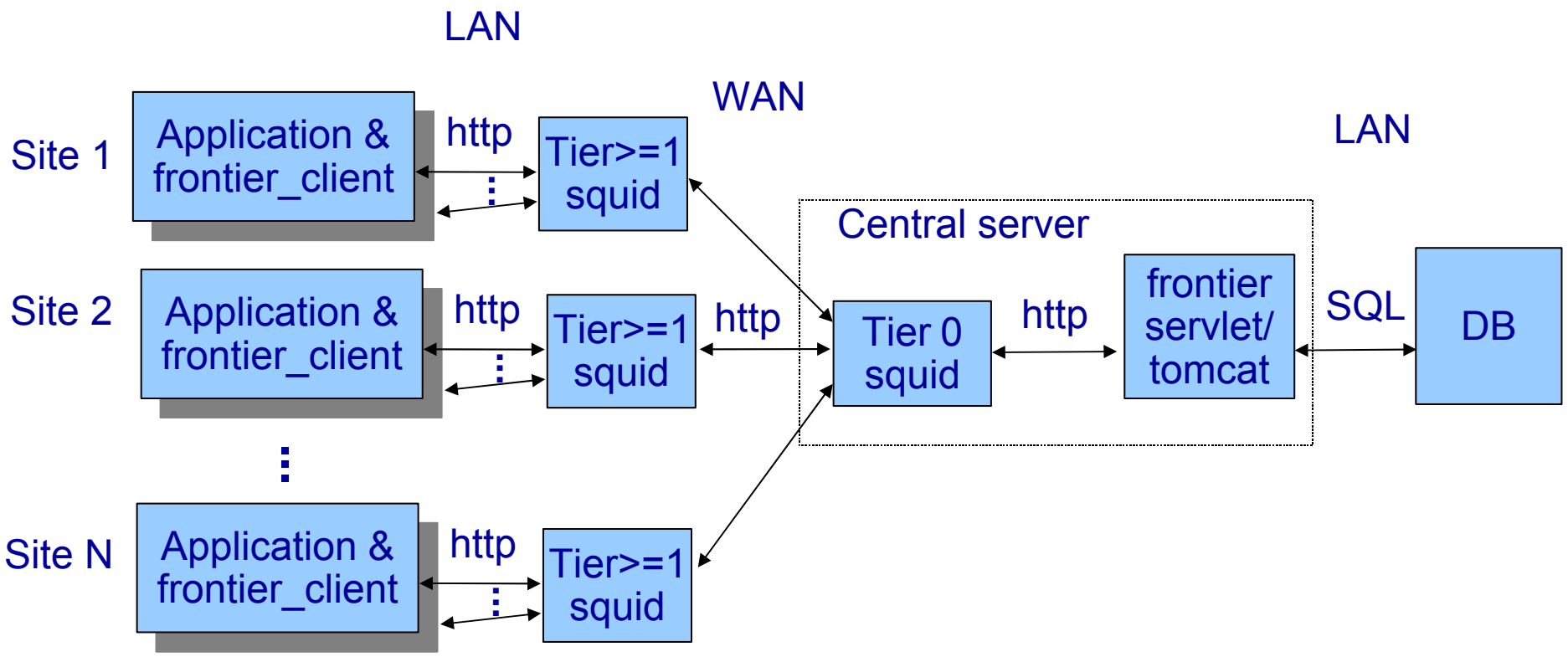
The problem

- Some applications need to get the same data to many jobs on a compute cluster
  - Inefficient or impractical to initially send with the job
  - Data too large or too many jobs for all to retrieve over a WAN directly from the source
  - Sometimes too much even for a single LAN source

The Frontier+Squid solution

- Distributes rarely-changing data from central databases to many clients around the world
  - Many sites
  - Many nodes/jobs at each site using same data
- Transfers with http to take advantage of standard tools
- Uses squid to cache centrally and locally at each site
  - also add-on monitoring tools MRTG and awstats
- Name comes from “N-tier”
- Developed for CDF at Fermilab
- Being adapted for CMS at CERN (by Fermilab)

Frontier architecture



Frontier/Squid performance

- Not designed for fast cache loading – 7MB/s or less to load from Oracle DB through tomcat servlet
- Can compress data: slows cache loading, speeds network
- Adds 1/3 net overhead for hex encoding to fit into http/xml
- Performs well for multiple clients once data is locally cached, especially for objects a couple megabytes or larger
  - 430MB/s total throughput for 2 squid servers each with 2 bonded gigabit interfaces and 2 squids
- However, that may not be enough for many jobs or large datasets that are loaded at nearly the same time
  - 100MB * 1000 jobs / 430MB/s = ~4 minutes
  - 10GB * 100 jobs / 430MB/s = ~40 minutes

Starting many jobs at once - problem

- CMS has an “Online” application with tight requirements:
  - 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 at once - solution

- Solution for CMS Online: squid on every node
  - Configured to pre-load simultaneously in tiers
  - Each squid feeding 4 means 6 tiers for 2000 nodes
    - 50 racks reached in 3 tiers, 3 tiers inside each rack
  - Measurements on test cluster indicate requirements can be met
    - bottleneck becomes the conversion from DB to http
    - 10-second loading always reads from pre-filled local squid

Grid environment

- Much less controlled than CMS Online, but still may need to load same data to many jobs at nearly the same time
  - One non-Frontier case needs 14GB of data, nearly 1GB at a time during a long-running job, hundreds of jobs
    - Must have at least one local cache
    - If all load from same cache, 1GB * 100 nodes / 100MB/s (1 Gbit/s) would be 16+ minutes
- Many different sites administered by many different people
  - Needs to be easy to configure

Grid environment proposal

- It's easy to make data available on an http server
- Proposal:
  - Primary (& possibly secondary) squid at each grid site
  - squid on every grid worker node
    - configured to find primary (& secondary) for site
    - automatic discovery of node peers because static configuration is impractical

Automatic discovery of squid peers

- squid-users suggested using existing multicast feature to locate objects already cached in peers
  - doesn't scale to hundreds or thousands of nodes
- Proposal: modify multicast peer discovery as follows
  - only peers that have objects respond
    - only when not heavily loaded
  - keep track of fastest responders and use unicast queries most of the time
  - also keep track of fastest throughput and give them priority to make best use of asymmetrical network
  - use existing TTL limit feature on multicast

Further info

- **Frontier home page:** http://frontier.cern.ch/
- **Performance details on CMS Twiki**
  https://twiki.cern.ch/twiki/bin/view/CMS/FrontierPerformanceImprovements