Any Data
Any Time
Any Where

Sudhir Malik

University of Nebraska-Lincoln & Fermilab, U.S.A.

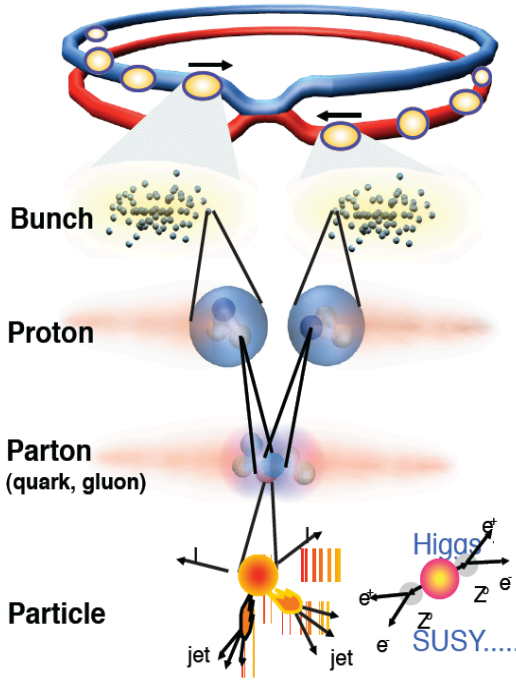
On behalf of the CMS collaboration

ICHEP 2012, Melbourne, Australia – 4-11 July 2012

The CMS Experiment

- **CMS** – **C**ompact **M**uon **S**olenoid detector at
- **LHC** – **L**arge **H**adron **C**ollider accelerator at
- **CERN -** Organisation Européenne pour la Recherche Nucléaire

Highlights of CMS –
- 3.8 Tesla Magnetic Field
- 14000 tonnes
- Studies proton-proton and heavy ion collisions
- Search for Higgs Boson
- Extra dimensions
- Dark Matter
- Discover the Unexpected

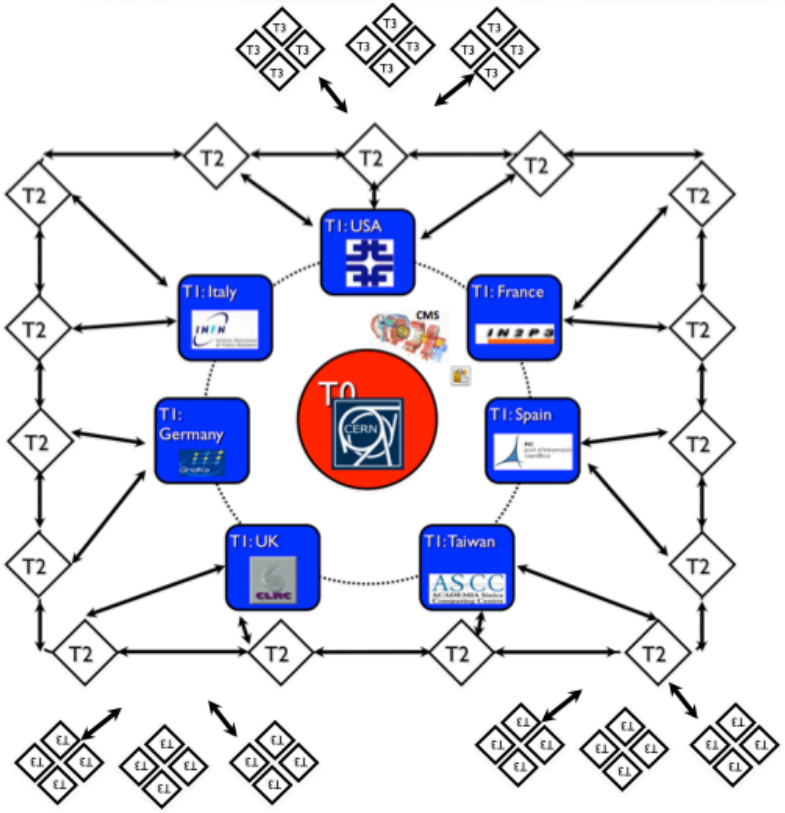


Proton- Proton	2835 bunch/beam
Protons/bunch	10^{11}
Beam energy	8 TeV (8×10^{12} eV)
Luminosity	$10^{34}\,cm^{-2}s^{-1}$
Crossing rate	40 MHz
Collision rate	~10^9 Hz
New physics rate	~ 0.00001 Hz

Event Selection:
1 in 10,000,000,000,000

Current computing model

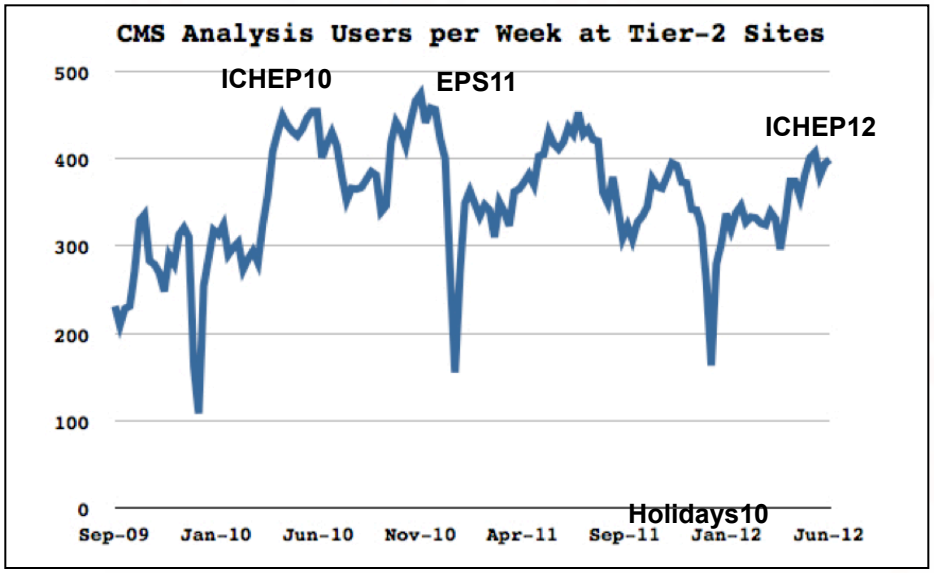
- The CMS computing model is highly distributed, both by design and by necessity
- Challenges in making this model work, but worth it

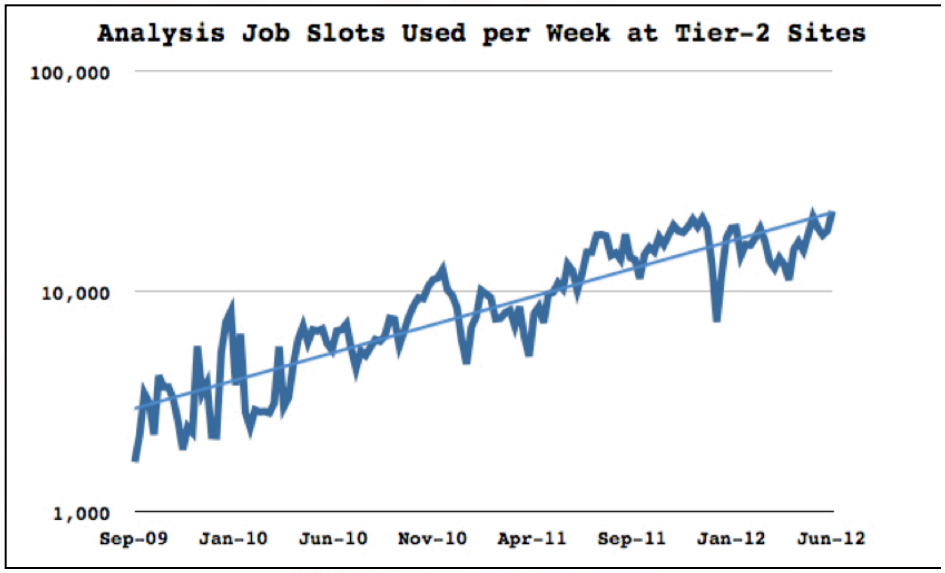


- T0 - Prompt Reconstruction, Calibration Streams Archival Copy of RAW and First RECO data
- T1 - Archival and Served Copy of RECO Re-Reconstruction, Skimming Archival Storage for Simulation
- T2 - Simulated event production and primary resource for analysis by physicists
  - Tier-2s have large enough disk pools
  - Users have access to Tier-2s disk and CPU
- T3 - Source of user analysis, user defined event content, copy reduced dataset to your favorite machine

Success of current model

- Each year 2010-2012 has brought increasing challenges for computing
- Increase in luminosity(6E33), pileup (16), event size (0.8 MB/event RECO format, 0.2 MB/event AOD), event processing time (quadruple), high trigger rate(~400 Hz)
- Computing successfully met all challenges successfully with the current available resources
- Average of ~400 users/week using grid resources





- Over 1M analysis jobs/week
- 33K datasets in DBS
- Resulting in over 120 physics publications

Limitations

- The Tier-2 facilities are the primary resource for all physics analysis
  - Data needs to be "hosted" here to be accessible to users that requires a human to make access request, and another one, the receiving site's data manager, to accept the request
  - This is where the various physics groups, and individual physicists have disk space allocated, produce group-level skims, develop and test new reconstruction algorithms, perform physics analysis
- Need to co-locate storage systems that host data with the processors that analyze them
  - Leads to inefficiencies in CPU usage
  - Physicists tend to insist on having the data they think they will need on disk, and given the size and distributed nature of the collaboration, making the distinction between what is expected to be needed and what is actually used is difficult
  - Some datasets not or rarely used occupy space that could be filled with more popular datasets and sites remain underutilized
  - Some sites are routinely saturated, provide more opportunistic resources than pledged or have larger queues of pending job
  - Restricts analysis to centers with resources and expertise to operate large storage infrastructure
  - If data gets deleted at a site chosen(has to be done statically) by a job, that job would fail

Tier-3s and beyond

- Current model renders Tier-3s not useful for physics data analysis
- Facilities at this tier have no operational responsibility towards CMS
- Driven entirely by the scientific goals of the institutions that own and operate them
- Great heterogeneity in size and functionality.
  - The facilities range from being just a few compute nodes to clusters on the scale of a Tier-2 facility
- The requirement of data co-location under-utilizes both the computational and intellectual resources of the Tier-3 universities.
- Physicists need improvements that will reduce the operational costs of storage, strengthen training and support, and ease access to the data
- We also want to expand the usage of resources beyond those controlled by physicists at Tier-3 to opportunistic resources that might be located on their campus, or anywhere else in the world
- To do so will require technologies that do not yet exist in CMS or in grid computing in general, including the ability to transparently add computing resources in grids and clouds
- LHC datasets are growing rapidly and the computing model must be adjusted and optimized to make the best use of the limited resources

Strategy

- Wide-area networking is more reliable now
- One is inclined to forget co-location and to think big
- CMS has made much progress in optimizing reading of data files over the network, there is very little additional cost compared to reading a file in the same room
- We need to build software tools with infrastructure with following features
  - No requirement of co-locating storage and processing resources
  - Reliability
    - No I/O error or failure for end-user unless no site can service
    - Catching failures early and re-directing I/O to different site
  - Transparency
    - Underlying system actions automatic – catalog, lookups..
    - Same workflow to access data – "close by" versus half-way around the world
  - Usability
    - Any solution should seamlessly integrate with CMS application framework, must not degrade event process significantly and complement the currently deployed data management system in CMS
  - Efficiency
    - dynamic data access and placement, access data in Tier-2 caches from Tier-3
- greater access to data by reducing financial and operational burden
- use of "opportunistic grid" and "commercial cloud"
  - with collaboration of computer scientists, applied mathematicians, physicists
- **analyze any data, any time, any where (AAA)**

Solution

- To achieve these goals CMS has begun using a distributed architecture based upon the Xrootd protocol and software developed by SLAC
- Architecture similar to the current data management architecture of the ALICE experiment.
- No intention on replacing current CMS data access methods for production
- Will greatly reduce the difficulty of data access for physicists on the small or medium scale
- The global Xrootd infrastructure would provide
  - fallback option for grid jobs in case of overflow (deal with resource limitation)
    - data access for jobs running at the Tier-2s
    - provided that fallback is enabled at the destination and data source exists
    - forced fallback allows “backfill” of otherwise idle CMS analysis CPUs
    - work around non optimal data distribution
  - interactive access for CMS physicists
  - a disk-free data access system for Tier-3s

Xrootd Architecture

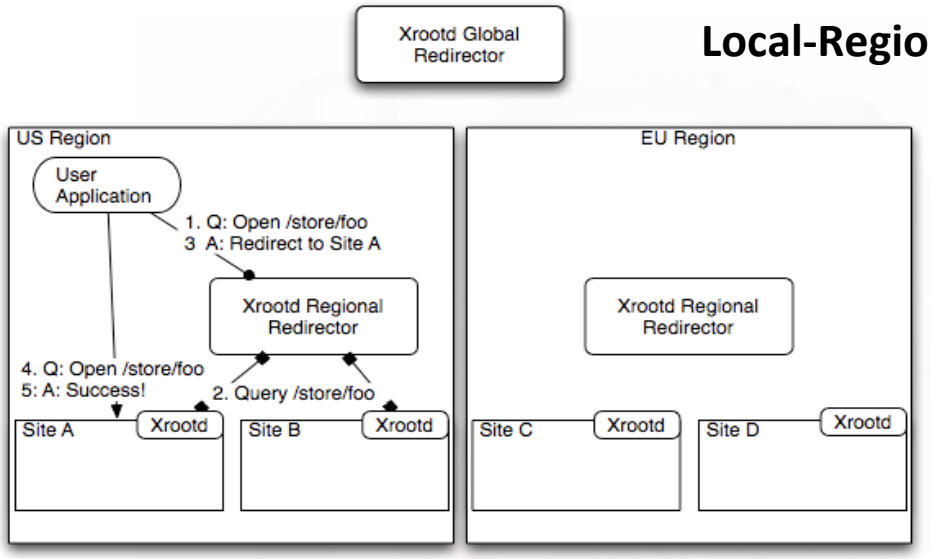
- Key element here is the direct remote access of files
- The basic feature of Xrootd - redirectors - allow jobs to find data a remote sites without any action from the user
- Mature project for remote-I/O.
- Almost always integrated into ROOT.
- Has the security mechanisms required by WLCG
- Time to open event interactively is limited to network latency
- Servers can be dynamically added or removed
- No restriction to using central sever
- Allows to decentralize architecture, provide high availability
- No “catalog synchronization” and “catalog re-building” issues
- Breaks “data-locality” dependency for jobs
- Transparent to user - exit only if no new data server can be found after a certain amount of time/retries

How does it work

- Scalla Software Suite – creates distributed data access infrastructure
- Uses Xrootd protocol – storage operations, custom protocol – load balance and redirect from global Xrootd to Xrootd data server
- Xrootd service – single Xrootd protocol endpoint, atleast one server at each Tier-2 site (like “google.com” – worldwide distributed infrastructure and is not a single server)
- Server at Tier-2 (controlled by Tier-2 admins) translates from LAN protocol to Xrootd protocol
- transparently translate CMS logical filename (LFN) to local physical filename (PFN), site specific namespace differences invisible, export uniform “global namespace”
- Users applications accesses central Xrootd endpoint
- Endpoint checks its info in cache to determine for file location , redirects client to the site
- If file location not known, queries all connected Scalla servers and redirects client to the first location found
- Client connects to the site-specific Xrootd server, authenticates, downloads data
- time between contacting central endpoint to data download is very fast - tens of milliseconds due to – compact binary protocol

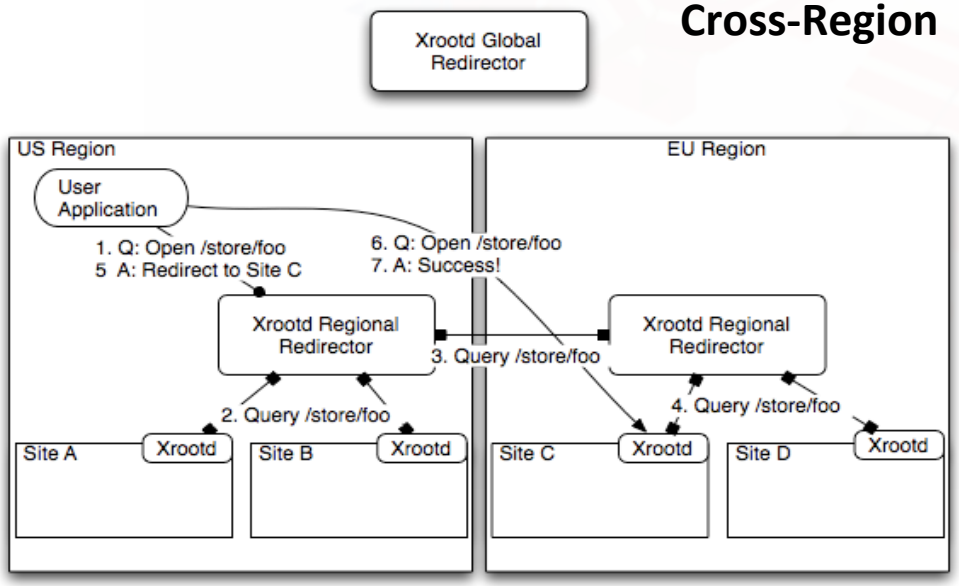
Redirectors

Local-Region



- First (1) the user application attempts to open the file in the regional redirector
- If the regional redirector does not know the file's location, it will then query all of the logged-in sites (2)
- Site A responds that it has the file, so the redirector redirects (3) the client to Site A's Xrootd server
- Finally, the client contacts Site A (4) and starts reading data (5)
- This is all implemented within the Xrootd client; no user interaction is necessary

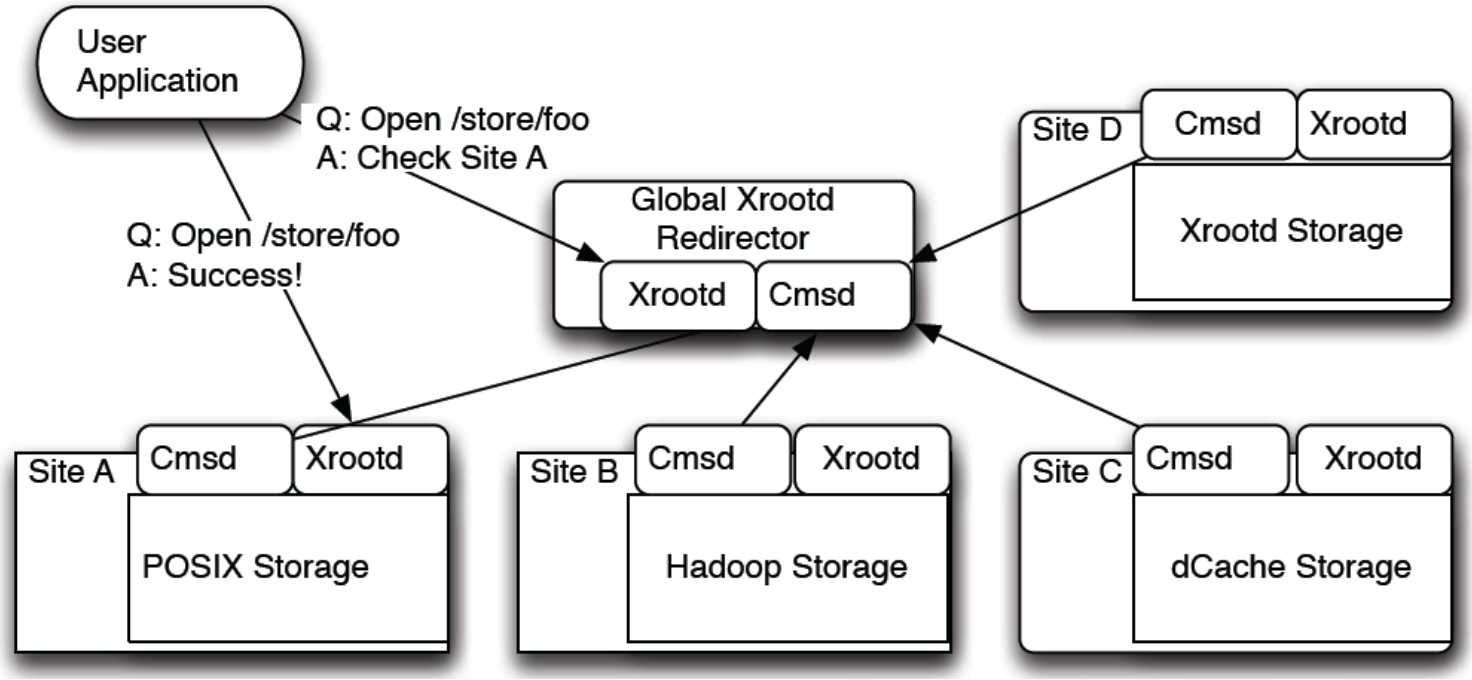
Cross-Region



- The image to the lower-left shows the communication paths for a user application querying the regional redirector when the desired file is not within the region
- Proceeds as in the case above , except all local sites respond they do not have the file
- Then, the regional redirector will contact the other regions (3); if the file location is not in cache
- The other regional redirector will query its sites (4)
- In this example, the user is redirected to Site C (5) and successfully opens the file (6 and 7).

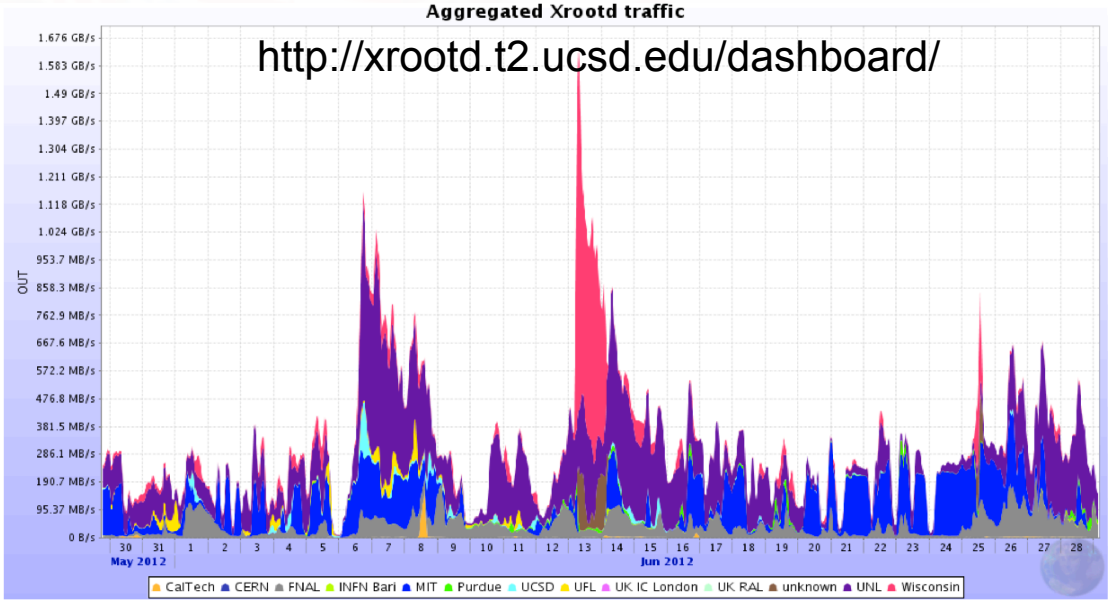
Federation

- Remote access gives us data for one site
- We need a federation to access all sites across all CMS sites



Operational Experience

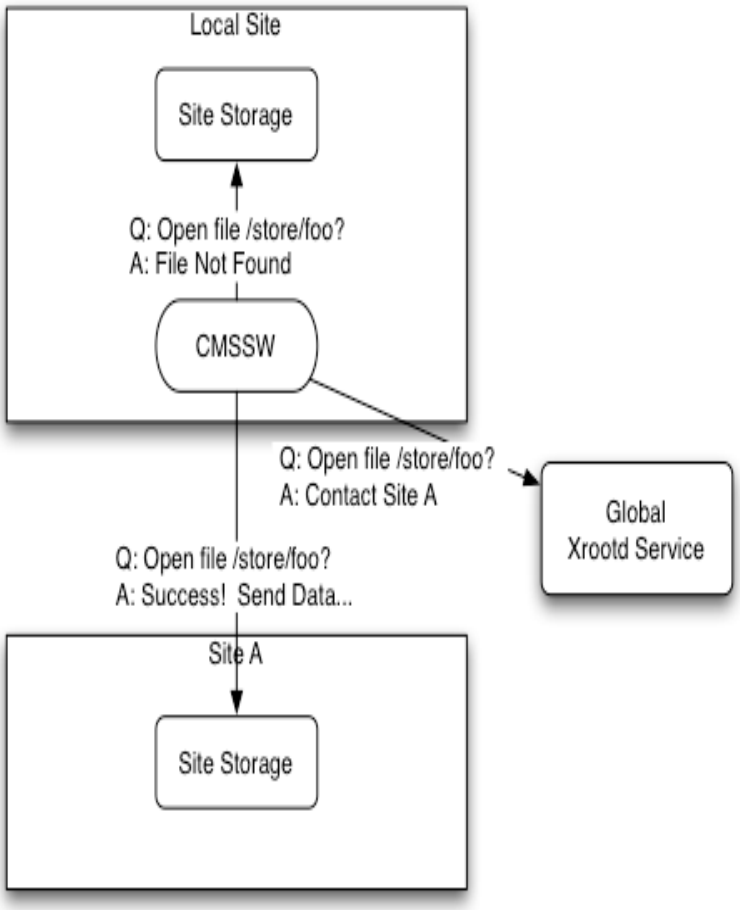
- A prototype of Xrootd architecture has been tested in the CMS sites in U.S.A.
- Included the FNAL Tier-1 (dCache) and 8 Tier2s (5 HDFS, 1 dCache, 1 Lustre, 1 L-Store).
- Evolving to include more CMS sites worldwide and all the relevant storage technologies
- Redirectors at UNL(Nebraska, U.S.A.), Bari (Italy) and CERN.
- During last April, monitoring recorded:
  - Over 300 unique users
  - 900K file transfers
  - 300TB moved
- Site Integration - sites integrate via installing a plugin specific to their storage system
- Merging Namespaces - To provide a uniform namespace, each site must export the global filename, not the local filename. This is achieved through a Xrootd plugin
  - For CMS, mapping achieved through a list of mapping rules and regular expressions



- Authorization - GSI-based authentication used, has a plugin for mapping the GSI credentials, AAA passes the DN and VOMS attributes to the site mapping service
- Monitoring – two streams - summary and detailed - http://xrootd.t2.ucsd.edu/dashboard/

Fallback Option

Besides interactive case, Xrootd can provide a fallback option for file access for grid jobs



- If grid job fails to open a file, don't fail the application
  - Try again reading from re-director
  - Loss of efficiency but job does not crash
- Storage Healing - federation as a source to re-download the broken file
- File Caching - Remote I/O expensive in terms of WAN bandwidth, especially if the file re-read many times
  - Instead of asking the redirector for the file, ask a local Xrootd install
  - On the file open failure, the local Xrootd will stage the file to a local disk and on next access, file will be local

Using additional resources

In case CMS wants to explore (not the case now) additional grid and opportunistic resources it would

- Require on-demand addition of CPU that should integrate seamlessly into its structure
  - But grid submission systems now used by CMS (Condor-G or gliteWMS) bind the user's job to a specific grid site, making it difficult to dynamically take advantage of opportunistic cores when available
  - However, we do have some experience with dynamically, late binding system
    - Univ. of Wisconsin Tier-2 uses Condor flocking mechanisms to export excess jobs to the large opportunistic resources available throughout the UW campus
    - CMS Analysis Operations group at UCSD operates a Condor glidein-based analysis system dynamically gathering compute resources from CMS Tier-2s worldwide to be used for data analysis jobs through the CMS CRAB infrastructure
    - Both not optimal but one can extract best of the two to find solution for on-demand CPU
- Also need to provide access to CMS software releases, solve the data access problem, and make the data access solution scalable, or at least regulate its scale
  - Having CMS software releases pre-installed on all sites is fundamentally not an option for opportunistic resources
- Need to efficiently use the combination of WAN bandwidth and access to network and storage resources
  - Jobs preferentially scheduled at CPU resources that have data already cached nearby, if any such cache exists

Conclusions

- Current CMS computing model is a great success and currently able to meet the demand of its users
- Challenge in near future is perhaps not in expanding the total resources available, but making sure that available resources are being used optimally
- Xrootd represents such a direction and CMS is quickly working on imbibing its potential
- After successfully performing prototype test at CMS sites in the U.S.A., CMS is expanding the model to all its sites
- It is also working on meeting the challenge of optimally using the opportunistic CPU and storage resources
- The success of CMS in adapting its computing model to improve resource use suggests that these efforts will be successful in the future too
- The success of this adaption will truly lead to access of any data, any time and any where for CMS users
- Thinking beyond CMS and HEP, this could serve as a prototype for the other "big data" applications outside of HEP

THANKS