Federated Storage Workshop

https://indico.fnal.gov/conferenceDisplay.py?confId=7207

Summary
For pre-GDB (Data Access) Meeting 5/13/14

Andrew Hanushevsky
SLAC National Accelerator Laboratory

April 10-11, 2014
SLAC

Status

- ALICE (Alien)
  - Long history of using federated storage
    - Catalog driven read-write exclusively XRootD federation
      - Though other protocols supported for various types of files
    - Heavy production usage
      - Past 6 months: 12PB written (0.85GB/s) & 126PB read back (8.4GB/s)
    - Small reads cause a large WAN I/O performance hit.
      - Local storage avg: 1.44 MB/s & Remote storage avg: 0.54 MB/s
      - Average CPU efficiency: 55%
  - Waiting for XRootD R4 with planned migration

Status

- ATLAS (FAX)
  - Rocky start due to LFC bottleneck in Name2Name translation
    - Now, sites use Rucio name convention so very stable
      - 47 sites deployed
    - Hammer Cloud testing is an ongoing activity
      - Architectural issues as SLAC & SWT will be fixed for improvement
  - PANDA now enabled for FAX failover
    - Saves a significant number of jobs that would otherwise fail
  - Continuous monitoring and improvement whenever possible
    - This is essentially production now

Status

- CMS (AAA)
  - Took a lot of work, especially to improve WAN I/O
    - Required for a workable large-scale federation
  - Today over 80% of T1/T2 sites are federated
    - Those that are not are poorly performing sites, so OK
    - Federated access at the same I/O level as bulk file transfer
  - Used for fallback and small-scale opportunistic scheduling
    - Read-only now looking to see how to do read-write

Status

- COEPP (Australia)
  - Initially read-only federation
  - Problems:
    - WAN transfer rates problematic when TTree cache is turned off
    - Caching is problematic using FRM facility
    - Federated write access
- UK
  - Impact on infrastructure is an open topic
    - Need to research and understand the access patterns
  - Doing stress tests via hammer cloud
    - 100-200 jobs work fine on well connected sites

Monitoring

- Analysis
  - Predict impact of federated access on EOS
    - Understand relationship between I/O and CPU
    - Propose events/sec as efficiency measure
      - Does it matter as users don't seem to care as long as it runs
        - E.G. not turning on vector reads, Ttree-cache, etc
    - Impact of fail over on CERN is minimal (<1%)
      - More of a CPU issue as more people take up slots

Monitoring

- Developing a cost matrix is essential
  - Cost of data transfer from any endpoint pair
  - Essential for scheduler to use this to not over-commit a site
    - May require some kind of throttling anyway
- Monitoring data rapidly increasing
  - 1 TB of monitoring data collected so far
  - The increasing rate and variety is causing scaling issues
  - Provides wealth of information so new uses being developed
    - File popularity, space resource usage, etc

WAN Access

- Effective access requires retooling the application
  - Latency ranges from 30 to 300 ms
- Caching is a must but roots's TTreeCache is not always sufficient
  - During training (first 20) every object is read with a separate network access
    - A typical CMS 1000 branch file -> 20 minutes training time
      - At 130 ms latency WAN file access takes too much time
  - Solution is bulk loading then training, reduces startup time
    - Bulk training yields an order of magnitude improvement
      - Changes will be integrated into the core by the root team

Access Patterns

- Crucial to understand
  - Develop new computing models and caching proxies
    - Can be used as a simulation for a caching proxy
  - Can be used to improve CPU efficiency
- Data access in user areas is least predictable
- Needs to periodically redone because it changes
  - Sensitive to data formats

Scale Testing

- Federated file open rate
  - Generally, will scale to CMS needs
    - File open needs to be 100HZ but tested to 200HZ
  - However, very sensitive to SE type
    - Investigation on the way as to why such high variability
- Read rate
  - US sites do well
    - However large variability even with similar SE's
      - This may be due to local load

Workload Management

- Panda
  - Finding a spot for federated storage in an exabyte environment
    - Rescuing failed transfers
      - Done and working well
    - Mitigating bottlenecks in job queues near local data
      - Opportunistic scheduling (alpha testing successful)
        - Significantly reduced wait time
    - Actively using federated storage for transfers
      - May need to throttle transfer rate
        - Already doing this by limiting the number of jobs
    - Many other possibilities

Recent Development

- HTTP plug-in for XRootD
  - Additional access mode (HTTP/HTTPS/WebDav)
- Full plugin environment in XRootD client
  - Additional application-specific processing modes
- Caching XRootD Proxy Server
  - Reduce network latency
  - Automatically manage site storage
  - Repairing local storage
- HTTP based federations
  - Another way of accessing federated storage

Next Meeting

- Target is January 2015
  - UK or Rome