Issues of remote stage-out protocol

Ian Fisk

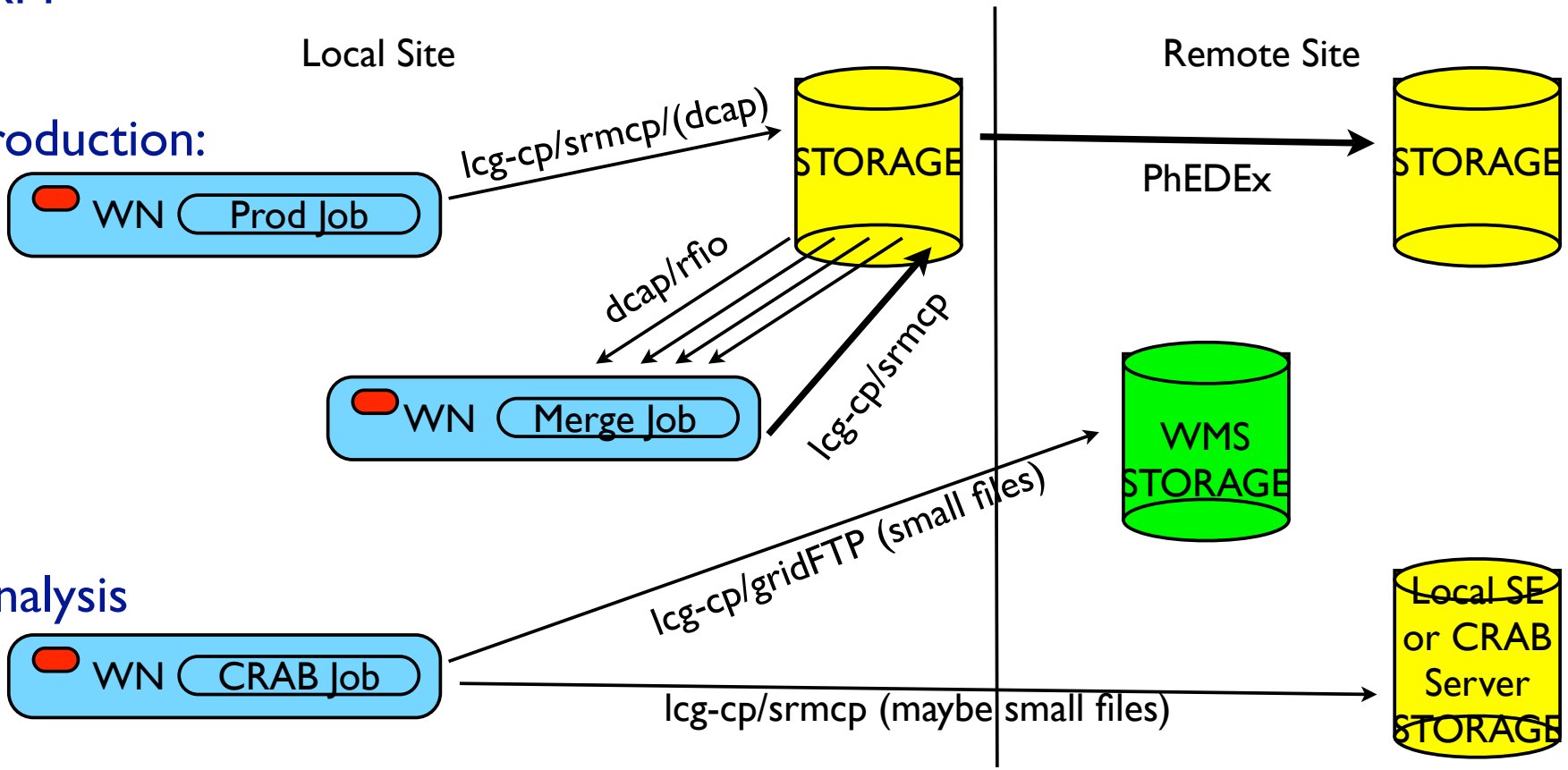
February 5, 2009

The CMS Model

The output sandbox for the WMS is too small for much of our work

- 50-100MB of output is easy to fill
- It's easy to fill the WMS temporary storage

CMS has used for some time a model where files are staged back through SRM



Issues in SRM

Each SRM implementation has a different number of transactions it can support before problems occur

- A new dCache/SRM installation can handle about 5Hz
  - The previous version could handle 1-2
- Castor is reported to scale to 50-100Hz
  - I have not measured this
- I don't have reliable numbers from BestMan and STORM

SRM transactions include

- Production transfers with PhEDEx and any activity from staging files
- A check the directory exists, a transfer and a verification of success are 3 transactions
  - Issues of the reliability of the SRM exit code have made this worse by adding the need for a separate step to check success.

Even if the server doesn't fail and simply slows down, we make very inefficient use of the CPU as workers wait to stage out and retry

Problems for production

At FNAL we ran into issues when the farm got to ~3000 cores

- Some shorter production workflows finished in ~1 hour
- Roughly 1Hz plus regular traffic
  - Made worse by checks of success
  - Made worse by skimming workflows with more than 1 output module
- Skimming workflows are quick and should have as many output modules as possible to reduce the job of data ops
  - Job length and number of outputs has been generally limited by the memory needed in the framework
    - May be improved in CMSSW_3

Currently FNAL is about 5500 cores

- Multiple checks and multiple outputs exceed the capability for SRM even at 5Hz for all but long running jobs
  - Some of the transaction rate is taken by the wide area transfers

Solution at FNAL

Use a local transfer protocol (dcap) to stage the high volume of unmerged files into the storage system

Advantages

- Lighter weight
  - No grid authentication (job itself has already been authenticated
  - Higher achievable transaction rate
  - More reliable exit code

Disadvantages

- SRM is a common protocol at all grid sites, the incoming job now needs to know the local protocol to use. Development work for the ProdAgent
- Involved some setup on the local site to enable a special door for writing into dCache with dcap

Issues for other sites

FNAL is one of the largest clusters available to CMS, but this will impact other sites eventually

- For skimming I think we need to assume we will get to 10 output modules
- 2 transactions per output file (copy and verify)
- Estimate for the CTDR is 0.25kSI2k s for skims (8Hz on a 2kSI2k core)
  - 2 hour jobs would process > 50k events
- This would be 20 transactions per core every 2 hours.
  - 3Hz for 1000 node cluster plus the normal traffic
    - Worse if it's 1 hour jobs, worse if it's 3 transactions per output file

Toss up whether we need to switch to a local stage-out protocol for all Tier-1 sites. May be needed for some

- Tier-2s run primarily simulation which runs longer. No obvious need for local protocol stage out for production

Issues for Analysis

In the proposed /store/user space the rate of the SRM transactions doesn't scale as the size of the site, it scales as the activity level of the supported users

- Users who get many cores on remote systems will be staging data back
- Files can be small and the number of transactions is related to how finely a user has split the workflow
- We have reasonable numbers during the spring 09 activities about the number of users running on a site, but not much about the number of users staging data back to a site.
  - Nominal Tier-2 supports 40 people. If 25% are active, that's 10 per T2. CTDR predicts 100-200k jobs per day. 2000 people in the collaboration, 25% active. 10 people per T2 out of 500 active people total would be 4000 jobs per day coming back to a Tier-2 (400 jobs per user per day)
    - No problem, but it's not what we see
    - Users submit lots of small jobs

Ways forward

For analysis the primary issue is that we have a model where the SRM endpoint needs to scale with how users split jobs

- In the current environment of many resources users are encouraged to split because the task gets done faster
- If we were more constrained you would want to hold the CPU you got for longer

I see two alternatives

- Try to encourage more reasonable length jobs and subsequently lower stage-out load
  - May require some some development and work on policy and education and may not work
- Implement a solution for local protocol stage-out and FTS driven copies after the job completes
  - Requires a fair bit of development. In order to significantly improve the number of transfers we would need a merging step too.