Service Challenge 3
CMS Goals

Lassi A. Tuura

Northeastern University

CMS Service Challenge Goals

- An ***integration test*** for next production system
- Main output for SC3 ***data transfer and data serving infrastructure*** known to work for ***realistic use***
  - Including testing the workload management components: the resource broker and computing elements
  - Bulk data processing mode of operation
- Crucial step toward SC4, CMS DC06 and LHC
  - Failure of any major component at CERN or at a Tier-1 site would make it difficult to recover and still be on track with increased scale and complexity in SC4 and CMS DC06
  - Focus on alternatives with reasonable expectation of success: need to leave SC3 with functional system with room to scale

CMS Service Challenge Goals Explained

- An integration test for next production system
  - **Full experiment software stack** – not a middleware test
    - "Stack" = s/w required by transfers, data serving, processing jobs
  - **Checklist on readiness for integration test**
    - *Complexity and functionality tests already carried out,* no glaring bugs
    - Ready for system test with other systems, throughput objectives
    - (Integration test cycles of ~three months – two during SC3)
  - **Becomes next production service** if/when tests pass
- Main output: data transfer and data serving infrastructure known to work for realistic use cases
  - **Using realistic** storage systems, files, transfer tools, …
    - *Prefer you to use standard services* (SRM, …), but given a choice between system not reasonably ready for 24/7 deployment and reliable more basic system, *CMS prefers success with the old system to failure with the new one*
  - Due to limited CMS resources, please confirm and coordinate with us your infrastructure so we can reach the objectives without excessive risk

Some Observations

- Give yourself enough time to put services into production
  - Our experience is that it takes months to bring a site up
  - Reserve enough time (read: months) to debug completely new systems before expecting great results
- You are expected to support what you put into production
  - Don’t plan for heroic one-time effort for throughput phase, you will kill yourself in the service phase
- Choose a services suite that is ready for integration test
  - CMS needs at least a month after large-scale functionality milestone for deployment into the experiment integration test
  - For throughput test, everything fully debugged by end of June
  - Decision to pick fallbacks latest by mid-June (this workshop?)
- Seek to “Evaluate what works, not find out what doesn’t”

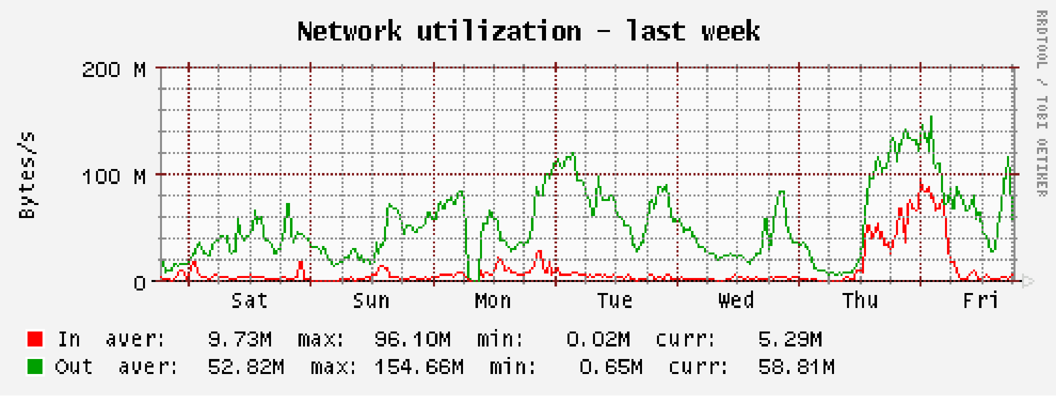
Input Parametres (I)

- CMS DC04
  - Tier 0 to Tier 1 sites
    - Rate 25 Hz = run completed every ~40 sec
    - Output ~250 MB/run (19 files) =~ 6 MB/s, ~0.5 TB/day

- CMS Computing TDR
  - Nominal Tier 1 (peak rates to/from tape)
    - From storage 800 MB/s
    - WAN 5.7 Gb/s in, 3.5 Gb/s out to regional centres
    - Peak data in 1.8 Gb/s (FEVT+AOD 0.7, AOD re-reco 1.0, MC 0.1)
    - Peak data out 0.9 Gb/s (serving events to Tier-2s)
  - Nominal Tier 2
    - From storage 1 GB/s (32 Mb/s per KSI2K)
    - WAN 1 Gb/s
    - Peak data in 5 TB/day
    - Peak data out 1 TB/day (up to 8 TB/day)
  - Estimate factor of five from now to C-TDR values

Input Parametres (II)

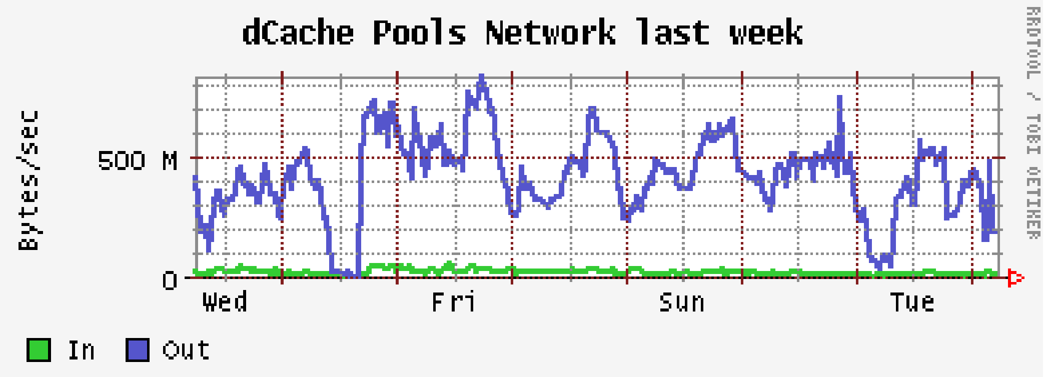
- Anecdotal statistics: data recently served from production storage systems at CERN, FNAL
  - Caveat: this is from system network usage monitoring, we don't actually know how much was delivered into applications



🟥 In aver: 9.73M max: 96.10M min: 0.02M curr: 5.29M
🟩 Out aver: 52.82M max: 154.66M min: 0.65M curr: 58.81M

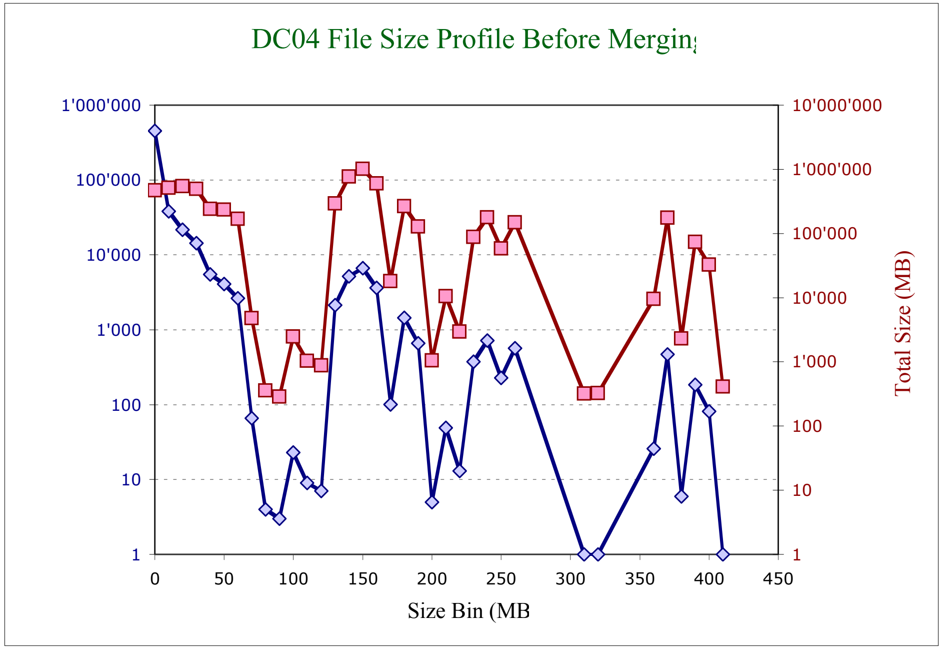
CERN stagecms last week, # jobs unknown

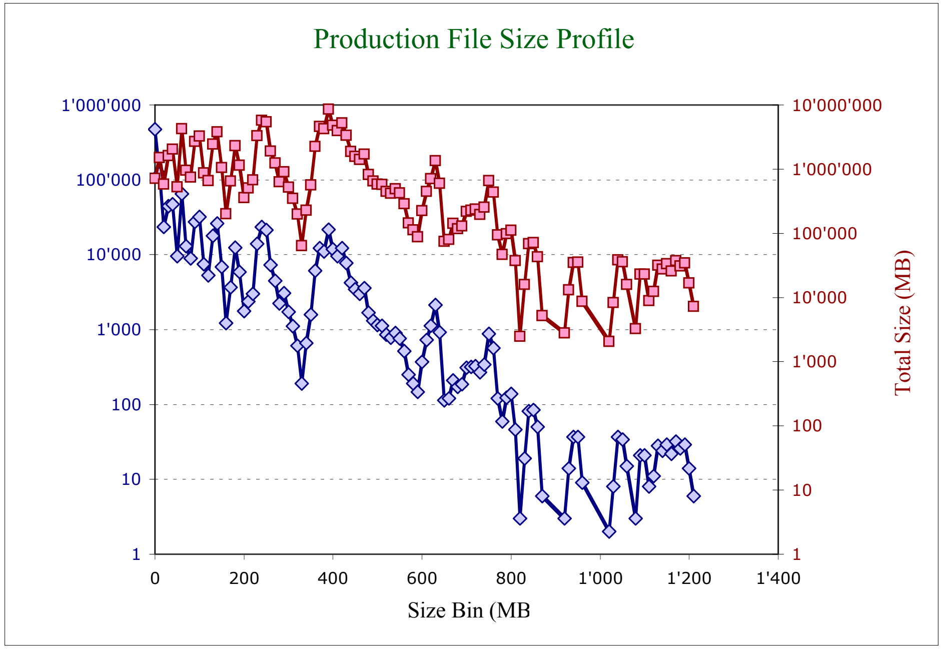
FNAL CMS pool last week, 200-300 jobs?



Input Parametres (III)

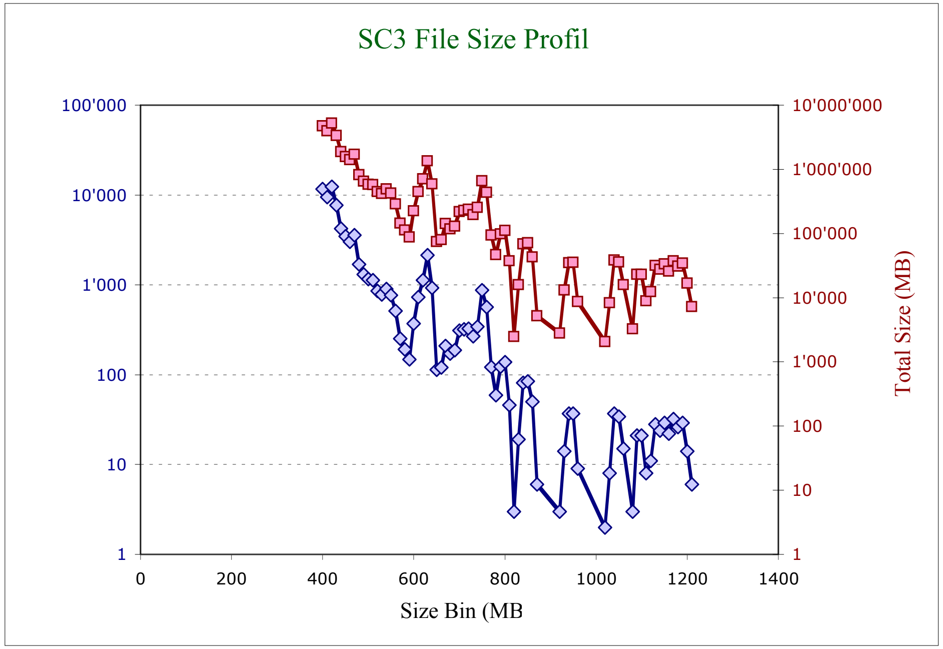
- File size distribution
  - DC04 files
  - Current production files

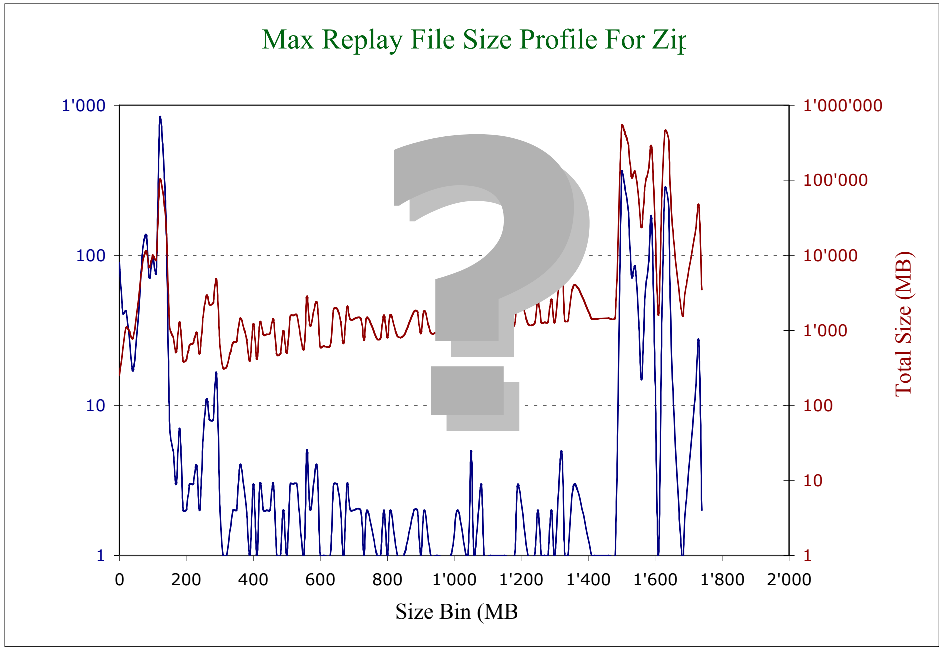

DC04 File Size Profile Before Merging


Production File Size Profile

Input Parametres (IV)

- File size distribution
  - Files for SC3 throughput phase (selected >= 400 MB)
  - Files for SC3 service phase (from merging)





Qualitative Goals

Throughput Phase

- Overview of throughput exercise
  - Throughput to disk and tape at Tier-1s from CERN Tier-0 disk
  - Fan out transfers to selected Tier-2s, same data but less of it
  - Target: transfer and storage systems work and are tuned
    - Using real CMS files and production systems (or to-be production)
    - Sustained operation at required throughput without significant operational interference / maintenance
- Concretely
  - Part 1: Data from disk buffer at CERN first to Tier-1/2 disks
    - Tier-2s will be subscribed subset of the data going to Tier-1s
    - Data to Tier-2s are routed via Tier-1s
  - Part 2: Same, but data goes to tape at Tier-1s
  - Transfers managed by PhEDEx
  - Files registered to local file catalogue
  - Sufficient monitoring

Quantitative Goals

Throughput Phase

- Rates defined in Jamie's document

Tier 0 disk to Tier 1 disk	150 MB/s sustained
Tier 0 disk to Tier 1 tape	60 MB/s sustained
Tier 1 disk/tape to Tier 2 disk	? MB/s sustained
Tier 2 disk to Tier 1 disk (tape?)	<1 MB/s (!?) sustained
Suggest informally 30 MB/s T1 to T2 if bandwidth is available	

- In addition: service quality

Transfer failures should have no significant impact on rate	
Transfer failures	<0.1% of files more than 5
Catalogue failures after transfer	<0.1% of files
File migration to tapes	(keep up with transfers)

Qualitative Goals

Service Phase

- Overview of service exercise
  - Structured data flow executing CMS computing model
  - Simultaneous data import, export and analysis
  - Job throughput at Tier 2 sites
- Concretely
  - Data produced centrally and distributed to Tier 1 centres (MSS)
  - Strip jobs at Tier 1 produce analysis datasets ("fake" COBRA jobs)
    - Approximately 1/10th of original data, also stored in MSS
  - Analysis datasets shipped to Tier 2 sites, published locally
    - May involve access from MSS at Tier 1
  - Tier 2 sites produce MC data, ship to Tier 1 MSS ("fake" COBRA jobs)
    - May not be the local Tier 1
  - Transfers between Tier 1 sites
    - Analysis datasets, 2nd replica of raw for failover simulation
  - Implied: software installation, job submission, harvesting, monitoring, VO + group roles

Quantitative Goals: Tier 1

Service Phase

- For two periods of at least one week each, sustain

Same service quality goals as with throughput phase	
All transfers and data serving are to/from tape at Tier 1s	
Data served to worker node jobs: bytes read by instrumented CMS apps (ROOT), not dcap/rfio/… (excludes file transfers!)	200 MB/s
Data stored from worker node jobs	12 MB/s
Transfers from Tier 0	3 TB/day (~36 MB/s)
Transfers to Tier 2s (all if more than one)	1.5 TB/day (~18 MB/s)
Transfers to Tier 2s (each)	1 TB/day (~12 MB/s)
Transfers to Tier 2s (each, minimum)	>10 MB/s [24+ hours]
Transfers to Tier 2s (each, if bandwidth exists)	30 MB/s [24+ hours]
Transfers from Tier 2s (each)	2.5 MB/s
Time from Tier 0 file availability to available for analysis applications at Tier 1	10% <15 min 33% <30 min
Skim data to 1/10th and store to tape	(keep up with input)
Job success rate	>95%? (to be defined)
Job throughput	?/day (to be defined)

Quantitative Goals: Tier 2

Service Phase

- For two periods of at least one week each, sustain

Same service quality goals as with throughput phase	
Data served to worker node jobs: bytes read by instrumented CMS apps (ROOT), not dcap/rfio/… (excludes file transfers!)	100 MB/s
Data stored from worker node jobs	2.5 MB/s
Transfers from Tier 1	1 TB/day (~12 MB/s)
Transfers to Tier 1	0.2 TB/day (~2.5 MB/s)
Time from Tier 1 file availability to available for analysis applications at Tier 2	10% <15 min 33% <30 min
Job success rate	>95%? (to be defined)
Job throughput	?/day (to be defined)

Quantitative Goals: Other

Service Phase

- Various constraints
  - Tier 1 strip jobs to keep up with incoming data
  - Tier 1 tape system able to migrate files at incoming rate (T0 + T2s)
  - Tier 1 data export able to keep up with data-producing jobs
  - Tier 2 data export able to keep up with data-producing jobs
- Other components

Resource broker able to accept jobs	N secs (to be defined)
RB and CEs/WNs able to process jobs	N/day (to be defined)
Grid infrastructure-related job failure rate	<5% (to be defined)

- Still undefined (or monitored) quantities
  - Latency from data block request to delivery
  - Number of data requests processed by Tier 1
  - File delay from request to start of transfer for MC and hosted data
  - Time for file to sit in Tier-2 cache
  - Frequency of Tier-2 cache refresh

Checklist Goals

Service Phase

- Automatic installation of CMS software works
- PhEDEx available, all file transfers executed with PhEDEx
- PubDB available, automatically updated from PhEDEx, updates RefDB
- Harvesting of job output files works: injected to PhEDEx, transferred
- File catalogue operational
  - Automatically updated by file transfers, harvesting
  - Functional for all jobs running on worker node
- UI installed with access to CMS software, test data samples accessible
  - Can compile, test, debug and submit CMS jobs to all sites from UI
  - Can receive jobs from all other CMS sites
  - "All sites" = "All CMS sites participating in the challenge"
  - "Submit" = "Submit using CRAB", "Run" = "As submitted fro CRAB"
- Worker nodes have access to CMS environment
  - Software, site configuration scripts, file catalogue, harvest agents, …
- General monitoring sufficient (to be defined)
- Optional: BOSS job monitoring provided (UI, database) and works

Test Data

Service Phase

- Total data capacity

50 TB	from CERN to at least two Tier 1 sites
~10 TB	from CERN to other Tier 1 sites
~5 TB	to each Tier 2
5-10 TB	T1/T1 analysis dataset transfers
50 TB	T1/T1 2nd raw replica transfers (for simulating Tier 1 failover)

- Data from both throughput and service phase can be discarded after a while
  - Data for service phase may need to be kept for a while (month)
  - Data for throughput phase can be recycled after a day or so
- *Most likely* no need for CPU capacity *dedicated to the service phase*
  - Submitting jobs to normal worker nodes, expect access to SC storage
  - Reasonable capacity available for two or three periods of a week at a time
- When integration tests have passed, services can go into production
  - Resources expected to remain for testbed environment

SC3 Services In Test

Services for all sites (I)

- Data storage
  - dCache, Castor or other (xrootd, gpfs, …)
  - SRM interface highly desirable, but not mandatory if unrealistic
- Data transfer
  - PhEDEx + normally SRM, can be + GridFTP – see Daniele's presentation
  - CMS will test FTS from November with other experiments (ATLAS, LHCb)
- File catalogue
  - The "safe" choice is POOL MySQL catalogue
  - Big question will catalogue scale for worker node jobs
    - Currently using XML catalogues from worker nodes
  - LCG favours LFC, but first step to CMS validation not even started
    - LFC exists, but no POOL version that can use it, and thus no CMS software
    - Existing CMS software to date will not be able to use LFC
  - US-CMS will test Globus RLS instead of LFC / MySQL on some sites
    - Same caveats as with LFC
  - Not planning to test EGEE Fireman yet
  - Note: in future possibly "trivial file catalogue" (= storage name space)

SC3 Services In Test

Services for all sites (II)

- Software packaging, installation, publishing into information system
  - Either central automated installation, or using local service
  - So far, central automated is not really very automated…
- Computing element and worker nodes
  - In particular, how the CE obtains jobs (RB, direct submission?)
  - Interoperability between different grid variants
- Job submission
  - Including head node / UI for submitting
  - Interoperability between different grid variants
- Job output harvesting
  - CMS agents, often configured with PhEDEx

- (These services require solutions for all grid variants)

SC3 Services In Test

Services for some sites

- PubDB / DLS
  - Backend MySQL database + web server interface for PubDB
- Job monitoring and logging
  - BOSS + MySQL database + local agents
- File merging
  - Agents running at the site producing data

- (These will evolve and be replaced with middleware improvements)

Support servers (I)

- Server-type systems required *at each site*
  - UI / head node for job submission (public login)
  - Storage space for CMS software installation (single root for all)
  - "Small databases" server for CMS services (see below, MySQL)
  - File catalogue database server (presumably MySQL on most sites)
  - Gateway-type server for PubDB, PhEDEx, job output harvesting
    - PubDB needs web server, PhEDEx local disk (~20 GB sufficient)
    - Typically installed as UI, but not public login (CMS admins only)
    - For SC3, one machine to run all agents is enough
    - For SC3, requires outbound access, plus access to local resources
      - PubDB requires inbound HTTP access, can install under any web server
    - The agents do not require substantial CPU power or network bandwidth, "typical" recent box with local disk and "typical" local network bandwidth should be enough (CERN gateway dual 2.4GHz PIV, 2 GB memory – plenty)

Support servers (II)

- Optional gateway services *at some sites*
  - BOSS job monitoring and logging
    - Local MySQL / SQLite backend per user on UI (MySQL can be shared)
    - Optional real-time monitoring database – to be discussed
    - BOSS itself does not require gateway server, only databases
  - File merging

- Service + operation of CMS services by CMS people at the site
  - Co-operation of local site admins and CMS people at the site
  - May have help from CMS people at your Tier 1, ask

Site Service Choices

Tier 0/1s

- CERN
  - Storage: Castor/SRM
  - Transfers: PhEDEx/SRM (srmcp)
  - File catalogue: POOL LFC Oracle
  - Does CERN participate as T1?
- FNAL
  - Storage: dCache/SRM
  - Transfers: PhEDEx/SRM (srmcp)
  - File catalogue: POOL Globus RLS
- CNAF
  - Storage: Castor/SRM
  - Transfer: PhEDEx/SRM (srmcp)
  - File catalogue: POOL LFC (Type?)
- RAL
  - Storage: dCache/SRM
  - Transfers: PhEDEx/SRM (srmcp)
  - File catalogue: POOL LFC (Type?)
- PIC
  - Storage: Castor/SRM
  - Transfers: PhEDEx/SRM (srmcp)
  - File catalogue: POOL LFC? (Type?)
- FZK
  - Storage: dCache/SRM
  - Transfers: PhEDEx/SRM (srmcp)
  - File catalogue: POOL LFC? (Type?)
- ASCC
  - Storage: Castor/SRM
  - Transfers: PhEDEx/SRM (srmcp)?
  - File catalogue: POOL LFC? (Type?)

Site Service Choices

Tier 2s

- US: Florida, Wisconsin, San Diego, Caltech (+ Purdue, Nebraska, MIT?)
  - Storage: dCache/SRM
  - Transfers: PhEDEx/SRM (srmcp)
  - File catalogue: POOL Globus RLS (POOL MySQL at some?)
- Italy: Legnaro
  - Storage: Castor?
  - Transfer: PhEDEx/Globus?
  - File catalogue: ?
- Spain: CIEMAT
  - Storage: Castor?
  - Transfer: PhEDEx/Globus?
  - File catalogue: ?
- UK: Imperial
  - Storage: dCache/SRM
  - Transfer: PhEDEx/SRM (srmcp)
  - File catalogue: POOL MySQL?
- Germany: DESY
  - Storage: dCache/SRM (+ tape)
  - Transfer: PhEDEx/SRM (srmcp)
  - File catalogue: ?
- Taiwan: ?

Typical Configuration

Service Suite

- One UI for job preparation etc.
  - Or "AFS UI"-like shared installation as available for CERN lxplus
- One CMS-dedicated UI-installed gateway system
  - ~20 GB local disk required
  - Runs PhEDEx, PubDB tools, output harvesting
  - Plus any other CMS-specific services (e.g. merging agent)
- One MySQL database server
  - Runs database for PubDB, BOSS
  - Runs database for file catalogue
  - Should not be the gateway server
  - In future, assumed to be CMS-dedicated, not required in SC3
- Web server
  - For PubDB, can be the gateway or another box

+ Accessible monitoring of all of this!

Typical Configuration
PhEDEx

- Single UI-installed system, ~20 GB local disk required
  - Follow deployment guide to install everything on local disk, avoid network file systems to avoid unnecessary agent crashes

```
Deployment/InstallOracleClient $BASE $TOOLS
Deployment/InstallPerlModules $TOOLS
Deployment/InstallPOOL -standalone -arch SLC3 $TOOLS
emacs Custom/MySiteName/Config        # follow guide
emacs Schema/DBParam                  # follow guide
Utilities/Master -config Custom/MySiteName/Config start
```

  - Load your certificate proxy to your local MyProxy server
    - See Custom/CERN/ProxyRenew `cron` script
  - Archive your transfer logs into some secure backed-up location
    - See Custom/CERN/LogArchive `cron` script
  - Watch the monitoring at http://cern.ch/cms-project-phedex
  - Watch the logs :-)

Summary

- Integration test for the next production service
  - Testing many new components ready for the step
  - Choose new components and fallbacks wisely
  - Many completely new systems rather a concern
  - When will CERN be tested as something more than a Tier-0 site?
- Aimed for data transfer and data serving infrastructure
  - CMS welcomes many new sites to join!
  - Opportunity for significant increase the infrastructure available for physicists in painless manner and readiness towards LHC startup!

Contact Information

- CMS main points of contact

Wiki	https://uimon.cern.ch/twiki/bin/view/CMS/SWIntegration
List	<cms-computing-sc@cern.ch>

- Overall service challenge coordination

Jamie Shiers	<jamie.shiers@cern.ch>
General	<service-challenge-tech@cern.ch>

- CMS computing coordination

Lothar Bauerdick	<bauerdick@fnal.gov>
David Stickland	<david.stickland@cern.ch>

- CMS overseers for challenge / integration

Ian Fisk	<ifisk@fnal.gov>
Stefano Belforte	<stefano.belforte@ts.infn.it>
Lassi A. Tuura	<lassi.tuura@cern.ch>