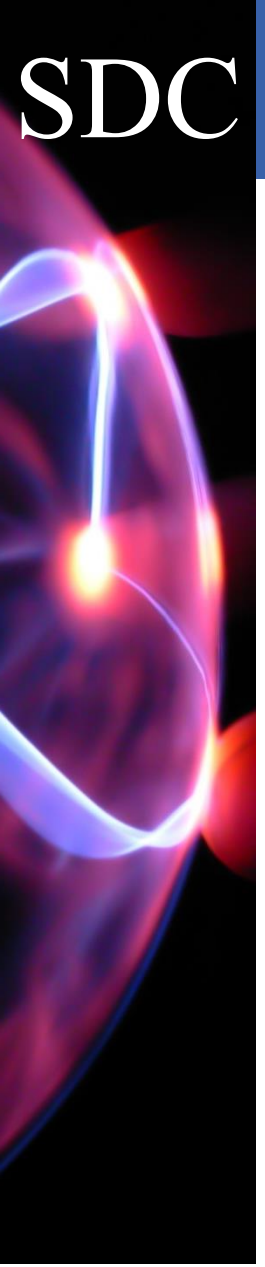


AGIS: Topology & SchedConfig parameters

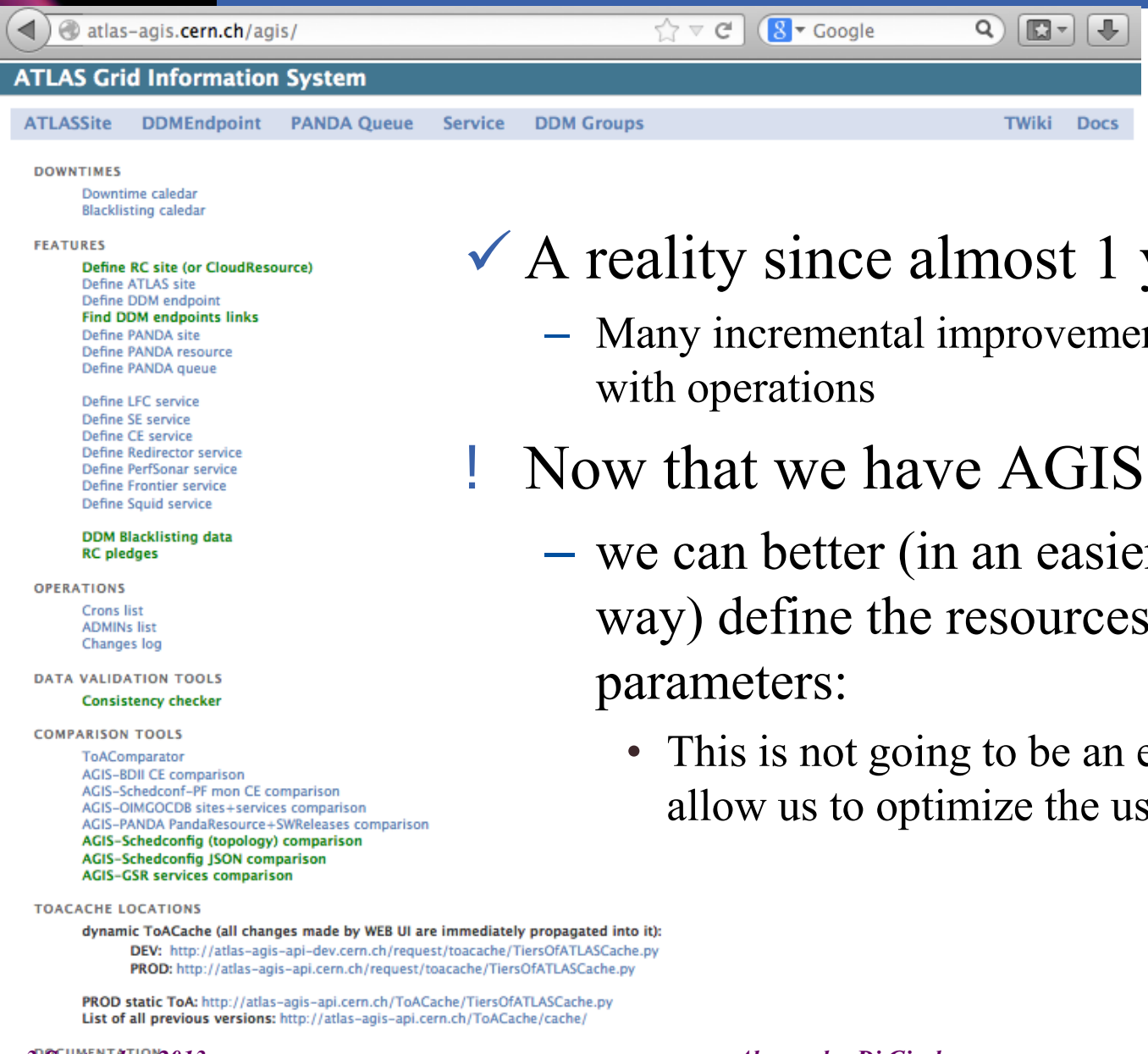
Production /Analysis share configuration

The HTTP ecosystem

Alessandro Di Girolamo CERN IT/SDC

+ slides/points from Simone Campana (Prod/Analy share), Fabrizio Furano (HTTP ecosystem), Maria Alandes (the GSR) and IUeda (SchedConfig params + pd2p observations)

ATLAS Grid Information System



- ✓ A reality since almost 1 year!
  - Many incremental improvements without interfering with operations
- ! Now that we have AGIS,
  - we can better (in an easier more maintainable way) define the resources and their parameters:
    - This is not going to be an easy process but it can allow us to optimize the use of all the resources

AGIS and the GSR

- AGIS collects info from various places:
  - BDII, OIM, GOCDB, plus AGIS webUI itself
  - And provides the abstraction layer needed by ATLAS to know what to use where.
- WLCG Information System Task Force:
  - create a single entry point for the all the static information of the resources

WLCG Global Service Registry

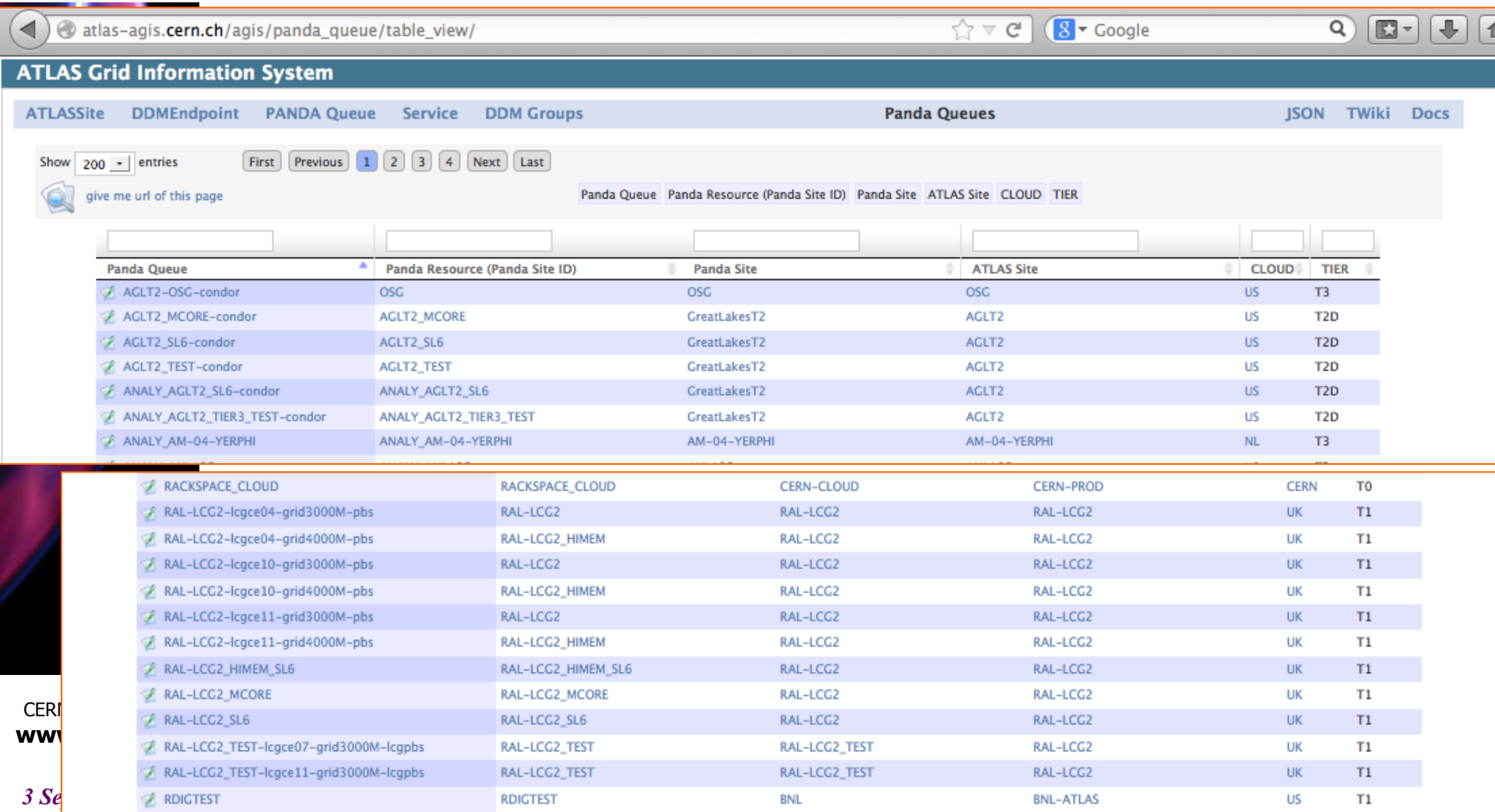
- Single entry point to get information about WLCG resources
  - Pledged resources
  - Actual resources
- Integrates resources from OSG and EGI
  - Independent from GLUE schema version
  - Although GLUE 2.0 is the internal integration model
- Easier for the experiments to gather the information they need in their databases
  - Saves them from dealing with different sources of information
  - Enables the possibility of changing the sources of information in a transparent way
- Single central repository
  - Single place where information is processed
  - Avoids duplication of effort
  - Reduces inconsistencies
  - Improves quality as the processing is done by experts
  - Easier to update and maintain

✓ AGIS setup a collector from GSR to validate the information

SDC Topology

CERN IT Department

- PandaQueue – PandaResources – PandaSite – ATLASSite
  - Do we really need all these? E.g. Who does use the PandaSite?



Panda Queue	Panda Resource (Panda Site ID)	Panda Site	ATLAS Site	CLOUD	TIER
AGLT2-OSG-condor	OSG	OSG	OSG	US	T3
AGLT2_MCORE-condor	AGLT2_MCORE	GreatLakesT2	AGLT2	US	T2D
AGLT2_SL6-condor	AGLT2_SL6	GreatLakesT2	AGLT2	US	T2D
AGLT2_TEST-condor	AGLT2_TEST	GreatLakesT2	AGLT2	US	T2D
ANALY_AGLT2_SL6-condor	ANALY_AGLT2_SL6	GreatLakesT2	AGLT2	US	T2D
ANALY_AGLT2_TIER3_TEST-condor	ANALY_AGLT2_TIER3_TEST	GreatLakesT2	AGLT2	US	T2D
ANALY_AM-04-YERPHI	ANALY_AM-04-YERPHI	AM-04-YERPHI	AM-04-YERPHI	NL	T3
RACKSPACE_CLOUD	RACKSPACE_CLOUD	CERN-CLOUD	CERN-PROD	CERN	T0
RAL-LCG2-lcgce04-grid3000M-pbs	RAL-LCG2	RAL-LCG2	RAL-LCG2	UK	T1
RAL-LCG2-lcgce04-grid4000M-pbs	RAL-LCG2_HIMEM	RAL-LCG2	RAL-LCG2	UK	T1
RAL-LCG2-lcgce10-grid3000M-pbs	RAL-LCG2	RAL-LCG2	RAL-LCG2	UK	T1
RAL-LCG2-lcgce10-grid4000M-pbs	RAL-LCG2_HIMEM	RAL-LCG2	RAL-LCG2	UK	T1
RAL-LCG2-lcgce11-grid3000M-pbs	RAL-LCG2	RAL-LCG2	RAL-LCG2	UK	T1
RAL-LCG2-lcgce11-grid4000M-pbs	RAL-LCG2_HIMEM	RAL-LCG2	RAL-LCG2	UK	T1
RAL-LCG2_HIMEM_SL6	RAL-LCG2_HIMEM_SL6	RAL-LCG2	RAL-LCG2	UK	T1
RAL-LCG2_MCORE	RAL-LCG2_MCORE	RAL-LCG2	RAL-LCG2	UK	T1
RAL-LCG2_SL6	RAL-LCG2_SL6	RAL-LCG2	RAL-LCG2	UK	T1
RAL-LCG2_TEST-lcgce07-grid3000M-lcgpbs	RAL-LCG2_TEST	RAL-LCG2_TEST	RAL-LCG2	UK	T1
RAL-LCG2_TEST-lcgce11-grid3000M-lcgpbs	RAL-LCG2_TEST	RAL-LCG2_TEST	RAL-LCG2	UK	T1
RDIGTEST	RDIGTEST	BNL	BNL-ATLAS	US	T1

SchedConfig parameters

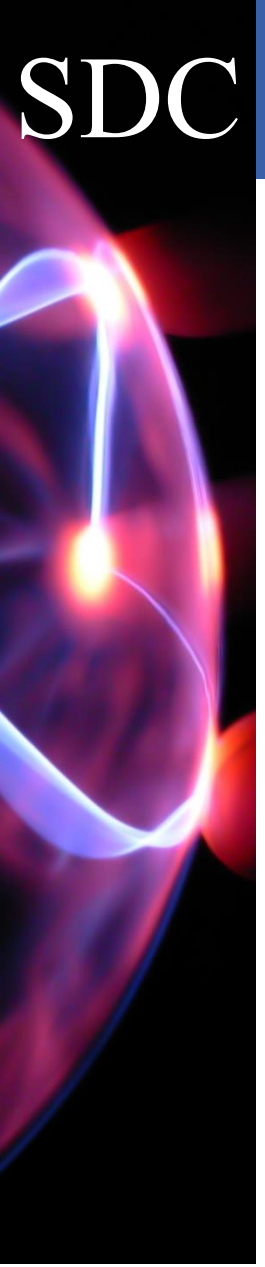
https://twiki.cern.ch/twiki/bin/viewauth/AtlasComputing/SchedconfigParameterDefinitions

- Ok-ish! … but need to clearly identify which are:
  - Needed by the Panda Server
  - Needed by the Pilot
  - Needed by the APF
  - Needed by some combination of the above, and/or needed by someone else
  - Not needed, just obsolete
- Moreover, by who should be filled?
  - by the site admin/cloud squad,
  - by ATLAS,
  - automatically by DDM (from AGIS)?
- Documentation need also examples on syntax

SchedConfig params - 2

- Other parameters could be useful to optimize the brokering
  - Input transfers to a site
  - Output transfers from a site
  - ? Do we need I/O between SE and WNs?
  - Let's see Artem presentation on the Network matrix:
    - Tools like Pd2p could/should benefit of the info on the transfer throughput, to avoid sending urgent datasets (triggered by waiting jobs) to slow sites may not be optimal
- At which level do we need those parameters?
  - We have been doing work to consolidate situation with many PQs behind a PR in one single PQ. -→ 1 PQ – 1 PR
    - Not true for NDGF (yet)
- How many of those parameters we may have common between the same ATLASSite/PandaSite ?
  - MCORE?

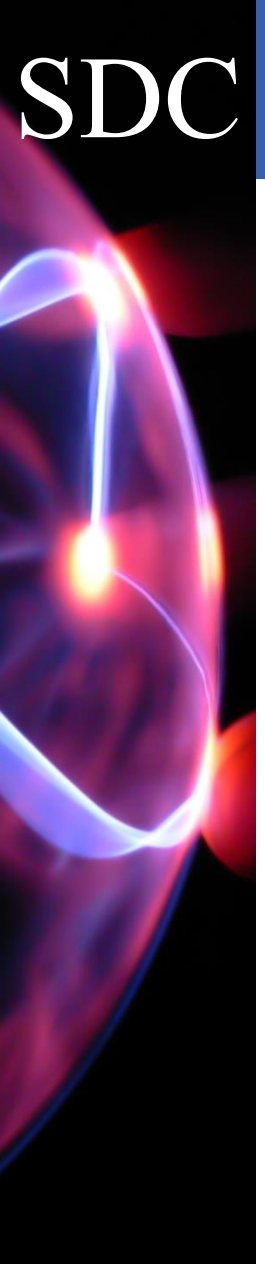
SchedConfig + Topology summary



- Topology: simplification could be beneficial!
  - Less objects: can we merge PandaSite to ATLASSite?
  - Share part of the parameters
- SchedConfig parameters: review of "needed by" could be useful
  - Setup a small group of people (e.g. Tadashi, Paul, Alden and few more…) to review the list and cleanup the obsolete?
- Other info could be useful to improve the brokering,
  - e.g. network info: let's discuss this after Artem talk

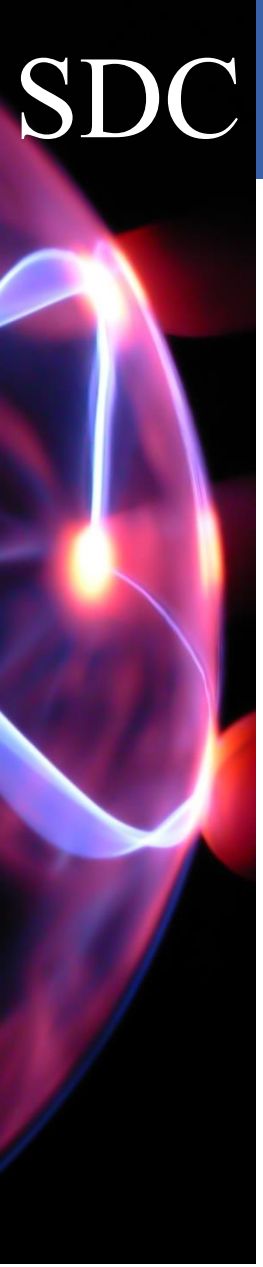
PD2P observations

- PD2P as of today can compete / affect:
  - transfers for production jobs -- PD2P triggered by waiting jobs use 'production' share/priority
  - transfers of analysis job output to the destSE
- Other observations:
  - Data transfer throughput is not infinite → not all the PD2P complete
  - Disk capacity is not infinite, primary data are occupying disks → PD2P replicas may be deleted before used
  - The number of analysis jobs varies largely depending on the period.
- Suggestions:
  - prioritize different PD2P categories, take the most useful ones as far as possible, may not be all.
  - normalize the numbers used in PD2P against the total number of jobs.
- Discussion ended 13 February 2013, but no conclusion: time to revive it?



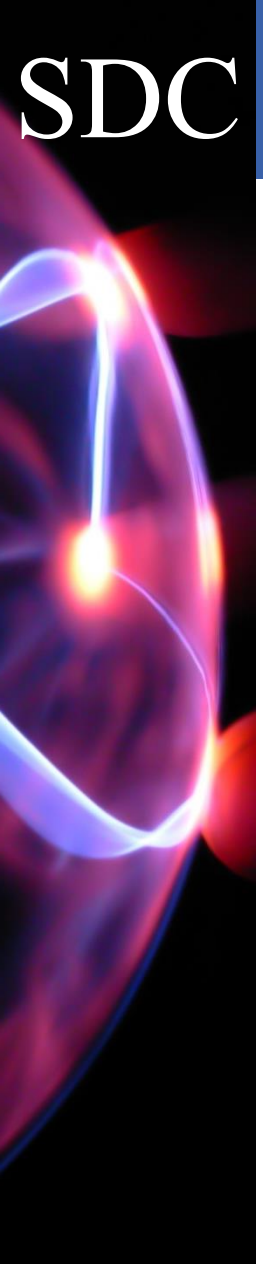
Production and Analysis queues

Production and Analysis queues



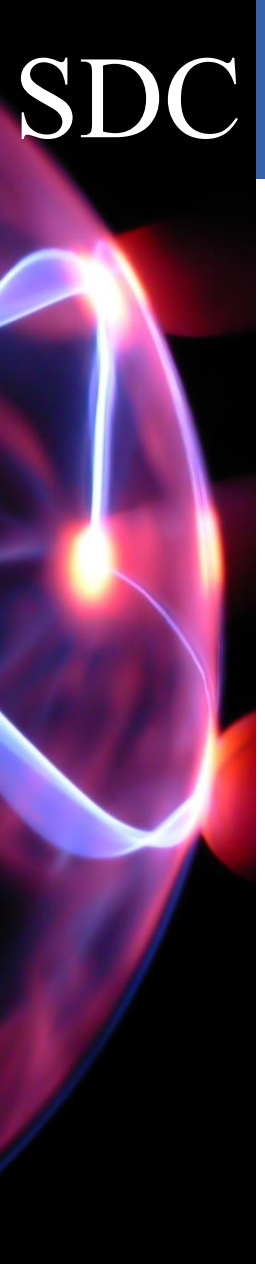
- Today hard partitioning at the site level
  - Batch system configured to provide 50/50 shares for analysis and production
- If we want to provide more resources for analysis we need
  - Decision of CREM
  - Ask sites to reconfigure
- Reconfiguring the BS shares is non trivial exercise at many sites with many VOs
- Then, all our effort of commissioning cloud resources and HPC for production to offload Grid and boost analysis looses a lot effectiveness
  - No obvious way to throttle production vs analysis for the same site in Panda

Production and Analysis queues



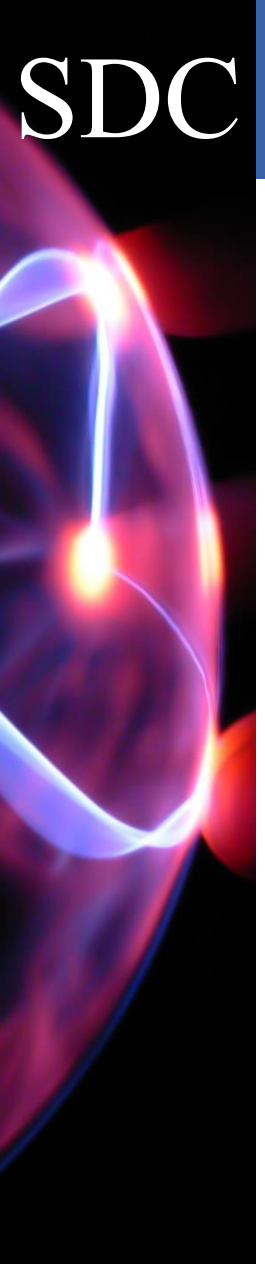
- ATLAS would really benefit if:
  - The same panda site could be used for both analysis and prod
  - The prod vs analysis shares for a given site (and globally) become a parameter in PanDA
  - PanDA dispatches jobs to sites not only based on priority but priority + shares
- Then we do not need to ask the sites to do anything
  - and a CREM decision can be implemented in zero time

Analysis and Production queues

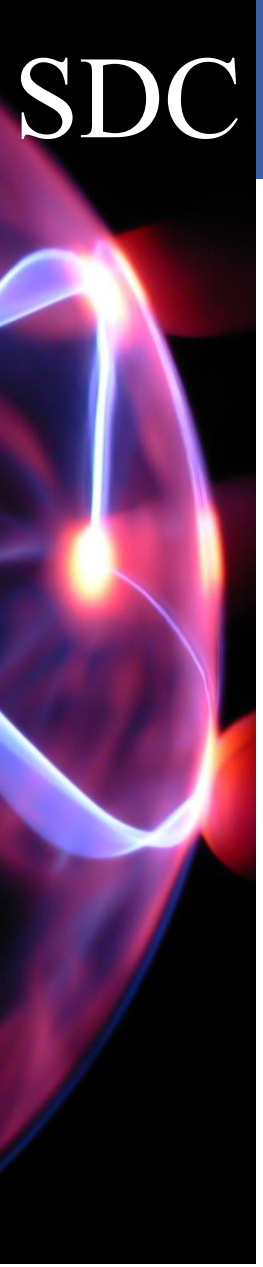


- What would we need:
  - No ANALY_ or production queues. Only one queue for both → Simplification
  - No pilots for production or analysis anymore, just one flavor → Simplification.
  - The pilot asks for a job. Can be analysis or prod. Then needs to get the right type of proxy (with or w/o prod role) and protect it.

Analysis and Production queues

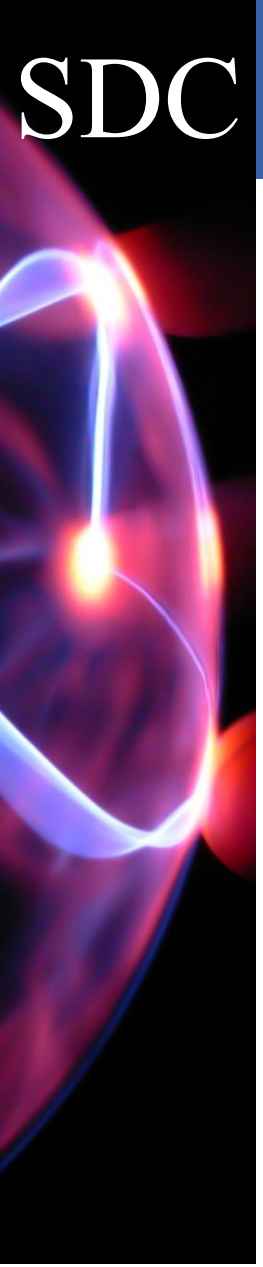


- Issues:
  - "Protecting a proxy" would probably mean gLexec. Other ideas?
  - ANALY_ and prod queues are fairly different in many respects
    - The main difference is in the data access method (copy-to-scratch vs direct I/O, different protocols, etc …)
    - But also other functionalities might apply for production and analysis only (e.g job recovery, FAX fallback, etc…)
    - In practice, some restructuring/rethinking at the level of AGIS/SchedConfig/pilot is needed
- This is work. But it brings many benefits. Please discuss.



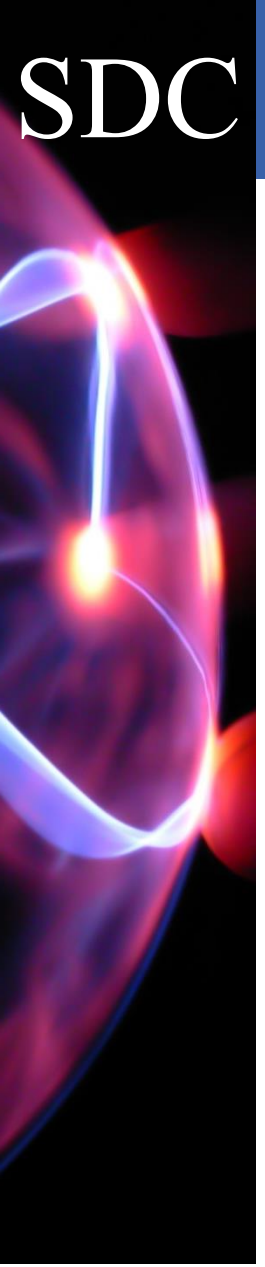
The HTTP Ecosystem

HTTP/WebDAV for HEP



- EMI provided the context for evolving towards HTTP/DAV
- This has been a starting point to evolve the Grid components towards "standard protocols", in particular HTTP
  - dCache, DPM/LFC, STORM are there
- HEP is a very demanding environment, we are 'just used' to things that are far beyond the needs of a Web farm or a Web developer
  - X509, proxies, large clusters with one entry point, swarms of jobs (greedier than an user browsing a page)…
- At the same time, HTTP is moving far more bytes than HEP, and has a far larger user base
- Our goal is to make HTTP/WebDAV distributed data access and management possible in the context of HEP and HPC, supporting well all the things that make HEP advanced
- Attractive for a professional to be formed on advanced uses of mainstream things
- Sharable easily by other communities than HEP
- There are use cases for which using mainstream apps may be desirable, browsers is just one of them

Where the features reside



- HTTP has interesting technical features
  - Multitalented, covers most existing use cases, while allowing new stuff
  - Applications just go straight to the data, wherever they are running
  - Staging GET/PUT (the Web case) or direct chunked access (the HPC case)
  - Direct data access can be made to work over WANs too
- We have the needed WebDAV DM features in the SEs
- We contributed a component that does HTTP/WebDAV storage federations
- We are contributing a client that implements ALL the HTTP things that HEP needs
  - This is DAVIX (in EPEL-testing now) and TDavixFile/TDavixSystem will be in ROOT 5.34 and 6
  - HTTP allows everything we need, mainstream clients are limited to the tech needs of a Web dev. What an user sees is limited only by the features of his client.
  - X509, proxies, S3, DFS, performance, chunked access, full redirection support, posix API and many others
- We are contributing a component that gives HTTP/WebDAV to the Xrootd framework (hey, testers wanted outside CERN!)
- We are rapidly approaching a ‘critical mass’ for an ecosystem to self-sustain
- IT tech forum on the HTTP ecosystem on the 25th of October

Thanks for the invitation!

Q?