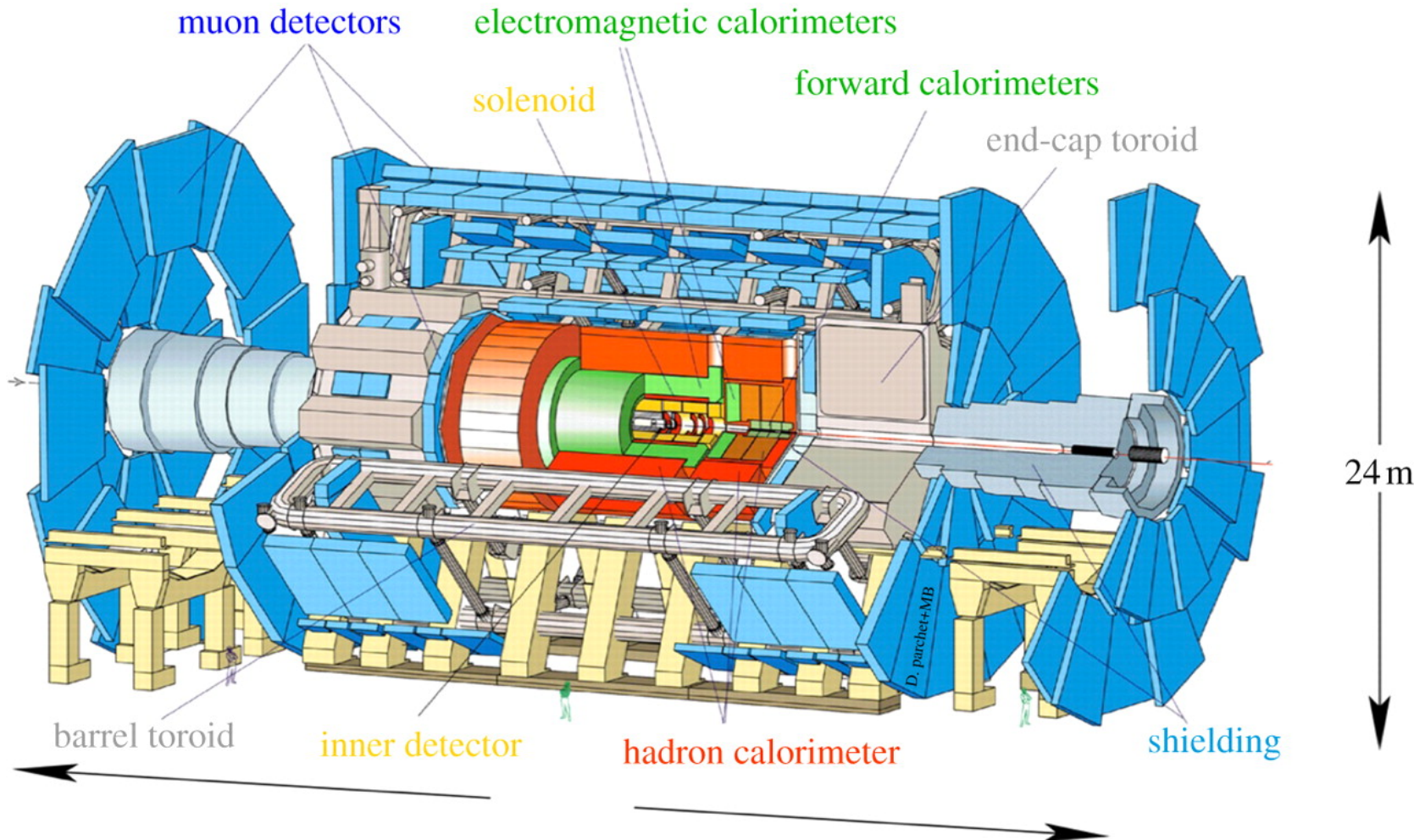


ATLAS production system

F Barreiro, M. Borodin, K. De, D. Golubkov,
A. Klimentov, T. Maeno, R. Mashinistov,
S.Padolski, T. Wenaus

GRID'2016

The ATLAS detector



~1/10th of its members

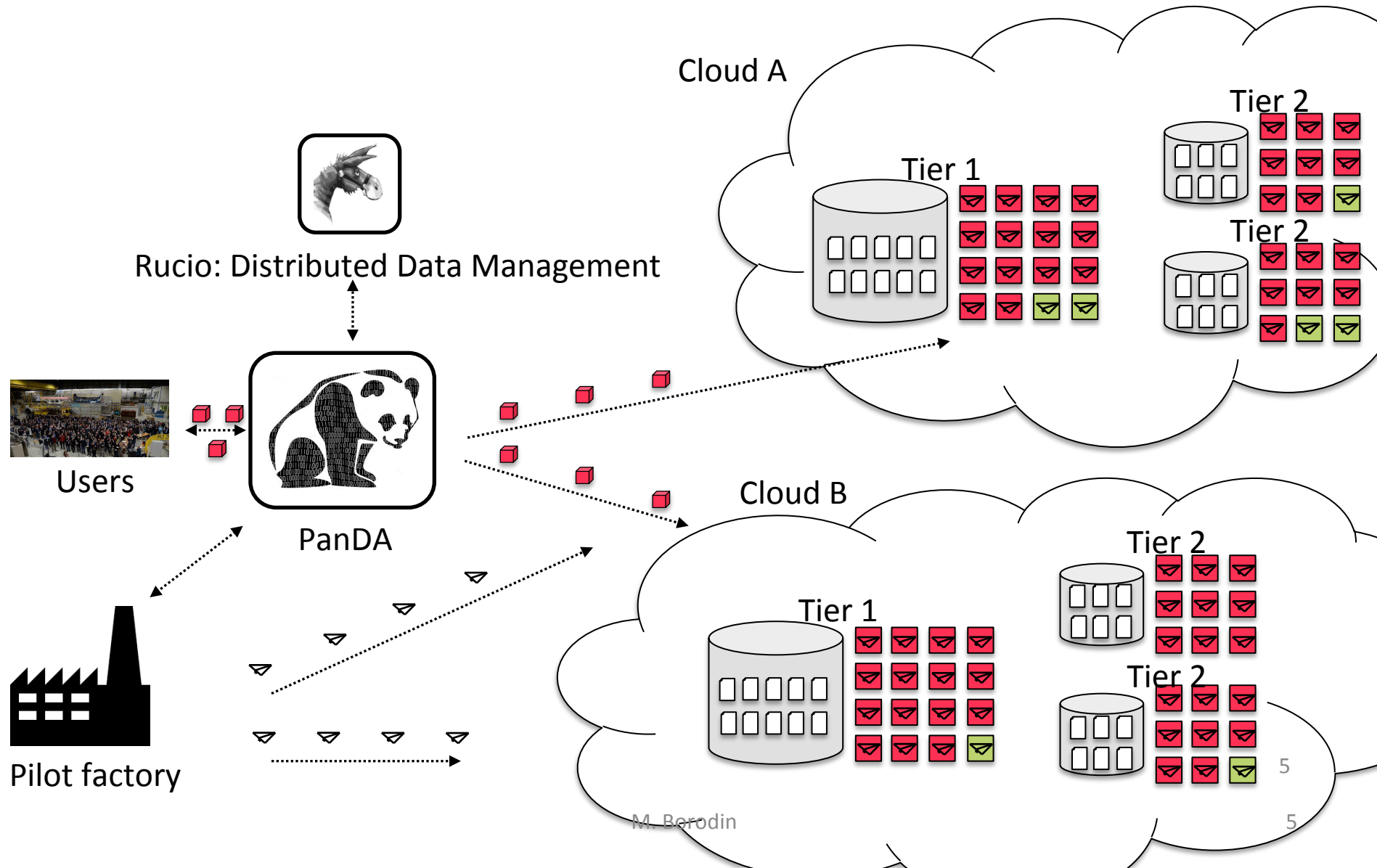


Production system

- **P**roduction and **D**istributed **A**nalysis system developed for ATLAS - PanDA
- Now also used by **AMS**, **ALICE**, **LSST**, **COMPASS** and others
- Many international partners: DoE HEP, DoE ASCR, NSF, CERN IT, OSG, ASGC, NorduGrid, European grid projects, Russian grid projects...

<http://news.pandawms.org/>

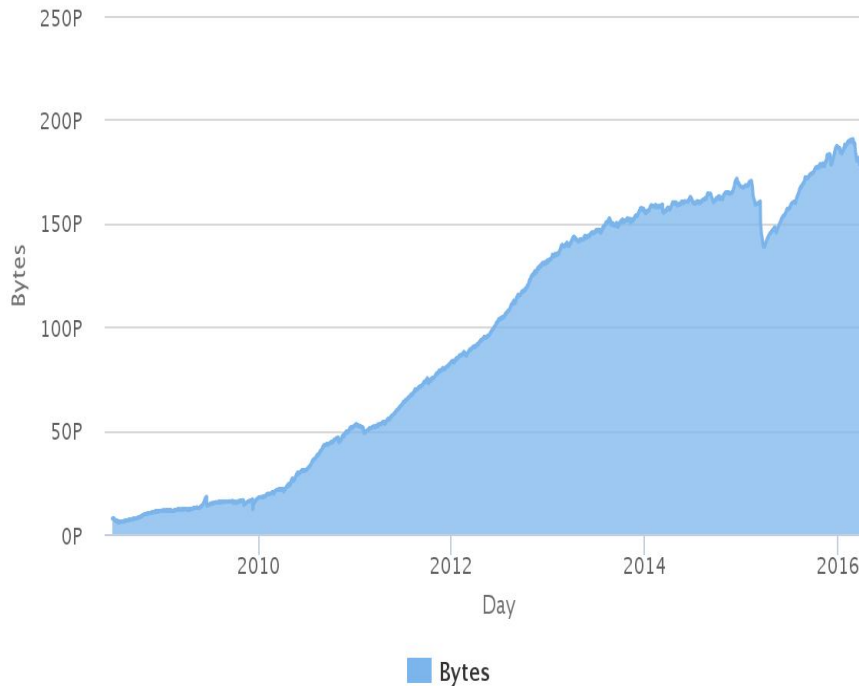
Production system in a glance



Orders of magnitude

ATLAS Data Overview

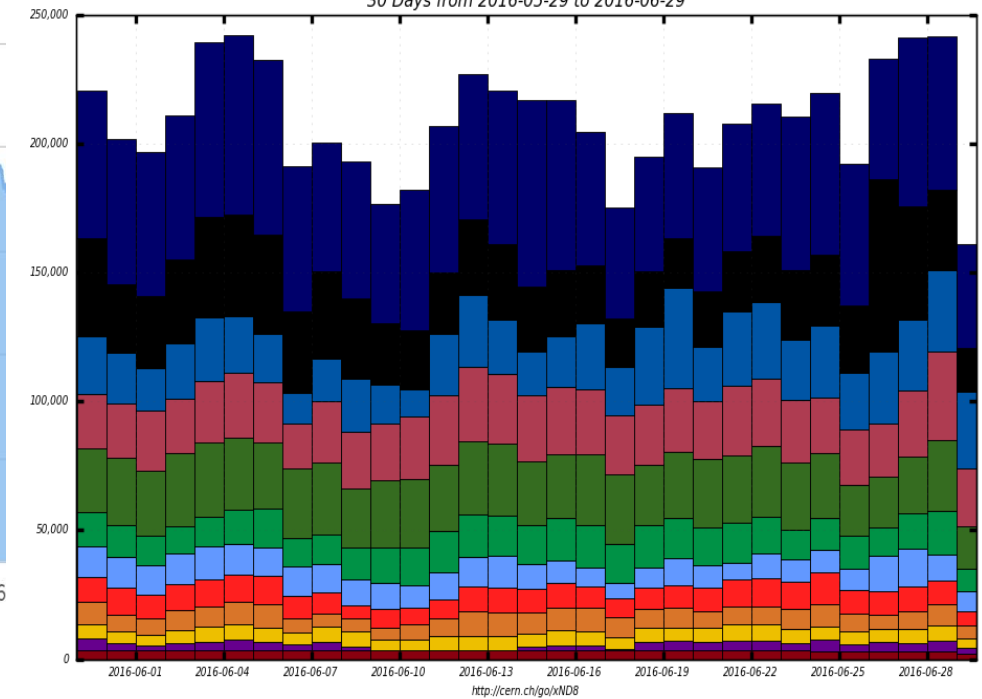
Worldwide



200 PB of ATLAS data is stored



Slots of Running Jobs 30 Days from 2016-05-29 to 2016-06-29



US DE FR IT CERN UK
RU ND CA NL ES
Maximum: 242,107, Minimum: 0.00, Average: 202,485, Current: 161,093

More than 200K
simultaneous jobs in the
system

Core idea in PanDA

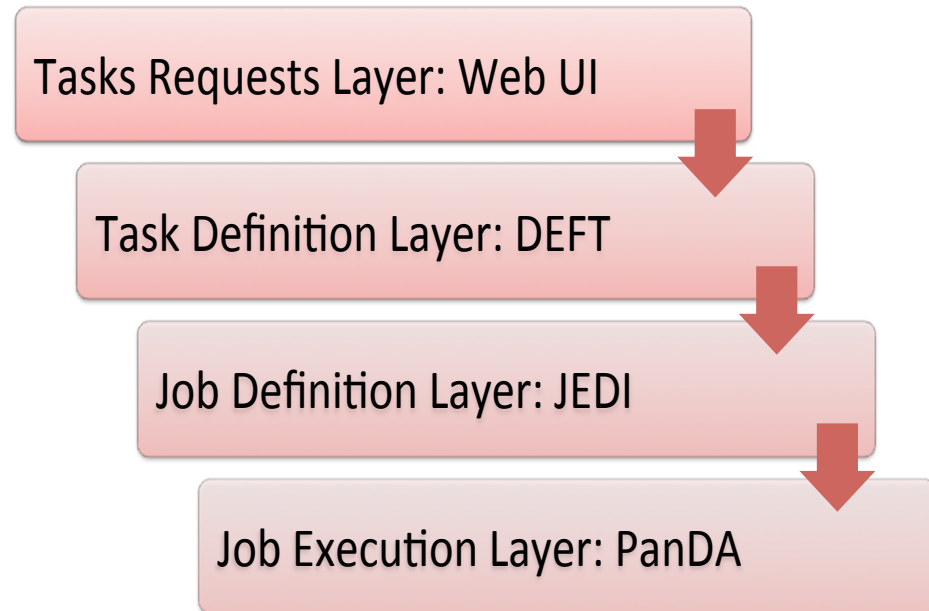
- Single entry point to the WLCG provide a central queue for users – similar to local batch systems
 - Make hundreds of distributed sites appear as local
- Reduce site related errors and reduce latency
 - Build a pilot job system – late transfer of user payloads
 - Crucial for distributed infrastructure maintained by local experts
- Hide middleware while supporting diversity and evolution
 - Atlas production system interacts with middleware – users see high level workflow
- Hide variations in infrastructure
 - Atlas production system presents uniform ‘job’ slots to user (with minimal sub-types)
 - Easy to integrate grid sites, clouds, HPC sites ...
- Production and Analysis users see same system
 - Same set of distributed resources available to all users
 - Highly flexible system, giving full control of priorities to experiment

Key features of ATLAS production system development

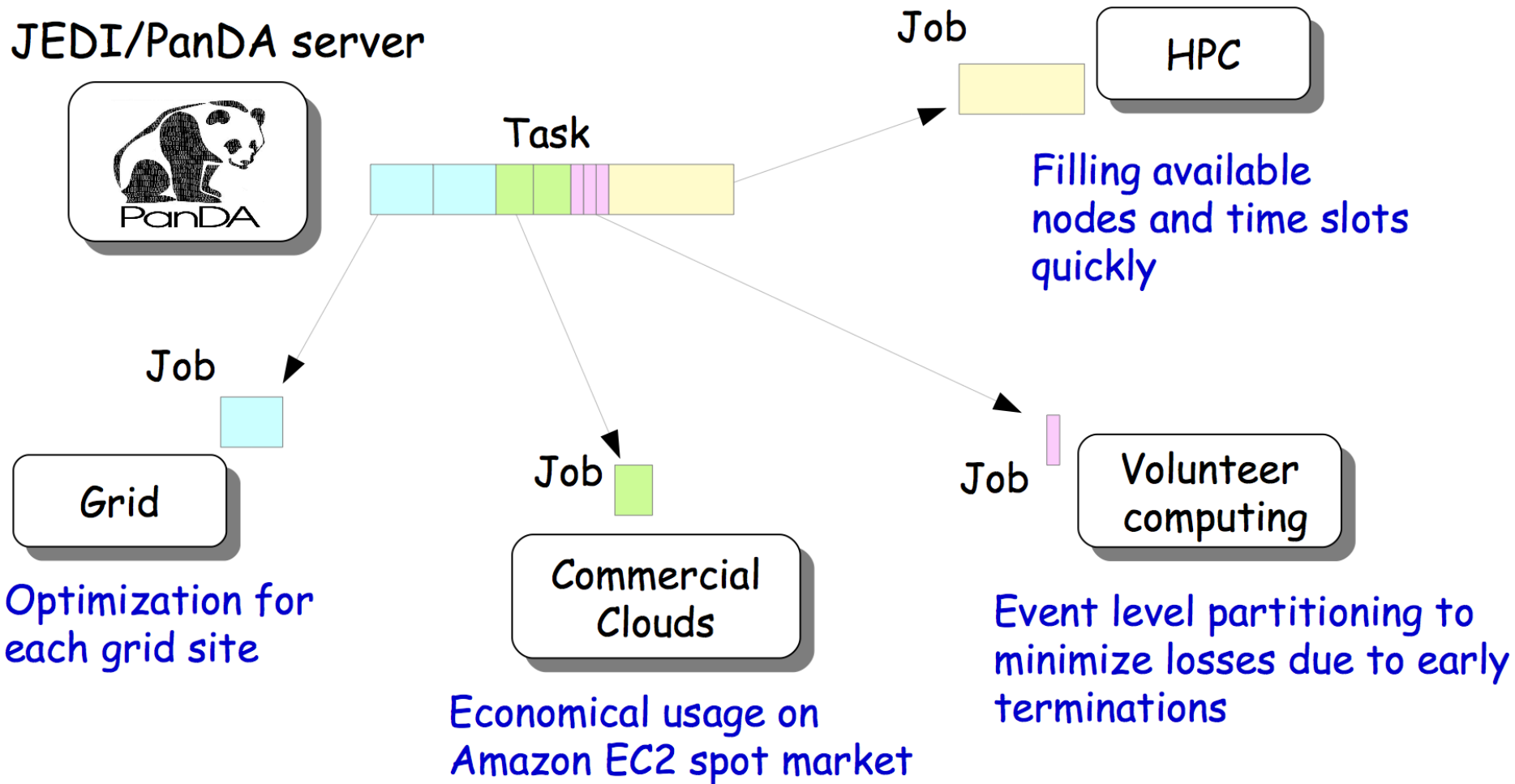
- Workflow is maximally asynchronous
- Pilot based job execution system
 - Condor based pilot factory
 - Payload is sent only after execution begins on CE
 - Minimize latency, reduce error rates
- Central job queue
 - Unified treatment of distributed resources
 - SQL DB keeps state - critical component
- Automatic error handling and recovery
- Extensive monitoring
- Modular design
- RESTful communications
- GSI authentication
- Use of Open Source components

ATLAS production system components

- Web UI for Managers and Users provides the interface for task and production request managing and monitoring at the higher level
- Database Engine for Tasks (DEFT): is responsible for formulating the tasks, **chains** of tasks and also task groups (**production request**), complete with all necessary parameters
 - It also keeps track of the state of production requests, chains and their constituent tasks
- Job Execution and Definition Interface (JEDI): is an intelligent component in the panda server to have capability for **task-level** workload management.
 - Key part of it is **‘Dynamic’** job definition, which highly optimizes resources usage compare to ‘Static’ model used in ProdSys1.
 - Dynamic job definition in JEDI is also crucial for multi-core, HPC's and other new requirements



Dynamic Job definition



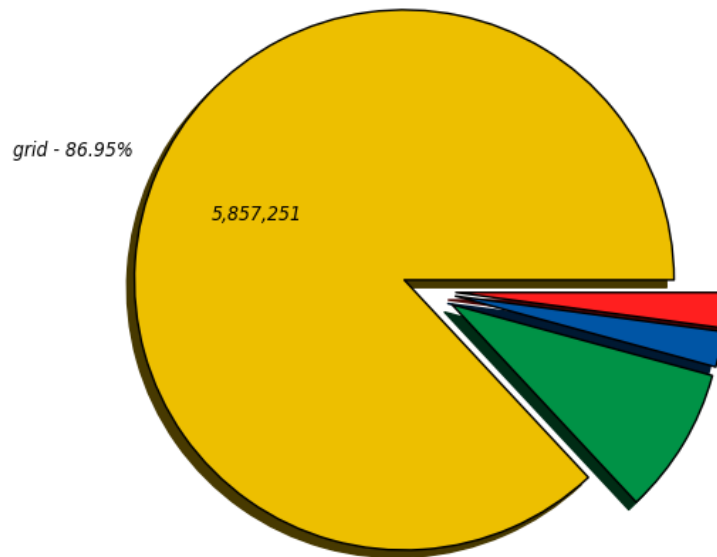
Dynamic job definition benefits

- Excluding requirements from users of detailed knowledge on computing resources
 - Especially for heterogeneous resources, e.g., many CPU cores, very short walltime limit, etc
- Self-optimization of job parameters
 - Real job metrics are collected using scout jobs
 - A small number (~ 10) of jobs (\equiv scout jobs) are generated for each task with minimum input chunks
 - Job parameters are optimized using job metrics for the rest of input
- Simplification of client tools and centralization of user functions

Extending beyond the Grid



Completed jobs (Sum: 6,736,023)



Cloud and HPC resources are steadily gaining territory

Example for 22-29 June 2016

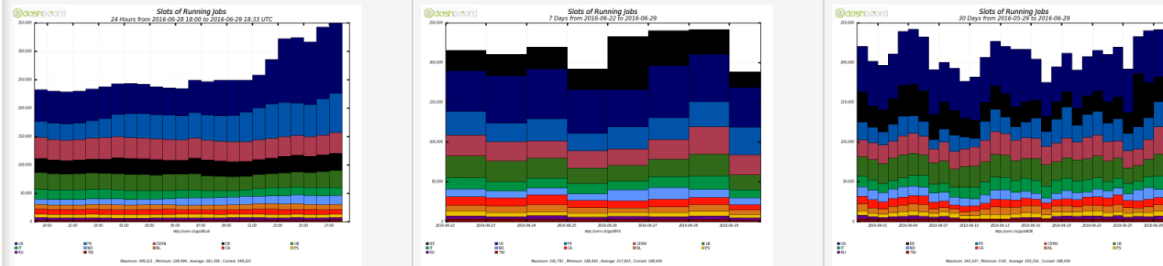
■ grid - 86.95% (5,857,251) ■ local - 8.82% (593,886) ■ hpc - 2.13% (143,377) ■ cloud - 2.10% (141,509)

Monitoring

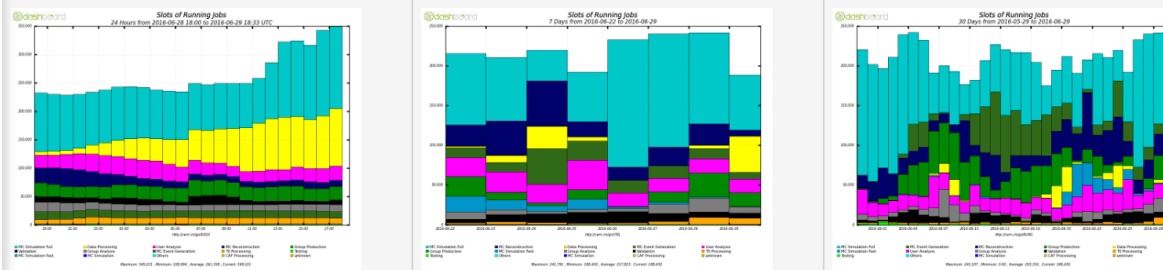
ATLAS PanDA Dash - Tasks - Jobs - Errors - Users - Sites - Incidents - Search Admin

ATLAS PanDA monitor home

Global concurrent running job core counts, all sites, all job types, by cloud, last 1, 7, 30 days



Global concurrent running job core counts, all sites, all job types, by activity, last 1, 7, 30 days



Search

PanDA job ID or name	<input type="text"/>	Submit
Batch ID	<input type="text"/>	Submit
Task ID	<input type="text"/>	Submit
Task name	<input type="text"/>	Submit
Request ID	<input type="text"/>	Submit
Tasks for Request IDs	from <input type="text"/> to <input type="text"/>	Submit
Jobs for Request IDs	from <input type="text"/> to <input type="text"/>	Submit

News

- 20150318: Memory information added to jobs and tasks pages
- 20150316: RW metric added to dashboard
- 20150205: Response time of tasks display improved
- 20150205: Wildcard search of jobs on job parameters added ([ATLASPANDA-133](#))
- 20150205: Dataset information added to JSON response ([ATLASPANDA-109](#))
- 20150205: Wildcard search on jobs added ([ATLASPANDA-40](#))
- 20141229: Main page plots show all jobs by cloud and activity
- 20141219: curl dumps of job params. See job list page help.
- 20141216: Request ID shown for jobs, range search added
- 20141215: Sort by duration option for job lists

Tasks requests - use-cases

- Production system is a workflow driven system and it's used for dealing with the all variety of ATLAS activities:
 - Simulation (MC production)
 - Data processing (“Tier0” processing and reprocessing)
 - High Level Trigger (HLT) reprocessing
 - Derivation and train production (slimming, skimming...)
 - Event Index
- Typical ATLAS workflow composed of many data transformation steps, e.g. the Monte Carlo simulations workflow is composed of many steps: generate hard-processes, hadronize signal and minimum-bias events, simulate energy deposition in the ATLAS detector, digitize electronics response, simulate triggers, reconstruct data, transform the reconstructed data into reduced forms for physics analysis



MC production request creation

- Creating of production request is one of the examples how different workflows can be integrated to the system
- For MC Google spreadsheet were used by user to provide a data for MC tasks. In ProdSys2 it was adopted so user can submit input data in the same format as they use before.

	A	B	C	D	E	F	G	H	I
1	brief	datasetNum	ESD,RD O	Joboptions	Event (FullSim) Config 25ns	Event (AF2) Config 25ns	Priority	Evgen	Sirn
2	Powheg+Pythia6 nominal ttbar (hdamp = mtop)	410000		MC15.410000.PowhegPythiaEvtGen_P2012_ttbar_hdamp172p5_nonallhad.p y	30000000		0	e3698	s2608
3	Powheg+Pythia6 nominal ttbar (hdamp = mtop)	410000		MC15.410000.PowhegPythiaEvtGen_P2012_ttbar_hdamp172p5_nonallhad.p y		30000000	0	e3698	a777
4									

+ MC15.410000.PowhegPythiaEvtGen_P2012_ttbar_hdamp172p5_nonallhad.py
 (Fullsim)Extension of ttbar nominal - additional 30M events events: 30000000

e3698	s2608	s2183		r6765	r6282				partially_submitted	edit (saved)
-------	-------	-------	--	-------	-------	--	--	--	---------------------	------------------------------

T: running
^ext.^

+ MC15.410000.PowhegPythiaEvtGen_P2012_ttbar_hdamp172p5_nonallhad.py
 (Atlfast)Extension of ttbar nominal - additional 30M events events: 30000000

e3698	a766						a777	r6282	not_submitted	edit (saved)
-------	------	--	--	--	--	--	------	-------	---------------	------------------------------

Reprocessing production request creation

- Reprocessing workflow has a tree structure, where output of one task can be an input for several more tasks – interface for creating such structure was developed for it.

Slice #0 step #0

Datasets list:
data12_8TeV.00204158.express_express.merge.RAW

Or search datasets in ddm/producers by filter:
dataset pattern

Find datasets

Dataset Name **events**

AMI tag: r6461
Output: AOD.ESD

cmtconfig: default
project: lumib1

Files per job: 1
GB per job:

Destination token:

JEDI internal merging:

Add step **Remove step** **Find**

0 data12_8TeV.00204158.express_express.merge.RAW
r6461 submitted [edit \(saved\)](#)
T: finished

1
r6461 p2298 p2301 p2301 submitted [edit \(saved\)](#)
T: done done done

2
r6461 p2299 p2299 submitted [edit \(saved\)](#)
T: done done

Slice #1 step #0

Parent step: Slice #0 step #0
Inp: AC

AMI tag: r6461
Output formats (e.g. AOD.ESD): AOD.TAG
Events per job: 1000
Tot: -1

HLT production request creation

- HLT reprocessing has a well defined workflow, so interface for HLT production request creation includes only a few fields.

Dataset:

Short description(request title):

Link to JIRA ticket:

Two step reco:

Outputs:

AOD

ESD

HIST

HIST_HLTMON

NTUP_TRIGCOST

NTUP_TRIGRATE

Priority:

Sites: CERN-PROD_SHORT,FZK-LCG2_HIMEM,IN2P3

Common project mode:
cmtconfig=x86_64-slc6-gcc48-opt;cloud=CERN;skipsout=yes;

Reco specific project mode addition:
useRealNumEvents=yes;tgtNumEventsPerJob=500;

[Proceed](#)

Expert mode

Request ID:	Description:	Reference:	Manager:	Physic group:	Project:	Status:
menu(=) 4173	Reprocessing 25ns EB with CAFHLT 20.2.3.2.3 and MC Menu	ATR-12348	damazio Me	THLT	data15_13TeV	processed

last comment: - [New comment](#)

[Show/hide long description](#)

Total input: 6, from them approved: 6

[Select All](#) Filter by: Filter: Filter by status: Sort:

[Show](#) [Bind](#)

	0	1	2	3	4	5	6	7	8	9

Replace empty

0	data15_13TeV.00276952.physics_EnhancedBias.merge.RAW	7101									submitted	edit (saved)
T:												
1	data15_13TeV.00276952.physics_EnhancedBias.merge.RAW	7101	p2418	p2418							submitted	edit (saved)
T:												
2	data15_13TeV.00276952.physics_EnhancedBias.merge.RAW	7101	p2418	p2418							submitted	edit (saved)
T:												
3	data15_13TeV.00276952.physics_EnhancedBias.merge.RAW	7101	p2365	p2365							submitted	edit (saved)
T:												
4	data15_13TeV.00276952.physics_EnhancedBias.merge.RAW	7101	p2365								submitted	edit (saved)
T:												
5	data15_13TeV.00276952.physics_EnhancedBias.merge.RAW	7101	p2417								submitted	edit (saved)
T:												

[Select All](#)

Derivation production request creation

- Derivation is using so called “train” model, there each input runs on some of many predefined outputs. To manage with complexity, pattern request is created first, and system is using this pattern to generate possible options for the specific derivation production request creation interface.

15 DAOD_SUSY1 DAOD_SUSY10 DAOD_SUSY11 DAOD_SUSY4 DAOD_SUSY5 DAOD_SUSY9

16 DAOD_SUSY2 DAOD_SUSY3 DAOD_SUSY6 DAOD_SUSY7 DAOD_SUSY8

17 DAOD_TAUP1 DAOD_TAUP3

18 DAOD_TCAL1

19 DAOD_TOPQ1 DAOD_TOPQ2 DAOD_TOPQ3 DAOD_TOPQ4

Datasets:

```
mc15_13TeV:mc15_13TeV.301325.Pythia8EvtGen_A14NNPDF23LO_zprime1000_tt.merge.AOD.e4061_s2608_s2183_r7000_r6282/
mc15_13TeV:mc15_13TeV.301329.Pythia8EvtGen_A14NNPDF23LO_zprime2000_tt.merge.AOD.e4061_s2608_s2183_r6987_r6282/
```

Cancel Save

8	mc15_13TeV:mc15_13TeV.301325.Pythia8EvtGen_A14NNPDF23LO_zprime1000_tt.merge.AOD.e4061_s2608_s2183_r7000_r6282/	p2419	submitted	edit (saved)
T:	done			
17	mc15_13TeV:mc15_13TeV.301325.Pythia8EvtGen_A14NNPDF23LO_zprime1000_tt.merge.AOD.e4061_s2608_s2183_r7000_r6282/	p2419	submitted	edit (saved)
T:	done			
1	mc15_13TeV:mc15_13TeV.301329.Pythia8EvtGen_A14NNPDF23LO_zprime2000_tt.merge.AOD.e4061_s2608_s2183_r6987_r6282/	p2419	submitted	edit (saved)
T:	done			
50	mc15_13TeV:mc15_13TeV.301329.Pythia8EvtGen_A14NNPDF23LO_zprime2000_tt.merge.AOD.e4061_s2608_s2183_r6987_r6282/	p2419	submitted	edit (saved)
T:	running			

Work with production request

- Tasks requests Web U/I provides many general and experiment specific features:
 - **Bookkeeping.**
 - **Approve management.** E.g. MC production request required several levels of approval.
 - **Monitoring.** User can easily follow progress of a running tasks.
 - **Error Handling.** Task could failed because of many permanent (e.g. bug in software) and temporal (storage is down) reasons. To be able quickly understand root of the problem and fix it by redefining the task is one of the major feature of the production system.
 - **Chaining** one production to the other. E.g. derivation production could be chained to MC or reprocessing task, that significantly speed up them.
 - **Automation** task submission. User can defined a pattern and when new data appears tasks are started automatically.
 - ...

Summary

- Production system has performed well for *ATLAS* including the LHC Run 1 data taking period
- New components and features have been delivered to *ATLAS* before LHC Run 2
- Many developments and challenges to come
 - New resources as Clouds and HPC
 - Full integration of network as a resource in workload management