The second generation of the ATLAS Production System: expertise and future evolution

•••

M. Borodin (U of Iowa)
On behalf of F. Barreiro, M. Golosova, D. Golubkov, A. Klimentov,
T. Maeno, P. Nilsson, M. Titov and ATLAS collaboration

ACAT, Saas-Fee, 13 Mar 2019

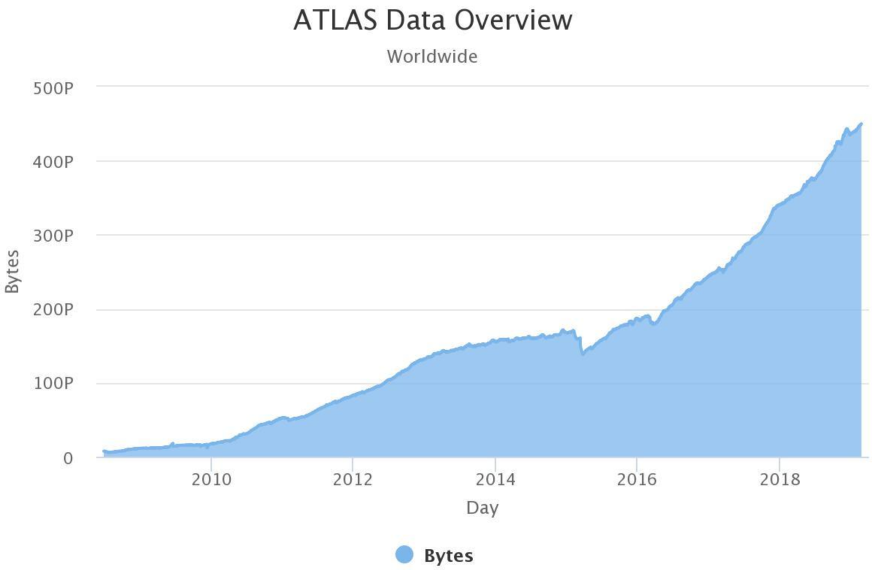
Introduction

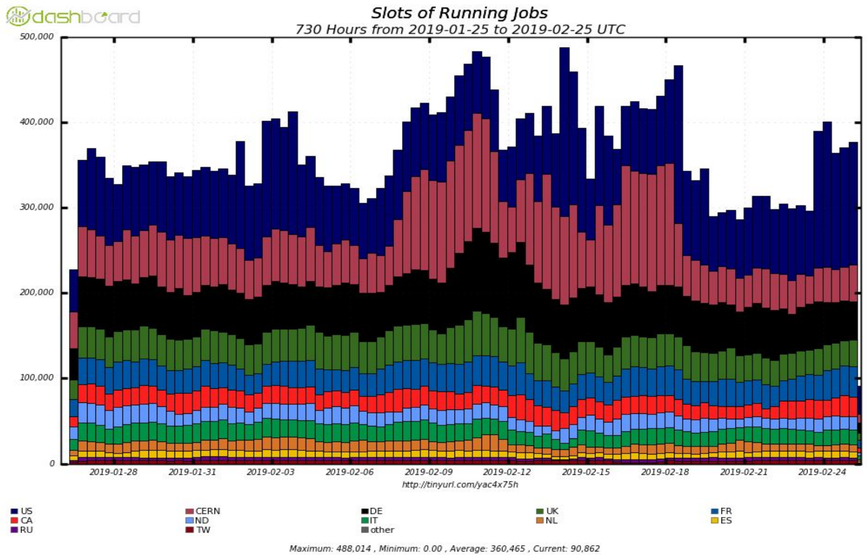
- PanDA - **P**roduction **an**d **D**istributed **A**nalysis System
  - Designed to meet ATLAS production/analysis requirements for a data-driven workload management system capable of operating at LHC data processing scale
- New generation of ATLAS production system was developed for Run 2 and beyond – **ProdSys2**
  - Improved resource utilization
  - New types of computing resources: HPC, Clouds
  - Improved usability and robustness

ATLAS production system design goals

- Deliver transparency of data processing in a distributed computing environment
- Achieve high level of automation to reduce operational effort
- Flexibility in adapting to evolving hardware, computing technologies and network configurations
- Scalable to the experiment requirements
- Support diverse and changing middleware
- Insulate user from hardware, middleware, and all other complexities of the underlying system
- Support custom workflow of individual physics groups
- Incremental and adaptive software development

Orders of magnitude





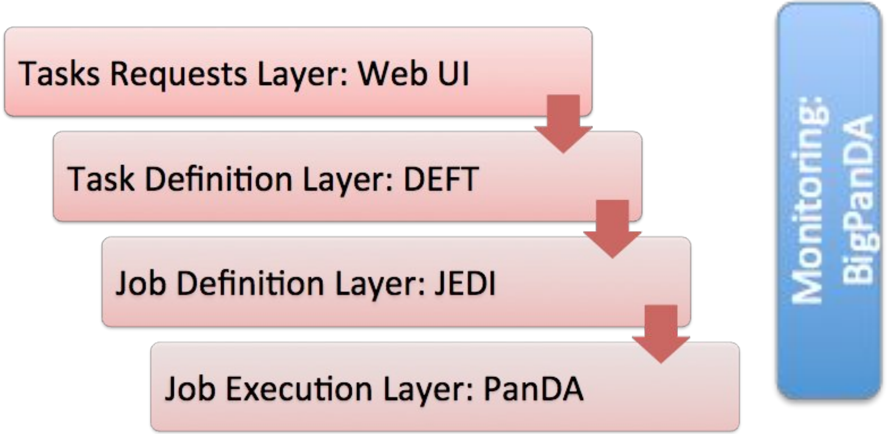
https://bigpanda.cern.ch

400 PB of data is managed by ATLAS DDM system (Rucio)

More than 300K cores used by simultaneously running jobs in the system

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


Tasks Requests Layer: Web UI
Task Definition Layer: DEFT
Job Definition Layer: JEDI
Job Execution Layer: PanDA
Monitoring: BigPanDA

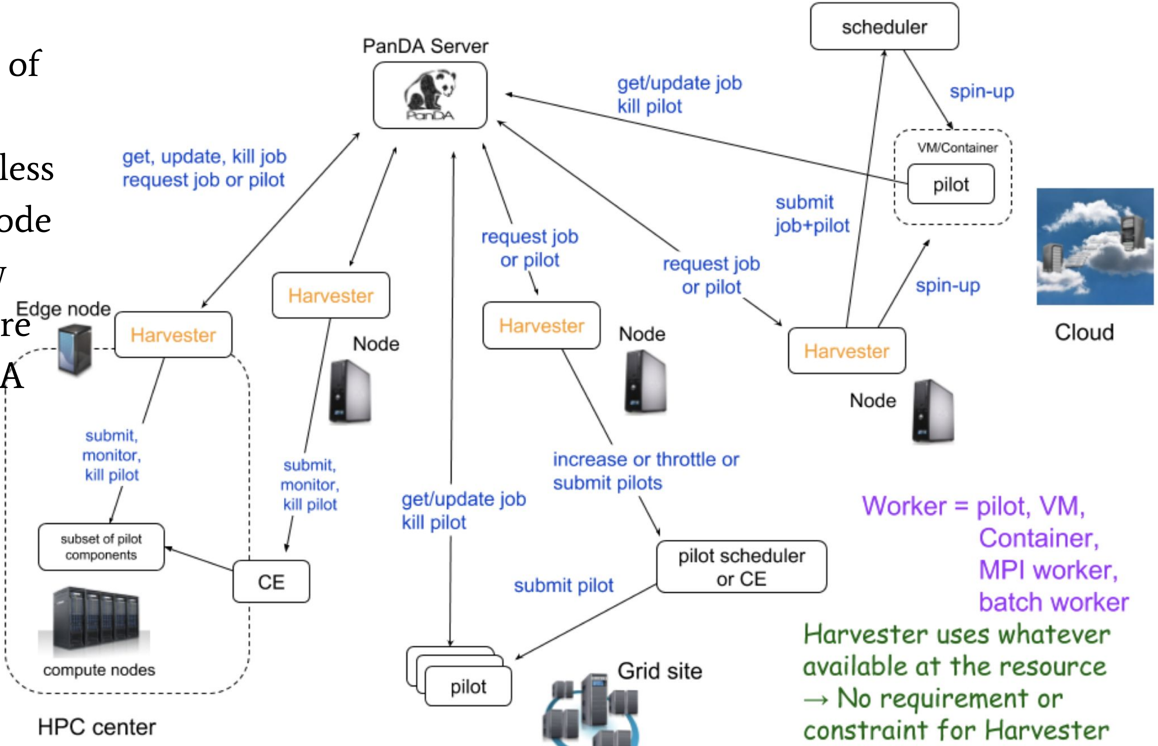
*Task consists of jobs that all run the same program.

ATLAS production system components (cont.)

- 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 compared to ‘Static’ model used in ProdSys1.
    - Dynamic job definition in JEDI is also crucial for multi-core, HPCs and other new requirements
- Monitoring (**BigPanDA**): progress, status and error diagnostics for all components.
- The PanDA **pilot** is an execution environment used to prepare the computing element, request the actual payload (a production or user analysis job), execute it, and clean up when the payload has finished. Input and output are transferred from/to storage elements, including object stores.

Harvester

Harvester is a resource-facing service between the PanDA server and collection of pilots for resource provisioning and workload shaping. It is a lightweight stateless service running on a VObox or an edge node of HPC centers to provide a uniform view for various resources. The following picture shows how harvester interacts with PanDA and resources.

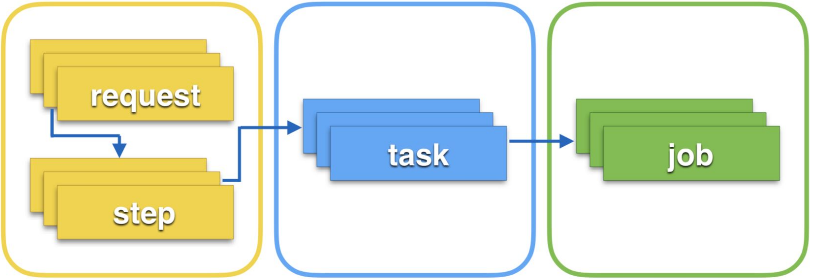


Worker = pilot, VM, Container, MPI worker, batch worker

Harvester uses whatever available at the resource
→ No requirement or constraint for Harvester deployment

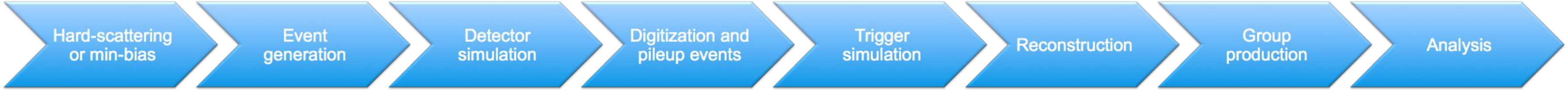
DEFT data model

- Model is represented by multilevel relational instances:
  - **Request** -> **Slice**(chain of steps) -> **Step** -> **Task**
  - Depending on workflow each instance could play a role of a template
  - Tasks are created by initiating a step instance.
  - **Hashtags** are used to union an arbitrary number of tasks



DEFT workflows

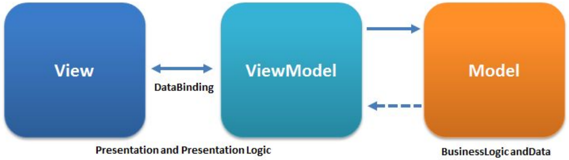
- ATLAS production workflows were implemented in chosen model
  - **MC simulation** is composed of many steps: generate hard-processes, hadronize signal and minimum-bias events, simulate energy deposition in the ATLAS detector, digitize electronics response, simulate triggers, reconstruct data, transform the reconstructed data into reduced forms for physics analysis



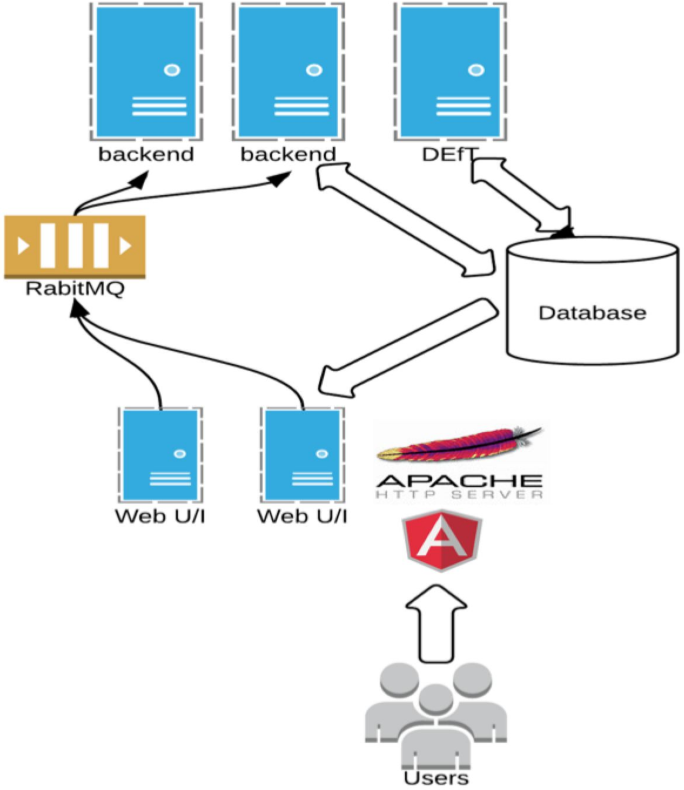
- **Data Reprocessing** workflow has a tree structure, where output of one task can be an input for several more tasks
- **Derivation** is using so called "train" model, there each input runs on some of many predefined outputs
- **Tier-0** workflow
- **HLT, EventIndex,** ...

DEFT and web UI development and deployment

- Key development points
  - Agile methodology: continuous meetings with the main users and often releases
  - Using open source
    - Django, Celery, AngularJS
  - «Model View ViewModel» approach

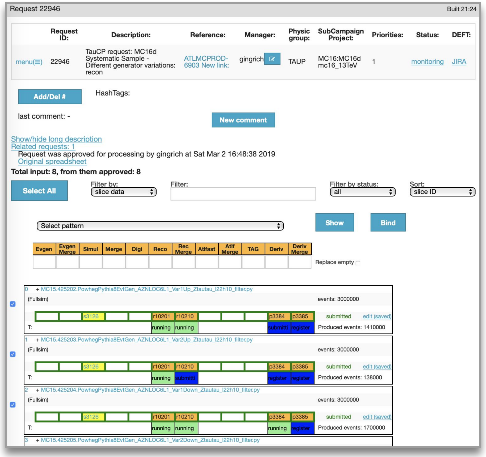


- Using CERN SSO for authentication and authorization



Web UI

Request management



Request 22946 Built 21:24

	Request ID:	Description:	Reference:	Manager:	Physic group:	SubCampaign Project:	Priorities:	Status:	DEFT:
menu(≡)	22946	TauCP request: MC16d Systematic Sample - Different generator variations: recon	ATLMCPROD-6903 New link:	gingrich	TAUP	MC16:MC16d mc16_13TeV	1	monitoring	JIRA

Add/Del # HashTags:

last comment: - New comment

Show/hide long description
Related requests: 1
Request was approved for processing by gingrich at Sat Mar 2 16:48:38 2019
Original spreadsheet

Total input: 8, from them approved: 8

Select All Filter by: slice data Filter: Filter by status: all Sort: slice ID

Select pattern Show Bind

Evgen	Evgen Merge	Simul	Merge	Digi	Reco	Rec Merge	Atlfast	Atlf Merge	TAG	Deriv	Deriv Merge

Replace empty □

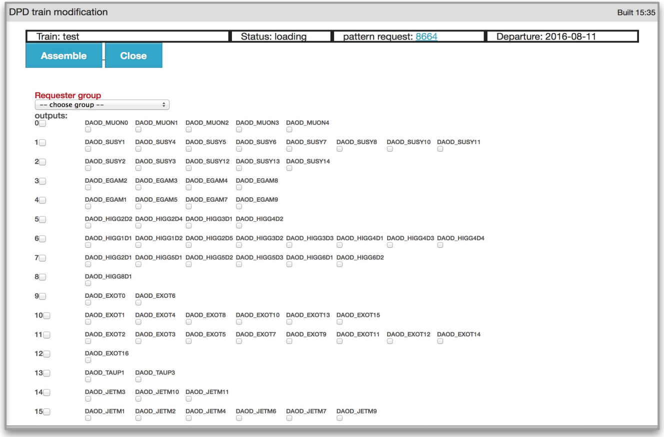
☑ 0 + MC15.425202.PowhegPythia8EvtGen_AZNLOC6L1_Var1Up_Ztautau_l22h10_filter.py
(Fullsim) events: 3000000
s3126 r10201 r10210 p3384 p3385 submitted edit (saved)
T: running running submitti register Produced events: 1410000

☑ 1 + MC15.425203.PowhegPythia8EvtGen_AZNLOC6L1_Var2Up_Ztautau_l22h10_filter.py
(Fullsim) events: 3000000
s3126 r10201 r10210 p3384 p3385 submitted edit (saved)
T: running submitti register register Produced events: 138000

☑ 2 + MC15.425204.PowhegPythia8EvtGen_AZNLOC6L1_Var1Down_Ztautau_l22h10_filter.py
(Fullsim) events: 3000000
s3126 r10201 r10210 p3384 p3385 submitted edit (saved)
T: running running running register Produced events: 1700000

3 + MC15.425205.PowhegPythia8EvtGen_AZNLOC6L1_Var2Down_Ztautau_l22h10_filter.py

Request creation interface



DPD train modification Built 15:35

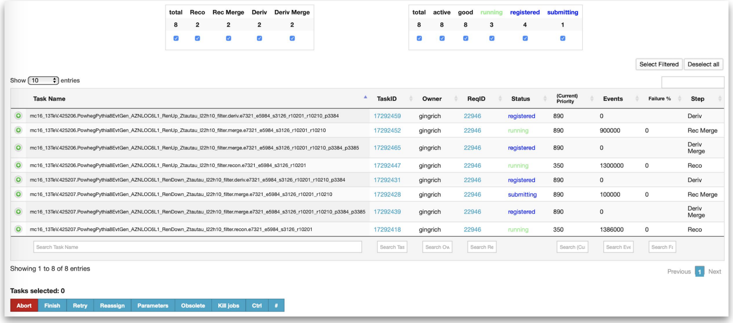
Train: test | Status: loading | pattern request: 8664 | Departure: 2016-08-11

Assemble Close

Requester group
-- choose group --

outputs:
0□ DAOD_MUON0 □ DAOD_MUON1 □ DAOD_MUON2 □ DAOD_MUON3 □ DAOD_MUON4 □
1□ DAOD_SUSY1 □ DAOD_SUSY4 □ DAOD_SUSY5 □ DAOD_SUSY6 □ DAOD_SUSY7 □ DAOD_SUSY8 □ DAOD_SUSY10 □ DAOD_SUSY11 □
2□ DAOD_SUSY2 □ DAOD_SUSY3 □ DAOD_SUSY12 □ DAOD_SUSY13 □ DAOD_SUSY14 □
3□ DAOD_EGAM2 □ DAOD_EGAM3 □ DAOD_EGAM4 □ DAOD_EGAM8 □
4□ DAOD_EGAM1 □ DAOD_EGAM5 □ DAOD_EGAM7 □ DAOD_EGAM9 □
5□ DAOD_HIGG2D2 □ DAOD_HIGG2D4 □ DAOD_HIGG3D1 □ DAOD_HIGG4D2 □
6□ DAOD_HIGG1D1 □ DAOD_HIGG1D2 □ DAOD_HIGG2D5 □ DAOD_HIGG3D2 □ DAOD_HIGG3D3 □ DAOD_HIGG4D1 □ DAOD_HIGG4D3 □ DAOD_HIGG4D4 □
7□ DAOD_HIGG2D1 □ DAOD_HIGG5D1 □ DAOD_HIGG5D2 □ DAOD_HIGG5D3 □ DAOD_HIGG6D1 □ DAOD_HIGG6D2 □
8□ DAOD_HIGG8D1 □
9□ DAOD_EXOT0 □ DAOD_EXOT6 □
10□ DAOD_EXOT1 □ DAOD_EXOT4 □ DAOD_EXOT8 □ DAOD_EXOT10 □ DAOD_EXOT13 □ DAOD_EXOT15 □
11□ DAOD_EXOT2 □ DAOD_EXOT3 □ DAOD_EXOT5 □ DAOD_EXOT7 □ DAOD_EXOT9 □ DAOD_EXOT11 □ DAOD_EXOT12 □ DAOD_EXOT14 □
12□ DAOD_EXOT16 □
13□ DAOD_TAUP1 □ DAOD_TAUP3 □
14□ DAOD_JETM3 □ DAOD_JETM10 □ DAOD_JETM11 □
15□ DAOD_JETM1 □ DAOD_JETM2 □ DAOD_JETM4 □ DAOD_JETM6 □ DAOD_JETM7 □ DAOD_JETM9 □

Tasks management



total	Reco	Rec Merge	Deriv	Deriv Merge
8	2	2	2	2
☑	☑	☑	☑	☑

total	active	good	running	registered	submitting
8	8	8	3	4	1
☑	☑	☑	☑	☑	☑

Select Filtered Deselect all

Show 10 entries

	Task Name	TaskID	Owner	ReqID	Status	(Current) Priority	Events	Failure %	Step
⊕	mc16_13TeV.425206.PowhegPythia8EvtGen_AZNLOC6L1_RenUp_Ztautau_l22h10_filter.deriv.e7321_e5984_s3126_r10201_r10210_p3384	17292459	gingrich	22946	registered	890	0		Deriv
⊕	mc16_13TeV.425206.PowhegPythia8EvtGen_AZNLOC6L1_RenUp_Ztautau_l22h10_filter.merge.e7321_e5984_s3126_r10201_r10210	17292452	gingrich	22946	running	890	900000	0	Rec Merge
⊕	mc16_13TeV.425206.PowhegPythia8EvtGen_AZNLOC6L1_RenUp_Ztautau_l22h10_filter.merge.e7321_e5984_s3126_r10201_r10210_p3384_p3385	17292465	gingrich	22946	registered	890	0		Deriv Merge
⊕	mc16_13TeV.425206.PowhegPythia8EvtGen_AZNLOC6L1_RenUp_Ztautau_l22h10_filter.recon.e7321_e5984_s3126_r10201	17292447	gingrich	22946	running	350	1300000	0	Reco
⊕	mc16_13TeV.425207.PowhegPythia8EvtGen_AZNLOC6L1_RenDown_Ztautau_l22h10_filter.deriv.e7321_e5984_s3126_r10201_r10210_p3384	17292431	gingrich	22946	registered	890	0		Deriv
⊕	mc16_13TeV.425207.PowhegPythia8EvtGen_AZNLOC6L1_RenDown_Ztautau_l22h10_filter.merge.e7321_e5984_s3126_r10201_r10210	17292428	gingrich	22946	submitting	890	100000	0	Rec Merge
⊕	mc16_13TeV.425207.PowhegPythia8EvtGen_AZNLOC6L1_RenDown_Ztautau_l22h10_filter.merge.e7321_e5984_s3126_r10201_r10210_p3384_p3385	17292439	gingrich	22946	registered	890	0		Deriv Merge
⊕	mc16_13TeV.425207.PowhegPythia8EvtGen_AZNLOC6L1_RenDown_Ztautau_l22h10_filter.recon.e7321_e5984_s3126_r10201	17292418	gingrich	22946	running	350	1386000	0	Reco
	Search Task Name	Search Tas	Search Ow	Search Re		Search (Cu	Search Eve	Search Fa	

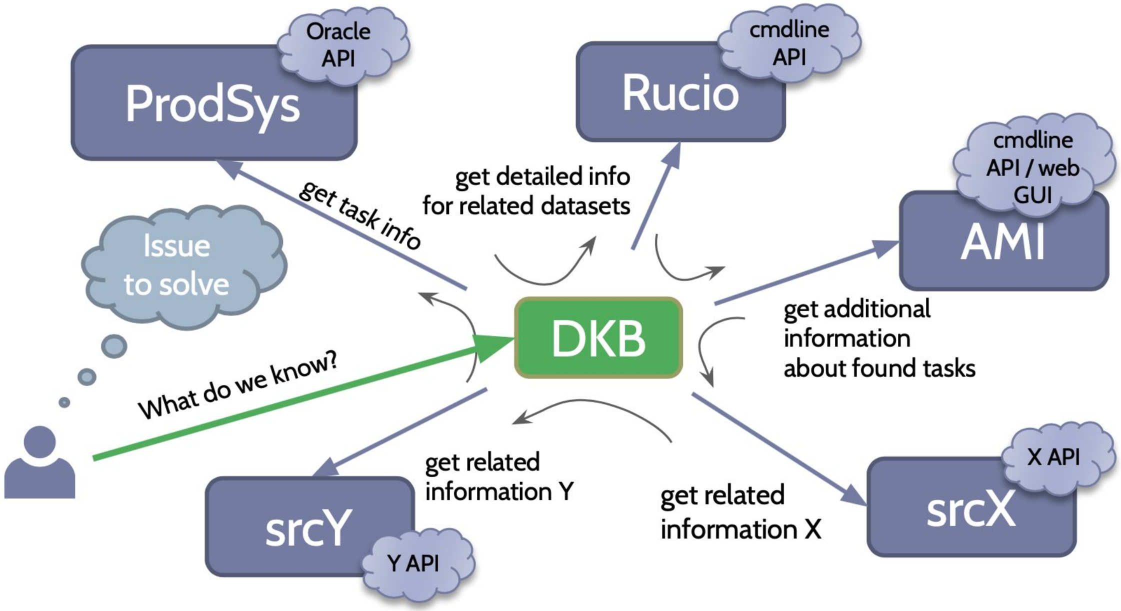
Showing 1 to 8 of 8 entries Previous 1 Next

Tasks selected: 0

Production request processing

- Task request Web UI provides many general and experiment specific features:
  - **Bookkeeping**. Storing metadata, including arbitrary hashtags, allows to provide fine tuning statistics for running and historical tasks.
  - **Approval 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 fail because of many permanent (e.g. bug in software) and temporal (storage is down) reasons. To be able to quickly understand the root of the problem and fix it by redefining the task is one of the major features of the production system.
  - **Chaining** one production to the other. E.g. derivation production could be chained to MC or reprocessing task, that significantly speeds them up.
  - **Automation** of task submission. User can define a pattern and when new data appears tasks are started automatically.
  - ...

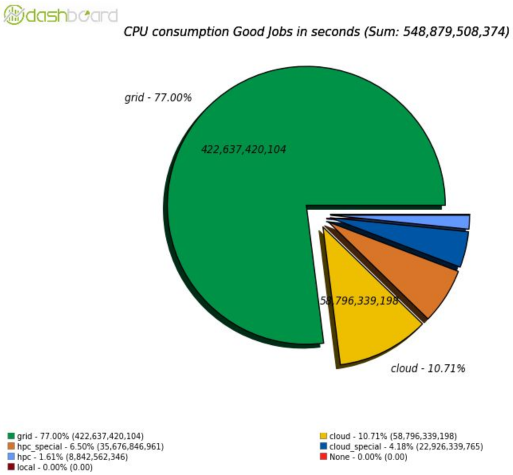
DKB - Data Knowledge Base


ProdSys
Oracle API
Rucio
cmdline API
AMI
cmdline API / web GUI
Issue to solve
What do we know?
DKB
get task info
get detailed info for related datasets
get additional information about found tasks
get related information Y
get related information X
srcY
Y API
srcX
X API

DKB is ElasticSearch based system. It is being developed to consolidate different metadata which are related to the ATLAS production system. It's useful for troubleshooting, statistics, workflows optimization.

Addressing future challenges

- ATLAS Distributed Computing was very successful in the last years with clouds, HPC and HTC integration and using opportunistic **computing resources** for the Monte Carlo production



- The HL-LHC era **data storage** estimated requirements are several times bigger than the present forecast of available resources, based on flat budget assumption
  - "Data Carousel" is a new project, which should allow orchestration between workload management, data management and storage services whereby a bulk production campaign with its inputs resident on a cheaper storage(e.g. tape), is executed by staging and promptly processing a sliding window of inputs

Conclusion

- Constantly increasing luminosity and always limited computing budget require to find ways for further efficient and economical use of traditional and new computing resources

- The ATLAS production system development and operation experience gained during LHC Run 2 creates an excellent base to face upcoming challenges