ATLAS Production System Top-level Layer to Manage Tasks

D. Golubkov and M. Borodin, K. De, J. Garcia, A. Klimentov, T. Maeno, A. Vaniachine

GRID2014, 1 July 2014

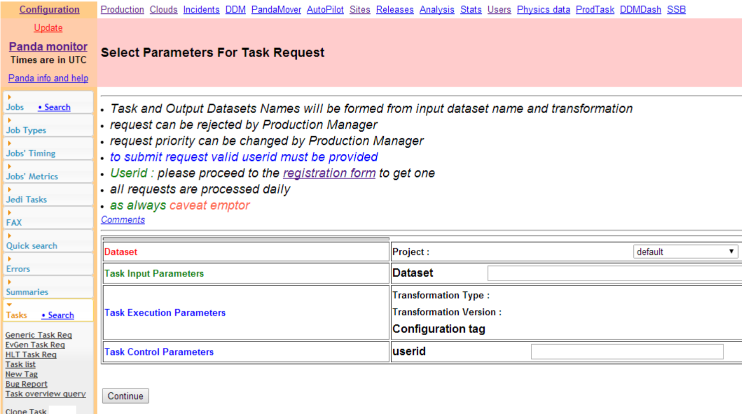
JINR, Dubna, Russia

Overview

- The ATLAS Production System (ProdSys)
  - automated system for running jobs in the PanDA
  - additional level of abstraction for PanDA
  - originated in 2006
  - handles O(1M) tasks per year
- The PanDA Production ANd Distributed Analysis system
  - data-driven workload management system for production and distributed analysis
  - about 50k jobs per day and 14k CPU wall-time hours per day for production at 100 sites (in 2009)

ATLAS Production System – ProdSys1

- Originated in 2006 and handles O(1M) tasks per year
- Task and job definition
- Task and tag request interfaces
- Limited task/job control interfaces

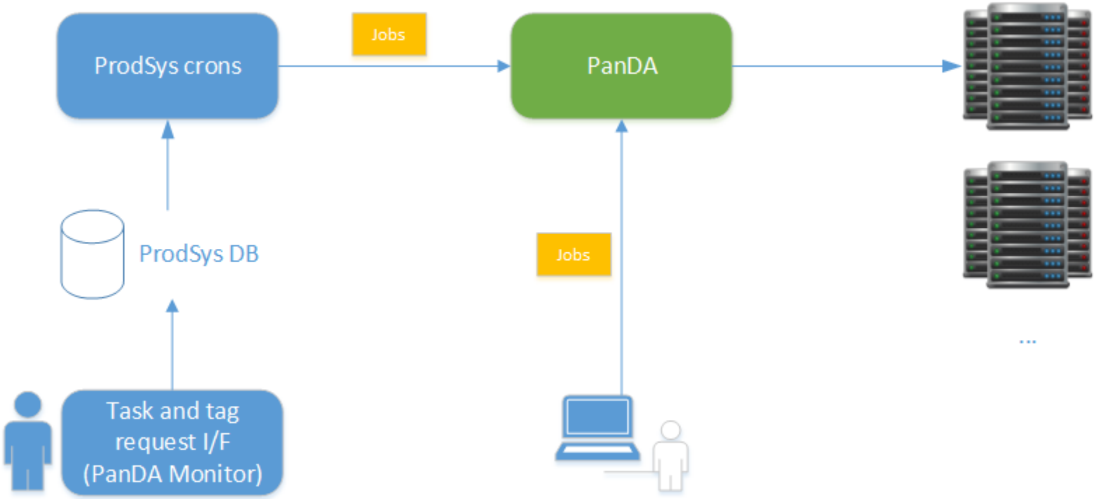


Select Parameters For Task Request

- *Task and Output Datasets Names will be formed from input dataset name and transformation*
- *request can be rejected by Production Manager*
- *request priority can be changed by Production Manager*
- *to submit request valid userid must be provided*
- *Userid : please proceed to the registration form to get one*
- *all requests are processed daily*
- *as always caveat emptor*

Comments

Dataset	Project : default ▾
Task Input Parameters	**Dataset**
Task Execution Parameters	Transformation Type : Transformation Version : **Configuration tag**
Task Control Parameters	**userid**

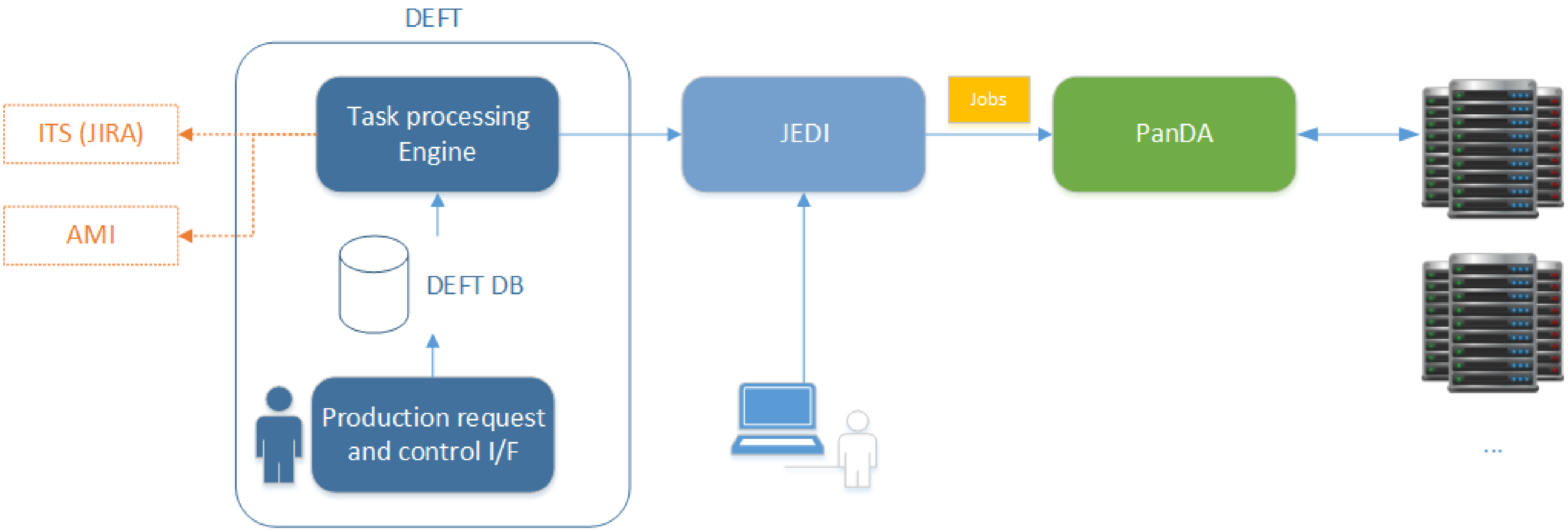


Evolution

- The current LHC shutdown provides an opportunity to rethink the architecture of ProdSys
- Exponential growth of the task submission rate
- Growing number of users and support requests
- New requirements from ATLAS main areas: Physics, Trigger, Data Preparation and Software & Computing
- The main features which should be fully revised and improved
    - Scalability and maintainability
    - Flexibility of job definition
    - Ease of use

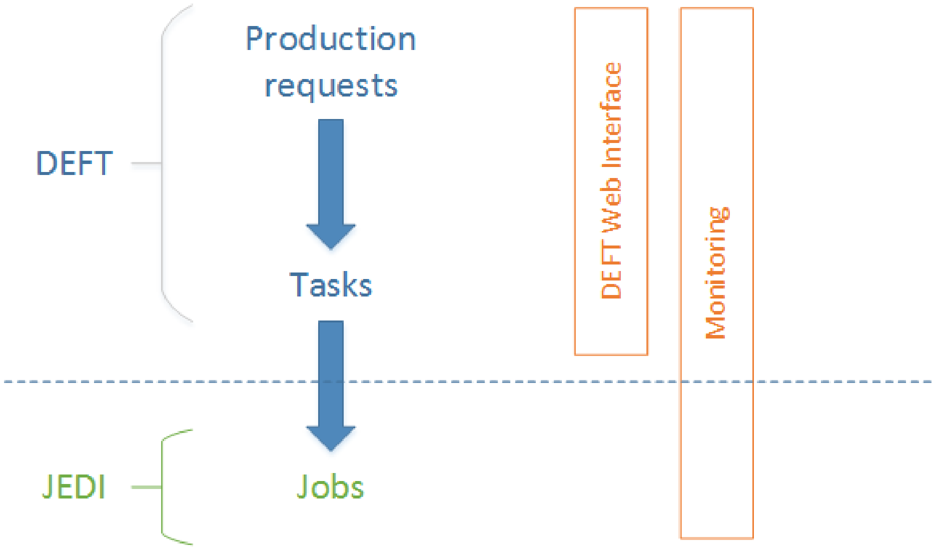
Next generation of ATLAS Production System – ProdSys2

- Developing started in the end of 2013
- Multilevel workflow management system
- Different levels of data abstraction
- DEFT (Database Engine for Tasks) and JEDI (Job Execution and Definition Interface)



DEFT

- Top-level layer of ProdSys2
- Defines the tasks complete with all necessary parameters and transfers these tasks to JEDI
- Implements the tasks handling
- Provides unified access to the user to run different types of tasks

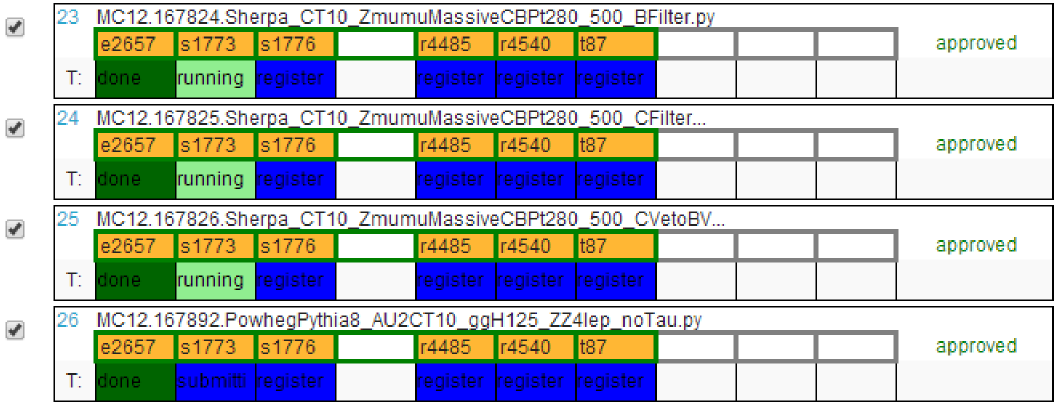


Design

- MVC (Model-view-controller pattern)
- Abstraction levels: request level, step level, task level
- Core components
  - Authentication and authorization subsystem
  - Web interface
  - Task processing engine
  - Post-production crons and database
- Integration with Issue Tracking system (ITS) and ATLAS Metadata Interface (AMI)

Web interface

- CERN Single Sign-On service and VOMS integration
- Each request form for each processing type based on same core:
    - MC Production request
    - DPD Production request
    - Reprocessing request
    - High-Level Trigger request
- Task manipulation: abort, clone, finish, change priority, reassign and etc.
- Different access levels



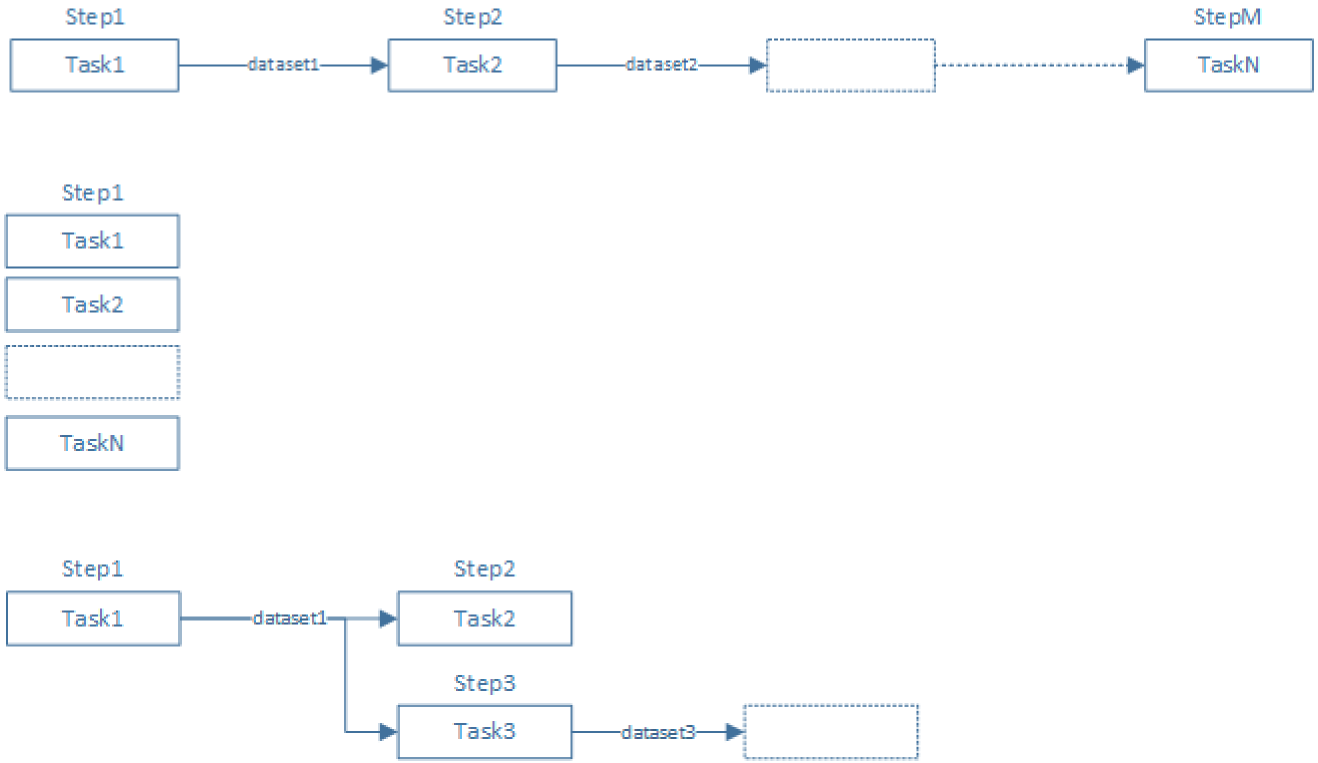
☑	23	MC12.167824.Sherpa_CT10_ZmumuMassiveCBPt280_500_BFilter.py										
		e2657	s1773	s1776		r4485	r4540	t87				approved
	T:	done	running	register		register	register	register				
☑	24	MC12.167825.Sherpa_CT10_ZmumuMassiveCBPt280_500_CFilter...										
		e2657	s1773	s1776		r4485	r4540	t87				approved
	T:	done	running	register		register	register	register				
☑	25	MC12.167826.Sherpa_CT10_ZmumuMassiveCBPt280_500_CVetoBV...										
		e2657	s1773	s1776		r4485	r4540	t87				approved
	T:	done	running	register		register	register	register				
☑	26	MC12.167892.PowhegPythia8_AU2CT10_ggH125_ZZ4lep_noTau.py										
		e2657	s1773	s1776		r4485	r4540	t87				approved
	T:	done	submitti	register		register	register	register				

Task Processing engine

- Translates production requests to tasks
- Django based, REST API
- Prepares input/output data list (using ProdSys1 database and DQ2)
- Gets transform parameters (from AMI or ProdSys1 database)
- Constructs the task name and generates job parameters structure and passes it to JEDI using JSON protocol
- Registers the task output in the database
- Logging of all user actions in ITS and database

Common task workflow types

- DEFT supports several types of task workflows including a chain of tasks and independent single tasks



Conclusions

- Most requirements are implemented
- Support for main processing types and task workflows
- Preparing for stress-testing and running in production mode

References

- https://twiki.cern.ch/twiki/bin/view/AtlasComputing/PanDA
- https://twiki.cern.ch/twiki/bin/viewauth/AtlasComputing/ProdSys