

# PanDA for COMPASS: processing data via Grid

Artem Petrosyan (JINR)

July 7, 2016

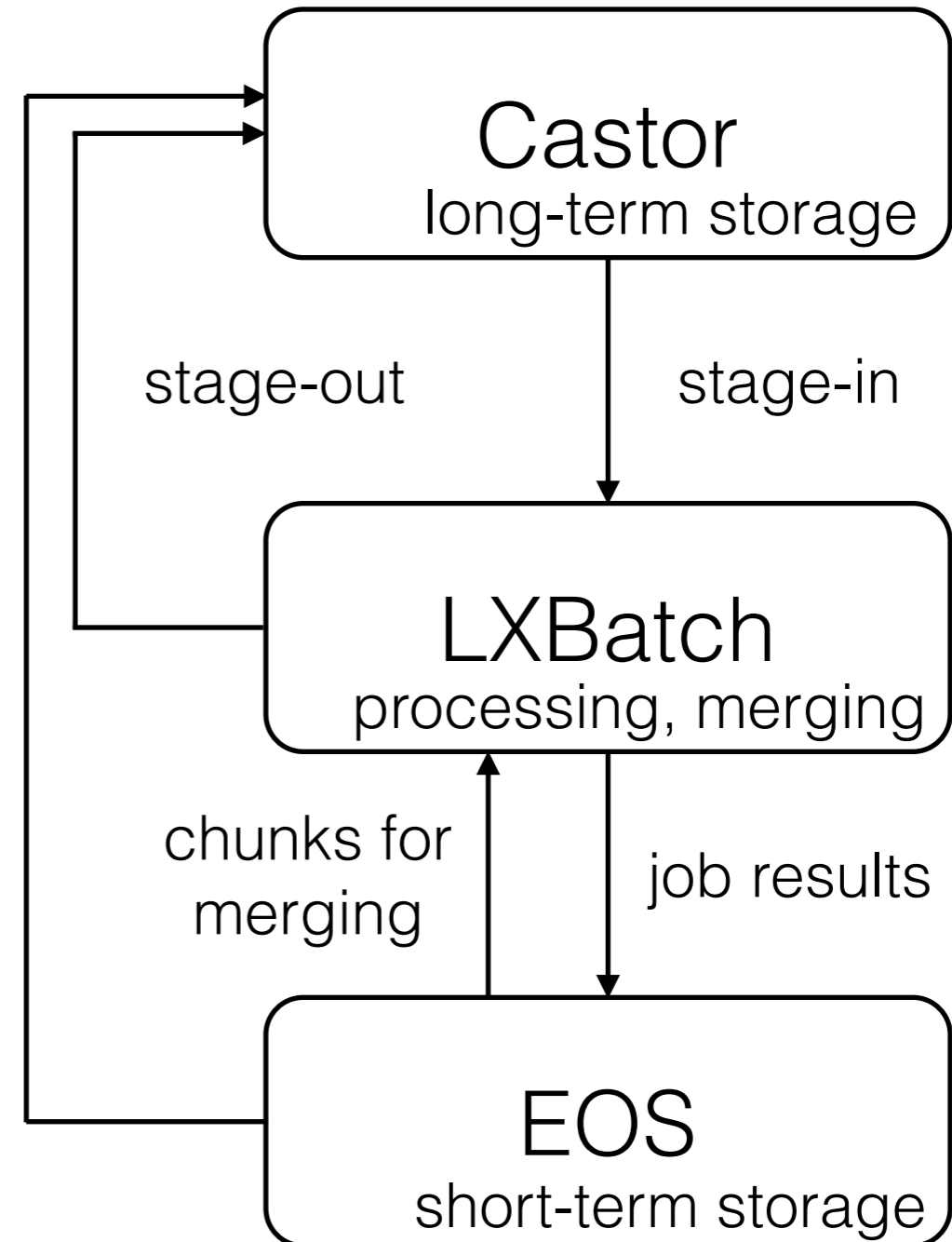
GRID`2016, LIT, JINR, Dubna, Russia

# What is COMPASS

- **C**Ommon **M**uon **P**roton **A**pparatus for **S**tructure and **S**pectroscopy (COMPASS) is a high-energy physics experiment at a Super Proton Synchrotron (SPS) at CERN
- The purpose of the experiment is the study of hadron structure and hadron spectroscopy with high intensity muon and hadron beams
- First data taking run started in summer 2002 and sessions are continue
- Each data taking session containing from 1.5 to 3 PB of data
- More than 200 physicists from 13 countries and 24 institutes are the analysis user community of COMPASS

# COMPASS production dataflow

- All data stored on Castor
- Data is being requested to be copied from tapes to disks before processing (may take ~6 hours)
- Task moves files directly from Castor to lxbatch for processing, several programs are used for processing
- After processing results are being transferred to EOS for merging or short-term storage or directly to Castor for long-term storage
- Merging
- Results are being copied to Castor for long-term storage



# Motivation items

- Data management is done by a set of scripts, deployed under production account on AFS
- Execution of user analysis jobs and production jobs are separated and managed by different sets of software
- Number of jobs which can be executed by the collaboration at lxbatch is limited
- Available space on home of COMPASS' production user at lxplus is limited and strictly managed
- Although COMPASS data flow has conditions to have distributed computing, it is implemented as single-site processing which uses only one computing facility
- Absence of monitoring does not allow to see how users work with data

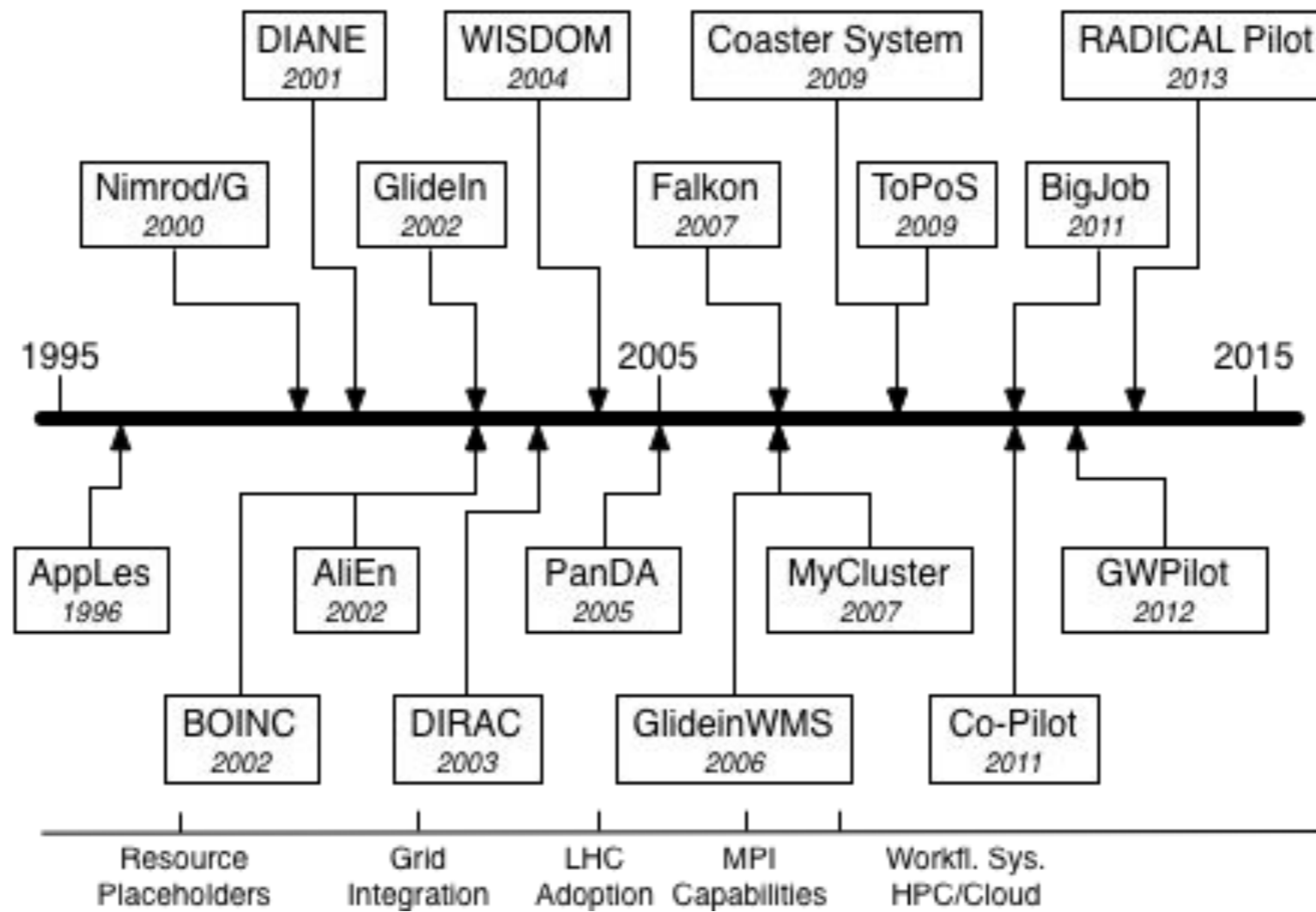
703638624	na58dst	PEND	2nw	lxplus0048.	*538429.28	Sep 29 16:53
703638628	na58dst	PEND	2nw	lxplus0048.	*538429.68	Sep 29 16:53
703638631	na58dst	PEND	2nw	lxplus0048.	*3538430.1	Sep 29 16:53
703638635	na58dst	PEND	2nw	lxplus0048.	*3538430.5	Sep 29 16:53
703638638	na58dst	PEND	2nw	lxplus0048.	*538430.91	Sep 29 16:53
703638641	na58dst	PEND	2nw	lxplus0048.	*538431.32	Sep 29 16:53
703638644	na58dst	PEND	2nw	lxplus0048.	*538431.74	Sep 29 16:53
703638647	na58dst	PEND	2nw	lxplus0048.	*538432.29	Sep 29 16:53
703638650	na58dst	PEND	2nw	lxplus0048.	*538432.75	Sep 29 16:53
703638652	na58dst	PEND	2nw	lxplus0048.	*538433.19	Sep 29 16:53
703638654	na58dst	PEND	2nw	lxplus0048.	*538433.59	Sep 29 16:53
703638660	na58dst	PEND	2nw	lxplus0048.	*3538434.0	Sep 29 16:53
703638663	na58dst	PEND	2nw	lxplus0048.	*3538434.4	Sep 29 16:53
703638667	na58dst	PEND	2nw	lxplus0048.	*538434.79	Sep 29 16:53
703638673	na58dst	PEND	2nw	lxplus0048.	*3538435.2	Sep 29 16:53
703638677	na58dst	PEND	2nw	lxplus0048.	*538435.67	Sep 29 16:53
703638683	na58dst	PEND	2nw	lxplus0048.	*538436.25	Sep 29 16:53
703638686	na58dst	PEND	2nw	lxplus0048.	*538436.66	Sep 29 16:53
703638693	na58dst	PEND	2nw	lxplus0048.	*538437.09	Sep 29 16:53
703638702	na58dst	PEND	2nw	lxplus0048.	*538437.67	Sep 29 16:53

**We need a WMS!**

# What is WMS?

- WMS — workload management system
- Providing a central queue for all users, makes hundreds of distributed sites appear as local
- Hide middleware while supporting diversity and evolution
  - WMS interacts with middleware, users see only high level workflow
  - Automation engines built in WMS, not exposed to users
- Hide variations in infrastructure
  - WMS presents uniform 'job' slots to user
  - Easy to integrate grid sites, clouds, HPC sites
- Use the same system for simulation, data processing and users analysis
- Similar ideas have been implemented in several independent systems developed by LHC experiments: AliEn, Dirac, PanDA

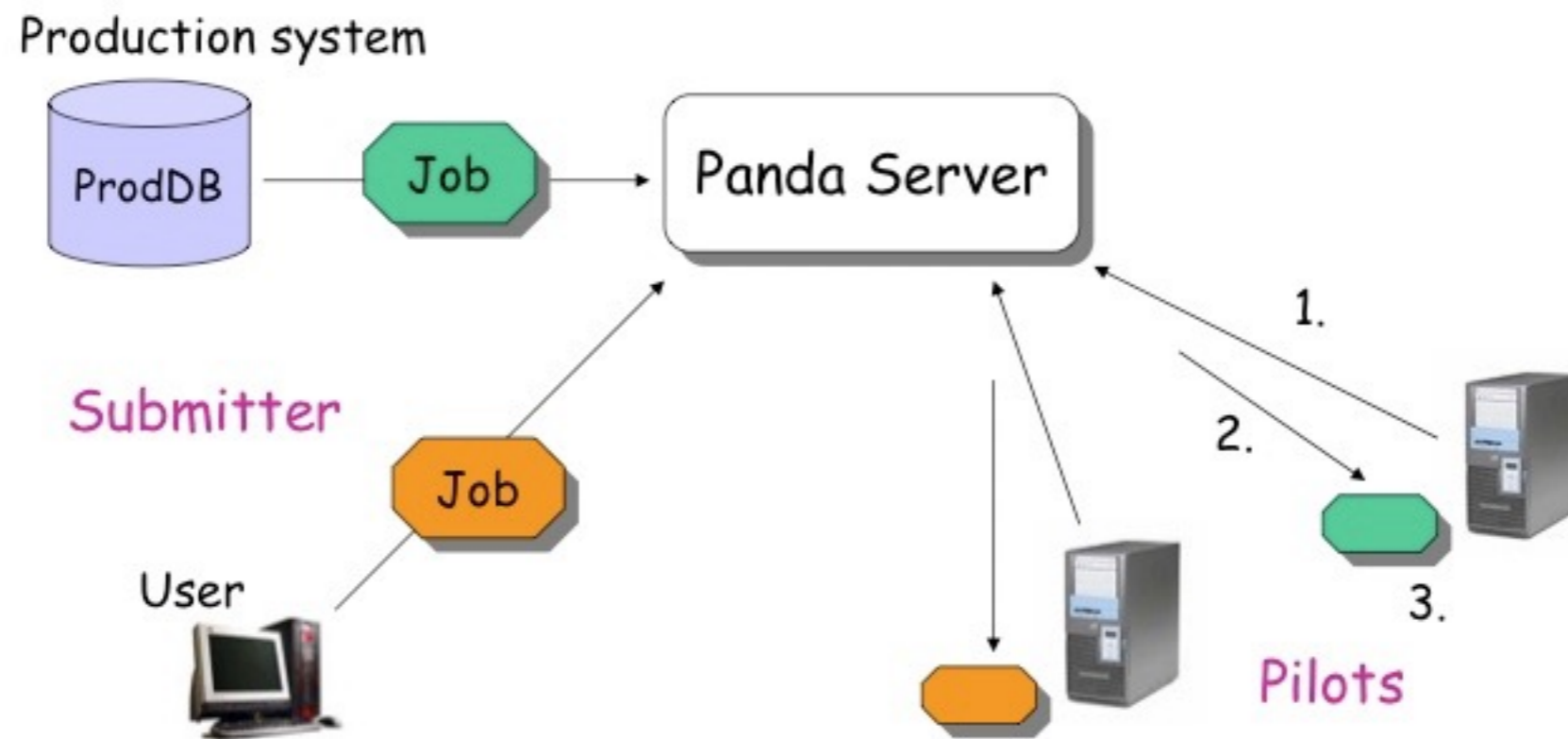
# WMS timeline



# What is PanDA?

- The PanDA **P**roduction **an**d **D**istributed **A**nalysis System has been developed by ATLAS to meet requirements of data-driven workload management system for production and distributed analysis processing capable at LHC data processing scale
- PanDA manages both user analysis and production jobs via same interface
- PanDA processing rate is 250-300K jobs on ~170 sites every day
- The PanDA ATLAS analysis user community numbers over 1400

# PanDA job workflow



Each pilot runs on a worker node

1. send a request
2. receives a job
3. runs the job



# Action items

- PanDA instance installation
- Production chain management software
- Grid environment setup
- Validation of COMPASS software in Grid
- Production jobs execution by PanDA expert
- Physics validation
- Production by COMPASS production manager

# PanDA instance at JINR

- Goal: evaluate the possibility of COMPASS jobs to be executed via PanDA server installed at JINR on local cloud service
- Steps were performed:
  - PanDA server was deployed on a cloud instance which was provided by LIT JINR cloud service
  - Several PanDA queues were defined
  - Several users were registered from both LIT and COMPASS sides
  - To define COMPASS-specific logic, several extensions were implemented in PanDA Pilot:
    - COMPASSExperiment.py
    - COMPASSSite.py

# Production chain management software

- We needed to redefine job submission and task management for PanDA
- COMPASS data has a good files naming convention
  - Each filename defines its year, run, chunk
- Combining file name with production parameters it's quite easy to define any job

```
$ python panda_compass_prod_coral.py CERN_COMPASS_PROD /castor/cern.ch/compass/data/2014/raw/T05/cdr11002-254440.raw 0 7 /  
afs/cern.ch/compass/scratch/d17/objsrvvy/ generalprod/testcoral/dy14T05t4PANDA
```

- Executing this command is cycle for whole production files set allows to define a task

```
TMPDSTFILE=mDST-254442-13026-0-7.root;export TMPHISTFILE=hist_254442-13026.root;export  
TMPRICHFILE=gfile_254442-13026.gfile;xrdcp -np root://castorpublic.cern.ch//castor/cern.ch/compass/data/2014/raw/T05/  
cdr13026-254442.raw\?svcClass=compassuser .;coralpath=/afs/cern.ch/compass/scratch/d17/objsrvvy/generalprod/testcoral/  
dy14T05t4PANDA/coral;echo $coralpath;cd -P $coralpath;export coralpathsetup=$coralpath"/setup.sh";echo  
$coralpathsetup;source $coralpathsetup;cd $ppwd;$CORAL/./phast/coral/coral.exe /afs/cern.ch/compass/scratch/d17/objsrvvy/  
generalprod/testcoral/dy14T05t4PANDA/template.opt;rm cdr13026-254442.raw;xrdcp -np $ppwd/mDST-254442-13026-0-7.root  
xroot://eoscompass.cern.ch//eos/compass/generalprod/testcoral/dy14T05t4PANDA/mDST/mDST-254442-13026-0-7.root;xrdcp -np  
$ppwd/hist_254442-13026.root xroot://eoscompass.cern.ch//eos/compass/generalprod/testcoral/dy14T05t4PANDA/histos/  
hist_254442-13026.root;xrdcp -np $ppwd/testevtdump.raw xroot://eoscompass.cern.ch//eos/compass/generalprod/testcoral/  
dy14T05t4PANDA/evtdump/slot0/evtdump0-254442-13026.raw
```

# Grid environment

- AFS COMPASS group
  - Production account
- Local batch queue
- EOS directory
- AFS directory to deploy production software
- Virtual organization
  - Production role
- Computing element
- EOS storage element
- CVMFS

# Test production

<b>computingsite</b>	CERN_COMPASS_PROD (54)	<h2>Queue summary, running and recently finished jobs</h2>
<b>destinationse</b>	local (54)	
<b>jobstatus</b>	finished (30) holding (1) running (23)	
<b>prodsourcelabel</b>	prod_test (54)	
<b>produsername</b>	Artem Petrosyan (54)	
<b>transformation</b>	export TMPRAWFILE=cdr11008-254440.raw; (1) export TMPRAWFILE=cdr11011-254440.raw; (1) export TMPRAWFILE=cdr11015-254440.raw; (1) export TMPRAWFILE=cdr11021-254440.raw; (1) export TMPRAWFILE=cdr11022-254440.raw; (1) export TMPRAWFILE=cdr11029-254440.raw; (1) export TMPRAWFILE=cdr11032-254440.raw; (1) export TMPRAWFILE=cdr11033-254440.raw; (1) export TMPRAWFILE=cdr11034-254440.raw; (1) export TMPRAWFILE=cdr11043-254440.raw; (1) export TMPRAWFILE=cdr11045-254440.raw; (1) export TMPRAWFILE=cdr12002-254440.raw; (1) export TMPRAWFILE=cdr12007-254440.raw; (1) export TMPRAWFILE=cdr12013-254440.raw; (1) export TMPRAWFILE=cdr12015-254440.raw; (1) export TMPRAWFILE=cdr12016-254440.raw; (1) export TMPRAWFILE=cdr12023-254440.raw; (1) export TMPRAWFILE=cdr12024-254440.raw; (1) export TMPRAWFILE=cdr12025-254440.raw; (1) export TMPRAWFILE=cdr12028-254440.raw; (1) export TMPRAWFILE=cdr12030-254440.raw; (1) export TMPRAWFILE=cdr12034-254440.raw; (1) export TMPRAWFILE=cdr12038-254440.raw; (1) export TMPRAWFILE=cdr13002-254440.raw; (1) export TMPRAWFILE=cdr13004-254440.raw; (1) export TMPRAWFILE=cdr13008-254440.raw; (1) export TMPRAWFILE=cdr13009-254440.raw; (1) export TMPRAWFILE=cdr13016-254440.raw; (1) export TMPRAWFILE=cdr13017-254440.raw; (1) export TMPRAWFILE=cdr13018-254440.raw; (1) export TMPRAWFILE=cdr13019-254440.raw; (1) export TMPRAWFILE=cdr13020-254440.raw; (3) export TMPRAWFILE=cdr13022-254440.raw; (1) export TMPRAWFILE=cdr13025-254440.raw; (1) export TMPRAWFILE=cdr13029-254440.raw; (1) export TMPRAWFILE=cdr13036-254440.raw; (1) export TMPRAWFILE=cdr14001-254440.raw; (1) export TMPRAWFILE=cdr14004-254440.raw; (1) export TMPRAWFILE=cdr14008-254440.raw; (1) export TMPRAWFILE=cdr14010-254440.raw; (1) export TMPRAWFILE=cdr14012-254440.raw; (1) export TMPRAWFILE=cdr14013-254440.raw; (1) export TMPRAWFILE=cdr14014-254440.raw; (1) export TMPRAWFILE=cdr14017-254440.raw; (1) export TMPRAWFILE=cdr14019-254440.raw; (1) export TMPRAWFILE=cdr14022-254440.raw; (1) export TMPRAWFILE=cdr14025-254440.raw; (1) export TMPRAWFILE=cdr14026-254440.raw; (1) export TMPRAWFILE=cdr14027-254440.raw; (1) export TMPRAWFILE=cdr14029-254440.raw; (1) export TMPRAWFILE=cdr14040-254440.raw; (1) export TMPRAWFILE=cdr14043-254440.raw; (1)	
<b>vo</b>	vo.compass.cern. (54)	

Owner / VO	Task ID	PanDA ID	Transformation	Status	Created	Start	End	Site	Priority	Job info
Artem Petrosyan / vo.compass.cern.	0	4625	export TMPRAWFILE=cdr14012-254440.raw;	finished	2016-07-05 15:13	07-05 17:19	07-06 00:38	CERN_COMPASS_PROD	1000	
Artem Petrosyan / vo.compass.cern.	0	4624	export TMPRAWFILE=cdr14019-254440.raw;	running	2016-07-05 15:13	07-05 17:19		CERN_COMPASS_PROD	1000	
Artem Petrosyan / vo.compass.cern.	0	4623	export TMPRAWFILE=cdr13002-254440.raw;	running	2016-07-05 15:13	07-05 17:17		CERN_COMPASS_PROD	1000	
Artem Petrosyan / vo.compass.cern.	0	4622	export TMPRAWFILE=cdr14008-254440.raw;	finished	2016-07-05 15:13	07-05 17:14	07-05 23:28	CERN_COMPASS_PROD	1000	
Artem Petrosyan / vo.compass.cern.	0	4621	export TMPRAWFILE=cdr12028-254440.raw;	running	2016-07-05 15:13	07-05 17:14		CERN_COMPASS_PROD	1000	
Artem Petrosyan / vo.compass.cern.	0	4620	export TMPRAWFILE=cdr13036-254440.raw;	holding	2016-07-05 15:13	07-05 17:13	07-06 04:15	CERN_COMPASS_PROD	1000	

# Summary

- Grid environment has prepared for COMPASS
  - We're now ready to send jobs to any Grid site after software validation and enabling VO on a site
- COMPASS production jobs are being managed by PanDA in Grid environment
  - All PanDA components behave well
- Bunch of infrastructure work was performed during evaluation
- New production management was prepared: we can submit thousands jobs by one command
- More than 5000 jobs were executed
- PanDA server deployed on JINR cloud service handles the load well
  - Instance running on a single virtual machine shows impressive productivity and reliability

# Bright future :)

- Production system?
  - Production management environment which would cover all possible combinations of software, data files and execution parameters is a natural need
- More Grid sites?
  - JINR T2 ready to receive COMPASS jobs
- HPC?
  - Work on enabling 13.5 petaFlops Blue Waters facility has started
- DDM?
  - Evaluation of ATLAS Rucio distributed management system is being discussed

# Links

- COMPASS home:

<http://wwwcompass.cern.ch/>

- PanDA home:

<https://twiki.cern.ch/twiki/bin/view/PanDA/PanDA>

- Monitoring link to COMPASS PanDA queue:

[http://vm127.jinr.ru/bigpandamon/jobjobs/?computingsite=CERN\\_COMPASS\\_PROD](http://vm127.jinr.ru/bigpandamon/jobjobs/?computingsite=CERN_COMPASS_PROD)



# Acknowledgements

- COMPASS team
  - Elena Zemlyanichkina, Sergei Gerassimov, Vladimir Frolov
- PanDA team
  - Alexei Klimentov, Fernando Harald Barreiro Megino, Danila Oleynik
- RCKI
  - Ruslan Mashinistov
- JINR team
  - Nikolay Kutovskiy
- Rutgers University
  - Shantenu Jha

Thanks!