COMPASS PanDA test instance at JINR

Artem Petrosyan (JINR)

October 2, 2015

NEC`2015, Budva, Bečići, Montenegro

What is COMPASS

- COmmon Muon Proton Apparatus for Structure and Spectroscopy (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

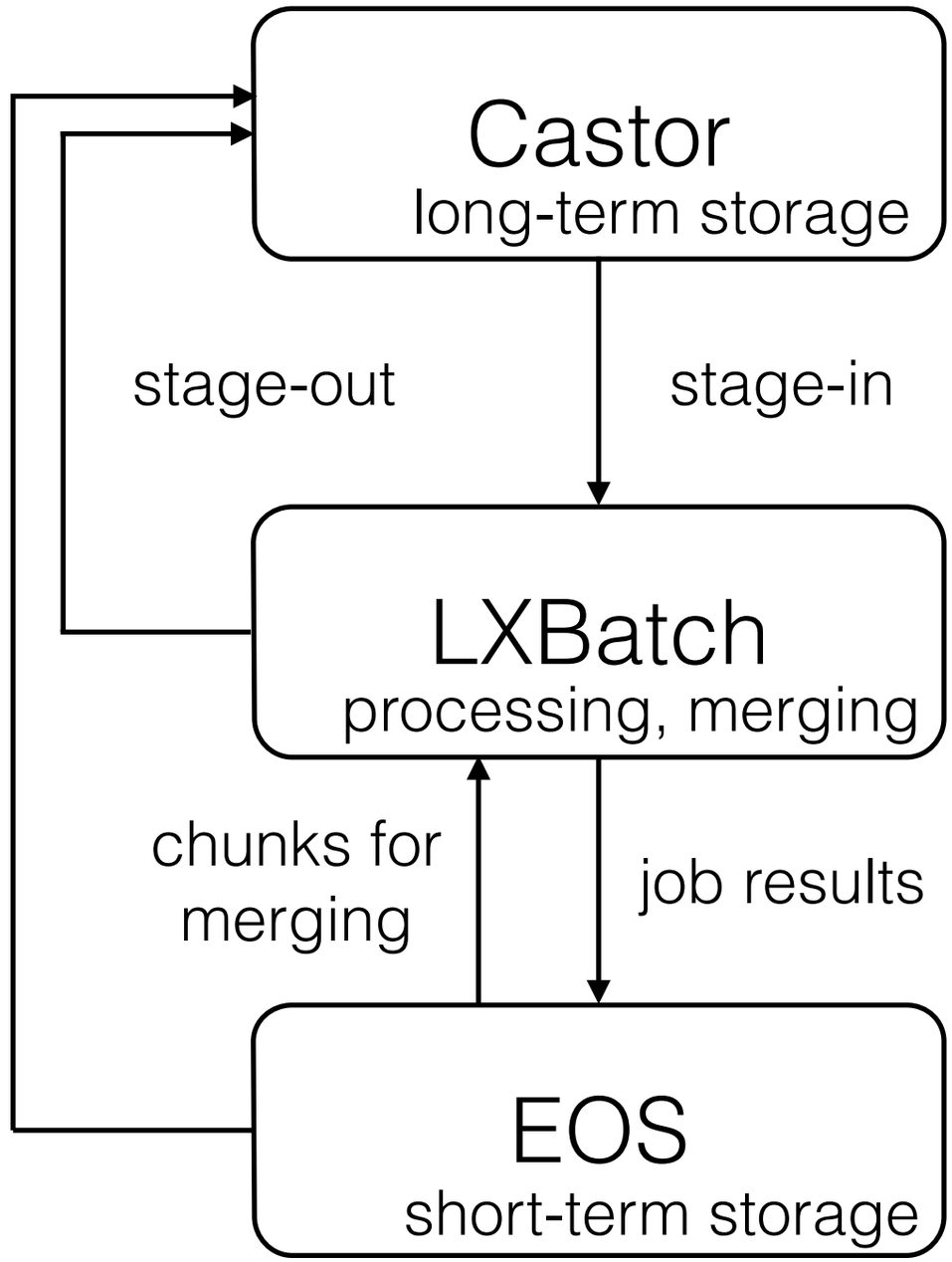
- First data taking run started in summer 2002 and sessions are continue

- 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



COMPASS dataflow limitations

- Data management is done by a set of scripts, deployed under production account on AFS, which are not organised as a common system
- 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 and Castor is limited and strictly managed
- Castor is a storage system which stores data on tapes and definitely not designed for random access reading from many users simultaneously
- 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
703638705 na58dst PEND  2nw        lxplus0048.             *538438.72 Sep 29 16:53
703638707 na58dst PEND  2nw        lxplus0048.             *538439.13 Sep 29 16:54
703638711 na58dst PEND  2nw        lxplus0048.             *538440.46 Sep 29 16:54
703641861 na58dst PEND  2nw        lxplus0050.             *539036.46 Sep 29 17:03
703641867 na58dst PEND  2nw        lxplus0050.             *539036.84 Sep 29 17:03
703641870 na58dst PEND  2nw        lxplus0050.             *3539037.2 Sep 29 17:03
703641872 na58dst PEND  2nw        lxplus0050.             *539037.59 Sep 29 17:03
703641875 na58dst PEND  2nw        lxplus0050.             *539037.96 Sep 29 17:03
703641878 na58dst PEND  2nw        lxplus0050.             *539038.32 Sep 29 17:03
703641880 na58dst PEND  2nw        lxplus0050.             *539038.67 Sep 29 17:03
703641885 na58dst PEND  2nw        lxplus0050.             *539039.03 Sep 29 17:03
703641889 na58dst PEND  2nw        lxplus0050.             *539039.38 Sep 29 17:03
703641897 na58dst PEND  2nw        lxplus0050.             *539039.74 Sep 29 17:03
703641903 na58dst PEND  2nw        lxplus0050.             *3539040.1 Sep 29 17:04
703641907 na58dst PEND  2nw        lxplus0050.             *539040.48 Sep 29 17:04
703641917 na58dst PEND  2nw        lxplus0050.             *539042.35 Sep 29 17:04
703641919 na58dst PEND  2nw        lxplus0050.             *3539042.7 Sep 29 17:04
703641921 na58dst PEND  2nw        lxplus0050.             *539043.06 Sep 29 17:04
703641923 na58dst PEND  2nw        lxplus0050.             *539043.51 Sep 29 17:04
[na58dst1@lxplus0001 ~]$ bjobs -q 8nh
JOBID     USER    STAT  QUEUE      FROM_HOST   EXEC_HOST   JOB_NAME   SUBMIT_TIME
703516433 na58dst RUN   8nh        lxplus0058. b6ec28b2eb  bsub.sh    Sep 29 10:06
703627778 na58dst RUN   8nh        lxplus0064. b619034a91  bsub.sh    Sep 29 16:00
703627790 na58dst RUN   8nh        lxplus0064. b6e496854d  bsub.sh    Sep 29 16:01
703627803 na58dst RUN   8nh        lxplus0064. p05795827n9 bsub.sh    Sep 29 16:01
703627811 na58dst RUN   8nh        lxplus0064. b6bd2898ae  bsub.sh    Sep 29 16:01
703627818 na58dst RUN   8nh        lxplus0064. lxbsu2603   bsub.sh    Sep 29 16:01
[na58dst1@lxplus0001 ~]$
```

What is PanDA?

- The PanDA Production and Distributed Analysis 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

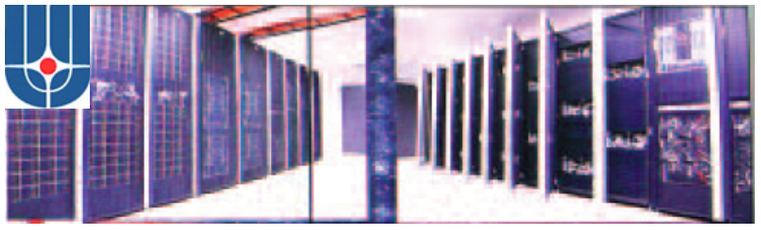
- PanDA processing rate is 250-300K jobs on ~170 sites every day

- The PanDA ATLAS analysis user community numbers over 1400

Resources supported by PanDA

WLCG
Worldwide LHC Computing Grid

Many Others



Beyond ATLAS and LHC Grid

- At 2012 the BigPanDA project was started
- Main goal of the project is to bring PanDA outside ATLAS
- The efforts during the project were concentrated on making PanDA modules distributable, adopted to work with several backends, on HPCs, on clouds (Amazon, Google, etc.), allow several virtual organisations to use one physical instance, better use of network infrastructure and better monitoring capabilities

Growing PanDA ecosystem

COMPASS

Google Cloud Platform

Alpha Magnetic Spectrometer AMS-02 Europe • Asia North America

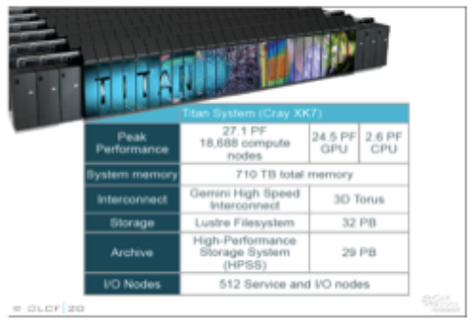
ALICE

ATLAS

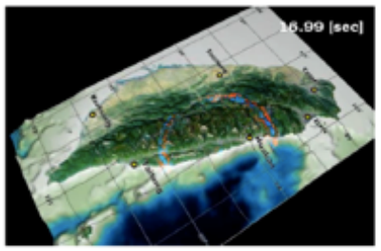
NICA

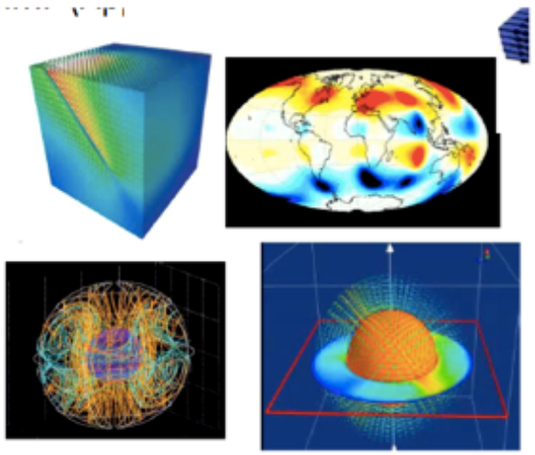
nectar

amazon web services™ EC2

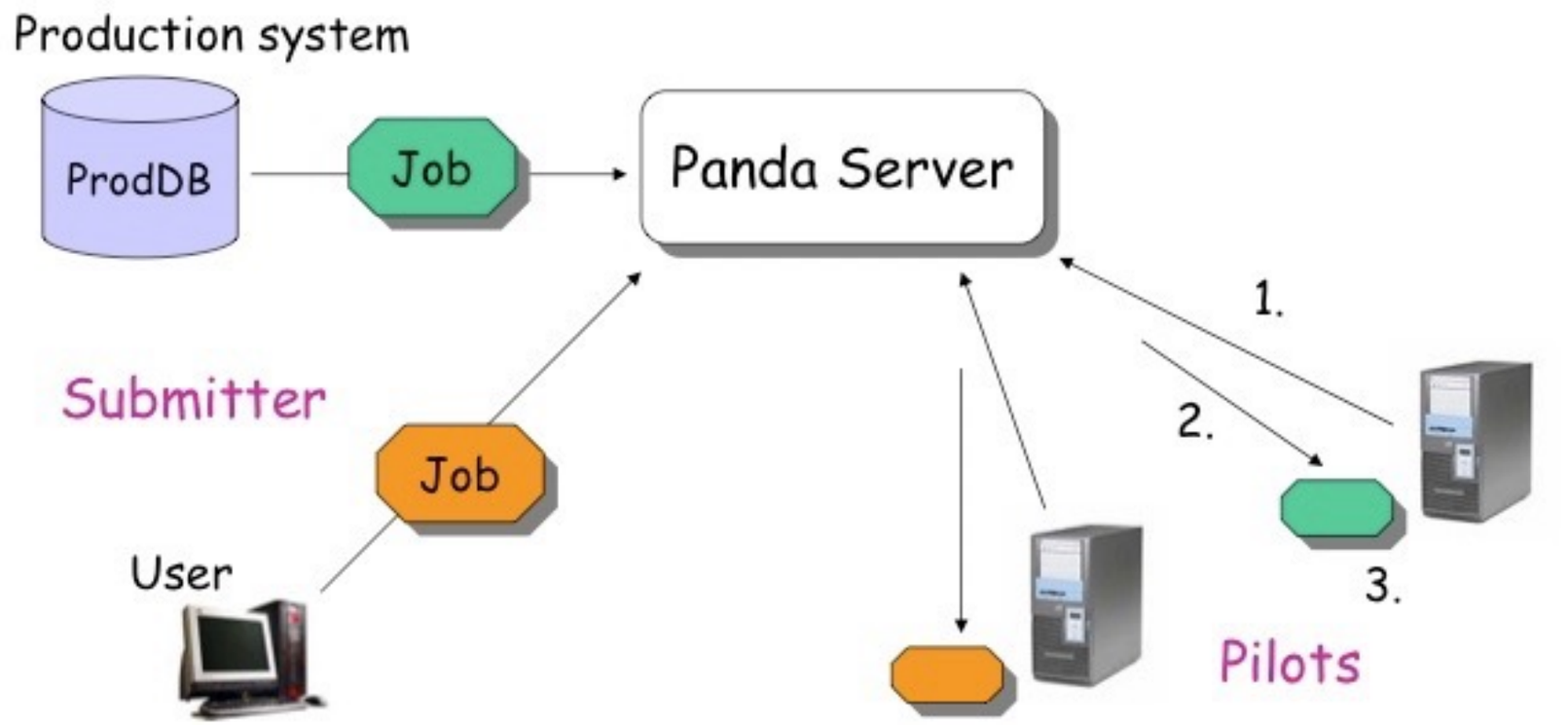


LSST Large Synoptic Survey Telescope





PanDA job workflow



Each pilot runs on a worker node
1. send a request
2. receives a job
3. runs the job

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 infrastructure 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

PanDA queue for COMPASS setup at CERN

- All production workload is done via COMPASS production use
- Steps were performed:
  - Space to store output was allocated
  - COMPASS analysis software (PHAST) was deployed, environment setup was adapted to work under bash
  - Robot certificate was requested and installed for production user account
  - Wrapper to send jobs to lxbatch was prepared
  - Set of root files to be defined and sent as a PanDA jobs was prepared

COMPASS job definition for PanDA

- For user job definition looks the same as before

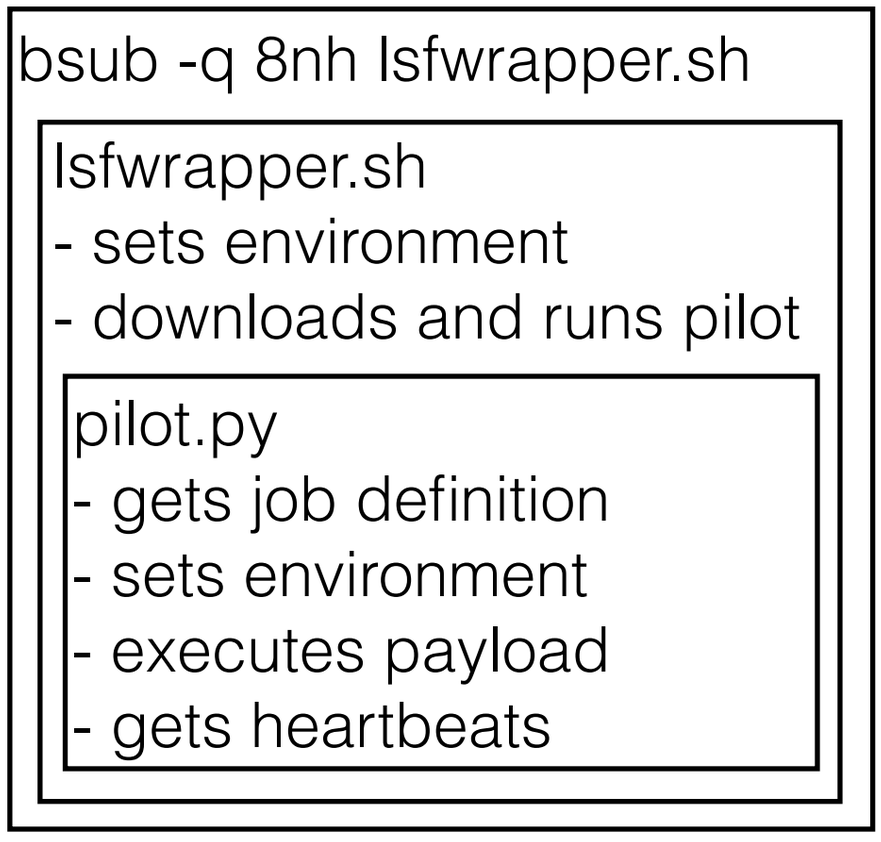
```
job = JobSpec()
job.jobDefinitionID   = 0
job.jobName = "lxplus_test_job" # - any string
#job.jobName           = "%s" % commands.getoutput('uuidgen')
job.transformation    = 'source /afs/cern.ch/work/n/na58dst1/panda/phast.7.148/misc/setup.lxplus_slc6_64.sh;/afs/cern.ch/work/n/na58dst1/panda/phast.7.148/phast
'  # payload (can be URL as well)
job.destinationDBlock = datasetName
job.destinationSE     = destName
job.currentPriority   = 1000
#SP job.prodSourceLabel   = 'test'
job.prodSourceLabel   = 'panda'
job.computingSite     = site
```

- Job submission

```
-bash-4.1$ python getntomlinesandsendjob.py ANALY_CERN_COMPASS_PROD 116 1000
number of lines: 3652
queue: ANALY_CERN_COMPASS_PROD
from: 116
amount: 1000
/castor/cern.ch/compass/generalprod/testcoral/hadron2009t86/megaDST/megaDST-80616-0-7.root.022

0
PandaID=280
/castor/cern.ch/compass/generalprod/testcoral/hadron2009t86/megaDST/megaDST-80616-0-7.root.023

0
PandaID=281
/castor/cern.ch/compass/generalprod/testcoral/hadron2009t86/megaDST/megaDST-80616-0-7.root.024

0
PandaID=282
/castor/cern.ch/compass/generalprod/testcoral/hadron2009t86/megaDST/megaDST-80616-0-7.root.025

0
PandaID=283
```

Running COMPASS payload via PanDA

- Job submission done by a registered PanDA user
- All execution made via COMPASS' production account
- Manual wrapper script submission from lxplus to lxbatch
- Wrapper sets environment, downloads pilot from PanDA server, unpacks and executes
- Pilot runs as usual
- Data is staged-out to a defined in job directory on lxplus



Monitoring 1/3

- PanDA comes with a monitoring, implemented as a Python-based application which uses Django framework and adopted to be set on Apache with Oracle and MySQL RDBMS

- Monitoring uses the same database schema where PanDA lives

- Execution details on each job are shown at monitoring pages

Monitoring 2/3

computingsite	ANALY_CERN_COMPASS_PROD (17)
destinationse	local (17)
jobstatus	activated (4) finished (3) holding (4) running (3) sent (3)
prodsourcelabel	panda (17)
produsername	Artem Petrosyan (17)
transformation	phast (17)

Queue summary, running and recently finished jobs

Owner / VO	Task ID	PanDA ID	Transformation	Status	Created	Start	End	Site	Priority	Job info
Artem Petrosyan	1	181	phast	activated	2015-09-24 11:58			ANALY_CERN_COMPASS_PROD	2000	
Artem Petrosyan	1	180	phast	activated	2015-09-24 11:57			ANALY_CERN_COMPASS_PROD	2000	
Artem Petrosyan	1	179	phast	activated	2015-09-24 11:57			ANALY_CERN_COMPASS_PROD	2000	
Artem Petrosyan	1	178	phast	activated	2015-09-24 11:57			ANALY_CERN_COMPASS_PROD	2000	
Artem Petrosyan	1	177	phast	sent	2015-09-24 11:57	09-24 00:00		ANALY_CERN_COMPASS_PROD	2000	

Monitoring 3/3

Owner	PandaID	TaskID	Status	Created	Start	End	Site	Priority
Artem Petrosyan	457	1	finished	2015-09-24 13:28	09-28 00:00	09-28 00:00	ANALY_CERN_COMPASS_PROD	2000

Job name: lxplus_test_job

Status **finished** indicates that the job has successfully completed.

View the job's stdout, job outputs
Download the job cache tarball containing the job execution scripts

Details of the job

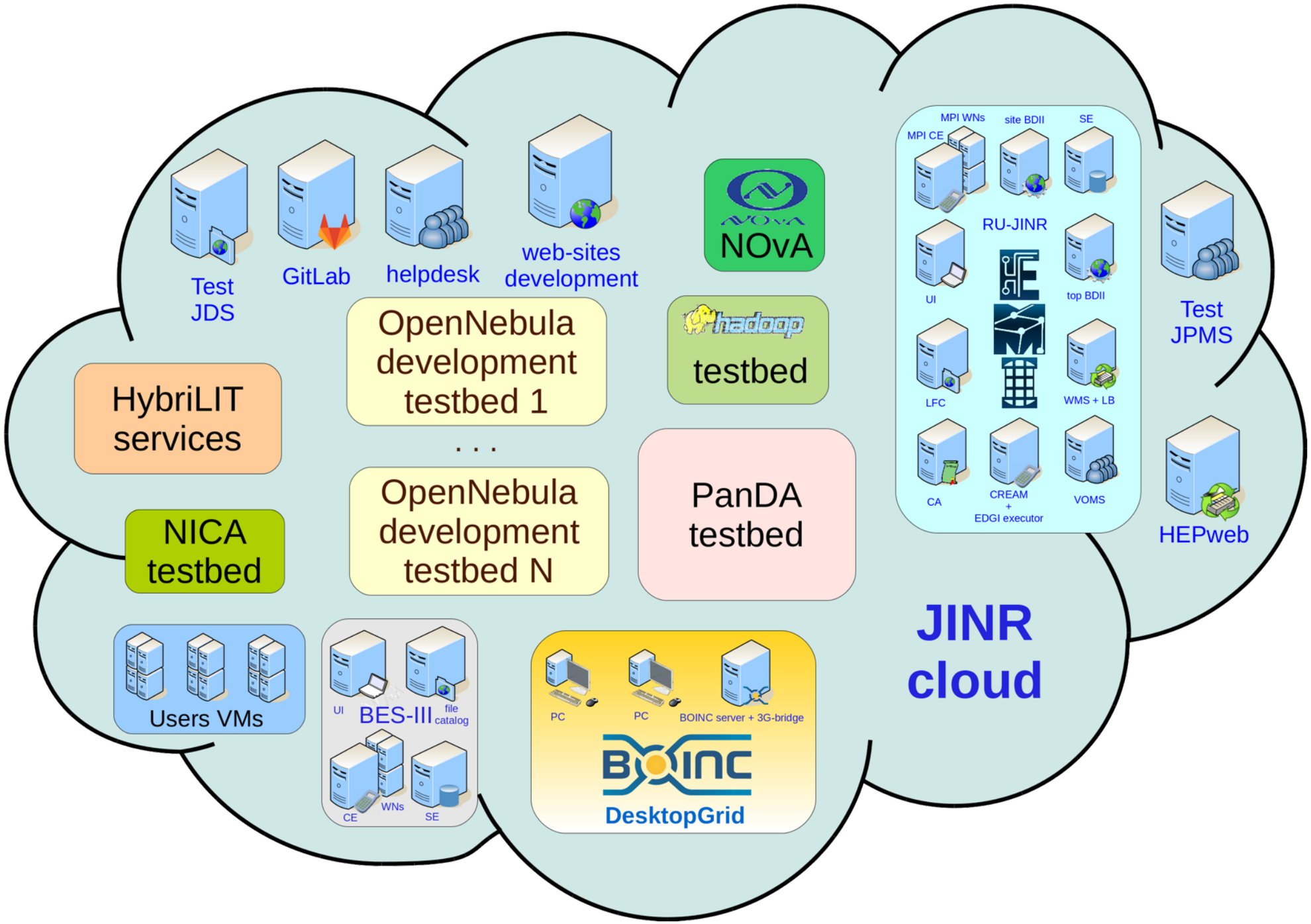
Job files

Filename (Type)	Size (bytes)	Status	Dataset
lxplus_test_job_0af413a0-3e1b-4927-8060-68359b101c68.job.log.tgz (log)	14937	ready	panda.destDB.0af413a0-3e1b-4927-8060-68359b101c68
out_file_megaDST-80625-0-7.root.005 (output)	11974606	ready	panda.destDB.0af413a0-3e1b-4927-8060-68359b101c68

Other key job parameters

Job type	panda
Payload script (transformation)	phast
Output destination	local
CPU consumption time (s)	462
Job parameters	-u0 /castor/cern.ch/compass/generalprod/testcoral/hadron2009t86/megaDST/megaDST-80625-0-7.root.005 -h out_file_megaDST-80625-0-7.root.005
Pilot ID	xtestP001|703148972|LSF|PR|PICARD 62.1
Batch ID	703148972

JINR cloud infrastructure



Status

- Goal was achieved: COMPASS analysis jobs may be executed through PanDA
- PanDA setup was evaluated
- HowTo documentation for COMPASS users was prepared and evaluated by Elena Zemlyanichkina
- More than a thousand jobs were executed successfully during the evaluation
- Test cloud infrastructure at JINR manages the load produced by PanDA server instance well

Next steps

- In next couple weeks
    - Evaluate a real set of production jobs (CORAL) execution via existing setup
    - Enable Auto Pilot Factory (APF) to enable automatic PanDA Pilot submission to several sites
- Soon, i.e. in next couple months
    - Implement a Distributed Data Management (DDM) PanDA plug-ins to open Grid for COMPASS: a common stage-in, stage-out and management of data on several sites through Grid infrastructure
- Future
    - Build a production setup, i.e. migrate from cloud virtual machines to real ones to avoid possible scalability and reliability problems which may arise in future
    - Connect JINR as a computing site for COMPASS through PanDA, i.e. allocate a space and define PanDA queue at JINR's computing infrastructure
- Bright future
    - Define PanDA queues in other participating in COMPASS institutes and make its data analysis distributed

Links

- COMPASS home:

  http://wwwcompass.cern.ch/

- PanDA&BigPanDA home:

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

- Monitoring link to COMPASS PanDA queue at JINR:

  http://vm127.jinr.ru/bigpandamon/jobjobs/?computingsite=ANALY_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