Supercomputer integration in the ALICE workflow

Pavlo Svirin (National Academy of Sciences of Ukraine (UA))

Andrey Kondratyev (Joint Inst. for Nuclear Research (RU))

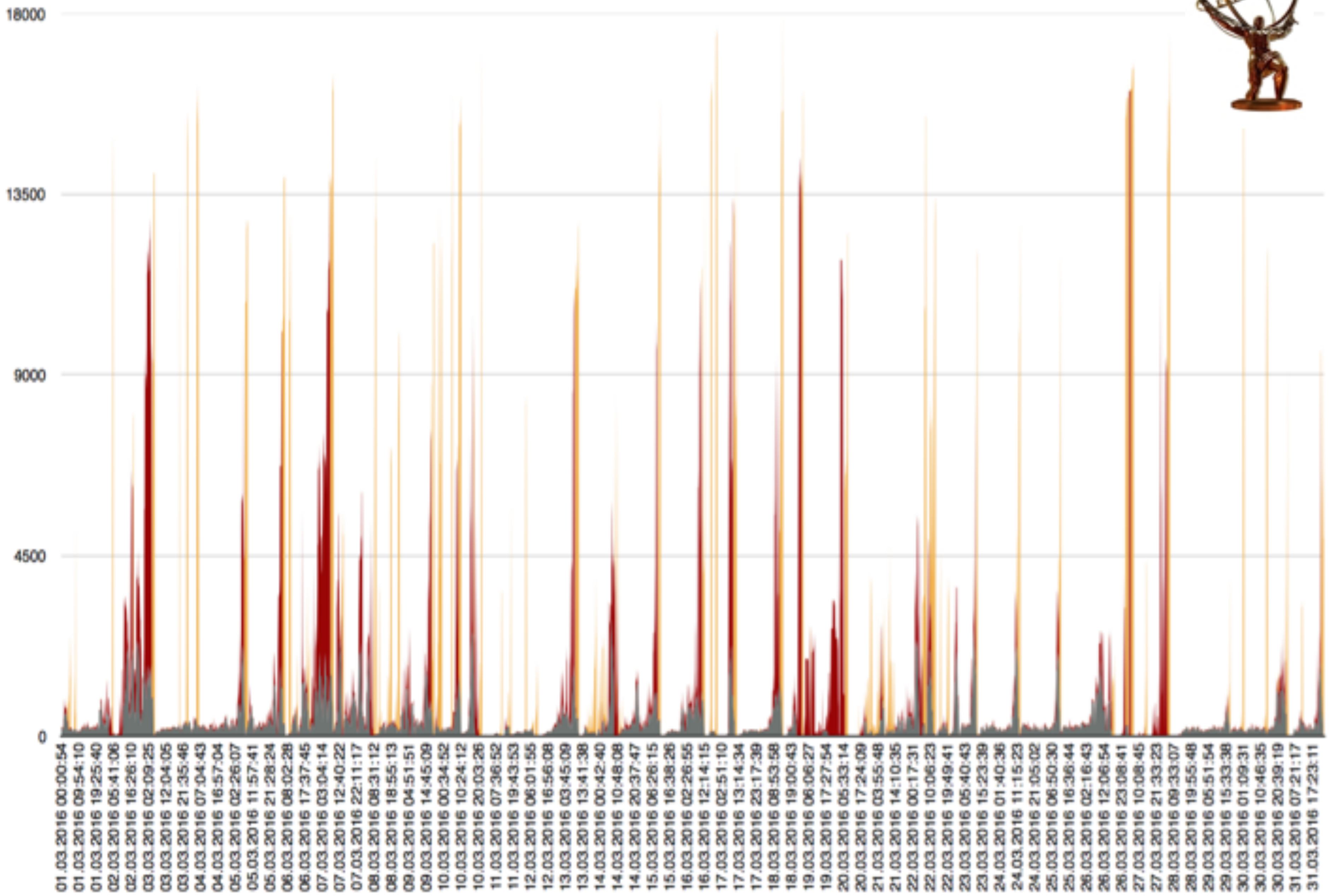
Titan general information

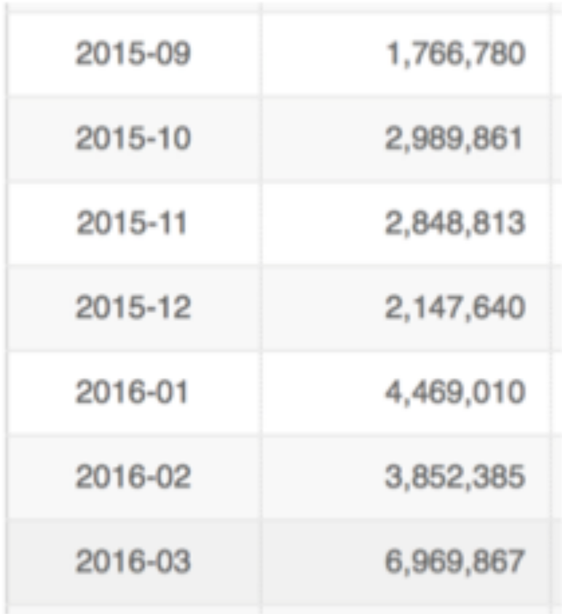
Architecture	18,688 AMD Opteron 6274 16-core CPUs, 18,688 Nvidia Tesla K20X GPUs
Operating system	Traditional Linux and Cray Linux Environment (modified SuSE Linux 11) on worker nodes
Memory	693.5 TiB (584 TiB CPU and 109.5 TiB GPU)
Disk storage	32 PB, 1.4 TB/s IO Lustre filesystem
Peak performance	27.1 PF (18,688 compute nodes, 24.5 GPU + 2.6 PF CPU)
I/O Nodes	512 service and I/O nodes

- 2GB RAM/core
- ‘Free’ resources (in addition to the T2 allocation), potentially up to 10% of the Titan capacity
- Will be used in AliEn environment for Monte-Carlo jobs

CSC108 utilization in March, 2016

- ATLAS is presently obtaining about 4 M Titan core hours per month
- 10,500,000 core hours was an initial quota for CSC108 project, now it’s removed

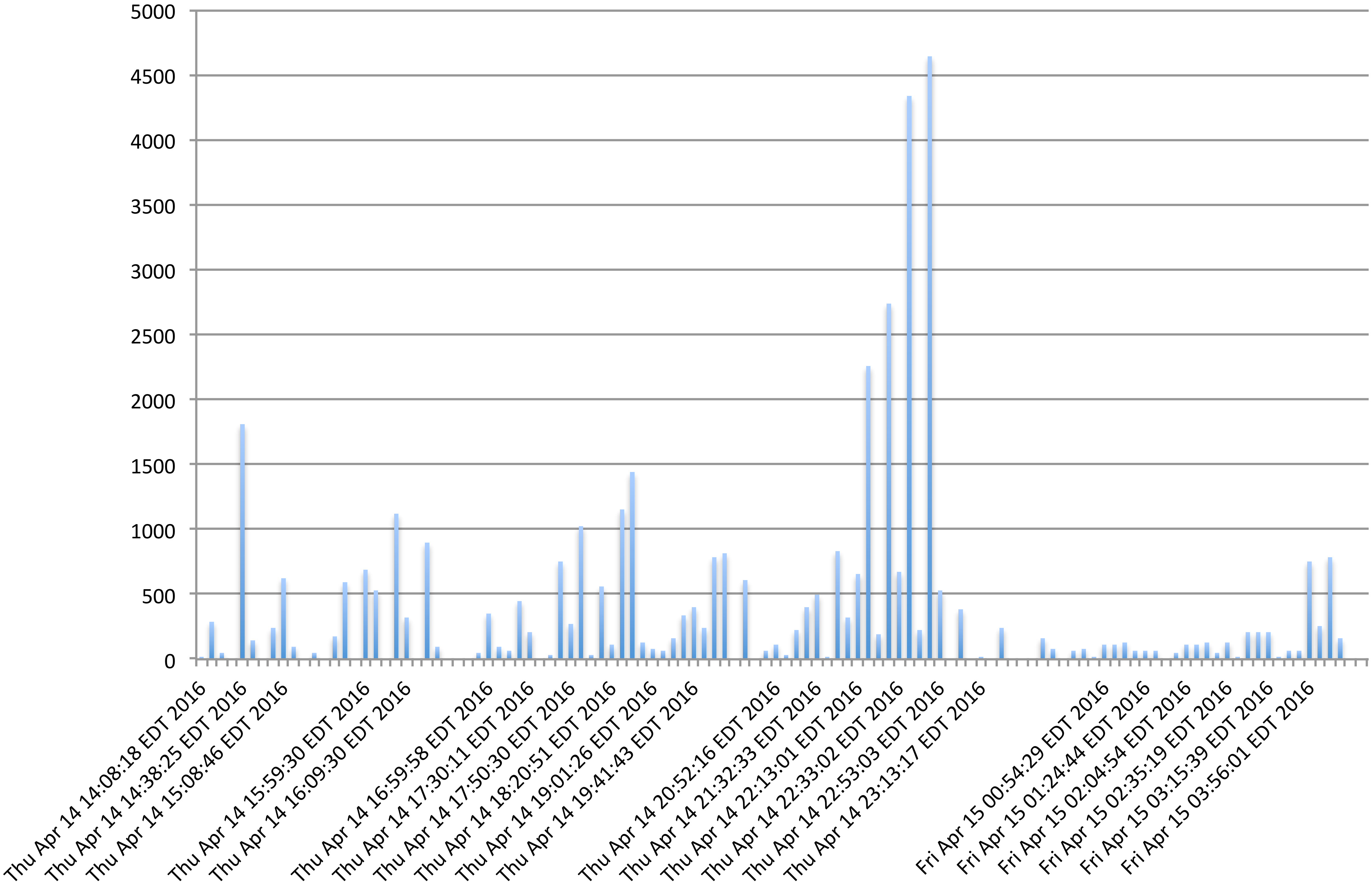




2015-09	1,766,780
2015-10	2,989,861
2015-11	2,848,813
2015-12	2,147,640
2016-01	4,469,010
2016-02	3,852,385
2016-03	6,969,867

(provided by Danila Oleynik (BigPANDA))

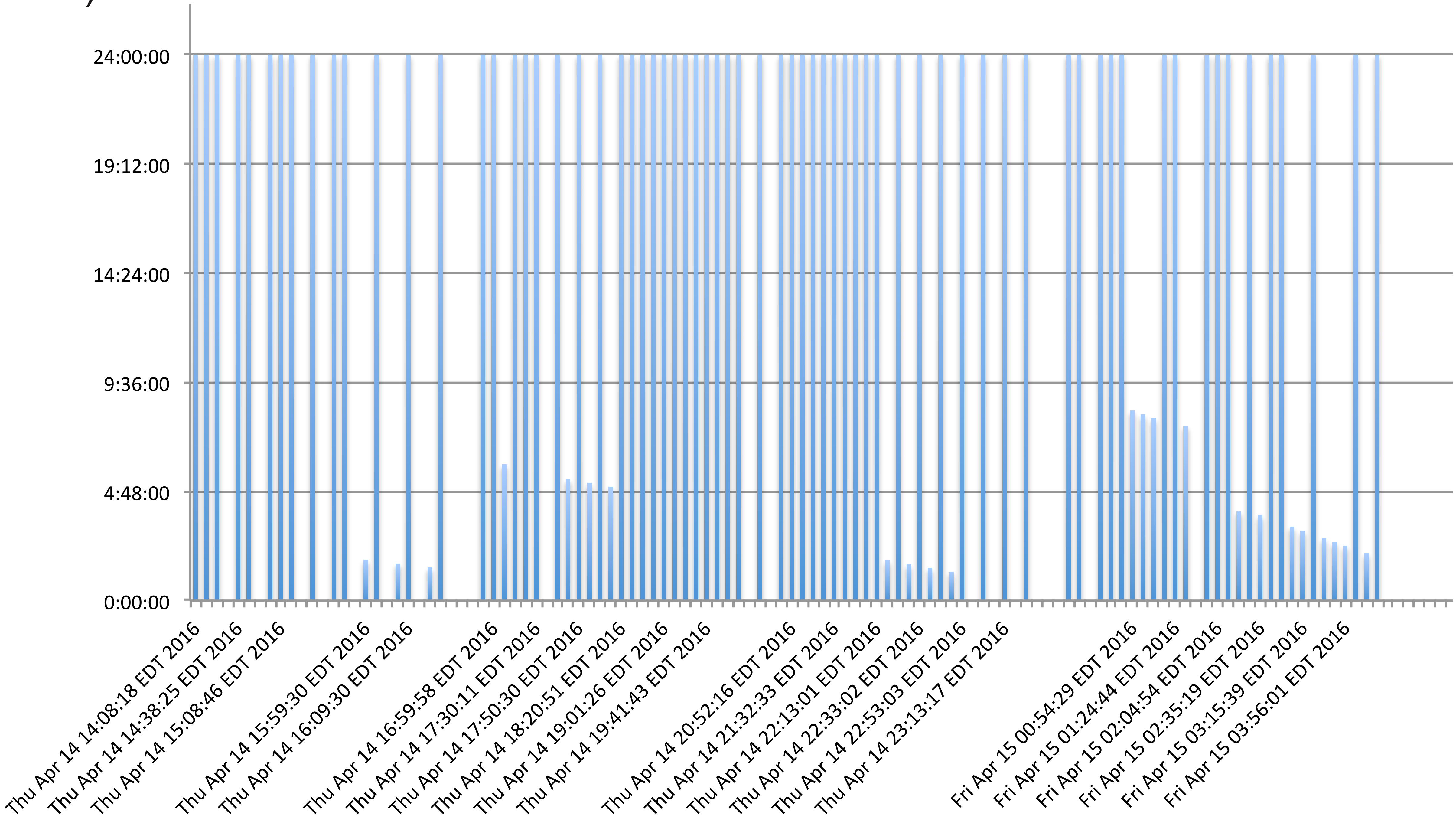
Available free cores count distribution



Measurements taken on April 14 2016.

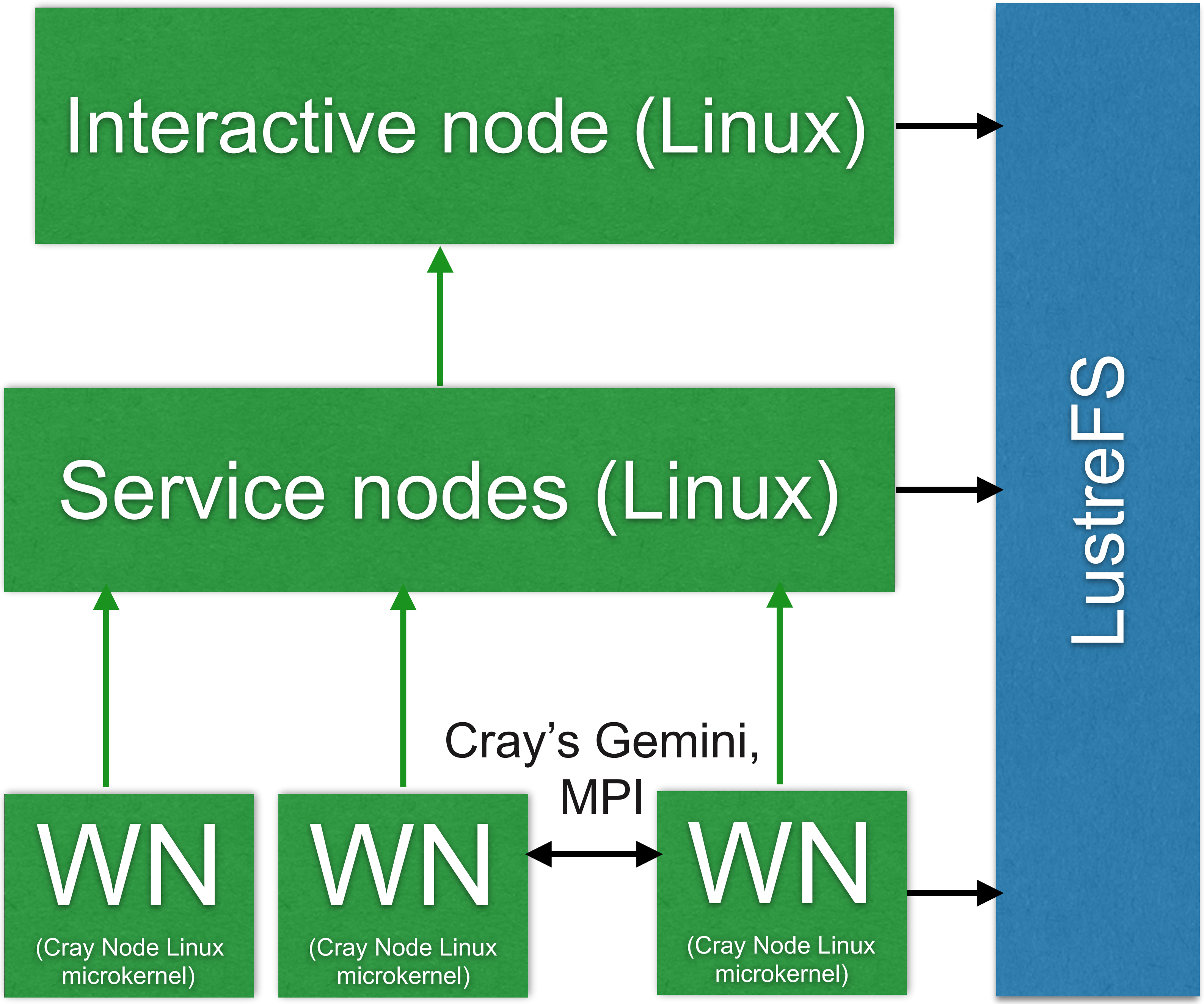
Available nodes timespan distribution

We still able to book (in theory!) 520,000 core hours per day (average timespan ~ 18 hrs)



Measurements taken on April 14 2016.

Titan logical architecture


Interactive node (Linux)
Service nodes (Linux)
Cray's Gemini, MPI
WN (Cray Node Linux microkernel)
WN (Cray Node Linux microkernel)
WN (Cray Node Linux microkernel)
LustreFS

- Interactive nodes (Login and DTN) and Service nodes are the only ones having internet access
- Communication between IN/SN and worker nodes (WN) is done via a file system
- A static copy of CVMFS shared through LustreFS

Batch script for Titan

```
psvirin@titan-ext3:/ccs/home/psvirin/tmp/qsub> qsub -q titan test.pbs
```

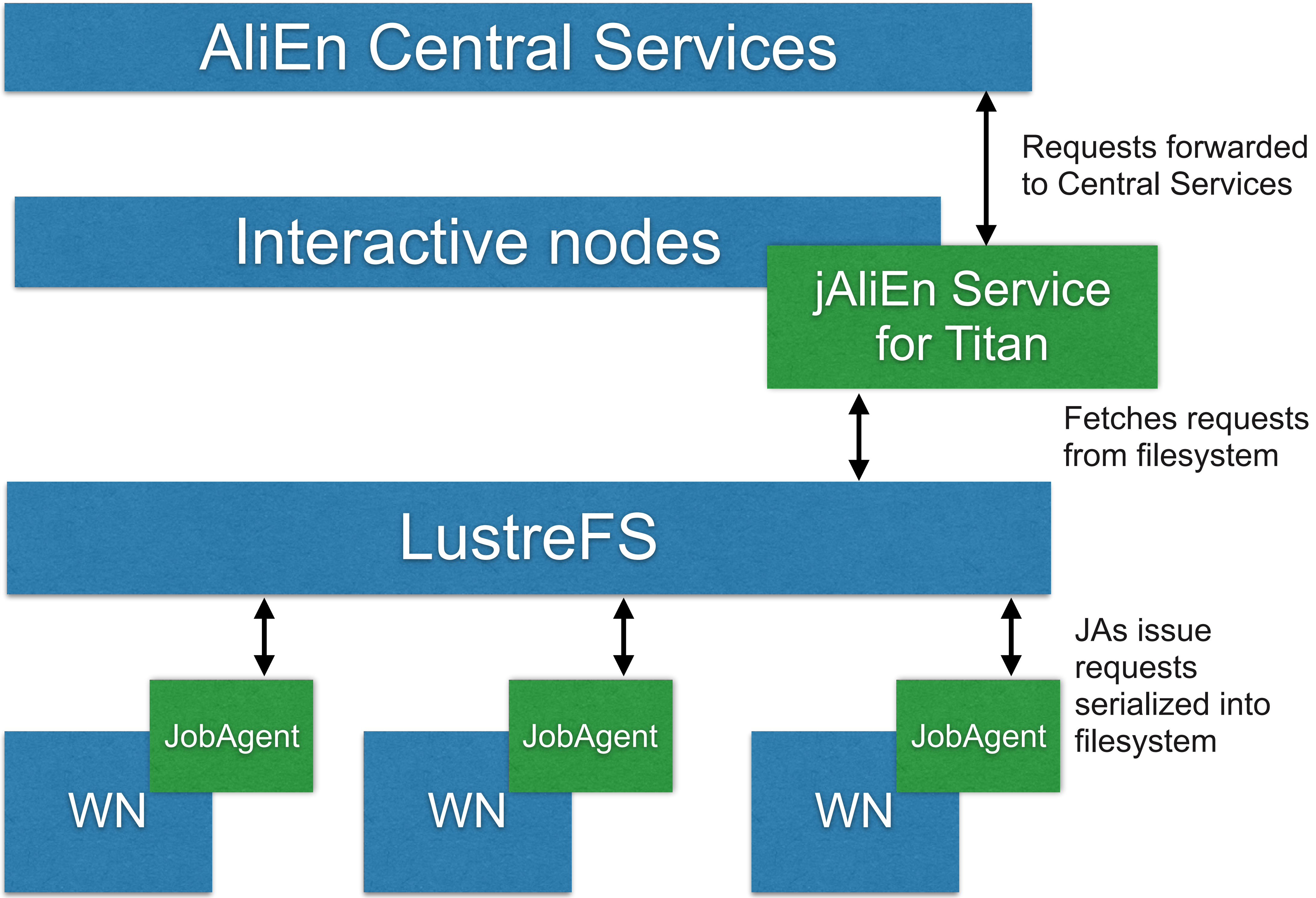
```
#!/bin/bash
#PBS -A CSC108
#PBS -N alien_job
#PBS -j oe
#PBS -l walltime=00:30:00,nodes=160
#PBS -l gres=atlas1

cd $MEMBERWORK/csc108

module load cray-mpich/7.2.5
module load python/3.4.3
module load python_mpi4py/1.3.1

aprun -n 2560 ./get_rank_and_exec_job.py
```

Approach 1 to integrate Titan



Requests forwarded to Central Services

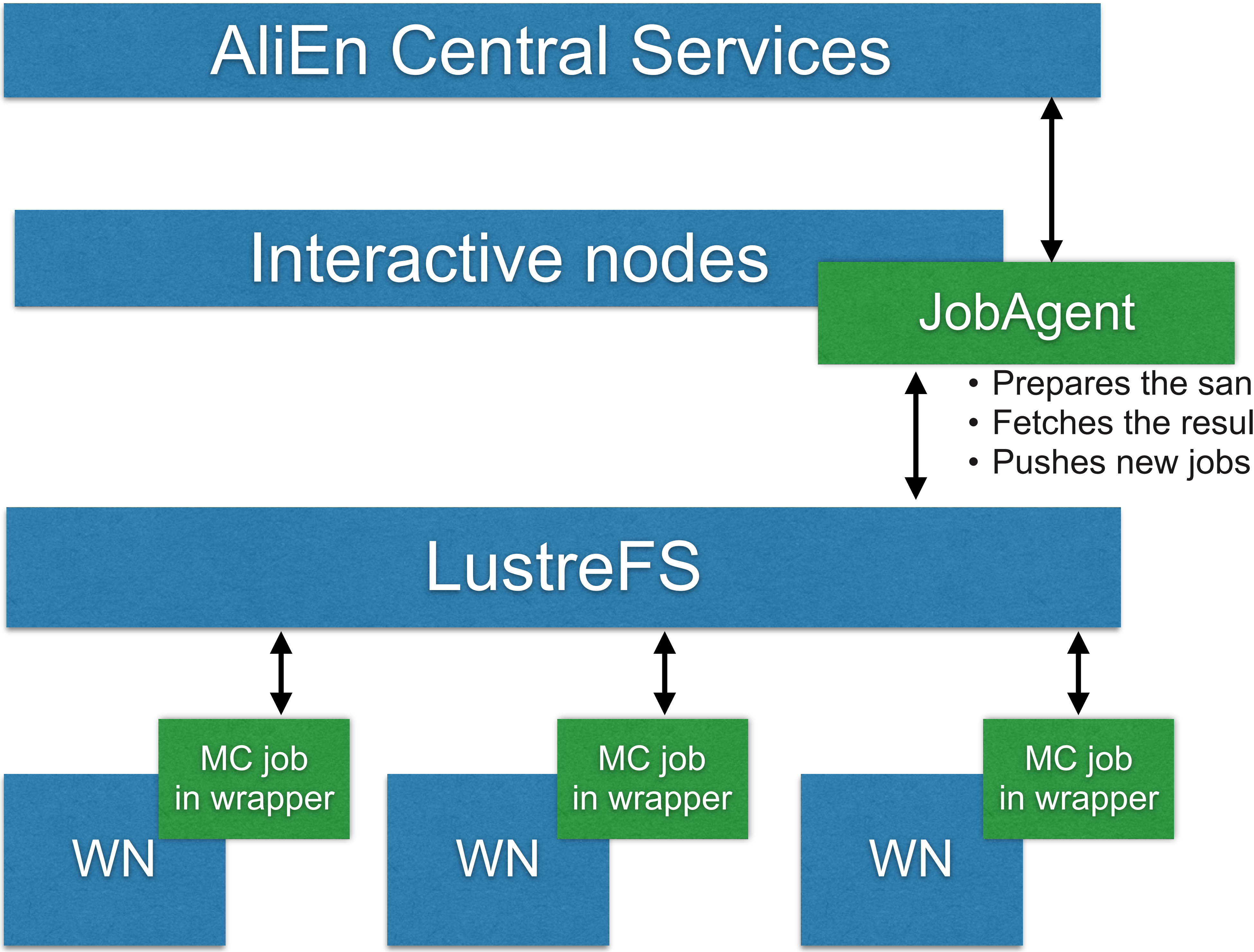
Fetches requests from filesystem

JAs issue requests serialized into filesystem

Approach 1 in details

- we’ve implemented the service that translates jAliEn network calls from filesystem to network and vice-versa
- we have to eliminate all of the direct network calls from a JobAgent (some LDAP calls are still present)
- a bit slow approach (the execution has to wait for the requests to be fetched by the service from a filesystem and then serialized back as a response) but seems to be reliable
- JobAgent has to be profiled to study it’s memory consumption for real jobs

Approach 2 to integrate Titan



- Prepares the sandbox
- Fetches the results
- Pushes new jobs if TTL allows

Approach 2 in details

- we have to modify existing jAliEn so it will be able to handle multiple parallel jobs
- all of the files have to be present in job's folder before the start of the job
- a faster approach as jobs on the worker nodes do not spend time on network I/O
- a prototype for this approach has been tested using 2560 cores (160 nodes); we simulated the information exchange between interactive and worker nodes using SQLite database.

ALICE binary software

- CrayOS is a microkernel modification on SuSE Linux 11.3
- No TCP/IP stack for worker nodes
- We will make a port of existing software for Titan for CRAY OS (Dario is working on it)
- CVMFS is not possible to mount on LustreFS, we will use periodical rsync through parrot
- Only the software versions compatible with Titan will be downloaded into the shared folder on LustreFS
- Titan will be used for specific (likely large-scale) Monte-Carlo cycles

AliEN/Titan integration specifics

- We are preparing a new generation of JobAgent (in Java implemented by Miguel)
- Titan is providing entire node per submission
- OCDB usage is necessary now for Monte-Carlo jobs (we have to find a workaround)
- Specific payload JDLs for Titan
- Explicit target SE for output data
- To be combined with the newly available 3rd party transfer methods
- AliEn software works on Titan
- So far we got a simple CE module for Vobox which allows to run test batches on Titan

PanDA WMS

- was originally developed for the ATLAS experiment at CERN's Large Hadron Collider (LHC) by teams at Brookhaven National Laboratory and the University of Texas at Arlington.

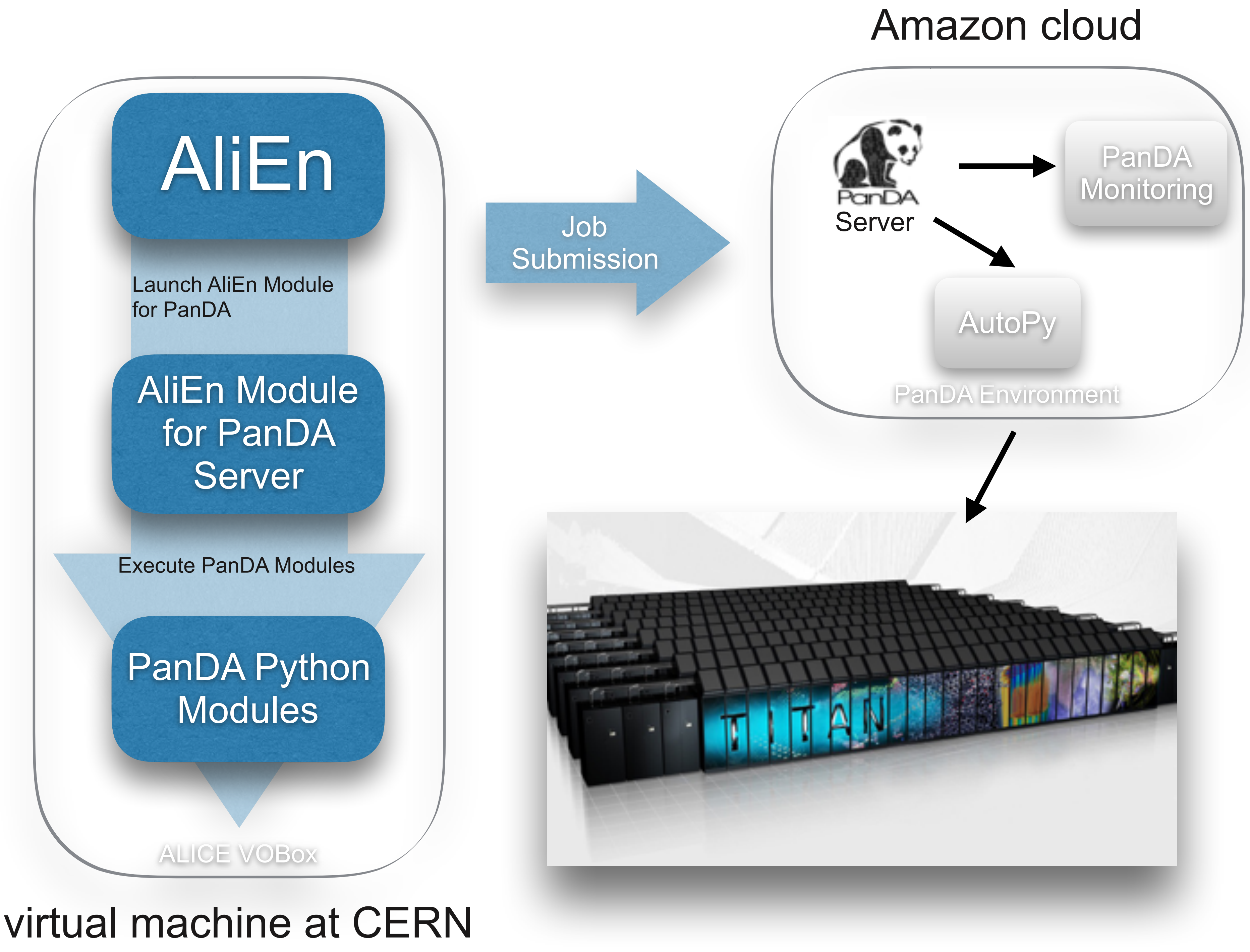
- it is used for the massively scaled distributed data-intensive production and analysis processing of the experiment at over 100 sites globally
  - 150k concurrent jobs around the clock,
  - a million jobs a day,
  - analyzing a data set currently 150 petabytes in size

- it is also being generalized, extended and packaged for use by other scientific communities through the BigPanDA project supported by the US Department of Energy.

AliEn – PanDA interface

- Implemented as a CE module for AliEN by by Andrey Kondratyev (JINR)
- Uses PANDA CLI tools to submit jobs and retrieve statistics about them
- Two PanDA test instances, @Amazon, @JINR
- `hello world` job submission OK

AliEn-PanDA-Titan workflow



AliEn

Launch AliEn Module for PanDA

AliEn Module for PanDA Server

Execute PanDA Modules

PanDA Python Modules

ALICE VOBox

virtual machine at CERN

Job Submission

Amazon cloud

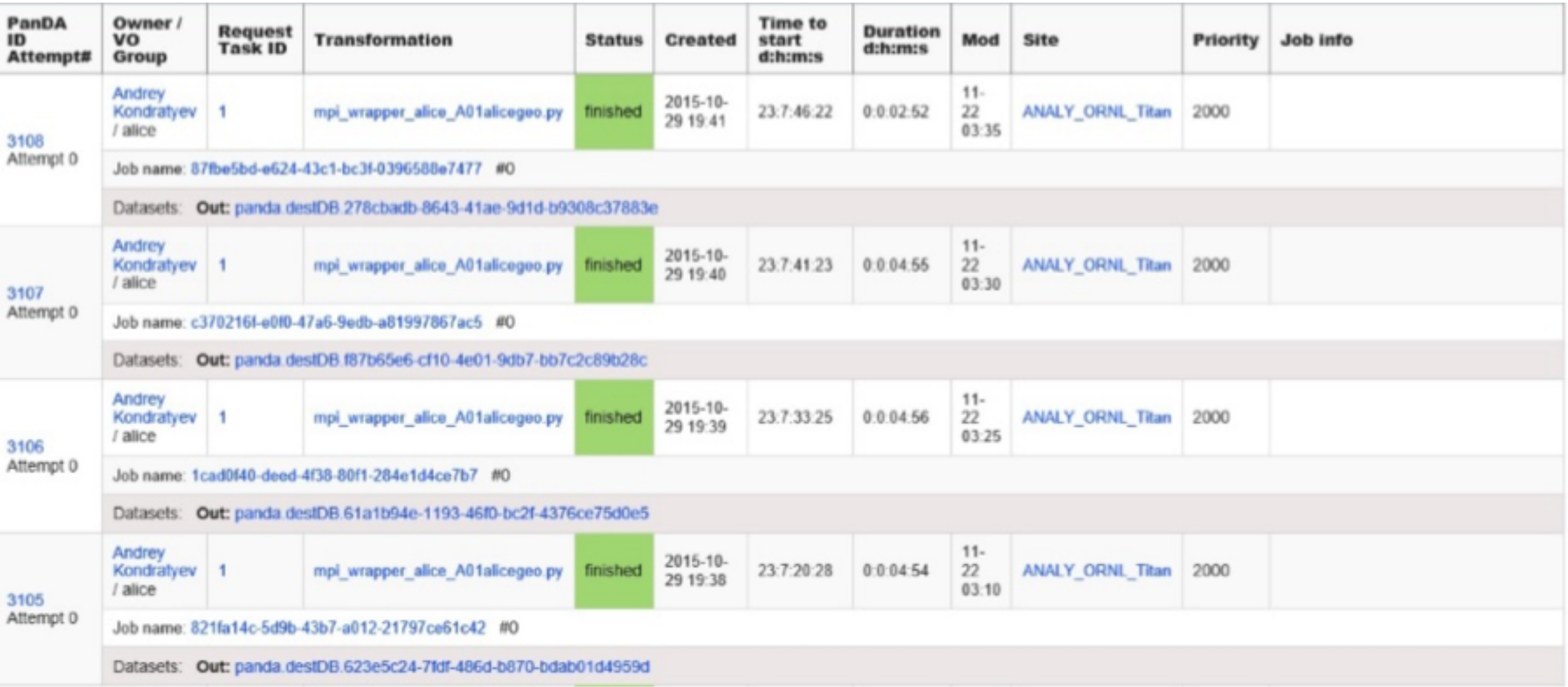
PanDA Server

PanDA Monitoring

AutoPy

PanDA Environment

Jobs submitted to Titan via PANDA



PanDA ID Attempt#	Owner / VO Group	Request Task ID	Transformation	Status	Created	Time to start d:h:m:s	Duration d:h:m:s	Mod	Site	Priority	Job info
3108 Attempt 0	Andrey Kondratyev / alice	1	mpi_wrapper_alice_A01alicegeo.py	finished	2015-10-29 19:41	23:7:46:22	0:0:02:52	11-22 03:35	ANALY_ORNL_Titan	2000	
	Job name: 87fbe5bd-e624-43c1-bc3f-0396588e7477 #0										
	Datasets: **Out:** panda.destDB.278cbadb-8643-41ae-9d1d-b9308c37883e										
3107 Attempt 0	Andrey Kondratyev / alice	1	mpi_wrapper_alice_A01alicegeo.py	finished	2015-10-29 19:40	23:7:41:23	0:0:04:55	11-22 03:30	ANALY_ORNL_Titan	2000	
	Job name: c370216f-e0f0-47a6-9edb-a81997867ac5 #0										
	Datasets: **Out:** panda.destDB.f87b65e6-cf10-4e01-9db7-bb7c2c89b28c										
3106 Attempt 0	Andrey Kondratyev / alice	1	mpi_wrapper_alice_A01alicegeo.py	finished	2015-10-29 19:39	23:7:33:25	0:0:04:56	11-22 03:25	ANALY_ORNL_Titan	2000	
	Job name: 1cad0f40-deed-4f38-80f1-284e1d4ce7b7 #0										
	Datasets: **Out:** panda.destDB.61a1b94e-1193-46f0-bc2f-4376ce75d0e5										
3105 Attempt 0	Andrey Kondratyev / alice	1	mpi_wrapper_alice_A01alicegeo.py	finished	2015-10-29 19:38	23:7:20:28	0:0:04:54	11-22 03:10	ANALY_ORNL_Titan	2000	
	Job name: 821fa14c-5d9b-43b7-a012-21797ce61c42 #0										
	Datasets: **Out:** panda.destDB.623e5c24-7fdf-486d-b870-bdab01d4959d										

Conclusions and future work

- we got our first experience with real Titan
- make jAliEn pilot compatible with worker nodes (remove network calls completely)
- make ALICE binary software compatible with Titan
- finish testing ALICE/PanDA/Titan integration for simple case (running JobAgent on worker nodes)
- Resources-rich environment - thousands of CPU/ hours/day opportunistically

THANK YOU!