A Large Ion Collider Experiment

ALICE

Supercomputers and HPCs in ALICE

L.Betev
ORNL BigPanda workshop
31/03/2017

Acknowledgements

- Part of the slides are borrowed from Pavlo Svirin
- Technical work done by Pavlo Svirin and Andrey Condratyev

Existing projects

- Titan at ORNL (Supercomputer)
    - Most advanced
- CORI at LBNL (Supercomputer)
    - Local use, Gridification not advanced
- Centre de calcul intensif des Pays de la Loire (CCIPL, Nantes)
    - Co-location of computing equipment, common use
    - Managed remotely from a T2
    - Economy of scale for equipment purchases

Common goals

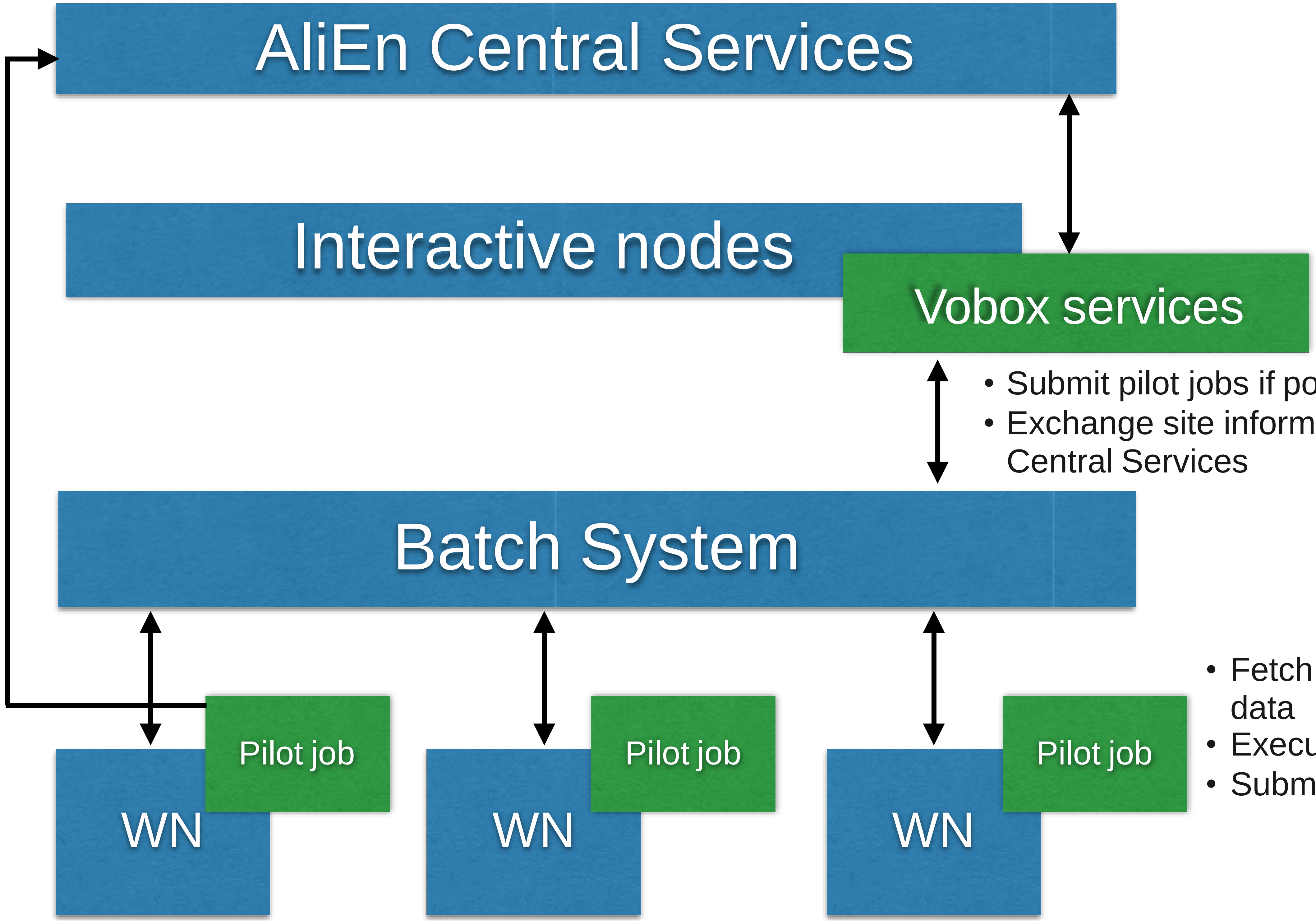
- Use opportunistically or allocated resources
- Include seamlessly into the Grid operations model
  - Adapt the existing Grid middleware to the new system(s)
  - Use resources without adding manpower
- Look for commonalities and partner with other Vos
  - For example ATLAS PanDa

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

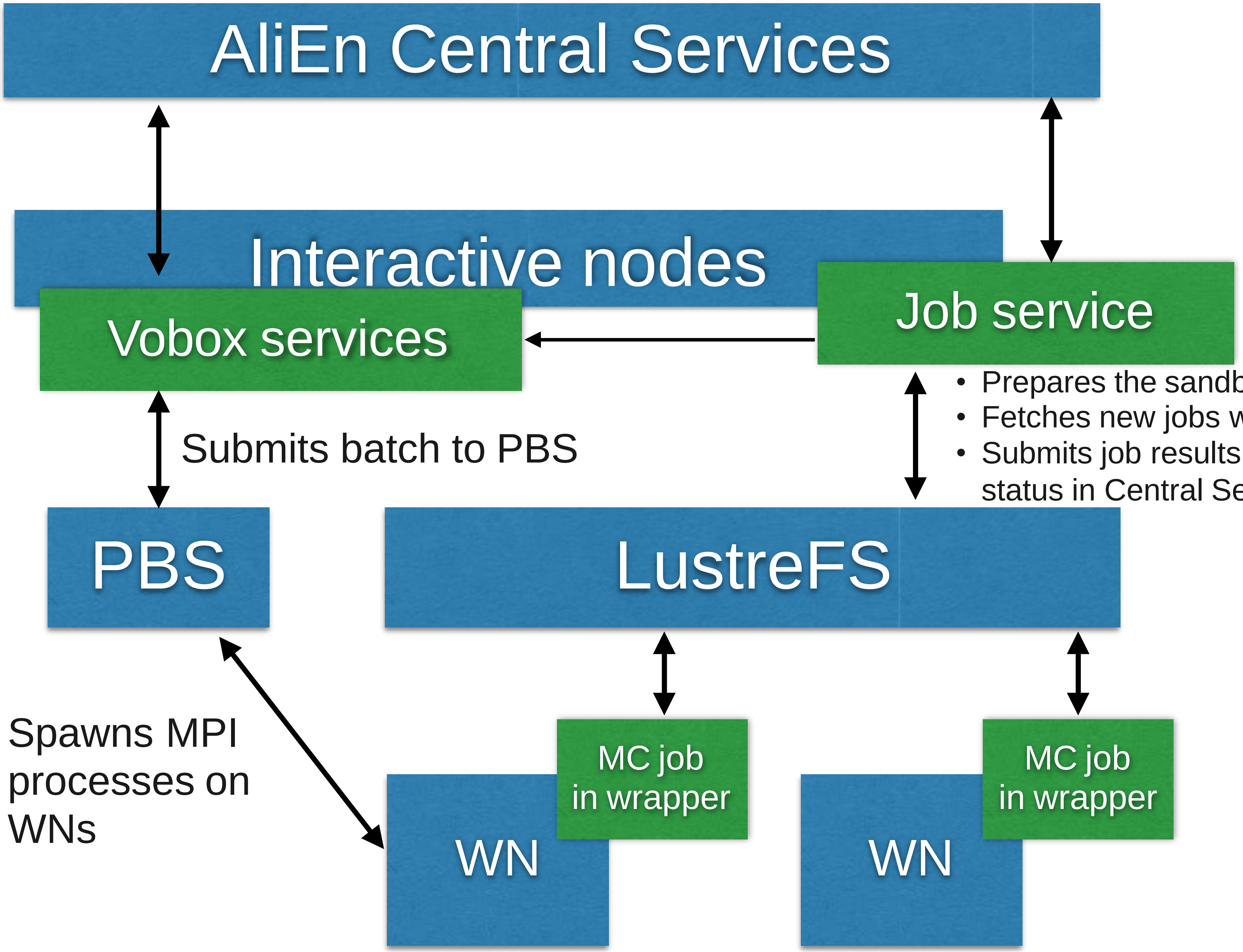
Usual ALICE Grid Environment elements



- Submit pilot jobs if possible
- Exchange site information with Central Services

- Fetch jobs and its data
- Execute jobs
- Submit job results

ALICE Grid Infrastructure and ORNL Titan

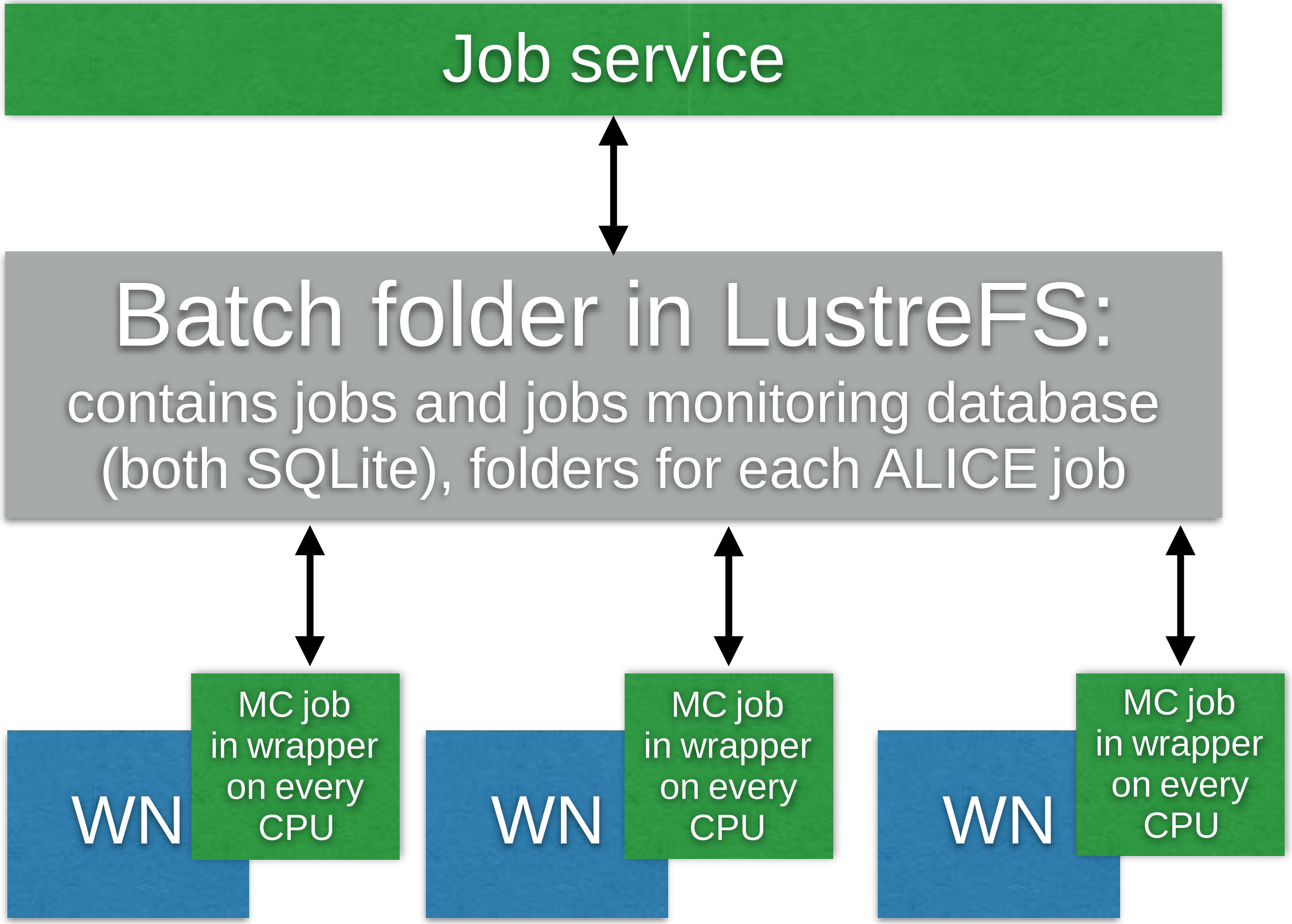


- Prepares the sandbox
- Fetches new jobs when batch starts
- Submits job results, controls job status in Central Services

Submits batch to PBS

Spawns MPI processes on WNs

Titan job service - batch interaction


Job service
Batch folder in LustreFS:
contains jobs and jobs monitoring database (both SQLite), folders for each ALICE job
WN
MC job in wrapper on every CPU
WN
MC job in wrapper on every CPU
WN
MC job in wrapper on every CPU

```
Service_Working_Folder
|____
|____2995314
| |____jalien-job-741337413
| |____jalien-job-741338229
| |____jalien-job-741338682
| |____……
| |____jobagent.db
| |____jobagent.db.monitoring
```

```
| |____jalien-job-741337413
| | |____jdl
| | |____environment
| | |____ OCDBsim.root
| | |____OCDBrec.root
| | |____fifo
| | |____aliroot_dpgsim.sh
| | |____validation.sh
```

Software distribution

- Application software build in the standard ALICE framework (for Titan)
  - No special compiler directives were used to build software
- Distributed through shared FS
  - CVMFS repository subset on Titan, updated every hour
  - publisher script had to be brushed up because Titan was cutting too frequent outbound network connections
- Similar approach for CORI

Running ALICE MC jobs on Titan

- CPU-intensive Pb-Pb production

JobTag = {"comment:Pb-Pb, 5.02 TeV - HIJING min bias - General-purpose Monte Carlo production anchored to Pb-Pb 5.02 TeV runs (LHC15o), ALIROOT-6784"};

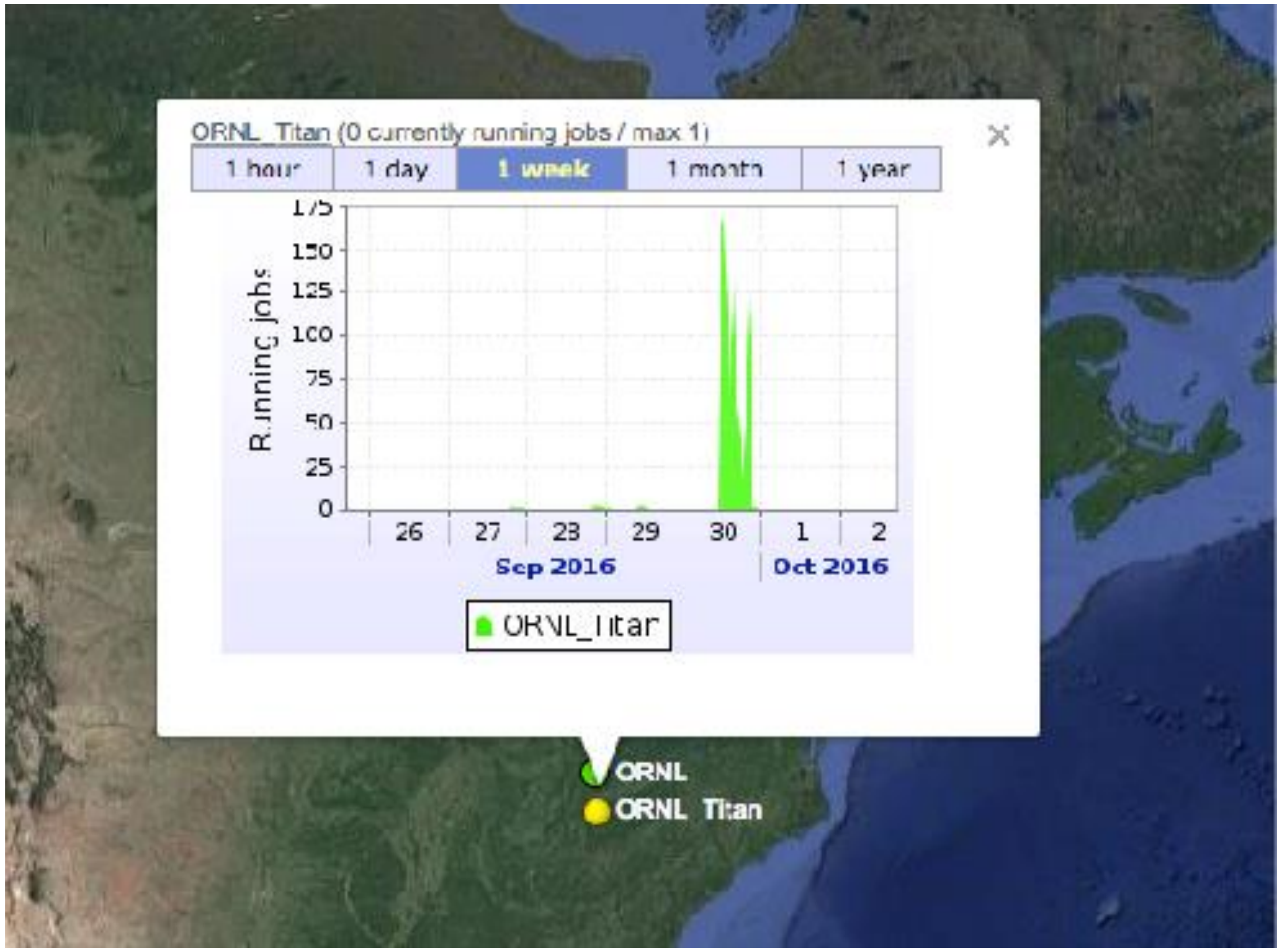
processing takes up to 3 hours for 1 event

- LHCbMarks: 5.56 for worker node CPU, corresponds to estimated 0.35/events per hour, 7.60 on interactive nodes

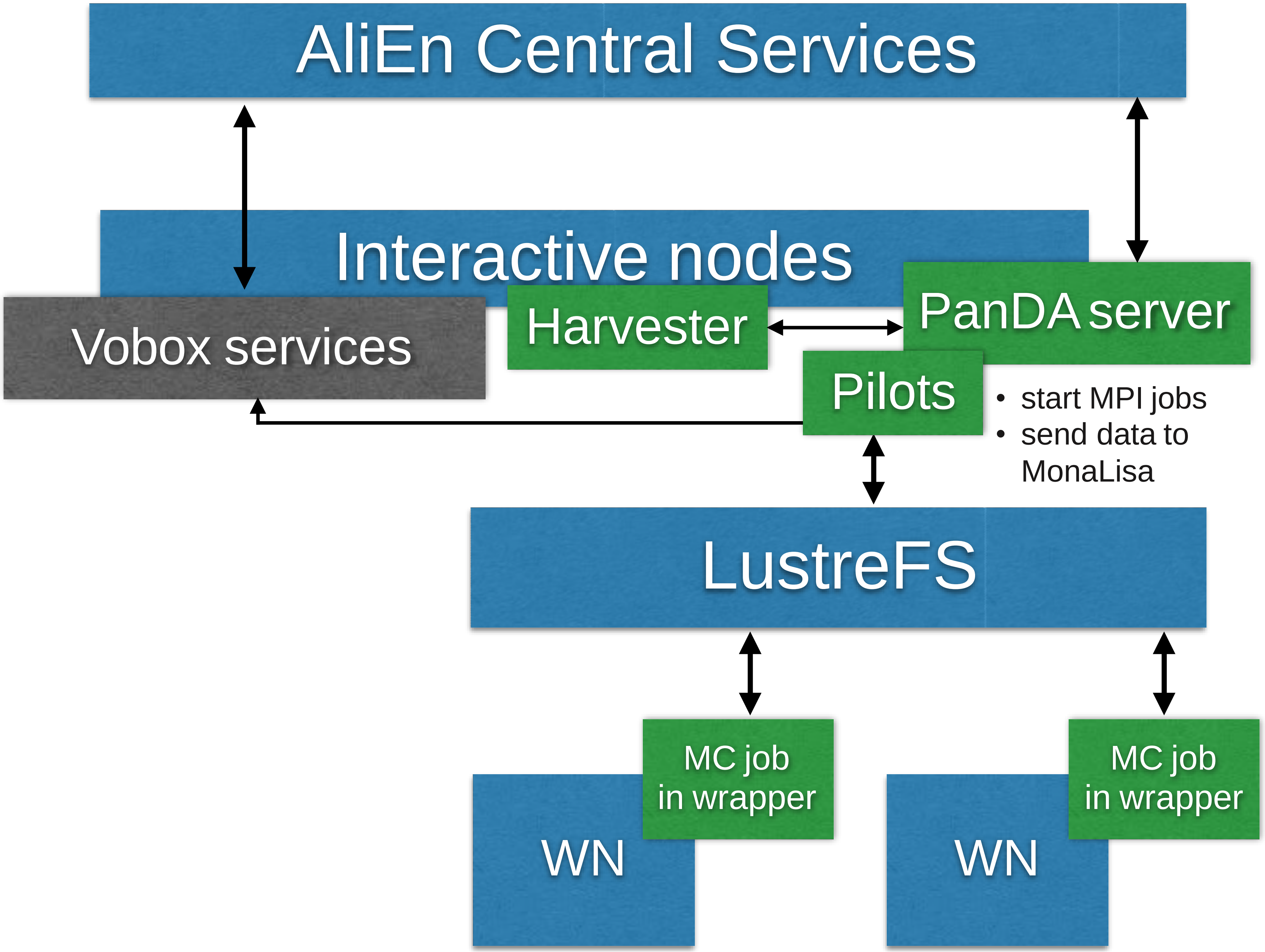
  - For comparison, average CERN CPU core is 12LHCbMarks
- Jobs we can not profit from pure backfill (usually less than 2 hours), CSC108 project has the lowest priority
- Successful in requesting 125 nodes/5:45h slots which can be ok for 2 events (theoretically up to 10 slots per day)

Titan in ALICE monitoring system

ORNL
ORNL Titan
UNAM T1
UNAM



PanDA integration and Pilot2 (draft design)



- start MPI jobs
- send data to MonaLisa

- Fetches jobs from Central job queue, uploads finished jobs
- Fetches pilot descriptions from Harvester service
- Runs pilots

PanDA integration: details and challenges

- PanDA server takes pilot description from Harvester service (more: https://indico.cern.ch/event/526308/contributions/2247704/attachments/1318598/1976659/Harvester.pdf)
- uses pre-binding for jobs: jobs need to be kept in ASSIGNED state for a certain period
- possible to play with "—mode" job option to split the job stages between the time slots (has to be tested)
- we can use HTTP/JSON calls for running jAliEn commands through Tomcat (approach has been tested in August 2015)
- bash job wrapper ready for PanDA

Conclusions and future work

- ALICE Grid software adapted to run on Titan
- Application software (MC) is adapted to network-less environment
- Software distribution is not ideal – CVMFS on the nodes would definitely help
- Backfill mode does not suit well the standard jobs – will have to find another set of tasks or ask for specific allocation
- Integration with the new PanDA structure is starting

- Many thanks to Alexei Klimentov for the technical and human resources support!