ALICE

ALICE use of HPC Facilities

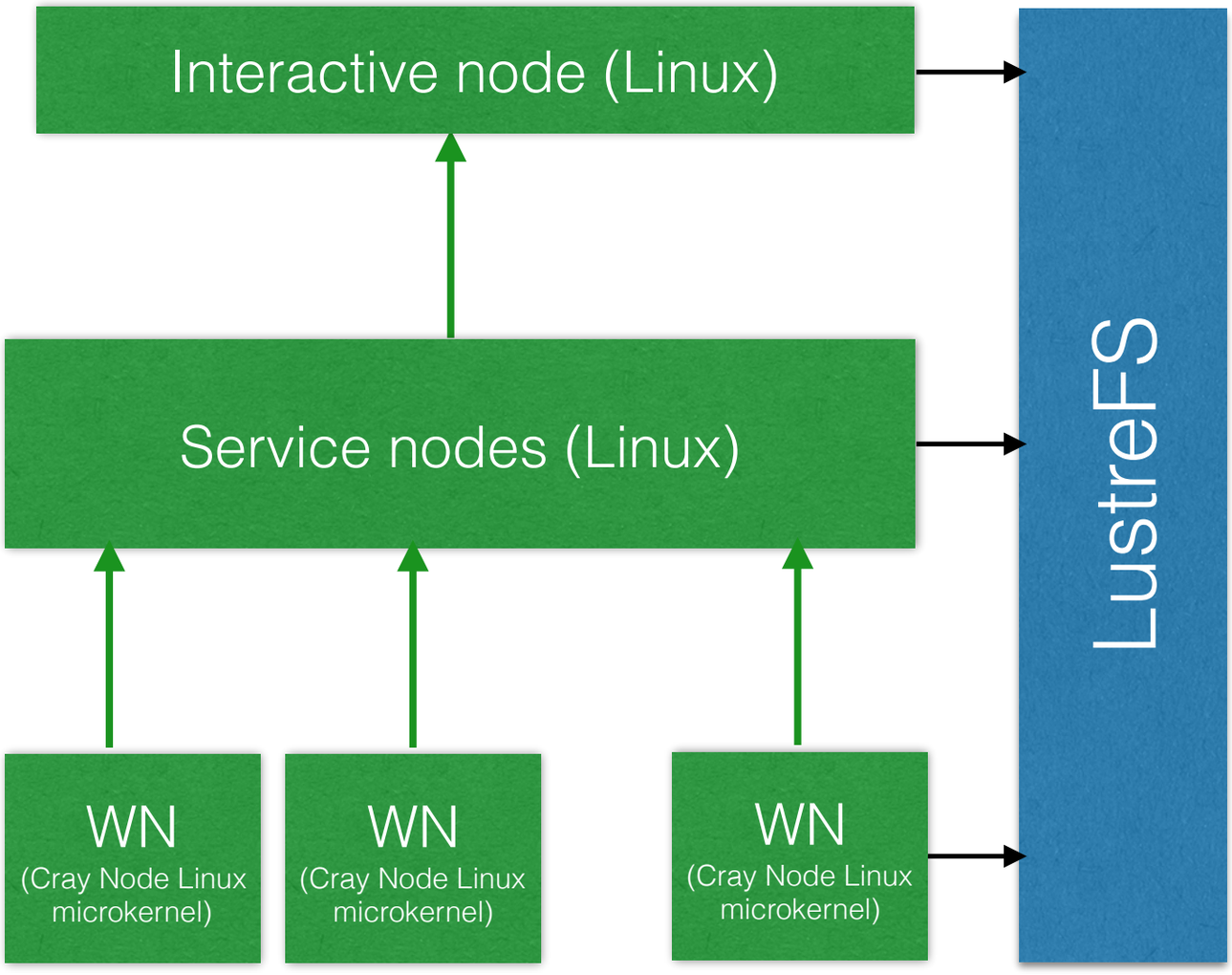
ALICE

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

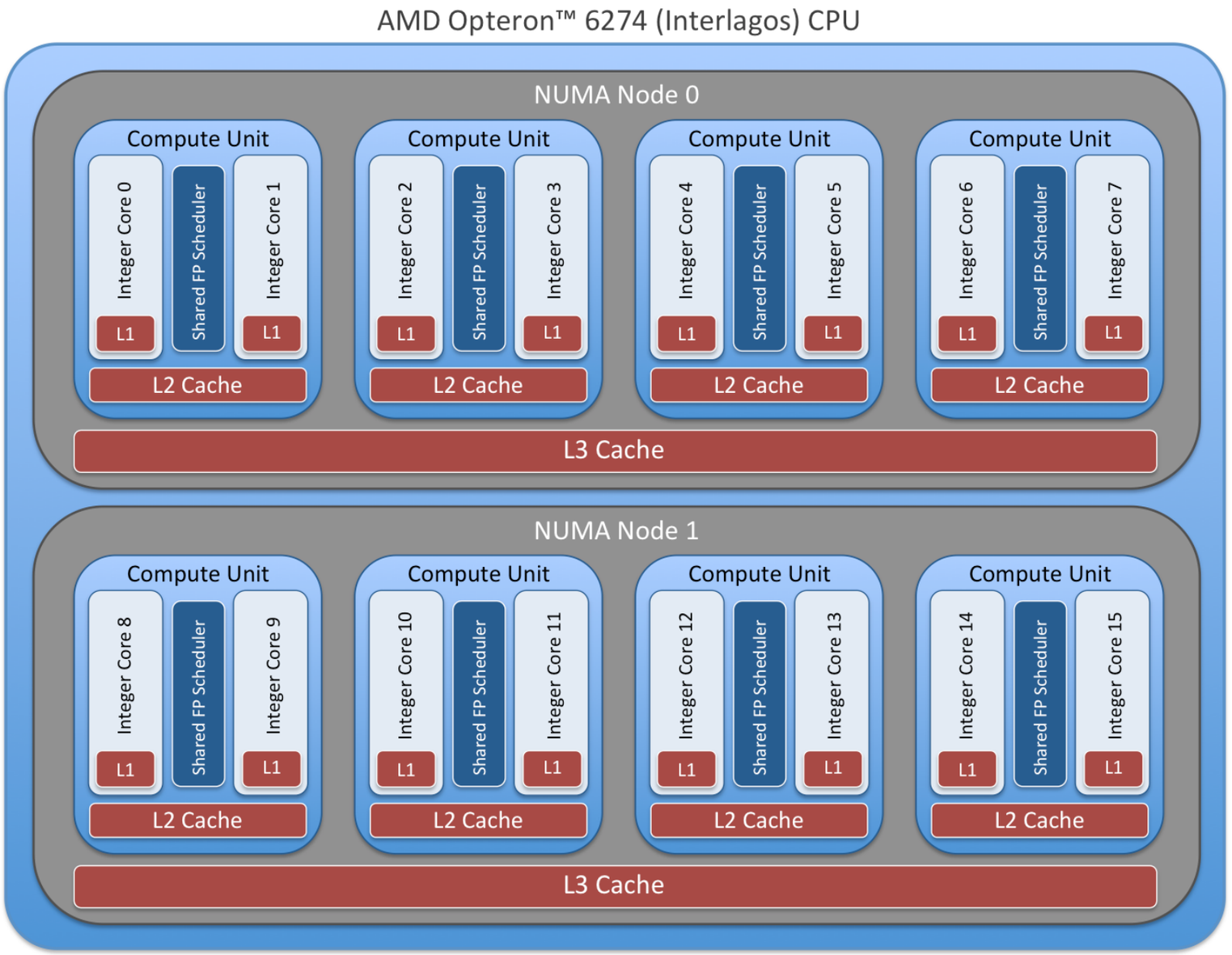
- 2GB RAM/core
- ‘Free’ resources (in addition to the T2 allocation), potentially up to 10% of the Titan capacity
- Will be used in AliEn environment for Monte-Carlo jobs

Titan logical architecture



- Interactive nodes and Service nodes are the only one 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

Titan Worker Node Architecture



Each Titan compute node contains :

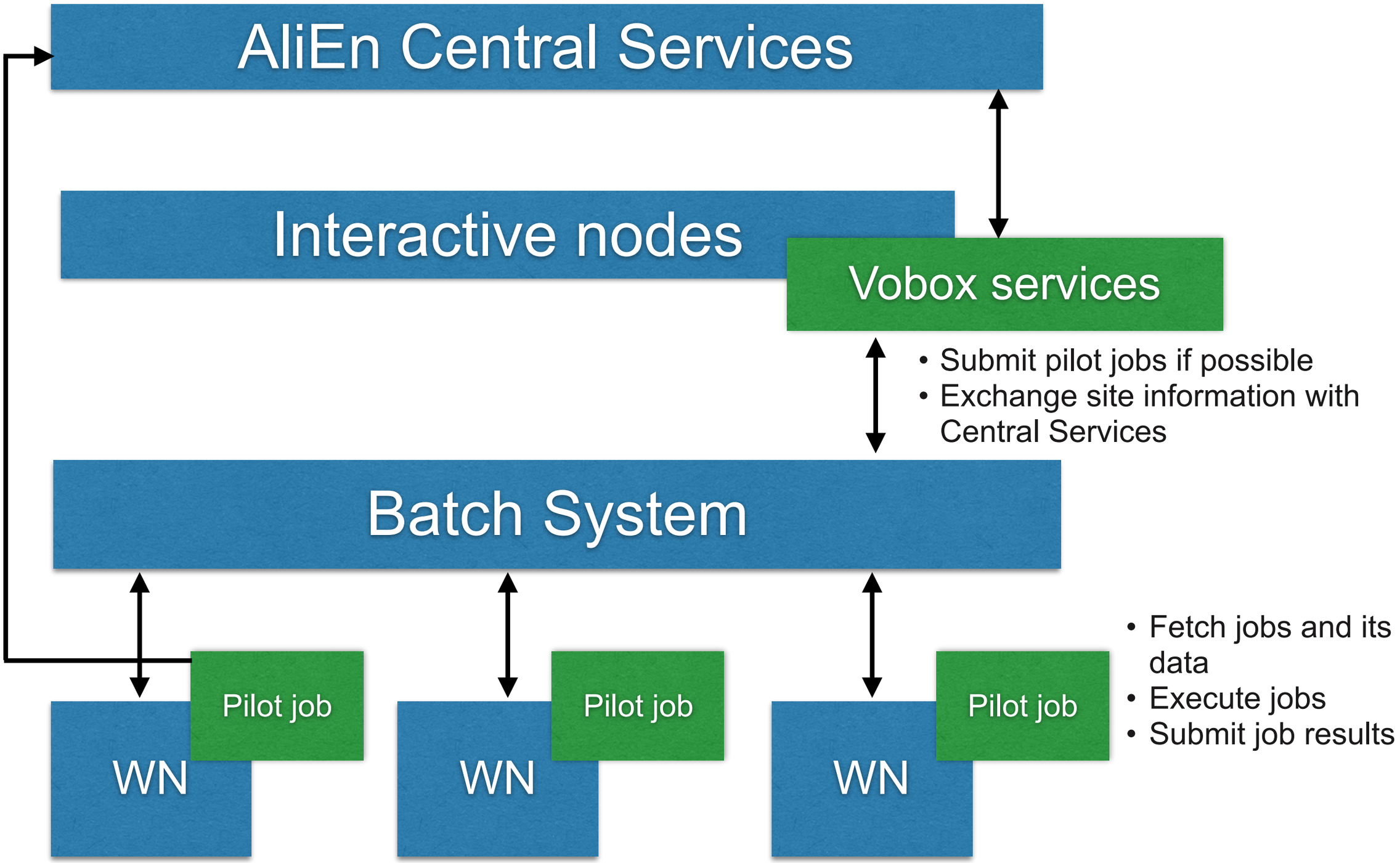
(1) AMD Opteron™ 6274 (Interlagos) CPU. Each CPU contains (2) die.

Each die contains (4) "bulldozer" compute units and a shared L3 cache.

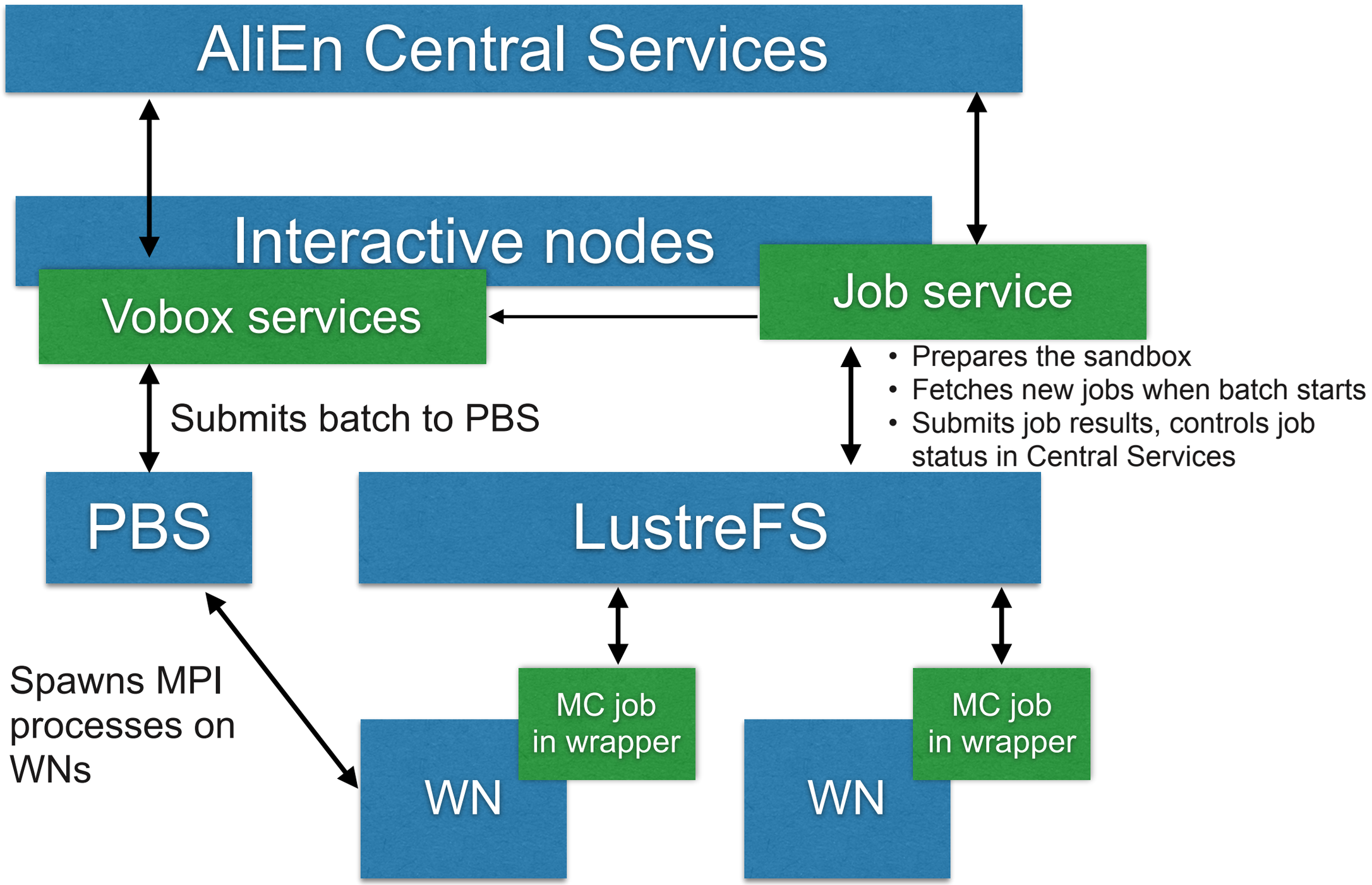
Each compute unit contains (2) integer cores (and their L1 cache), a shared floating point scheduler, and shared L2 cache. To aid in task placement, each die is organized into a NUMA node.

Each compute node contains (2) NUMA nodes. Each NUMA node contains a die's L3 cache and its (4) compute units (8 cores).

Usual ALICE Grid Environment elements



ALICE Grid Infrastructure and ORNL Titan



Batch life cycle

Batch submitted to PBS, waiting

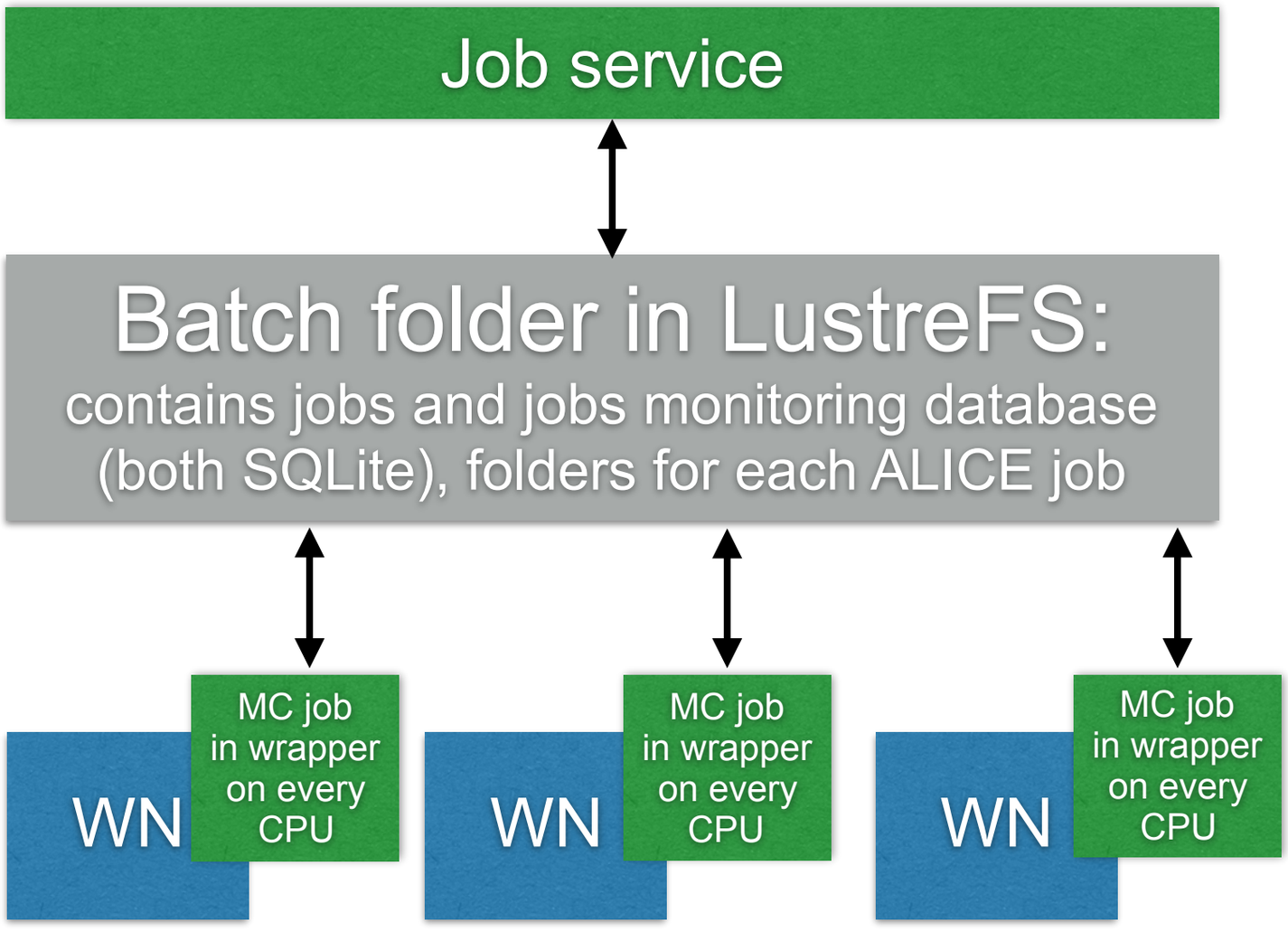
Batch started
creating batch folder, batch info database, waiting for a response from service (second part of the tables are created)

Database structure ready
spawning MPI processes, waiting for job information to be available in the database

Job execution

Cleaning up on exit

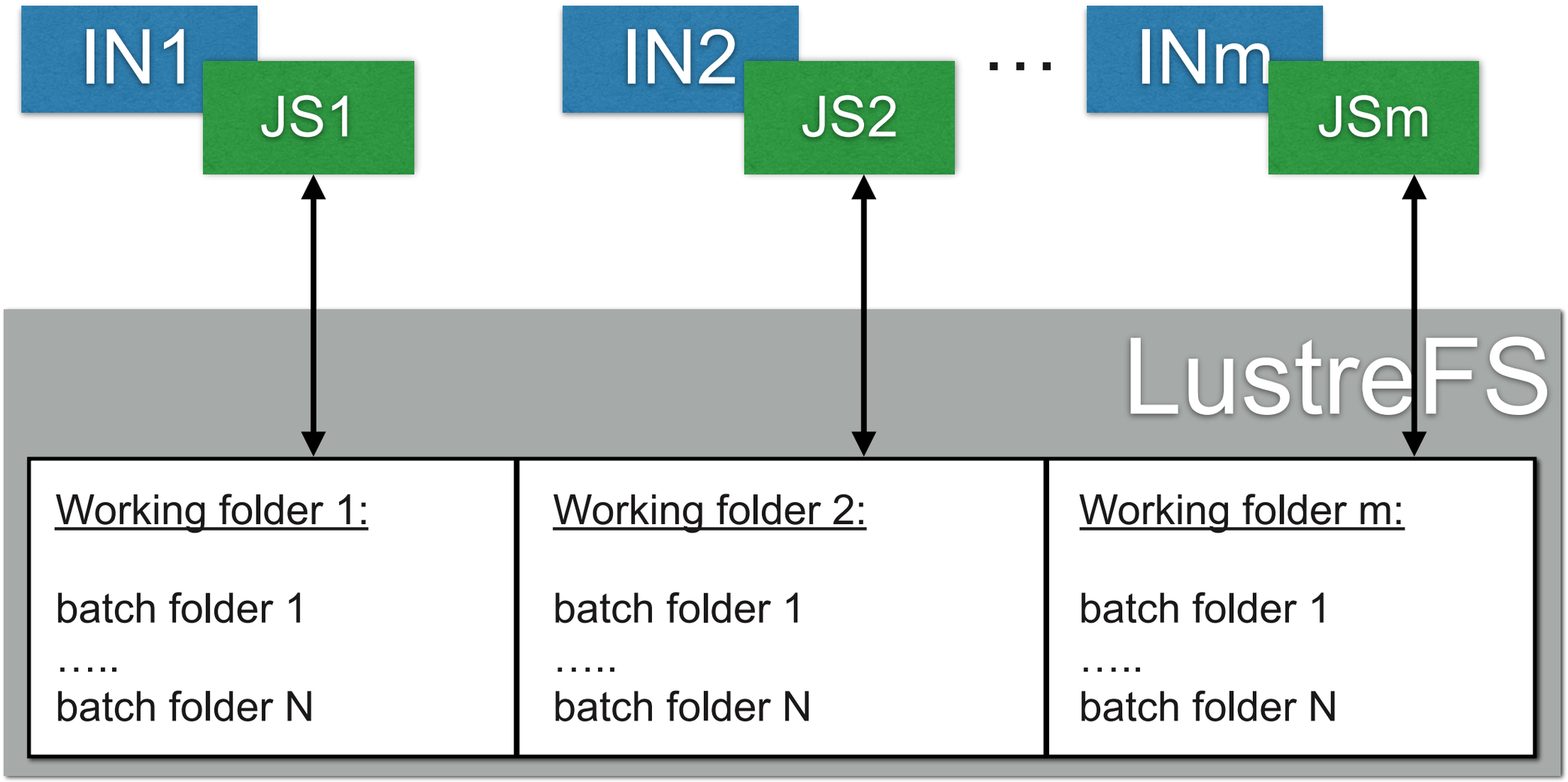
Titan job service - batch interaction




```
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
```

Possible scaling for Job Services



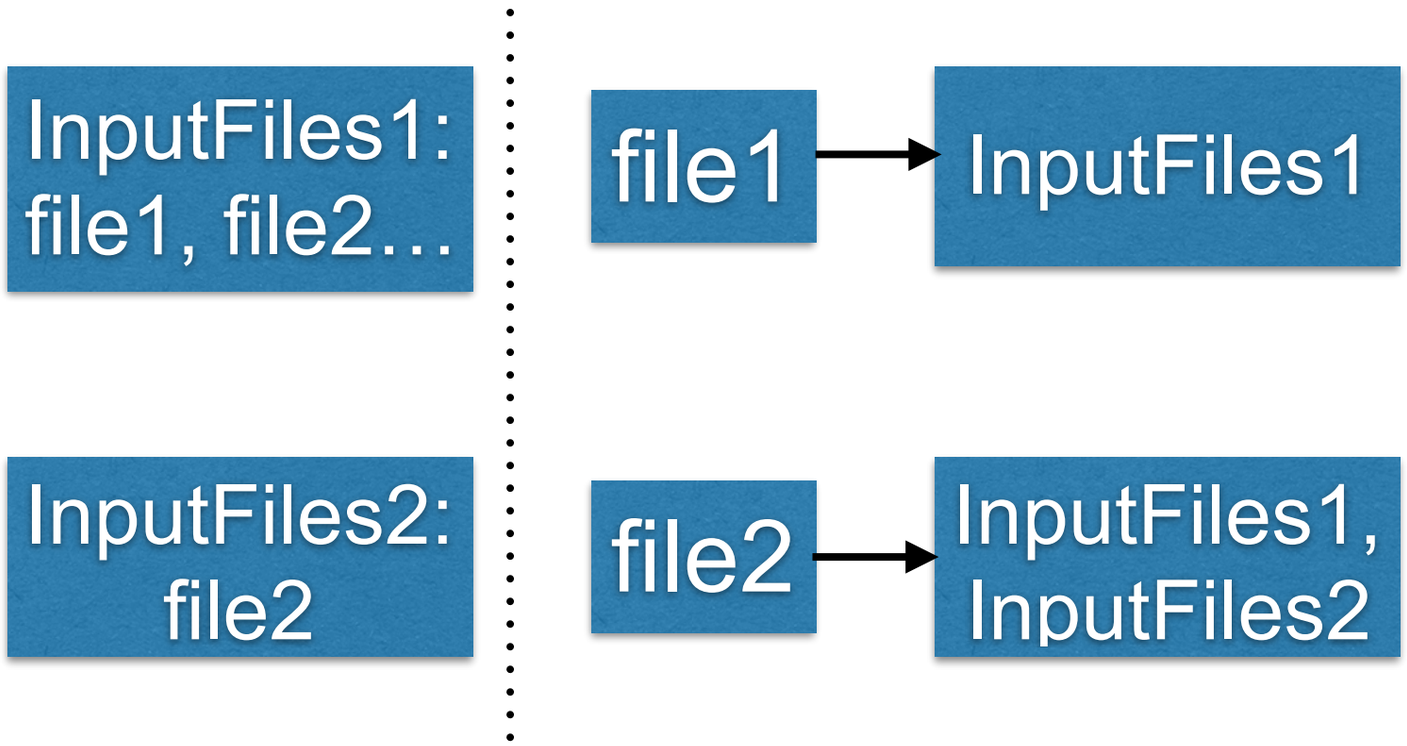
Working folder is selected by Vobox CE service before batch submission (for example, using a simple round-robin)

Titan job service

- persistent service, originated from JobAgent implementation in Java
- operates with multiple jobs in PBS batches
- does not follow the usual ALICE JobAgent workflow completely:
  - spawns multiple connections to Central Services
  - own component for input file download
  - takes completed jobs from the separate stage-out folder when all of the downloads for just started batches finished

File Download

- job input files are fetched at the same moment of time, possible situations to spawn thousands xrdcp processes
- each file from InputFile element in each JDL fetched is downloaded only once, then it is copied from file cache to job working folder
- can be useful for environments with limited network bandwidth



Job wrapper

- implemented in Bash
- communicates with the batch databases, reads information about job to be run, reports to the database about job exit codes
- sets the environment variables, starts the executable and validation scripts
- reads resource stats for the job being executed
- provides resource stats in format suitable for sending to AliEn Central Services
- instead of real “uploading” files for done jobs makes hard links into the special “stage-out folder” (takes around 0.5 sec for all of the files)
- can be used with other workflow management systems like PanDa

ALICE simulation software

- managed to build needed software for Titan (thanks to Dario Berzano and ALICE build masters)
- no special compiler directives were used to build software
- there is a CVMFS repository subset on Titan, updated every hour (mounted under other folder than /cvmfs)
- publisher script had to be brushed up because Titan was cutting too frequent outbound network connections
- now trying to set up publisher script for NERSC Cori

ALICE simulation software challenges

- work on network calls removal succeeded, now the jobs communicate only with snapshots (took a lot of time from July to end of October to fix it for QA and AOD stages)
- network leaks were encountered in QAtrainsim.C and AODtrainsim.C, fixed, merged into master branch
- there was a discussion concerning compilation for ACLiC by HLT on validation stage, may be continued in the future

Running ALICE MC jobs on Titan

- Tried with:

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
- with Pb-Pb jobs we can not participate in pure backfill (usually less than 2 hours), CSC108 project has the lowest priority (-527911), thus, Titan tools do not show accurate info about start time
- quite successful in requesting 125 nodes/5:45h slots which can be ok for 2 events (theoretically up to 10 slots per day)
- does not seem too problematic to get 125 nodes/3:30 slots (max waiting time observed - around 3 hours)
- more suitable for running p-p MC jobs (no benchmarks yet due to a problem on simulation stage)

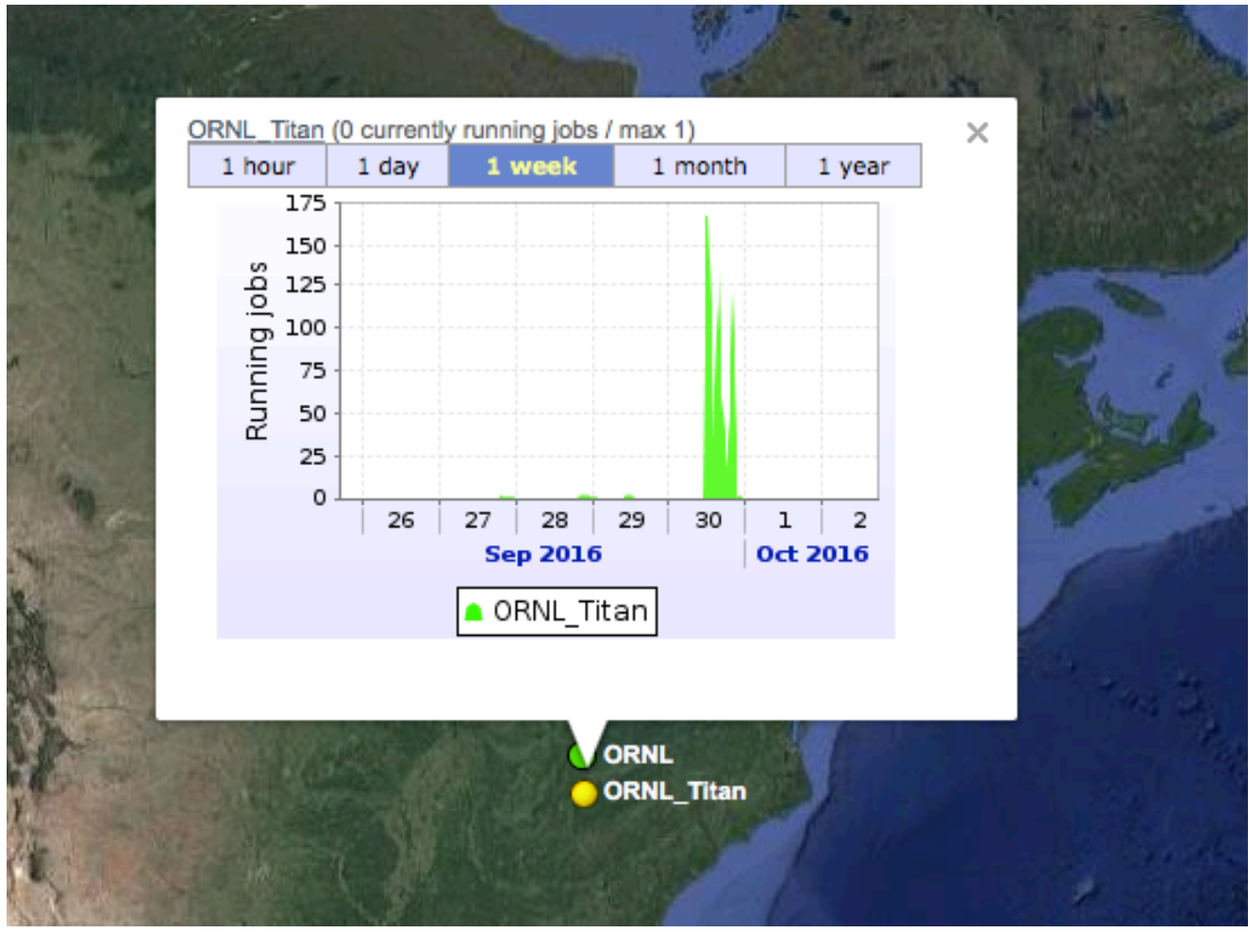
Overhead estimation for starting 2000 jobs from the same masterjob

- Downloading 2 snapshot files around 130Mb: ~2 mins (downloads run in parallel)
- Copying 2 snapshot files to job working folder on LustreFS: ~0.4 sec
- Fetching 2000 JDLs: ~4 mins
- **Total time (worst case) : 4*60 + 2*60 + 0.4*2000 = ~1160 sec = ~20 mins**
- Making hard link for a 2 files instead of copying: 0.006 sec
- If we use recommendation for LustreFS “1 file per 1 node”:
- **Total time: 4*60 + 2*60 + 0.4*2000/16 + 0.006*2000 = ~422 sec = 7 mins**

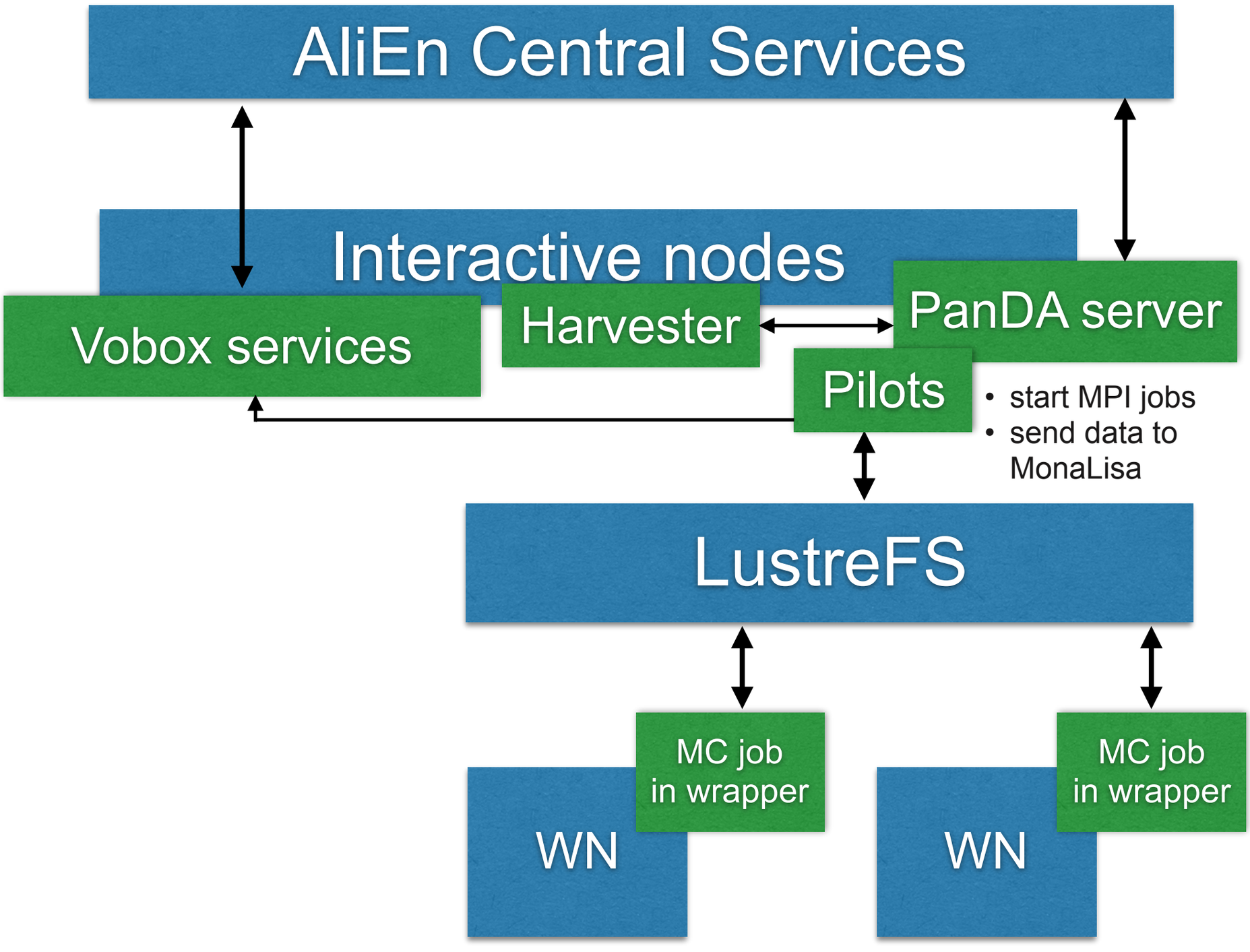
Overhead can be optimized by:

- using local storage element for xrootd (e.g. on CADES) or running service on a DTN node
- extending JDL fetching from central queue from “give me 1 job” to, e.g., “give me 50 jobs”

Titan in ALICE monitoring system



PanDA integration and Pilot2 (draft design)



- 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

Submitted jobs to Titan, Pilot 1.0

PanDA jobs

pandawms.org/dev/jobs/?jobstatus=finished

ВКонтакте | mail.ru почта | LostFilm.TV. | Mail Services | JINR Webmail | RuTracker | iTop Login | Battle | Сериал Теория | PanDA PanDA | BigPanDAinstancel

Job list Sort by PandaID, ascending mod time, descending mod time, priority, attemptnr, duration

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										
	Andrey Kondratyev	1	mpi_wrapper_alice_A01alicegeo.py	finished	2015-10-	23:7:16:20	0:0:02:54	11-	ANALY_ORNL_Titan	2000	

Conclusions and future work

- a new generation of ALICE Grid software running on Titan (however, needs more testing)
- Titan marked as functioning in our monitoring system (more testing needed)
- we managed to remove network calls from simulation software
- takes a long time to process Pb-Pb jobs, benchmark needed for p-p
- payload JDLs have to be discussed taking into account the benchmarks, current and future data specifics
- started a closer interaction with PanDA team

THANK YOU!