HTC and HPC as a parts of distributed computing system

Danila Oleynik

HTC AND HPC: WHERE IS DIFFERENCES?

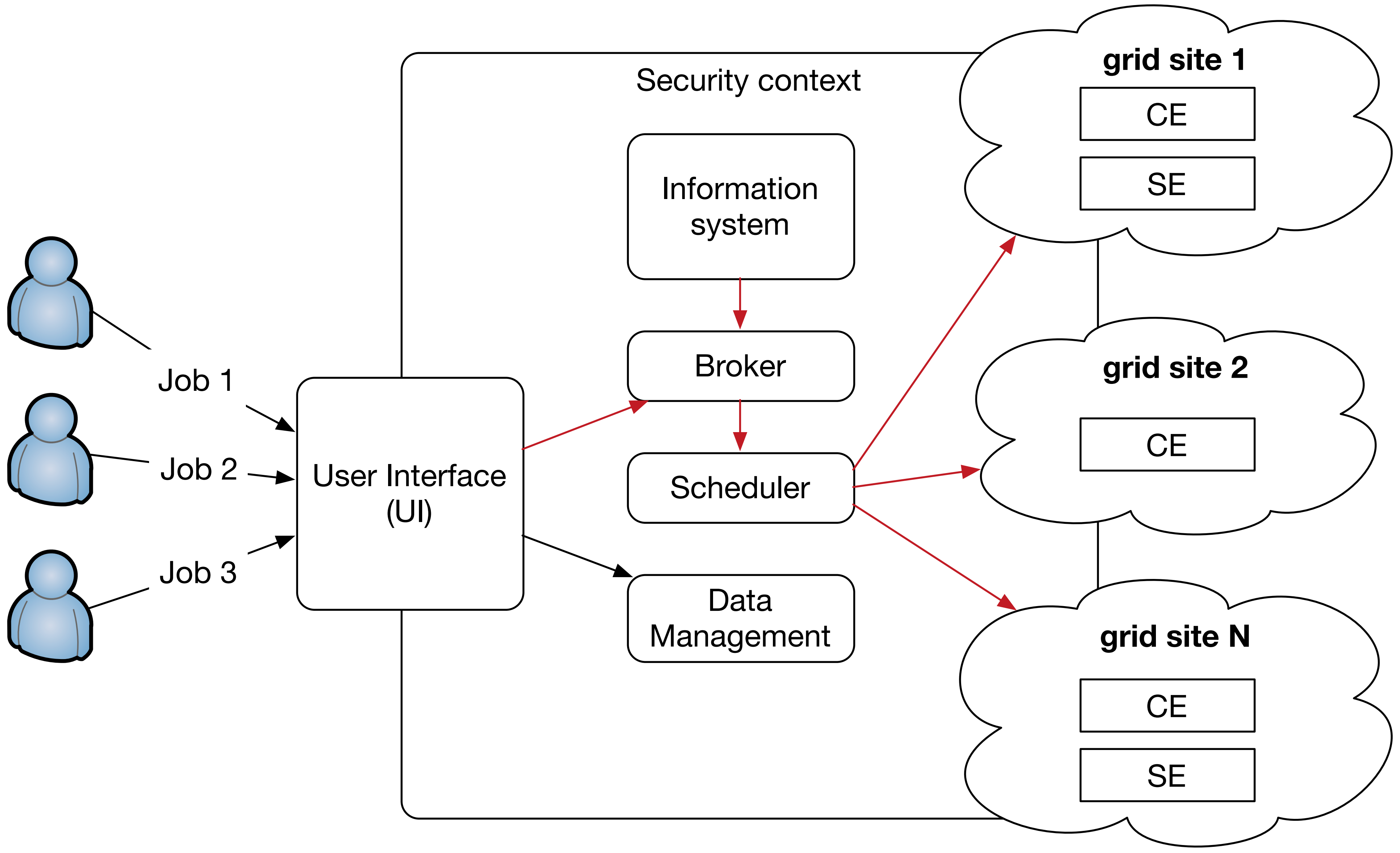
HPC tasks are characterized as needing large amounts of computing power for short periods of time, whereas HTC tasks also require large amounts of computing, but for much longer times (months and years, rather than hours and days). HPC environments are often measured in terms of FLOPS.

The HTC community, however, is not concerned about operations per second, but rather operations per month or per year. Therefore, the HTC field is more interested in how many jobs can be completed over a long period of time instead of how fast an individual job can complete.

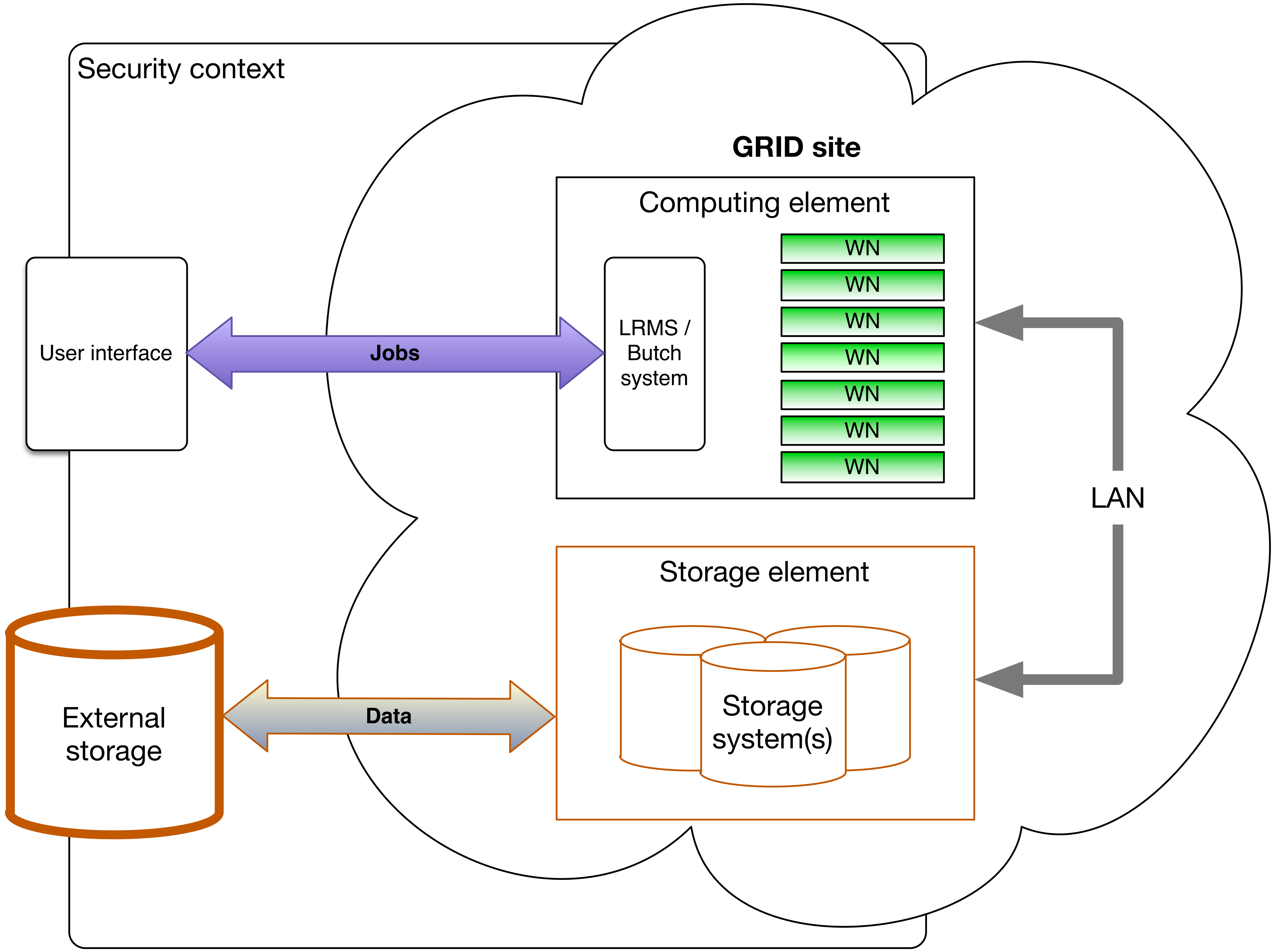
As an alternative definition, the European Grid Infrastructure defines HTC as "a computing paradigm that focuses on the efficient execution of a large number of loosely-coupled tasks", while HPC systems tend to focus on tightly coupled parallel jobs, and as such they must execute within a particular site with low-latency interconnects. Conversely, HTC systems are independent, sequential jobs that can be individually scheduled on many different computing resources across multiple administrative boundaries. HTC systems achieve this using various grid computing technologies and techniques.

HTC: GRID COMPUTING

- ***Grid computing*** *is the collection of computer resources from multiple locations to reach a common goal. The grid can be thought of as a distributed system with non-interactive workloads that involve a large number of files. Grid computing is distinguished from conventional high performance computing systems such as cluster computing in that grid computers have each node set to perform a different task/application.*



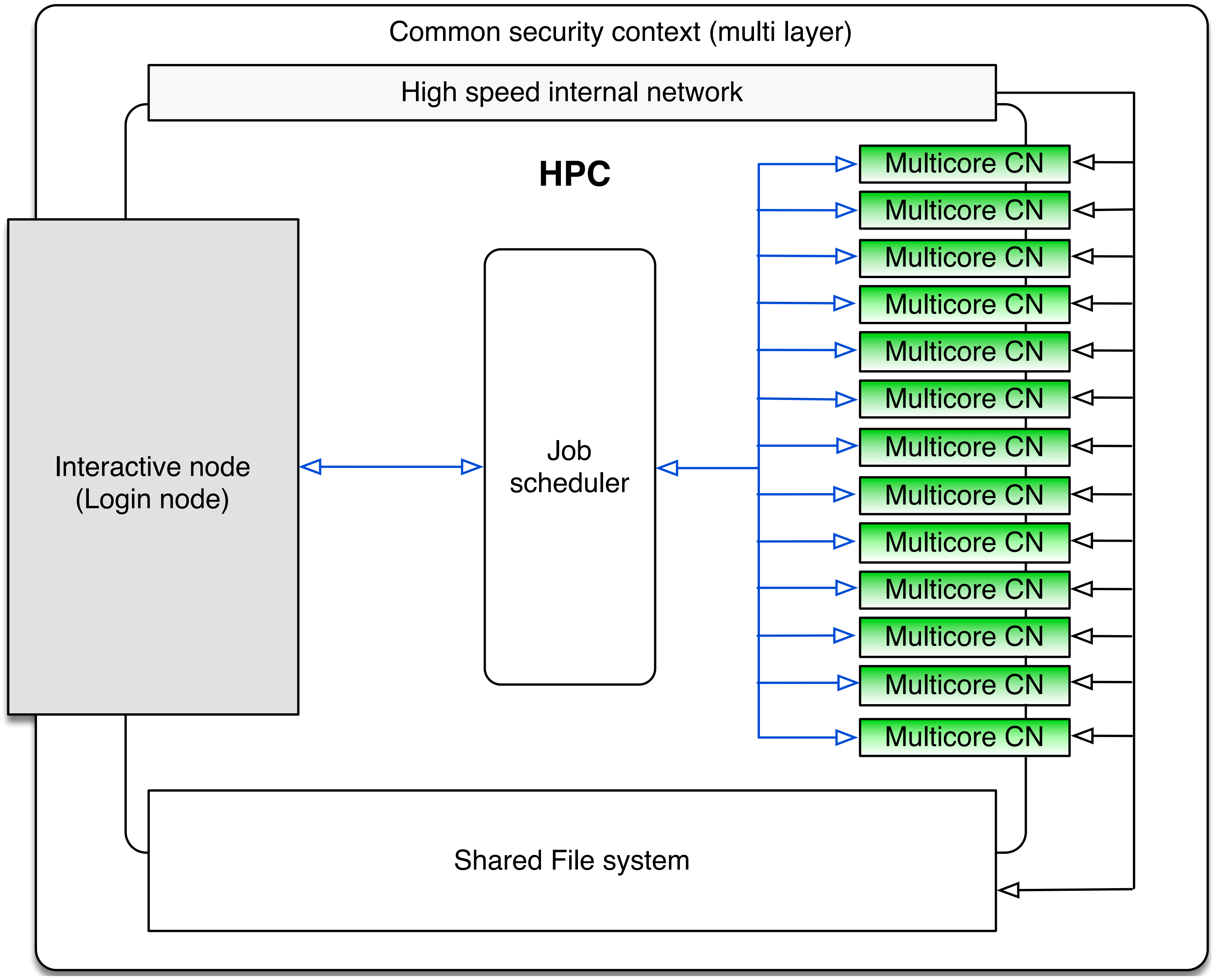
GRID SITE



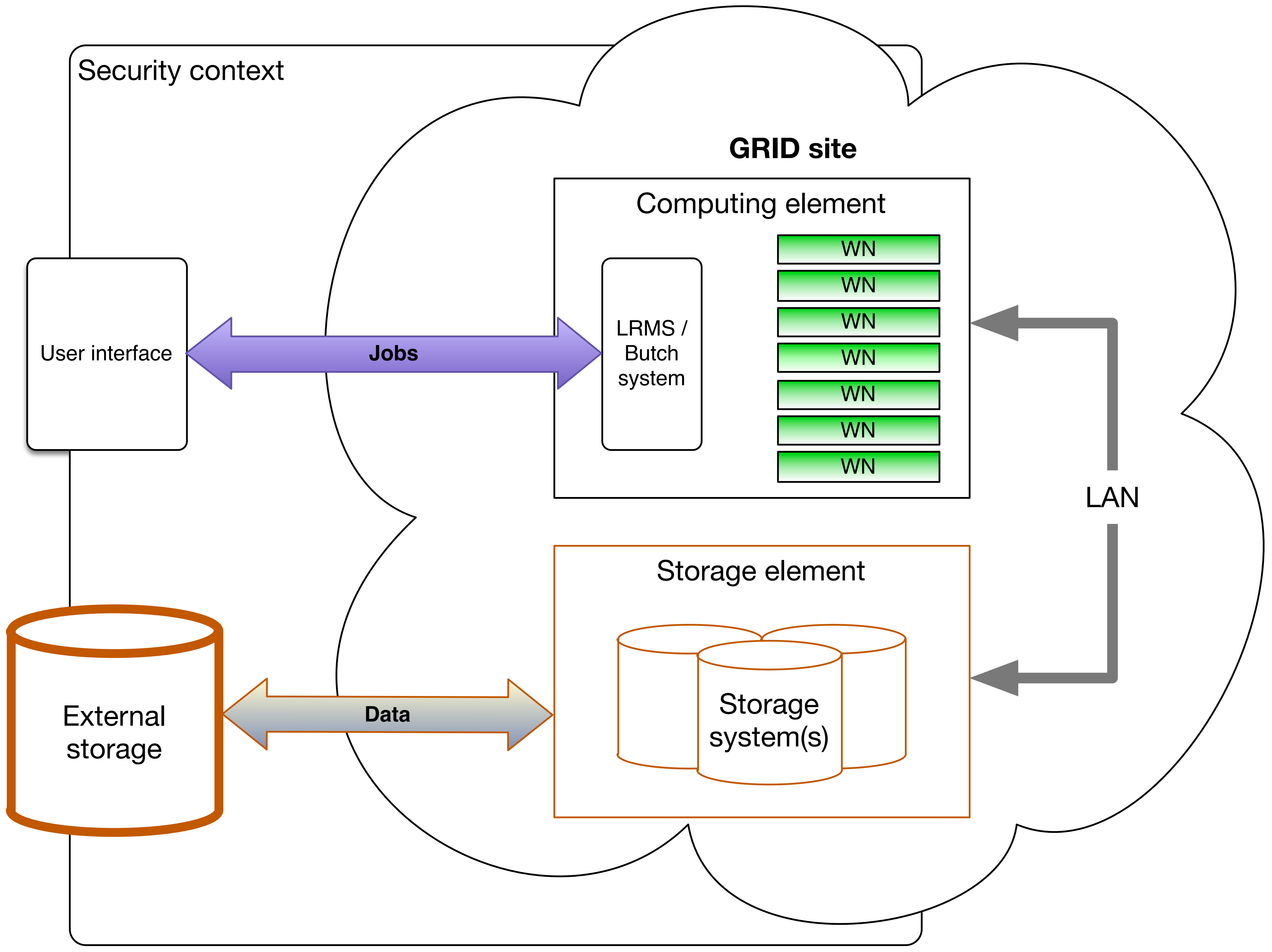
Security context
GRID site
Computing element
User interface
Jobs
LRMS / Butch system
WN
WN
WN
WN
WN
WN
WN
LAN
Storage element
External storage
Data
Storage system(s)

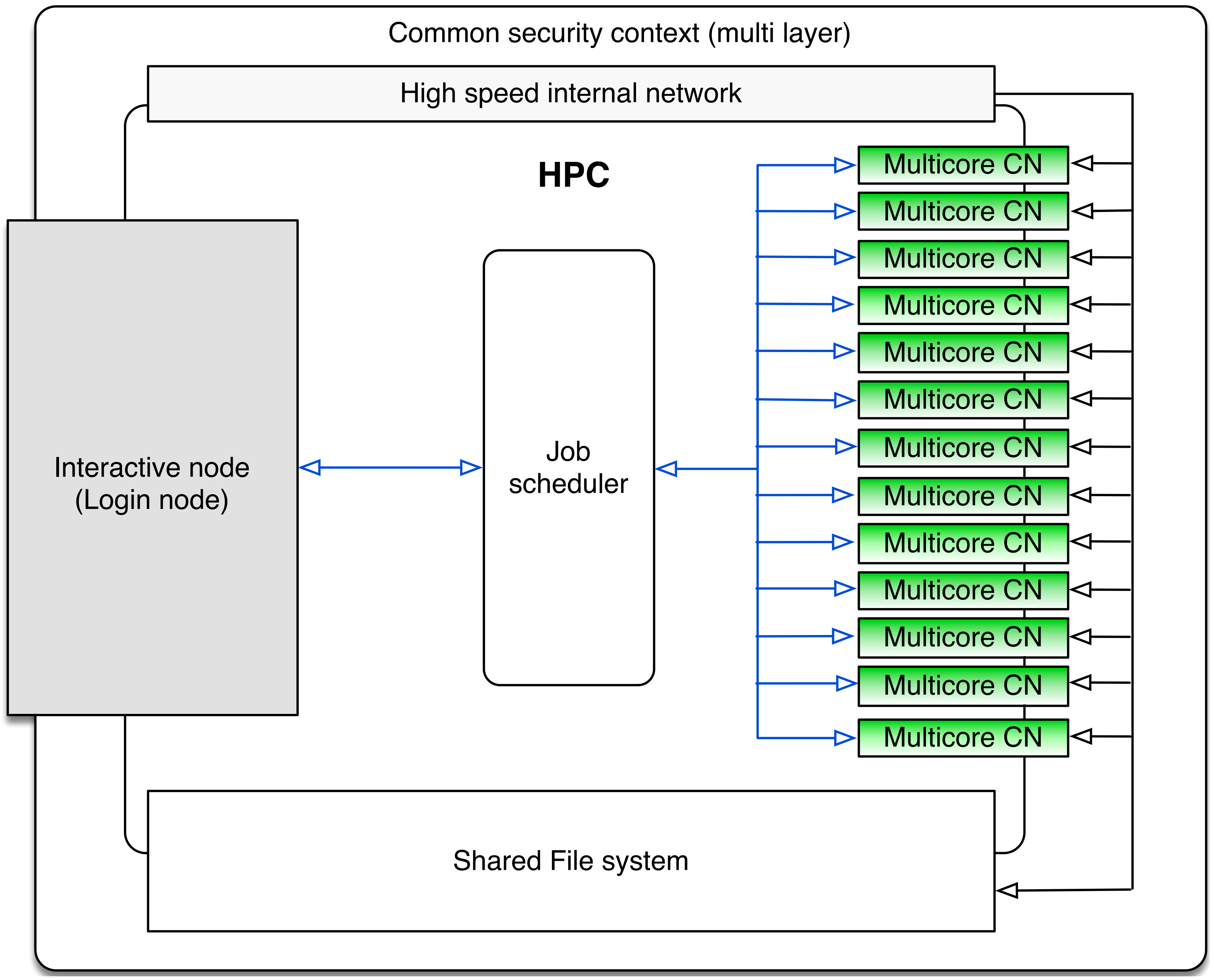
HPC (SUPERCOMPUTERS)

- A **supercomputer** is a computer with a high-level computational capacity compared to a general-purpose computer. Performance of a supercomputer is measured in floating-point operations per second (FLOPS) instead of million instructions per second (MIPS). Supercomputers which can perform up to (hundreds of) quadrillions of FLOPS already existed and we are on the way to quintillion of FLOPS (ExaFLOP).



SIMILAR SCHEMAS?





But very different policies…

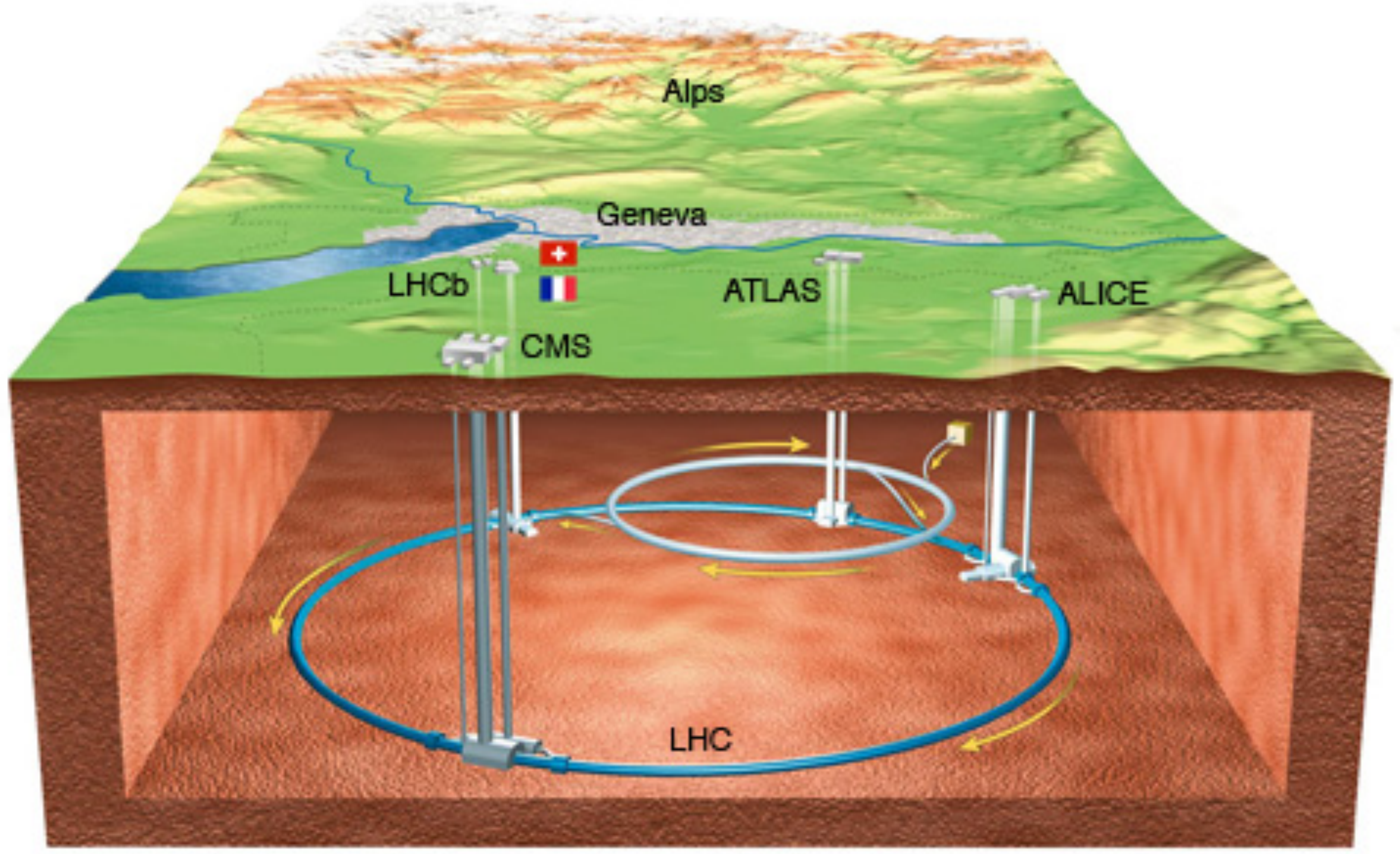
- *GRID Sites - hetorogenuose architectures, with static size of job slot*
- *HPC - homogenous architecture with dynamic size of job slot*

WLCG

The Worldwide LHC Computing Grid (WLCG) project is a global collaboration of more than 170 computing centers in 42 countries, linking up national and international grid infrastructures.

The mission of the WLCG project is to provide global computing resources to store, distribute and analyse petabytes of data annually generated by the Large Hadron Collider (LHC) at CERN

The Large Hadron Collider (LHC) is the world's largest and most powerful particle accelerator. It first started up on 10 September 2008, and remains the latest addition to CERN's accelerator complex.



WLCG

42 countries
170 computing centres
2 million tasks run every day
800,000 computer cores
500 petabytes on disk and 400 petabytes on tape

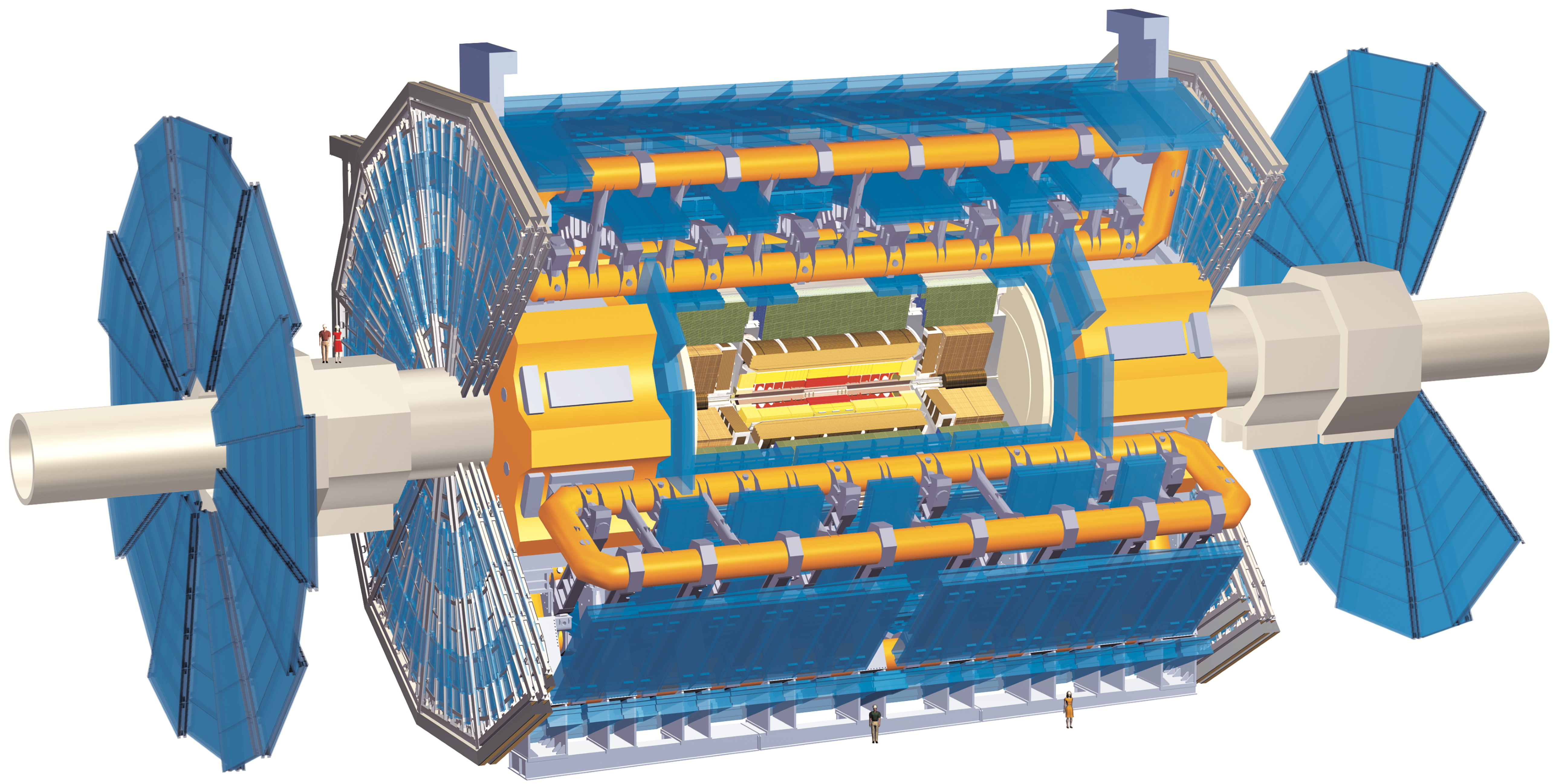
RMAX (TFLOP/S)	RPEAK (TFLOP/S)
17173.2	20132.7

TOP 500 The List.

2018	#7
2017	#4
2016	#3
2015	#2
2014	#2
2013	#2
2012	#1

Titan System Configuration	
Architecture:	Cray XK7
Processor:	16-Core AMD
Cabinets:	200
Nodes:	18,688 AMD Opterons
Cores/node:	16
Total cores:	**299,008** Opteron Cores
Memory/node:	32GB + 6GB
Memory/core:	2GB
Interconnect:	Gemini
GPUs:	18,688 K20X Keplers
Speed:	27 PF
Square Footage	4,352 sq feet

THE ATLAS EXPERIMENT AT THE LHC



The ATLAS Detector	
Diameter:	25 m
Length:	46 m
Barrel toroid	26 m
Overall weight:	7000 tonnes
~ 100 million electronic channels	
~ 3 000 km of cables	

THE ATLAS EXPERIMENT AT THE LHC

The Nobel Prize in Physics 2013

Photo: A. Mahmoud
François Englert
Prize share: 1/2

Photo: A. Mahmoud
Peter W. Higgs
Prize share: 1/2

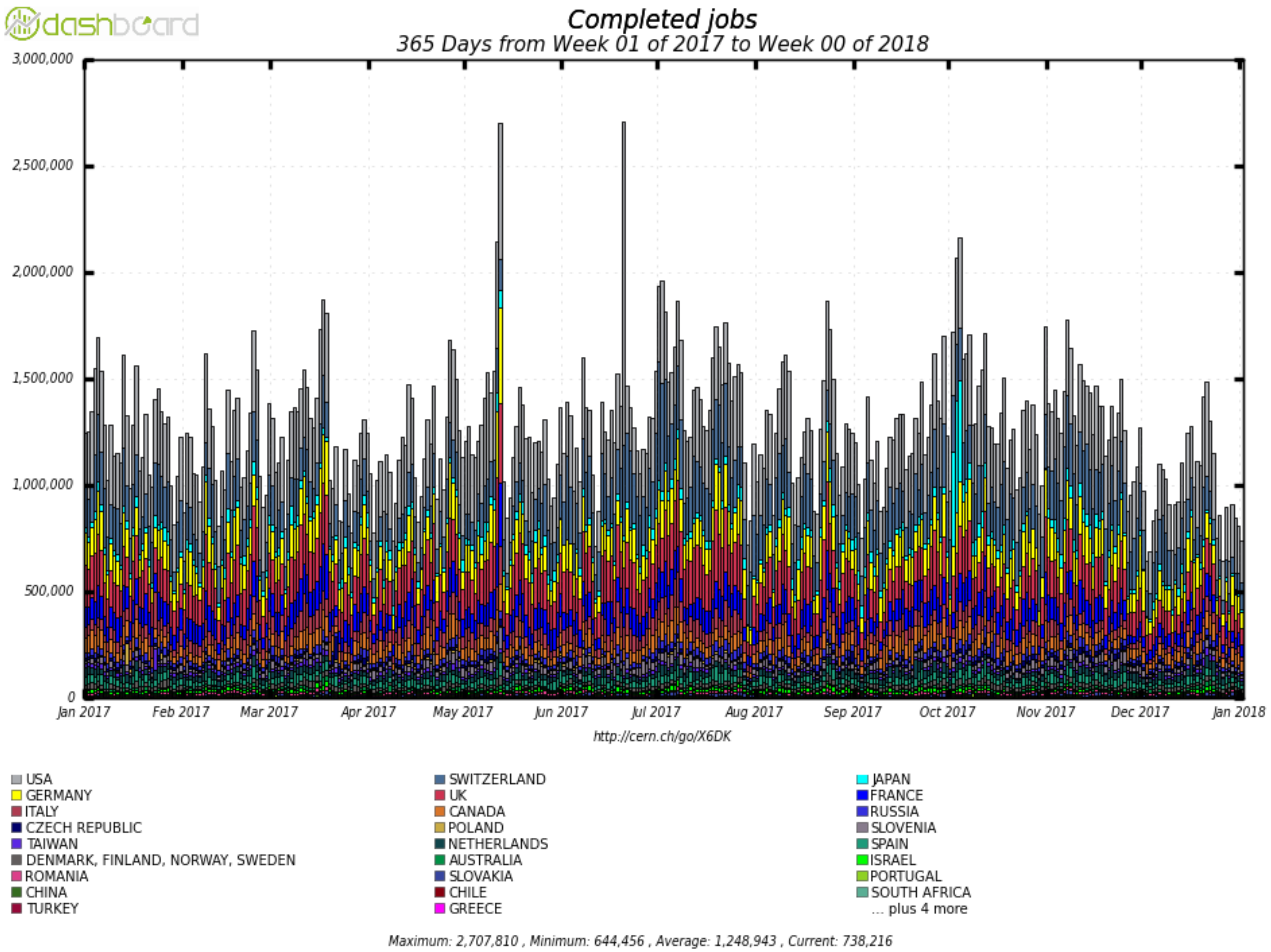
The Nobel Prize in Physics 2013 was awarded jointly to François Englert and Peter W. Higgs *"for the theoretical discovery of a mechanism that contributes to our understanding of the origin of mass of subatomic particles, and which recently was confirmed through the discovery of the predicted fundamental particle, by the ATLAS and CMS experiments at CERN's Large Hadron Collider"*

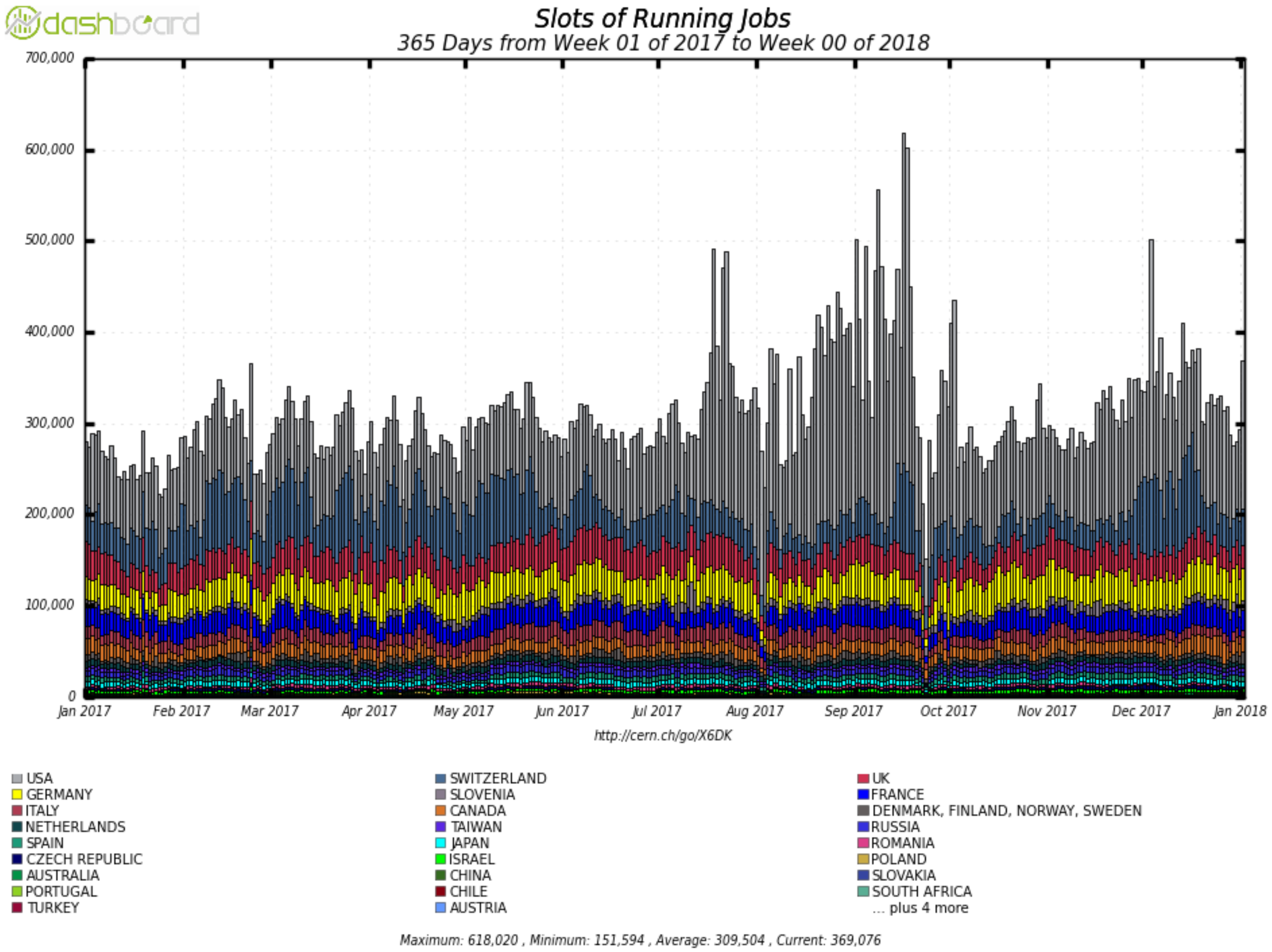
3000 scientists

38 countries

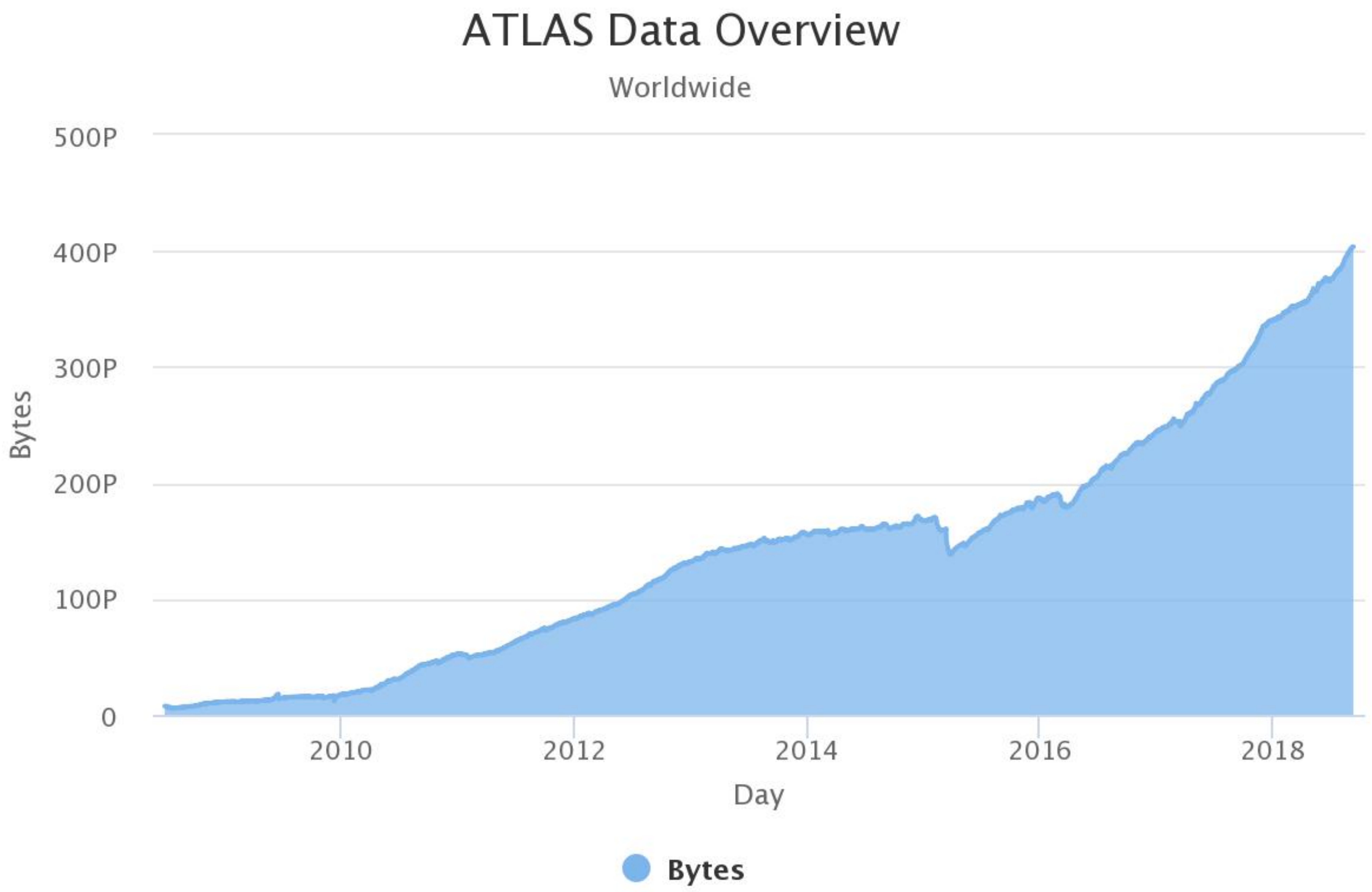
174 universities and

GRID production for ATLAS





BIG DATA: OFTEN JUST A BUZZ WORD, BUT NOT WHEN IT COMES TO ATLAS…



Current ATLAS data set, considering all data products and replicas

>400 PB

PanDA

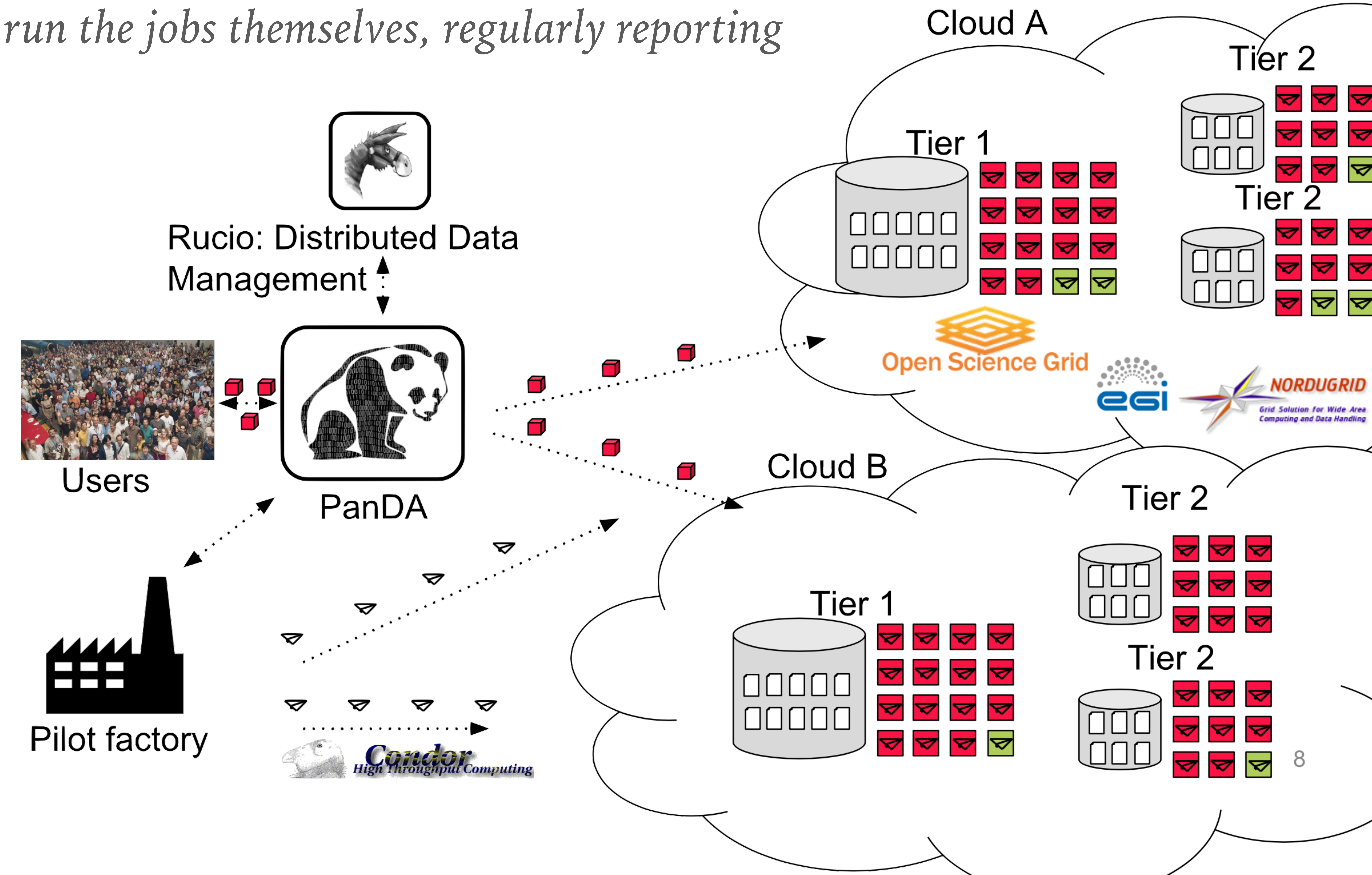
*The **PanDA P**roduction **AN**d **D**istributed **A**nalysis system has been developed by ATLAS since summer 2005 to meet ATLAS requirements for a data-driven workload management system for production and distributed analysis processing capable of operating at LHC data processing scale. ATLAS processing and analysis places challenging requirements on throughput, scalability, robustness, efficient resource utilization, minimal operations manpower, and tight integration of data management with processing workflow.*

PanDA throughput has been rising continuously over the years. In 2009, a typical PanDA processing rate was 50k jobs/day and 14k CPU wall-time hours/day for production at 100 sites around the world, and 3-5k jobs/day for analysis. In 2017, PanDA processes about a 1,2M jobs per day, with about 120,000 jobs running at any given time. The PanDA analysis user community numbers over 1400.

PanDA

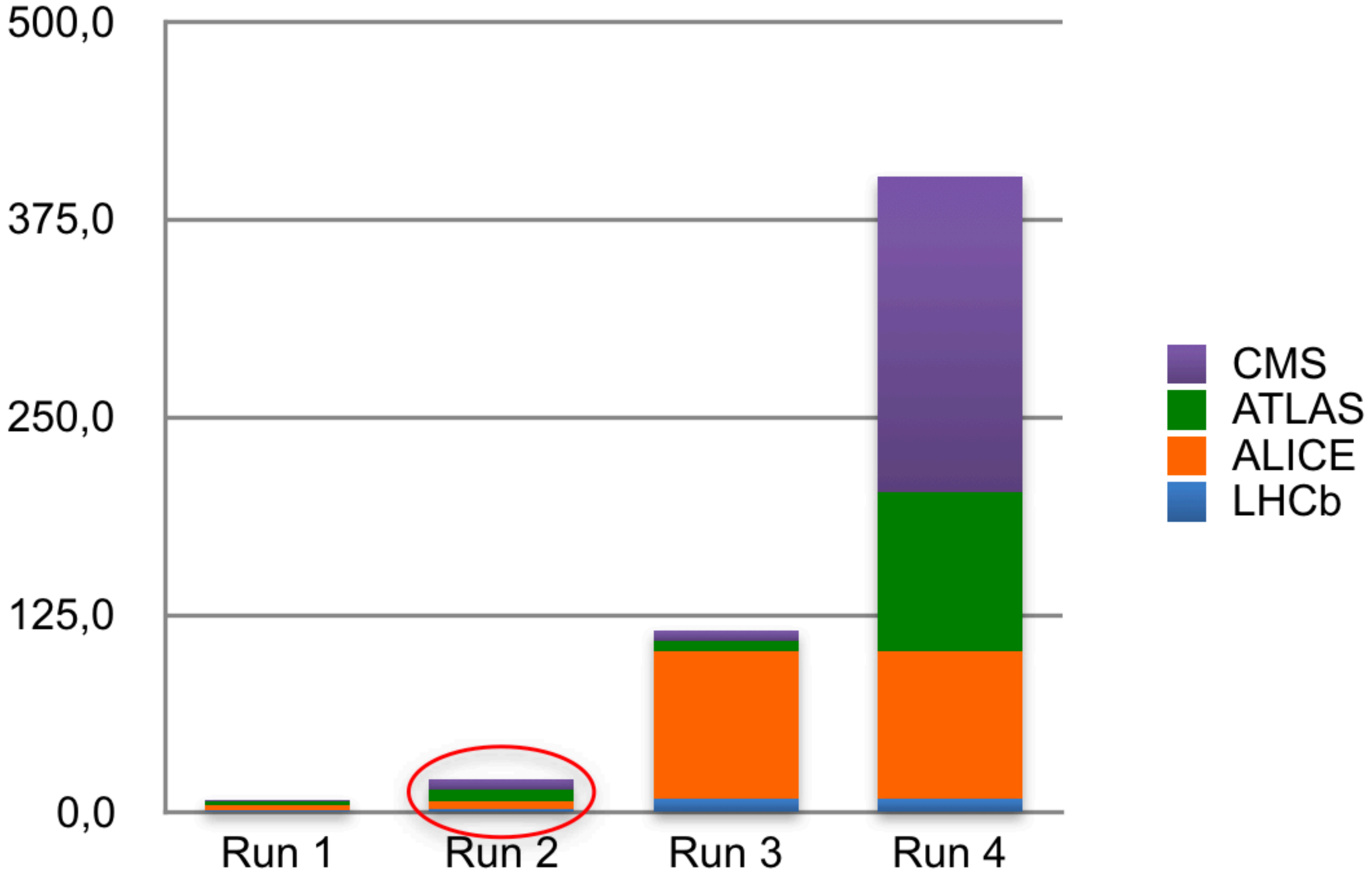
PanDA Pilot - the execution environment (effectively a wrapper) for PanDA jobs. Pilots request and receive job payloads from the dispatcher, perform setup and cleanup work surrounding the job, and run the jobs themselves, regularly reporting status to PanDA during execution

The interest in PanDA by other big data sciences brought the primary motivation to generalize PanDA, aka the BigPanDA project, providing location transparency of processing and data management, for High Energy Physics community and other data-intensive sciences, and a wider exascale community.



LHC UPGRADE 2019-2021. COMPUTING NEEDS

- CPU needs (per event) will grow with track multiplicity (pileup) and energy
- Storage needs are proportional to accumulated luminosity
- Grid resources are limited by funding and fully utilized

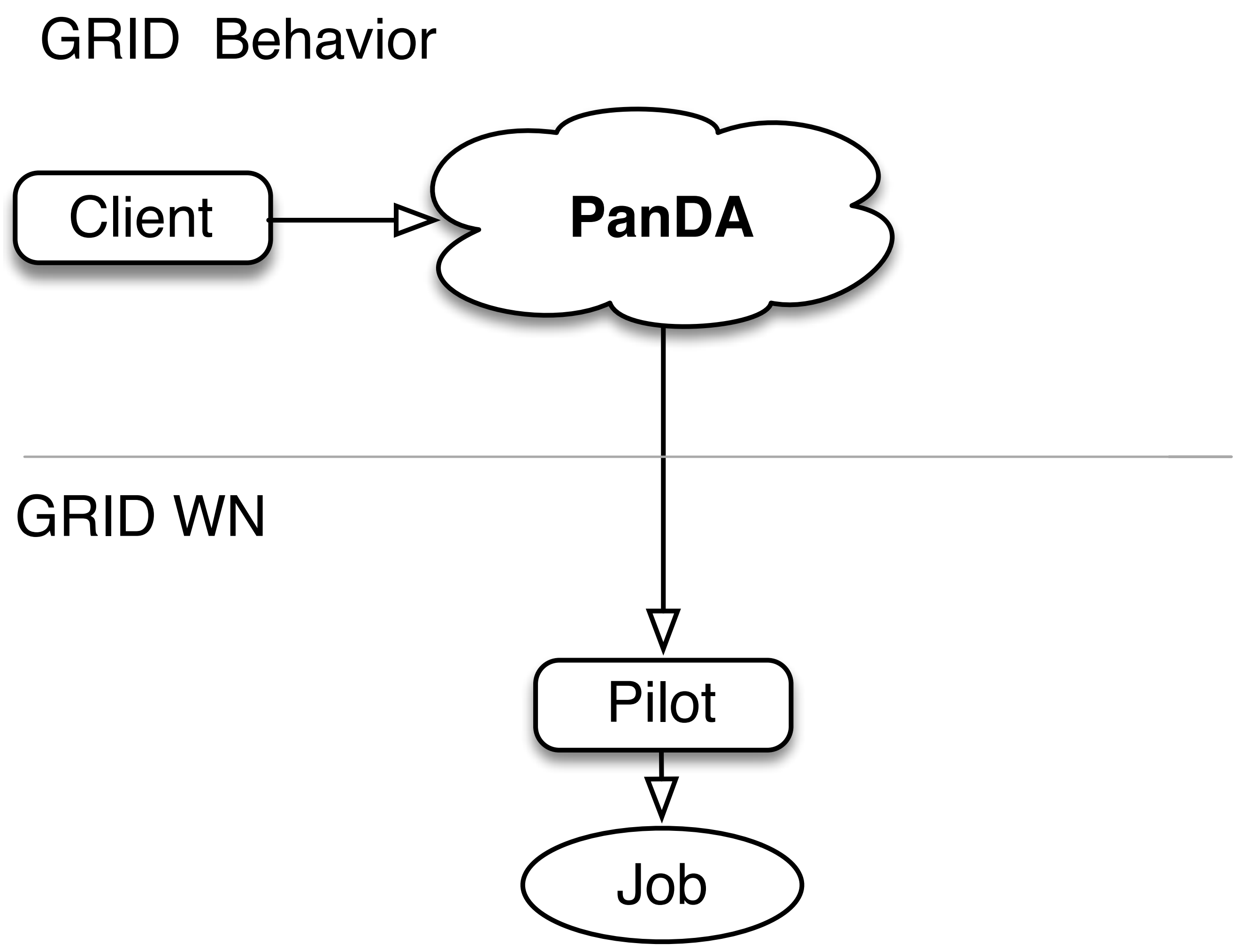


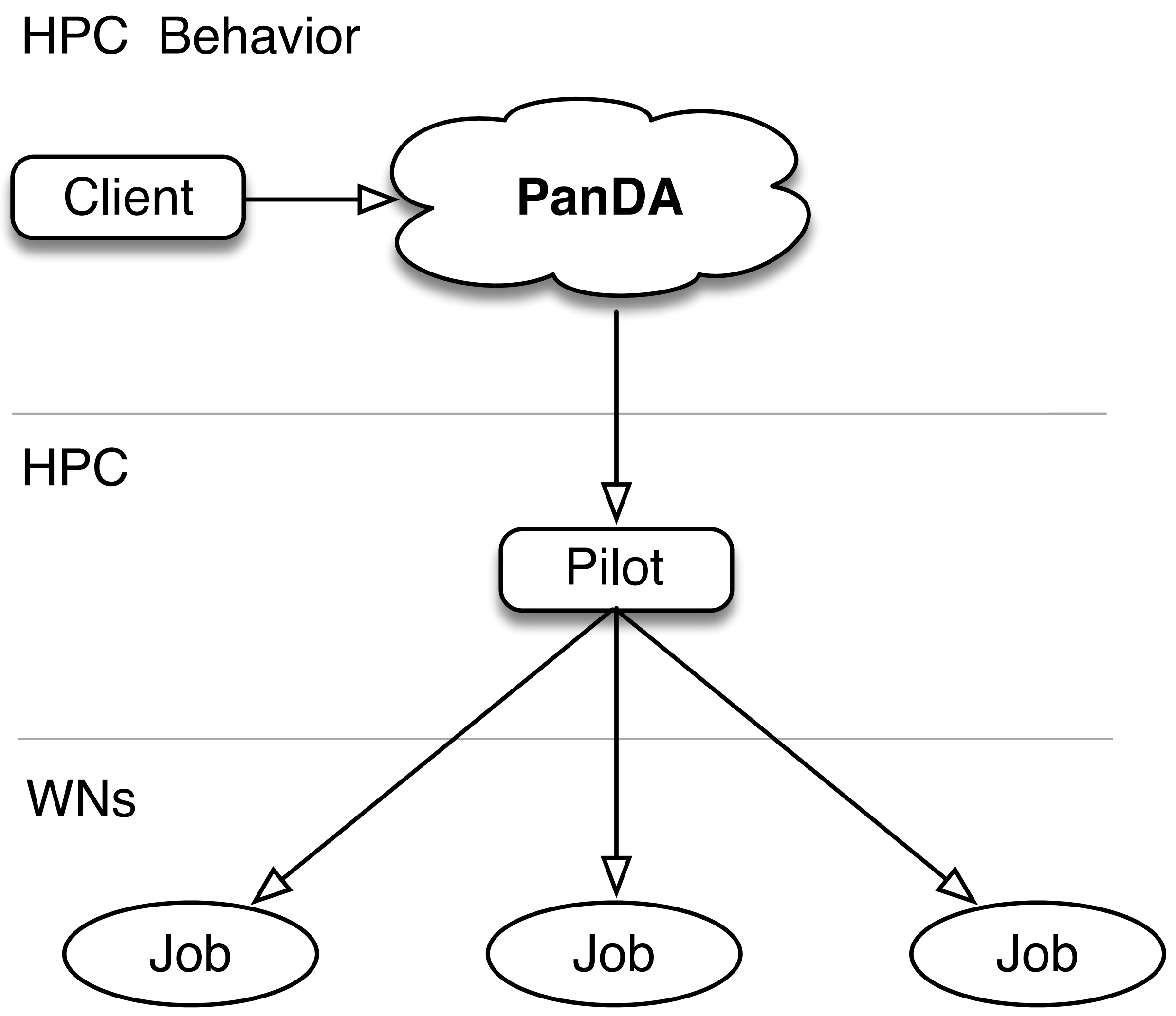
- Highly restricted access. One-time password interactive authentication
  - No portals, gatekeepers. Pilot needs to run on Titan's interactive nodes
- No network connectivity from worker nodes to the outside world
  - Pilot can not run on worker nodes, needs a new mechanism for batch workload management
- Limit on number of submitted jobs in batch queue per user and limit on number of running jobs per user
  - Sequential submissions of single node jobs is not an option
  - Have to use MPI in some form!
- Specialized OS (SUSE based CNL) and software stack
- Highly competitive time allocation. Geared toward leadership class projects and very big jobs
  - Creates opportunity for backfill

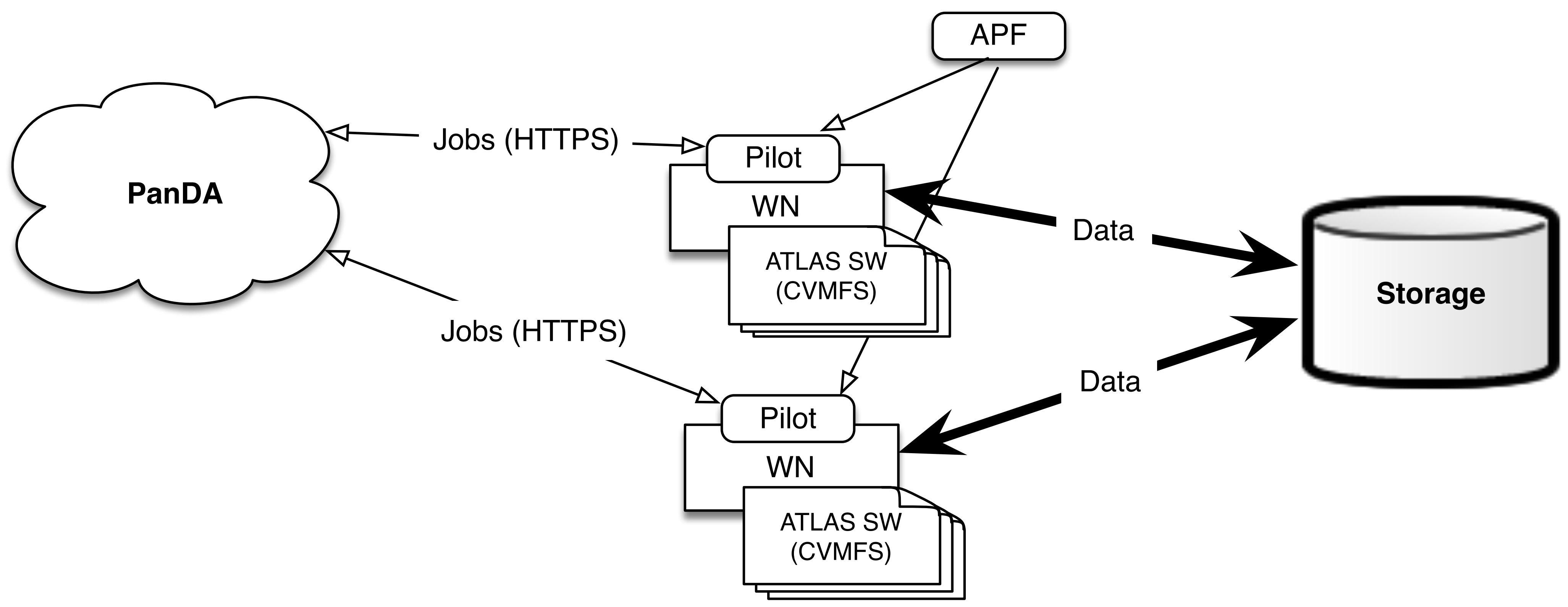
«BACKFILL»

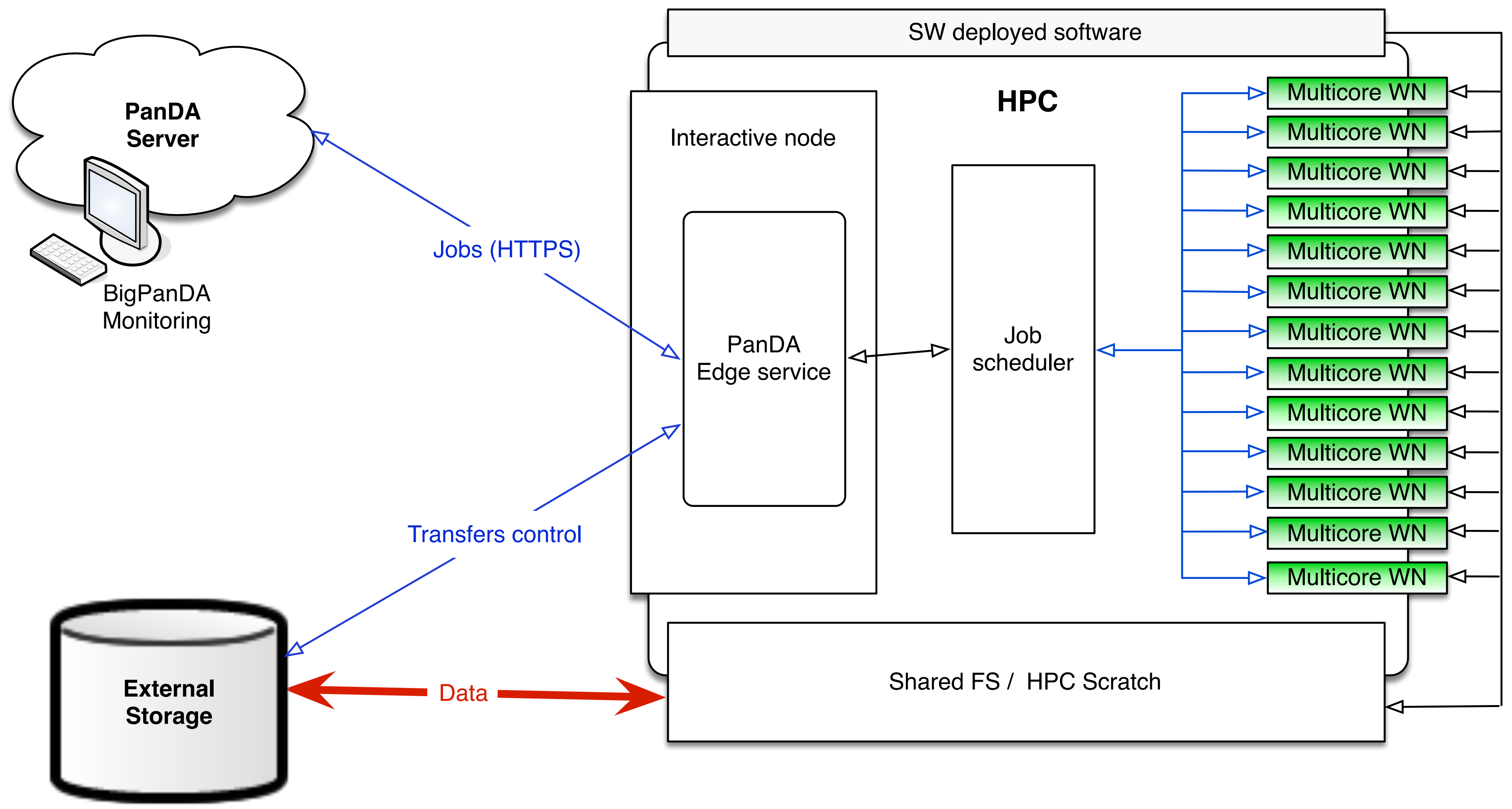
- *Typical LCF facility is ran on average at ~90% occupancy*
  - *Necessary outcome of prioritizing large jobs execution*
  - *On a machine of the scale of Titan that translates into ~400M unused core hours per year*
- ***Anything that helps to improve this number consistent with LCF mission is very useful***
- *PanDA Pilot was instrumented with capability to collect, in near real time, information about current free resources on Titan*
  - *Both number of free worker nodes and time of their availability*
- *Based on that information Pilot can define job submission parameters when forming PBS script for Titan, thus tailoring the submission to the available resource.*
  - *Takes into account Titan's scheduling policies*
  - *Can also take into account other limitations, such as workload output size, etc*
  - *Modular architecture, adaptable to other HPC facilities*

PanDA integration with Titan

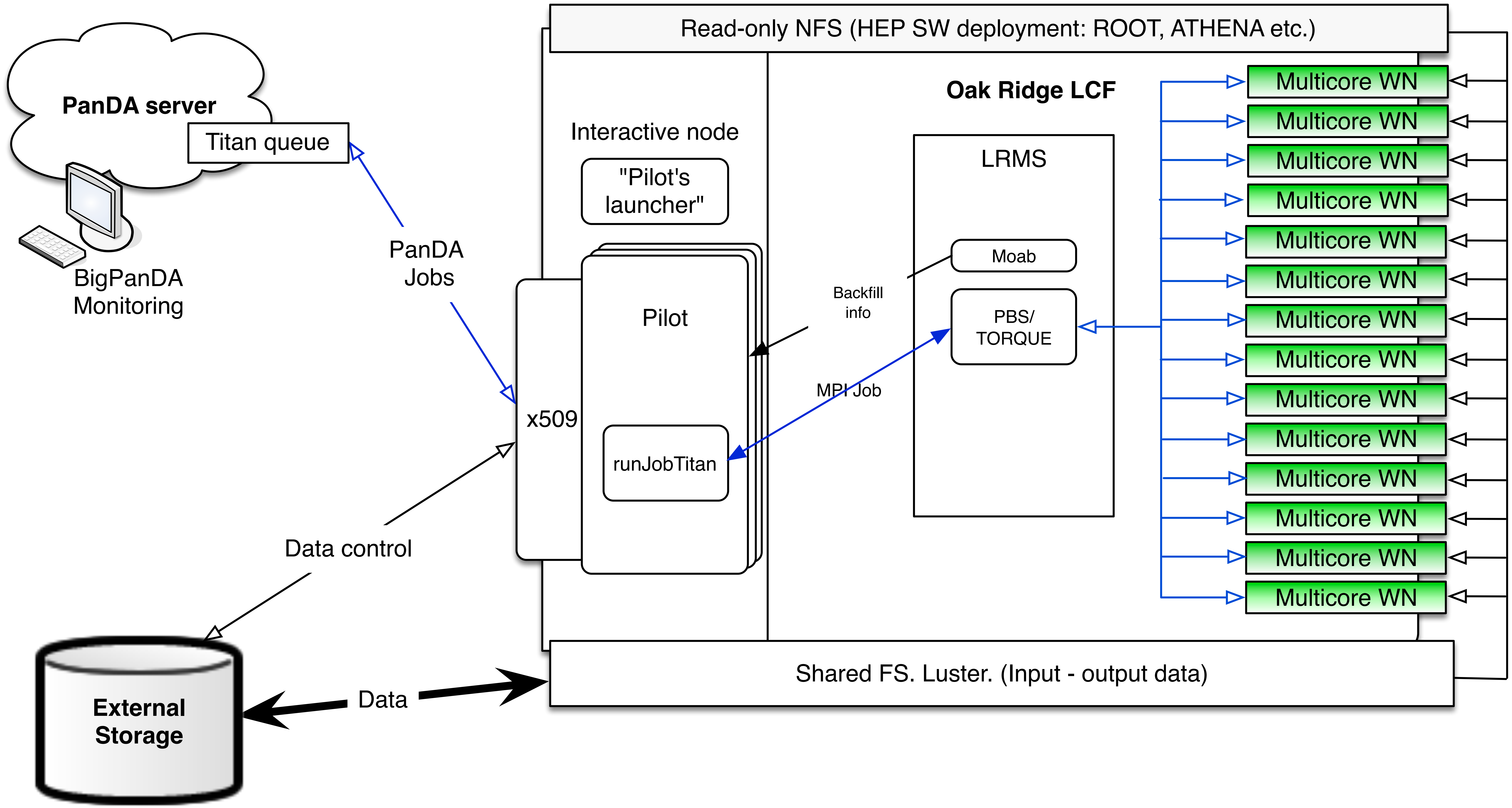




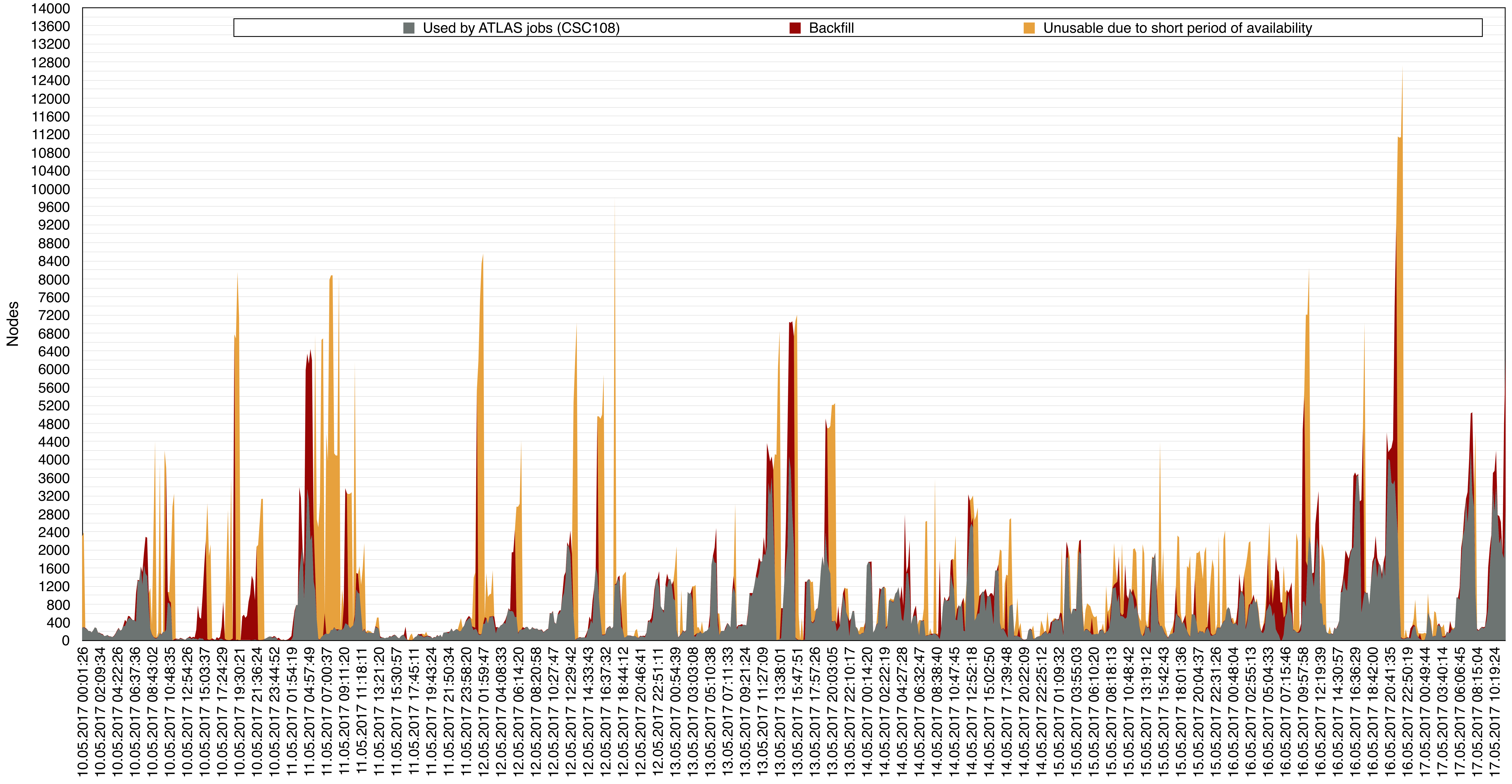




PanDA integration with Titan



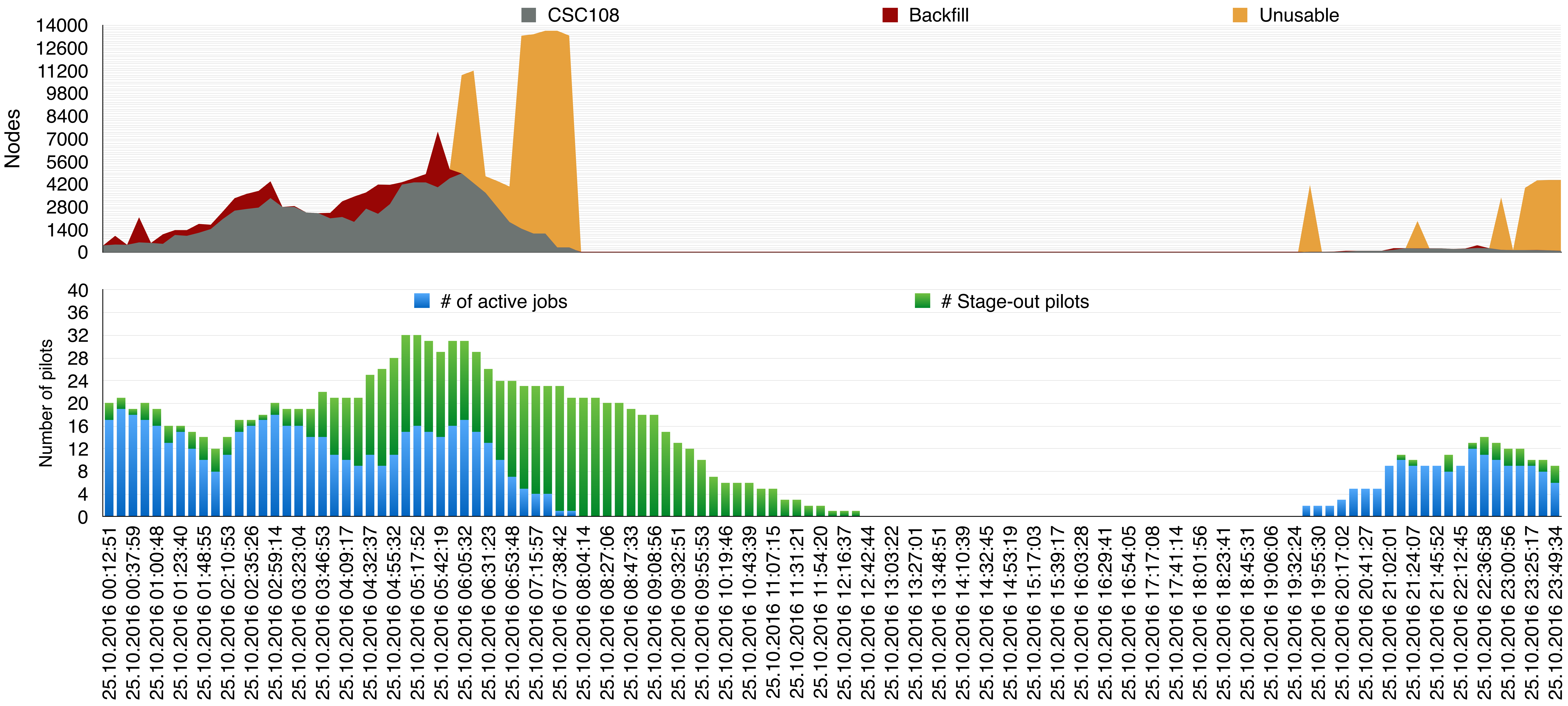
Backfill consumption on Titan



ISSUES

- Why we limit with low maximum number PanDA jobs per chunk?
  - StageIn and StageOut consecutive operations and for big amount of jobs may increase loading to data transfers infrastructure. During this operations some backfill resources will be missed
- Architecture of Pilot was not designed to server multiply simultaneous jobs at nutshell
  - Design of Pilot core was done at the begging of project and did not had major changes in decades

ISSUES



CSC108
Backfill
Unusable
Nodes
# of active jobs
Stage-out pilots
Number of pilots

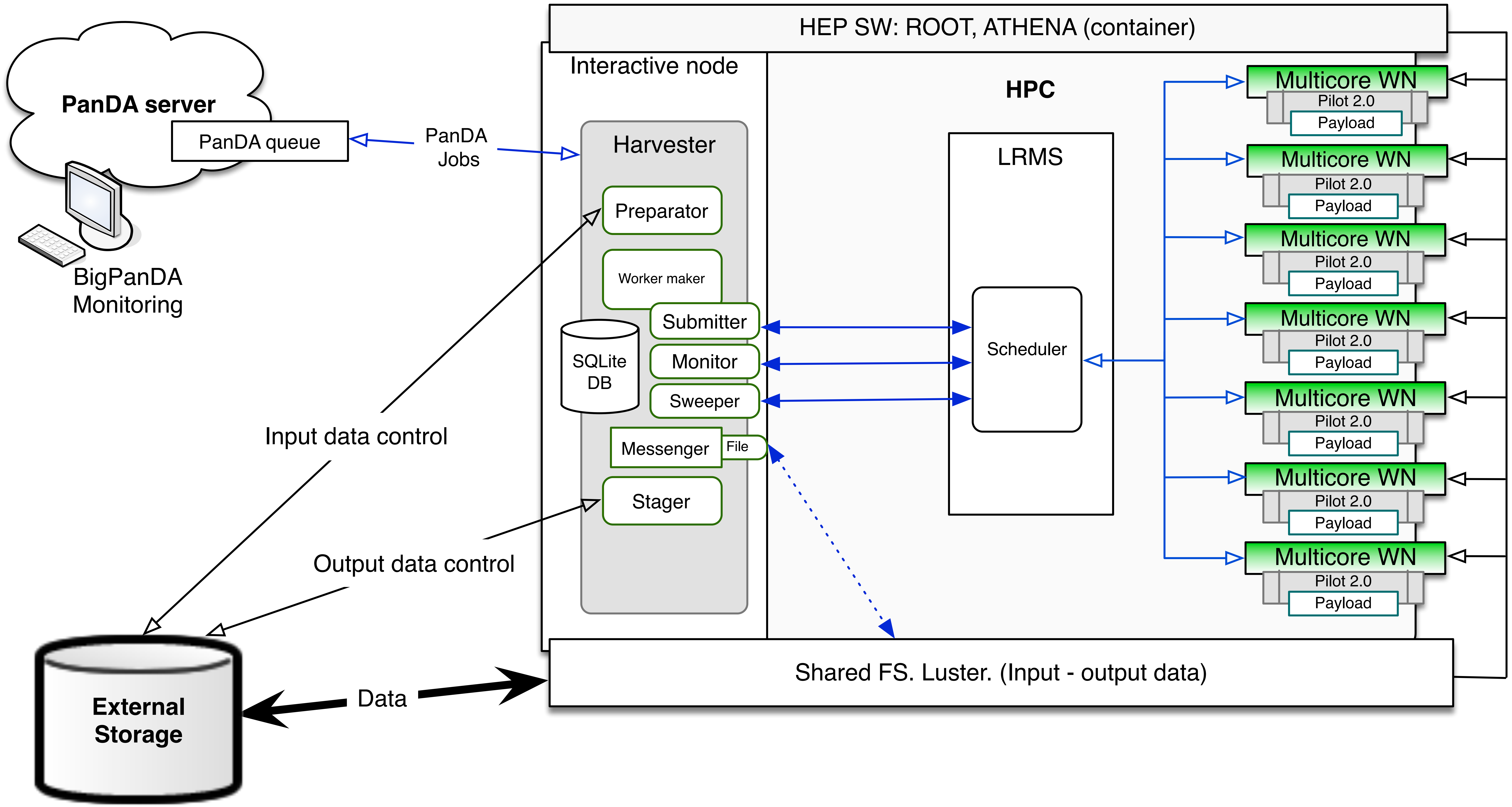
EVALUATION OF PANDA: PILOT 2.0

- Pilot 2.0 Motivation:
  - Some of the Pilot 1.0 code base is getting a bit too old and is difficult to maintain
    - Refactoring is a slow process that has already been going on for years and does not always have highest priority
    - New features/workflows are often challenging to implement/support
    - "Complete" rewrite
  - Getting rid of all legacy code and outdated mechanisms
  - Rethink of basic pilot flow
  - New PanDA Pilot Project launched in April 2016

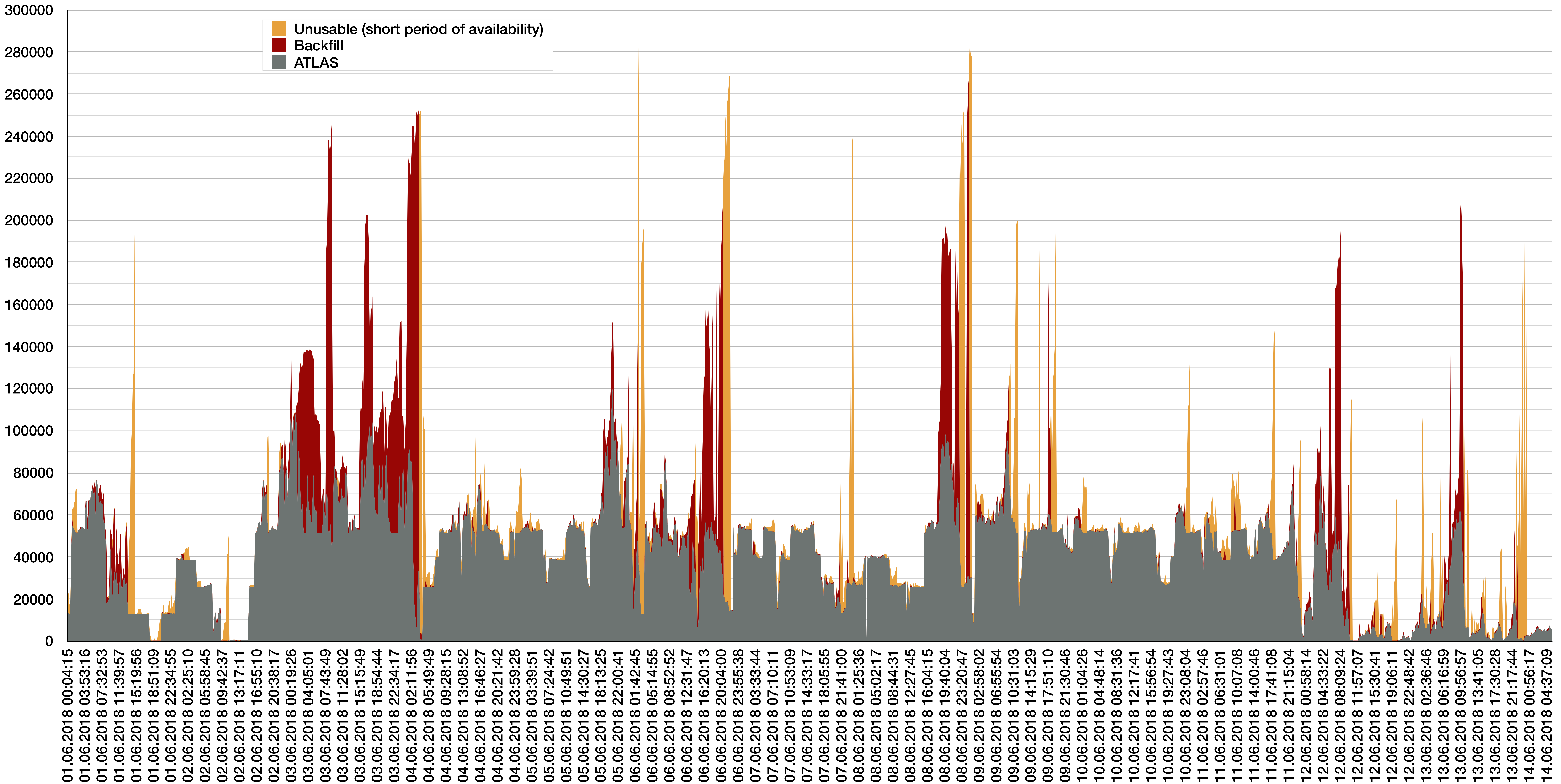
EVALUATION OF PANDA: HARVESTER

- Motivation for Harvester
  - PanDA currently relies on server-pilot paradigm
    - PanDA server maintains state and manages workflows with various granularities, such as task, job, and event – Pilots are job-centric and independently run on worker nodes with limited view of local resource
  - Works well for the grid with 250k cores 24x7 as underlying resources are not very heterogeneous
    - But missing capability to dynamically optimize resource allocation among differences of architectures (limitations by number of cores, amount of RAM per core, limitations of wall time etc.)
  - Not very well for HPC or large-scale clouds
    - Each HPC has a different edge service and operational policy, leading to over-stretched pilot architecture and incoherence in implementation at different HPCs
    - PanDA itself has no means of managing and monitoring cloud utilization by using native cloud API which is far more optimal than that of an intermediate service like condor

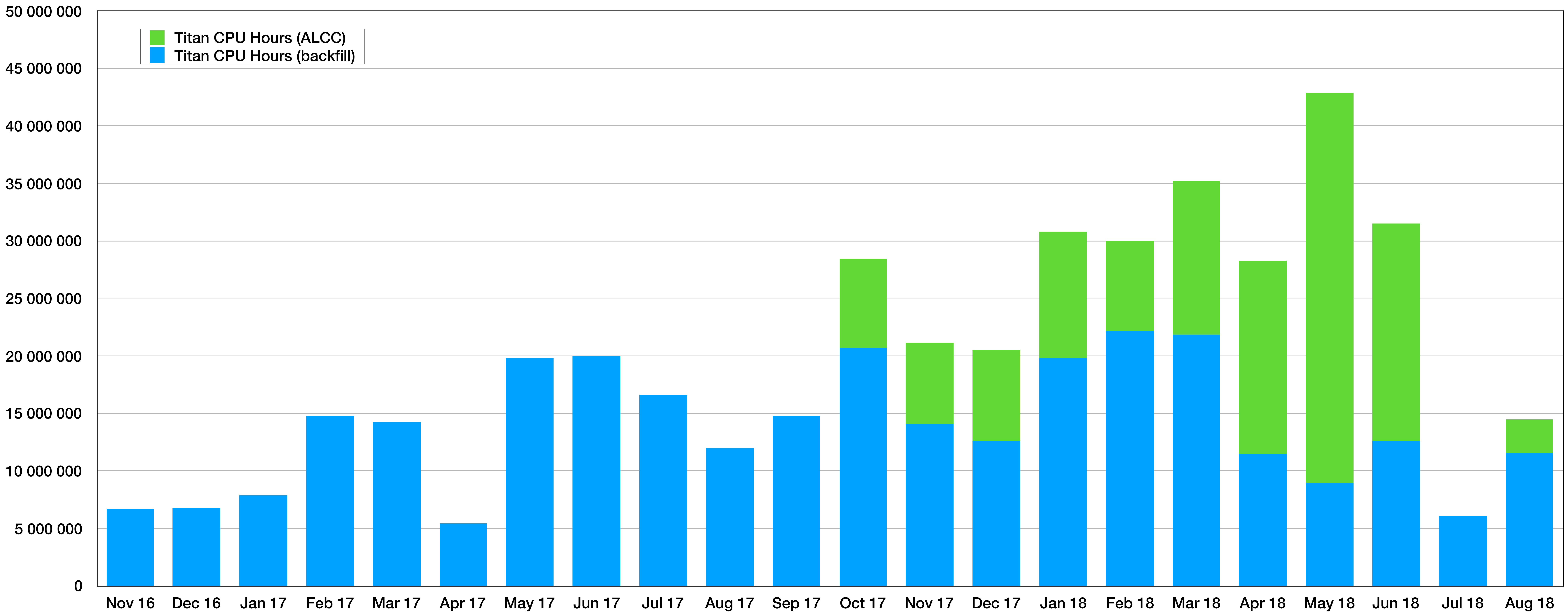
Harvester and Pilot 2.0 on Titan



Resources consumption on Titan with Harvester



ACHIEVEMENTS



>300M Titan core hours in a year

~8M PanDA jobs competed in a year

PLANS: №1 TOP500 The List.

- **Summit**, an IBM-built supercomputer now running at the Department of Energy's (DOE) Oak Ridge National Laboratory (ORNL), captured the number one spot with a performance of **122.3 petaflops** on High Performance Linpack (HPL), the benchmark used to rank the TOP500 list. Summit has 4,356 nodes, each one equipped with two 22-core **Power9** CPUs, and six **NVIDIA Tesla V100 GPUs**. The nodes are linked together with a Mellanox dual-rail EDR InfiniBand network.

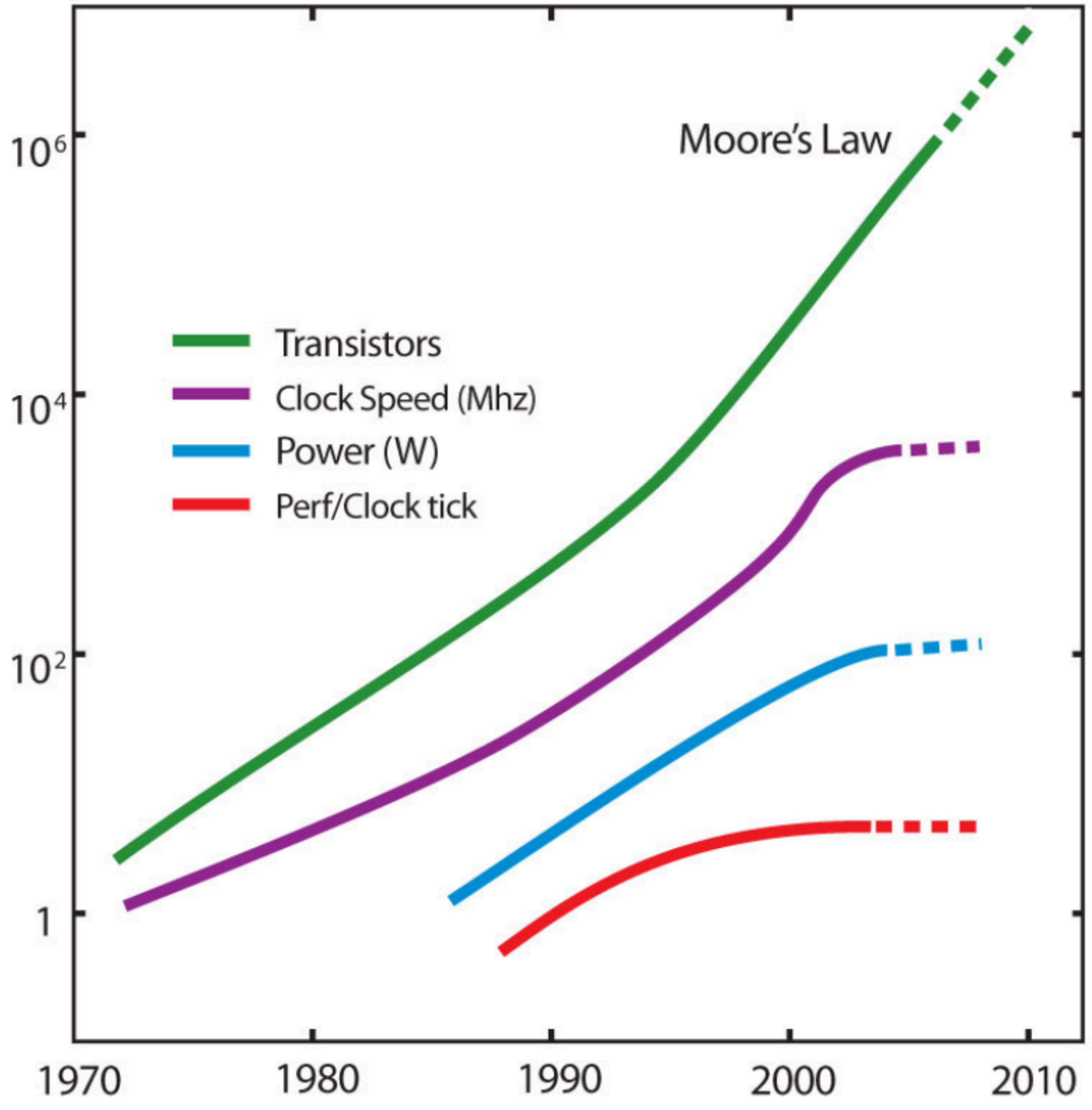
Processor (2 per node):	**IBM POWER9™**
GPUs (6 per node):	NVIDIA Volta
Nodes:	4,608
Node Performance:	42TF
Memory/node:	512GB DDR4 + 96GB HBM2
NV Memory/node:	1600GB
Total System Memory:	>10PB DDR4 + HBM + Non-volatile
Interconnect Topology:	Mellanox EDR 100G InfiniBand, Non-blocking Fat Tree
Peak Power Consumption:	13MW

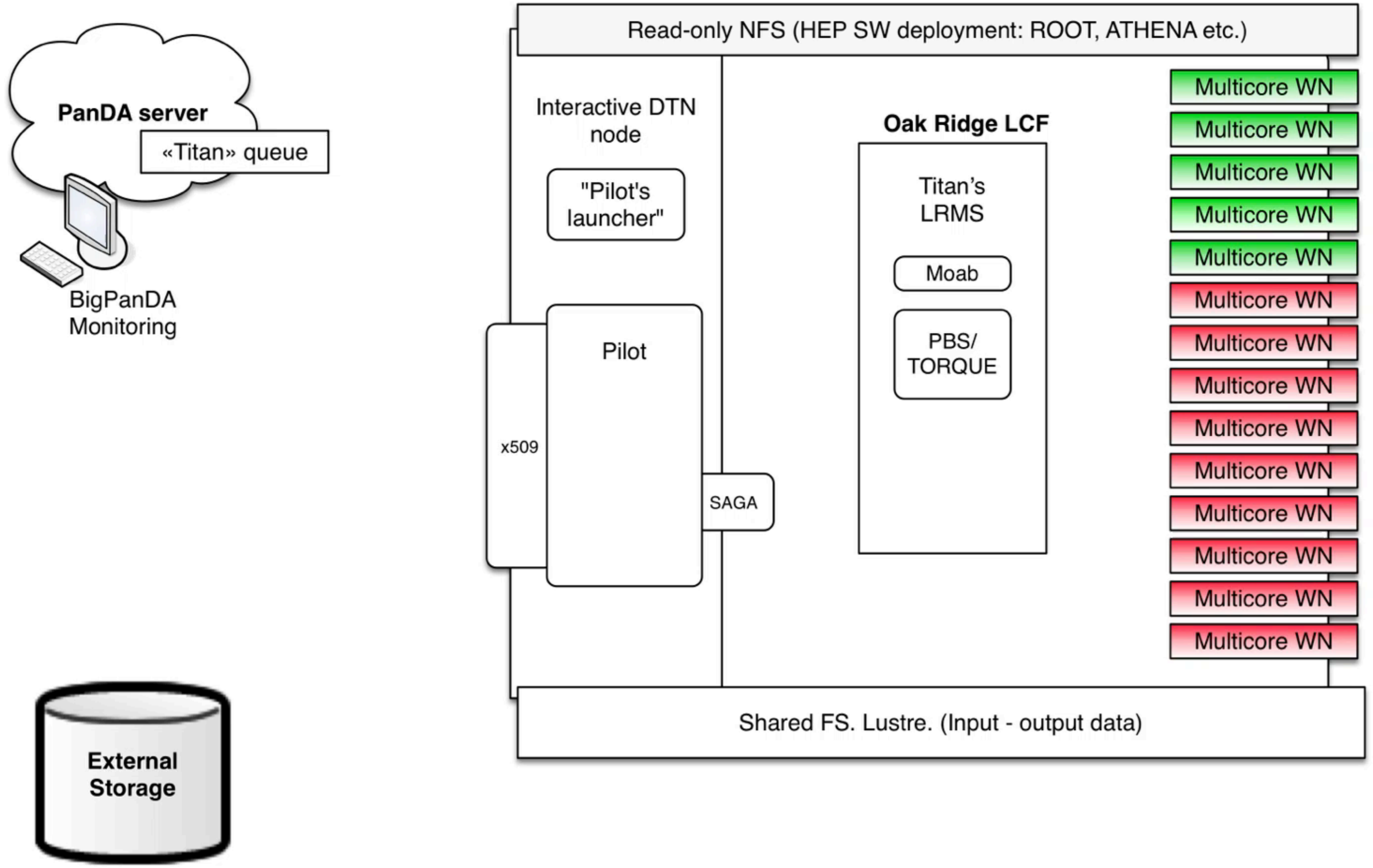
BACKUP SLIDES

FREE LAUNCH OVER TEN YEARS AGO

- Moore's law is still in charge, but
  - Clock rates no longer increase
  - Performance gains only through increased parallelism
- Optimizations of applications more difficult
  - Increasing application complexity
    - Multi-physics
    - Multi-scale
  - Increasing machine complexity
    - Hierarchical networks / memory
    - More CPUs / multi-core

☞ Every doubling of scale reveals a new bottleneck!




PanDA server
«Titan» queue
BigPanDA
Monitoring
External
Storage
Read-only NFS (HEP SW deployment: ROOT, ATHENA etc.)
Interactive DTN
node
"Pilot's
launcher"
Pilot
x509
SAGA
Oak Ridge LCF
Titan's
LRMS
Moab
PBS/
TORQUE
Multicore WN
Multicore WN
Multicore WN
Multicore WN
Multicore WN
Multicore WN
Multicore WN
Multicore WN
Multicore WN
Multicore WN
Multicore WN
Multicore WN
Multicore WN
Multicore WN
Shared FS. Lustre. (Input - output data)

SLOTS FROM HPC