Distributed Computing Evolution of High-Performance Computing

(with ATLAS & WLCG in focus)

Andrej Filipčič

Two paths of large-scale computing

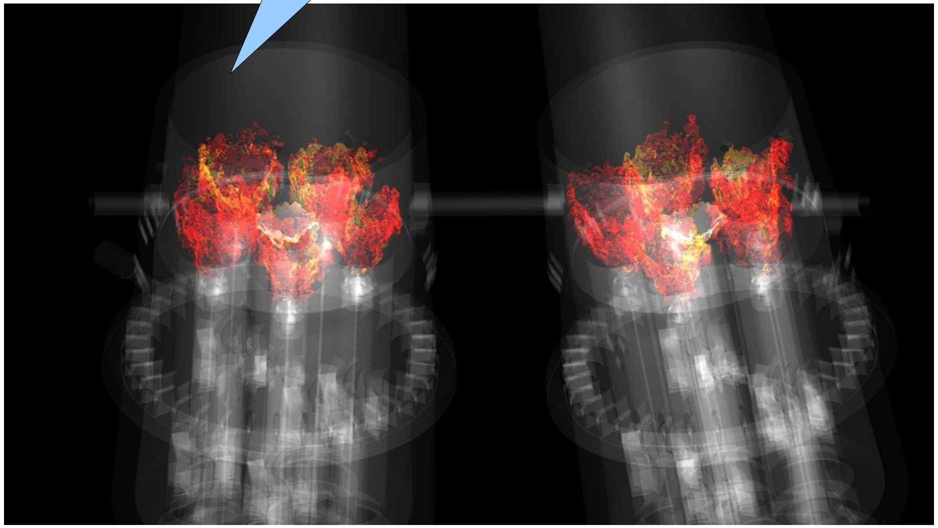
- Supercomputing (HPCs)
  - Massively parallel jobs, extremely CPU intensive, less data intensive
  - High-end hardware
  - Concentration of computing power at one place
  - Closed environment

- Grid computing
  - Serial jobs, trivially parallel jobs, medium CPU intensive, extremely data intensive
  - Commodity hardware
  - Resources distributed over many 100 computing sites
  - Open environment

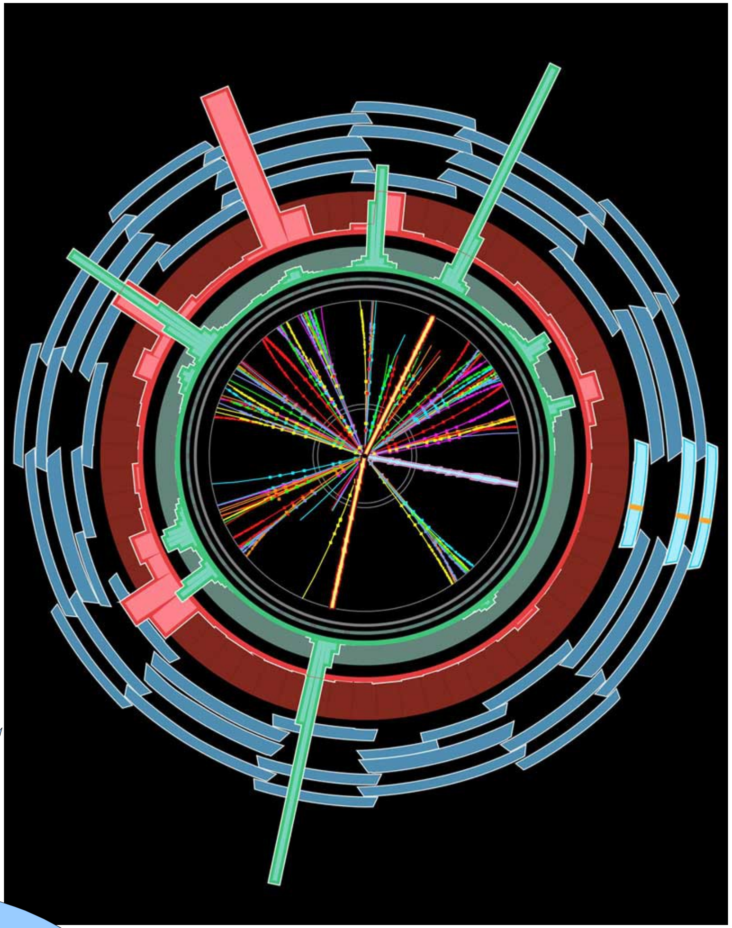
What about Clouds?
Roughly at the same level as Grid

HPC vs GRID

Titan - fuel combustion simulation



Parallel processing of single "long" event
1M hours



ATLAS – detector response simulation

Parallel processing of billions of independent "short" events
2M*0.1hour

Is the old paradigm still true today?

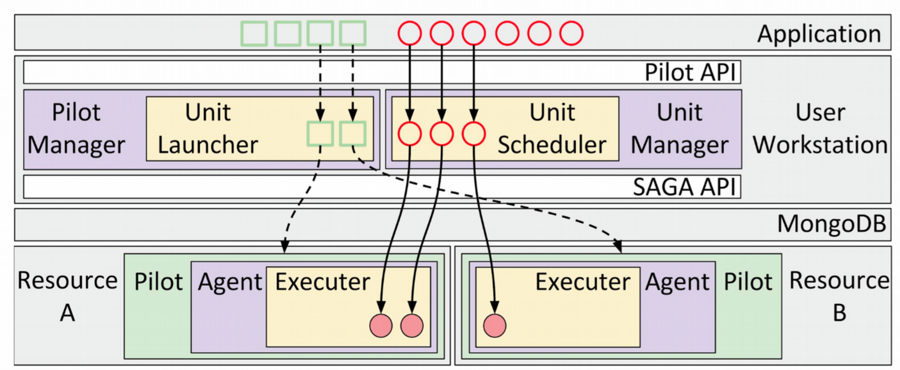
- Many HPC applications
  - Are I/O intensive
  - require processing of large amount of short independent tasks
  - Require access to external information (databases) or storage
- Many GRID applications
  - Are becoming multithreaded, parallel
  - Require huge amount of CPU – 2200Mhours/year for ATLAS Experiment
  - Large amount of memory (10GB/core)
- The difference between GRID and HPC (and Cloud) is shrinking!!!

HPC vision

HPC Requirements / Goals

- Workload with *heterogeneous* tasks
  - Varying core count
  - Varying application kernels
  - MPI / non-MPI
- Dynamic workload with workload unknown a priori
  - Dynamic: Tasks (workload) and task relations
- Control over concurrency of tasks
  - Might be loosely coupled (e.g. replica exchange)
- Multiple dimensions of scalability
  - O(100k) concurrent tasks

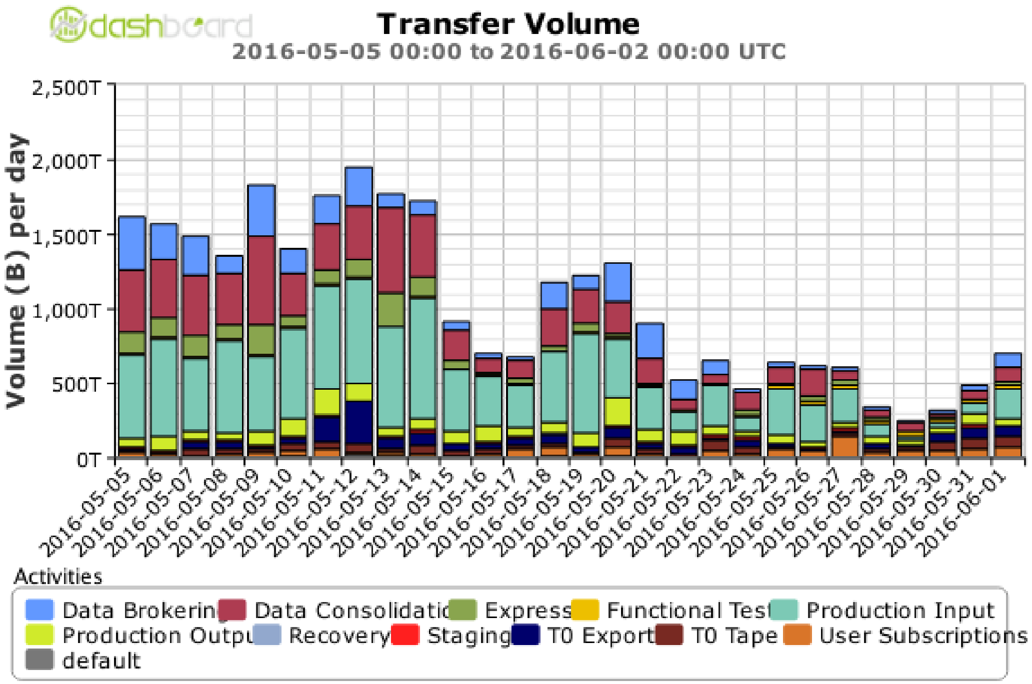
RADICAL-Pilot Architecture (2)



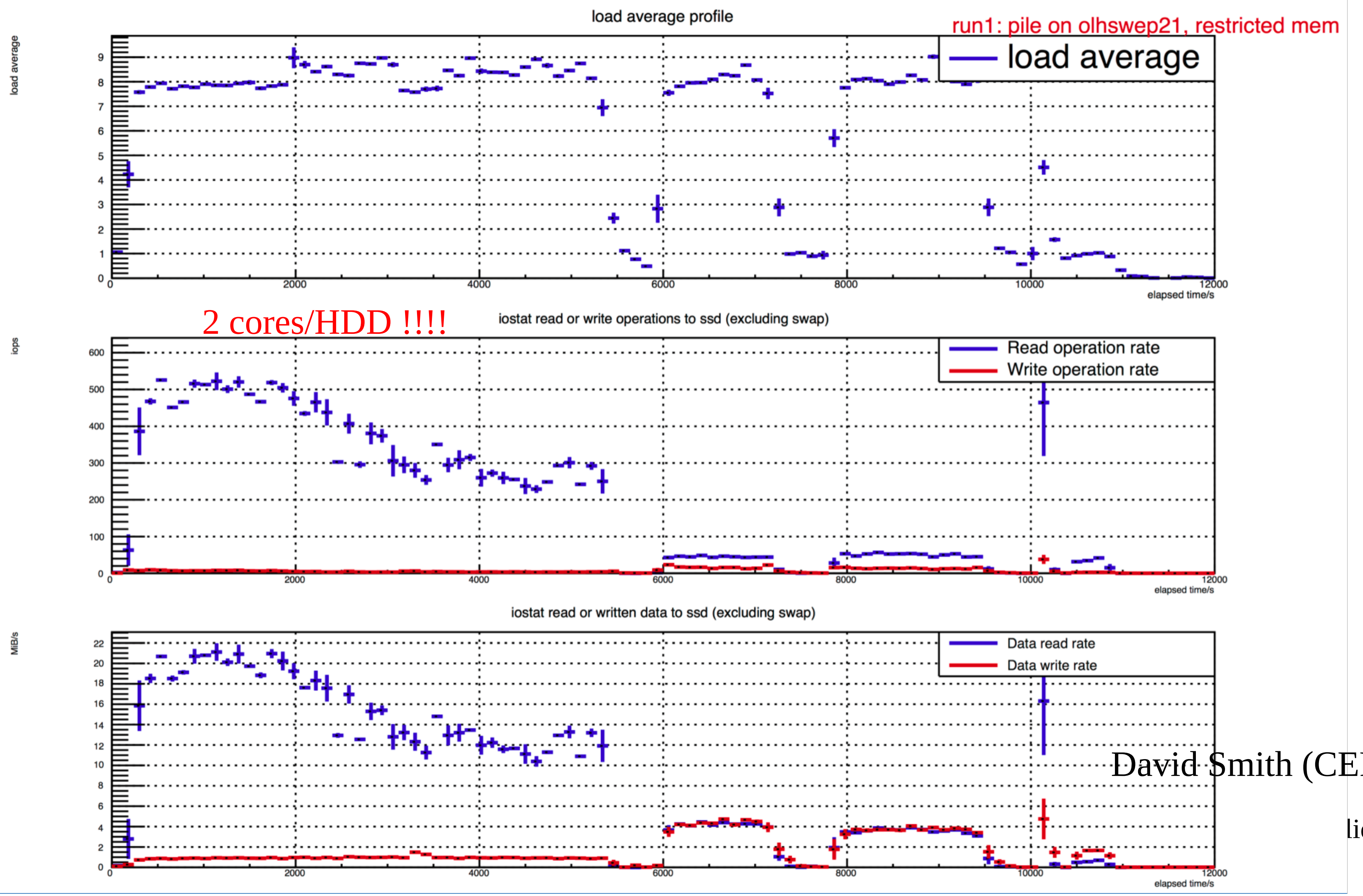
Shantenu Jha (Rutgers)

ATLAS Experiment

- Detailed description later today…
- Just few computing facts: ~100 distributed sites
  - 250k cores used all the time
  - 200PB of storage space
  - 1M jobs/day
  - 2PB of data is transferred per day between computing sites
  - Sites include: WLCG GRID sites, HPCs, Clouds, Volunteer computing



ATLAS - MC Reconstruction job profile – Heavy jobs

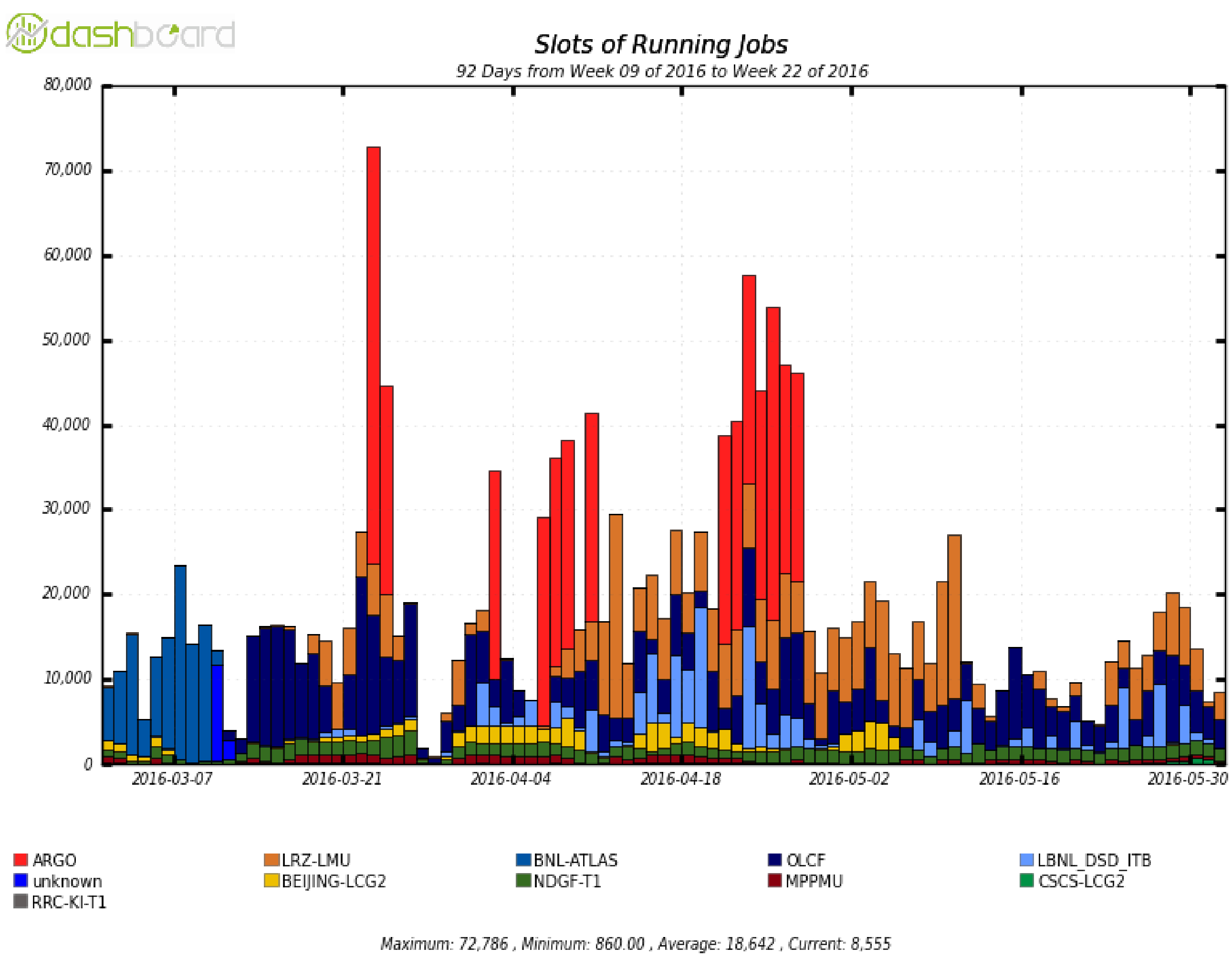


David Smith (CERN)

In not so distant past

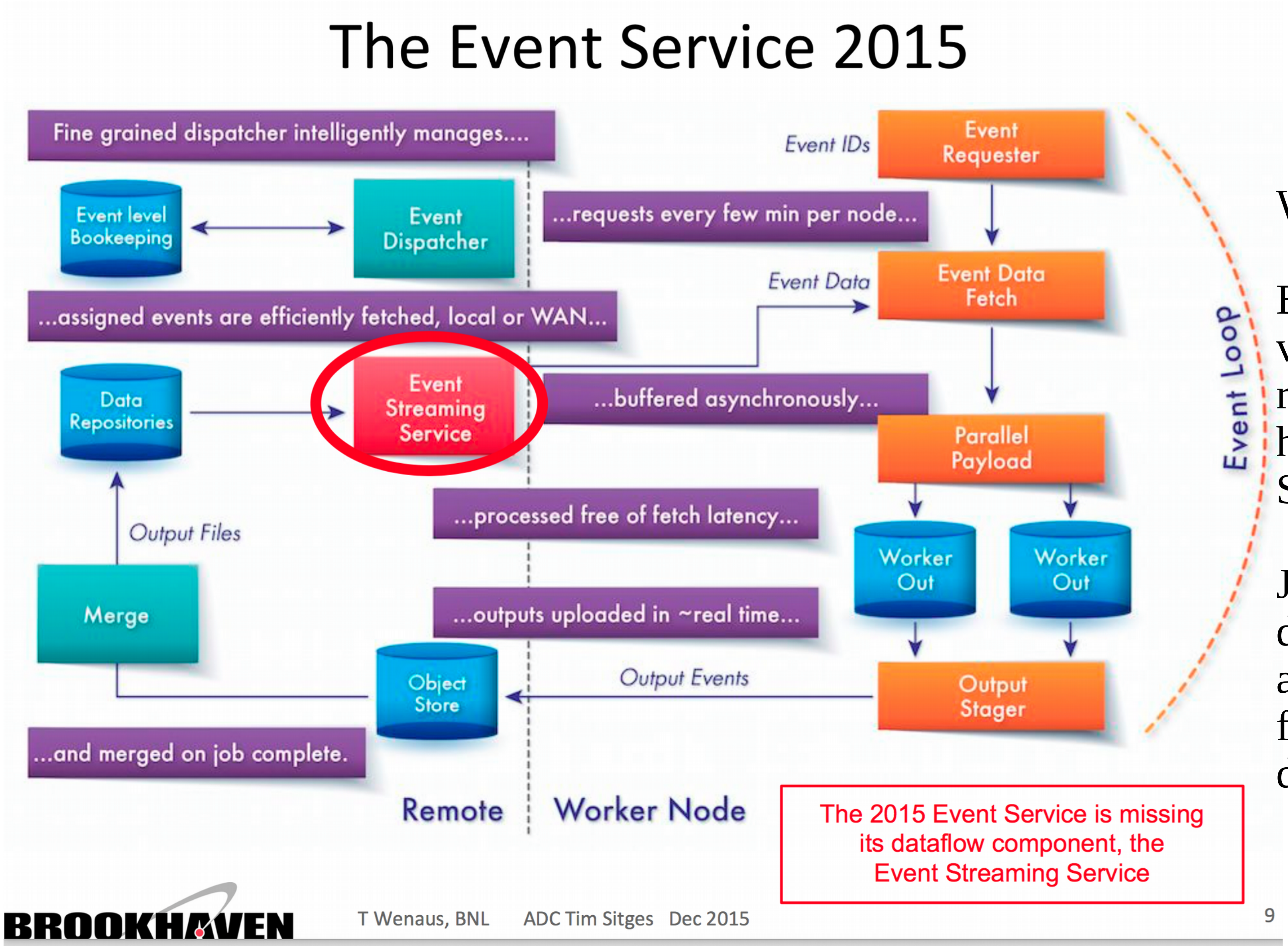
- Can we use your HPC for ATLAS?
  - Your software sucks
- Can we access external network on the nodes?
  - Are you crazy?
- Can we install grid services on the HPC?
  - No way
- Can we submit the jobs through ssh?
  - Try and we are going to sue you
- Can we install 2TB of software on the file system?
  - Is your data embedded in your software?
- Can we use 30k cores in the next year?
  - Huh?

Today: ATLAS – Running HPC cores



- ARGO – Mira
- LRZ_LMU – SuperMUC
- BNL-ATLAS,OLCF – Titan
- LBNL_DSD_ITB – NERSC
- BEIJING-LCG2 – ERA, Tianhe-1A
- NDGF-T1 – Triolith,Abel,DCSC
- MPPMU – Hydra
- CSCS-LCG2 – Piz Daint/Dora

Moving to event level granularity – 10min payload/core

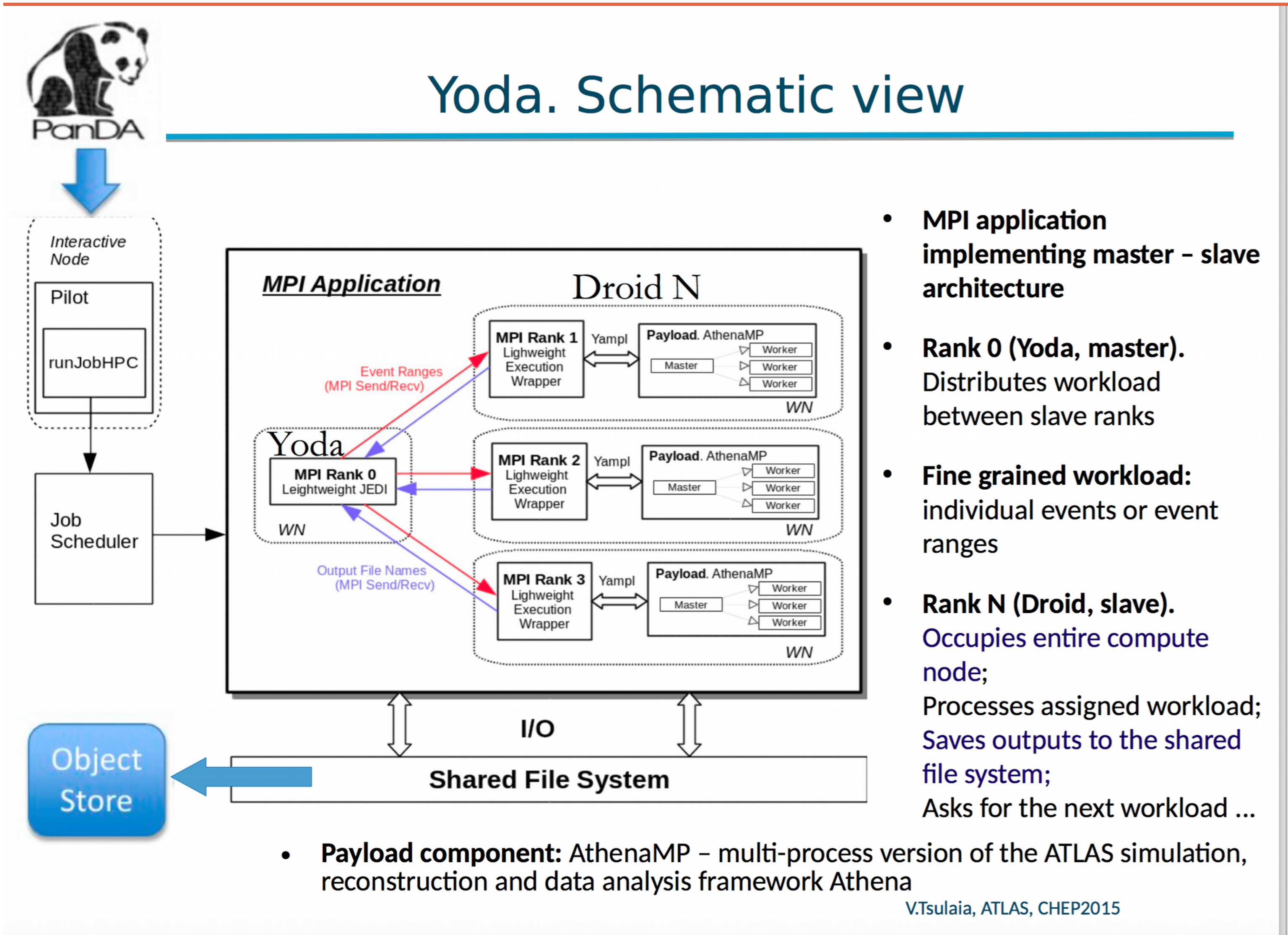


Why doing this?

Better suited for various types of resources, from home PCs to SuperComputers

Jobs are fully dynamic and they are not sensitive to a fixed execution deadline.

Executing short payloads on HPC



The master process on Rank 0 controls execution of short payloads on allocated nodes.

A single big parallel job scales well up to 265k cores, the main bottleneck is the filesystem throughput

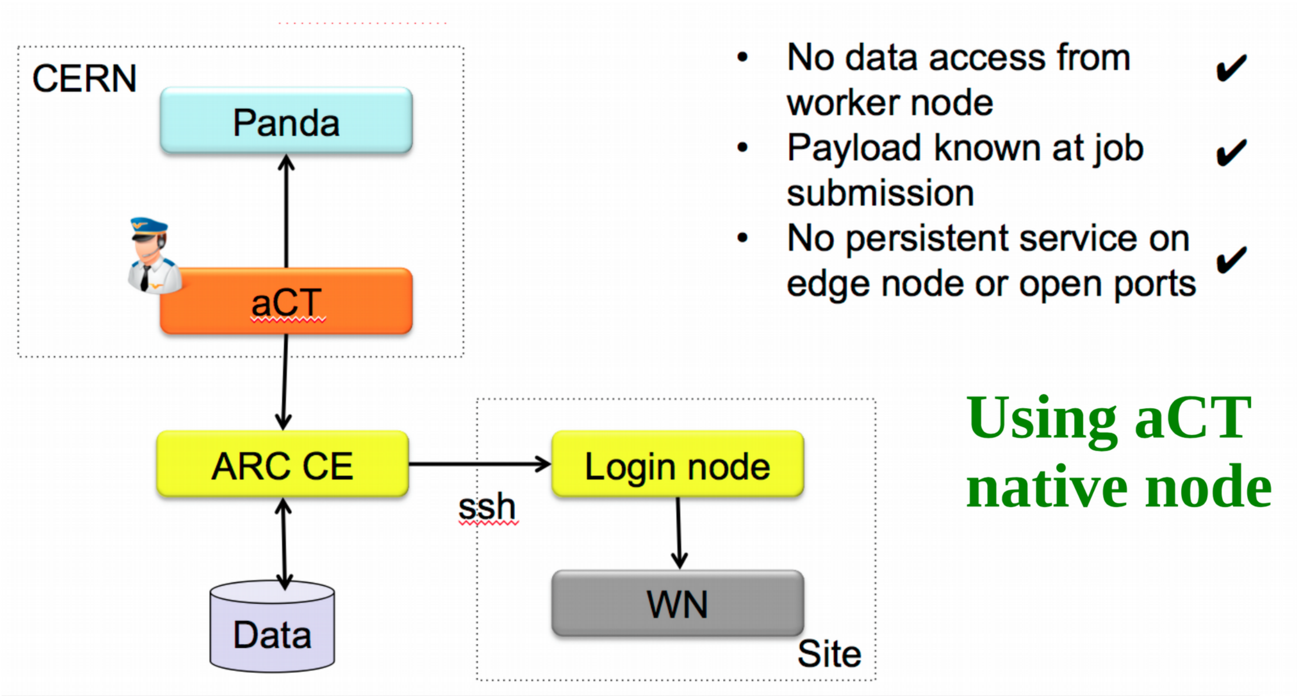
Where are the challenges with HPCs? EVERYWHERE !!!

- Software Delivery and Setup
  - A single ATLAS release is 20GB
  - Installed on cvmfs networked http-based RO file system
  - Many different releases used – deployment and setup is non-trivial
- Data Delivery:
  - ATLAS jobs are data oriented: input needs to be fetched from permanent storage and output stored there as well
    - For real data processing: 0.2MB/s/core IN, 0.1MB/s/core OUT

Where are the challenges with HPCs (2)?

- Outbound connectivity
  - Jobs require access to external ATLAS databases – too big to copy
  - Jobs need to communicate with central ATLAS scheduler
  - Jobs sometimes need access to files stored on external permanent storage
- Authorization
  - WLCG relies on x509 standard, HPCs are usually provide ssh only access, sometimes with short-lived keys – automation is difficult
- External access
  - No general external access to HPCs – using custom edge services is extremely limited
  - In some cases, an automated access through key-based ssh is allowed
  - In some cases, no push of any kind to HPCs is allowed – everything must be managed internally (eg request for data, communication with distributed services)

How to treat the limited HPC accessibility?



- One possibility:
- Panda
  - central scheduler
- aCT
  - central dispatcher
- ARC-CE
  - HPC batch gateway
- Used on some HPC sites
- Some restrictive sites require dispatcher and gateway inside the HPC to satisfy no push policy

Towards a common solution for distributed HPCs

- Many aspects of ATLAS computing are shared amongst other scientific fields as well. Some examples of development in this direction:
- User oriented application execution – virtualization, containers
  - Shifter@NERSC
  - Virtualization plans of IBM, Cray
- Data delivery services
  - US LCFs are building gridftp-based service network for data transfers
  - Many EU HPCs integrate ARC-CE to allow external job submission and managed data transfer
- Common gateway to HPC sites
  - SCEAPI in China is a restful interface for job submission to a network of 15 HPCs
- Opportunistic usage and backfilling
  - Many HPCs have empty resources (~10%) due to large job scheduling
  - Using them with short dynamically-sized jobs is encouraged
- Outbound access
  - Some HPCs already opened either directly to the outside world, or through custom proxies

Generalizing the requirements of future large-scale applications on HPCs

- Dynamic execution:
  - Resource allocation and payload delivery need to be separate – execution ordering driven by application
  - Applications will be self-adaptable to available resources which can dynamically change during execution
- Global information access
  - Task execution will require access to external distributed information sources (eg databases)
- Application driven job/task scheduling
  - o(100k) parallel task execution cannot be driven by site services
  - Applications will use their own custom task schedulers
- Continuous data delivery
  - Applications will demand a constant flow of input data to process
  - Output data will need to be uploaded in a managed way to a desired location
- Data management
  - Organization of input and output data needs to be application oriented

- The large computing centers of the future will need to address all this requirements and need to focus on common building blocks for application oriented services.

Conclusions

- The computing approach is rapidly changing these days and the old paradigm of HPC/GRID/Cloud distinction makes no sense any more
- Future computing will be more and more user oriented – applications are becoming more complex and cannot be contained any more in a single site or environment
- Computing resources need to evolve to provide seamless integration into scientific community frameworks.