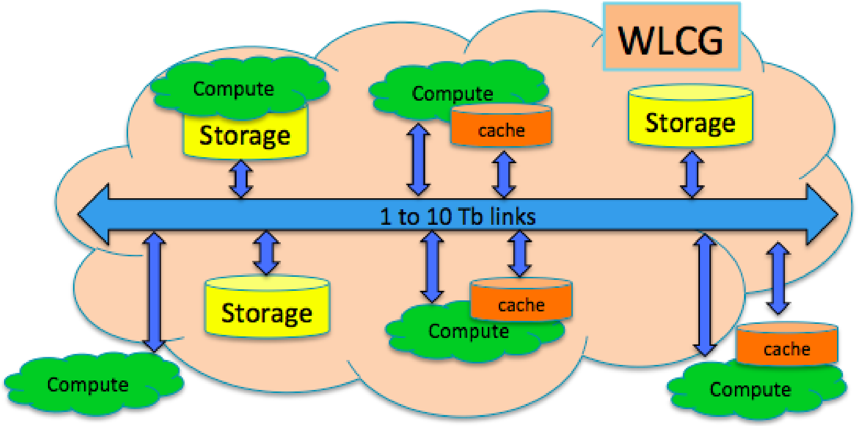
Review of the WLCG experiments compute plans

Simone Campana (CERN)

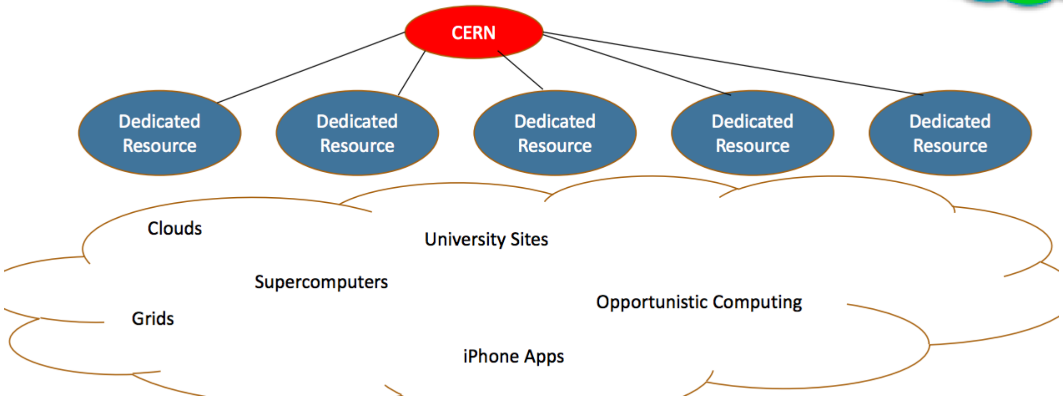
Evolution of the WLCG infrastructure

Evolution in the direction of

- Network centric model
- Consolidation of storage
- Diversification of facilities
- …



WLCG at HL-LHC



WLCG at HL-LHC (I. Fisk's representation)

- … diversification of compute resources

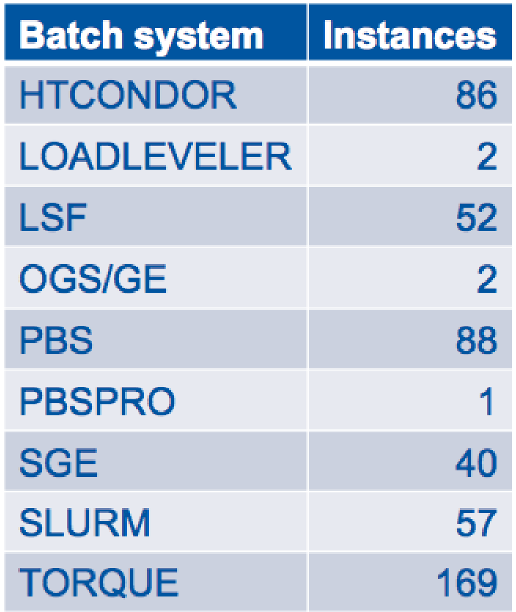
No need to wait 2026 for this

Compute Resources and experiments Computing Model

- Diversification of Compute Resources
  - Provisioning interface: Grid CE, cloud web service, login to batch head-node, none …
  - Resource availability: pledged vs opportunistic, flat vs elastic, …
  - Resource retention: from long living resources to highly volatile
- Diversification of Computing Models
  - LHC experiments are different. More obvious for Alice and LHCb (special physics focus)
  - Also true for ATLAS and CMS (different detector layouts and different sociologies)
  - This has an impact on the computing models, and motivates different choices and different focuses

Job Requirements

- LHC experiments run a mix of single and multi core jobs
- Generally the jobs require largely less than 2GB of memory but there are special cases (e.g. Heavy Ions and Upgrade samples) requiring more
- Flexibility to provision what you need at low/no extra cost will be increasingly important
- A challenge for site administrators and for the batch systems
- Do we have the right (modern) tools? http://cern.ch/go/jx9S



Batch system	Instances
HTCONDOR	86
LOADLEVELER	2
LSF	52
OGS/GE	2
PBS	88
PBSPRO	1
SGE	40
SLURM	57
TORQUE	169

Containers

- A lightweight technology to provide isolation, with many benefits for experiments and sites. E.g. :
  - Payload isolation from pilot credentials
  - SL6 environment on other distributions

- A lot of (coherent) interest from the experiments

- Focus on Singularity for WN environment

- This workshop is a great opportunity to discuss and push this further

Clouds

- LHC experiments leverage cloud resources for data processing. Two options:
  1. The experiment workflow management system instantiates VMs (generally through Condor) and the VMs join an experiment Condor pool to which pilots are submitted
  2. A grid site instantiates VMs and the VMs join the site batch system

- Effectively the same thing, but in (2) the facility administers the WNs while in (1) the experiment operates the WNs

- I prefer (2)

- Condor generalizes the provisioning and access to cloud processing units and proved to be reliable. A standard de facto.

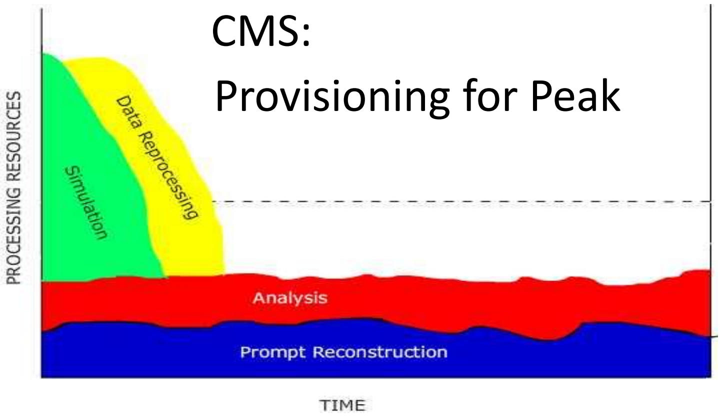
Elastic Resources

- Different interests due to different needs

- ATLAS needs large CPU resources for relatively low priority G4 simulation (Calorimeters)

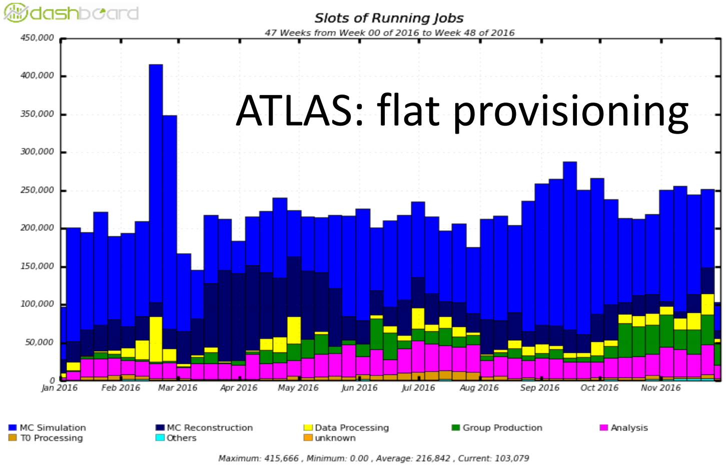
  => High priority tasks pushed through processing shares with flat provisioning

- CMS G4 simulation is faster, so a big interest in provisioning for peaks for bursty activities



CMS:

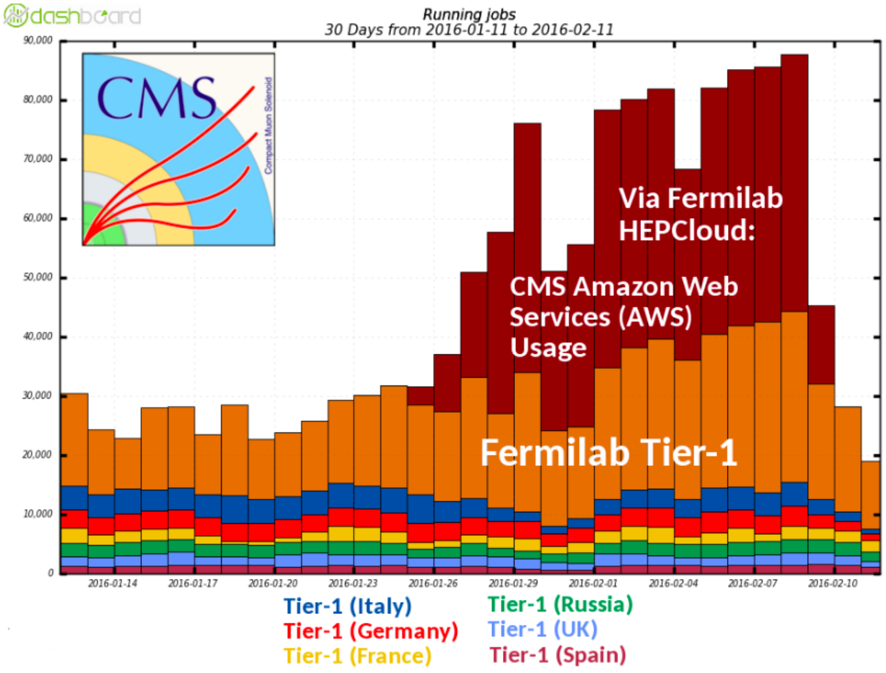
Provisioning for Peak



ATLAS: flat provisioning

Elastic Resources

- All experiments started integrating cloud resources for simulation
- But the goal and real advantage consist in leveraging cloud flexibility for all use cases. Including the challenging ones
  - High memory, many cores, long jobs, ..
- CMS run the full production chain on commercial clouds, carefully considering also the economic model



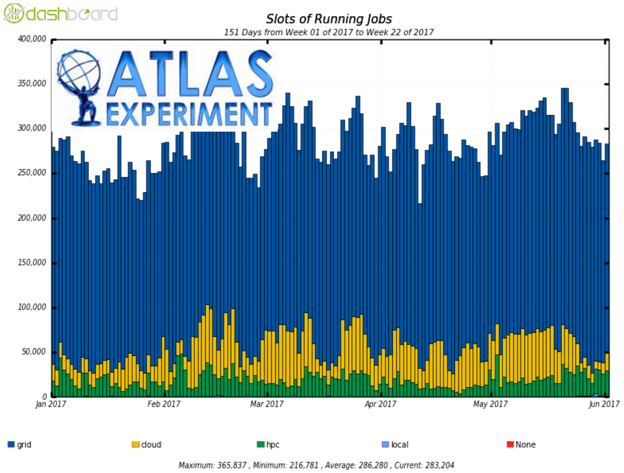
Via Fermilab HEPCloud:

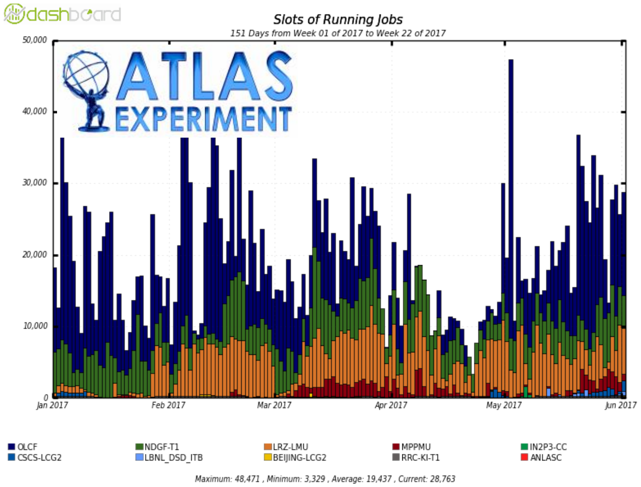
CMS Amazon Web Services (AWS) Usage

Fermilab Tier-1

HPCs

- Rather complicated resources to exploit …
- The main challenge comes from diversity (e.g. site policies and CPU architectures)
- HPCs are built for a use case not very suitable for our embarrassingly parallel use case
- … but potentially a large number. So experiments invest on integrating them
- Invest in common tools. Software distribution, data handling, edge services. Try to negotiate consensually policies



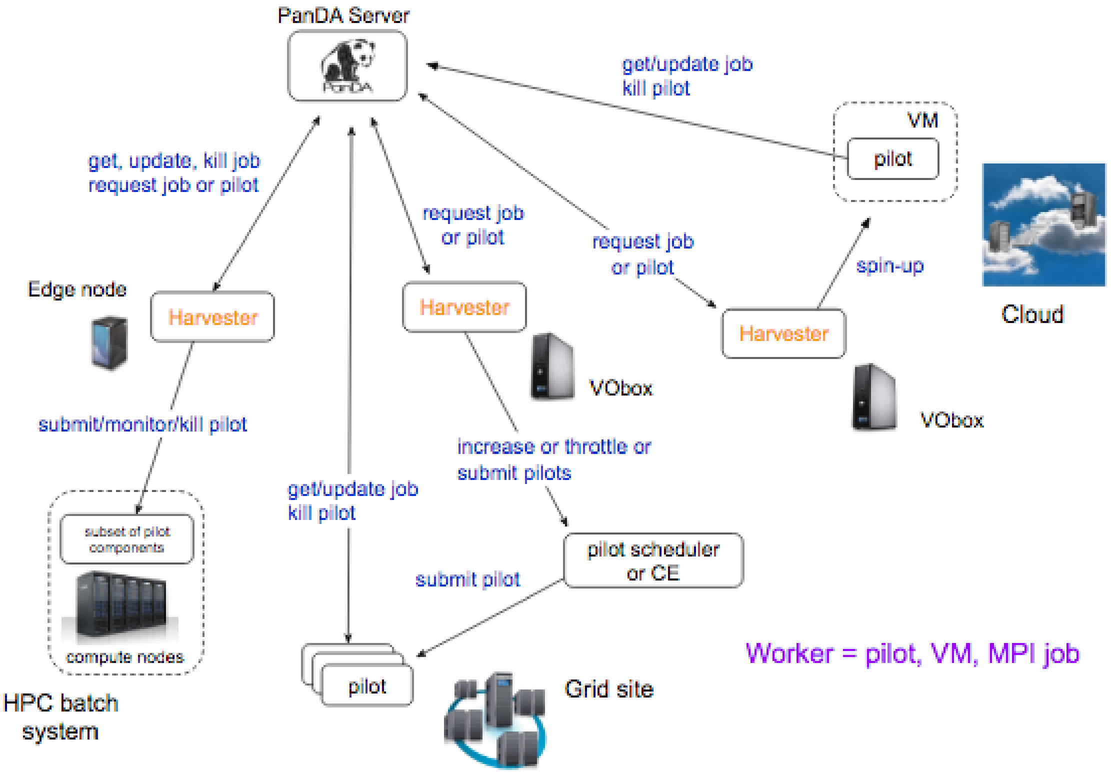


Rationalize provisioning layer

- In ATLAS, the strong need to rationalize resource provisioning at HPCs initiated Harvester project

- An edge service, creating a communication channel between resources and WMS (PanDA)

- A plugin-based architecture allows to leverage many of the functionalities for Grids and Clouds as well



Worker = pilot, VM, MPI job

Volatile resources

- Considerable processing capacity can be exploited for short periods of time (or with reduced QoS)
  - HLT farms between fills, Spot Market on Clouds, Backfill in HPCs, …
- In many cases, for such resources one can expect pre-emption at any time
  - Could be softer (SIGTERM followed by SIGKILL) or harder (SIGKILL with no merci nor regret)
- Several solutions on the table, with different advantages and different challenges
  - Machine Job Feature to size at runtime the workload depending on what the resource can offer
  - Reduce the data processing granularity in a check-pointable way to one(few) event(s)
  - Ad-hoc short jobs + extra care in retry, monitoring and allarming

Lightweight sites: Vac and @HOME

- Vac - clusters of autonomous hypervisors manage VMs
  - In production at many UK sites, VMs in production for ALICE, ATLAS, LHCb, and VOs of the GridPP DIRAC service
  - Major component for LHCb integrating Clouds, HLT,
  - Generic VMCondor can connect VMs to an HTCondor pool
  - Report usage to APEL as “virtual” CEs

- LHC@HOME - interesting outreach project
- Provisions non negligible processing capacity (1-2% in ATLAS) and a technology for lightweight sites
  - Based on Boinc and various solutions for data handling