Data Access at HPC
ATLAS/CMS Perspective

S. Jezequel (LAPP), Frank Würthwein (UCSD)

Credits

- D. Hufnagel : JLab presentation ‘**Review on current workflows and production on HPC centers**’

- ATLAS slides : D. Benjamin (Presentation in ATLAS site Jamboree), A. Klimentov, D. Oleynik, A. Filipcic

Software/frontier access

- CMS
  - Use local Squids for Frontier at HPC centers.
  - Use local CVMFS Stratum-R or containers for CMS software
- ATLAS
  - Some sites provide cvmfs/squid
  - For the ones without squids : Run only simulation

ATLAS : Harvester Data Access

- Large HPC’s (DOE and NSF) :multi Data Transfer Nodes
  - Multi-factor authorization (DOE) → increase complexity for automatic TPC
  - Globus is available at all HPC centers https://www.globus.org/

- Data In/Out
  - – PanDA instructions Rucio to transfer files to/from RSE (Datadisk) associated with HPC PanDA queue
    - Titan (BNL-OSG2) , NERSC – ALCF (SLACXRD)
  - – Harvester running on an edge node uses plugins used to transfer data to HPC local shared storage
  - – Harvester plugins clean up old space (sweeper)
  - – At Titan/Summit – Harvester run edge node – use Rucio client code
  - – ALCF and NERSC – Harvester run on login node and use Globus Python SDK to trigger transfer via Globus

ATLAS : ARC Data Access

- HPC Local storage : essentially a cache managed by ARC code
  - Cache is configured with watermarks – deleting last recently used files from cache
  - Jobs typically run from scratch space which is cleaned up when job finishes

- Data In
  - Harvester/aCT pulls jobs from PanDA and sends them ARC-CE with required input files
  - ARC CE queries Rucio for files and downloads them
    - Data is copied with gridftp, xrootd to ARC-CE with what ever is preferred protocol

- Data out
  - ARC CE uploads the job output files to preferred grid storage – in Event Service case – CERN Object Store, Grid RSE otherwise

CMS : Data Access

- Remote streaming for input and remote stageout
  - Also do static placement of certain samples (pileup)
  - Adopting Rucio → benefit from common ATLAS/HPC/Rucio integration

- RAW input data or pre-mixing data as input preferably via remote read.
  - In practice, pre-staged ~600TB pre-mixing library with Rucio to NERSC at 100TB/day using multiple NERSC DTNs in parallel.

Going forward

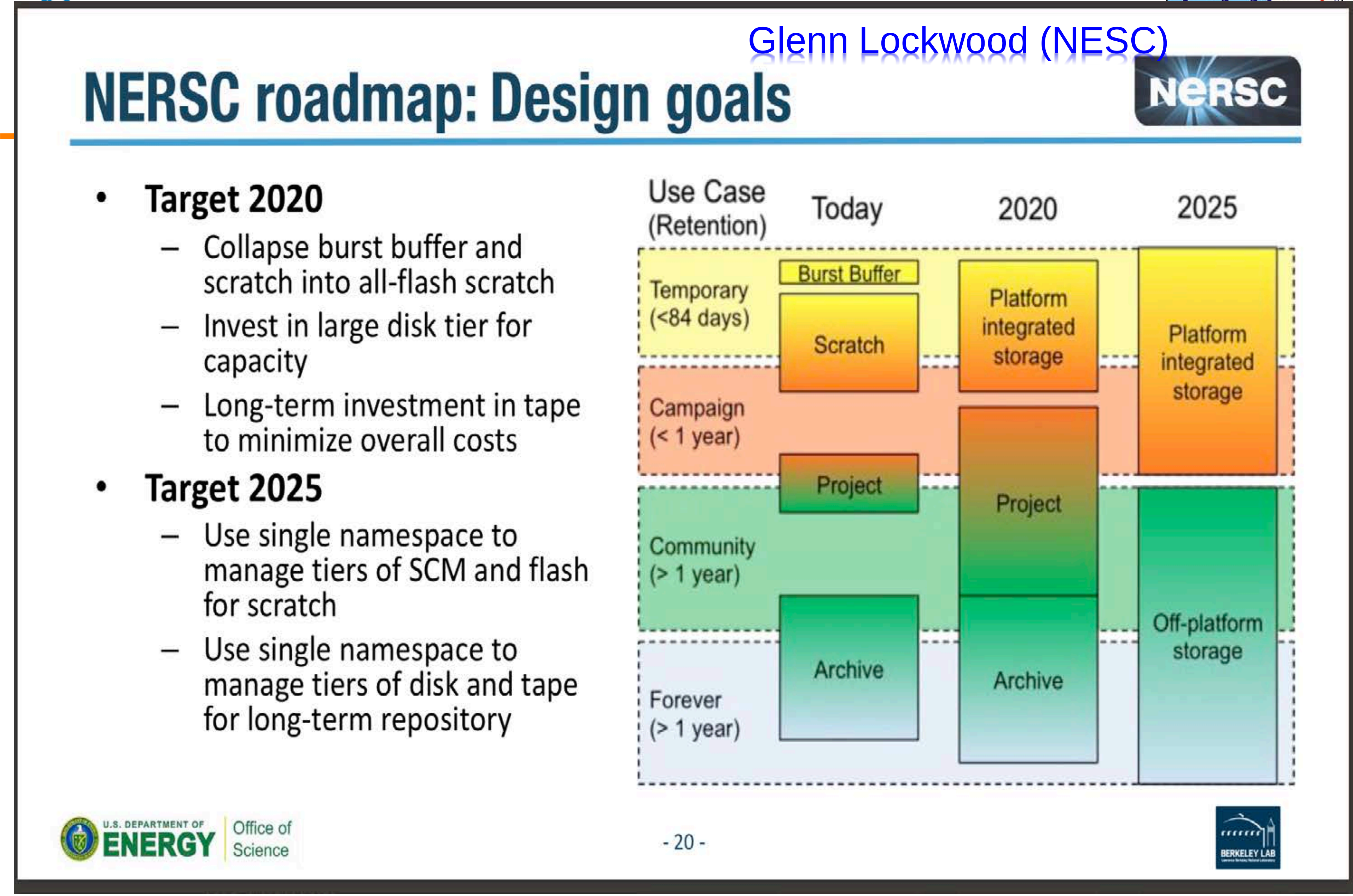
- Improve robustness of dual use Globus Endpoints/Rucio SE
  - working on second endpoint at BNL.
  - testing using OSRIS Object Store for Event Service output.

- Dream : Develop universal solution
  - Integrate the HPC Data Transfer Nodes into Rucio
  - Event Streaming Service for the data flow into the HPC disks
  - ➢ Using access tools from sites → avoids hickup from technology changes

Interest to document WLCG requirements

Backup

Glenn Lockwood (NESC)



NERSC roadmap: Design goals

- **Target 2020**
  - Collapse burst buffer and scratch into all-flash scratch
  - Invest in large disk tier for capacity
  - Long-term investment in tape to minimize overall costs
- **Target 2025**
  - Use single namespace to manage tiers of SCM and flash for scratch
  - Use single namespace to manage tiers of disk and tape for long-term repository

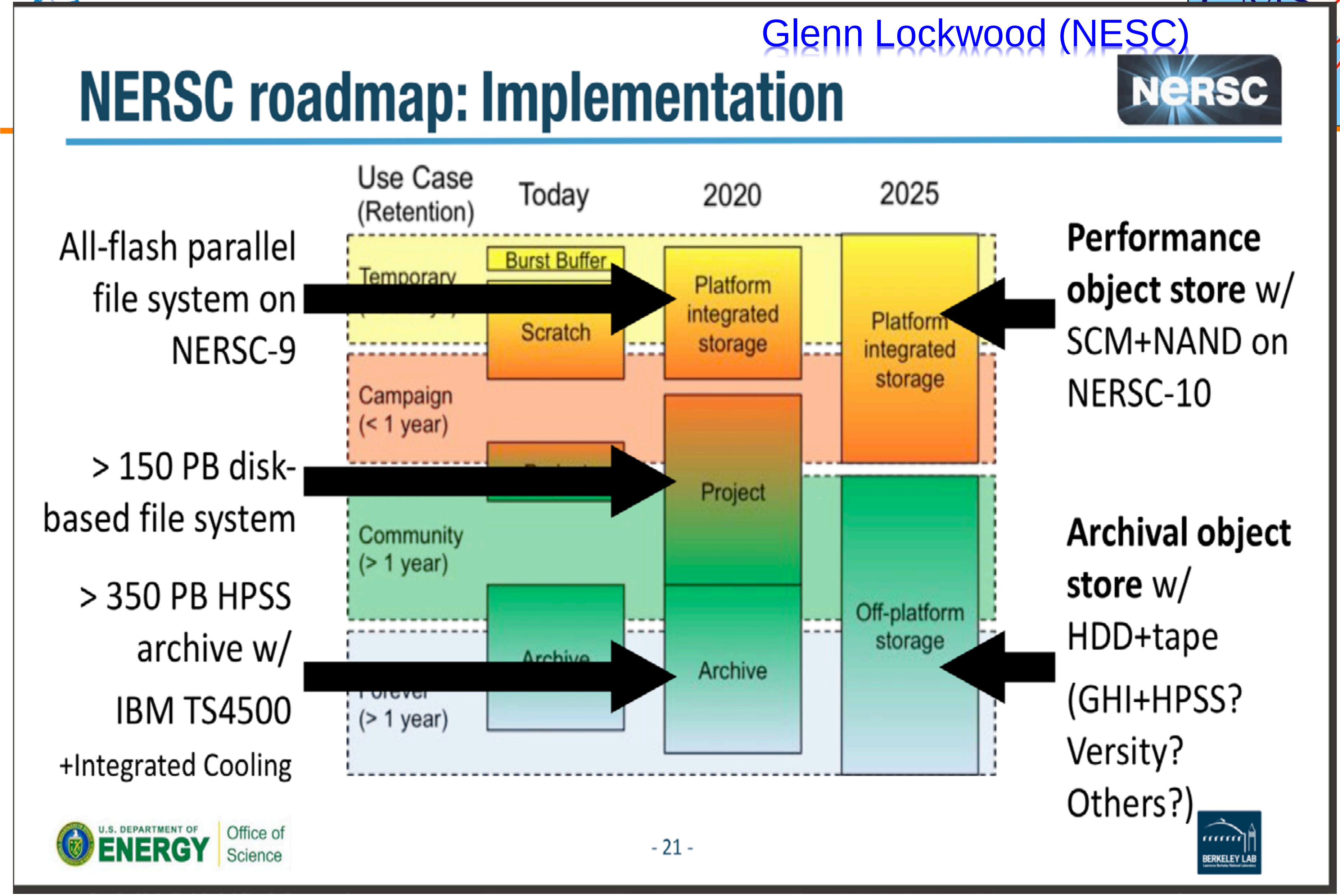
Use Case (Retention)	Today	2020	2025
Temporary (<84 days)	Burst Buffer Scratch	Platform integrated storage	Platform integrated storage
Campaign (< 1 year)	Project	Project	
Community (> 1 year)			Off-platform storage
Forever (> 1 year)	Archive	Archive	

U.S. DEPARTMENT OF ENERGY | Office of Science

- 20 -

BERKELEY LAB

Glenn Lockwood (NESC)



NERSC roadmap: Implementation

All-flash parallel file system on NERSC-9

> 150 PB disk-based file system

> 350 PB HPSS archive w/ IBM TS4500 +Integrated Cooling

Performance object store w/ SCM+NAND on NERSC-10

Archival object store w/ HDD+tape (GHI+HPSS? Versity? Others?)

U.S. DEPARTMENT OF ENERGY | Office of Science

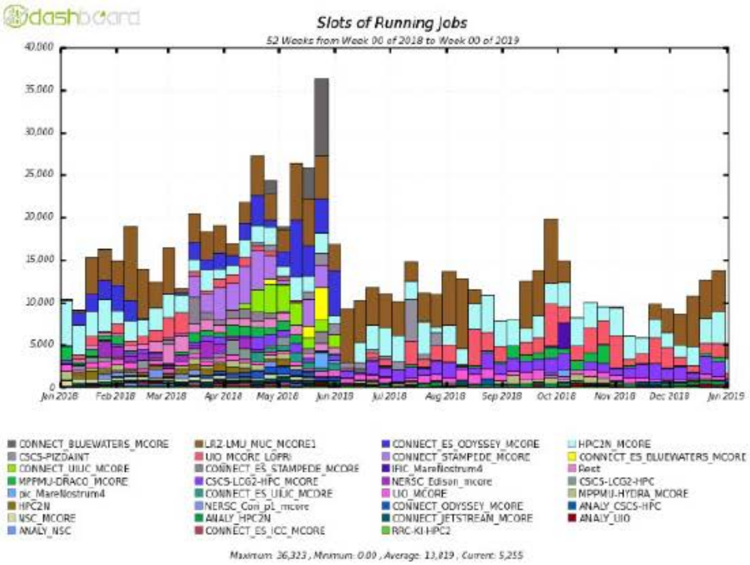
- 21 -

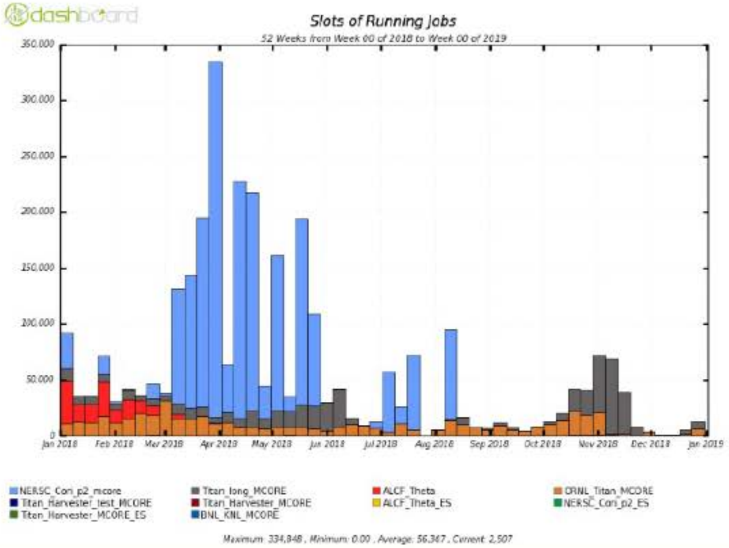
BERKELEY LAB

HPC Use : ATLAS

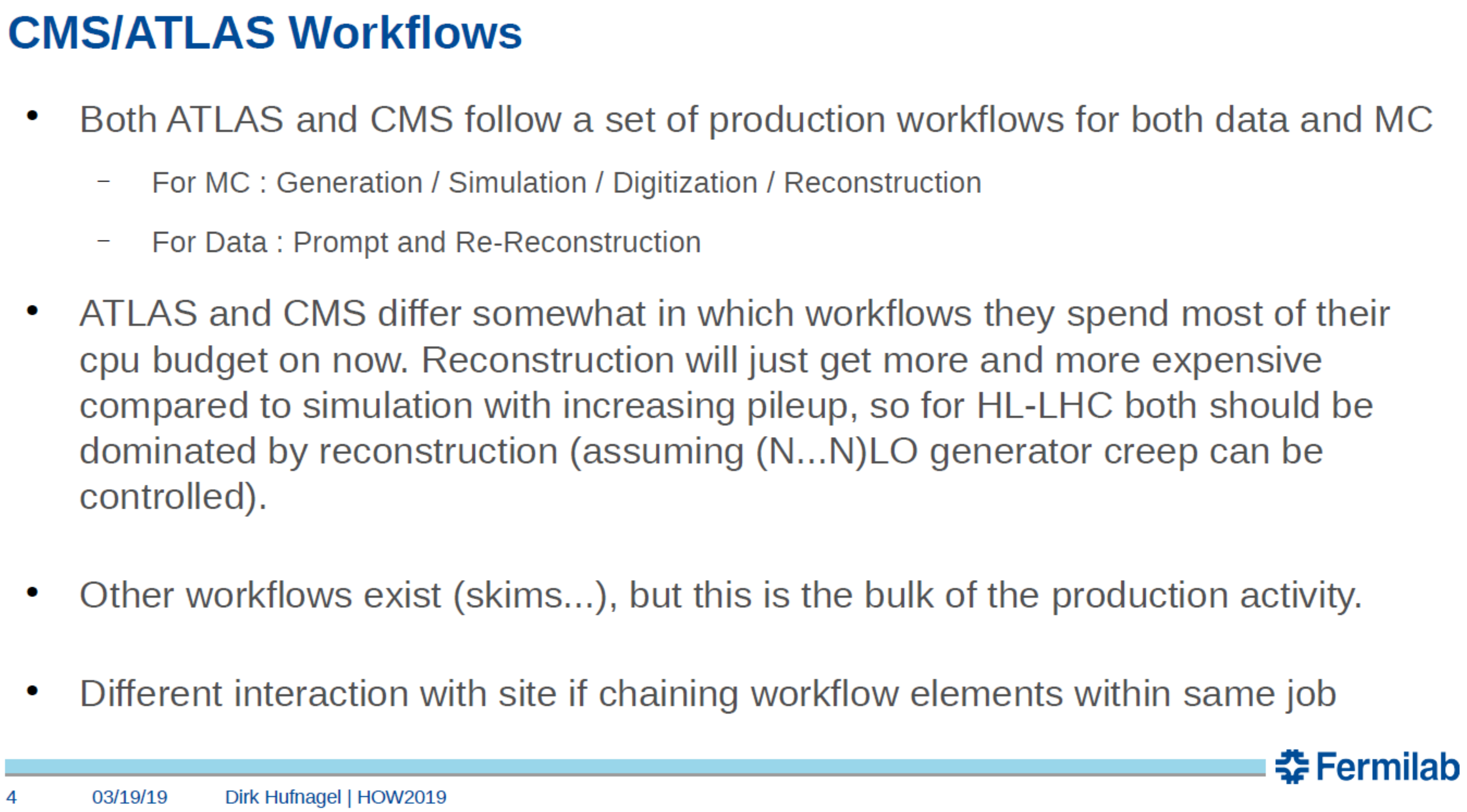
Scale of operations ATLAS

- In Europe using a variety of HPC, altogether 10k to few 10k cores
- In US using NERSC, ALCF, OLCF and others, total 2018 use multiple 100M hours





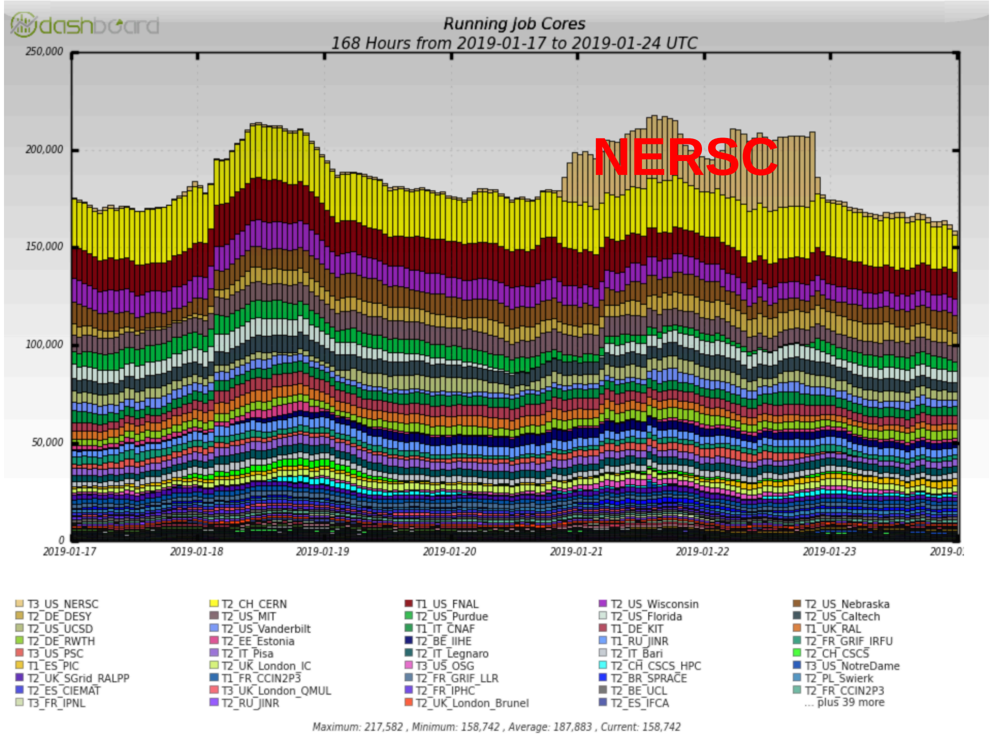
Relevant Workflows



CMS/ATLAS Workflows

- Both ATLAS and CMS follow a set of production workflows for both data and MC
  - For MC : Generation / Simulation / Digitization / Reconstruction
  - For Data : Prompt and Re-Reconstruction
- ATLAS and CMS differ somewhat in which workflows they spend most of their cpu budget on now. Reconstruction will just get more and more expensive compared to simulation with increasing pileup, so for HL-LHC both should be dominated by reconstruction (assuming (N...N)LO generator creep can be controlled).
- Other workflows exist (skims...), but this is the bulk of the production activity.
- Different interaction with site if chaining workflow elements within same job

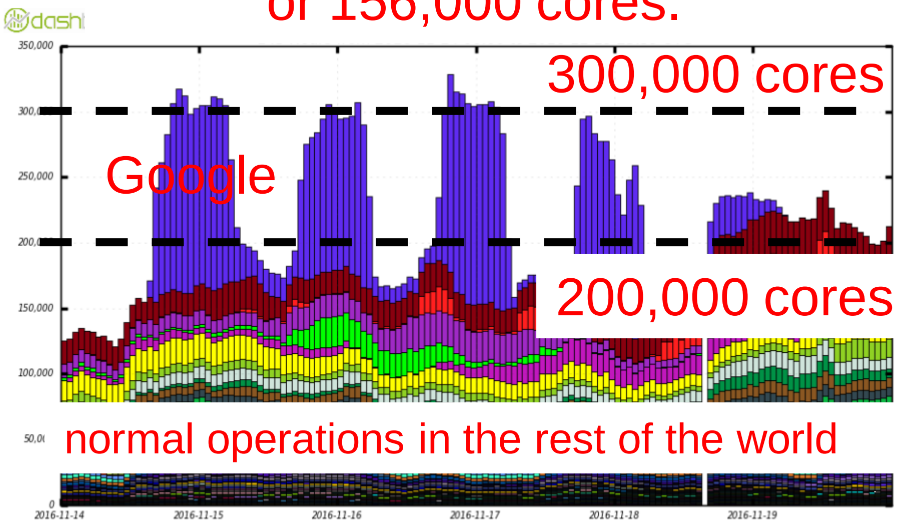
HPC and Cloud Use : CMS



NERSC

In addition to NERSC, CMS has allocations at Stampede, Comet, Bridges, Jetstream, and Piz Daint.

x2 elastic scale out on Google or 156,000 cores.



300,000 cores

Google

200,000 cores

normal operations in the rest of the world

So far, we run primarily full chain Gen-Sim-Digi-Reco at HPC/Cloud.

This implies O(1)PB pre-mixing input sample for digitization.

Relevant Workflows-backup

- Workflows domination HPC:
  - Data Reconstruction, i.e. (re-)processing of RAW data.
  - Gen-Sim-Digi-Reco, i.e. the entire chain from generator through reconstruction.
- Now, and even more so in the future, both are completely CPU dominated. IO is small enough that we routinely do remote IO.