GridKa 2018 - Aug 30, 2018

Six Key Challenge Areas Driving Innovation in Distributed High Throughput Computing

Todd Tannenbaum
Center for High Throughput Computing
Department of Computer Sciences
University of Wisconsin-Madison

CENTER FOR HIGH THROUGHPUT COMPUTING

HTCondor

University of Wisconsin Center for High Throughput Computing

HTCondor

- Open source software to enable distributed High Throughput Computing (HTC)
- Full featured, mature production system (1M+ LOC)
- Widely deployed
  - Used in production at hundreds of universities, government labs, commercial companies to manage compute clusters in science, engineering, finance, …
  - Components used to federate compute clusters into campus grids and wide-area computing grids, e.g. Open Science Grid, WLCG, …

Six Challenge Areas

- We have identified six challenge areas directing innovation in HTC technologies.
- We use these to guide our efforts on HTCondor.
- *Survey questions concluding this talk!*

CHALLENGE AREA #1

HARDWARE COMPLEXITY

It all starts with a server

- A server: The building block of a HTC environment
- Fundamental task of a DHTC environment: manage the workload (jobs), and manage the resources (servers):
  - Schedule server resources to jobs
  - Manage server resources (utilization, isolation, monitoring, fault detection, …)

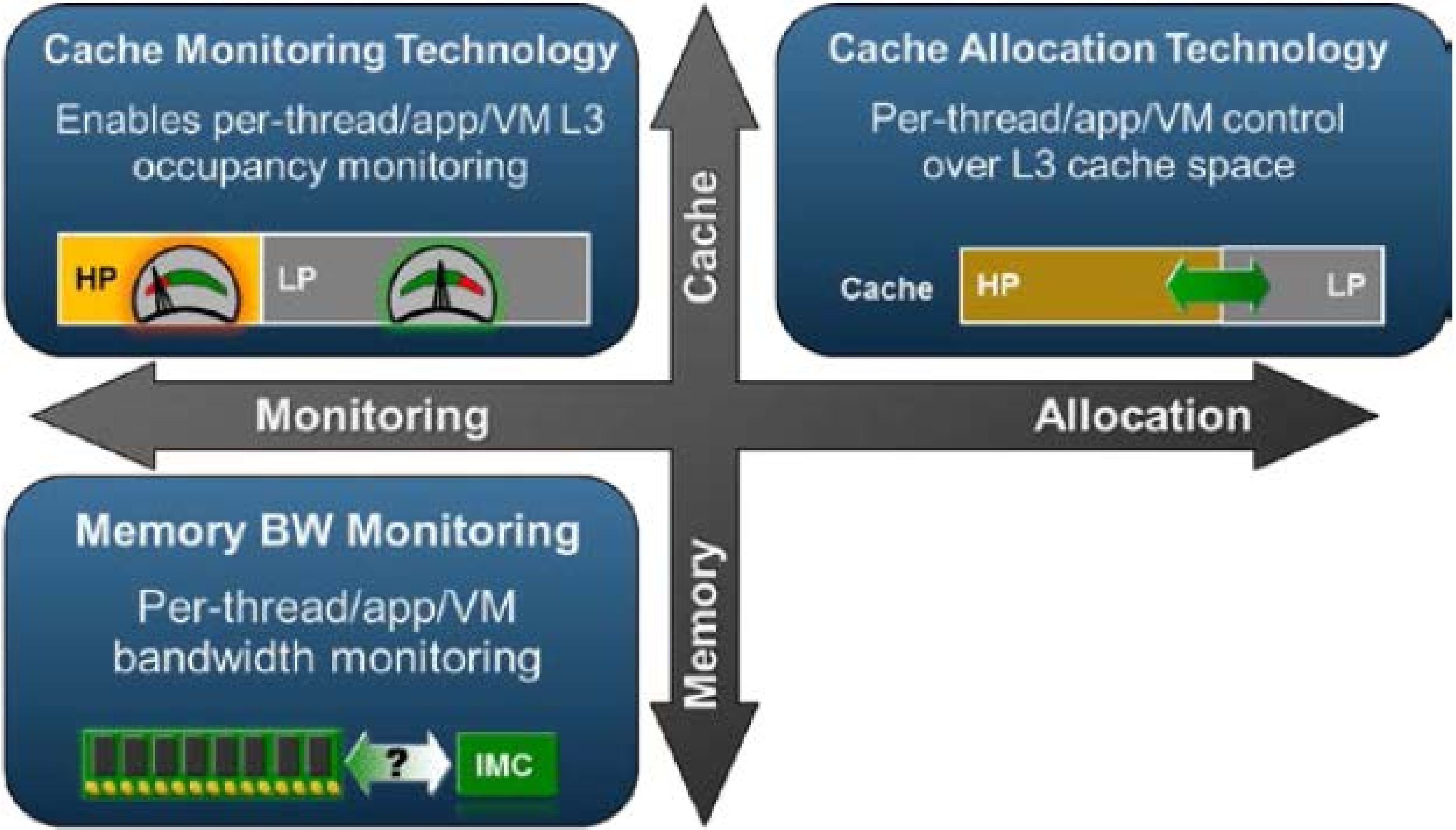
A simpler time…

- Commodity Intel CPU server resources back in the day before 2006
  - CPU w/ one core
  - Memory
- *One core → one job per server*
- Existing workloads gained substantial speed-ups just from processor upgrades
  - higher clock speeds
  - supposedly "smarter" processors (e.g. from simple in-order processors to complex superscalar out-of-order processors)

Things start to get complicated

- CPU clock speeds stagnated (due to heat)
- Industry moved to CPUs with multiple cores in Yr 2006, and servers with multiple CPUs
- *Multiple cores → multiple jobs per server*
- More scheduling complexity
- More monitoring/isolation complexity
  - Linux Control Groups (cgroups) : memory, cpu, io, network
  - Still a work in progress…

Intel Resource Director Technology (RDT)



Death of Moore's Law making even more complexity

› Yr 2016 or so Moore's Law died, killed by
  - Economics (Rock's Law)
  - Physics (5nm ~ 150 atoms between features?)
  - "Smart" processors? (Foreshadow, Spectre, Meltdown, …)

› Result? GPUs, Knight Landing, FPGAs, coming soon quantum computing …

› Now we have multi-axis co-scheduling complexity (and recursion with nVidia Volta)

› …not just complexity, heterogeneity! And the speed of change!

CHALLENGE AREA #2

EVOLVING RESOURCE ACQUISITION MODELS

Public Cloud Services

- Cloud services enable fast acquisition of compute infrastructure for short or long periods of time.
  - Enables homogenous, single user, single purpose clusters. *How do we best leverage that capability?*
  - Enables elasticity: augment existing on-site cluster with resources offered by competing commercial clouds. *Best on-the-fly capacity planning mechanisms? Budget-based scheduling? To cloud or not to cloud?*

Not just public clouds…

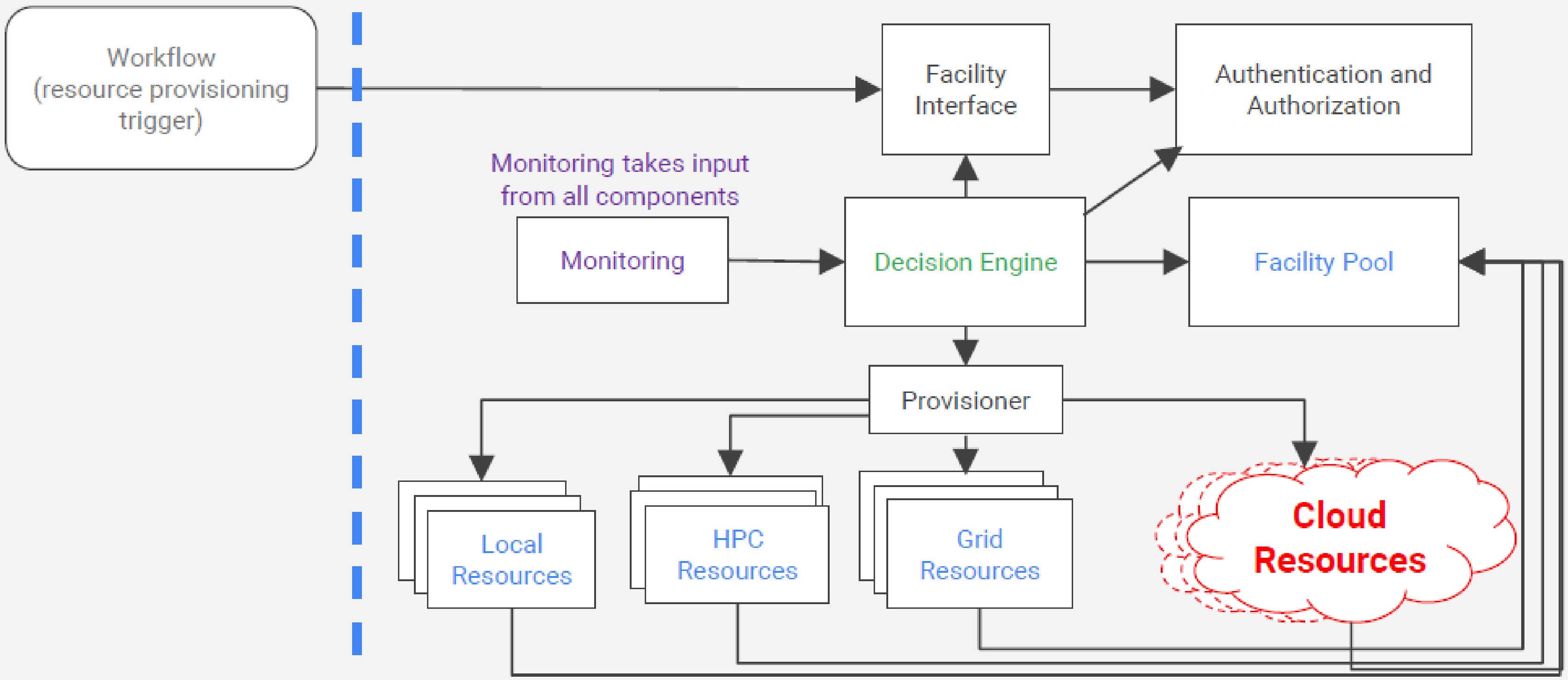
- HPC Supercomputers
  - Often have special considerations, like no network connectivity!
- Grids (federations of compute clusters across institutions), e.g. WLCG, Open Science Grid
- Private Clouds
  - To consolidate IT servers, IT data centers at home institutions increasingly using 'private cloud' services
  - OpenStack, Apache Mesos, Docker Swarm, *Kubernetes!*
- Technologies are changing rapidly!

HTCondor has long allowed dynamic augmentation of a cluster…

- HTCondor can be used to submit and track "pilot" or "glidein" jobs (aka the HTCondor daemons themselves) to remote cluster/cloud services to add execute nodes
  - Amazon EC2, Google Cloud, Microsoft Azure, SLURM, PBS/Torque, SGE, HTCondor, ...
- Cluster augmentation frameworks evolving
  - glideinWMS, pyglidein, HEPCloud, COBalD, …

HEPCloud

HEPCloud Architecture



(B. Holtzman; FNAL)

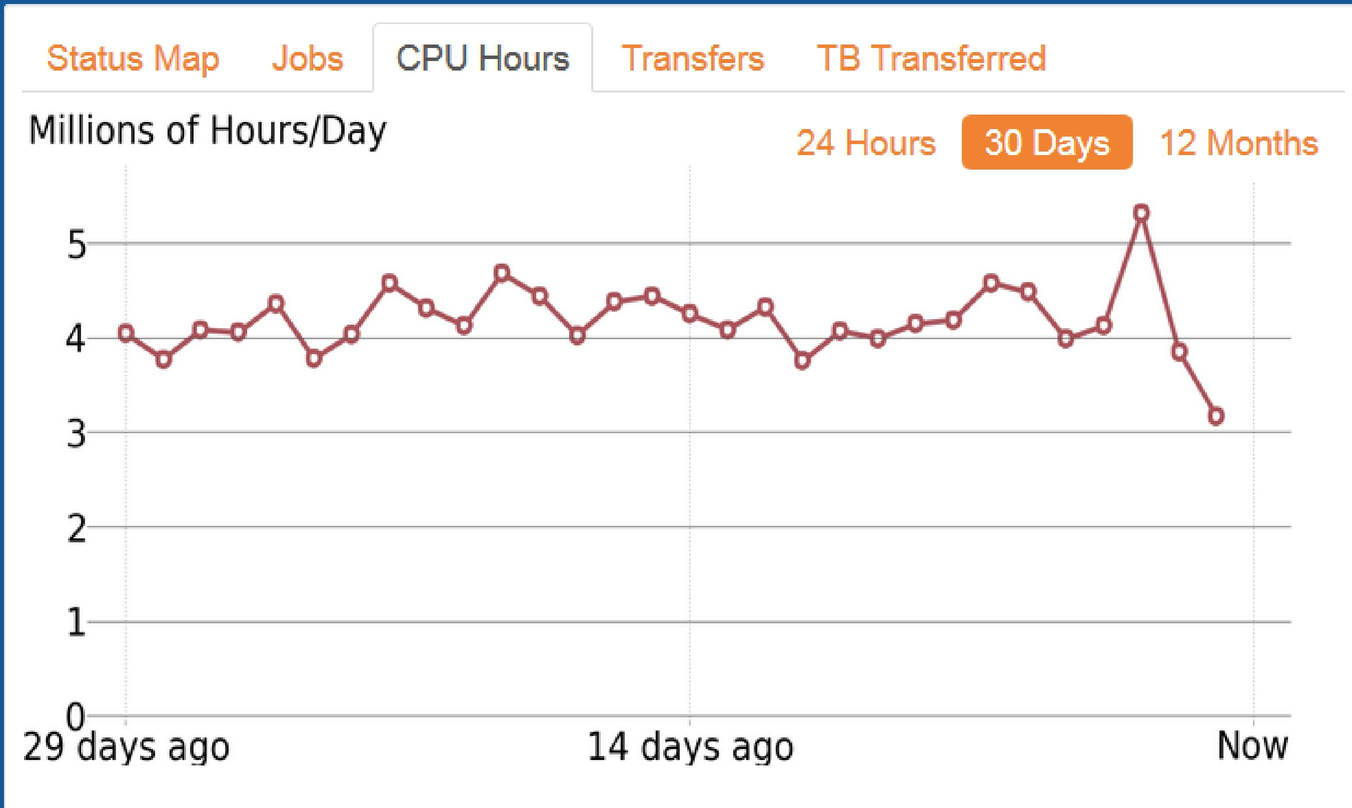
Fermilab 50 Years of Discovery

CENTER FOR HIGH THROUGHPUT COMPUTING

HTCondor

Some Examples

Open Science Grid



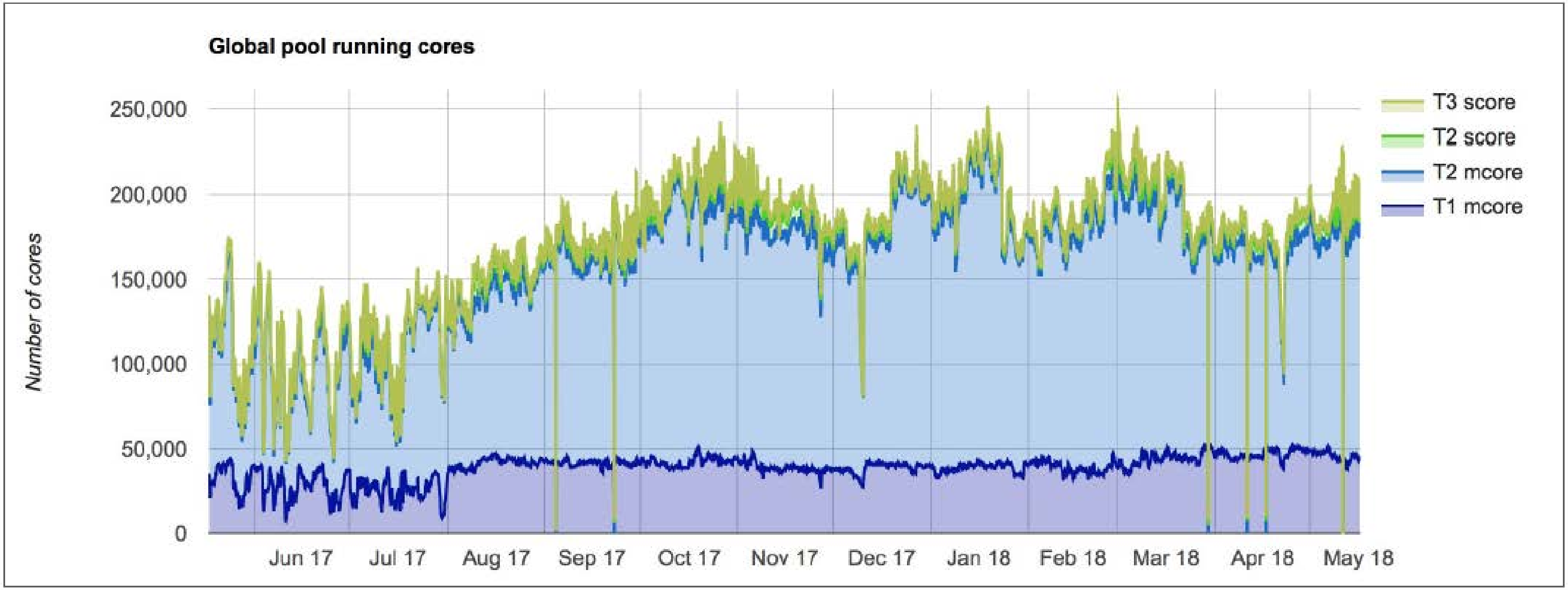
In the last 24 Hours	
330,000	Jobs
4,668,000	CPU Hours
4,722,000	Transfers
368	TB Transfers
In the last 30 Days	
7,729,000	Jobs
125,707,000	CPU Hours
12,517,000	Transfers
3,496	TB Transfers
In the last 12 Months	
127,924,000	Jobs
1,618,355,000	CPU Hours
2,083,299,000	Transfers
234,000	TB Transfers

OSG delivered across 117 sites

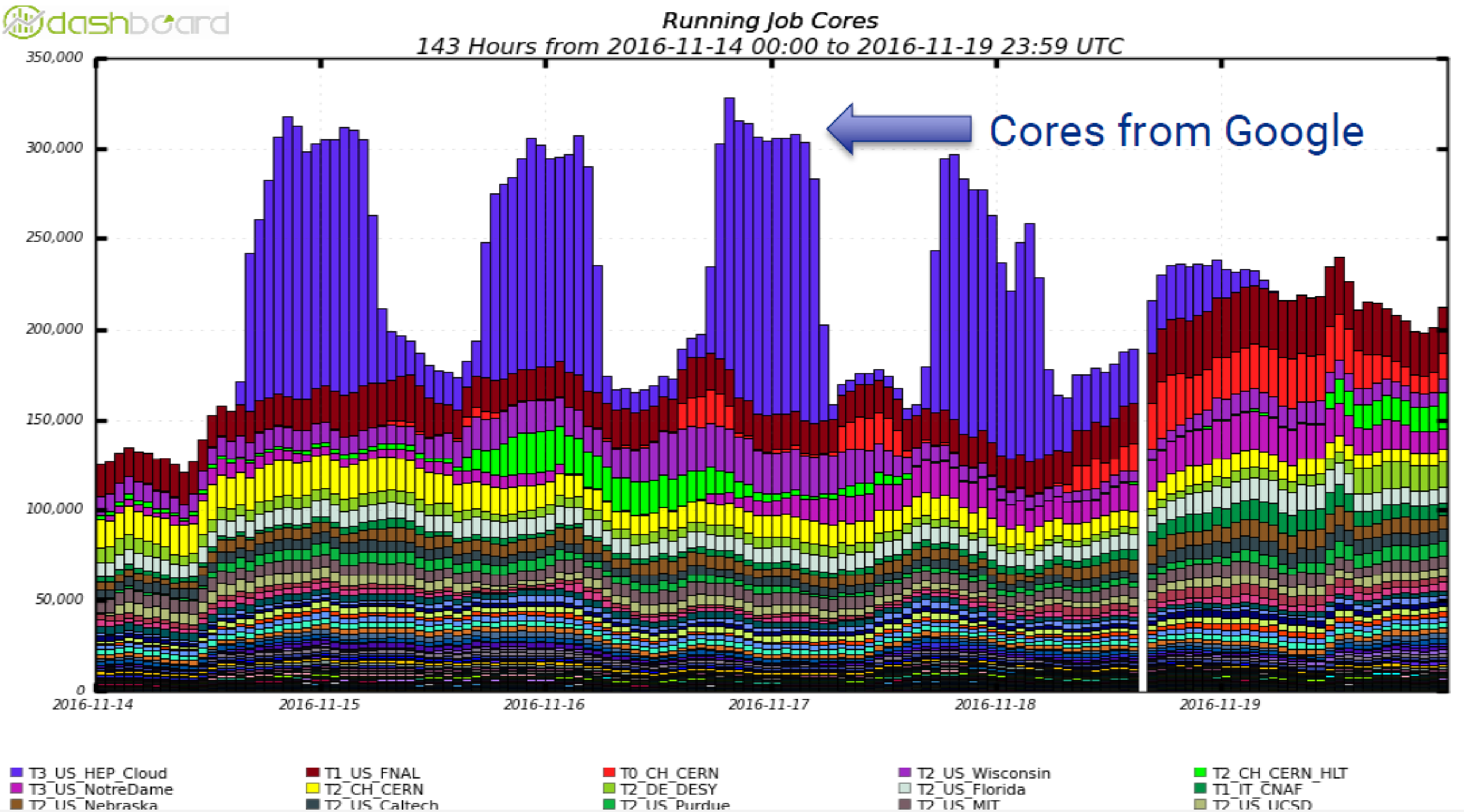
http://display.opensciencegrid.org

CMS Global Pool

- Dynamic cluster, ~200k - 250k cores pulled in from sites worldwide (including KIT!)



Bursting into Google Cloud @ SC16



Bursting into HPC

Fermilab comp
NOvA evidence
consumed

July 3, 2018

Share Tweet G+

How do you arrive at the physical
data on a renegade particle that in
light-years of lead? You call on the

The NOvA neutrino experiment, in collaboration with the Department of Energy's Scientific Discovery through Advanced Computing (SciDAC-4) program and the HEPCloud program at DOE's Fermi National Accelerator Laboratory, was able to perform

b augments
Condor
th nodes
ERSC
mputer to
rm the
antineutrino
lysis ever
in record time.

http://news.fnal.gov/2018/07/fermilab-computing-experts-bolster-nova-evidence-1-million-cores-consumed/

CHALLENGE AREA #3

SCALABILITY

No shortage of work to do here

- Growth everywhere you look
  - manufacturers give us more cores
  - researchers bring us more jobs
  - universities and science communities bring us more users
  - cloud providers offer us more machines
- Operational scaling
- Grouping for both operations and user/admin interactions
  - Who can effectively reason about millions of individual jobs or machines?
- Reliability, damage from "black holes" nodes

CHALLENGE AREA #4

WIDELY DISPARATE USE CASES

A growing spectrum of scenarios

- Increased demand for higher throughput, HTC technologies are being called upon to serve in a growing spectrum of scenarios:
  - Large multi-purpose institutional clusters managed by IT experts
  - Ephemeral overlay clusters atop other batch systems (grid computing)
  - Purpose built clusters from a cloud provider
  - Cycle scavenging server farms (K8), desktops
  - Manage a workflow on a single server (laptop)
  - *Non-batch interactive computing environments such as Jupyter Lab / Notebook*
- And a growing spectrum of user backgrounds

CHALLENGE AREA #5

BLACK BOX APPLICATIONS

HTC Users aren't just Unix Wizards anymore

- Contemporary HTC users, many with no experience with large scale computing, are much less knowledgeable about the codes they run than their predecessors.

Our Goal:

You do not need to be a computing expert in order to benefit from HTC!

- Meeting this goal requires HTC frameworks that can effectively manage work when the user cannot state the
  - Software dependencies
  - Resource requirements
  - Compute time

of their application.

"YOU SHOULD HAVE SAID YOU WANTED CHAIRS WHEN YOU BOOKED THE TABLE!"

CHALLENGE AREA #6

DATA INTENSIVE COMPUTING

Manage data movement *and* manage data storage.

- We all know the story: more data pouring in from everywhere.
- Everyone fixated about managing and scheduling data transfers… but what about the storage?
- All the software we write assumes that disk space is infinite. Once you remove that assumption, what happens?

Thank you!

And now time for your feedback!

Results from survey at GridKa School 2018