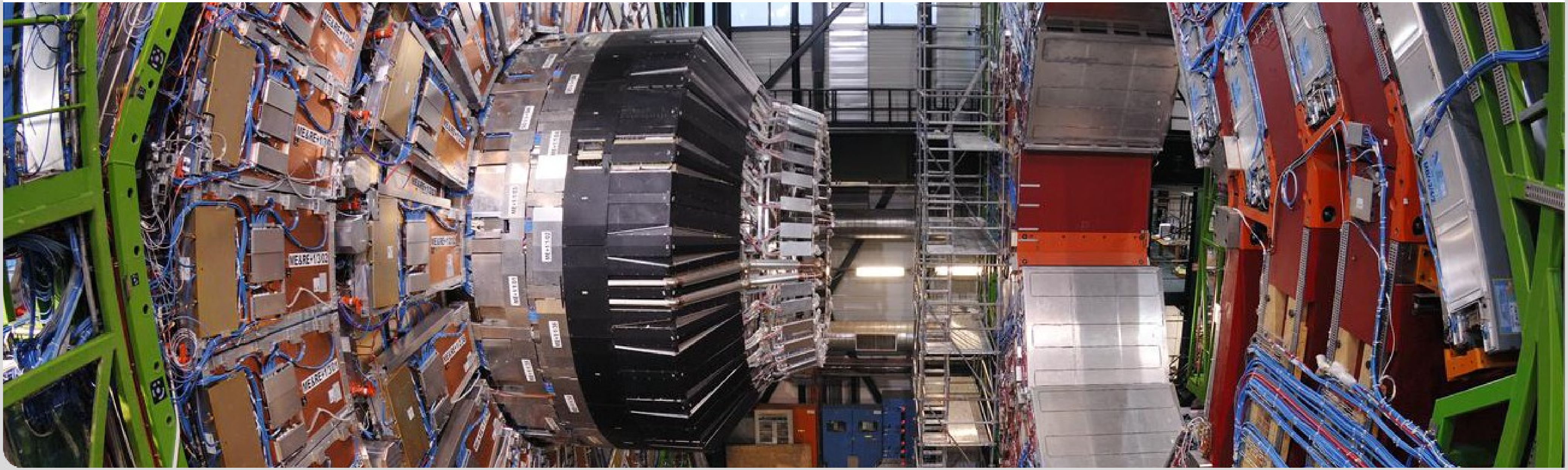
Using Opportunistic Resources at HPC Centers

2.12.2014 - 8th Annual Meeting of the Helmholtz Alliance "Physics at the Terascale" - Hamburg

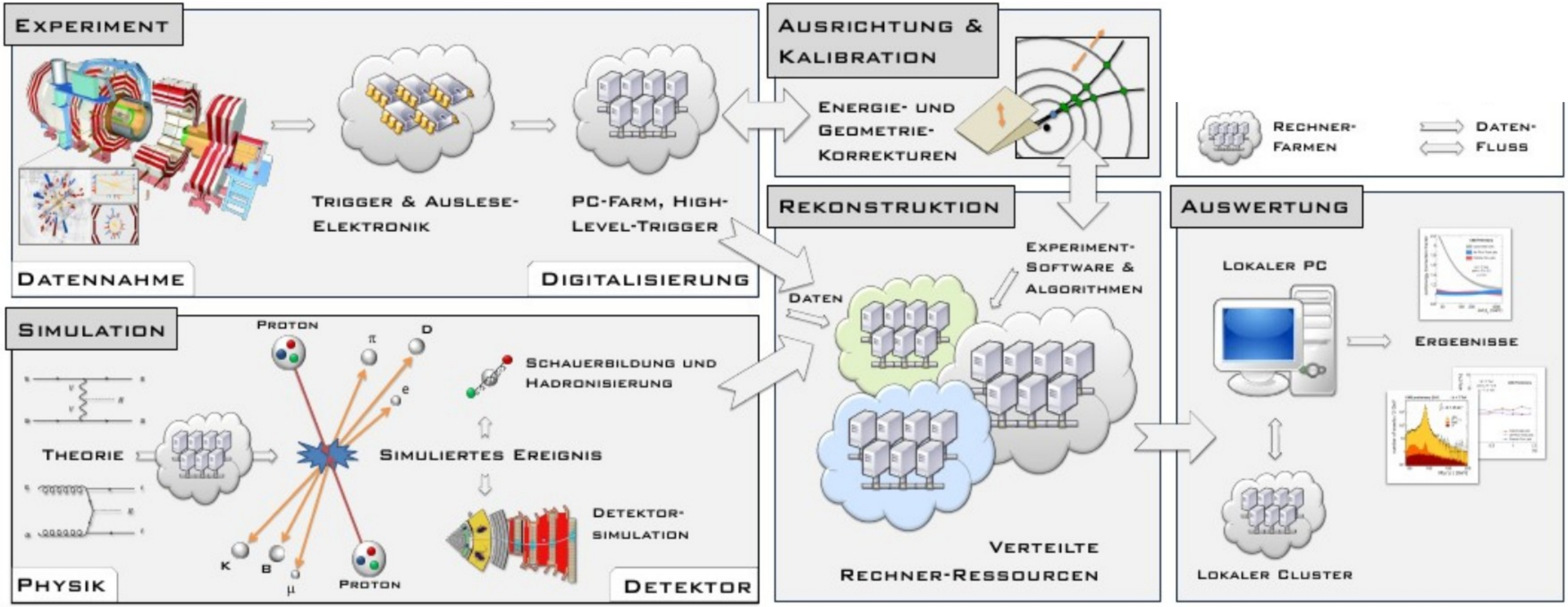
Thomas Hauth, Günter Quast, Manuel Giffels, Max Fischer, Frank Polgart
Institut für Experimentelle Kernphysik (IEKP), KIT



Overview

- High Energy Physics computation in the domain of Scientific Computing
- Virtualization as a doorway to new computing resources
- Making opportunistic HPC resources available to the Karlsruhe HEP users
- Conclusion and Outlook

Diverse Computing in HEP

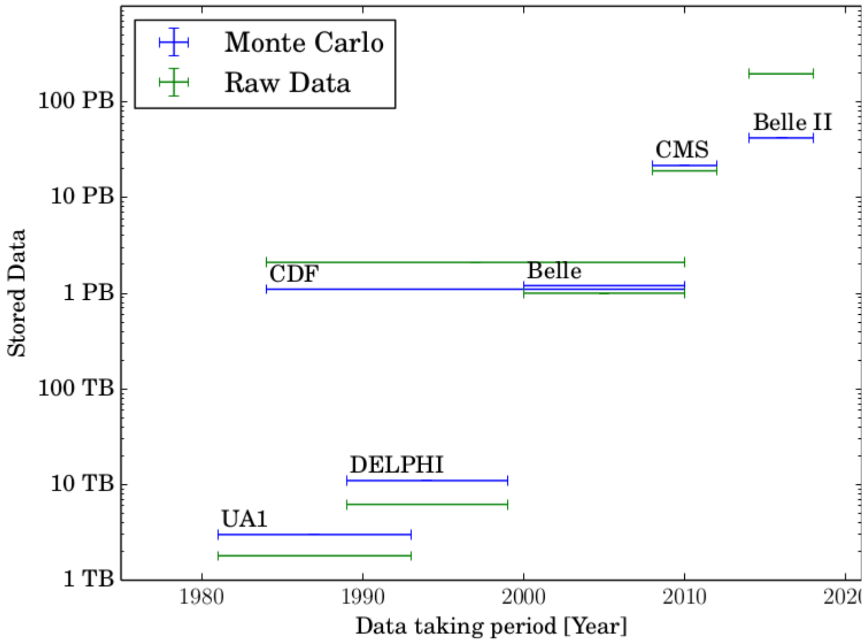


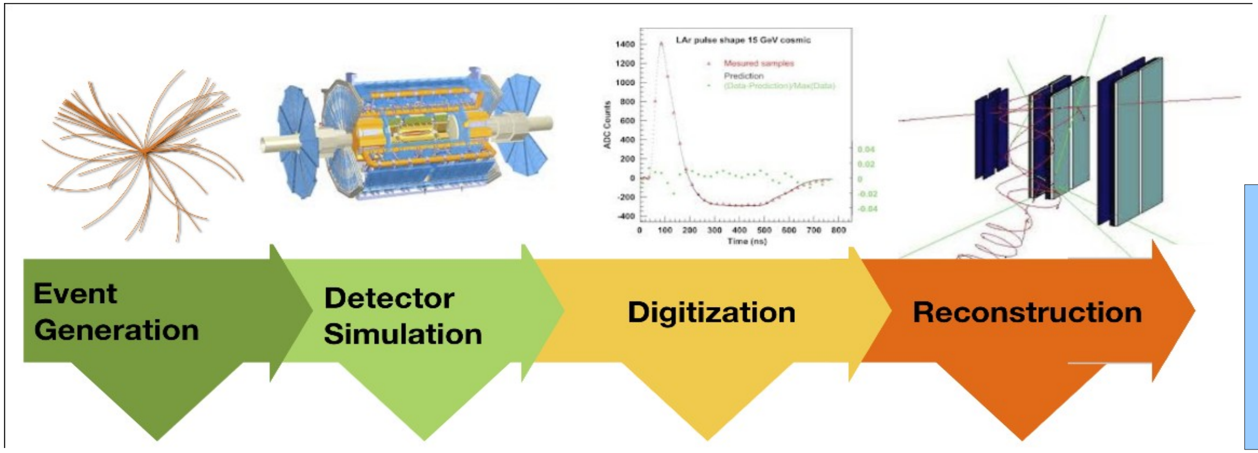
- HEP software applications are very diverse with different requirements in terms of I/O and compute
- Some applications need to be located at specific sites
  - HLT farm must be close to the detector
  - Event reconstruction needs fast access to the measurements stored on disk
- Simulation and analysis can be located more freely

A Detailed Look: Monte Carlo Simulation

- Monte Carlo simulated events are an essential tool to compare detector measurement with theory prediction
- About the same number of simulated as measured events necessary to make statistically significant conclusions
- Chain of multiple simulation programs involved
- Size of one Monte Carlo dataset: ~ 10 GB

  O(10 Mio. Events) with 1MB per Event





Event Generation → Detector Simulation → Digitization → Reconstruction

Storage on disk for later analysis

http://iopscience.iop.org/1742-6596/513/2/022006

High [Performance | Throughput] Computing

High Performance Computing (HPC)

“focuses on the efficient execution of compute intensive, tightly-coupled tasks.” *

High Throughput Computing (HTC)

“focuses on the efficient execution of a large number of loosely-coupled tasks.” *

Nearly all High Energy Physics workloads belong to the HTC category

- No shared data between individual work items
  - Local concurrency: Multi-core jobs with up to 8 threads possible
- The size of one work item can be freely chosen via the number of processed events
  - Typical runtime of one job ~ 8h

This has advantages for cluster operators

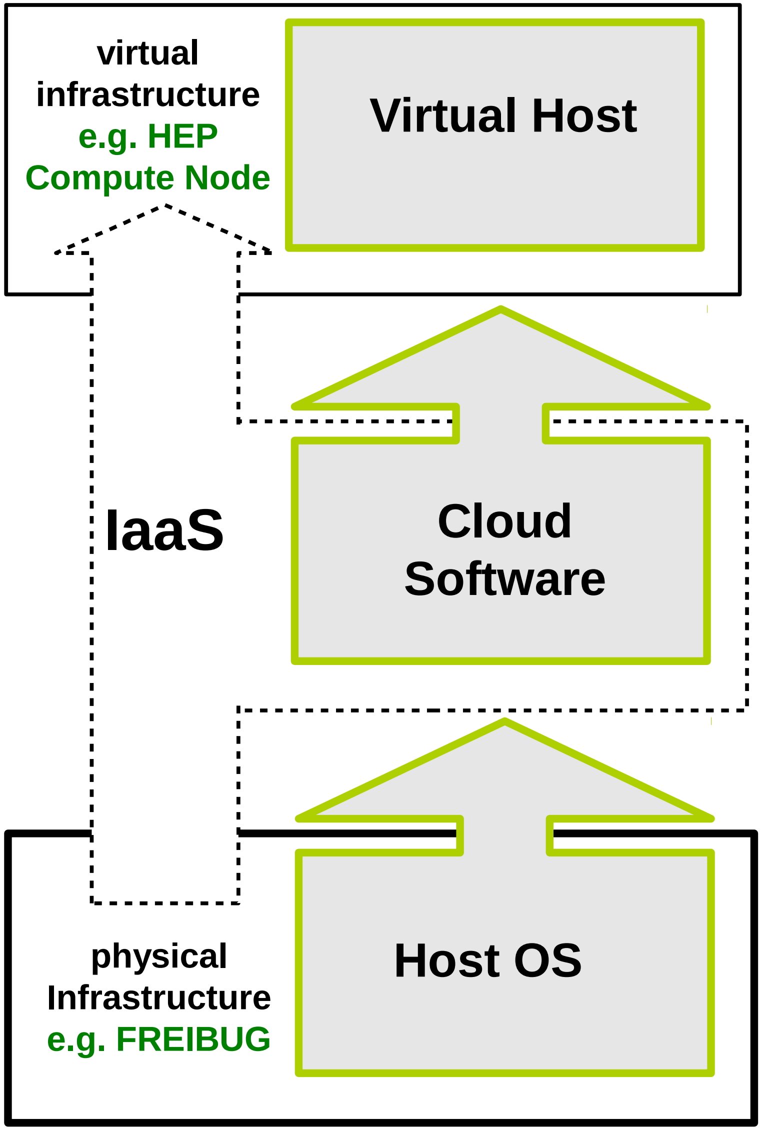
- HEP batch jobs can be placed anywhere (no fast interconnect to related jobs needed)
- Backfill of HEP jobs can be run also in smaller quantities to fully load a partially occupied cluster

But typical centers HPC have a different user profile

- Exclusive booking of the large parts of the cluster by one user
- Long running (~days), inter-dependent computations by only a few jobs

* according to the European Grid Infrastructure (EGI) - https://wiki.egi.eu/wiki/Glossary_V1#High_Throughput_Computing

From Physical to Virtual Infrastructure



The Infrastructure-as-a-Service (Iaas) model

- Infrastructure (e.g. machines, network) is virtualized
- Decouples complexities of hardware maintenance and specific software setup

- The life cycle of this virtual infrastructure is managed by a Cloud system:
  - Virtual machine images are managed
  - The user can upload and start custom virtual machines
  - Storage blocks can be attached to these VMs

OpenStack
Cloud Operating System

- Complete open source **IaaS framework**
- Backed by lots of big companies (IBM, Red Hat, Rackspace, HP, ...)
- **Standardized API** (Amazon EC2, fits with HEP workflow tools)
- Acceptable to HPC Centers (already in use)
- Active Involvement of the HEP Community (CERN Personnel in OpenStack Foundation Board)

proven software

long term support can be expected

Private OpenStack installation in Karlsruhe

- High-performant desktop machines (4-core, 16 Gb RAM, Ubuntu) run Scientific Linux VMs
- VMs are integrated into the institute's batch system
  - Desktop machines rarely fully loaded by users
  - The extra load generated by batch jobs barely noticeable by desktop users
- **Working example of opportunistic resource usage**

Introduction HEP batch systems

Pilots are jobs that integrate multiple computing resources by launching remote batch worker nodes and registering them locally.

- **HTCondor** as local and remote **batch system**
  - Client-server-architecture
  - Free & open source
  - Specifically designed for HTC workloads
  - Implements pilot-jobs called “GlideIns”

- **GlideinWMS** as **automation wrapper** for on-demand GlideIn creation.
  - Cloud-aware: can provision VMs via EC2 interface
  - KIT runs a GlideInWMS system
    - Integrated with local OpenStack: pilot jobs can be send to virtual machines

HEP Examples of Cloud technology usage

To access opportunistic resources, the OpenStack cloud platform has become very popular

- OpenStack overlay was installed on HLT machines
- KVM HyperVisor with Scientific Linux 5

  64-bit image is used
- CERN VM FS is used to deliver the experiment software to the machines
- Virtual machines of the batch jobs do not interfere with the HLT-configuration of the individual Nodes
- Virtualization allows for fast switching between HLT and VM mode of the cluster

CMS High-Level-Trigger Farm

- Not used during LHC downtimes
- Can offer significant compute capabilities (13k cores) during these times
- 20Gbps link to CERN IT compute center
- HLT software runs natively on machines
- No interruption of primary HLT duty acceptable
- OpenStack overlay installed on HLT machines

ATLAS High-Level-Trigger Farm

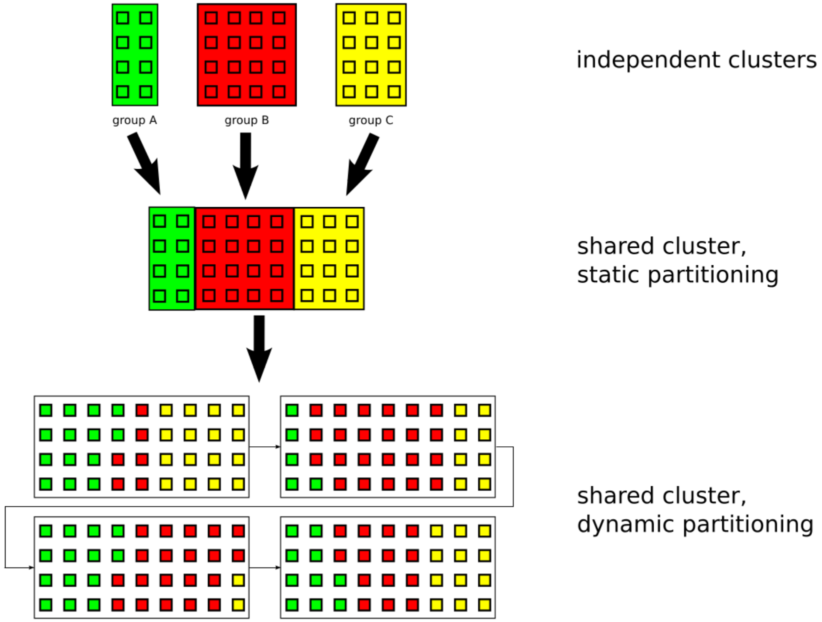
- Also using OpenStack overlay on 15k cores
- ~ 45 minutes time to provision all virtual machines
- ~ 10 minutes to shutdown VM and restore the HLT-mode of the cluster
  - Dynamic usage of resources, even possible to exploit short LHC shutdowns

http://indico.cern.ch/event/273593/contribution/1/material/slides/1.pdf
http://iopscience.iop.org/1742-6596/513/3/032019/

Opportunistic Computing Experience in Karlsruhe

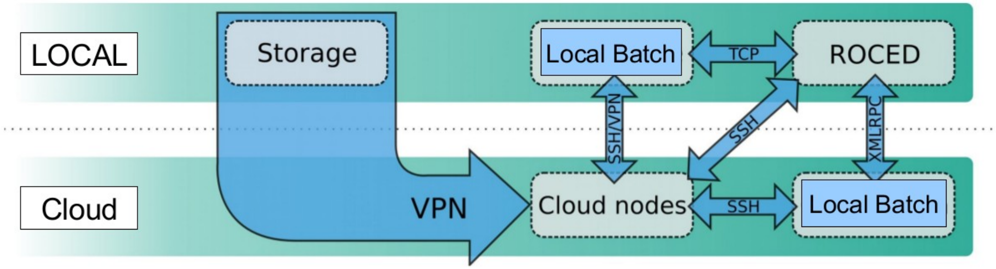
ViBatch Project

- System to dynamically partition computing cluster
- Pre-Cloud era: virtual machines are started by ViBatch
- User job are automatically transferred and executed inside the virtual machine
- Fully transparent to the user
- Deployed at KIT to allow HEP users to utilize a cluster which is shared among many institutes



Cloud Integration via ROCED

- Transparent integration of local or remote cloud resources in batch server
- Modular design allows for flexible batch server and cloud provider configurations
- Supported cloud providers:
  - OpenNebula
  - Amazon EC2 and compatible
  - OpenStack Nova API (in development)
- Supported batch servers:
  - Oracle Grid Engine
  - Torque



More Information:

Dynamic Extension of a Virtualized Cluster by using Cloud Resources
Oliver Oberst et al 2012 J. Phys.: Conf. Ser. 396 032081
http://iopscience.iop.org/1742-6596/396/3/032081

Both projects profited from Alliance funding.

HPC Resources at the Compute Center Freiburg

- The **bwForCluster** installation is a HPC system which will be shared by 3 diverse user groups:
  - Particle Physics
  - Neuroscience
  - Microsystems technology

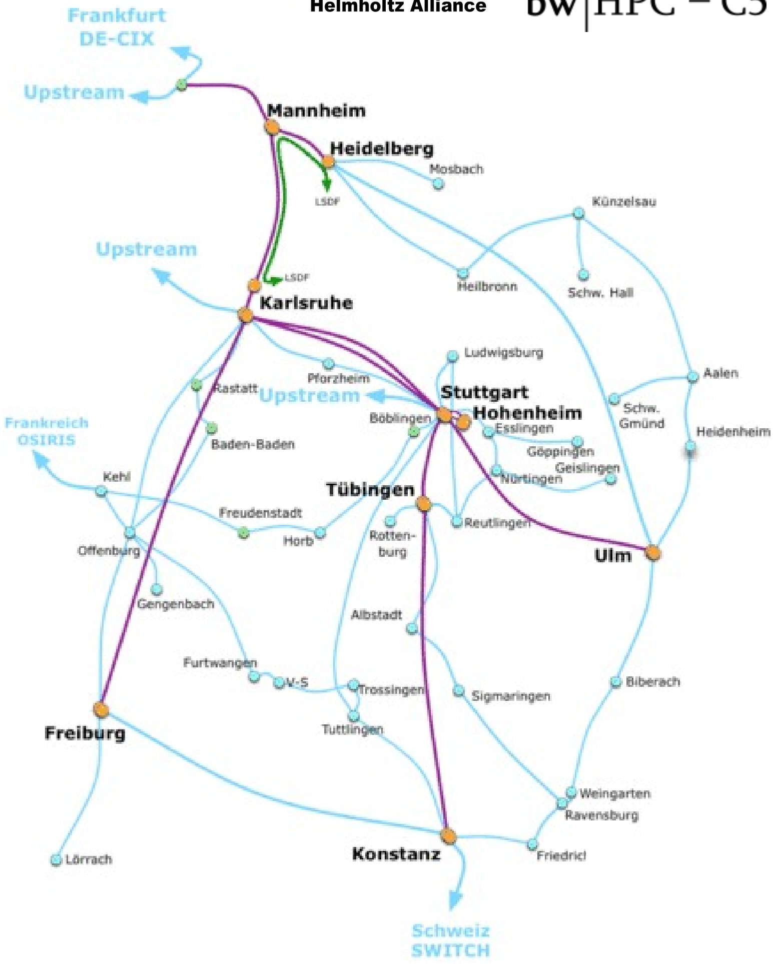
- State funding has been secured this month and the full installation with ~8000 cores is expected to be available in May 2015
- Already in the design phase, it became clear that virtualization will be a key technology to allow for a efficient sharing among the user groups
- A 10Gbit BelWü link between Karlsruhe and Freiburg Universities allow for an efficient data transfer (possible to upgrade to 100Gbit)

Current Test Installation

- The Freiburg Rechenzentrum group is providing a test system similar to the bwForCluster installation today
- Fully functional OpenStack setup with 400 cores

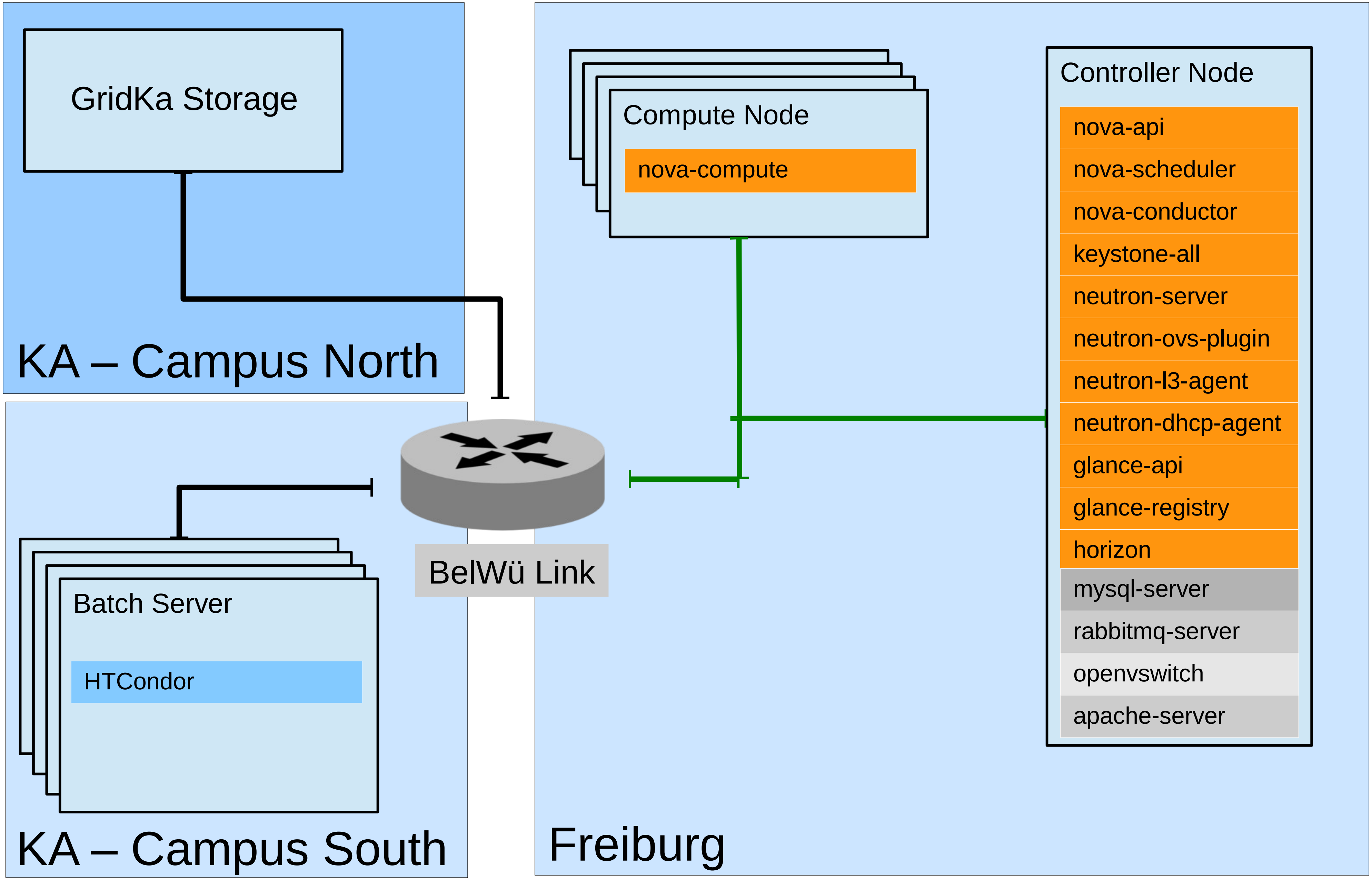
PHYSICS AT THE TERA SCALE
Helmholtz Alliance

bw|HPC – C5



http://www.belwue.de/

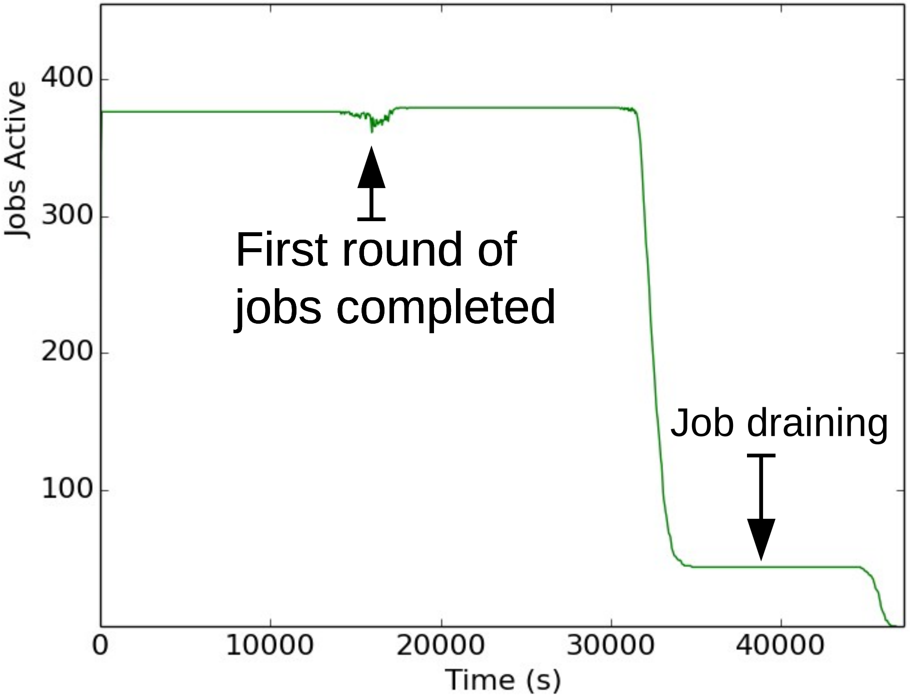
Topology Karlsruhe <> Freiburg



CMS Production Test @ bwForCluster Test System

- Testing the overall system (Condor, OpenStack) with a CMS Monte Carlo simulation of ttbar decays
- Output is written to the storage element at Tier1 GridKa at the KIT Campus North
- 800 Jobs with 200 Events each = 160k events in total
- Runtime per job: ~ 4h

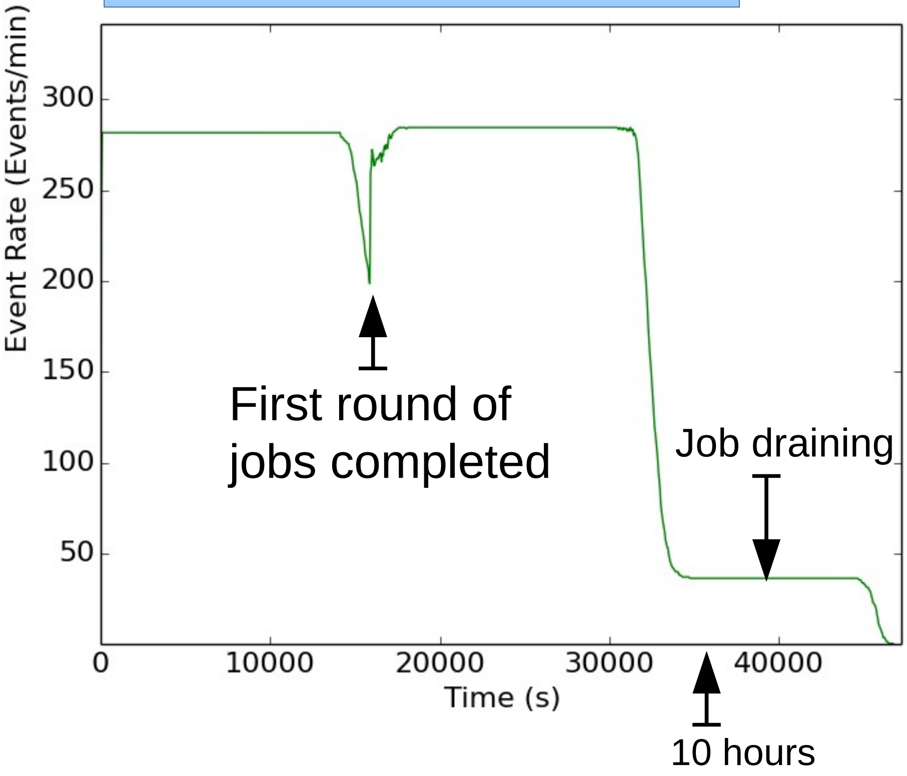
Active Jobs



First round of jobs completed

Job draining

Total Event Production Rate



First round of jobs completed

Job draining

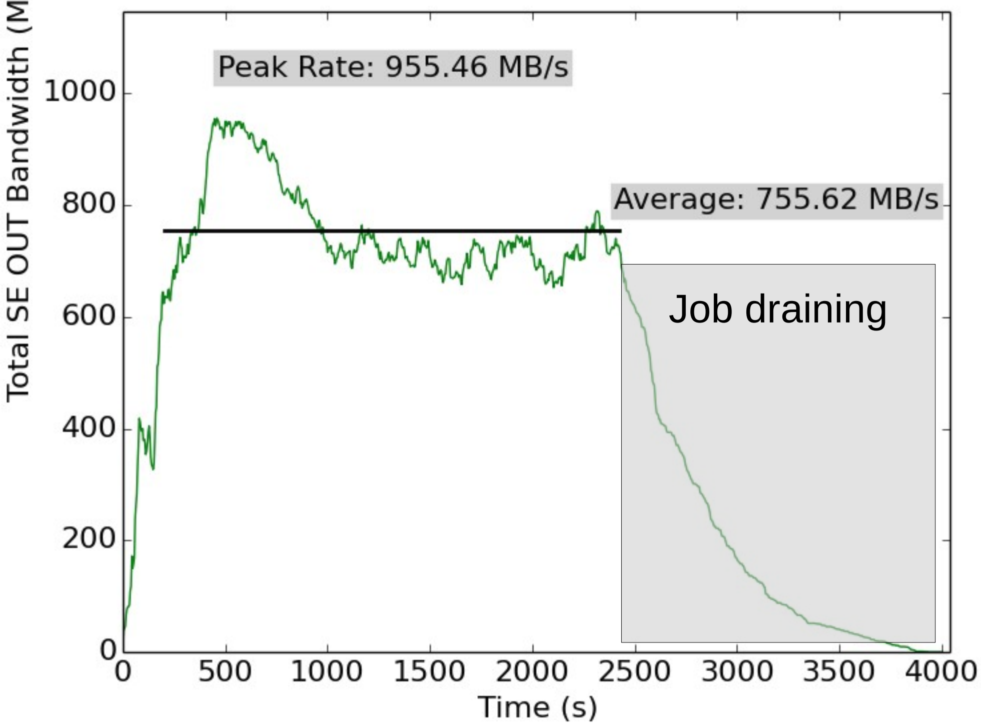
10 hours

Result: Stable performance of the Condor/bwForCluster test system

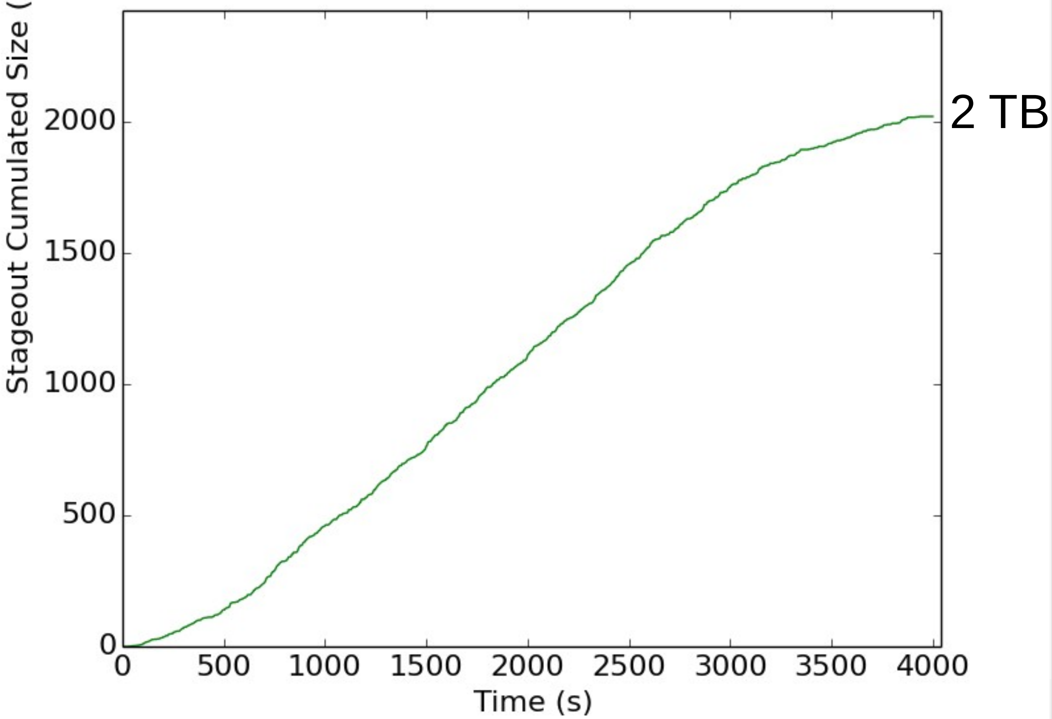
Evaluation of file transfers from Freiburg Virtual Machines to GridKa

- To saturate the link and use all network ports of the Freiburg router, sufficient jobs must perform stage-outs at the same time
- On average 320 jobs store files via GridFTP at GridKa at the same time
  - Realistic scenario for large-scale Monte Carlo productions

Total Transfer Bandwidth



Cumulated Transferred Data



- An **average bandwidth of 755.62 MB/s** was achieved during test period (truncated at start & end)
- 2 TB have been transferred at the end of the test
- The evaluation shows that sufficient bandwidth between the Freiburg VMs and Grid-Ka is available
- **Result: Large-scale Monte Carlo productions can be performed on the Freiburg site**

Conclusion

- Virtualization is essential to allow access for the HEP community to shared HPC resources
- Mature software for virtualization (KVM, OpenStack) and flexible work management (HTCondor) exists today
  - And even in HEP: CVMFS, GlideinWMS
- The bwForCluster project uses virtualization to solve the diverse requirements of its tenants
- First evaluations of the bwForCluster test system successful

Outlook

- Accompany the bwForCluster procurement with our virtualization requirements in mind
- Discussions on governance and operational details with the other user groups of the bwForCluster are ongoing:
  - How to account for VM-hours in contrast to native batch slots
  - How to schedule VM in between multi-node batch requests

Backup

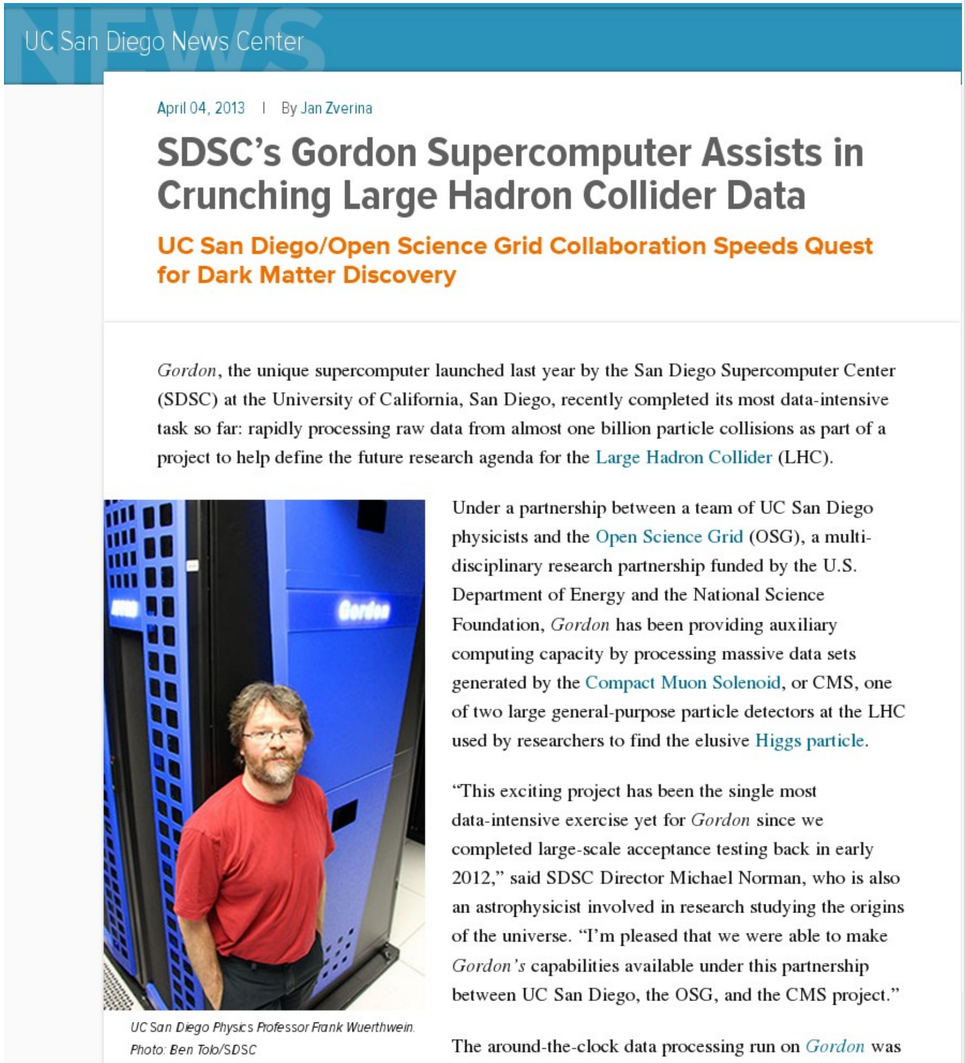
HEP Examples of HPC resource usage

- CMS uses San Diego Supercomputing Center

  “Gordon” system
  - 16k SandyBridge cores
  - Reconstruction of almost 1 Bil. CMS events witihin 4 weeks in April 2013

“Giving us access to the Gordon supercomputer effectively doubled the data processing compute power available to us,” added Lothar Bauerdick (U.S. software and computing manager for the CMS project)

- ATLAS has access to the SuperMUC system in Munich
  - 155k SandyBridge cores
  - OpenStack + ATLAS' custom job management system



UC San Diego News Center

April 04, 2013 | By Jan Zverina

SDSC’s Gordon Supercomputer Assists in Crunching Large Hadron Collider Data

UC San Diego/Open Science Grid Collaboration Speeds Quest for Dark Matter Discovery

Gordon, the unique supercomputer launched last year by the San Diego Supercomputer Center (SDSC) at the University of California, San Diego, recently completed its most data-intensive task so far: rapidly processing raw data from almost one billion particle collisions as part of a project to help define the future research agenda for the Large Hadron Collider (LHC).

Under a partnership between a team of UC San Diego physicists and the Open Science Grid (OSG), a multi-disciplinary research partnership funded by the U.S. Department of Energy and the National Science Foundation, *Gordon* has been providing auxiliary computing capacity by processing massive data sets generated by the Compact Muon Solenoid, or CMS, one of two large general-purpose particle detectors at the LHC used by researchers to find the elusive Higgs particle.

“This exciting project has been the single most data-intensive exercise yet for *Gordon* since we completed large-scale acceptance testing back in early 2012,” said SDSC Director Michael Norman, who is also an astrophysicist involved in research studying the origins of the universe. “I’m pleased that we were able to make *Gordon’s* capabilities available under this partnership between UC San Diego, the OSG, and the CMS project.”

The around-the-clock data processing run on *Gordon* was

UC San Diego Physics Professor Frank Wuerthwein. Photo: Ben Tolo/SDSC

http://acat2013.ihep.ac.cn/proceedings/papers/A118-68-ATLAS_Distributed_Computing__Experience_and_Evolution.pdf

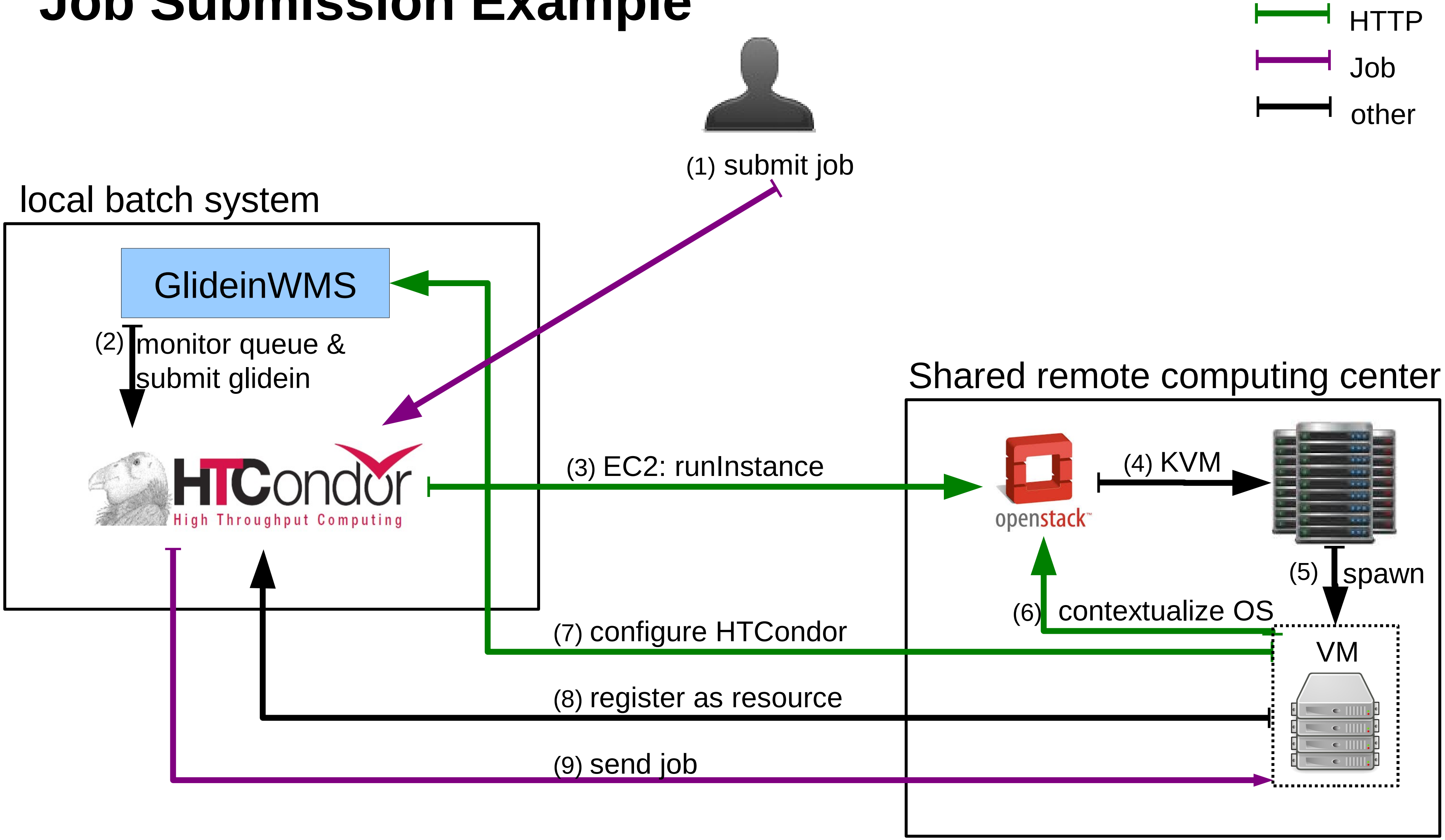
http://ucsdnews.ucsd.edu/pressrelease/sdscs_gordon_supercomputer_assists_in_crunching_large_hadron_collider_data

Custom CMS Virtual Machine Image

- VM images are built with the OZ toolkit
  - Installs an OS, software and configurations according to template files (.tdl)
- Advantages:
  - Reproducibility
  - Easy process (run `./buildImage.sh <template file>`)
  - Easy to adapt working templates to new sites > KA to FR (but still copy paste)
- We used the CERN OpenStack SL6.5 image templates as starting point:

  https://github.com/cernops/openstack-image-tools
- A lot of modifications:
  - Modified network config to accept changed MAC addresses (needed for cloud boot)
  - CVMFS for Grid UI & CMS software
  - Disabled yum auto-update
  - Installed GlideInWMS pilot launcher
  - Removal of unneeded files (/usr/share/man, /usr/share/doc, desktop backgrounds, cups)
  - Filled the free disk space with zeros > allows for better compression of the image
    - Final Image size (1.3 Gb, before 2.1 Gb)

Job Submission Example



HTTP
Job
other
(1) submit job
local batch system
GlideinWMS
(2) monitor queue & submit glidein
HTCondor High Throughput Computing
(3) EC2: runInstance
Shared remote computing center
openstack
(4) KVM
(5) spawn
(6) contextualize OS
VM
(7) configure HTCondor
(8) register as resource
(9) send job