- Motivation
- Past experience
- The LHConCRAY project
- System architecture and integration
- First preliminary performance data
- Outlook and Conclusions

BIG DATA SCIENCE ACCESSING HIGH-END HPC

SWISS EXPERIENCE

Gianfranco Sciacca
AEC - Laboratory for High Energy Physics, University of Bern, Switzerland

Digital Infrastructures for Research 2016
Thursday 29 October 2016, AGH University of Science and Technology, Kraków, Poland

MOTIVATION

The challenges of Large Hadron Collider computing for the next decade

- **The Worldwide LHC Computing Grid is made mainly of ad-hoc engineered computing sites**
- **The WLCG model doesn't scale for High Luminosity LHC (beyond 2020)**
- Need (considerably) more computing for the same money

MOTIVATION

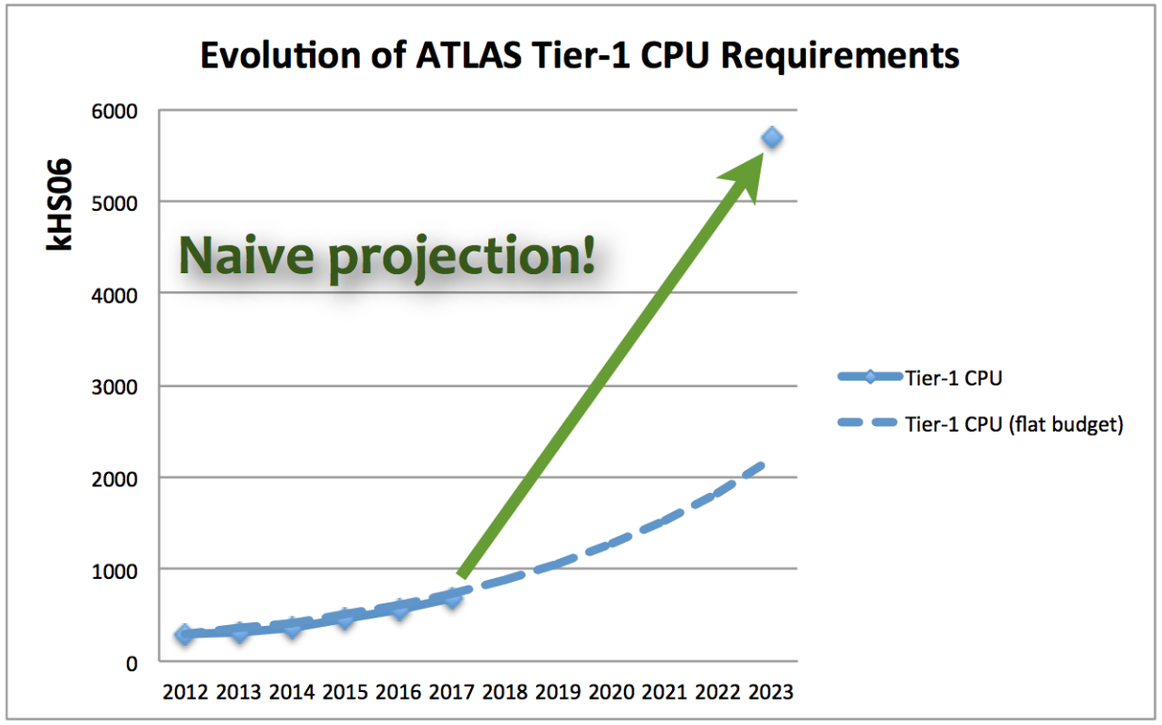
The challenges of Large Hadron Collider computing for the next decade

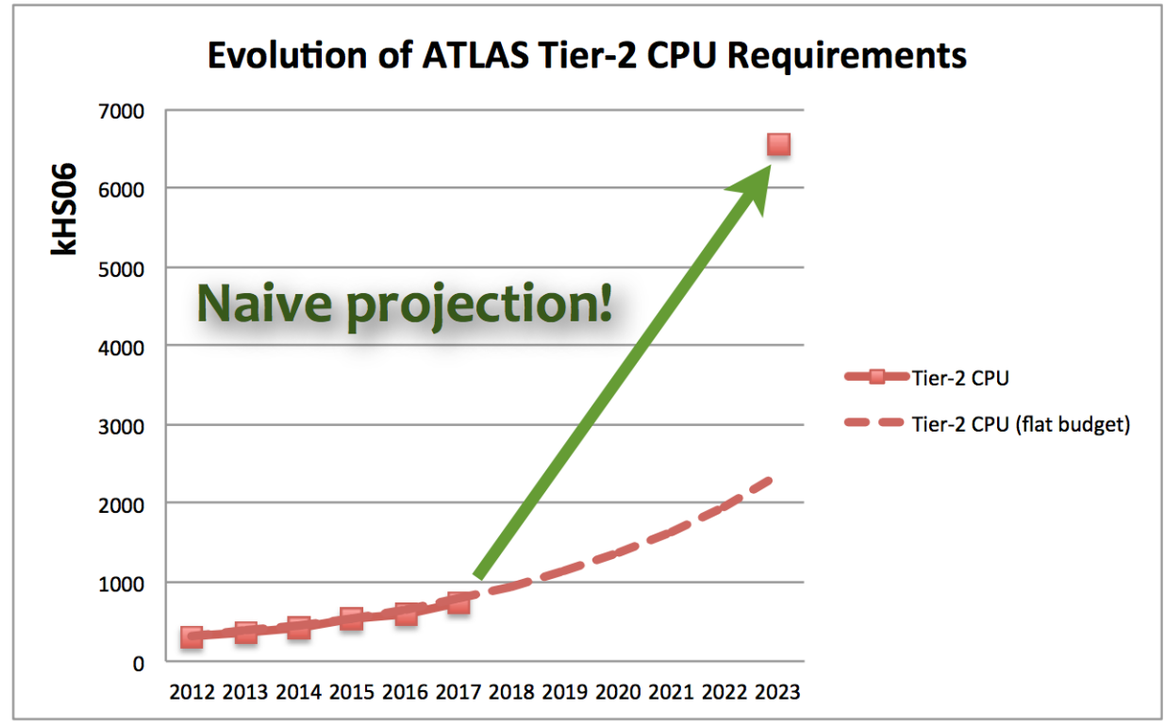
- The Worldwide LHC Computing Grid is made mainly of ad-hoc engineered computing sites
- The WLCG model doesn't scale for High Luminosity LHC (beyond 2020)
- Need (considerably) more computing for the same money

Assuming an order of magnitude increase in data and MC statistics

CPU: a drastic deviation from 'flat-budget'

BORUT KERSEVAN, JUNE '16





MOTIVATION

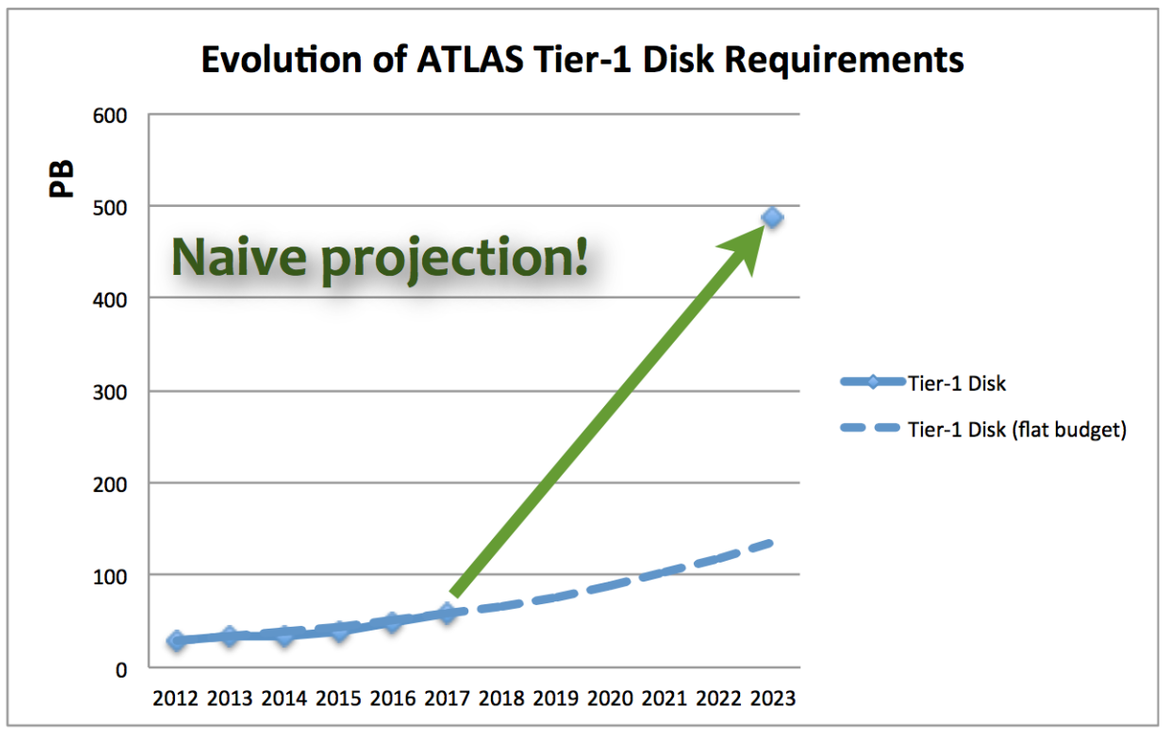
The challenges of Large Hadron Collider computing for the next decade

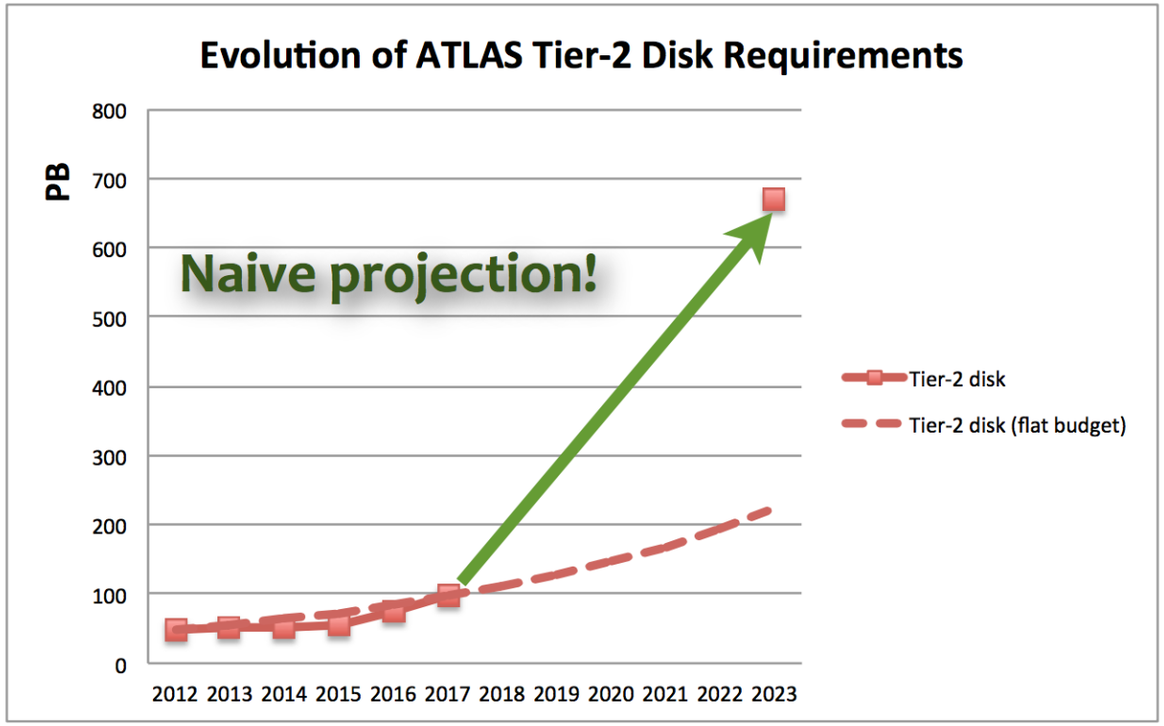
- The Worldwide LHC Computing Grid is made mainly of ad-hoc engineered computing sites
- The WLCG model doesn't scale for High Luminosity LHC (beyond 2020)
- Need (considerably) more computing for the same money

Assuming an order of magnitude increase in data and MC statistics

Disk: a drastic deviation from 'flat-budget'

BORUT KERSEVAN, JUNE '16





MOTIVATION

The challenges of Large Hadron Collider computing for the next decade

- **The Worldwide LHC Computing Grid is made mainly of ad-hoc engineered computing sites**
- **The WLCG model doesn't scale for High Luminosity LHC (beyond 2020)**
- Need (considerably) more computing for the same money

- **Part of the solution is to consolidate LHC computing**
  - Fewer but bigger sites world wide (operationally cheaper, high-end hw, etc)
  - General purpose HPC centres seem a good alternative to dedicated clusters
  - A lightweight operational approach can free up manpower a more effective exploitation of the resources
  - The computing site does "computing", exploiting the resources for the experiment needs is for the experts
  - Cheaper than having experiment experts at all computing sites
  - Leverage from economy of scales when procuring hardware (true? hopefully)

MOTIVATION

The challenges of Large Hadron Collider computing for the next decade

- **The Worldwide LHC Computing Grid is made mainly of ad-hoc engineered computing sites**
- **The WLCG model doesn't scale for High Luminosity LHC (beyond 2020)**
- Need (considerably) more computing for the same money

- **Several challenges arise**
  - **Processor architecture and/or OS might not always be suitable**
    complex software re-builds, environment tweaking, etc..
  - **Compliance with tight access rules**
    single-user access, username/password
  - **Application provisioning**
    a single ATLAS release is ~20GB, release cycles are very short/unpredictable
  - **Workload management integration**
    requires in general outbound IP connectivity
  - **Data input and retrieval**
    for real data processing: ~0.2MB/s/core IN, ~0.1MB/s/core OUT

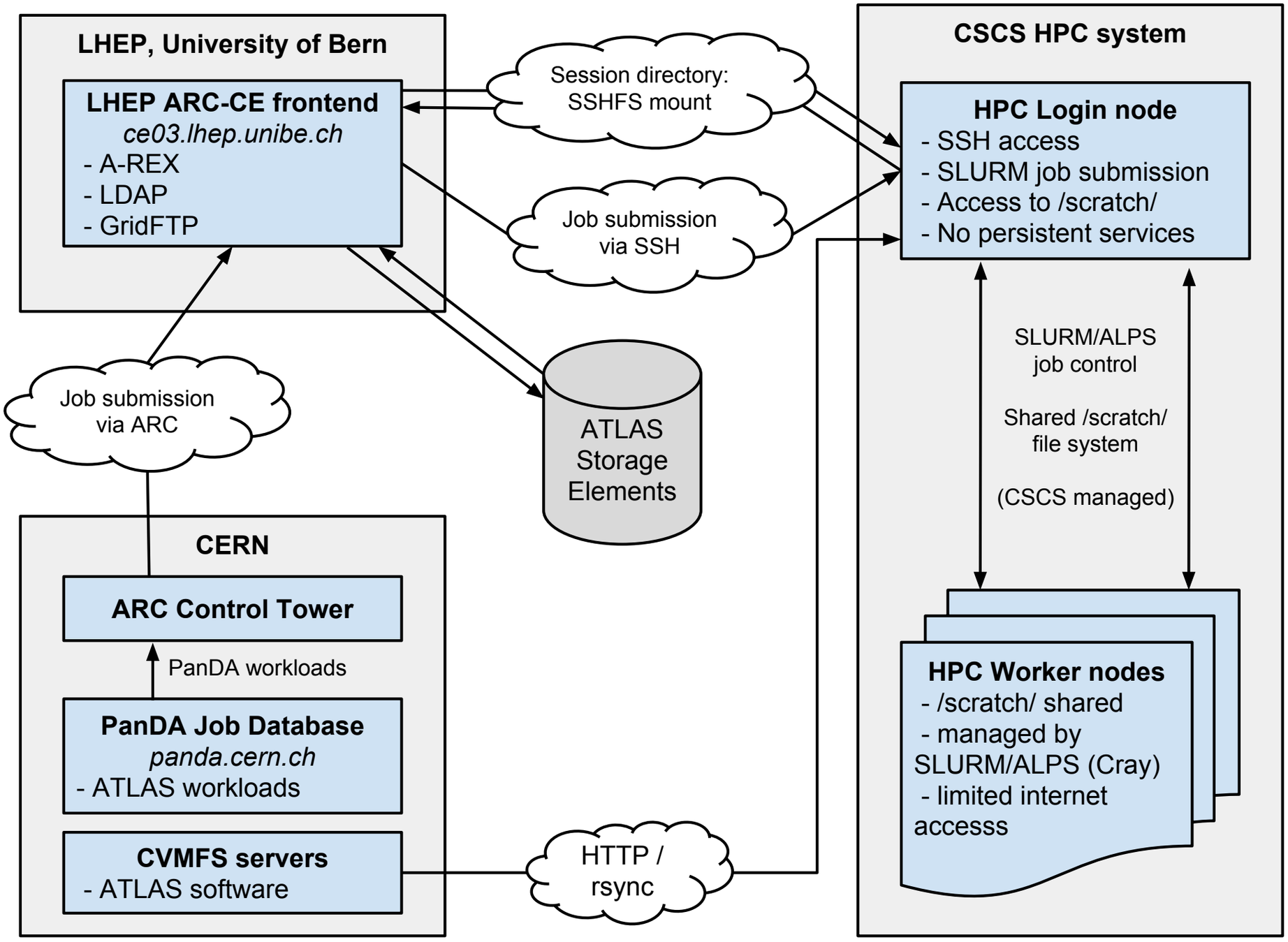
PAST EXPERIENCE

First approach to a Cray at CSCS, Lugano (2014/15)

System name	*Tödi*	*Piz Daint*	*Piz Dora*	*Monte Rosa*
Model	Cray XK7	Cray XC30	Cray XC40	Cray XE6
Description	Former CPU/GPU development and integration system.	Current flagship hybrid CPU/GPU system.	Flagship CPU-only system.	Former flagship CPU-only system.
Compute node configuration	• 16 core AMD Opteron CPU • 32 GB RAM • NVIDIA Tesla K20X GPU	• 8 core Intel Xeon CPU • 32 GB RAM • NVIDIA Tesla K20X GPU	• 2 x 12 core Intel Xeon CPUs • 64/128 GB RAM	• 2 x 16 core AMD Interlagos CPUs • 32 GB RAM
Number of compute nodes	272	5272	1256	1496
Total number of CPU cores	4352 + 272 GPUs	42176 + 5272 GPUs	30144	47872
Interconnect	Cray Gemini	Cray Aries	Cray Aries	Cray Gemini
Resource Manager / Scheduler	Cray SLURM / ALPS	Cray SLURM / ALPS	Cray SLURM / ALPS	Cray SLURM / ALPS

PAST EXPERIENCE

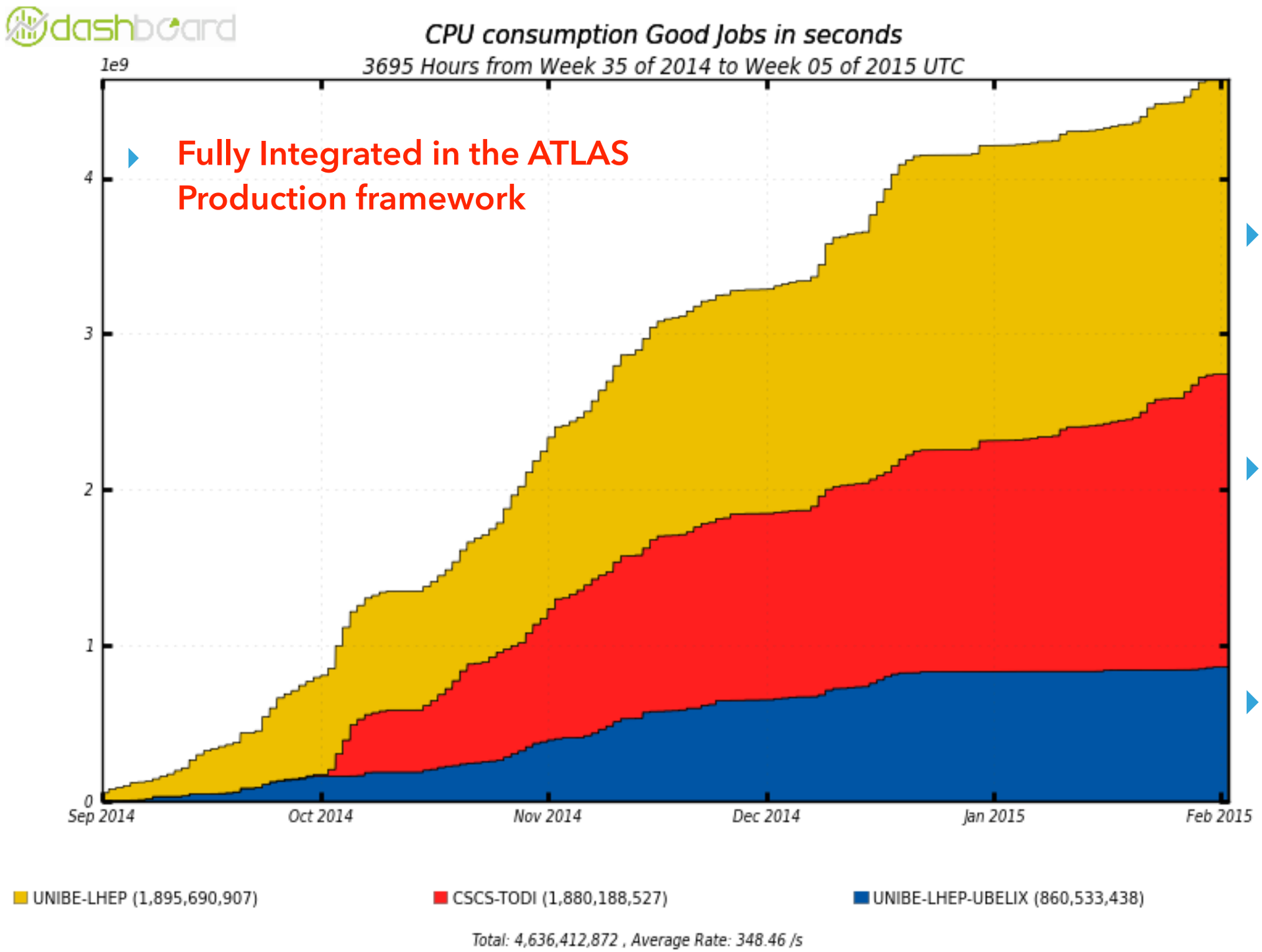
First approach to a Cray at CSCS, Lugano (2014/15)



- Master thesis work by Michael Hostettler, Universität Bern , 2015

PAST EXPERIENCE

First approach to a Cray at CSCS, Lugano (2014/15)



- Fully Integrated in the ATLAS Production framework
- Tier-2 at Bern
- Cray
- Tier-3 at Bern

PAST EXPERIENCE

First approach to a Cray at CSCS, Lugano (2014/15): lessons learned

General

- **The ARC Compute Element is well suited to targeting remote compute systems in a non-intrusive way**
  - Abiding to the strict access policies typical of a HPC centre
  - Does not need to be physically located inside the computing centre
  - Seamless integration of compute resources in the experiment workload management system
  - The data management system built in ARC allows for transparent delivery and retrieval of data
  - The modifications needed for this use case (e.g. SSH submission) will be included in a future release of ARC

Performance

- **We were able to run un-modified binaries out of the experiment sw repositories from CVMFS on the Cray :-)**
  - Pre-compiled gcc binaries (CVMFS) performed 30% better than binaries re-compiled with the Cray compiler
  - gcc + Cray recommended options brought only a marginal improvement (~5%) in processing time per event
- Linear job and processed event rate scaling from 10 to 100 compute nodes (as expected)
- Thread scaling is linear too (with a slight offset due to the initialisation and finalisation steps)
- Memory footprint for a ATLAS GEANT4 job running on 16 cores is considerably lower that the 32GB available

THE LHCONCRAY PROJECT: OVERVIEW

New approach to a Cray at CSCS, Lugano (2016)

- **Objective: be able to run all the Tier-2 Grid workloads on a share of general CSCS resources (Cray systems, central storage, etc)**
- **This includes:**
  - All supported WLCG VOs: ATLAS, CMS, LHCb
  - All experiment workloads (pilot, production, analysis, monitoring, etc)
  - Fully integrated with the experiment central factories (Crab, Panda, Dirac, etc)
  - Fully grid-aware (CE, SE, BDII, etc) and integrated into WLCG
  - "Green" for the VOs and the central EGI monitoring systems
- **This poses many challenges**
  - Difference in the OS and libraries
  - Supporting infrastructure such as CVMFS, scratch file system (10k files/job), network, etc.
  - Middleware readiness for ARC, Xrootd, GridFTP, HTTP, SRM, Infosys, Accounting, etc.
  - VO readiness (they might need to adapt as well!) <= **VERY IMPORTANT**
  - Administrative, contractual and support changes
- **If the project is successful and shows some cost advantage, the plan is to phase the current dedicated WLCG Tier-2 systems out and run all the WLCG workloads on the Central Systems at CSCS**

THE LHCONCRAY PROJECT: OVERVIEW

New approach to a Cray at CSCS, Lugano (2016)

The original Compute plan

Phase 1
Feasibility study

Objective: Evaluate whether it is actually possible to run WLCG jobs in a Cray system

1. **Enabling job submission from the grid**
   - ✓ a. Adapt the infrastructure
   - ✓ b. Adapt CE software
2. **Software validation**
   - ✓ a. Base grid software
   - ✓ b. Application software

✓

Phase 2
Integration

Objective: Attempt to integrate WLCG jobs on CSCS infrastructure.

3. **Scheduling & Accounting**
   - ✓ a. Multiple jobs per node
   - b. Fairshare complexity

✓

Phase 3
Evaluation at scale

Objective: Evaluate and control impact of WLCG jobs on CSCS infrastructure.

4. **Scalability**
   - a. High rate of job submission/sec
   - b. High impact on I/O systems

THE LHCONCRAY PROJECT: OVERVIEW

New approach to a Cray at CSCS, Lugano (2016)

The original Compute plan

Phase 1
Feasibility study

Objective: Evaluate whether it is actually possible to run WLCG jobs in a Cray system

1. **Enabling job submission from the grid**
   - ✓ a. Adapt the infrastructure
   - ✓ b. Adapt CE software
2. **Software validation**
   - ✓ a. Base grid software

✓

Phase 2
Integration

Objective: Attempt to integrate WLCG jobs on CSCS infrastructure.

Main technical challenges and objectives:

- From proof-of-concept to pre-production
- Sharing nodes between different jobs/users
- Meet different scheduling requirements (HPC vs. HTC)
- Running unmodified VO applications

Phase 3
Evaluation at scale

Objective: Evaluate and control impact of WLCG jobs on CSCS infrastructure.

4. **Scalability**
   - a. High rate of job submission/sec
   - b. High impact on I/O systems

SYSTEM ARCHITECTURE AND INTEGRATION

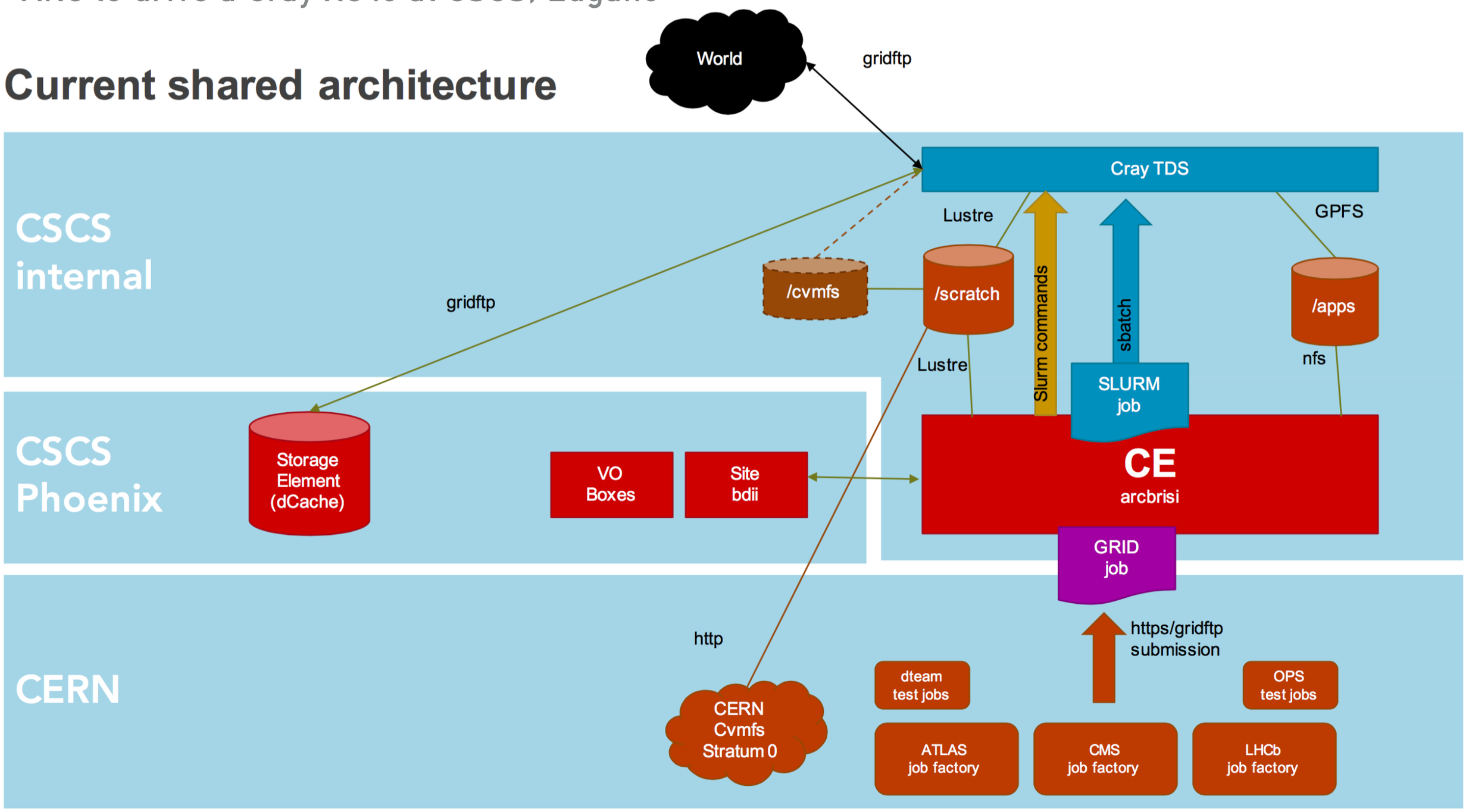
Brisi: Test Design System for Piz Dora

System name	*Tödi*	*Piz Daint*	*Piz Dora*	*Monte Rosa*
Model	Cray XK7	Cray XC30	Cray XC40	Cray XE6
Description	Former CPU/GPU development and integration system.	Current flagship hybrid CPU/GPU system.	Flagship CPU-only system.	Former flagship CPU-only system.
Compute node configuration	• 16 core AMD Opteron CPU • 32 GB RAM • NVIDIA Tesla K20X GPU	• 8 core Intel Xeon CPU • 32 GB RAM • NVIDIA Tesla K20X GPU	• 2 x 12 core Intel Xeon CPUs • 64/128 GB RAM	• 2 x 16 core AMD Interlagos CPUs • 32 GB RAM
Number of compute nodes	272	5272	1256	1496
Total number of CPU cores	4352 + 272 GPUs	42176 + 5272 GPUs	30144	47872
Interconnect	Cray Gemini	Cray Aries	Cray Aries	Cray Gemini
Resource Manager / Scheduler	Cray SLURM / ALPS	Cray SLURM / ALPS	Cray SLURM / ALPS	Cray SLURM / ALPS

- Native SLURM 16.05 on Brisi

SYSTEM ARCHITECTURE AND INTEGRATION

ARC to drive a Cray XC40 at CSCS, Lugano



SYSTEM ARCHITECTURE AND INTEGRATION

LHConCRAY: Test integration phase ongoing with the ARC Compute Element

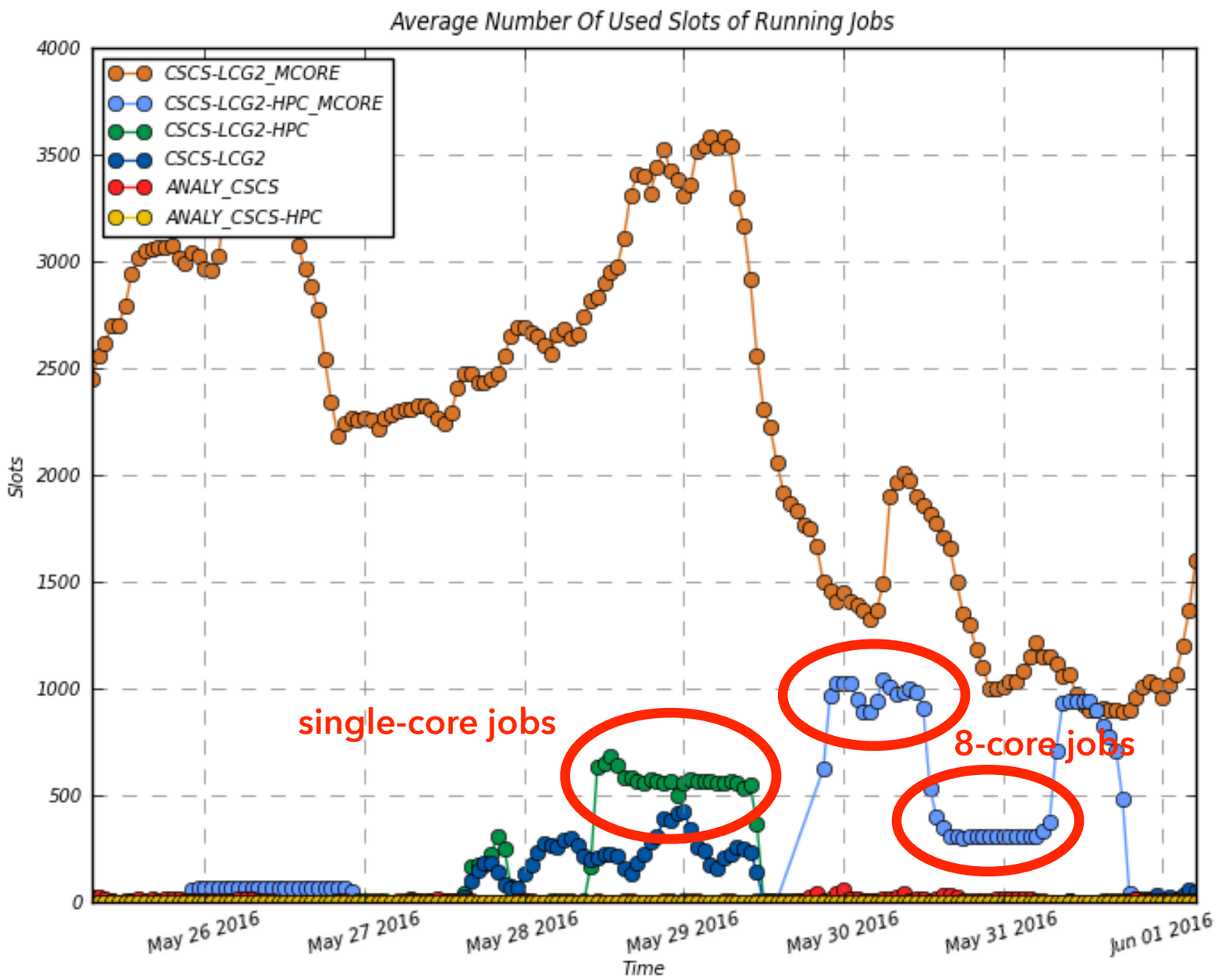
- Dedicated **ARC CE** in place (arcbrisi.cscs.ch): OPS, ATLAS, CMS and LHCb, submits directly to the LRMS.
- **Application provisioning**: Initial approach was to preload the **CernVM-FS** cache for all VOs on the shared file-system
- Compute: **Cray Test Design System XC40** (**Brisi**)
  - 1 SLURM partition (wlcg) for all VOs, each node may be shared by concurrent jobs.
  - **Broadwell** nodes with 64 HT-cores (Intel Xeon E5-2695 v4 @2.10GHz) and 128 RAM each.
  - **HEPSPEC'06 rating** measured to be **13.39**/core. This is 18% higher when compared to the average rating of **11.46** for the current Tier-2 cluster (Phoenix)
  - Jobs run within **Shifter containers** [1]. The container itself is a CentOS 6.8 full image with the same packages as in the dedicated T2 cluster (Phoenix) and configured accordingly.
    - It integrates seamlessly in the Cray environment. Can also run on any cluster
    - Leverages from current Docker layered image based environment
    - Security is enforced by design: all user processes run in userland
- Integrated in the ATLAS PanDA workload management: `CSCS-LCG2-HPC, CSCS-LCG2-HPC_MCORE, ANALY_CSCS-HPC`
- Also integrated in the CMS Crab factories and the LHCb Dirac factories

[1] http://www.nersc.gov/research-and-development/user-defined-images

FIRST PRELIMINARY PERFORMANCE DATA

LHConCRAY - Performance with ATLAS HammerCloud stress tests

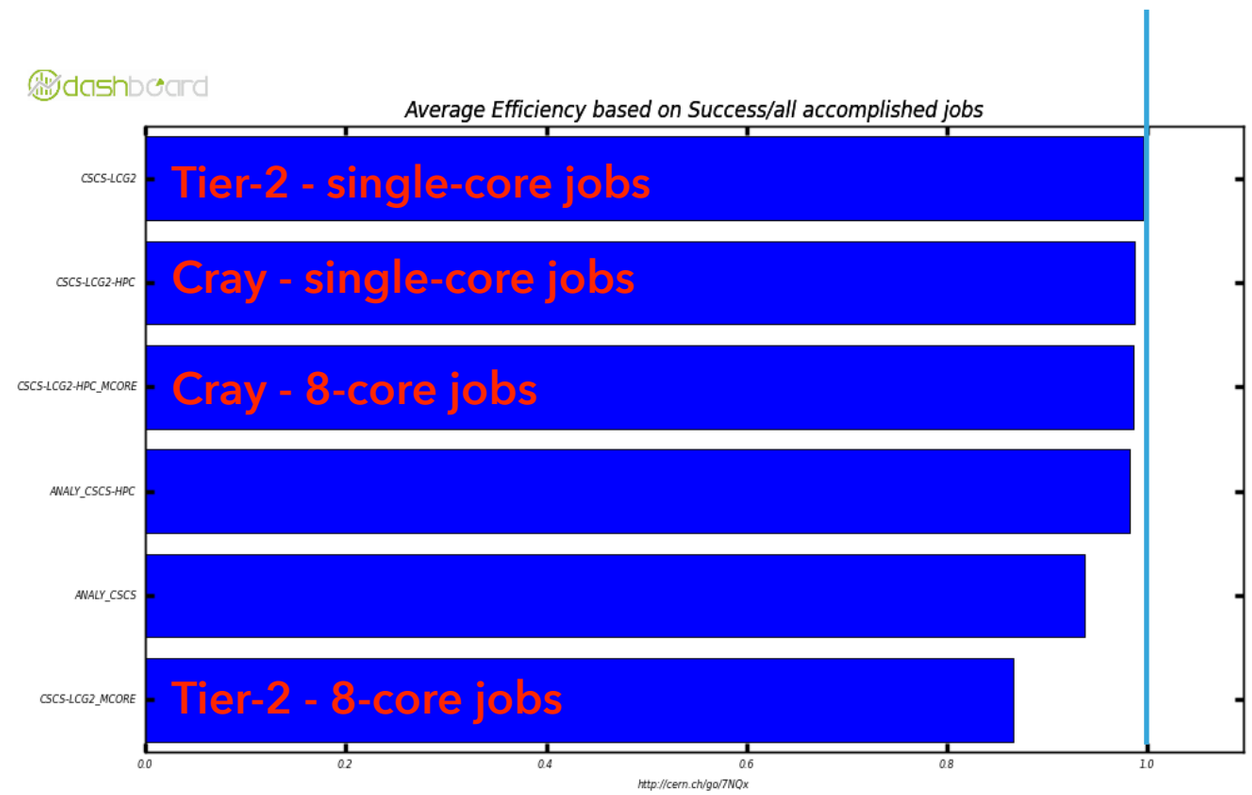
- Using the ATLAS HammerCloud testing framework to study performance for different workloads
- Allows for reproducible tests (same input data, 1k events for single-core and 8k events for 8-core jobs)



FIRST PRELIMINARY PERFORMANCE DATA

LHConCRAY - Performance with ATLAS HammerCloud stress tests: Job success rate

- CPU-bound workload: ATLAS detector simulation: **Job success rate**



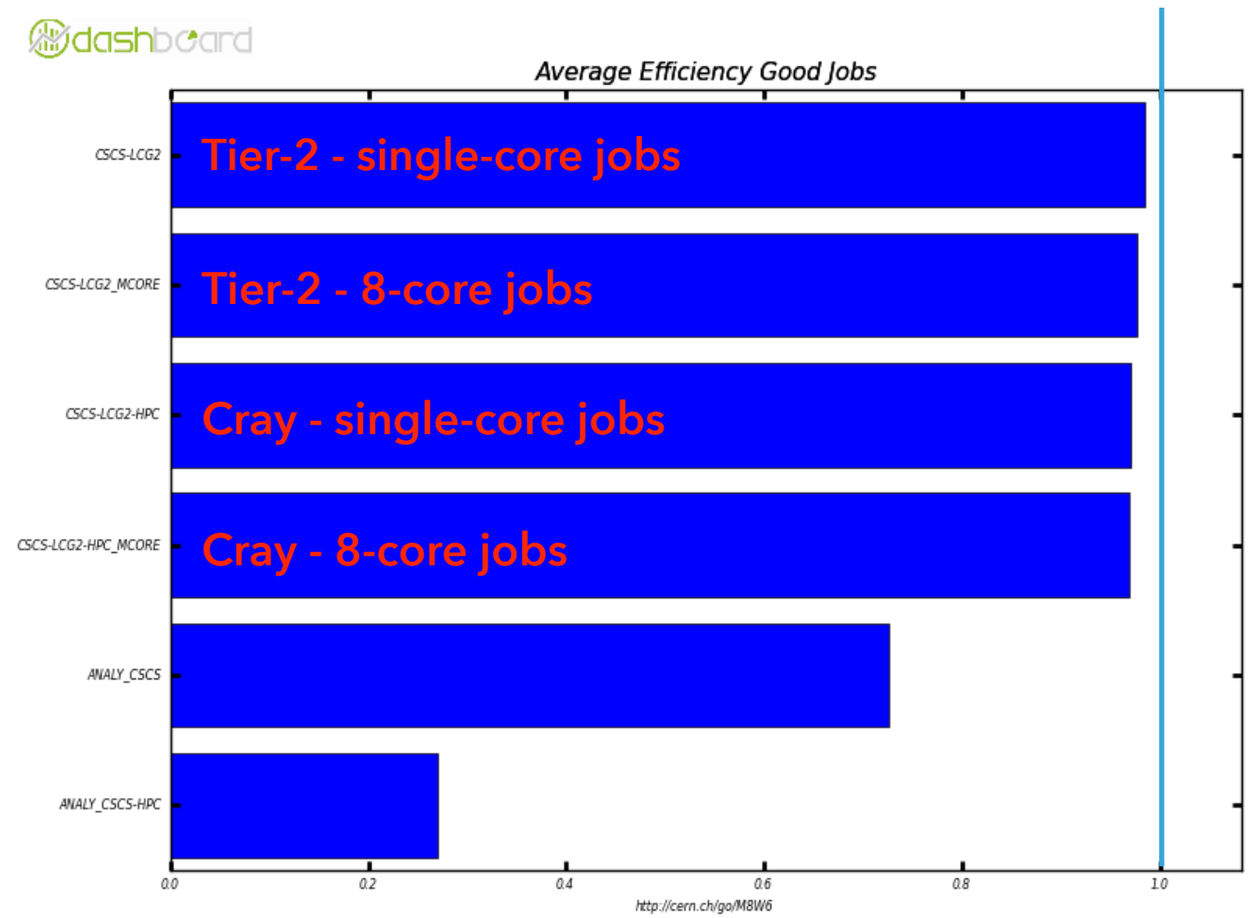
Job success Efficiency ——————————> 1.0

Time ——————————————————————>

FIRST PRELIMINARY PERFORMANCE DATA

LHConCRAY - Performance with ATLAS HammerCloud stress tests: CPU efficiency

- CPU-bound workload: ATLAS detector simulation: **CPU efficiency**



Time ——————————————————>

CPU/WT Efficiency ——————————> 1.0

FIRST PRELIMINARY PERFORMANCE DATA

LHConCRAY - Performance with ATLAS HammerCloud stress tests: WallClock

- CPU-bound workload: ATLAS detector simulation: **WallClock**

	Successful jobs	Failed jobs	HEPSPEC06 core	Mean wallclock
Cray single core	4316	56	13.39	10386
Phoenix T2 single core	2117	0	11.46	13450
Cray 8-core	340	12	13.39	17978
Phoenix T2 8-core	75	0	11.46	23029

Cray HEPSPEC rating is 18% better than the Phoenix tier-2 nodes

Cray Wallclock performance is 22 to 25% better than the Phoenix tier-2 nodes

OUTLOOK

LHConCRAY - System upgrades

During August, the Main Crays and the TDS (Brisi) have been heavily upgraded

- **CPU:**
  - **Broadwell** nodes with 64 HT-cores (Intel Xeon E5-2695 v4 @2.10GHz) and 128 RAM each.

- **CLE6.0, a new environment:**
  - **Cray Linux Environment 6.0 is based on SUSE 12**
    (CLE 5.2 UP04 was based on SUSE 11 SP3)
  - **Compute nodes run full OS**
    netroot or tmpfs
  - **Cray proprietary configuration** management tool (xtopview) **is replaced by** more or less standard **Ansible**
  - Nodes are eligible for **live package updates**
    zypper repos are accessible and can be used with system booted

OUTLOOK

LHConCRAY - System upgrades

During August, the Main Crays and the TDS (Brisi) have been heavily upgraded

- **Network changes**
  - Nodes inside the Cray High Speed Network now use **public IPs** with standard Linux IP packet forwarding with Pseudo-HA (was Realm Specific IP)
  - One compute node can get a max. of **10Gbps connection** (40Gbps in the future) iperf3 shows a sustained transfer rate of ~5Gbps to/from a compute node
  - **Active GridFTP/dcap transfers** have been successfully done to/from Phoenix's dCache
  - **CVMFS is much faster** in caching data
  - **The ARC CE has direct access to the compute nodes**, no more wrappers!

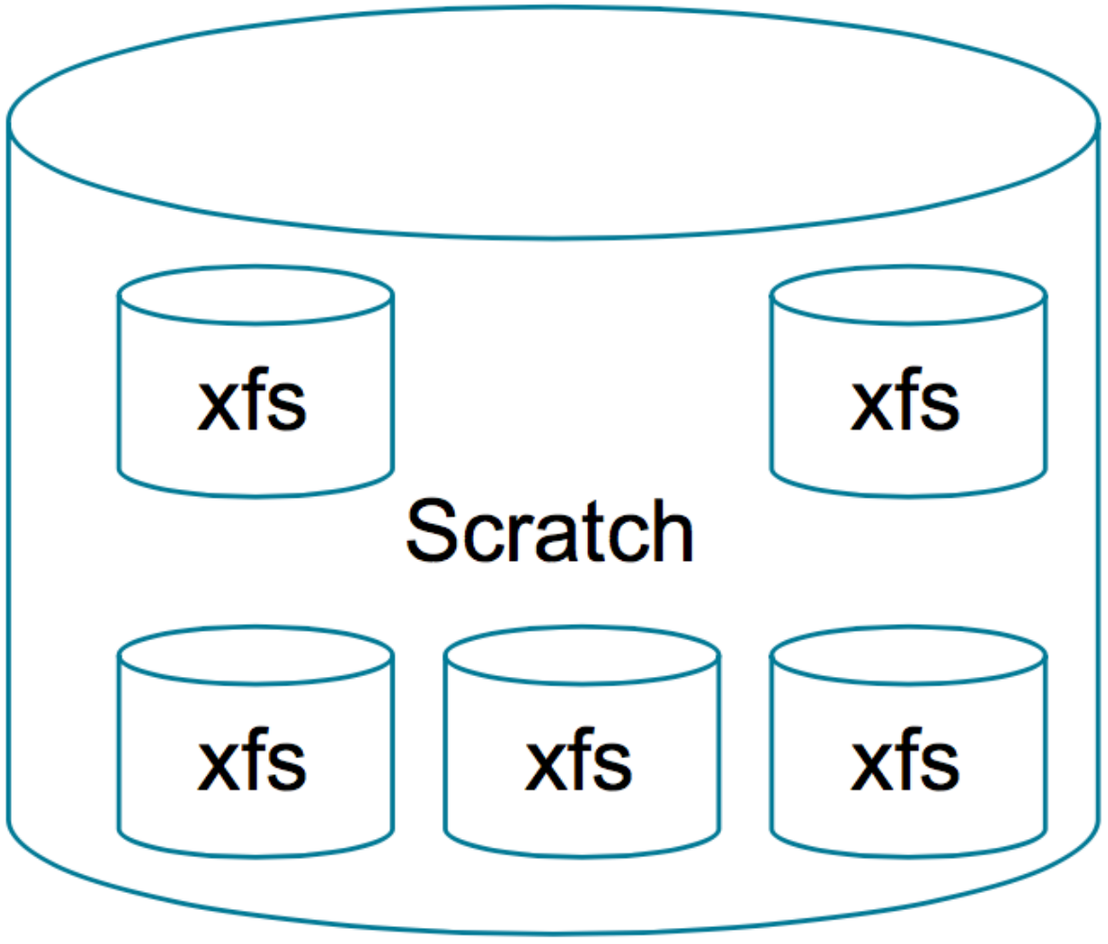
- **CVMFS changes**
  - No longer using a shared, preloaded cache
  - Each node has **its own XFS filesystem** for caching CVMFS data
  - The XFS filesystem is, in reality, **a sparse XFS file** that can be located *anywhere*
  - The underlying filesystem sees only one file: **minimal metadata operations**

OUTLOOK

LHConCRAY - System upgrades

During August, the Main Crays and the TDS (Brisi) have been heavily upgraded

- **Network changes**
  - Nodes inside the Cray High Speed Network now use **public IPs** with standard Linux IP packet forwarding with Pseudo-HA (was RSIP)
  - One compute node can get a max. of **10Gbps connection** (40Gbps in the future) iperf3 shows a sustained transfer rate
  - **Active GridFTP/dcap transfers** have b
  - **CVMFS is much faster** in caching data
  - **The ARC CE has direct access to the c**
- **CVMFS changes**
  - No longer using a shared, preloaded
  - Each node has **its own XFS filesystem**
  - The XFS filesystem is, in reality, **a spar**
  - The underlying filesystem sees only o



OUTLOOK

LHConCRAY - Next steps: Resume testing, benchmark and scale-up

- **Integrate more workflows** (e.g. I/O intensive, incl. user analysis jobs) while **testing the scalability**
- Perform **benchmarking** with more workloads to evaluate performance under different data access patterns
- **Ramp-up test**: from zero to x*k cores, ramp-up time (**ARC+shared File System performance**)
- Will need **tests beyond 1k-core scale** (TDS is limited in size)
- Evaluate and control the **impact** of WLCG jobs on the CSCS infrastructure

- Get more "real" jobs to run : -)
- **Slurm** fair-share and accounting **tuning** (partially conflicting VO ecosystems, HPC vs HTC, etc.)
  - diskless nodes and up to 2GB RAM/job slot
  - additional challenge: different mode of operation/computing models of the 3 experiments

- **Feed some of the information from above back to the cost study**

CONCLUSIONS

- We are trying to address some of the challenges of LHC computing for beyond the next 5-10 years
- The use of general purpose HPC systems is one of the possible ways forward

- The ARC technology provides us with the right tool
  - lightweight and non-invasive access to high-end computing resources
  - should contribute to bring down (considerably?) the operational costs

- Quite some experience gained doing real computations for ATLAS (with minimal support required)

- We have setup the LHConCRAY project to explore the feasibility and cost advantages of this model
- The main challenges arising from this approach have been addressed

CONCLUSIONS

- We are trying to address some of the challenges of LHC computing for beyond the next 5-10 years
- The use of general purpose HPC systems is one of the possible ways forward

- The ARC technology provides us with the right tool
  - lightweight and non-invasive access to high-end computing resources
  - should contribute to bring down (considerably?) the operational costs

- Quite some experience gained doing real computations for ATLAS (with minimal support required)

- We have setup the LHConCRAY project to explore the feasibility and cost advantages of this model
- The main challenges arising from this approach have been addressed:

  - Processor architecture and/or OS: **run applications from within Shifter containers**
  - Compliance with tight access rules: **access policies relaxed (project endorsed by the RC)**
  - Application provisioning: **CVMFS, with per node-sparse XFS FS for the cache**
  - Workload management integration: **leverage the ARC CE technology**
  - Data input and retrieval: **leverage the ARC CE technology**

CONCLUSIONS

- We are trying to address some of the challenges of LHC computing for beyond the next 5-10 years
- The use of general purpose HPC systems is one of the possible ways forward

- The ARC technology provides us with the right tool
  - lightweight and non-invasive access to high-end computing resources
  - should contribute to bring down (considerably?) the operational costs

- Quite some experience gained doing real computations for ATLAS (with minimal support required)

- We have setup the LHConCRAY project to explore the feasibility and cost advantages of this model
- The main challenges arising from this approach have been addressed:

- Integration is progressing well and test at scale are planned for the immediate future
- Test results are expected to provide input to the cost study

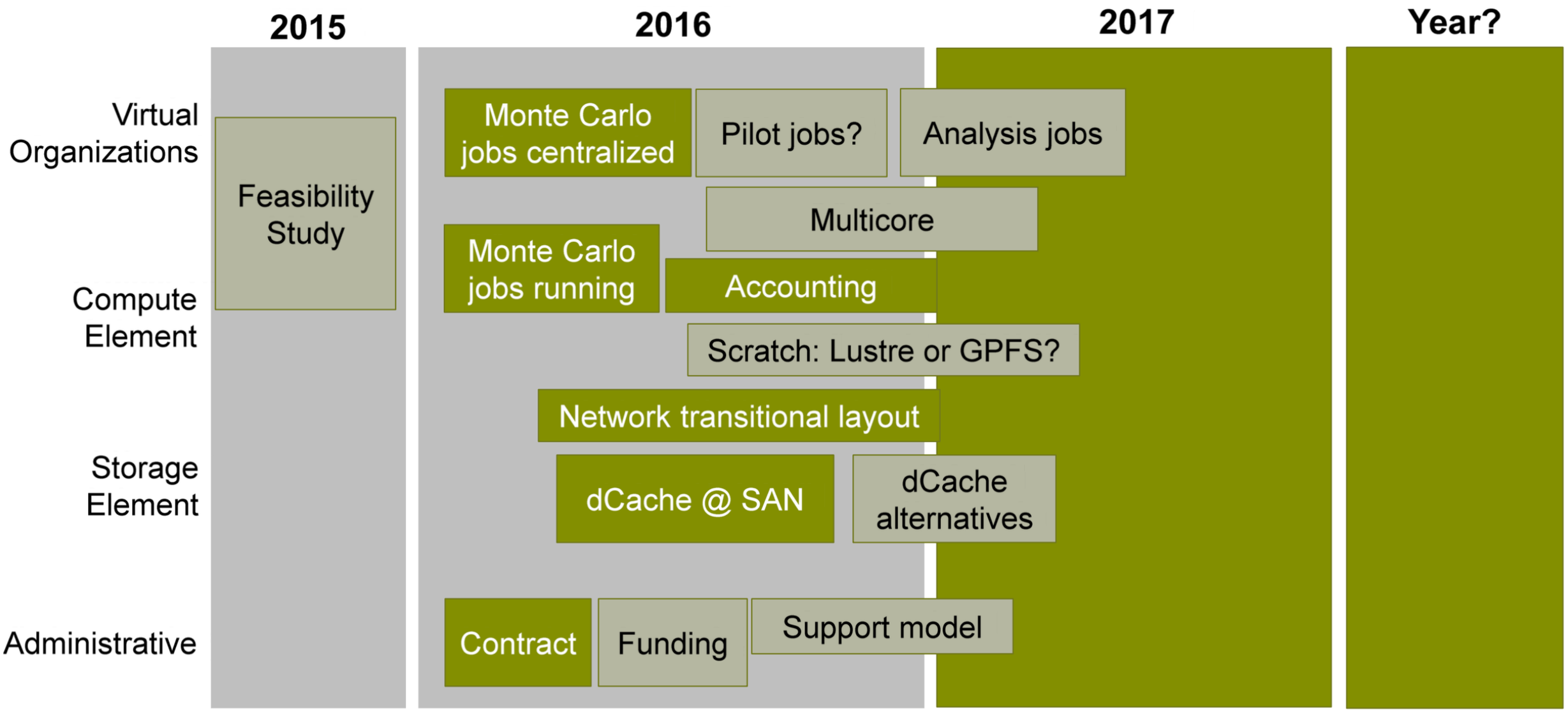
A decision on the way forward will be taken in a timescale of a few months

Thank you for your attention!

BACKUP

THE LHCONCRAY PROJECT: OVERVIEW

New approach to a Cray at CSCS, Lugano (2016)



BACKUP

Phase 2: Running unmodified applications with Shifter

- Shifter basically
  1. Pulls an image to a shared location (/scratch)
  2. Creates a loop device with the image (=container)
  3. Creates a chrooted environment on the loop device
  4. Runs our application in chrooted environment

- A container in our context is basically an image with a full CentOS distribution and a chroot

```
[miguelgi@brisi01]-[02:46:29]-[~]:-)$ salloc -t 01:00:00 -n1 --image=docker:centos:6.7 -N1
salloc: Granted job allocation 82463

[miguelgi@brisi01]-[02:57:20]-[~]:-)$ srun --pty shifter /bin/bash

[miguelgi@nid00035]-[01:57:30]-[~]:-)$ uname -r
3.0.101-0.46.1_1.0502.8871-cray_ari_c

[miguelgi@nid00035]-[01:57:31]-[~]:-)$ cat /etc/redhat-release
CentOS release 6.7 (Final)
```