- Motivation and challenges
- Non-invasive approach
- The LHConCRAY project
- System architecture and integration
- First preliminary performance data
- Summary and Outlook

ATLAS AND LHC COMPUTING ON CRAY

SWISS EXPERIENCE

Gianfranco Sciacca

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

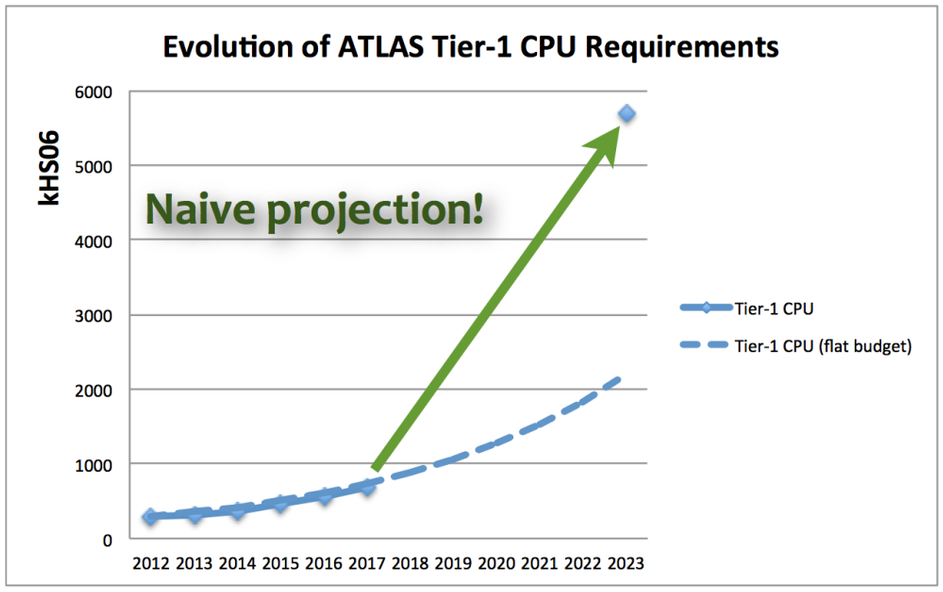
on behalf of the ATLAS Collaboration

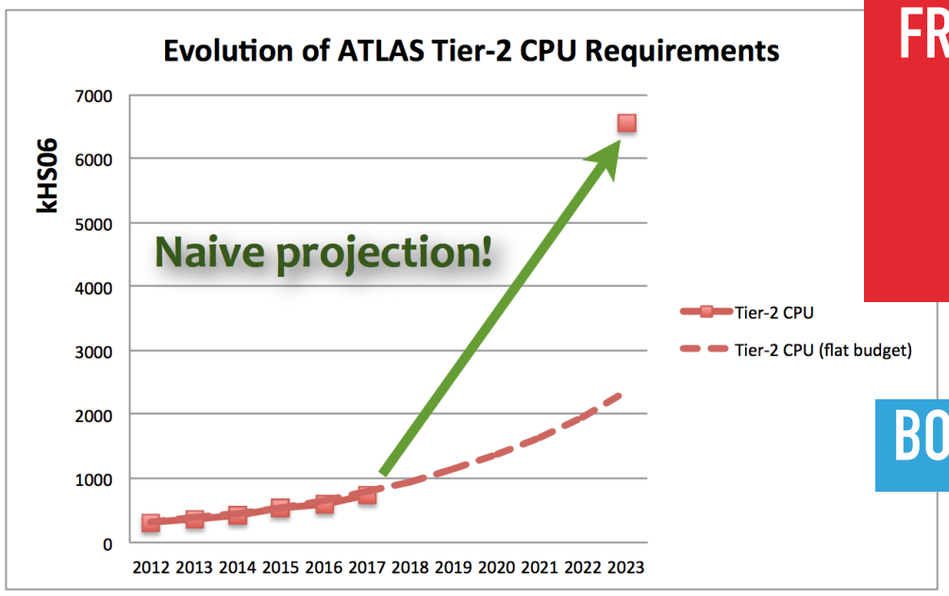
CHEP 2016 - 10-14 October 2016, San Francisco Marriott Marquis

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





EXPERIMENT REQUIREMENTS: CPU: A DRASTIC DEVIATION FROM WHAT 'FLAT-BUDGET' CAN AFFORD US

SAME FOR DISK

BORUT KERSEVAN, JUNE '16

- Part of the solution is to consolidate LHC computing
  - Less but bigger sites world wide (operationally cheaper, high-end hw, etc)
  - **General purpose HPC centres seem a good alternative to dedicated clusters**
  - Lightweight operational model, easier to focus the effort from the experiments
  - Leverage from economy of scales when procuring hardware (***true***? hopefully)

CHALLENGES

The challenges of Large Hadron Collider computing for the next decade

- **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

HPCS ARE VERY RESTRICTED AND SELF-CONTAINED ENVIRONMENTS!

CHALLENGES

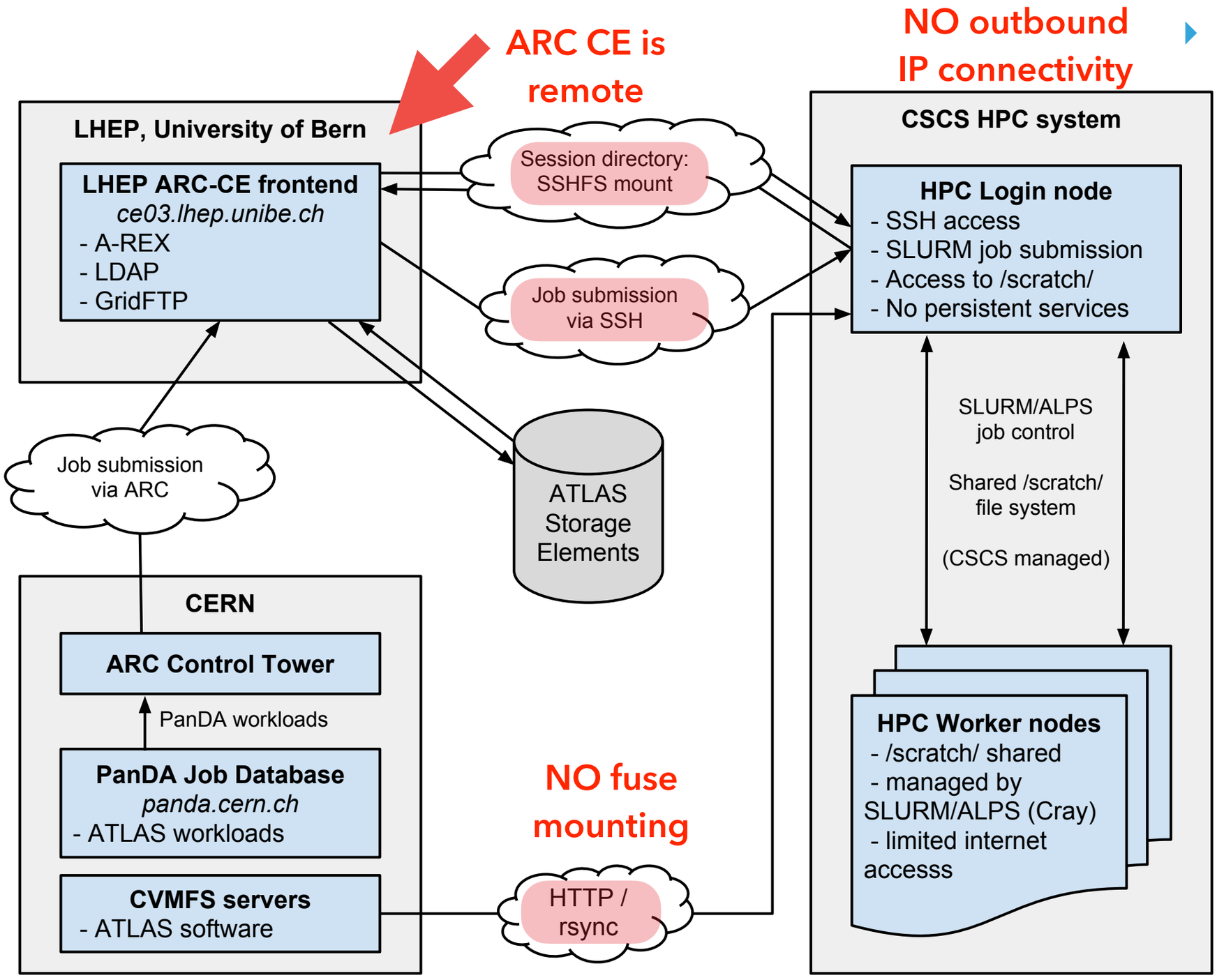
We have tried two approaches to the HPC centre

- **The "non-invasive" approach**
  - Obtain one user ssh access (relatively simple…, or relatively complex)
  - Perform the full lifecycle of the workloads from **OUTSIDE** the centre
  - Integrate with the experiment frameworks

- **The LHConCRAY project**
  - Gained endorsement by the centre (complex)
  - Can integrate more tightly **INSIDE** the centre
  - Solves several issues at once
  - Can now concentrate on performance and scalability

NON INVASIVE APPROACH

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



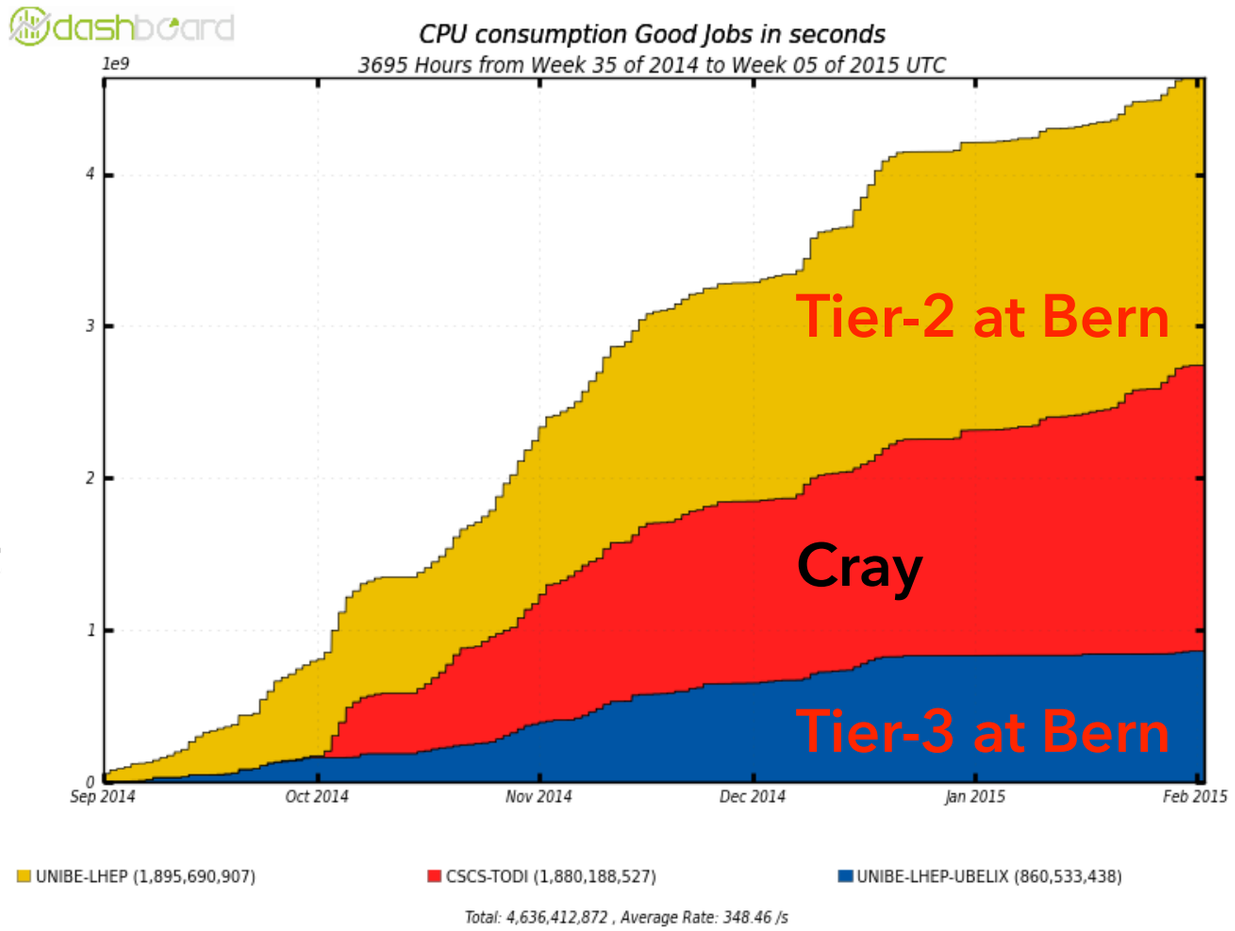
- **CRAY XK7 (Tödi)**
  - Former CPU/GPU development system
  - 16-core AMD Opteron CPU, 32GB RAM
  - CLE5
  - NVIDIA Tesla K20X GPU
  - 4352 cores + 272 GPUs
  - Cray Gemini interconnect
  - SLURM / ALPS
  - Same system as Titan

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

NON INVASIVE APPROACH

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

- **The versatility of the ARC CE [1] has been the key to "non-invasive" access to the HPC centre**
  - Addresses all the typical restrictions/policies of a HPC centre
  - Allows remote, compliant access, data management and job lifecycle management
  - Integrates seamlessly in the experiment frameworks
  - In production at centres in Germany, Cina (HPC) and Switzerland (cloud)
- **We were able to run un-modified binaries out of CVMFS on the Cray :-)**
  - Pre-compiled gcc binaries (CVMFS) performed 30% better than binaries re-compiled with the Cray compiler
  - gcc + Cray recommended options: only a marginal improvement (~5%) in processing time per event
  - Node and thread scaling: linear as expected

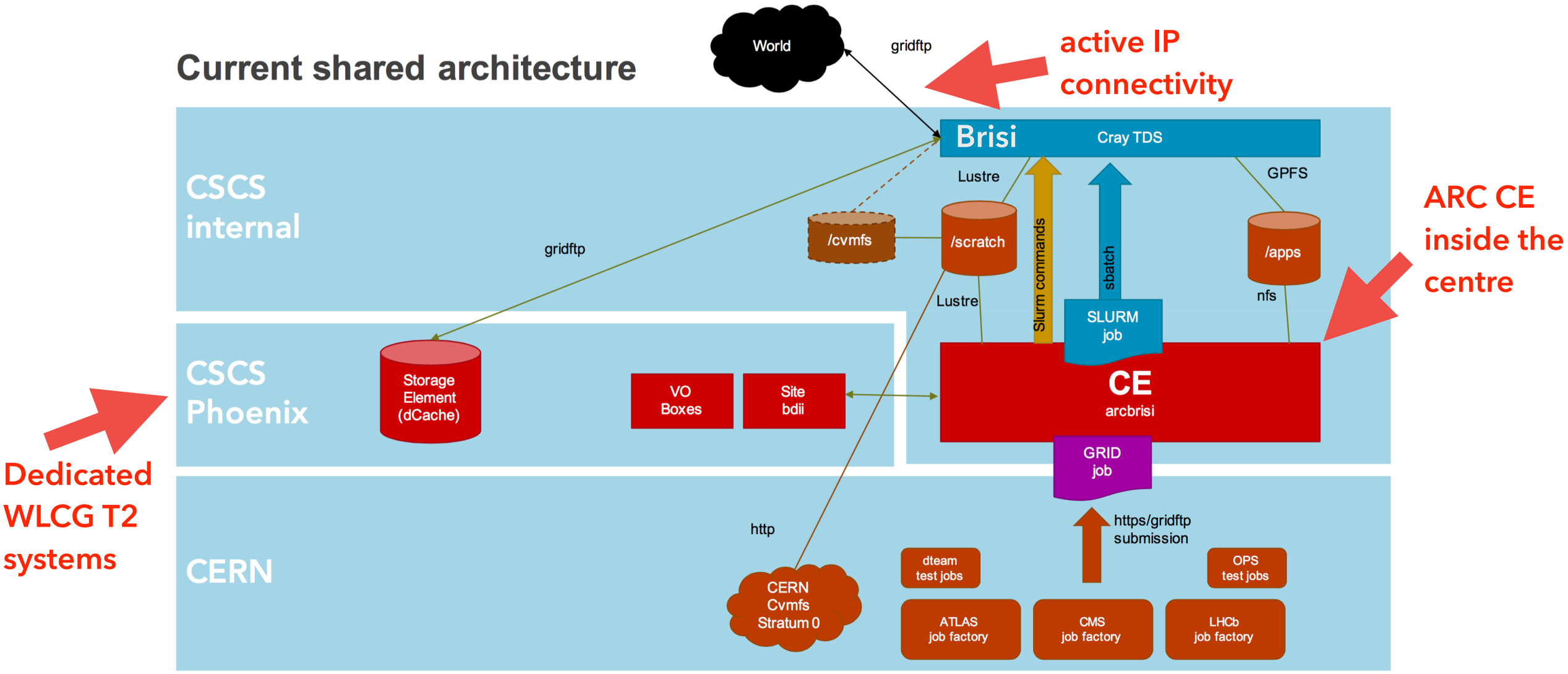


- [1] http://www.nordugrid.org/arc/

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) - Fully WLCG and EGI integrated



- If the project is successful and shows some cost advantage, the plan is to phase out the current dedicated WLCG Tier-2 systems at CSCS 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)

- **CRAY XC40 (Piz Dora)** - we have access to the Test Development System **(Brisi)**
  - **Broadwell** (Intel Xeon E5-2695 v4 @2.10GHz)
  - **64 HT-cores, 128GB RAM**
  - **HEPSPEC06 rating: 13.4/core**
  - **Diskless** nodes
  - **CLE6.0** (based on SUSE 12)
  - **Cray Aries** interconnect
  - **Native SLURM** 16.05 (**nodes may be shared**)

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.

From proof-of-concept to pre-production

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

SYSTEM ARCHITECTURE AND INTEGRATION

LHConCRAY: Test integration phase ongoing with the ARC Compute Element

✓ **Processor architecture and/or OS might not always be suitable**

jobs run within **Shifter containers** [1]. The container itself is a CentOS 6.8 full image with the same packages as in the dedicated WLCG T2 cluster (Phoenix) and configured accordingly.

✓ **Compliance with tight access rules**

access policies relaxed (project endorsed by the RC). Middleware **INSIDE** the centre

✓ **Application provisioning**

CVMFS, with one XFS FS (one single sparse XFS file) **for the cache per node**

✓ **Workload management integration**

leverage the ARC CE technology. Integrated seamlessly with the ATLAS, CMS and LHCb factories. Nodes inside the Cray High Speed Network now use **public IPs** with standard Linux IP packet forwarding

✓ **Data input and retrieval**

leverage the ARC CE technology. Will need testing at scale

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

FIRST PRELIMINARY PERFORMANCE DATA

LHConCRAY - Performance with ATLAS HammerCloud stress tests

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

	Stress test duration h	Successful jobs	Failed jobs	Success rate %	CPU/ wallclock	HEPSPEC 06 (core)	Mean wallclock
Cray single core	24	4316	56	98.8	0.9708	13.39	10386
Phoenix T2 single core	24	2117	0	100	0.9889	11.46	13450
Cray 8-core	48	340	12	96.6	0.9682	13.39	17978
Phoenix T2 8-core	48	75	0	100	0.9777	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

SUMMARY AND OUTLOOK

- We think that the use of general purpose HPC systems is one possible way forward in addressing some of the challenges of LHC computing for beyond the next 5-10 years

- We have identified the the ARC middleware technology as the right tool
  - provides lightweight and non-invasive access to high-end computing resources
  - ensures seamless integration with the experiment frameworks

- We have demonstrated "remote access" of a Cray at CSCS doing real computations for ATLAS
- We have setup the LHConCRAY project to integrate all WLCG workloads on a Cray at CSCS

  to explore the feasibility at a scale and cost advantages of this model, and have addressed the main challenges arising from this approach

- 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!

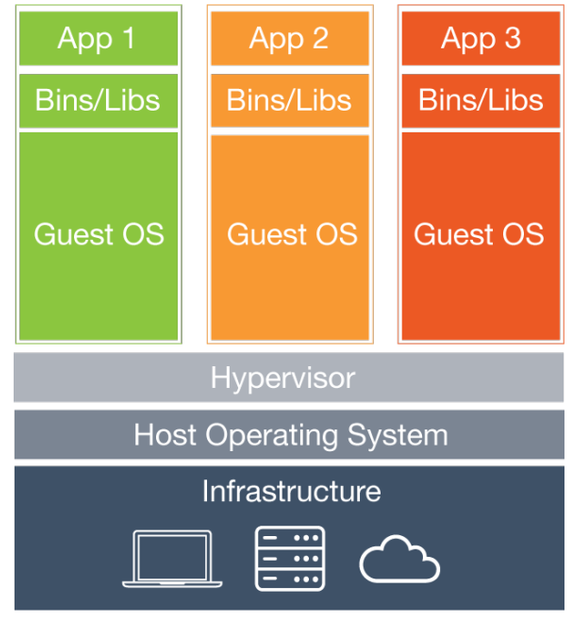
BACKUP

Phase 2: Running unmodified applications with Shifter

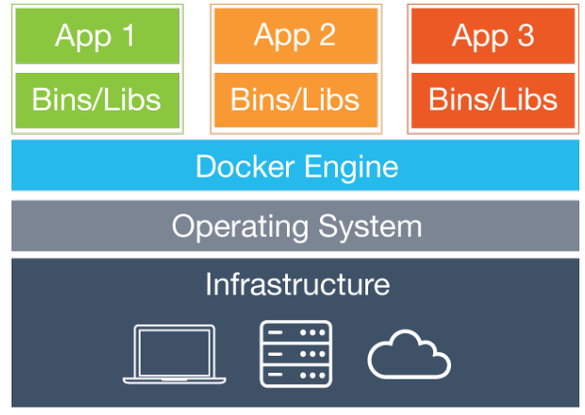
- Shifter basically
  1. Pulls an image to a shared location (/scratch)
  2. Creates a loop device with the image (=container)
  3. Creates a chrooted environment on the loop device
  4. Runs our application in chrooted environment

- 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)
```



Virtual Machines



Containers