Big data science accessing high-end HPC

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

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

**Evolution of ATLAS Tier-1 CPU Requirements**

- Naive projection!

**Evolution of ATLAS Tier-2 CPU Requirements**

- Naive projection!
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’**

*Evolution of ATLAS Tier-1 Disk Requirements*

*Evolution of ATLAS Tier-2 Disk Requirements*
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)
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)

<table>
<thead>
<tr>
<th>System name</th>
<th>Tödi</th>
<th>Piz Daint</th>
<th>Piz Dora</th>
<th>Monte Rosa</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Model</strong></td>
<td>Cray XK7</td>
<td>Cray XC30</td>
<td>Cray XC40</td>
<td>Cray XE6</td>
</tr>
<tr>
<td><strong>Description</strong></td>
<td>Former CPU/GPU development and integration system.</td>
<td>Current flagship hybrid CPU/GPU system.</td>
<td>Flagship CPU-only system.</td>
<td>Former flagship CPU-only system.</td>
</tr>
<tr>
<td></td>
<td>● 16 core AMD Opteron CPU</td>
<td>● 8 core Intel Xeon CPU</td>
<td>● 2 x 12 core Intel Xeon CPUs</td>
<td>● 2 x 16 core AMD Interlagos CPUs</td>
</tr>
<tr>
<td></td>
<td>● 32 GB RAM</td>
<td>● 32 GB RAM</td>
<td>● NVIDIA Tesla K20X GPU</td>
<td>● NVIDIA Tesla K20X GPU</td>
</tr>
<tr>
<td><strong>Compute node configuration</strong></td>
<td></td>
<td>● 64/128 GB RAM</td>
<td></td>
<td>● 32 GB RAM</td>
</tr>
<tr>
<td><strong>Number of compute nodes</strong></td>
<td>272</td>
<td>5272</td>
<td>1256</td>
<td>1496</td>
</tr>
<tr>
<td><strong>Total number of CPU cores</strong></td>
<td>4352 + 272 GPUs</td>
<td>42176 + 5272 GPUs</td>
<td>30144</td>
<td>47872</td>
</tr>
<tr>
<td><strong>Interconnect</strong></td>
<td>Cray Gemini</td>
<td>Cray Aries</td>
<td>Cray Aries</td>
<td>Cray Gemini</td>
</tr>
<tr>
<td><strong>Resource Manager / Scheduler</strong></td>
<td>Cray SLURM / ALPS</td>
<td>Cray SLURM / ALPS</td>
<td>Cray SLURM / ALPS</td>
<td>Cray SLURM / ALPS</td>
</tr>
</tbody>
</table>
First approach to a Cray at CSCS, Lugano (2014/15)

- **LHEP, University of Bern**
  - LHEP ARC-CE frontend
    - `ce03.lhep.unibe.ch`
    - A-REX
    - LDAP
    - GridFTP
  - Job submission via ARC

- **CERN**
  - ARC Control Tower
    - PanDA workloads
  - PanDA Job Database
    - `panda.cern.ch`
    - ATLAS workloads
  - CVMFS servers
    - ATLAS software

- **CSCS HPC system**
  - HPC Login node
    - SSH access
    - SLURM job submission
    - Access to /scratch/
    - No persistent services
  - HPC Worker nodes
    - /scratch/ shared
    - managed by SLURM/ALPS (Cray)
    - limited internet access
  - ATLAS Storage Elements
    - Shared /scratch/ file system
      - (CSCS managed)

- **Session directory**: SSHFS mount
- **Job submission**: via SSH

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

CPU consumption Good jobs in seconds
3695 Hours from Week 35 of 2014 to Week 05 of 2015 UTC

Total: 4,636,412,872, Average Rate: 348.46 /s
**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 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

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

4. **Scalability**  
   a. High rate of job submission/sec  
   b. High impact on I/O systems
The original Compute plan

1. **Enabling job submission from the grid**
   - a. Adapt the infrastructure
   - b. Adapt CE software

2. **Software validation**
   - a. Base grid software

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

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

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**

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

- **Native SLURM 16.05 on Brisi**
SYSTEM ARCHITECTURE AND INTEGRATION

ARC to drive a Cray XC40 at CSCS, Lugano

Current shared architecture

CSCS internal

CSCS Phoenix

CERN

Storage Element (dCache)

VO Boxes

Site bdii

http

cern Cvmfs
Stratum 0

Atlas test jobs

CMS job factory

LHCb job factory

https/gridftp submission

gridftp

World

Cray TDS

GPFS

scratch

Lustre

slurm commands

nfs

SLURM job

GRID job

OPS test jobs

dteam test jobs

ATLAS job factory

Big Data Science Accessing High-end HPC

15

Gianfranco Sciacca - AEC / LHEP Universität Bern • DI4R Conference 2016, 29 September 2016, Kraków
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

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**

![Graph showing job success efficiency for different systems and core counts](image)

- Tier-2 - single-core jobs
- Cray - single-core jobs
- Cray - 8-core jobs
- Tier-2 - 8-core jobs

**Job success Efficiency** 1.0
FIRST PRELIMINARY PERFORMANCE DATA

LHConCRAY - Performance with ATLAS HammerCloud stress tests: CPU efficiency

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

![Graph showing CPU efficiency and time for different workload configurations](image)

**Tier-2 - single-core jobs**
**Cray - single-core jobs**
**Cray - 8-core jobs**
**Tier-2 - 8-core jobs**

**Time** → **CPU/WT Efficiency**
## FIRST PRELIMINARY PERFORMANCE DATA

LHConCRAY - Performance with ATLAS HammerCloud stress tests: WallClock

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

<table>
<thead>
<tr>
<th></th>
<th>Successful jobs</th>
<th>Failed jobs</th>
<th>HEPSPEC06 core</th>
<th>Mean wallclock</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cray single core</td>
<td>4316</td>
<td>56</td>
<td>13.39</td>
<td>10386</td>
</tr>
<tr>
<td>Phoenix T2 single core</td>
<td>2117</td>
<td>0</td>
<td>11.46</td>
<td>13450</td>
</tr>
<tr>
<td>Cray 8-core</td>
<td>340</td>
<td>12</td>
<td>13.39</td>
<td>17978</td>
</tr>
<tr>
<td>Phoenix T2 8-core</td>
<td>75</td>
<td>0</td>
<td>11.46</td>
<td>23029</td>
</tr>
</tbody>
</table>

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
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
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**
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 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

- **CVMFS changes**
  - No longer using a shared, preloaded cache
  - Each node has its own XFS filesystem
  - The XFS filesystem is, in reality, a sparse XFS file
  - The underlying filesystem sees only one file: minimal metadata operations
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!
THE LHConCRAy PROJECT: OVERVIEW

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