Swiss Institute of Particle Physics

AEC
ALBERT EINSTEIN CENTER
FOR FUNDAMENTAL PHYSICS

Running on Cray
Status and Thoughts

S. Haug, AEC-LHEP UNIBE, at CSCS 2015-01-29

Outline

1. Motivation

1. WLCG model doesn't scale for HL-LHC (beyond 2020)

2. Need more science/computing for same money

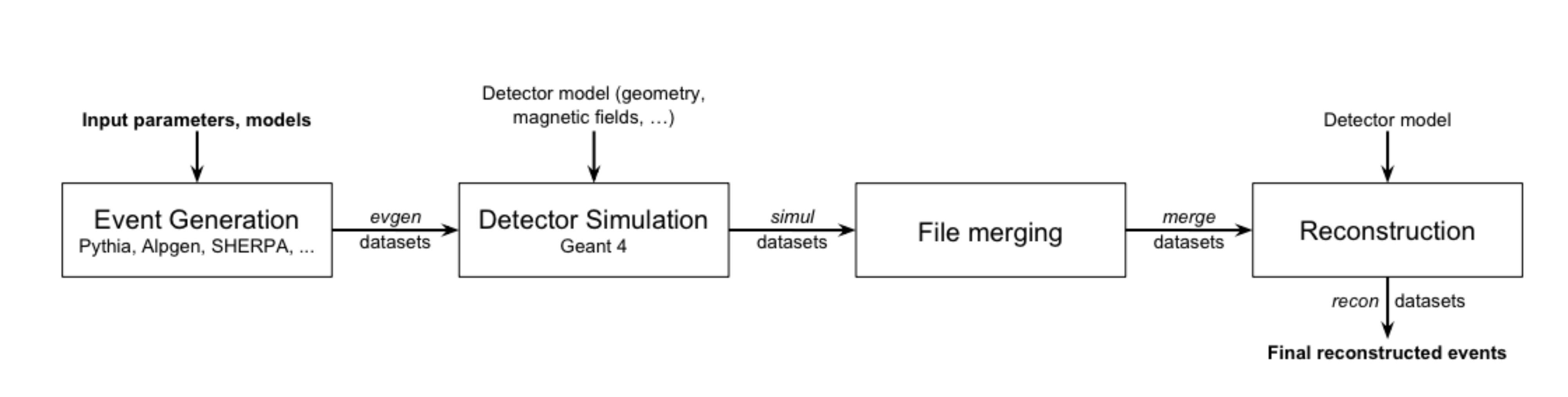
3. Part of the solution is to consolidate LHC computing
    1. Less but bigger sites world wide (operationally cheaper, better hw, etc). CSCS fits well as a world leading computing center
    2. Less but bigger systems in CSCS/CH (operationally cheaper, better hw, etc) is in CSCS interest
    3. Operational optimisation could go into sw optimisation / brain power

2.1 Target / development systems

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

1. Since a year we doing development and operational commissioning on Todi

2.2 Workflow steps



1. **Have studied and enabled event generation and detector simulation**
2. **These steps have moderate i/o (less than a GB per job, i.e. node)**

Full simulation time	~900 s/1 event
Memory usage	~2 GB
Job size	100 events
Input file size	< 300 MB/1000 events
Output file size	< 100 MB/100 events

Table 2.4: Typical ATLAS Geant4 full simulation job requirements

2.3 Compiling and SW Provisioning

1. First we had a 3 months CSCS preparatory project (two accounts on Todi) in which we successfully tested compilation and running of standalone Sherpa and GEANT4 ATLAS jobs

Part	Inode count
ATLAS software release (17.7.3)	427013
ATLAS condition database	8371
atlas-gcc	3062
Current ATLAS database release	1756
Total	**440202**

Table 2.3: Number of inodes (files and directories on the file system) used by the ATLAS CVMFS repository.

1. Enabled application sw access via Parrot (file system wrapper). Mounting /cvmfs as normal user. For multi-threaded jobs we had to move to rsync due to race conditions not handled by Parrot.
2. Default inode limit at CSCS was 0.5M, a bit close to limit

2.4 Performance - RAM and Threads

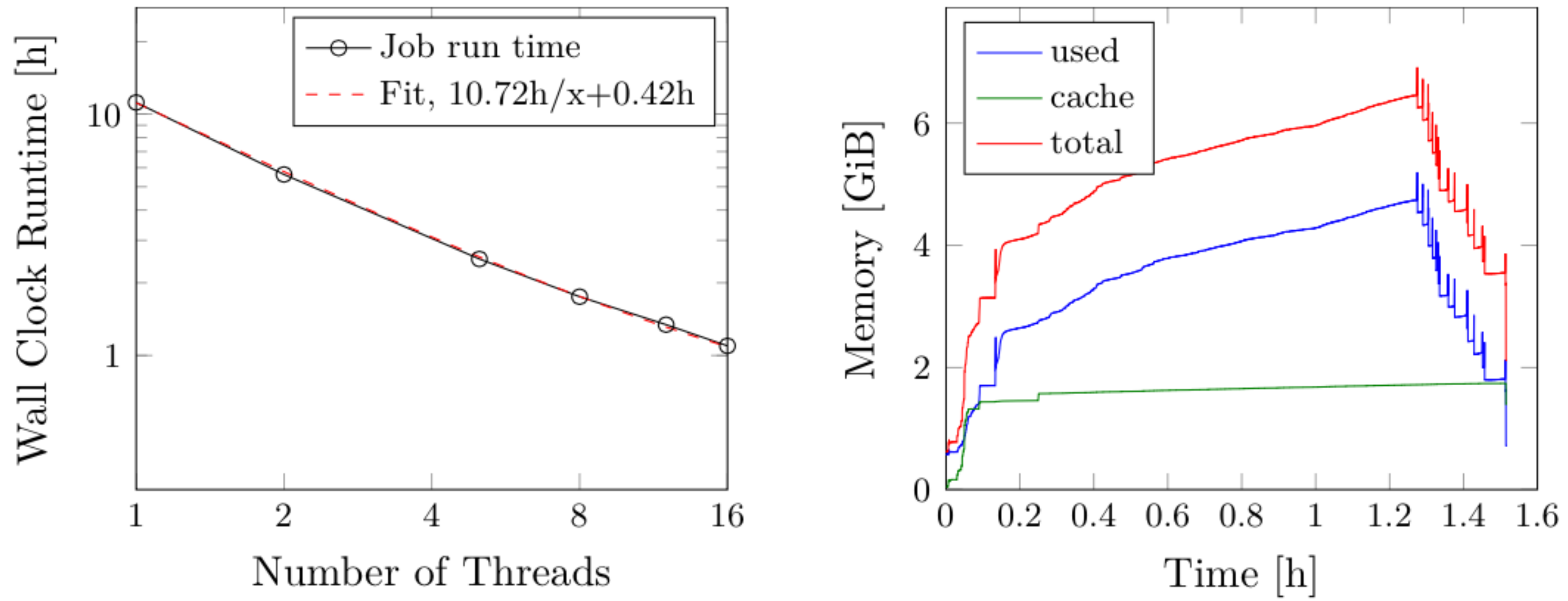


Figure 2.5: Thread-scaling of jobs. The scaling is near-perfect (linear), with a slight offset due to the initialization and finalization steps. The total memory usage of a 16-threaded job processing 100 events is much lower than the 32 GB available per node.

1. **The usage of a full node (16/32 CPU cores) scales well and does not hit memory limits due to multi-threading (one job per node)**

2.5 Performance - Nodes and Comp

Requested simultaneous jobs	10	100
Average running jobs	10 ± 0	95.3 ± 4.5
Completion rate [jobs/h]	7.28 ± 3.04	68.8 ± 13.9

Table 2.5: Comparison of ATLAS simulation jobs running on 10 and 100 compute nodes in parallel.

1. **Large scale test (October) showed that the use of many nodes scales linearly (as expected)**

Compiling with Cray recommended options brings about 5%.

Cray compiler is worse than precompiled gcc

Random Seed	Precompiled	Optimized gcc	CrayCC
539155	880 s	834 s	1219 s
939155	879 s	833 s	1208 s
139155	887 s	840 s	1178 s

Table 2.6: Processing time per event for different ATLAS Geant4 builds.

2.6 GPU Usage

1. Detector simulation with GEANT4 is Monte Carlo base, i.e. throwing random numbers. However, one standard ATLAS GEANT simulation needs about 40 MB, i.e. 10x available GPU memory
2. We replaced the random number generator with one for GPUs. GPU then provides the numbers needed by the CPUs. The generation is 5 to 10 faster than with standard generator
3. However, the total achieved gain was about another 5%
4. This is little, however, we can use the GPUs
5. Possible next step is to export the Runge-Kutta solving for particle propagation in external magnetic fields to the GPU. Standalone tests indicates a factor 30 speed up, however, integration into GEANT is not straight forward.

2.7 Production system integration

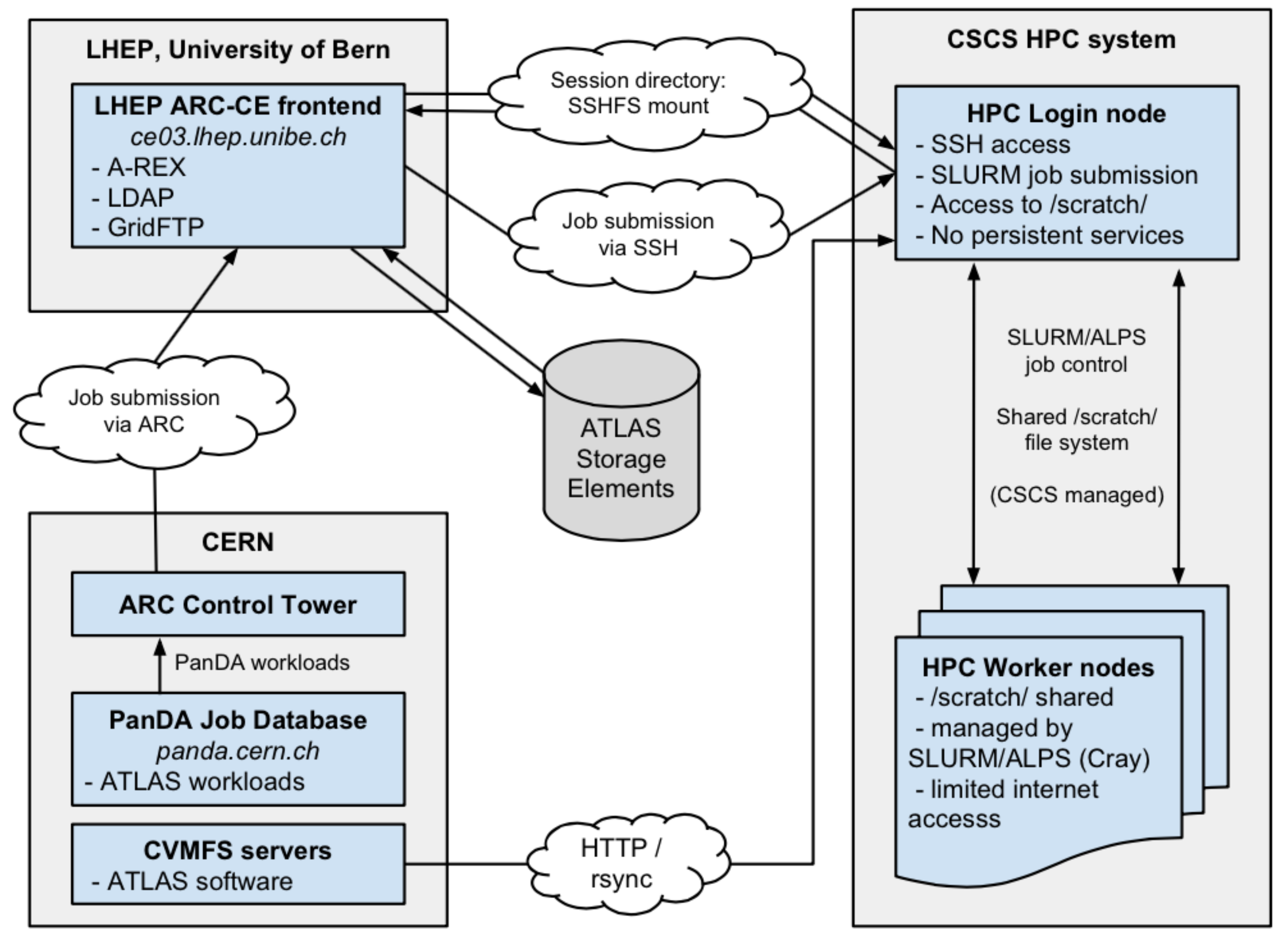
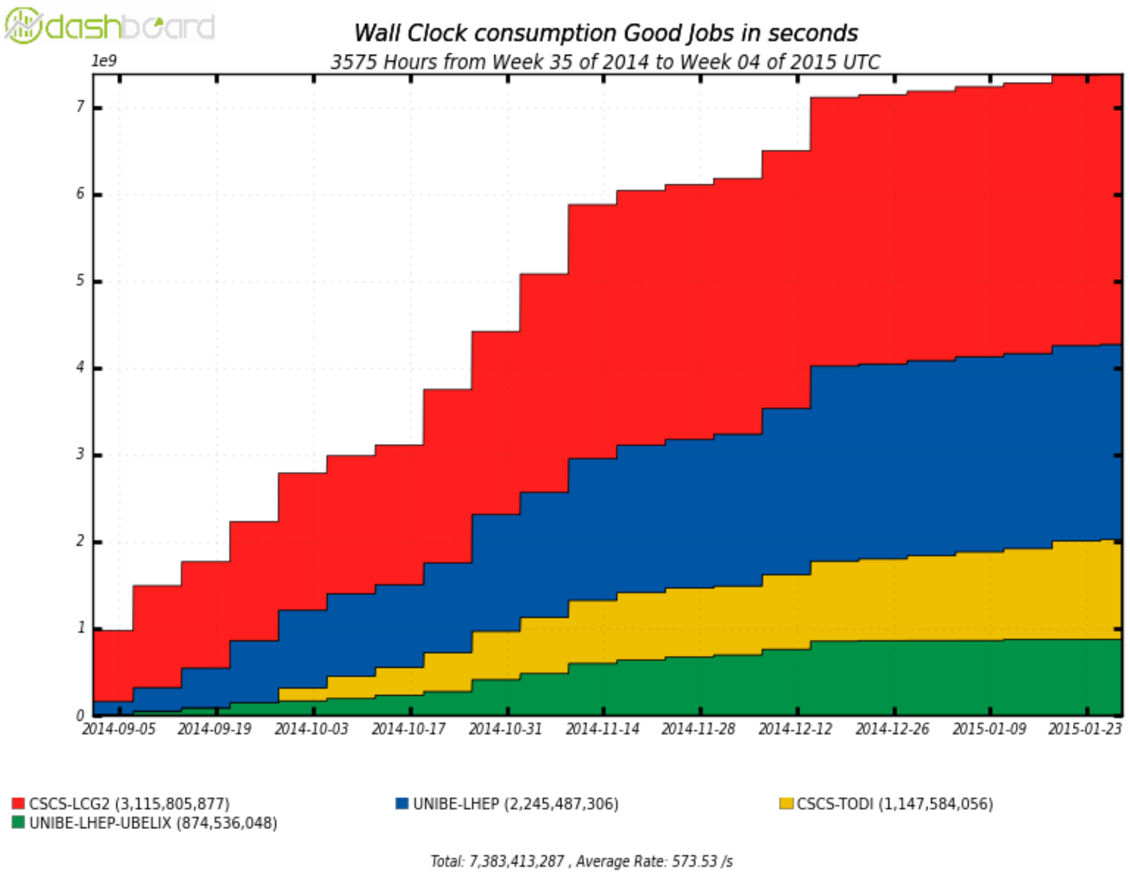
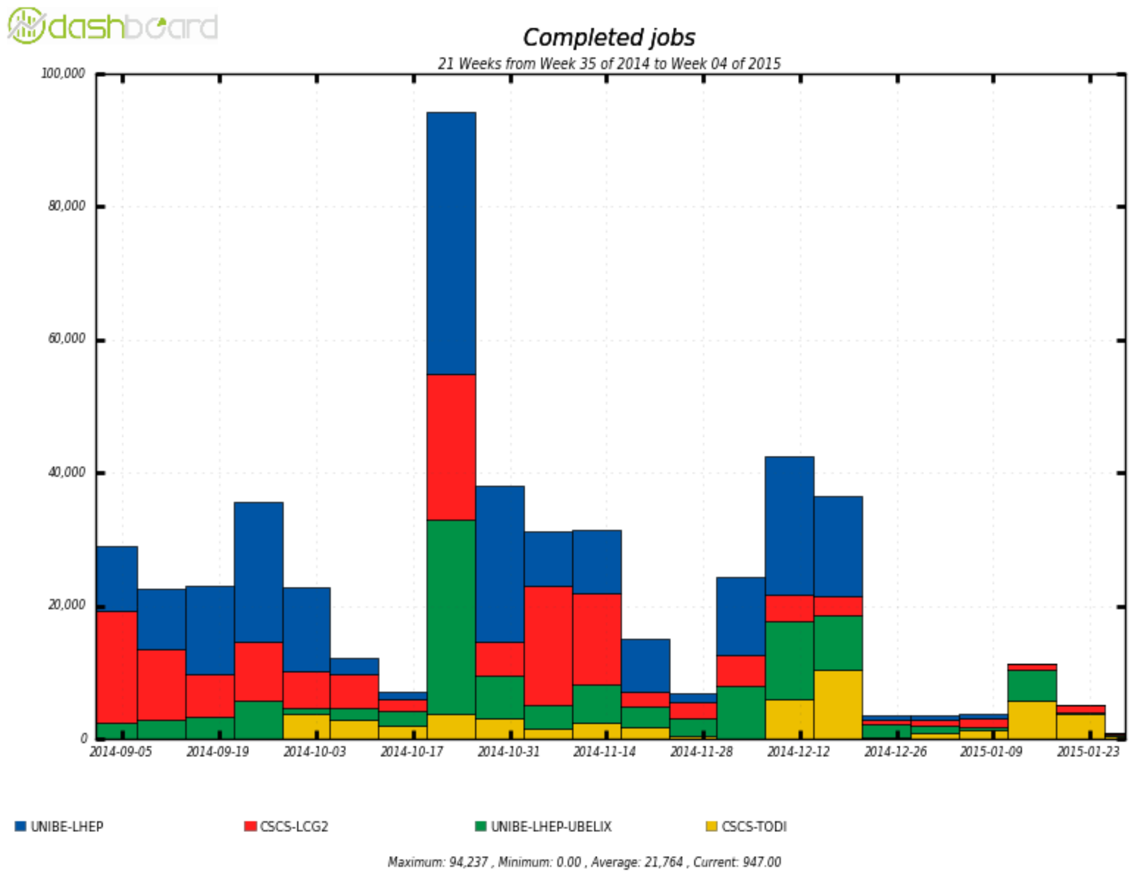


Figure 2.7: The architecture of the ARC-CE as a frontend to a remote HPC system. Clouds denote connections made over the internet.

1. This our solution is now used for SuperMUC (Munich), Hydra (Munich) and Pi (Shanghai/China).
2. The ssh ARC back-end may become standard in ARC

3. Accounting





4. Dissemination

1. ATLAS presentations
2. PASC14 (one poster and one talk)
3. To CHEP15 with two posters and proceedings
4. Ultimate test would be the 50 MCPU hour project (CHRONOS application). Then several presentations and publication planned.

4. Thoughts / Conclusions

1. The Crays can run LHC simulation jobs
2. Very cheep in operation, (close to?) no intervention since October
3. It is possible to consider a model running experiment production on multi-usage high-end HPC machines at CSCS
4. Probably operationally cheeper, machines faster and stable
5. User jobs and special cases could run at “home” (PSI, UNIBE/ UNIGE …)

4. Possible next steps

1. Await CHRONOS application decision
2. Anyway ask CSCS to continue to provide some back fill machine (Monte Rosa / Todi …) for further development and consolidation. Gradually move computation to the large HPC systems.
3. Help CMS and LHCb onto Cray/HPC (just need an ARC back-end to their production systems)?
4. Ask CSCS team to assess the feasibility of using HPC systems in future

Additional Material

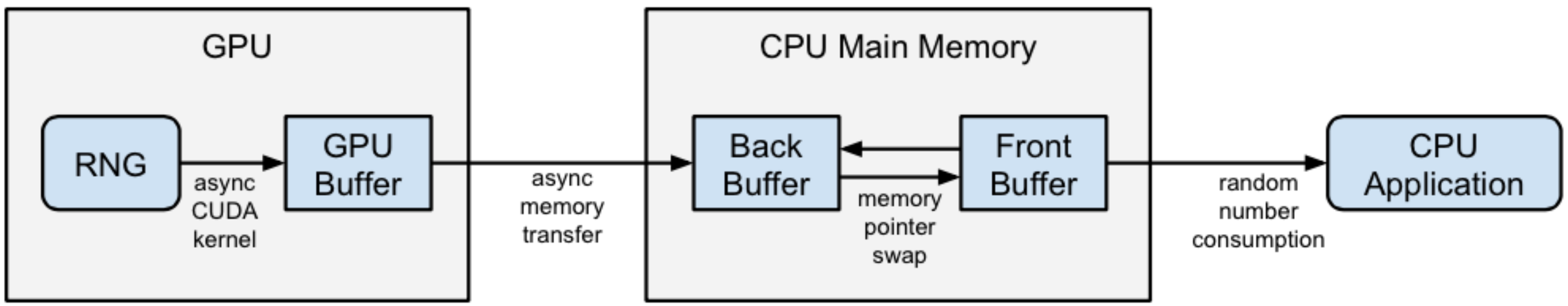


Figure 2.6: Principle of the double-buffered CUDA RNG

Directory	Contents	Required for production jobs
`ATLASLocalRootBase`	General, release-independent setup and software management scripts, e.g. for setting up specific versions of the various ATLAS software compontents. While users usually use this scripts to set up specific releases of the ATLAS software, production jobs set up the release directly and don't use ATLASLocalRootBase.	No
`conditions`	Symbolic link to the `atlas-condb` repository, which contains the ATLAS condition database, i.e. additional non-event data from the ATLAS detector [15]. It also holds detector parameters used for partially parametrized detector simulation (ATLAS fast simulation).	Yes
`dev`	Software development and testing area.	No
`sw/database`	Versioned ATLAS database, which contains e.g. the description of detector geometry and physical parameters. At least one (current) release of the database has to be provided in order to run production jobs.	Yes
`sw/atlas-gcc`	The GNU Compiler Collection (gcc) used to build the ATLAS software, including any associated libraries. Software dynamically linked need to access these libraries.	Yes
`sw/software`	Versioned ATLAS software stack. At the time of writing, the large-scale ATLAS Production tasks use the 17.7.3 and 17.7.4 releases of the ATLAS software for detector simulation and event generation, so at least these releases have to be provided in order to run current the considered steps of ATLAS production.	Yes

Table 2.2: The ATLAS CVMFS repository organization.

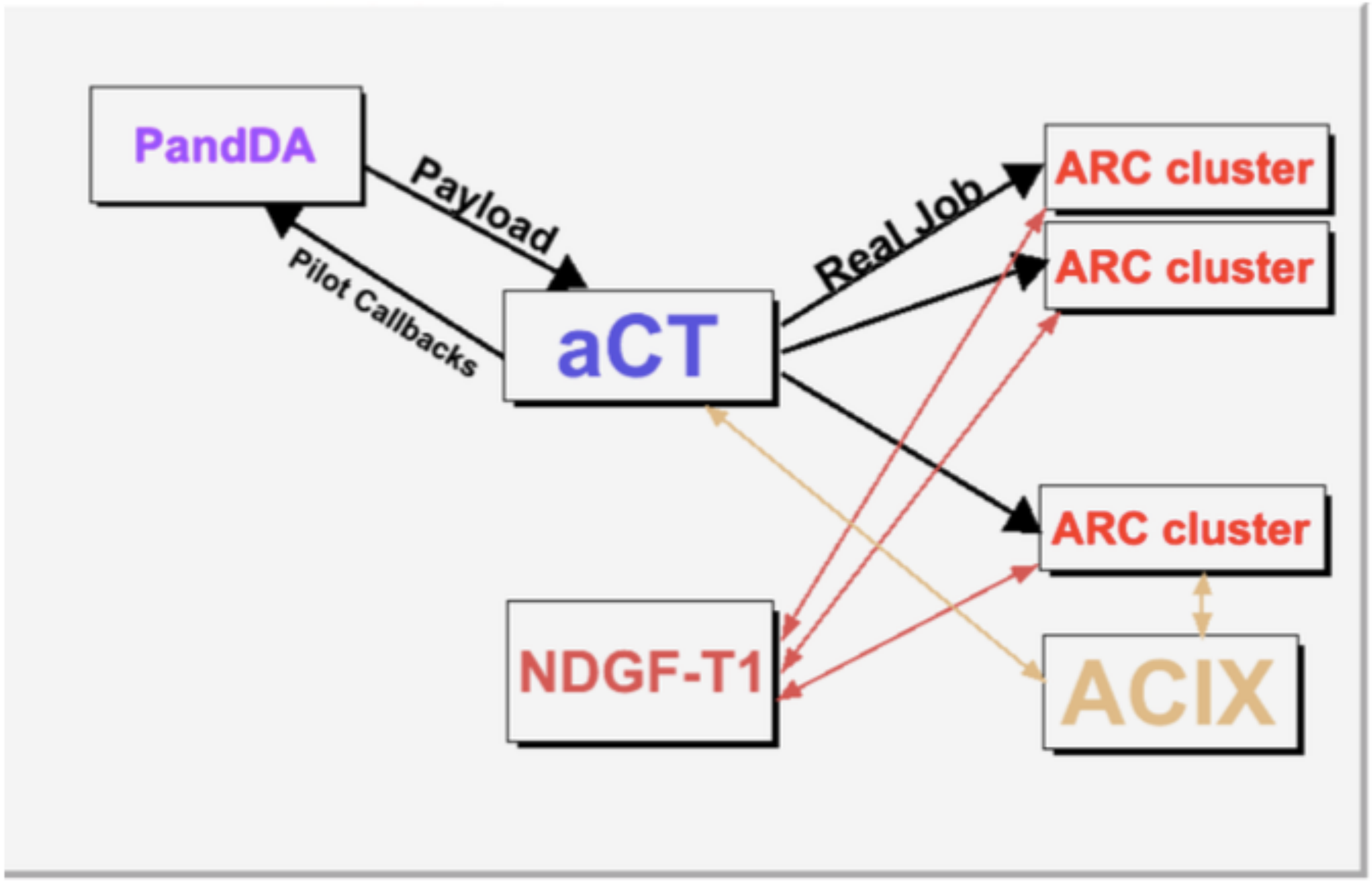


Figure 2.4: The job flow when using ARC and the arcControlTower. (Source: [13])

Country	Site	CPUs	Load (processes: Grid+local)
Denmark	Steno Tier 1 (DCSC/KU)	3476	735+2249
Germany	LRZ-C2PAP	4072	512+3108
	LRZ-LMU	288	0+0
	LRZ-LMU lcg-lrz-ce0	1824	8+13
	LRZ-LMU lcg-lrz-ce3	1824	13+8
	LRZ-LMU_MUC	3200	0+35
	RZG ATLAS HYDRA	167848	0+148086
	wuppertalprod	3684	145+1515
Norway	Abel C1(UiO/USIT)	10880	484+8995
	Abel C3(UiO/USIT)	10880	528+7342
Slovenia	Arnes	2280	1741+0
	SiGNET	2834	2165+5
Sweden	Abisko (HPC2N)	15936	547+13950
	Alarik (SweGrid, Luna>	3776	315+2839
	Triolith - Atlas (NSC)	25472	376+23902
Switzerland	ATLAS BOINC	17147	2913+1242
	Bern ce01 (UNIBE-LHEP)	1368	781+0
	Bern ce02 (UNIBE-LHEP)	776	449+10
	Bern LHEP HPC TEST	4208	336+3504
	Bern UBELIX T3	2600	203+2047
	Geneva (UNIGE-DPNC)	184	250+86
	Lugano PHOENIX T2	3098	0+2287
	Lugano PHOENIX T2	3098	12+2275
UK	arc-ce01 (RAL-LCG2)	13704	2110+8761
	arc-ce02 (RAL-LCG2)	13704	2160+8713
	arc-ce03 (RAL-LCG2)	13704	2316+8549
	cetest01 (UKI-LT2-IC->	4	39+1563
	t2arc01 UKI-SOUTHGRID>	1200	685+67
TOTAL	*28 sites*	*333069*	*19743 + 251151*

Figure 2.2: Screenshot of the ATLAS Nordugrid Monitor [11] on December 19, 2014. The ARC HPC frontend developed in this work shows up as „Bern LHEP HPC TEST". The frontend interfaces the „Todi" HPC system at CSCS and run ATLAS jobs on 336 of its 4208 cores when the screenshot was taken.

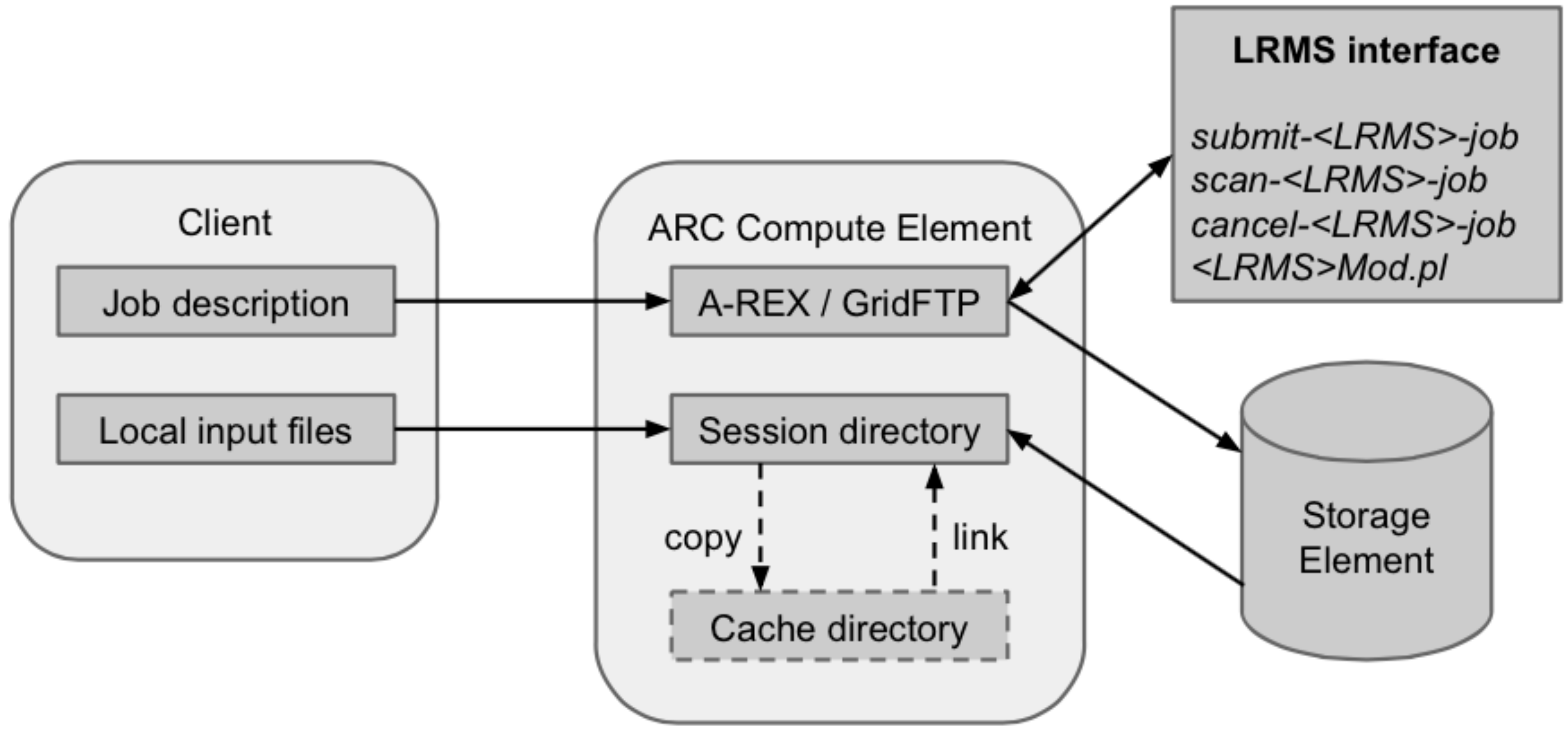


Figure 2.1: Overview of the ARC job information flow. The dashed caching part is an optional feature.