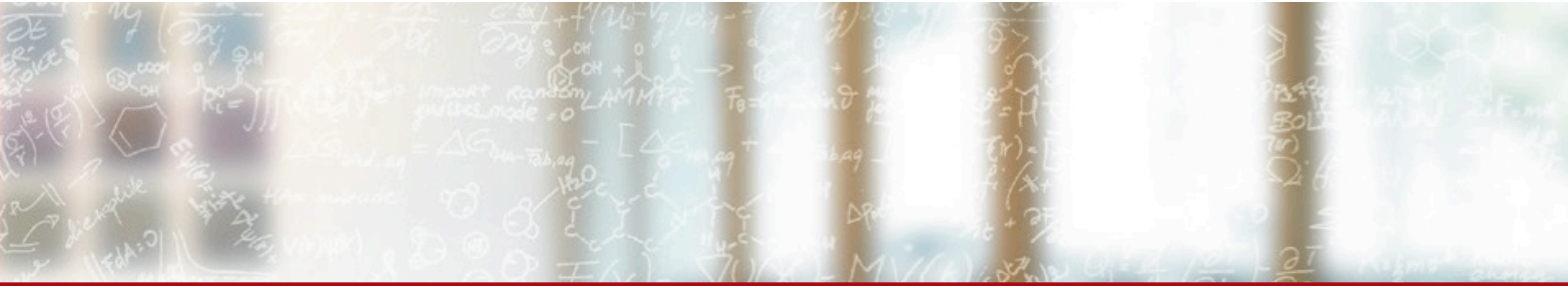
CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

ETH zürich



Experience running on Piz Daint @ CSCS

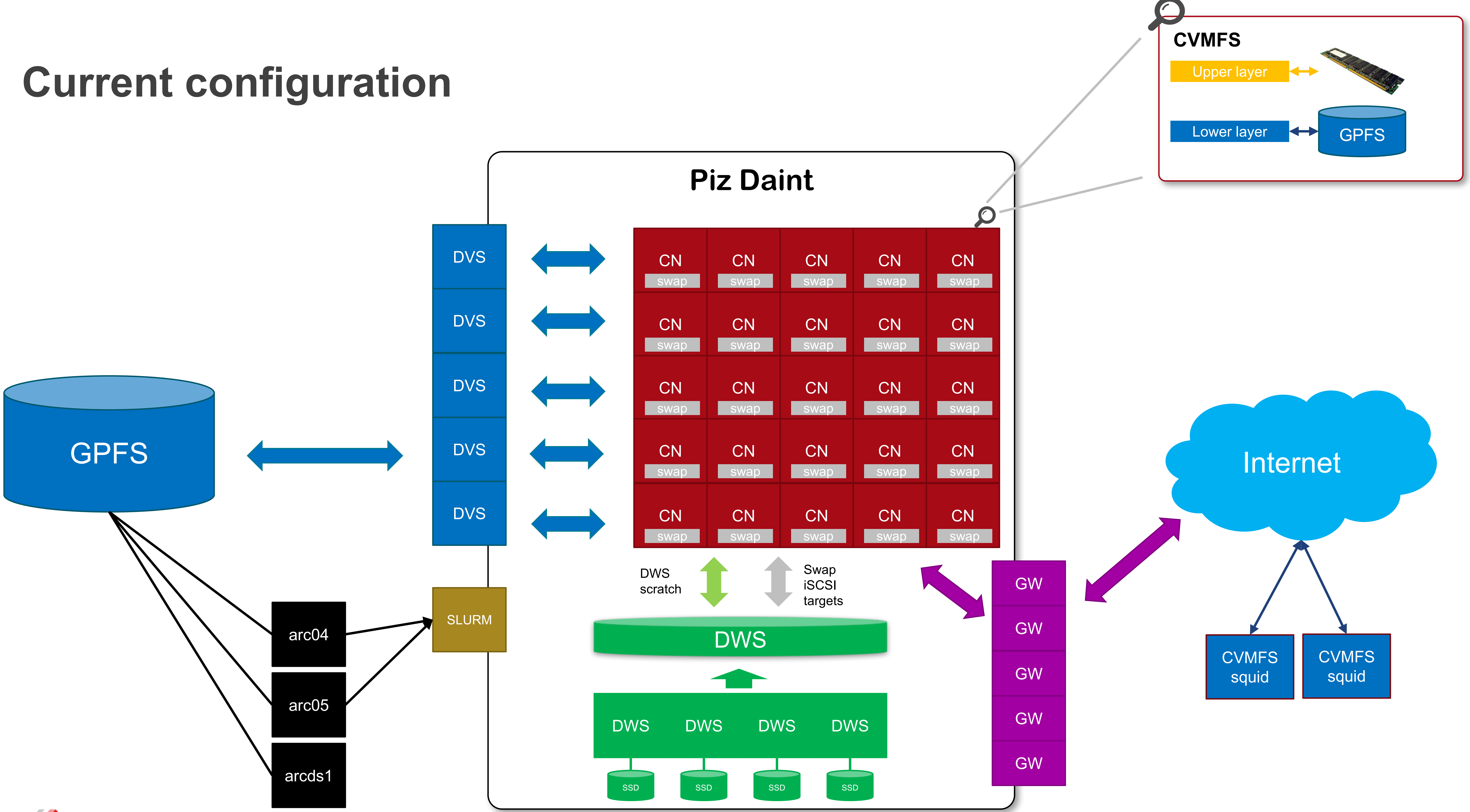
Gianfranco Sciacca
University of Bern

ATLAS TIM, CERN
October 2017

Operational challenges

- **Data delivery / access** ✓
  - network connectivity
- **Diskless nodes** ✓
  - scratch area, job workdirs, ARC sessiondirs
  - /tmp
  - swap
- **Memory management** ✓
  - .le. 2GB/core
- **Job scheduling** ✓
  - job prioritisation and fair-share in the global environment
- **Software provisioning** ✓
  - CVMFS cache performance
- **OS environment** ✓
  - Cray Linux Environment (stripped down SUSE)
- **Scalability** ?
  - depends on all of the above

Current configuration



Current configuration - data access, memory, scheduling, OS

- **25 compute nodes: 72 HT cores (Broadwell), 128GB RAM, diskless, 64-68 cores used**
  - nodes are dedicated and have IP connectivity with public IP addresses ✓
- **1 production ARC CE + 1 ARC data stager** + 1 test ARC CE (*internal*) - in **ARC native mode**
  - Perform full data staging I/O ✓
  - Can scale up the number of stagers as needed ✓
  - ARC **caching not enabled**: each job has its own copy of all files (*at least for now*) ✓✓
- **SLURM LRMS**
  - Dedicated WLCG partition (*jobs are not node-exclusive - 1-core or 8-core*) ✓
  - **Memory is not consumable**. Enforce 6GB/core limit for to catch rogue jobs ✓
  - **swap** on DataWarp **enabled**: one iSCSI device per node with 64GB each (*not really used yet*) ✓
  - Bypassing `--nice` in `submit-SLURM-job` : seems to break fair-share penalising ATLAS ✓
  - *When scheduling is disrupted due to rogue users, all suffer* ✓
- **OS environment: CLE6.0** (*based on SUSE 12*)
  - Jobs run in **Docker containers using Shifter** ✓
  - Image is a WLCG full WorkerNode (*CentOS6, EMI3, HEP_OSlibs_SL6*) ✓
  - `https://hub.docker.com/r/cscs/wlcg_wn:20170731`

Current configuration - shared file systems

- **Most critical pieces of the puzzle, ongoing work**

- Dedicated **GPFS file system** shared with the Phoenix T2 cluster
  - Used by ARC for input data staging ✓

- 5 **DVS** (Cray Data Virtualisation Service) nodes exposing GPFS to the CNs via 40GbE links
  - A few DVS related issues/bugs to deal with
  - Had to turn off ARC caching => issues with symlinks over DVS (will likely be fixed with the next CLE update) ✓
  - Issues when a file is accessed by multiple clients, performance degrades very quickly => job timeouts ✓

- 4 **DWS** (Cray Data Warp Service), SSD-based (`http://www.cray.com/datawarp`)
  - cannot mount on nodes external to the Cray, e.g. the ARC CEs for ARC sessiondirs
  - **swap** on DataWarp **enabled**: one iSCSI device per node with 64GB each (*not really used yet*) ✓
  - job workdir (`$RUNTIME_LOCAL_SCRATCH_DIR`) and /tmp: ongoing work ✓
    - the key is to distribute metadata operations to more servers
    - this requires creating dynamic allocations per job with a fixed size => CLE update on 27 Sep

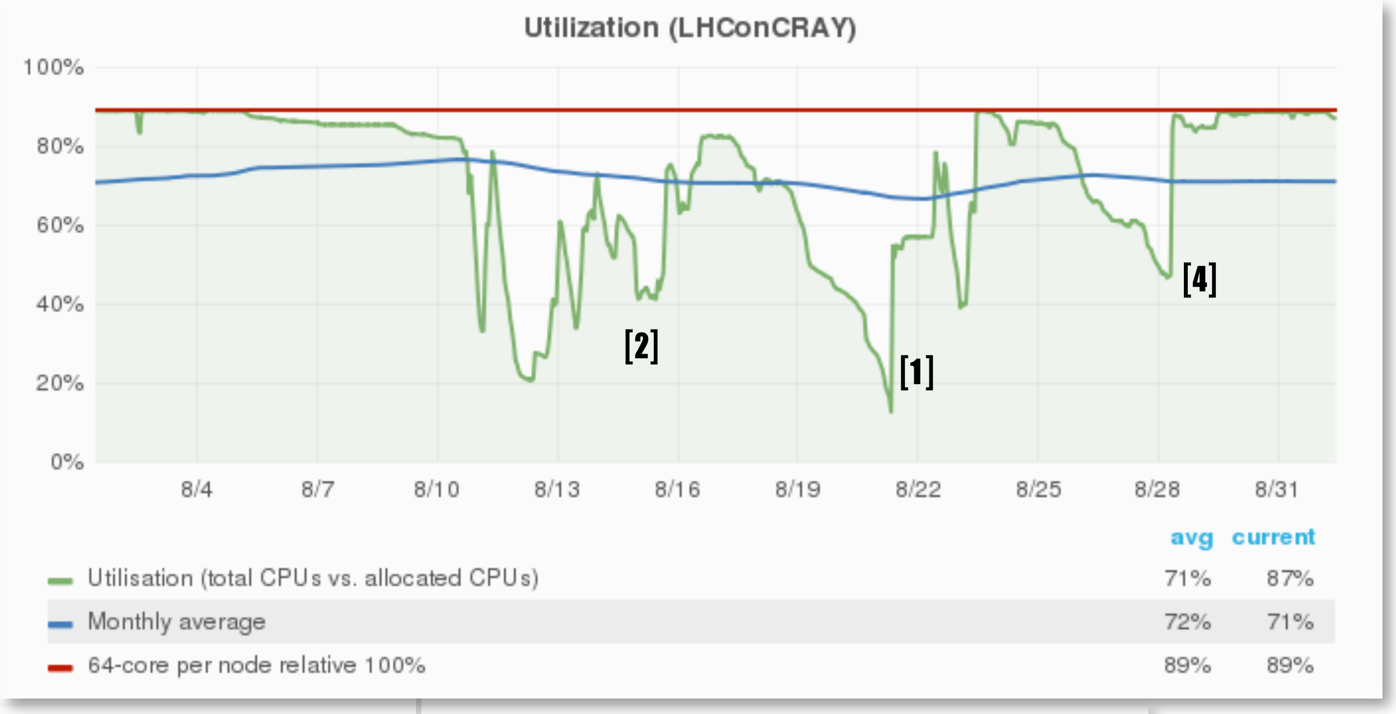
- **Docker images**
  - On the Cray Sonexion 1600 Lustre FS ✓
  - so far it has worked very well with no IO penalties because of being on Lustre ✓

Current configuration - CVMFS ✓

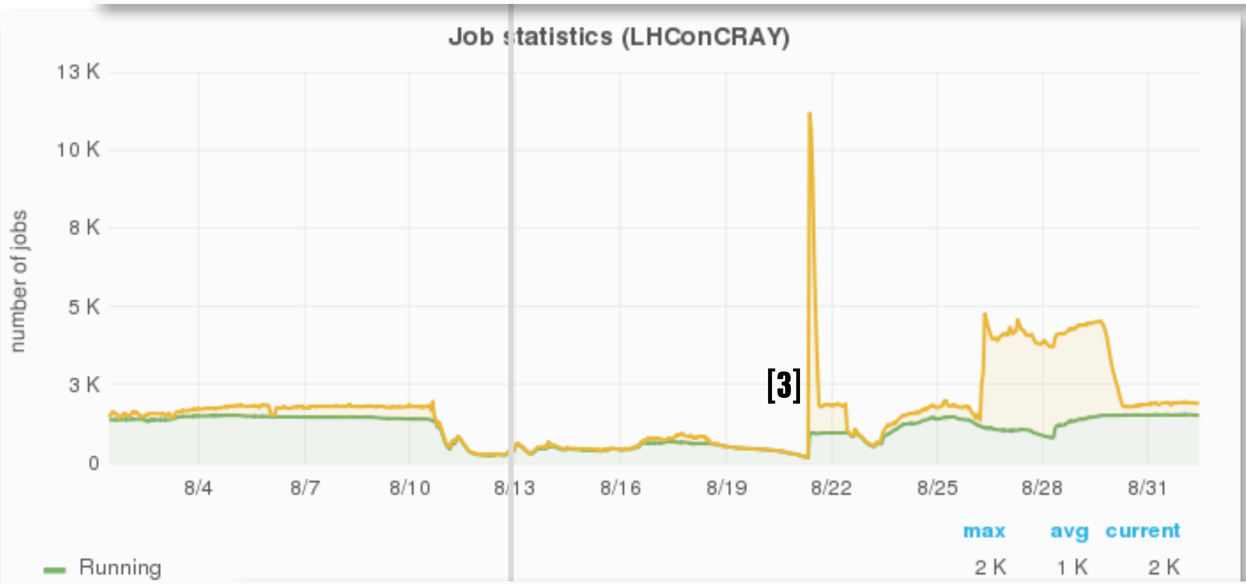
- CVMFS running natively on CNs using **workspaces** and **tiered cache**
- was previously configured to use a XFS loopback filesystem on top of DVS as local cache
  - the two new features from the CVMFS developers, allow us to store data directly on a DVS projected filesystem (no more XFS)
  - DVS does not support \`flock()\`, with the **workspace** setting it is now possible to set all locks relative to the cache local to the node (or ramdisk)
  - **tiered cache with in-ram storage**: it is now possible to instruct cvmfs to store its cache in memory, without the need for local storage. This can dramatically increase performance. **We have an upper layer of 6GB in-RAM per node (shared by all)**. Cache on DWS suffered from data corruption
  - **Lower layer on GPFS**: all needed cvmfs repos have been preloaded onto GPFS thanks to a new, fast service provided by CERN for HPC sites. This syncs several times a day. If a file is not found on the local caches, the query propagates to the outside.

System utilization and issues

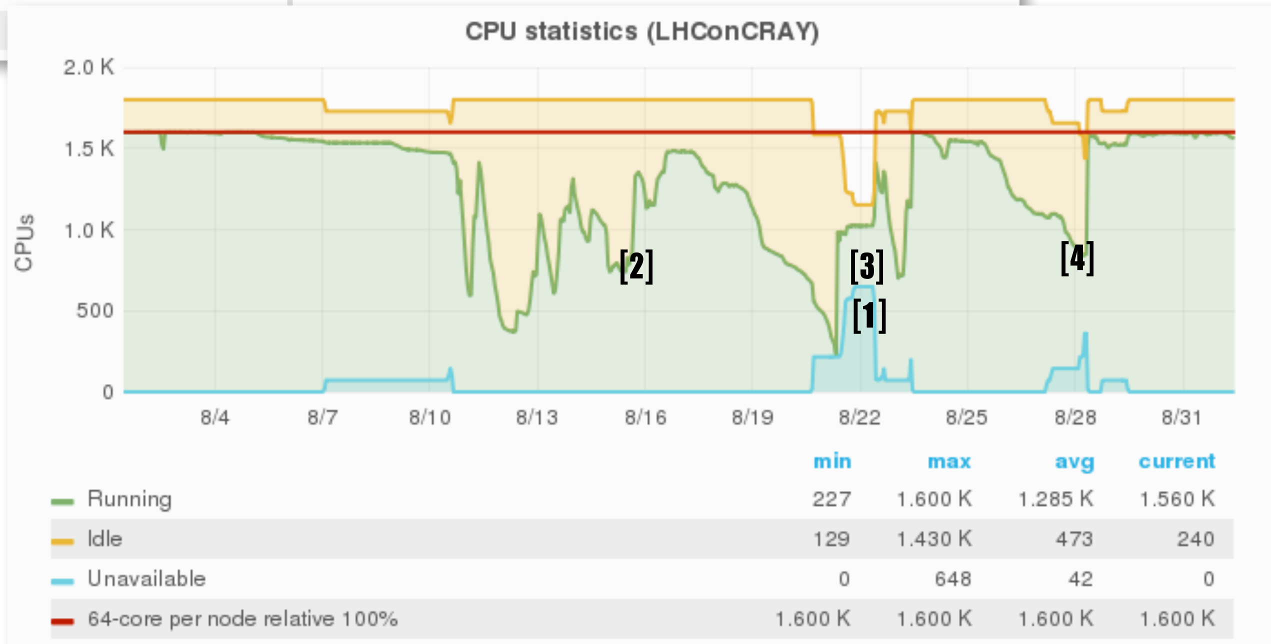
- Core allocation up to 100% relative to 64core/node (out of 72) for long periods of time
- Encountered certain issues with ARC delegations [1] and nodes becoming silently blackholes [2]
- LHCb submitted ~10K jobs because of a problem with ARC BDII [3]
- Non LHC users hammered Slurm consistently and this affected scheduling for a while [4]
- ATLAS has picked up on LHCb and CMS seems to be be consistently running a low number MC of jobs


Utilization (LHConCRAY)

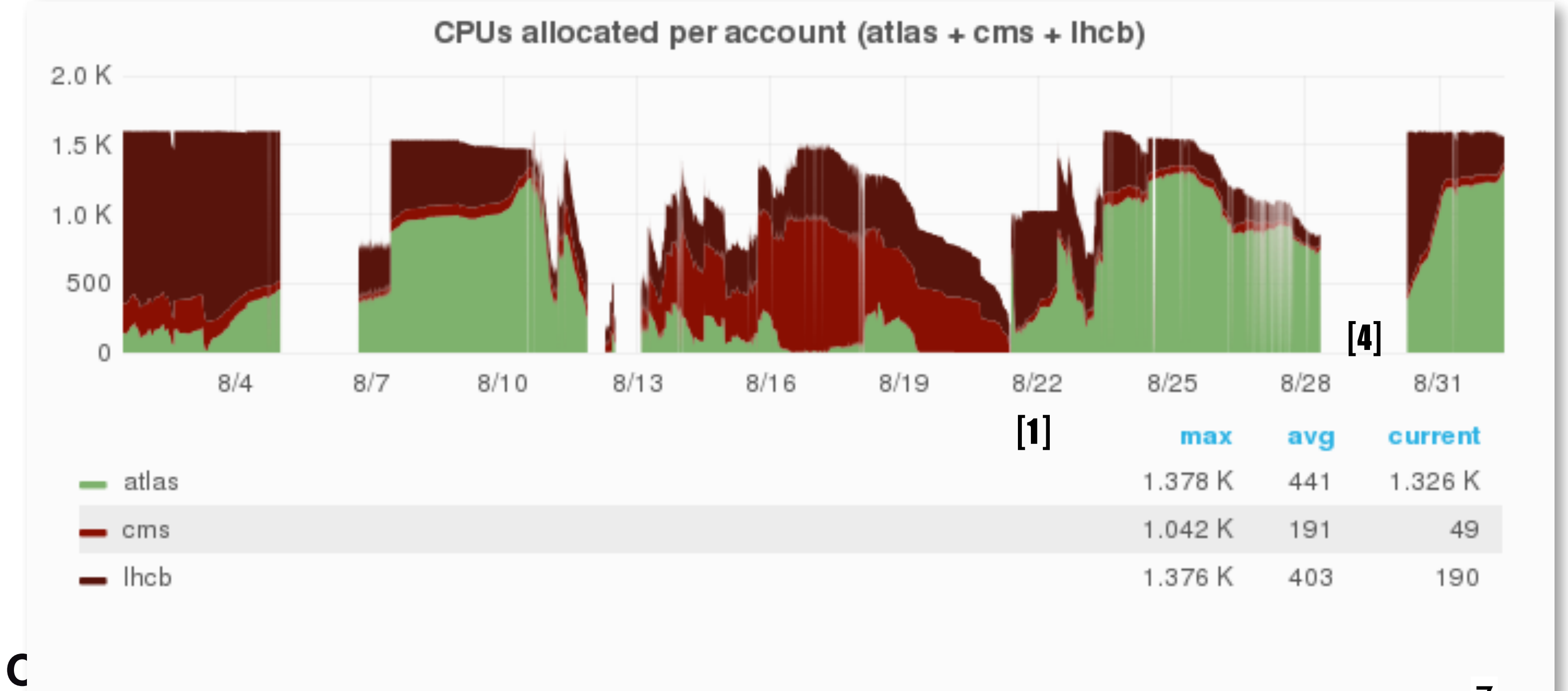
	avg	current
Utilisation (total CPUs vs. allocated CPUs)	71%	87%
Monthly average	72%	71%
64-core per node relative 100%	89%	89%


Job statistics (LHConCRAY)

	max	avg	current
Running	2 K	1 K	2 K
Pending			


CPU statistics (LHConCRAY)

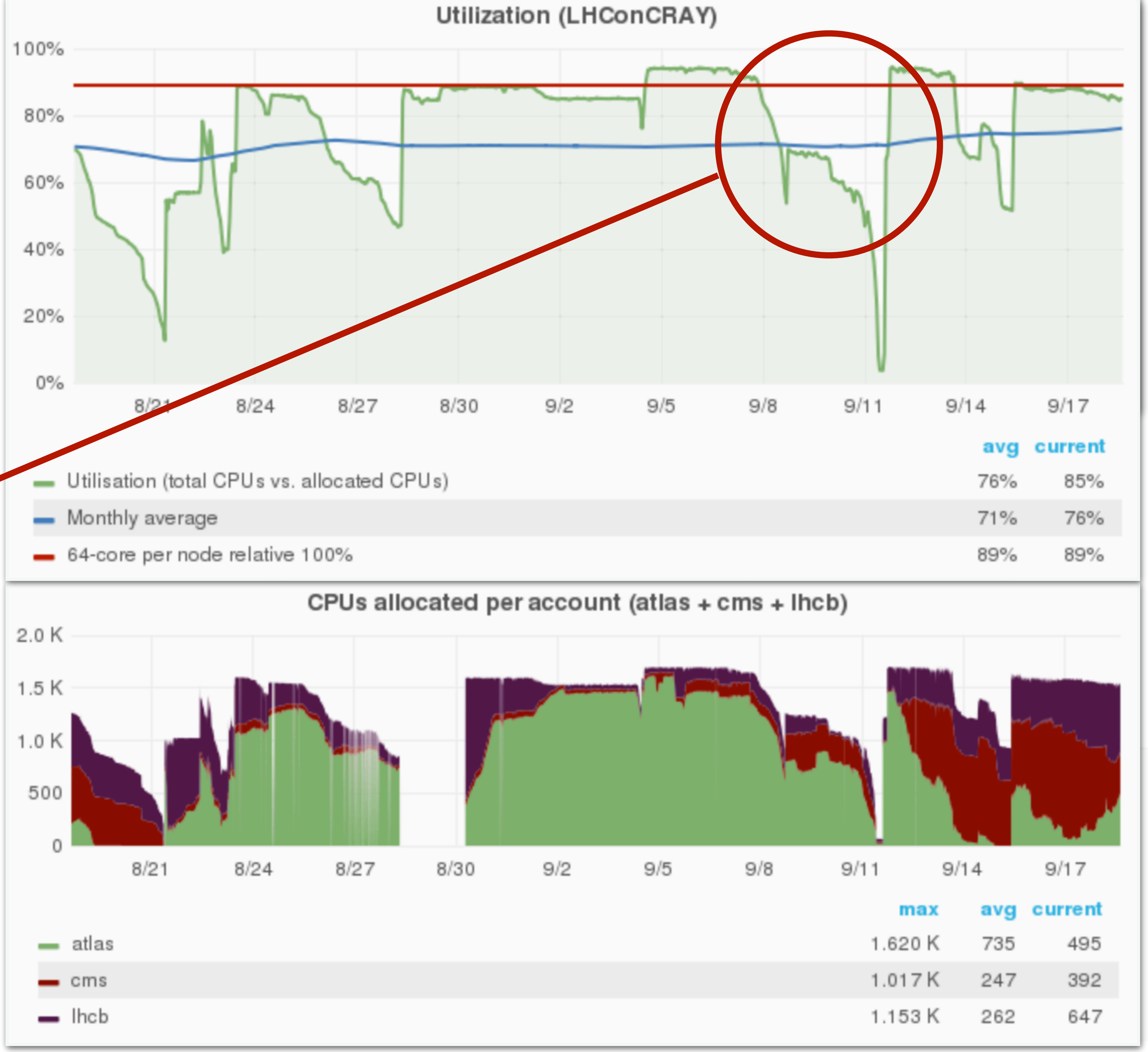
	min	max	avg	current
Running	227	1.600 K	1.285 K	1.560 K
Idle	129	1.430 K	473	240
Unavailable	0	648	42	0
64-core per node relative 100%	1.600 K	1.600 K	1.600 K	1.600 K


CPUs allocated per account (atlas + cms + lhcb)

	max	avg	current
atlas	1.378 K	441	1.326 K
cms	1.042 K	191	49
lhcb	1.376 K	403	190

System utilization and issues

- several CMS jobs writing 200k files each exhaust inodes on GPFS
- hard quota per VO set in place



Utilization (LHConCRAY)

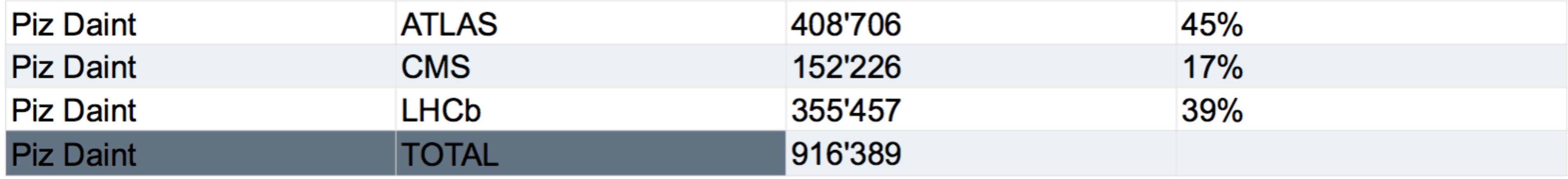
	avg	current
🟩 Utilisation (total CPUs vs. allocated CPUs)	76%	85%
🟦 Monthly average	71%	76%
🟥 64-core per node relative 100%	89%	89%

CPUs allocated per account (atlas + cms + lhcb)

	max	avg	current
🟩 atlas	1.620 K	735	495
🟥 cms	1.017 K	247	392
🟪 lhcb	1.153 K	262	647

Summary report

- *Relatively* stable operation, all VOs now capable of running jobs
- Overall CPU utilisation reaching the relative maximum
- Memory utilisation: about 30GB in cache, about 1GB free on average
- Swap not really used so far, might reduce the size
- CVMFS in RAM seems to work quite well, not a single issue since we have enabled it
- DVS and node load is high at times due to I/O
- Fair-share seems to work now, although must really understand if due to bypassing `--nice`
- CPUs were unavailable due to auto-drain or maintenance for 9% of the total CPUhours available (August)
- auto-drain algorithm made smarter, we expect improvements



Piz Daint	ATLAS	408'706	45%
Piz Daint	CMS	152'226	17%
Piz Daint	LHCb	355'457	39%
Piz Daint	TOTAL	916'389	

916'389 is **85**% of the total available time (1'075'200) !

- **Scalability is a concern at this stage:** handed in a proposal for a scalability test (~75k slots) to be carried out while the system is being drained ahead of the 27th Sept downtime