



CSCS

Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

ETH zürich



CUG PEAD SIG BOF

CSCS site update

May 26, 2017



CSCS

Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

ETH zürich

CSCS Overview

CSCS Cray/XC



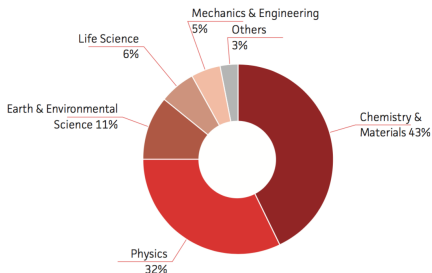
Piz Daint

- Hybrid/Multicore CrayXC/40+50, Aries (dragonfly):
- Each XC/50 compute node hosts 1 Intel Haswell CPU and 1 NVIDIA P100 GPU,
- Each XC/40 compute node hosts 2 Intel Broadwell CPUs,
- $\sim=9$ Pflops Rmax (#2 in Green500, #8 in Top500, 11/2016 list)

CSCS users (2016)

Usage by Research Field

Research Field	CPU h	%
Chemistry & Materials	11 482 049	43
Physics	8 664 124	32
Earth & Environmental Science	2 826 961	11
Life Science	1 568 041	6
Mechanics & Engineering	1 275 781	5
Others	951 620	3
Total Usage	26 768 576	100



Piz Daint

- 43 million compute node hours used in 2016
- 124 research projects, >1000 users
- Swiss and non Swiss research institutions



CSCS

Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

ETH zürich

Easybuild @ CSCS

Easybuild

- A software build and installation framework that allows you to **manage scientific software** on Cray (and non Cray) systems in an efficient way.
- **Making Scientific Software Installation Reproducible On Cray System Using EasyBuild:**
www.cug.org/proceedings/cug2016_proceedings/ → pap145-file2.pdf

Supported Scientific Applications:

- Amber/16-2016.11-CrayGNU-2016.11
- CP2K/4.1-CrayGNU-2016.11
- CPMD/4.1-CrayIntel-2016.11
- GROMACS/5.1.4-CrayGNU-2016.11
- LAMMPS/30Jul16-CrayGNU-2016.11
- NAMD/2.11-CrayIntel-2016.11
- QuantumESPRESSO/5.4.0-CrayIntel
- VASP/5.4.1-CrayIntel-2016.11

Libs and Tools:

- Boost/1.63.0-CrayGNU-2016.11-Python-3.5.2
- CDO/1.8.0rc6-CrayGNU-2016.11
- Charm++/6.7.1-CrayIntel-2016.11
- GDAL/2.1.0-CrayGNU-2016.11
- GEOS/3.6.0-CrayGNU-2016.11
- NCL/6.4.0
- NCO/4.6.2-CrayGNU-2016.11
- Octave/4.2.0-CrayGNU-2016.11
- R/3.3.2-CrayGNU-2016.11
- magma/2.2.0-CrayGNU-2016.11
- moose/b675783-CrayGNU-2016.11
- ncview/2.1.7-CrayGNU-2016.11

Python:

- Python/2.7.12-CrayGNU-2016.11
- Python/3.5.2-CrayGNU-2016.11
- h5py/2.6.0-CrayGNU-2016.11
- netcdf-python/1.2.7-CrayGNU-2016.11
- pycuda/2016.1.2-CrayGNU-2016.11

Analysis:

- Score-P/3.0-CrayGNU/Intel/PGI/CCE
- Scalasca/2.3.1-Cray*

Visualisation:

- ParaView/5.3.0-CrayGNU-2016.11
- VMD/1.9.3-egl
- VTK/7.1.0-CrayGNU-2016.11
- Visit/2.12.0-CrayGNU-2016.11

DL toolkits ****new****

- Caffe/1.0-CrayGNU-2016.11
- Caffe2/0.7.0-CrayGNU-2016.11
- CNTK/2.0.rc1-CrayGNU-2016.11
- Keras/2.0.3-CrayGNU-2016.11
- TensorFlow/1.0.0-CrayGNU-2016.11
- Theano/0.8.2-CrayGNU-2016.11

<https://github.com/eth-cscs/production>

Production repository:

- All CSCS EasyBuild recipes
- List of supported recipes on [Daint](#)
- <https://github.com/eth-cscs/production>



Jenkins

Monitor changes

Deploy new recipes





CSCS

Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

ETH zürich

Regression testing @ CSCS

What is Reframe ?

A new regression framework that

- allows writing portable HPC regression tests in Python,
- abstracts away the system interaction details,
- lets users focus solely on the logic of their test.

<https://github.com/eth-cscs/reframe>

CUG'17 talk

- **A regression framework for checking the health of large HPC systems:** Thu. May 11th @ 2pm (Session C)



CSCS

Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

ETH zürich

XALT: Understanding HPC Usage via Job Level Collection

XALT/0.7.6

Motivation

- What software do researchers use (or do not use) on our supercomputer ?
- How successful are they in their efforts to use it ?
- <http://github.com/Fahey-McLay/xalt.git>

Design

- Xalt intercepts the user link (ld) and job launcher (srun) lines,
- maps program name & libraries to modulefile names,
- records complete list of environment vars (site configurable),
- stores the results and provides reporting tools
- <https://github.com/kpl-grwl/xalt-portal>

XALT MySQL (Compilers)

```
mysql> select link_id,link_program,date from xalt_link
```

link_id	link_program	date
699925	driver.cc	2017-05-07 00:00:46
699926	g++	2017-05-07 00:01:02
699927	g++	2017-05-07 00:01:04
699928	icpc	2017-05-07 00:04:28
699929	gcc	2017-05-07 00:05:47
699930	icpc	2017-05-07 00:09:15
699931	icc	2017-05-07 00:10:51
699932	g++	2017-05-07 00:11:34
699933	g++	2017-05-07 00:11:36
699934	g++	2017-05-07 00:12:46

```
mysql> select job_id,date,syshost,num_cores,num_nodes,num_threads,exec_path from xalt_run where date="2017-05-08" limit 5;
```

job_id	date	syshost	num_cores	num_nodes	num_threads	exec_path
1536396	2017-05-08 00:00:06	daint	120	10	12	/users/wuwei/opt/hfqmc_1.24dreal.exec
1536394	2017-05-08 00:01:13	daint	120	10	12	/users/wuwei/opt/cdmft_1.15.exec
1536394	2017-05-08 00:01:24	daint	120	10	12	/users/wuwei/opt/hfqmc_1.24dreal.exec
1536395	2017-05-08 00:01:40	daint	120	10	12	/users/wuwei/opt/cdmft_1.15.exec
1536395	2017-05-08 00:01:52	daint	120	10	12	/users/wuwei/opt/hfqmc_1.24dreal.exec

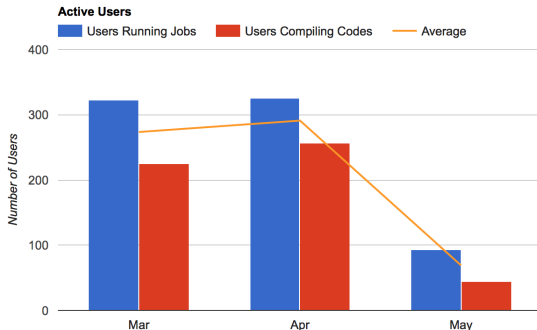
XALT portal (Users)

XALT User Environment Tracking and Problem Detection

Select syshost and date range for your queries.

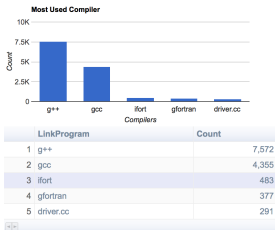
Select Syshost*

Active Users



XALT portal (Compilers)

Link Program Usage ▾



[Count = Number of executable linked to link_program]

Further Details ▾

List of User(s) (for given LinkProgram)

	Build Users	Earliest LinkDate	Latest LinkDate	Count
1	degomme	2017-05-05 11:38:08	2017-05-05 16:28:10	259
2	lucamar	2017-05-05 12:22:30	2017-05-05 23:59:58	72
3	asulrich	2017-05-05 07:55:54	2017-05-05 11:14:53	50
4	syzgants	2017-05-05 10:04:56	2017-05-05 15:46:52	30
5	jenscaca	2017-05-05 01:42:20	2017-05-05 23:47:18	16
6	albino	2017-05-05 14:09:20	2017-05-05 17:32:31	8
7	course01	2017-05-05 16:02:51	2017-05-05 16:04:53	7
8	ajajcj	2017-05-05 14:11:35	2017-05-05 18:33:40	6
9	merlo	2017-05-05 00:57:11	2017-05-05 20:10:25	5
10	pnayak	2017-05-05 13:40:43	2017-05-05 13:47:03	5

[Count = Number of Executable linked by the User]

XALT portal (Libs)

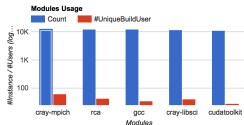
Usage Module and Compilers used over the given time

Select syshost and date range for your queries.

Select Syshost*

Number of Records*

Modules Usage



Modules	Count	#UniqueBuildUser
1 cray-mpich	13,039	62
2 rca	11,972	43
3 gcc	11,951	35
4 cray-libsci	11,528	41
5 cudatoolkit	10,987	28

Click Modules to get Version details

[Count = Number of time Object was Linked]

Further Details

List of Version(s) (for given Module)

Modules	Versions	Count	#UniqueBuildUser
1 cray-libsci	16.11.1	11,528	41

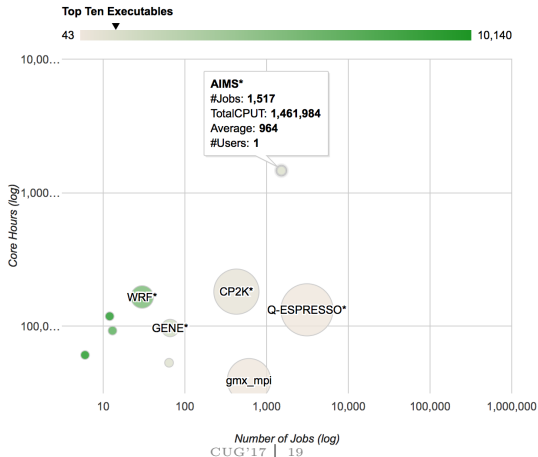
[Count = Number of time Object was Linked for given Module-Version]

XALT portal (Executables)

Select syshost and date range for your queries.

Select Syshost*

Top Ten Executables



Legend: radius=#users / left-right=#jobs / down-up=#cnh / white-green=#avgjobduration average=TotalCPUtime/#jobs

XALT reports (Modules)

```
./mostUsedModules.sh -s 2017-05-01 -e 2017-05-30
period = [ 2017-05-01 : 2017-05-30 ]
```

	MFile	%CNH	CNH	Jobs	Users
	AIMS*	35.37%	183389	351	3
	UNKNOWN	23.74%	123047	17927	152
	BIGDFT*	11.17%	57915	393	3
	CP2K*	9.81%	50866	2622	26
	Q-ESPRESSO*	4.79%	24840	31063	23
	GENE*	3.78%	19585	273	2
	VASP*	2.49%	12890	7242	16
	WRF*	1.77%	9167	600	13
	NAMD*	1.53%	7935	677	5
	GROMACS*	1.13%	5864	2141	15
	ECHAM*	1.10%	5688	174	2
	ENZO*	0.97%	5012	51	1
	LAMMPS*	0.88%	4544	678	10
	NVPROF*	0.76%	3959	142	5
	CPMD*	0.45%	2317	41	3
	PYTHON2*	0.14%	738	174	9
	RAMSES*	0.07%	380	203	10
	NCL*	0.04%	190	19	3
	PARAVIEW*	0.01%	49	83	5
	...				

	Total:	100.00%	518420	65719	326
request took 22 seconds					

```
mysql> select module_name as mf, ROUND(SUM(run_time*num_nodes)/3600) as CNH FROM xalt_run
+-----+-----+
| mf | CNH |
+-----+-----+
| NULL | 112797 |
| Amber/16-2016.11-CrayGNU-2016.11-cuda-8.0.54(daint-gpu) | 0 |
| Amber/16-2016.11-CrayGNU-2016.11-parallel(daint-mc) | 0 |
| CP2K/4.1-CrayGNU-2016.11(daint-mc) | 1836 |
+-----+-----+
```

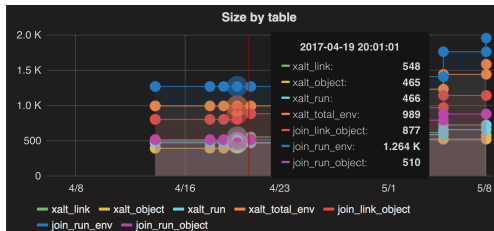
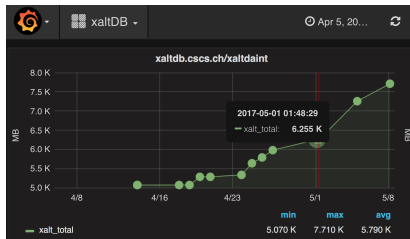
XALT @ CSCS

Past issues

- DB size growth (filtering)
- DB stability (lock issues, invalid column errors)
- WLM job informations (Slurm Jobid, OpenMP)
- Integrating xalt (RevMap) into CSCS workflow (easybuild)

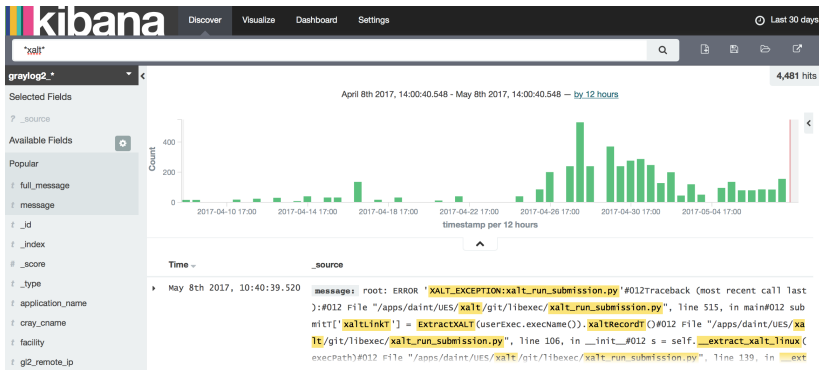


XALT: Database size



■ 21000 jobs per week = +1.4GB

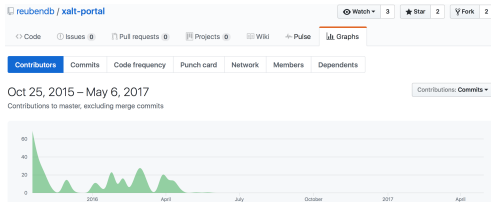
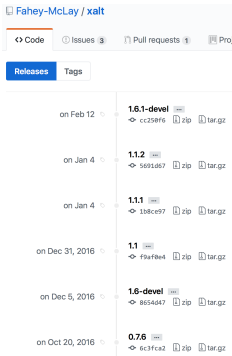
XALT: Database errors



XALT @ CSCS

On-going work

- Validating collected data
- Hunting down unknown users (xalt for all)
- Extending Reporting scripts and User portal (grafana ?)
- Upgrading from xalt/0.7.6 to latest version (what is yours ?)





CSCS

Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

ETH zürich

Thank you ! Questions ?
