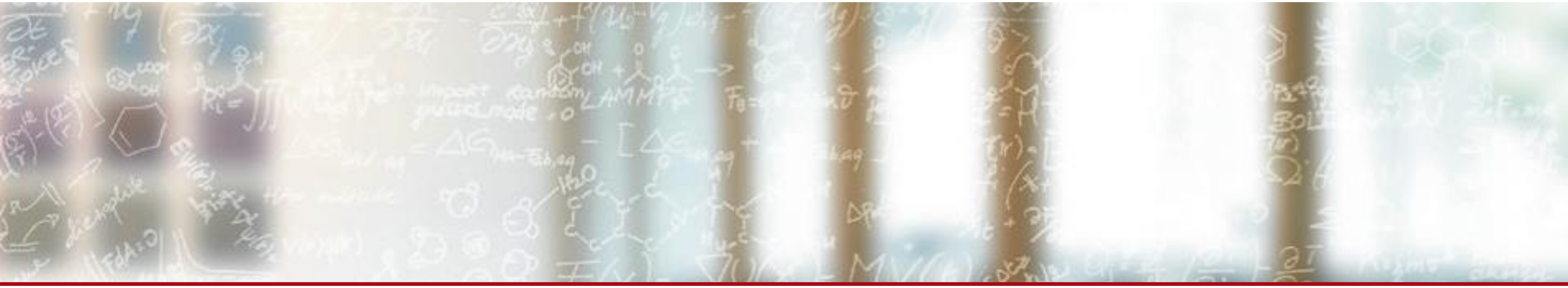
CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

ETH zürich



Piz Daint Upgrade

Webinar for CSCS User Community

Luca Marsella, CSCS

December 5th 2016

Outline of the Webinar

- Hybrid Cray **XC50** / XC40
  - Features of the upgraded system
  - Configuration of the SLURM scheduler
  - Enhancements in CLE 6.0 UP02
  - Documentation
- NVIDIA CUDA Toolkit
  - New features in v8.0
  - Performance Improvements
  - Documentation
- Cray Programming Environment
  - Cray PE November 2016
  - Changes to default modules
  - Easybuild Framework @ CSCS

CSCS office building in Lugano

Hybrid Cray XC50 / XC40

System Specifications

Model	Cray XC50/XC40
XC50 Compute Nodes	Intel® Xeon® E5-2690 v3 @ 2.60GHz (12 cores, 64GB RAM) and NVIDIA® Tesla® P100 16GB
XC40 Compute Nodes	Intel® Xeon® E5-2695 v4 @ 2.10GHz (18 cores, 64/128 GB RAM)
Login Nodes	Intel® Xeon® CPU E5-2650 v3 @ 2.30GHz (10 cores, 256 GB RAM)
Interconnect Configuration	Aries routing and communications ASIC, and Dragonfly network topology
Scratch capacity	Piz Daint scratch: 6.2 PB

File Systems
The $SCRATCH space, /scratch/snx3000/$USER, is connected via an Infiniband interconnect.
The shared storage under /project and /store is available from the login nodes only!

Filesystems features

	/scratch (Piz Daint)	/scratch (Clusters)	/users	/project	/store
Type	Lustre	GPFS	GPFS	GPFS	GPFS
Quota	Soft quota 1 M files	None	10 GB/user 100 K files	5 TB/group 250 K files	As per contract
Expiration	30 days	30 days	None	End of the project	As per contract
Data Backup	None	None	Active	Active	Active
Access Speed	Fast	Fast	Slow	Medium	Slow
Capacity	6.2 PB	1.4 PB	86 TB	5.7 PB	4.4 PB

Soft quotas:
The $SCRATCH space /scratch/snx3000/$USER has a **soft** quota set to prevent any excessive load. Users exceeding the soft quota will be **warned at submit time** and **will not be able to submit** new jobs.

Data transfer

- CSCS provides a data transfer service to get your files from/to CSCS file systems and a SLURM queue **xfer** available on Piz Daint to transfer files internally
- The SLURM queue **xfer** can submit jobs directly on the nodes of the cluster that supports the data transfer and can access most filesystems with copy commands
- The **xfer** usage **will not be charged** against your project allocation
- A set of jobs with dependencies can be started using the template **stage.sbatch**

```
$ sbatch --job-name=stage_in stage.sbatch ${PROJECT}/<source> ${SCRATCH}/<destination> production.sbatch
```

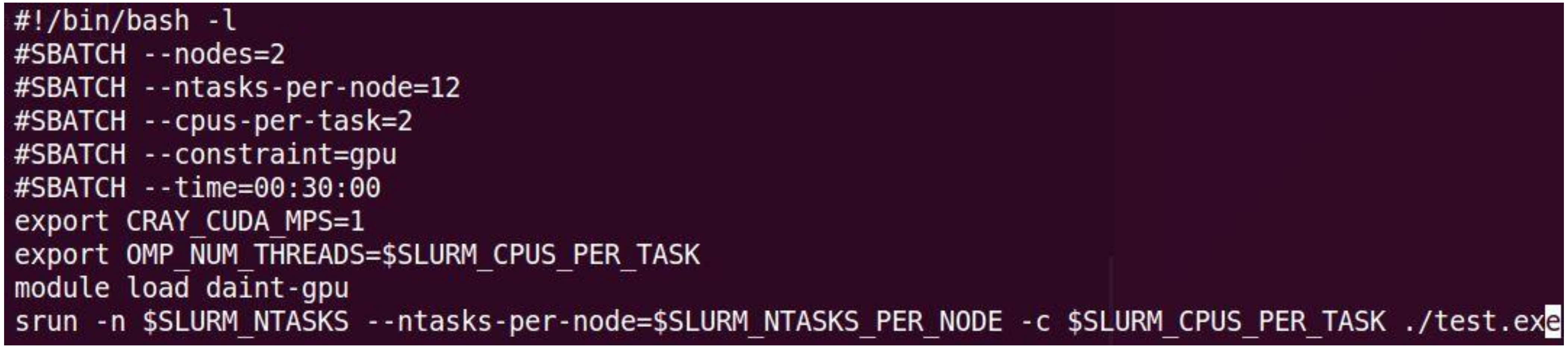
- For more information have a look at **Internal Data Transfers** on the User Portal

SLURM batch queues

Name of the queue	Max time	Max nodes	Brief Description
debug	30min	4	Quick turnaround for test jobs
low	6 h	2400	For use only when allocations are exhausted
normal	24 h	2400	Standard queue for production work
high	24 h	2400	High priority queue, time is charged double
large	12 h	4400	Large scale work, by arrangement only
prepost	30min	1	High priority pre/post processing
xfer	30min	1	Data transfer queue
total	2 h		CSCS maintenance queue (restricted use)

SLURM Scheduler Configuration

- Piz Daint uses SLURM for the submission, monitoring and control of parallel jobs
- Parallel programs compiled with cray-MPICH must be run using the **srun** command
- SLURM batch scripts need to be submitted with the **sbatch** command from the $SCRATCH folder: users are NOT supposed to run jobs from different filesystems due to the low performance
- The SLURM option **--constraint=gpu** makes sure that the SLURM scheduler will allocate nodes with GPU devices and will automatically set the option **--gres=gpu:1** to allocate the GPU device
- The module **daint-gpu** targets the XC50 architecture with craype-haswell and updates MODULEPATH



```
#!/bin/bash -l
#SBATCH --nodes=2
#SBATCH --ntasks-per-node=12
#SBATCH --cpus-per-task=2
#SBATCH --constraint=gpu
#SBATCH --time=00:30:00
export CRAY_CUDA_MPS=1
export OMP_NUM_THREADS=$SLURM_CPUS_PER_TASK
module load daint-gpu
srun -n $SLURM_NTASKS --ntasks-per-node=$SLURM_NTASKS_PER_NODE -c $SLURM_CPUS_PER_TASK ./test.exe
```

Cray Linux Environment 6.0 UP02

- Cray Linux Environment (CLE) is the operating system on Cray systems
- CLE 6.0 UP02 is based on the Novell SLES 12 base operating system
- CLE 6.0 UP02 software release is available on the upgraded Piz Daint
- Documentation: http://docs.cray.com/PDF/Whats_New_for_CLE_6.0_and_SMW_8.0_CLE60UP02_S-2573.pdf

Compatibilities and Differences

- Applications and binaries compiled for XC30 architecture **must be recompiled**

- Binaries compiled for the multicore XC40 architecture on earlier releases of CLE might run on CLE 6.0UP02 provided that the binaries are **dynamically linked**

- Statically linked binaries using directly or indirectly the network interface libraries (uGNI/DMAPP) **must be recompiled**:
  - This includes **applications using MPI or SHMEM libraries**, as well as the PGAS (Partitioned Global Address Space) languages such as **UPC, Fortran with Coarrays, and Chapel**
  - DMAPP (Distributed Shared Memory Application) and uGNI (user Generic Network Interface) are tied to specific kernel versions and no backward or forward compatibility is provided

Documentation

- Cray provides books and man pages that can be accessed in the following ways:
  - **CrayDoc** is the Cray documentation delivery system, enabling quick access and search of Cray books, man pages, and third-party documentation using HTML and PDF formats:
    - CrayDoc public website: http://docs.cray.com
  - **Man pages** are textual help files available from the command line on Cray machines. To access man pages, enter the **man** command followed by the name of the man page. For more information about man pages, see the man(1) man page by entering “**man man”** on the shell

CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

ETH zürich

NVIDIA CUDA Toolkit

Upgrade to NVIDIA CUDA Toolkit v8.0

- It features a comprehensive development environment to build **GPU-accelerated applications**
- It includes **compiler** for NVIDIA GPUs, **math libraries** and tools for debugging and optimizing application performance
- It provides **programming guides**, user manuals, API reference and **online documentation** to get started quickly
- NVIDIA developer portal:

  https://developer.nvidia.com/cuda-zone

NVIDIA Tesla P100 GPU Accelerator

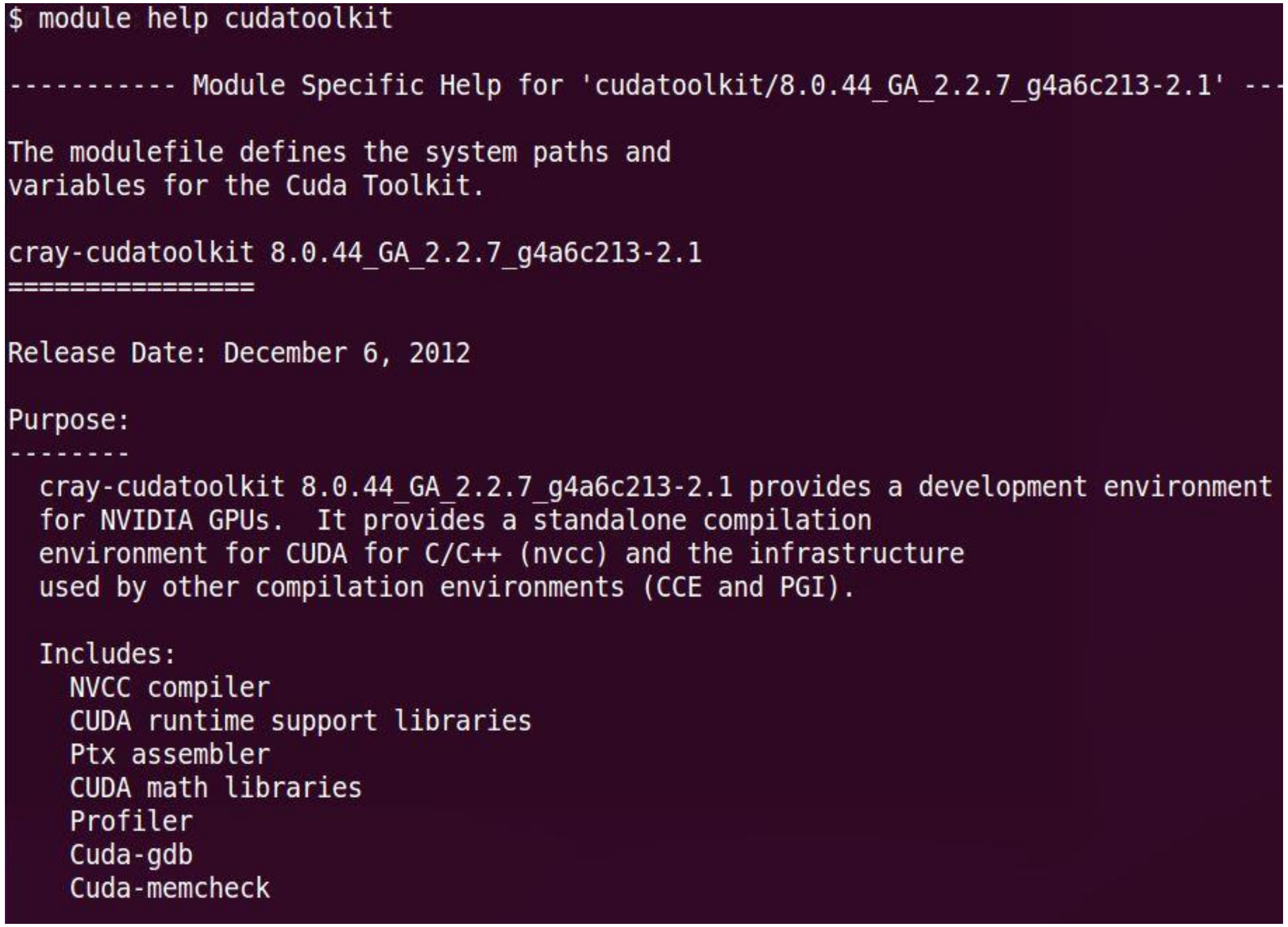
New Features Highlights in CUDA Toolkit v8.0

- General CUDA
  - you need to target the Tesla P100 architecture **sm_60** with NVCC gpu architecture flags
  - adds support for GPUDirect Async, improving application throughput
- CUDA Tools
  - CUDA compilers: Intel C++ Compilers 16.0 and 15.0.4 are now supported
  - CUDA profiler provides CPU profiling to identify hot-spot regions in the code
- CUDA Libraries
  - new built-in for fp64 atomicAdd() that cannot be overridden with a custom user function
  - nvGRAPH, a new library that is a collection of routines to process graph problems on GPUs

- Features and release notes of CUDA Toolkit v8.0 and Pascal GPU Architecture
  - https://devblogs.nvidia.com/parallelforall/cuda-8-features-revealed
  - http://docs.nvidia.com/cuda/cuda-toolkit-release-notes
  - https://developer.nvidia.com/pascal

Documentation

- NVIDIA Documentation Portal
  - http://docs.nvidia.com/
- CUDA Toolkit for Developers
  - https://developer.nvidia.com/cuda-toolkit
- System located documentation
  - **module help cudatoolkit**
  - NVIDIA compiler
    - **nvcc --help**
  - CUDA debugger
    - **cuda-gdb --help**



```
$ module help cudatoolkit

----------- Module Specific Help for 'cudatoolkit/8.0.44_GA_2.2.7_g4a6c213-2.1' ---

The modulefile defines the system paths and
variables for the Cuda Toolkit.

cray-cudatoolkit 8.0.44_GA_2.2.7_g4a6c213-2.1
================

Release Date: December 6, 2012

Purpose:
--------
  cray-cudatoolkit 8.0.44_GA_2.2.7_g4a6c213-2.1 provides a development environment
  for NVIDIA GPUs.  It provides a standalone compilation
  environment for CUDA for C/C++ (nvcc) and the infrastructure
  used by other compilation environments (CCE and PGI).

  Includes:
    NVCC compiler
    CUDA runtime support libraries
    Ptx assembler
    CUDA math libraries
    Profiler
    Cuda-gdb
    Cuda-memcheck
```

CSCS

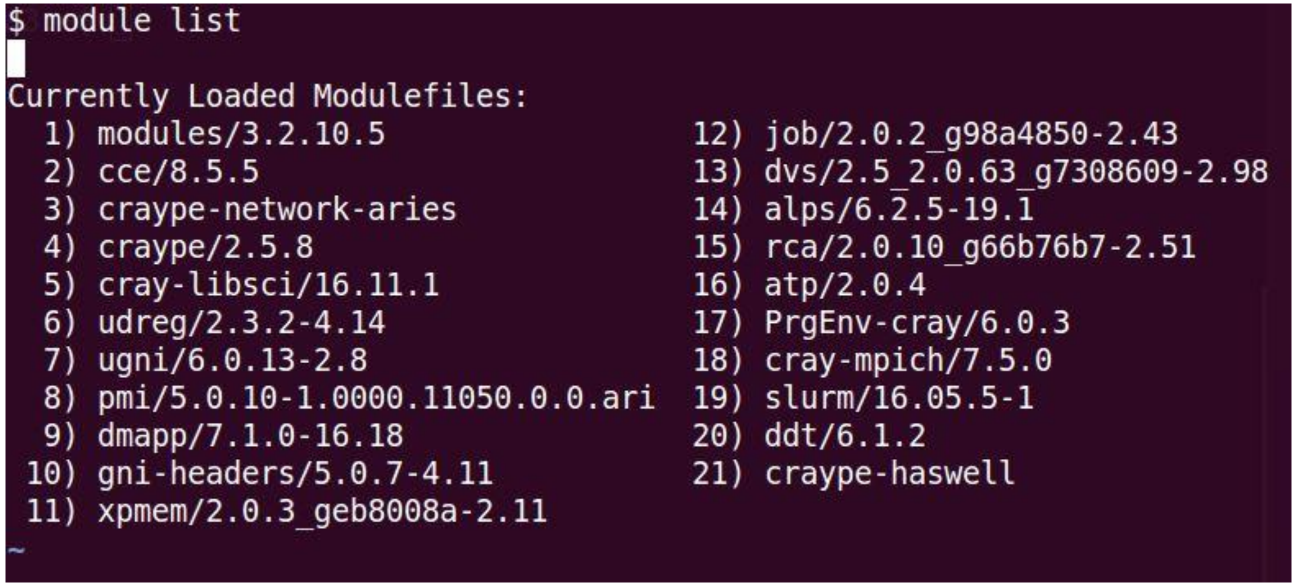
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

ETH zürich

Cray Programming Environment

The Cray Programming Environment on the hybrid Piz Daint

- Released on a monthly basis, it uses the modules framework for library path management
  - The environment contains a set of libraries for each supported compiler (e.g.: **PrgEnv-cray**):



```
$ module list

Currently Loaded Modulefiles:
  1) modules/3.2.10.5                    12) job/2.0.2_g98a4850-2.43
  2) cce/8.5.5                           13) dvs/2.5_2.0.63_g7308609-2.98
  3) craype-network-aries                14) alps/6.2.5-19.1
  4) craype/2.5.8                        15) rca/2.0.10_g66b76b7-2.51
  5) cray-libsci/16.11.1                 16) atp/2.0.4
  6) udreg/2.3.2-4.14                    17) PrgEnv-cray/6.0.3
  7) ugni/6.0.13-2.8                     18) cray-mpich/7.5.0
  8) pmi/5.0.10-1.0000.11050.0.0.ari     19) slurm/16.05.5-1
  9) dmapp/7.1.0-16.18                   20) ddt/6.1.2
 10) gni-headers/5.0.7-4.11              21) craype-haswell
 11) xpmem/2.0.3_geb8008a-2.11
~
```

  - The default target architecture is the XC50 with Intel Haswell processors: craype-haswell
  - Users can change the target architecture by loading one of the following modules:
    - **daint-gpu** it targets the XC50 architecture (Intel Haswell and P100 Tesla GPUS)
    - **daint-mc** it targets the XC40 architecture (Intel Broadwell multicore)
  - The modules above will update the MODULEPATH: use the module unload command to change environment!

Upgrade of the Cray XC Programming Environment

- The Cray XC PE 16.11 consists of the **Cray Developer Toolkit - CDT 16.11**
- The following products have been updated within this release:
  - **Compiling Environment - CCE 8.5.5**
    - CCE 8.5.5
  - **Cray Message Passing Toolkit - MPT 7.5.0**
    - MPT 7.5.0
  - **Cray Debugging Support Tools - CDST 16.11**
    - ATP 2.0.4
  - **Cray Performance Measurement & Analysis Tools - CPMAT 6.4.3**
    - Perftools 6.4.3
    - PAPI 5.5.0.1
  - **Cray Scientific and Math Libraries - CSML 16.11**
    - LibSci 16.11.1
    - LibSci_ACC 16.11.1
  - **Cray Environment Setup and Compiling support - CENV 16.11**
    - craype-installer 1.20.0
    - craype 2.5.8
  - **GCC** default modulefile addressing CUDA 8.0 dependencies is set to version **5.3.0**

New default modules for compilers, libraries and tools

- Compilers
  - cce/8.5.5
  - gcc/5.3.0
  - intel/17.0.0.098
  - pgi/16.9.0
- Communication Libraries
  - cray-ga/5.3.0.7
  - cray-mpich/7.5.0
  - cray-shmem/7.5.0
- Numerical Libraries
  - cray-libsci/16.11.1
  - cray-libsci_acc/16.11.1
  - fftw/3.3.4.10
  - cray-tpsl/16.07.1
  - cray-trilinos/12.6.3.3
- Performance tools
  - perftools/6.4.3
  - perftools-base/6.4.3
  - perftools-lite/6.4.3
  - papi/5.5.0.1
- I/O Libraries
  - cray-hdf5/1.10.0
  - cray-netcdf/4.4.1
  - cray-hdf5-parallel/1.10.0
  - cray-netcdf-hdf5parallel/4.4.1
- Debuggers
  - ddt/6.1.2
  - cray-lgdb/3.0.4

New default modules for scientific applications and libraries

daint-gpu

- Amber/16-2016.11-CrayGNU-2016.11-cuda-8.0
- Boost/1.61.0-CrayGNU-2016.11-Python-2.7.12
- CDO/1.7.2-CrayGNU-2016.11
- CP2K/4.1-CrayGNU-2016.11-cuda-8.0
- CPMD/4.1-CrayIntel-2016.11
- GROMACS/5.1.4-CrayGNU-2016.11-cuda-8.0
- GSL/2.1-CrayGNU-2016.11
- LAMMPS/30Jul16-CrayGNU-2016.11-cuda-8.0
- magma/2.2.0-CrayGNU-2016.11-cuda-8.0
- NAMD/2.11-CrayIntel-2016.11-cuda-8.0
- NCL/6.3.0
- NCO/4.6.0-CrayGNU-2016.11
- ncview/2.1.7-CrayGNU-2016.11
- QuantumESPRESSO/5.4.0-CrayIntel-2016.11-cuda-8.0
- R/3.3.1-CrayGNU-2016.11
- VASP/5.4.1-CrayIntel-2016.11-cuda-8.0

daint-mc

- Amber/16-2016.11-CrayGNU-2016.11-parallel
- Boost/1.61.0-CrayGNU-2016.11-Python-2.7.12
- CDO/1.7.2-CrayGNU-2016.11
- CP2K/4.1-CrayGNU-2016.11
- CPMD/4.1-CrayIntel-2016.11
- GROMACS/5.1.4-CrayGNU-2016.11
- GSL/2.1-CrayGNU-2016.11
- LAMMPS/30Jul16-CrayGNU-2016.11
- NAMD/2.11-CrayIntel-2016.11
- NCL/6.3.0
- NCO/4.6.0-CrayGNU-2016.11
- ncview/2.1.7-CrayGNU-2016.11
- QuantumESPRESSO/5.4.0-CrayIntel-2016.11
- R/3.3.1-CrayGNU-2016.11
- VASP/5.4.1-CrayIntel-2016.11

EasyBuild Framework @ CSCS

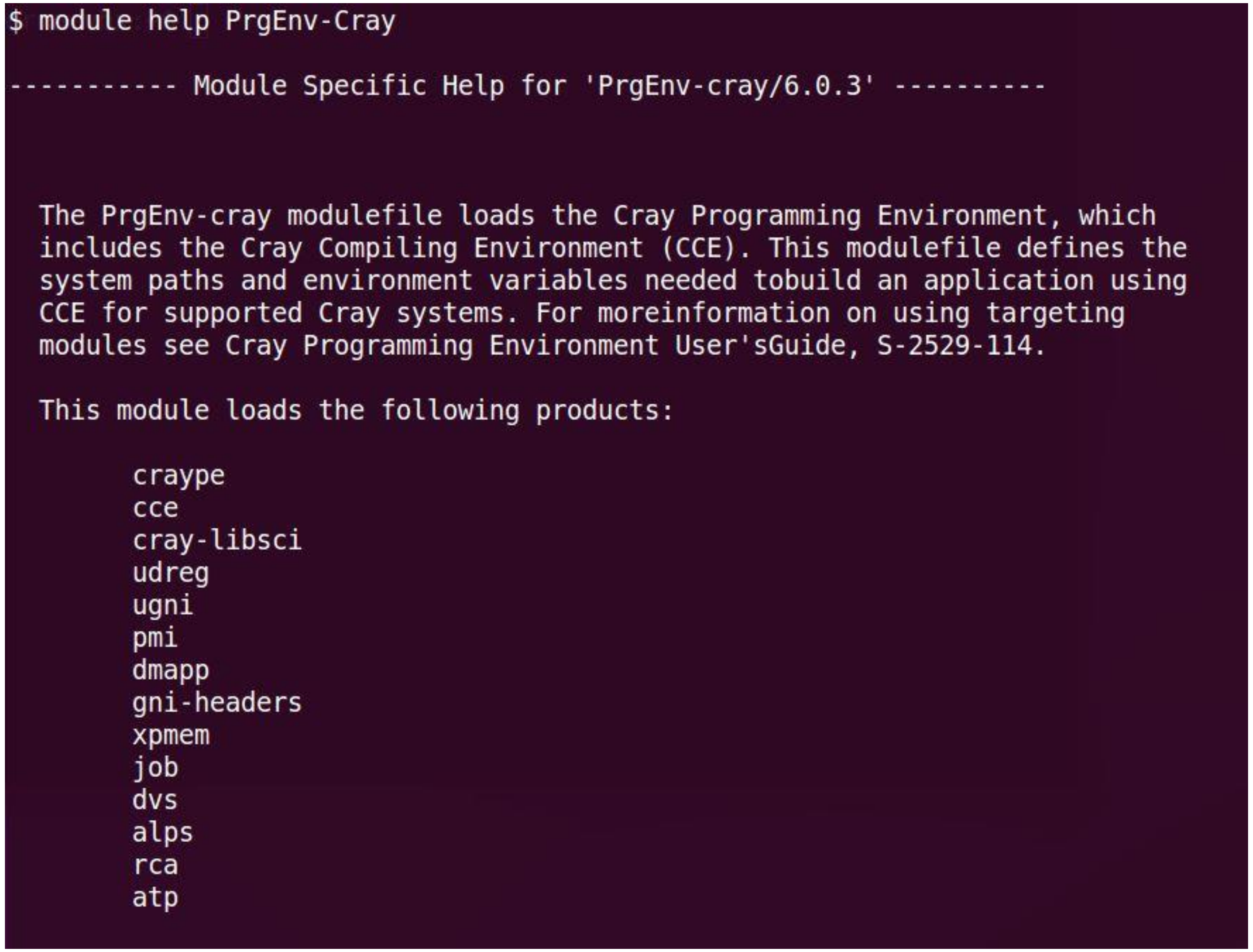
- EasyBuild is available through the module **EasyBuild-custom**. This module defines the location of the configuration files, the recipes that we provide and the install path of the software stack:
  - **$ module load EasyBuild-custom**
- On the upgraded Piz Daint you need to select which architecture should be targeted when building software. For instance you need to load the following to target the XC50 with GPUs:
  - **$ module load daint-gpu EasyBuild-custom**
- On Piz Daint, the EasyBuild software and modules will be installed by default on:
  - **$HOME/easybuild/daint/<haswell|broadwell>**
- You can override the default installation folder (EASYBUILD_PREFIX) and the default CSCS repository folder (EB_CUSTOM_REPOSITORY) by exporting the following variables:
  - **$ export EASYBUILD_PREFIX=/your/preferred/installation/folder**
  - **$ export EB_CUSTOM_REPOSITORY=/your/cscs/repository/folder**
  - **$ module load EasyBuild-custom**
- How to build a program resolving dependencies automatically:
  - **$ eb <name_version>.eb -r**

Documentation

- Manuals and User’s Guides on Cray PE are addressed by **CrayDoc**, **man** or **module help**

- Further details can be retrieved selecting specific modules of the Cray PE with **module help**:
  - **module help cce**

- The CSCS User Portal at http://user.cscs.ch gives basic information on how to compile your code on Cray systems under **Compiling Your Code**



```
$ module help PrgEnv-Cray

----------- Module Specific Help for 'PrgEnv-cray/6.0.3' ----------



  The PrgEnv-cray modulefile loads the Cray Programming Environment, which
  includes the Cray Compiling Environment (CCE). This modulefile defines the
  system paths and environment variables needed tobuild an application using
  CCE for supported Cray systems. For moreinformation on using targeting
  modules see Cray Programming Environment User'sGuide, S-2529-114.

  This module loads the following products:

        craype
        cce
        cray-libsci
        udreg
        ugni
        pmi
        dmapp
        gni-headers
        xpmem
        job
        dvs
        alps
        rca
        atp
```

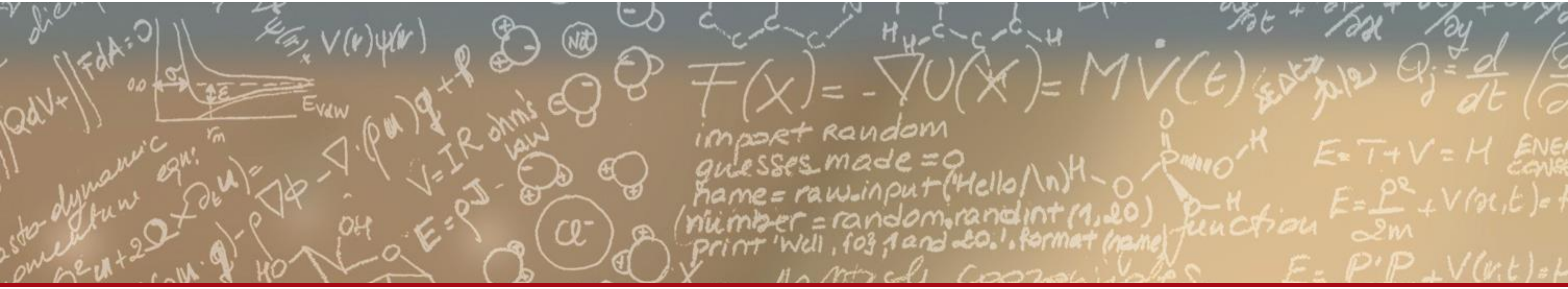
Further information

- CSCS User Portal:
  - http://user.cscs.ch
- Cray Documentation:
  - http://docs.cray.com
- NVIDIA Documentation:
  - http://docs.nvidia.com
- Contact us:
  - help@cscs.ch

Piz Daint in the machine room at CSCS

CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

ETH zürich



Thank you for your kind attention