Student HPC Hackathon 8/2018

J. Simon, C. Plessl

22. + 23. August 2018

Student HPC Hackathon 8/2018

- Get the most performance out of our new HPC system Noctua1
- Performance is measured with given Benchmark programs (+ your favorite App)
- Form groups of about max. two individuals
- HPC experts from PC² will support you
- Agenda
  - 1st day starts with
    - an introduction to the Noctua1 system, and
    - an overview of the Benchmark programs and running rules
  - Do your performance optimizations/measurements
  - 2nd day ends with
    - a presentation of the results of each group
- Date
  - August, 22nd + 23rd, 10:00-16:00 / 9:00-16:00
  - Room O2.267

Student HPC Hackathon 8/2018

Challenges

- New Hardware
  - Processor architecture (cache hierarchy, AVX512 instructions,..)
  - Communication network (OmniPath 100Gbps)
  - Storage System
- New Software
  - Programming Tools
  - Libraries
  - Resource management system “SLURM”
  - Lustre file system
- Planning your batch jobs
  - Optimal use of available resources

Noctua1

Frontside: Cold air intake

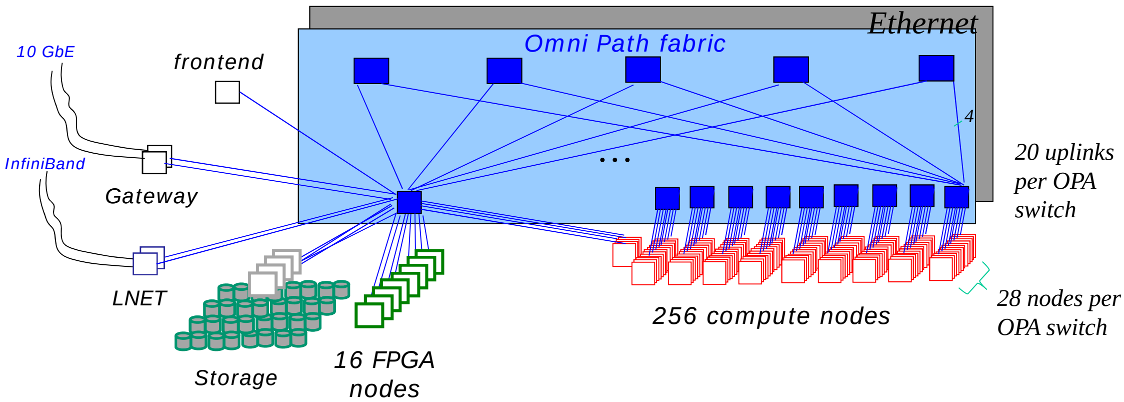
Backside: Cooled backdoors

Noctua1: Cray CS 500 Storm

- 256 compute nodes
  - 2x Intel Xeon Gold 6148, each 20C, 2.4 GHz
  - 192 GiB
- 16 FPGA nodes
  - Intel Xeon 6148+6148F, 192 GiB
  - each with 2 Nallatech Stratix 10
- Parallel file system
  - Lustre
  - 720 TB disk capacity
- Interconnect Intel Omni-Path
  - 100 Gbit/s network
  - Blocking faktor 1:1,4

Paderborn Center for Parallel Computing

Noctua1: Architecture



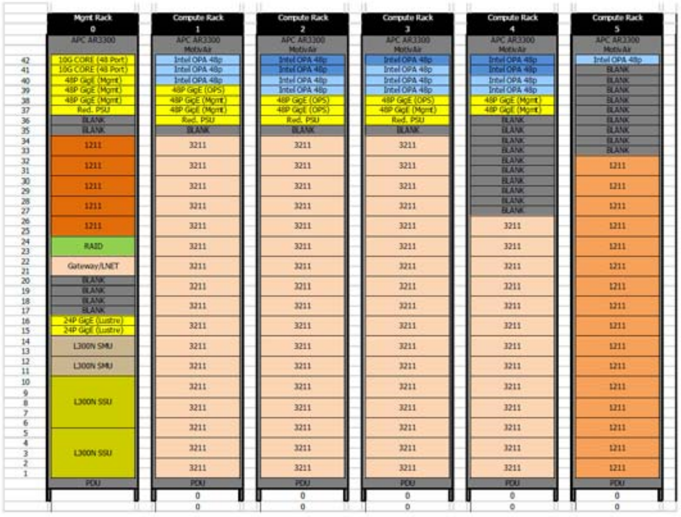
Paderborn Center for Parallel Computing

Noctua1: Rack Layout

5x water cooled rack

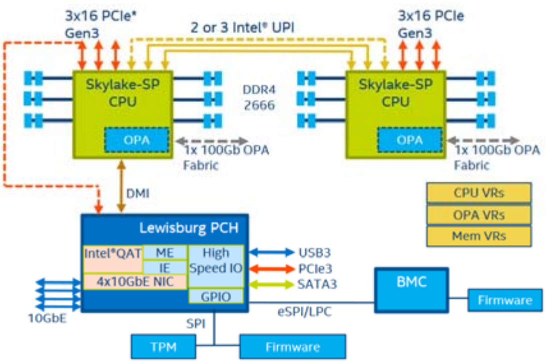
1x air cooled rack

~ 161 kW



Paderborn Center for Parallel Computing

Intel Xeon Scalable Processor

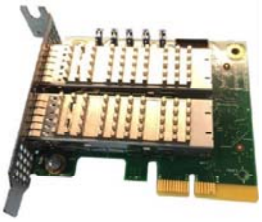


BMC: Baseboard Management Controller	PCH: Intel® Platform Controller Hub	IE: Innovation Engine
Intel® OPA: Intel® Omni-Path Architecture	Intel QAT: Intel® QuickAssist Technology	ME: Manageability Engine
NIC: Network Interface Controller	VMD: Volume Management Device	NTB: Non-Transparent Bridge

Feature	Details
Socket	Socket P
Scalability	2S, 4S, 8S, and >8S (with node controller support)
CPU TDP	70W – 205W
Chipset	Intel® C620 Series (code name Lewisburg)
Networking	Intel® Omni-Path Fabric (integrated or discrete) 4x10GbE (integrated w/ chipset) 100G/40G/25G discrete options
Compression and Crypto Acceleration	Intel® QuickAssist Technology to support 100Gb/s comp/decomp/crypto 100K RSA2K public key
Storage	Integrated QuickData Technology, VMD, and NTB Intel® Optane™ SSD, Intel® 3D-NAND NVMe & SATA SSD
Security	CPU enhancements (MBE, PPK, MPX) Manageability Engine Intel® Platform Trust Technology Intel® Key Protection Technology
Manageability	Innovation Engine (IE) Intel® Node Manager Intel® Datacenter Manager

Paderborn Center for Parallel Computing

OmniPath – Integrated Fabric



Intel Fabric through carrier card

Intel Fabric passive cable

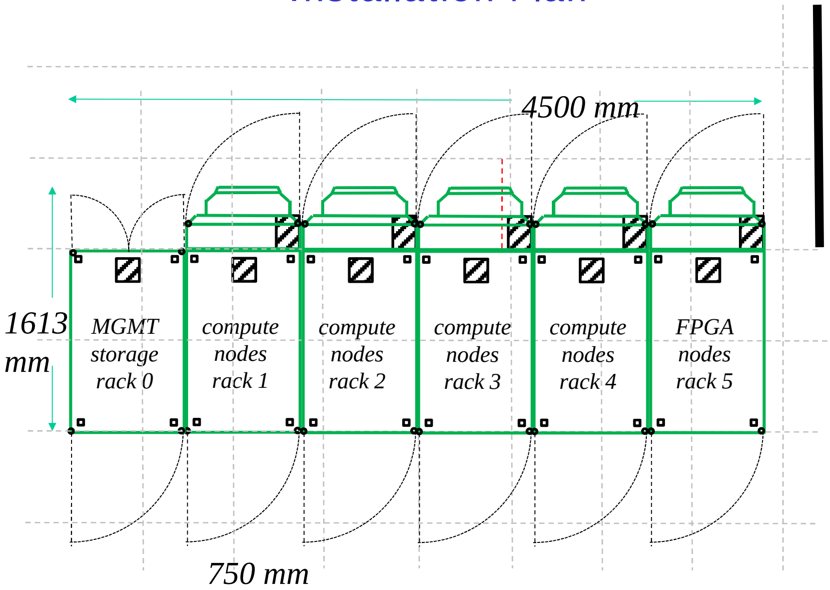
CPU with integrated Fabric

Noctua1 vs. OCuLUS

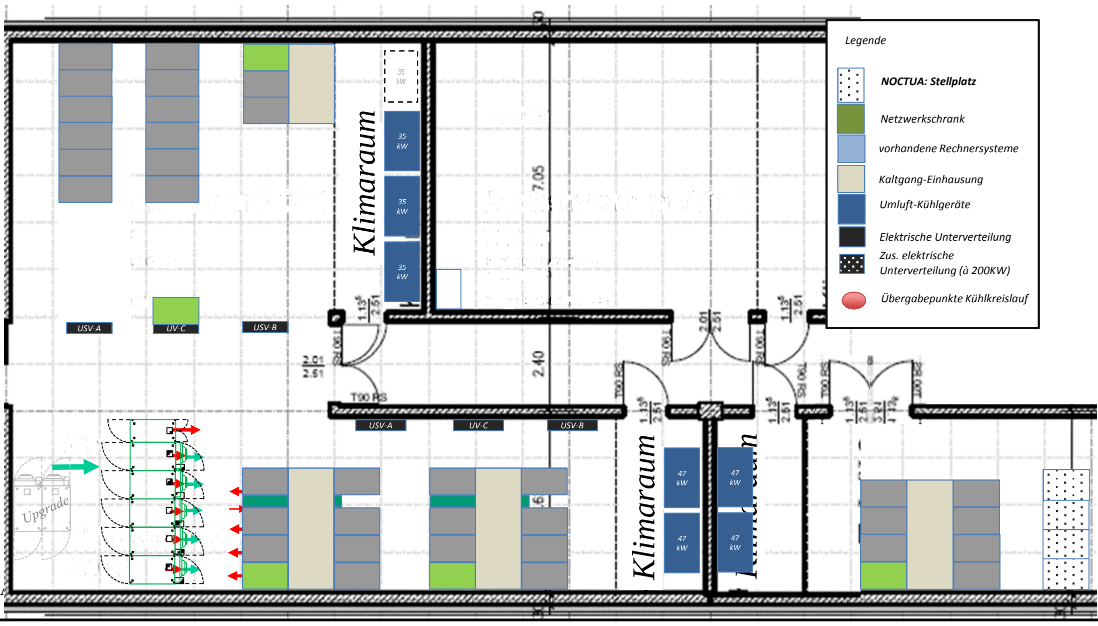
	Noctua1	OCuLUS	Diff.
# nodes	272	616	- 56%
# cores per node	2x20	2x8	x 2.5
Total # cores	10.880	9.856	+ 10%
Memory per node [GiB]	192	64	x 3
Total memory [TiB]	52,2	41,2	+ 27%
HP-Linpack [TFlop/s]	> 535	188,7	x 2.8
Accu. STREAM [GiB/s]	50.150	43.700	+ 15%
Accu. SpecFP 2006 base	381.000	295.700	+ 29%
Accu. SpecINT 2006 base	516.800	381.900	+ 35%
Scaled Spec MPI	> 560	?	
MPI latency [µs]	<1,34	2,1	- 36%
MPI bandwidth [GiB/s]	24,5	7	x 3.5

	Noctua 1	OCuLU S	Diff.
Storage Capacity [TB]	720	500	+ 44%
Storage bandwidth [GiB/s]	20	25	- 20%
MPI network blocking fact.	OPA 1:1,4	IB 1:2	
MPI network [Gbps]	100	40	x 2.5
Power consumption[kW]	164	230	- 29%

Installation Plan



Floorspace of DataCenter Building-O



PC2 Environment

- IMT Account
  - Member of group **HPC-LCO-SIMON**
  - **fileservice** activated (IMT serviceportal)
- Login server and Noctua1 frontend
  - fe-1....
- Standard Environment Settings
  - PC2FS
    - $PC2SW — Software installed by PC2
    - $PC2DATA/HPC-LCO-SIMON — shared data of group HPC-LCO-SIMON
    - $PC2SCRATCH/HPC-LCO-SIMON/<user> — shared filesystem OCuLUS / Noctua1
    - $PC2PFS/HPC-LCO-SIMON — Lustre parallel filesystem (use individual dirs)
    - $PC2SYSNAME — Noctua
  - SLURM
    - Version 17.11.8

Program Development Environments

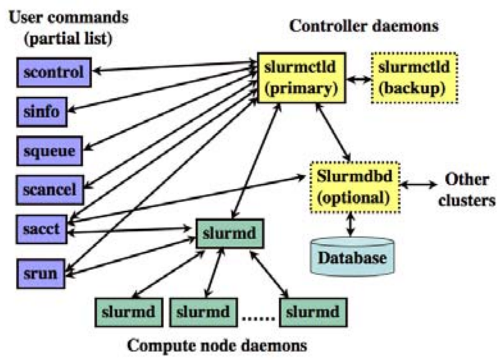
- Intel Parallel Studio Cluster Edition (modulefile intel/18.0.3)
  - C/C++ and Fortran
  - Python
  - Intel Math Kernel Libraries (MKL)
  - Intel MPI
  - Intel Data Analytics Acceleration Library, Integrated Performance Primitives, Threading Building Blocks
  - Intel VTune, Advisor, Inspector, Trace Analyzer and Collector
- Cray Compiler Environment “CCE”
  - C/C++ and Fortran 2008
  - OpenMP 4.1, MPI 2.2, UPC 1.2, OpenACC 2.0
  - LibSci, LibSci_ACC
- Cray Performance Measurement, Analysis, and Porting Tools
  - Performance and Analysis Tool CrayPAT
  - Visualization Tool Cray Apprentice2
  - Porting Tool Cray Reveal

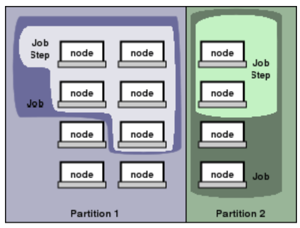
Cray Performance Tools

- CrayPAT profiles executables
  - Timing and hardware performance counter measurements
  - Collect and show program top time consumers and bottlenecks
  - Automatic generation of observations and suggestions
  - Data collection and presentation of computation, communication, I/O, and memory statistics
  - CrayPAT lite is a simplified, easy-to-use version of CrayPAT
- Visualization of performance data with Cray Apprentice2
  - Reports and graphical formats
  - GUI
  - Runs on Windows, MacOS, and Linux using the platform-independent data files
- Code-restructuring assistant Reveal
  - Helps developers to add additional levels of parallelism
  - Assists with parallelizing more complicated loops
  - Combining performance statistics and program source code

SLURM

- User commands
  - sacct, salloc, sattach, sbatch, sbcast, scancel, scontrol, sinfo, smap, squeue, srun, strigger, sview
- Managed entities
  - Jobs (allocation of resource assigned to a user for a specified amount of time)
  - Job steps (sets of parallel tasks within a job)
  - Resources are nodes, processors, memory, …
  - Nodes are logically organized into (possibly overlapping) partitions





SLURM Quick Start

- Information about the system

> sinfo

PARTITION	AVAIL	TIMELIMIT	NODES	STATE	NODELIST
compute*	up	infinite	16	alloc	cn-[0001-0016]
compute*	up	infinite	240	idle	cn-[0017-0256]
fpga	up	infinite	16	idle	fpga-[0001-0016]
all	up	infinite	16	alloc	fpga-[0001-0016]
all	up	infinite	256	idle	cn-[0017-0256],fpga-[0001-0016]

- State of jobs

> squeue

JOBID	PARTITION	NAME	USER	ST	TIME	NODES	NODELIST(REASON)
11618	compute	my.scrip	jens	PD	0:00	1	(Resources)

- Accounting information about active and completed jobs

> sacct

JOBID	JobName	Partition	Account	AllocCPUs	State	ExitCode
11617	IMB-MPI1	compute		120	COMPLETED	0:0
11618	my.scrip	compute			PENDING	

SLURM Quick Start (2)

- More information about nodes, partitions, jobs, job steps, configurations

> scontrol show partitions

```
PartitionName=compute
   AllowGroups=ALL AllowAccounts=ALL AllowQos=ALL
   AllocNodes=ALL Default=YES QoS=N/A
   DefaultTime=NONE DisableRootJobs=NO ExclusiveUser=NO GraceTime=0 Hidden=NO
   MaxNodes=UNLIMITED MaxTime=UNLIMITED MinNodes=1 LLN=NO MaxCPUsPerNode=UNLIMITED
   Nodes=cn-[0001-0256]
   PriorityJobFactor=1 PriorityTier=1 RootOnly=NO ReqResv=NO OverSubscribe=NO
   OverTimeLimit=NONE PreemptMode=OFF
   State=UP TotalCPUs=10240 TotalNodes=256 SelectTypeParameters=NONE
   DefMemPerNode=UNLIMITED MaxMemPerNode=UNLIMITED
...
```

- Resource allocation and launch the tasks for a job step in a single command

> srun –N3 –l /bin/hostname

```
cn-0019
cn-0018
cn-0017
```

SLURM Quick Start (3)

- Submit a script for later execution

```
> cat my.script
#!/bin/bash
#SBATCH -- time=1
/bin/hostname
srun –l /bin/hostname
srun –l /bin/pwd
```

```
> sbatch –N4 –o my.stdout my.script
sbatch: Submitted batch job 1234

> cat my.stdout
cn-0001
0: cn-0001
2: cn-0003
3: cn-0004
1: cn-0002
2: /upb/departments/pc2/user/j/jens/Tests/SLURM
0: /upb/departments/pc2/user/j/jens/Tests/SLURM
1: /upb/departments/pc2/user/j/jens/Tests/SLURM
3: /upb/departments/pc2/user/j/jens/Tests/SLURM
```


SLURM Quick Start (4)

- MPI batch script

```
#!/bin/bash
# Example with 80MPI tasks and 40tasks per node.
#
# Project/Account (use your own)
#SBATCH -A hpc2n-1234-56
#
# Number of MPI tasks
#SBATCH -n 80
#
# Number of tasks per node
#SBATCH --tasks-per-node=40
#
# Runtime of this jobs is less then 1 hours.
#SBATCH --time=1:00:00

module load intel/18.0.3
srun ./mpi_program

# End of submit file
```

SLURM Quick Start (5)

- Resource allocation and spawn job steps within that allocation

```
> salloc –N4 bash
$ sbcast a.out /tmp/pi
Granted job allocation 1234
$ srun /tmp/pi
Result is 3.14159
$ srun rm /tmp/pi
$ exit
salloc: Relinquishing job allocation 1234
```