Introduction to HPC2N, Kebnekaise and HPC

Birgitte Brydsö,
Pedro Ojeda May, and others at HPC2N

HPC2N
Umeå University

19. January 2022

HPC2N (HPC2N at a glance)

- **High Performance Computing Center North (HPC2N)** is a national center for Scientific and Parallel Computing

HPC2N (HPC2N at a glance)

- **High Performance Computing Center North (HPC2N)** is a national center for Scientific and Parallel Computing

- A part of **Swedish National Infrastructure for Computing (SNIC)**

HPC2N (HPC2N at a glance)

Provides state-of-the-art resources and expertise:

- Scalable and parallel **HPC**

HPC2N (HPC2N at a glance)

Provides state-of-the-art resources and expertise:

- Scalable and parallel **HPC**
- Large-scale **storage facilities** (Project storage (Lustre), SweStore, Tape)

HPC2N (HPC2N at a glance)

Provides state-of-the-art resources and expertise:

- Scalable and parallel **HPC**
- Large-scale **storage facilities** (Project storage (Lustre), SweStore, Tape)
- **Grid and cloud** computing (WLCG NT1, SNIC Cloud)

HPC2N (HPC2N at a glance)

Provides state-of-the-art resources and expertise:

- Scalable and parallel **HPC**
- Large-scale **storage facilities** (Project storage (Lustre), SweStore, Tape)
- **Grid and cloud** computing (WLCG NT1, SNIC Cloud)
- Support
  - Primary, advanced, dedicated
  - Application Experts (AEs)

HPC2N (HPC2N at a glance)

Provides state-of-the-art resources and expertise:

- Scalable and parallel **HPC**
- Large-scale **storage facilities** (Project storage (Lustre), SweStore, Tape)
- **Grid and cloud** computing (WLCG NT1, SNIC Cloud)
- Support
  - Primary, advanced, dedicated
  - Application Experts (AEs)
- International network for **research and development**

HPC2N (partners)

HPC2N has five **partners**:

- Luleå University of Technology
- Mid Sweden University
- Swedish Institute of Space Physics
- Swedish University of Agricultural Sciences (SLU)
- Umeå University

HPC2N (funding)

- Funded by **Swedish Research Council (VR)**, **SNIC** and various **partners**

HPC2N (funding)

- Funded by **Swedish Research Council (VR)**, **SNIC** and various **partners**

- Involved in several **projects and collaborations**
  - EGI, PRACE, EISCAT, eSSENCE, NOSEG, SNIC Science Cloud, ...

HPC2N (training and other services)

- **User support** (primary, advanced, dedicated)
  - Research group meetings @ UmU
  - Also at the partner sites

HPC2N (training and other services)

- **User support** (primary, advanced, dedicated)
  - Research group meetings @ UmU
  - Also at the partner sites
- **User training and education program**
  - 0.5 – 3 days; ready-to-run exercises
  - Introduction to HPC2N and Kebnekaise
  - Parallel programming and tools (e.g., OpenMP, MPI, debugging, performance analyzers, Matlab, R, MD simulation, Deep Learning, GPU, . . .)

HPC2N (training and other services)

- **User support** (primary, advanced, dedicated)
  - Research group meetings @ UmU
  - Also at the partner sites
- **User training and education program**
  - 0.5 – 3 days; ready-to-run exercises
  - Introduction to HPC2N and Kebnekaise
  - Parallel programming and tools (e.g., OpenMP, MPI, debugging, performance analyzers, Matlab, R, MD simulation, Deep Learning, GPU, . . .)
- NGSSC / SeSE & university courses

HPC2N (training and other services)

- **User support** (primary, advanced, dedicated)
  - Research group meetings @ UmU
  - Also at the partner sites
- **User training and education program**
  - 0.5 – 3 days; ready-to-run exercises
  - Introduction to HPC2N and Kebnekaise
  - Parallel programming and tools (e.g., OpenMP, MPI, debugging, performance analyzers, Matlab, R, MD simulation, Deep Learning, GPU, . . .)
- NGSSC / SeSE & university courses
- Workshops and seminars

HPC2N (personnel)

Management

- Paolo Bientinesi, director
- Björn Torkelsson, deputy director
- Lena Hellman, administrator

HPC2N (personnel)

Management

- Paolo Bientinesi, director
- Björn Torkelsson, deputy director
- Lena Hellman, administrator

Application experts

- Jerry Eriksson
- Pedro Ojeda May

HPC2N (personnel)

Management

- Paolo Bientinesi, director
- Björn Torkelsson, deputy director
- Lena Hellman, administrator

Application experts

- Jerry Eriksson
- Pedro Ojeda May

Others

- Bo Kågström
- Mikael Rännar (WLCG coord)
- Anders Backman
- Kenneth Bodin
- Claude Lacoursière (Algoryx)

HPC2N (personnel)

Management

- Paolo Bientinesi, director
- Björn Torkelsson, deputy director
- Lena Hellman, administrator

Application experts

- Jerry Eriksson
- Pedro Ojeda May

Others

- Bo Kågström
- Mikael Rännar (WLCG coord)
- Anders Backman
- Kenneth Bodin
- Claude Lacoursière (Algoryx)

System and support

- Erik Andersson
- **Birgitte Brydsö**
- Niklas Edmundsson (Tape coord)
- Ingemar Fällman
- Magnus Jonsson
- Roger Oscarsson
- **Åke Sandgren**
- Mattias Wadenstein (NeIC, Tier1)
- **Lars Viklund**

HPC2N (application experts)

- HPC2N provides advanced and dedicated support in the form of **Application Experts (AEs)**:

[1] https://www.snic.se/support/dedicated-user-support/

HPC2N (application experts)

- HPC2N provides advanced and dedicated support in the form of **Application Experts (AEs)**:

Jerry Eriksson	Profiling, Machine learning (DNN), MPI, OpenMP, OpenACC

[1] https://www.snic.se/support/dedicated-user-support/

HPC2N (application experts)

- HPC2N provides advanced and dedicated support in the form of **Application Experts (AEs)**:

Jerry Eriksson	Profiling, Machine learning (DNN), MPI, OpenMP, OpenACC
Pedro Ojeda May	Molecular dynamics, Profiling, QM/MM, NAMD, Amber, Gromacs, GAUSSIAN, R

[1] https://www.snic.se/support/dedicated-user-support/

HPC2N (application experts)

- HPC2N provides advanced and dedicated support in the form of **Application Experts (AEs)**:

Jerry Eriksson	Profiling, Machine learning (DNN), MPI, OpenMP, OpenACC
Pedro Ojeda May	Molecular dynamics, Profiling, QM/MM, NAMD, Amber, Gromacs, GAUSSIAN, R
Åke Sandgren	General high level programming assistance, VASP, Gromacs, Amber

[1] `https://www.snic.se/support/dedicated-user-support/`

HPC2N (application experts)

- HPC2N provides advanced and dedicated support in the form of **Application Experts (AEs)**:

Jerry Eriksson	Profiling, Machine learning (DNN), MPI, OpenMP, OpenACC
Pedro Ojeda May	Molecular dynamics, Profiling, QM/MM, NAMD, Amber, Gromacs, GAUSSIAN, R
Åke Sandgren	General high level programming assistance, VASP, Gromacs, Amber

- Contact through regular support or dedicated support form[1]
  - If you have a specific problem/question and/or need consultation (up to 100 h)

[1] `https://www.snic.se/support/dedicated-user-support/`

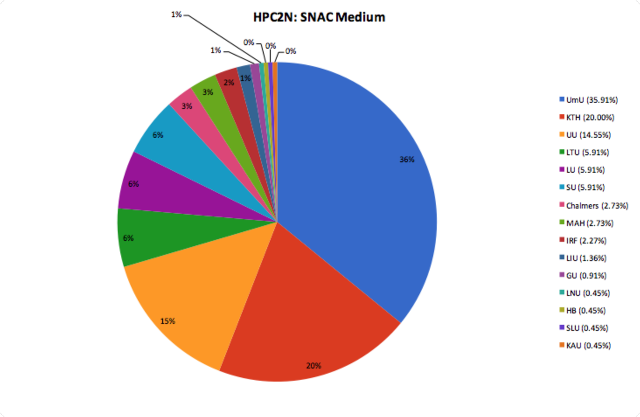
HPC2N (users by discipline)

- Users from several scientific disciplines:
  - Biosciences and medicine
  - Chemistry
  - Computing science
  - Engineering
  - Materials science
  - Mathematics and statistics
  - Physics including space physics
  - Deep learning and artificial intelligence

HPC2N (users by discipline, largest users)

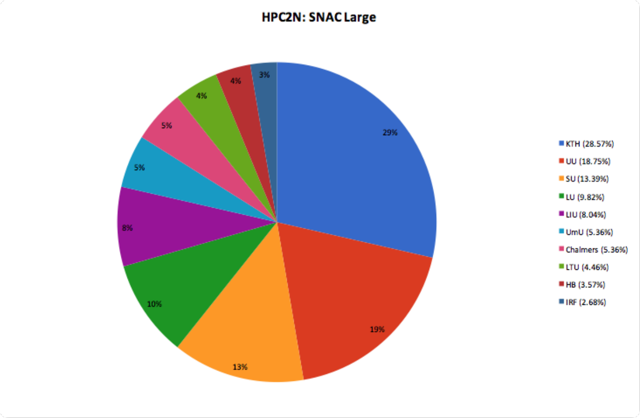
- Users from several scientific disciplines:
  - Biosciences and medicine
  - **Chemistry**
  - Computing science
  - Engineering
  - **Materials science**
  - Mathematics and statistics
  - **Physics including space physics**
  - Deep learning and artificial intelligence (several new projects)

HPC2N (medium users by university)



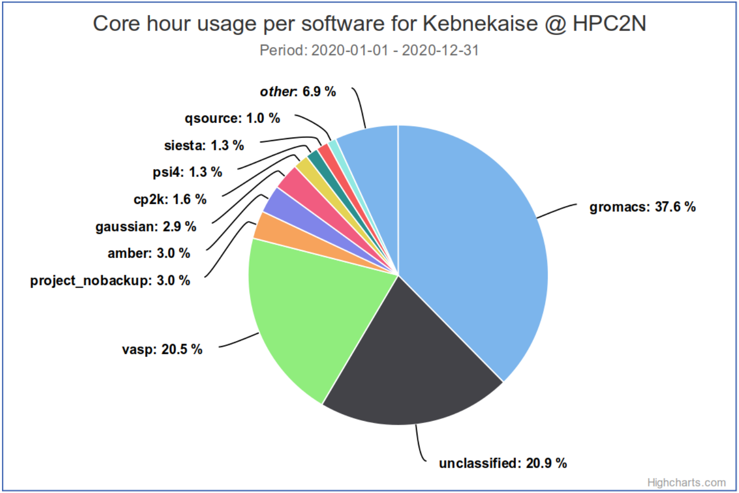
Projects with allocations at HPC2N: 2014-01-01 to 2016-05-30

HPC2N (large users by university)



Projects with allocations at HPC2N: 2014-01-01 to 2016-05-30

HPC2N (users by software)



Kebnekaise

- **Latest supercomputer at HPC2N**

Kebnekaise

- **Latest supercomputer at HPC2N**
- Named after a massif (contains some of Sweden's highest mountain peaks)

Kebnekaise

- **Latest supercomputer at HPC2N**
- Named after a massif (contains some of Sweden's highest mountain peaks)
- Kebnekaise was
  - delivered by Lenovo and
  - **installed during the summer 2016**

Kebnekaise

- **Latest supercomputer at HPC2N**
- Named after a massif (contains some of Sweden's highest mountain peaks)
- Kebnekaise was
  - delivered by Lenovo and
  - **installed during the summer 2016**
- Opened up for general availability on November 7, 2016

Kebnekaise

- **Latest supercomputer at HPC2N**
- Named after a massif (contains some of Sweden's highest mountain peaks)
- Kebnekaise was
  - delivered by Lenovo and
  - **installed during the summer 2016**
- Opened up for general availability on November 7, 2016
- In 2018, Kebnekaise was **extended** with
  - 52 Intel Xeon Gold 6132 (Skylake) nodes, as well as
  - 10 NVidian V100 (Volta) GPU nodes

Kebnekaise (compute nodes)

Name	#	Description
Compute	**432**	Intel Xeon E5-2690v4, **2 x 14 cores**, **128 GB**, FDR Infiniband

Kebnekaise (compute nodes)

Name	#	Description
Compute	**432**	Intel Xeon E5-2690v4, **2 x 14 cores**, **128 GB**, FDR Infiniband
Compute-skylake	52	Intel Xeon Gold 6132, 2 x 14 cores, **192 GB**, EDR Infiniband, **AVX-512**

Kebnekaise (compute nodes)

Name	#	Description
Compute	**432**	Intel Xeon E5-2690v4, **2 x 14 cores**, **128 GB**, FDR Infiniband
Compute-skylake	52	Intel Xeon Gold 6132, 2 x 14 cores, **192 GB**, EDR Infiniband, **AVX-512**
Large Memory	20	Intel Xeon E7-8860v4, **4 x 18 cores**, **3072 GB**, EDR Infiniband

Kebnekaise (compute nodes)

Name	#	Description
Compute	**432**	Intel Xeon E5-2690v4, **2 x 14 cores**, **128 GB**, FDR Infiniband
Compute-skylake	52	Intel Xeon Gold 6132, 2 x 14 cores, **192 GB**, EDR Infiniband, **AVX-512**
Large Memory	20	Intel Xeon E7-8860v4, **4 x 18 cores**, **3072 GB**, EDR Infiniband
KNL	36	Intel **Xeon Phi** 7250 (Knight's Landing), 68 cores, 192 GB, 16 GB MCDRAM, FDR Infiniband

Kebnekaise (GPU nodes)

Name	#	Description
2xGPU	32	Intel Xeon E5-2690v4, 2 x 14 cores, 128 GB, FDR Infiniband, **2 x NVidia K80** 4 x 2496 CUDA cores, 4 x 12 GB VRAM

Kebnekaise (GPU nodes)

Name	#	Description
2xGPU	32	Intel Xeon E5-2690v4, 2 x 14 cores, 128 GB, FDR Infiniband, **2 x NVidia K80** 4 x 2496 CUDA cores, 4 x 12 GB VRAM
4xGPU	4	Intel Xeon E5-2690v4, 2 x 14 cores, 128 GB, FDR Infiniband, **4 x NVidia K80** 8 x 2496 CUDA cores, 8 x 12 GB VRAM

Kebnekaise (GPU nodes)

Name	#	Description
2xGPU	32	Intel Xeon E5-2690v4, 2 x 14 cores, 128 GB, FDR Infiniband, **2 x NVidia K80** 4 x 2496 CUDA cores, 4 x 12 GB VRAM
4xGPU	4	Intel Xeon E5-2690v4, 2 x 14 cores, 128 GB, FDR Infiniband, **4 x NVidia K80** 8 x 2496 CUDA cores, 8 x 12 GB VRAM
GPU-volta	10	Intel Xeon Gold 6132, 2 x 14 cores, 192 GB, EDR Infiniband, **2 x NVidia V100**, 2 x 5120 CUDA cores, 2 x 16 GB VRAM, **2 x 640 Tensor cores**

Kebnekaise (in numbers)

- 602 nodes in 15 racks

UMEÅ UNIVERSITY SNIC HPC2N

Kebnekaise (in numbers)

- 602 nodes in 15 racks
- **19288 cores** (of which 2448 cores are KNL-cores)
  - 18840 available for users (the rest are for managing the cluster)

Kebnekaise (in numbers)

- 602 nodes in 15 racks
- **19288 cores** (of which 2448 cores are KNL-cores)
  - 18840 available for users (the rest are for managing the cluster)
- More than **136 TB memory**

Kebnekaise (in numbers)

- 602 nodes in 15 racks
- **19288 cores** (of which 2448 cores are KNL-cores)
  - 18840 available for users (the rest are for managing the cluster)
- More than **136 TB memory**
- 71 switches (Infiniband, Access and Managment networks)

Kebnekaise (in numbers)

- 602 nodes in 15 racks
- **19288 cores** (of which 2448 cores are KNL-cores)
  - 18840 available for users (the rest are for managing the cluster)
- More than **136 TB memory**
- 71 switches (Infiniband, Access and Managment networks)
- 728 TFlops/s Peak performance (expansion not included)

Kebnekaise (in numbers)

- 602 nodes in 15 racks
- **19288 cores** (of which 2448 cores are KNL-cores)
  - 18840 available for users (the rest are for managing the cluster)
- More than **136 TB memory**
- 71 switches (Infiniband, Access and Managment networks)
- 728 TFlops/s Peak performance (expansion not included)
- **629 TFlops/s** Linpack (all parts, except expansion)
  - 86% of Peak performance

Kebnekaise (HPC2N storage)

- Basically five types of storage are available at HPC2N:

Kebnekaise (HPC2N storage)

- Basically five types of storage are available at HPC2N:
  - Home directory
    - `/home/X/Xyz`, `$HOME`, ~
    - 25 GB, user owned

Kebnekaise (HPC2N storage)

- Basically five types of storage are available at HPC2N:
  - Home directory
    - `/home/X/Xyz`, `$HOME`, ~
    - 25 GB, user owned
  - Project storage
    - `/proj/nobackup/abc`
    - Shared among project members

UMEÅ UNIVERSITY SNIC HPC2N

Kebnekaise (HPC2N storage)

- Basically five types of storage are available at HPC2N:
  - Home directory
    - `/home/X/Xyz`, `$HOME`, ~
    - 25 GB, user owned
  - Project storage
    - `/proj/nobackup/abc`
    - Shared among project members
  - Local scratch space
    - `$SNIC_TMP`
    - SSD (170GB), per job, per node, "volatile"

Kebnekaise (HPC2N storage)

- Basically five types of storage are available at HPC2N:
  - Home directory
    - `/home/X/Xyz`, `$HOME`, ~
    - 25 GB, user owned
  - Project storage
    - `/proj/nobackup/abc`
    - Shared among project members
  - Local scratch space
    - `$SNIC_TMP`
    - SSD (170GB), per job, per node, "volatile"
  - SweStore — disk based (dCache)
    - part of SNIC Storage, **nationally accessible storage**

Kebnekaise (HPC2N storage)

- Basically five types of storage are available at HPC2N:
  - Home directory
    - `/home/X/Xyz`, `$HOME`, ~
    - 25 GB, user owned
  - Project storage
    - `/proj/nobackup/abc`
    - Shared among project members
  - Local scratch space
    - `$SNIC_TMP`
    - SSD (170GB), per job, per node, "volatile"
  - SweStore — disk based (dCache)
    - part of SNIC Storage, **nationally accessible storage**
  - Tape Storage
    - Backup
    - **Long term storage**

Kebnekaise (projects)

- In order to use Kebnekaise, you must be a member of a **compute project**

Kebnekaise (projects)

- In order to use Kebnekaise, you must be a member of a **compute project**
  - A compute project has a certain number of **core hours** allocated for it per month

Kebnekaise (projects)

- In order to use Kebnekaise, you must be a member of a **compute project**
  - A compute project has a certain number of **core hours** allocated for it per month
  - A regular CPU core cost 1 core hour per hour, other resources (e.g., GPUs) cost more

Kebnekaise (projects)

- In order to use Kebnekaise, you must be a member of a **compute project**
  - A compute project has a certain number of **core hours** allocated for it per month
  - A regular CPU core cost 1 core hour per hour, other resources (e.g., GPUs) cost more
  - Not a hard limit but projects that go over the allocation get lower priority

Kebnekaise (projects)

- In order to use Kebnekaise, you must be a member of a **compute project**
  - A compute project has a certain number of **core hours** allocated for it per month
  - A regular CPU core cost 1 core hour per hour, other resources (e.g., GPUs) cost more
  - Not a hard limit but projects that go over the allocation get lower priority
- A compute project contains a certain amount of storage
  - If more storage is required, you must be a member of a **storage project**

Kebnekaise (projects)

- In order to use Kebnekaise, you must be a member of a **compute project**
  - A compute project has a certain number of **core hours** allocated for it per month
  - A regular CPU core cost 1 core hour per hour, other resources (e.g., GPUs) cost more
  - Not a hard limit but projects that go over the allocation get lower priority
- A compute project contains a certain amount of storage
  - If more storage is required, you must be a member of a **storage project**
- I will cover more details in the next section, where we go more in to detail about HPC2N and Kebnekaise

High Performance Computing (definition)

"High Performance Computing most generally refers to the practice of **aggregating computing power** in a way that delivers much **higher performance** than one could get out of a typical desktop computer or workstation in order to **solve large problems** in science, engineering, or business."[2]

[2] https://insidehpc.com/hpc-basic-training/what-is-hpc/

High Performance Computing (opening the definition)

- **Aggregating computing power**
  - 602 nodes in 15 racks totalling 19288 cores
  - Compared to 4 cores in a modern laptop

[3] 728 trillion (billion)
[4] 200 billion (milliard)

High Performance Computing (opening the definition)

- **Aggregating computing power**
  - 602 nodes in 15 racks totalling 19288 cores
  - Compared to 4 cores in a modern laptop
- **Higher performance**
  - 728 000 000 000 000 arithmetical operations per second[3]
  - Compared to 200 000 000 000 Flops in a modern laptop[4]

[3] 728 trillion (billion)

[4] 200 billion (milliard)

High Performance Computing (opening the definition)

- **Aggregating computing power**
  - 602 nodes in 15 racks totalling 19288 cores
  - Compared to 4 cores in a modern laptop
- **Higher performance**
  - 728 000 000 000 000 arithmetical operations per second[3]
  - Compared to 200 000 000 000 Flops in a modern laptop[4]
- **Solve large problems**
  - When does a problem become large enough for HPC?
  - Are there other reasons for using HPC resources? (Memory, software, support, etc.)

[3] 728 trillion (billion)
[4] 200 billion (milliard)

High Performance Computing (large problems)

- A problem can be large for two main reasons:
  1. Execution time: The time required to form a solution to the problem is very long
  2. Memory / storage use: The solution of the problem requires a lot of memory and/or storage

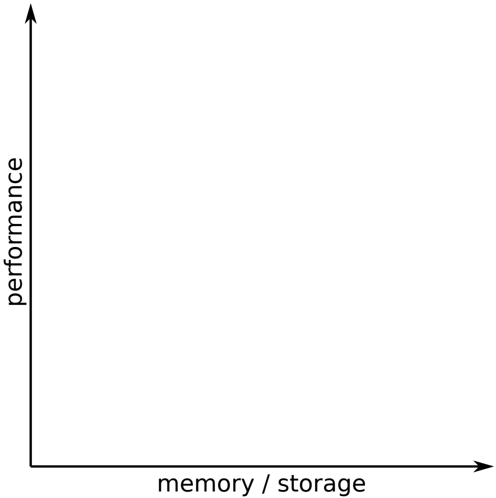
High Performance Computing (large problems)

- A problem can be large for two main reasons:
  1. Execution time: The time required to form a solution to the problem is very long
  2. Memory / storage use: The solution of the problem requires a lot of memory and/or storage
- The former can be remedied by **increasing the performance**
  - More cores, more nodes, GPUs, ...

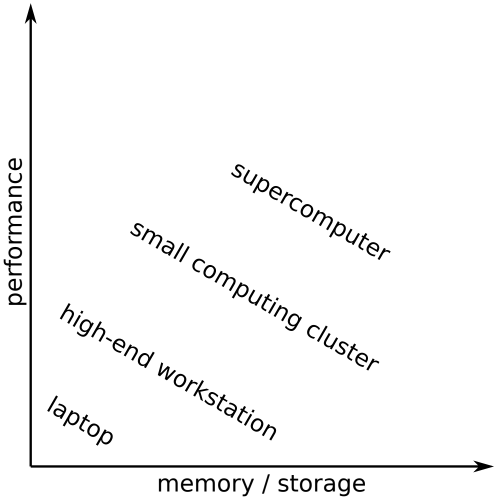
High Performance Computing (large problems)

- A problem can be large for two main reasons:
  1. Execution time: The time required to form a solution to the problem is very long
  2. Memory / storage use: The solution of the problem requires a lot of memory and/or storage
- The former can be remedied by **increasing the performance**
  - More cores, more nodes, GPUs, ...
- The latter by **adding more memory / storage**
  - More memory per node (including large memory nodes), more nodes, ...
  - Large storage solutions, ...

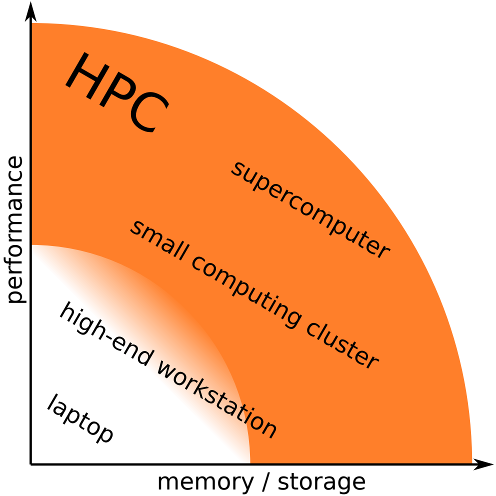
High Performance Computing (what counts as HPC)



High Performance Computing (what counts as HPC)



High Performance Computing (what counts as HPC)



High Performance Computing (other reasons)

- Specialized (expensive) hardware

High Performance Computing (other reasons)

- Specialized (expensive) hardware
  - GPUs, **Nvidia Tesla V100 GPUs** are optimized for AI

High Performance Computing (other reasons)

- Specialized (expensive) hardware
  - GPUs, **Nvidia Tesla V100 GPUs** are optimized for AI
  - Intel Xeon Phi

High Performance Computing (other reasons)

- Specialized (expensive) hardware
  - GPUs, **Nvidia Tesla V100 GPUs** are optimized for AI
  - Intel Xeon Phi
  - High-end CPUs (AVX-512 etc) and ECC memory

High Performance Computing (other reasons)

- Specialized (expensive) hardware
  - GPUs, **Nvidia Tesla V100 GPUs** are optimized for AI
  - Intel Xeon Phi
  - High-end CPUs (AVX-512 etc) and ECC memory
- Software
  - HPC2N holds **licenses** for several softwares
  - Software is **pre-configured and ready-to-use**

High Performance Computing (other reasons)

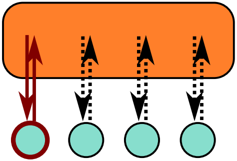
- Specialized (expensive) hardware
  - GPUs, **Nvidia Tesla V100 GPUs** are optimized for AI
  - Intel Xeon Phi
  - High-end CPUs (AVX-512 etc) and ECC memory
- Software
  - HPC2N holds **licenses** for several softwares
  - Software is **pre-configured and ready-to-use**
- **Support and documentation**

High Performance Computing (memory models)

- Two memory models are relevant for HPC:

High Performance Computing (memory models)

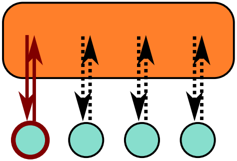
- Two memory models are relevant for HPC:
  - Shared memory: Single memory space for all data.



- **Everyone can access the same data**
- Straightforward to use

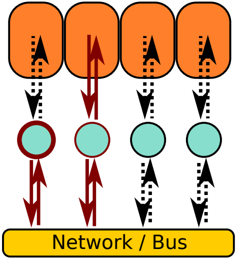
High Performance Computing (memory models)

- Two memory models are relevant for HPC:
  - Shared memory: Single memory space for all data.



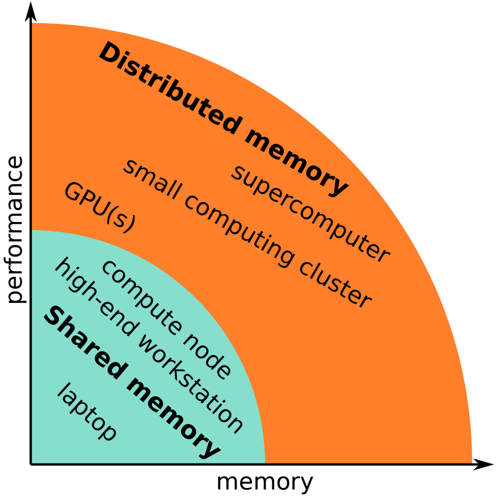
- **Everyone can access the same data**
- Straightforward to use

  - Distributed memory: Multiple **distinct** memory spaces.



- Everyone has direct access **only to the local data**
- Requires **communication**

High Performance Computing (memory models)



High Performance Computing (programming models)

- The programming model changes when we aim for extra performance and/or memory:

High Performance Computing (programming models)

- The programming model changes when we aim for extra performance and/or memory:
  1. Single-core: Matlab, Python, C, Fortran, ...
     - Single stream of operations

High Performance Computing (programming models)

- The programming model changes when we aim for extra performance and/or memory:
  1. Single-core: Matlab, Python, C, Fortran, ...
     - Single stream of operations
  2. Multi-core: Vectorized Matlab, pthreads, **OpenMP**
     - **Multiple streams** of operations

High Performance Computing (programming models)

- The programming model changes when we aim for extra performance and/or memory:
  1. Single-core: Matlab, Python, C, Fortran, ...
     - Single stream of operations
  2. Multi-core: Vectorized Matlab, pthreads, **OpenMP**
     - **Multiple streams** of operations
     - **Work distribution**, **coordination** (synchronization, etc), ...

High Performance Computing (programming models)

- The programming model changes when we aim for extra performance and/or memory:
  1. Single-core: Matlab, Python, C, Fortran, ...
     - Single stream of operations
  2. Multi-core: Vectorized Matlab, pthreads, **OpenMP**
     - **Multiple streams** of operations
     - **Work distribution**, **coordination** (synchronization, etc), ...
  3. Distributed memory: **MPI**, ...
     - Multiple streams of operations
     - Work distribution, coordination (synchronization, etc), ...

High Performance Computing (programming models)

- The programming model changes when we aim for extra performance and/or memory:
  1. Single-core: Matlab, Python, C, Fortran, ...
     - Single stream of operations
  2. Multi-core: Vectorized Matlab, pthreads, **OpenMP**
     - **Multiple streams** of operations
     - **Work distribution**, **coordination** (synchronization, etc), ...
  3. Distributed memory: **MPI**, ...
     - Multiple streams of operations
     - Work distribution, coordination (synchronization, etc), ...
     - **Data distribution and communication**

High Performance Computing (programming models)

- The programming model changes when we aim for extra performance and/or memory:
  1. Single-core: Matlab, Python, C, Fortran, ...
     - Single stream of operations
  2. Multi-core: Vectorized Matlab, pthreads, **OpenMP**
     - **Multiple streams** of operations
     - **Work distribution**, **coordination** (synchronization, etc), ...
  3. Distributed memory: **MPI**, ...
     - Multiple streams of operations
     - Work distribution, coordination (synchronization, etc), ...
     - **Data distribution and communication**
- GPUs: **CUDA**, OpenCL, OpenACC, OpenMP, ...

High Performance Computing (programming models)

- The programming model changes when we aim for extra performance and/or memory:
  1. Single-core: Matlab, Python, C, Fortran, ...
     - Single stream of operations
  2. Multi-core: Vectorized Matlab, pthreads, **OpenMP**
     - **Multiple streams** of operations
     - **Work distribution**, **coordination** (synchronization, etc), ...
  3. Distributed memory: **MPI**, ...
     - Multiple streams of operations
     - Work distribution, coordination (synchronization, etc), ...
     - **Data distribution and communication**
- GPUs: **CUDA**, OpenCL, OpenACC, OpenMP, ...
  - **Many lightweight** streams of operations

High Performance Computing (programming models)

- The programming model changes when we aim for extra performance and/or memory:
  1. Single-core: Matlab, Python, C, Fortran, ...
     - Single stream of operations
  2. Multi-core: Vectorized Matlab, pthreads, **OpenMP**
     - **Multiple streams** of operations
     - **Work distribution**, **coordination** (synchronization, etc), ...
  3. Distributed memory: **MPI**, ...
     - Multiple streams of operations
     - Work distribution, coordination (synchronization, etc), ...
     - **Data distribution and communication**
- GPUs: **CUDA**, OpenCL, OpenACC, OpenMP, ...
  - **Many lightweight** streams of operations
  - Work distribution, coordination (synchronization, etc), ...

High Performance Computing (programming models)

- The programming model changes when we aim for extra performance and/or memory:
  1. Single-core: Matlab, Python, C, Fortran, ...
     - Single stream of operations
  2. Multi-core: Vectorized Matlab, pthreads, **OpenMP**
     - **Multiple streams** of operations
     - **Work distribution**, **coordination** (synchronization, etc), ...
  3. Distributed memory: **MPI**, ...
     - Multiple streams of operations
     - Work distribution, coordination (synchronization, etc), ...
     - **Data distribution and communication**
- GPUs: **CUDA**, OpenCL, OpenACC, OpenMP, ...
  - **Many lightweight** streams of operations
  - Work distribution, coordination (synchronization, etc), ...
  - **Data distribution across memory spaces and movement**

High Performance Computing (software)

- Complexity grows when we aim for extra performance and/or memory/storage:

High Performance Computing (software)

- Complexity grows when we aim for extra performance and/or memory/storage:
  1. Single-core: LAPACK, ...
     - Load correct toolchain etc

High Performance Computing (software)

- Complexity grows when we aim for extra performance and/or memory/storage:
  1. Single-core: LAPACK, ...
     - Load correct toolchain etc
  2. Multi-core: LAPACK + parallel BLAS, ...
     - Load correct toolchain etc

High Performance Computing (software)

- Complexity grows when we aim for extra performance and/or memory/storage:
  1. Single-core: LAPACK, ...
     - Load correct toolchain etc
  2. Multi-core: LAPACK + parallel BLAS, ...
     - Load correct toolchain etc
     - **Allocate** correct number of cores, **configure** software to use correct number of cores, ...

High Performance Computing (software)

- Complexity grows when we aim for extra performance and/or memory/storage:
  1. Single-core: LAPACK, ...
     - Load correct toolchain etc
  2. Multi-core: LAPACK + parallel BLAS, ...
     - Load correct toolchain etc
     - **Allocate** correct number of cores, **configure** software to use correct number of cores, ...
  3. Distributed memory: ScaLAPACK, ...
     - Load correct toolchain etc

High Performance Computing (software)

- Complexity grows when we aim for extra performance and/or memory/storage:
  1. Single-core: LAPACK, ...
     - Load correct toolchain etc
  2. Multi-core: LAPACK + parallel BLAS, ...
     - Load correct toolchain etc
     - **Allocate** correct number of cores, **configure** software to use correct number of cores, ...
  3. Distributed memory: ScaLAPACK, ...
     - Load correct toolchain etc
     - Allocate correct number of **nodes and cores**, configure software to use correct number of **nodes and cores**, ...

High Performance Computing (software)

- Complexity grows when we aim for extra performance and/or memory/storage:
  1. Single-core: LAPACK, ...
     - Load correct toolchain etc
  2. Multi-core: LAPACK + parallel BLAS, ...
     - Load correct toolchain etc
     - **Allocate** correct number of cores, **configure** software to use correct number of cores, ...
  3. Distributed memory: ScaLAPACK, ...
     - Load correct toolchain etc
     - Allocate correct number of **nodes and cores**, configure software to use correct number of **nodes and cores**, ...
     - Data distribution, storage, ...

High Performance Computing (software)

- Complexity grows when we aim for extra performance and/or memory/storage:
  1. Single-core: LAPACK, ...
     - Load correct toolchain etc
  2. Multi-core: LAPACK + parallel BLAS, ...
     - Load correct toolchain etc
     - **Allocate** correct number of cores, **configure** software to use correct number of cores, ...
  3. Distributed memory: ScaLAPACK, ...
     - Load correct toolchain etc
     - Allocate correct number of **nodes and cores**, configure software to use correct number of **nodes and cores**, ...
     - Data distribution, storage, ...
- GPUs: MAGMA, TensorFlow, ...
  - Load correct toolchain etc
  - Allocate correct number of **cores and GPUs**, configure software to use correct number of **cores and GPUs**, ...

End (questions?)

Questions?