Beginner's guide to eX³

Tore H. Larsen
James Trotter

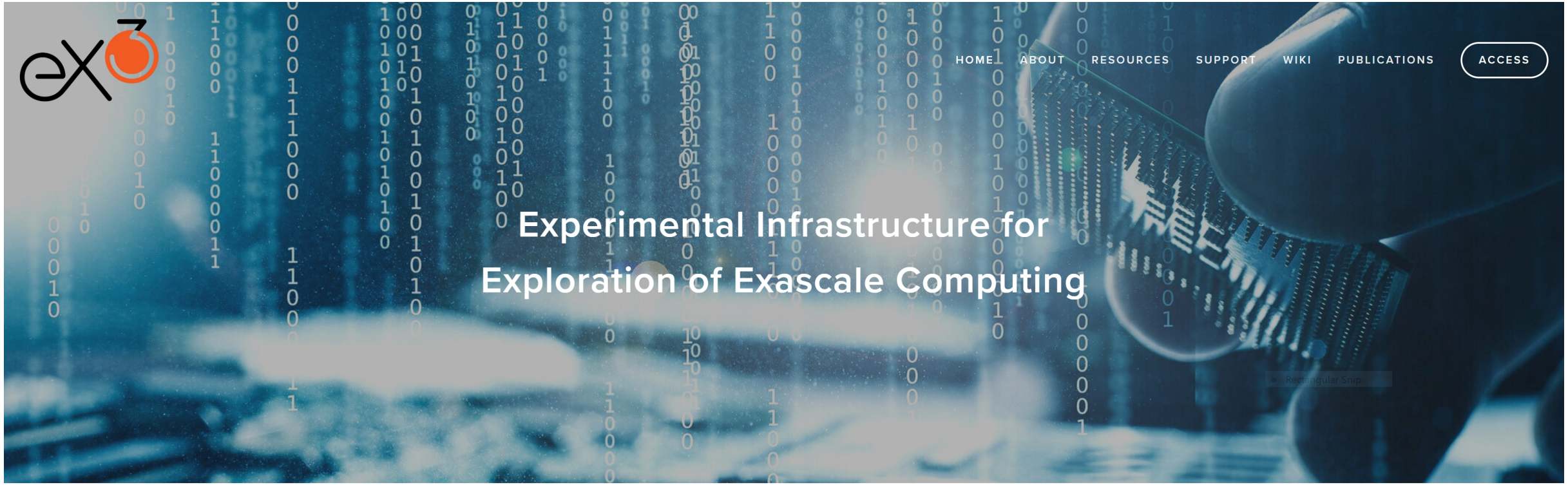
High-performance Computing,
Simula Research Laboratory

Tools-meetup: eX³ tutorial

21 November 2022

https://www.ex3.simula.no/

Primary objective of eX³



"The primary objective of the eX3 project is to provide an experimental, heterogeneous computational cluster as a national infrastructure that will help Norwegian HPC researchers, HPC/Big Data users, data center management, and HPC industry prepare for the coming exascale era of computing."

https://www.ex3.simula.no/

The eX³ infrastructure will provide the Norwegian HPC community and collaborators with a foundation for exascale preparations

Subgoals:

- Access to bleeding-edge technology
- Educate students, researchers and practitioners
- Develop powerful programming models, best practices, and software tools
- Provide objective measurements and analyses

By its nature, itself being the target for research, eX³ is a different infrastructure

2018 – 2023

eX³ - Research Council of Norway under contract 270053

Project team:

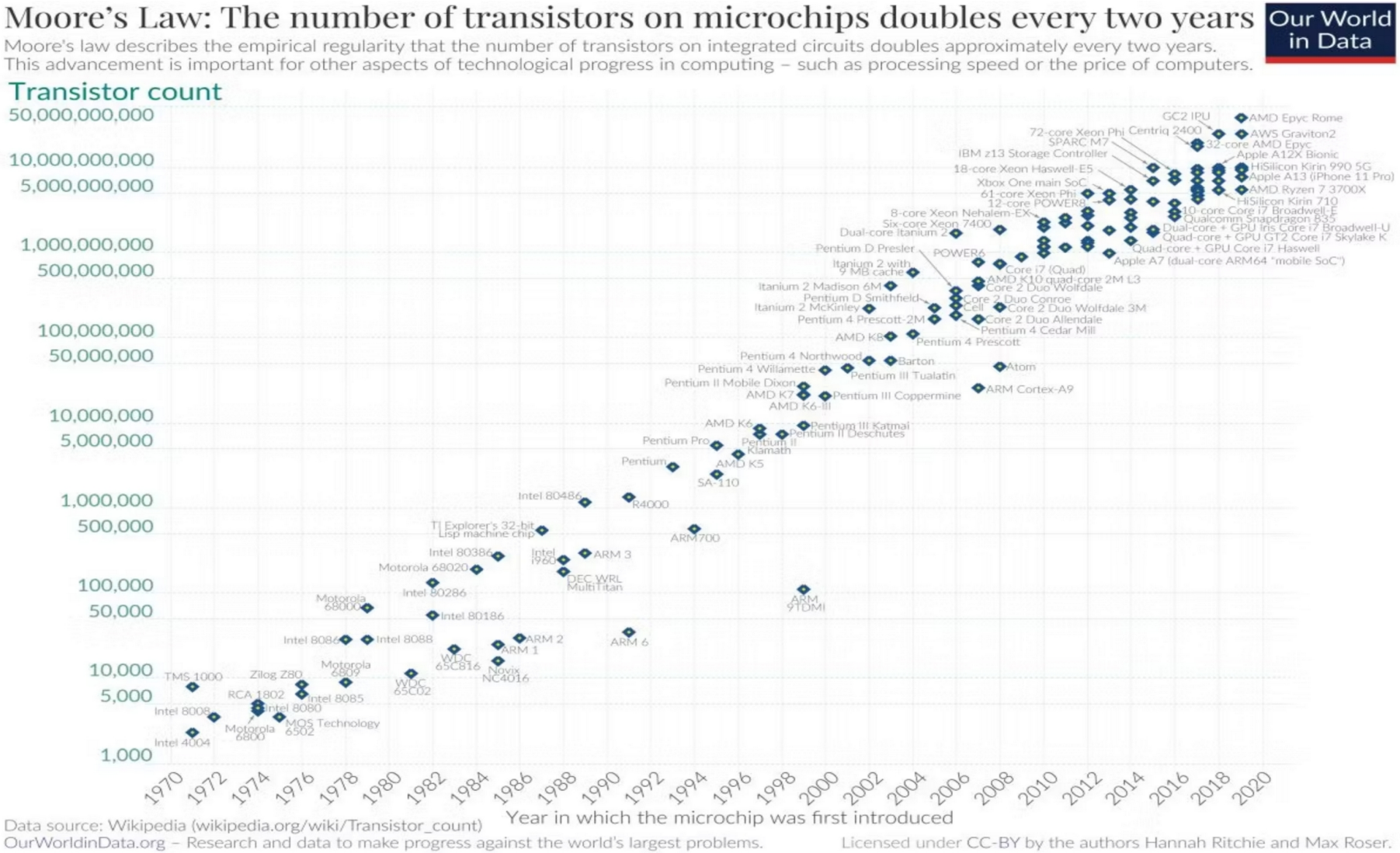
Prof. Are Magnus Bruaset (PL) (SRL)

Prof. Xing Cai (UiO, SRL HPC)

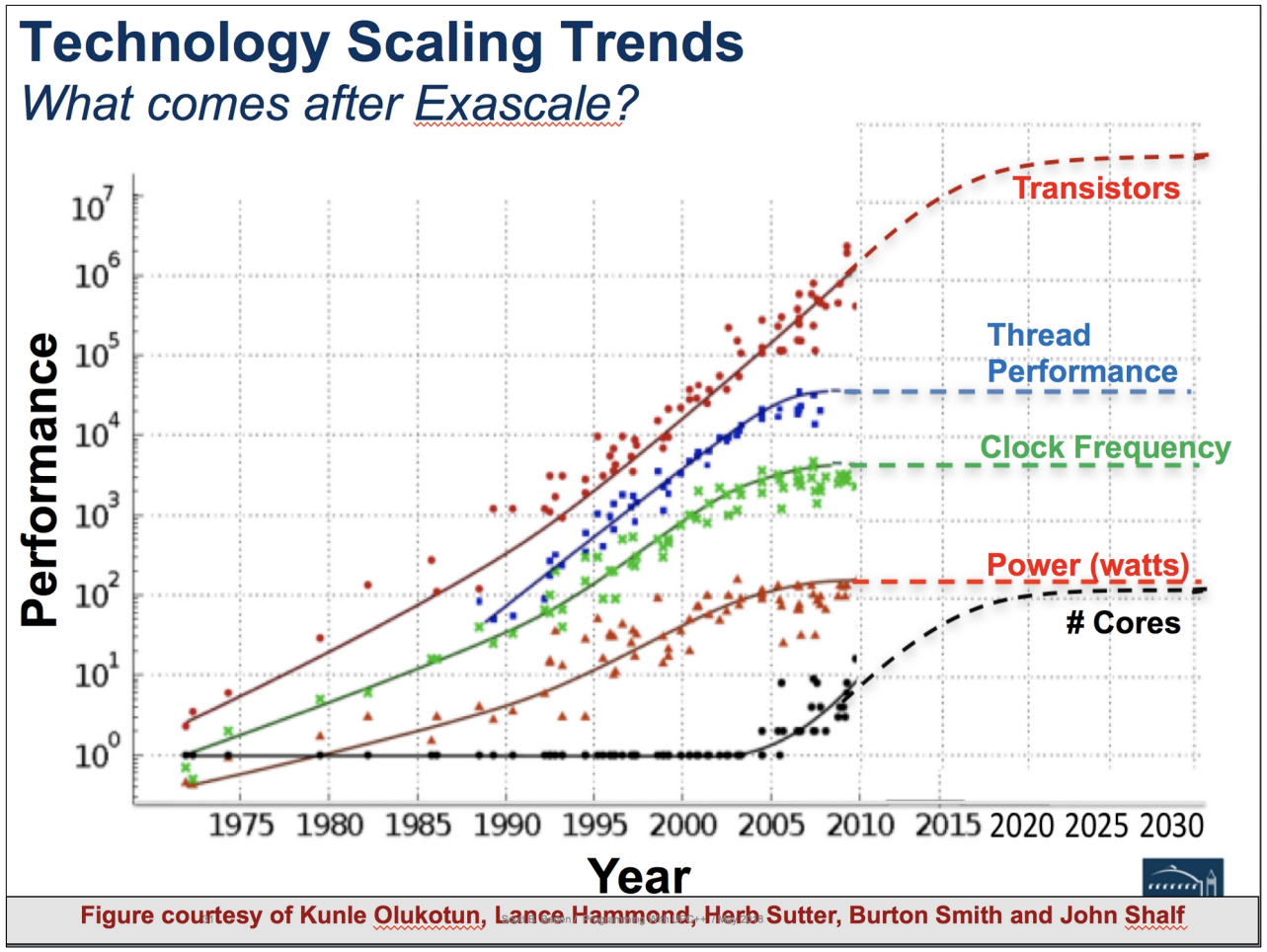
Ass Prof. Ernst G. Gran (NTNU, SRL HPC)

Tore H. Larsen, SRL HPC

https://www.ex3.simula.no/



https://en.wikipedia.org/wiki/Transistor_count

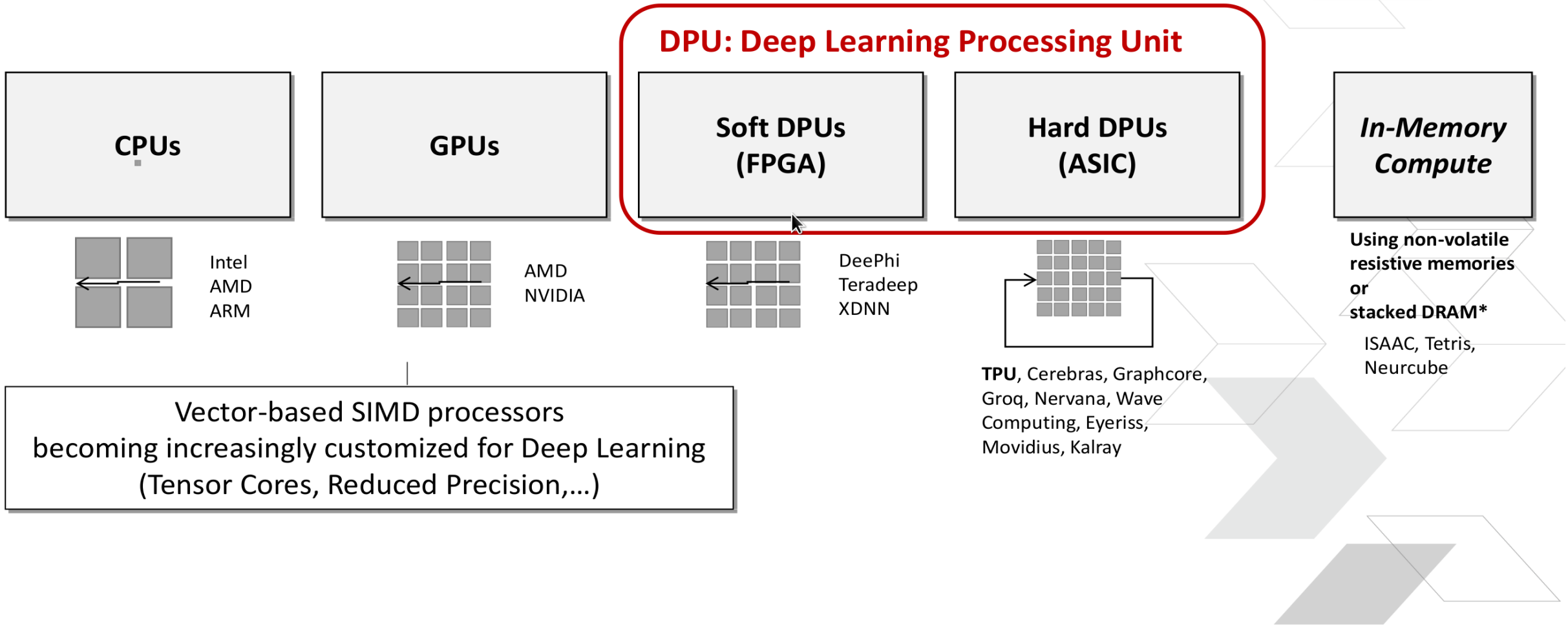


Rank	System	Cores	Rmax (PFlop/s)	Rpeak (PFlop/s)	Power (kW)
1	**Frontier** - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE DOE/SC/Oak Ridge National Laboratory United States	8,730,112	1,102.00	1,685.65	21,100
2	**Supercomputer Fugaku** - Supercomputer Fugaku, A64FX 48C 2.2GHz, Tofu interconnect D, Fujitsu RIKEN Center for Computational Science Japan	7,630,848	442.01	537.21	29,899
3	**LUMI** - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE EuroHPC/CSC Finland	2,220,288	309.10	428.70	6,016
4	**Leonardo** - BullSequana XH2000, Xeon Platinum 8358 32C 2.6GHz, NVIDIA A100 SXM4 64 GB, Quad-rail NVIDIA HDR100 Infiniband, Atos EuroHPC/CINECA Italy	1,463,616	174.70	255.75	5,610
5	**Summit** - IBM Power System AC922, IBM POWER9 22C 3.07GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband, IBM DOE/SC/Oak Ridge National Laboratory United States	2,414,592	148.60	200.79	10,096
6	**Sierra** - IBM Power System AC922, IBM POWER9 22C 3.1GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband, IBM / NVIDIA / Mellanox DOE/NNSA/LLNL United States	1,572,480	94.64	125.71	7,438
7	**Sunway TaihuLight** - Sunway MPP, Sunway SW26010 260C 1.45GHz, Sunway, NRCPC National Supercomputing Center in Wuxi China	10,649,600	93.01	125.44	15,371
8	**Perlmutter** - HPE Cray EX235n, AMD EPYC 7763 64C 2.45GHz, NVIDIA A100 SXM4 40 GB, Slingshot-10, HPE DOE/SC/LBNL/NERSC United States	761,856	70.87	93.75	2,589

Top 8 of Top 500, Nov 2022

Two answers: heterogeneous systems and domain-specific processors

“This is a golden age for computer architecture“ - Patterson**



*Gfx from “Design Trade-offs for Machine Learning Solutions on Reconfigurable Devices”, Michaela Blott, Xilinx Research, ASAP July 2018

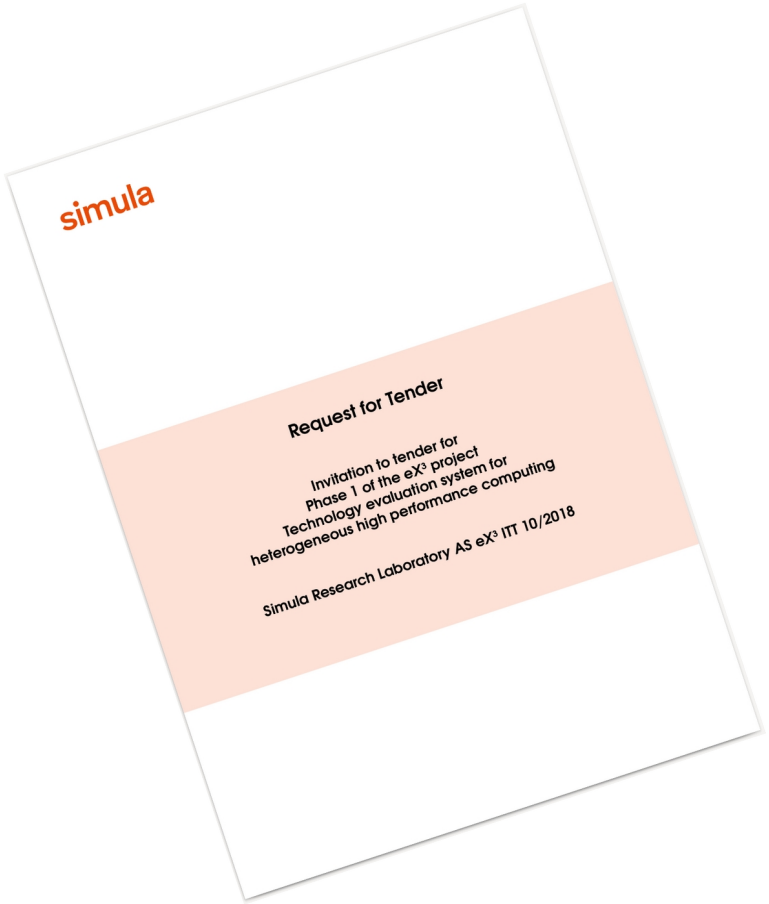
** David Patterson says It’s Time for New Computer Architectures and Software Languages, IEEE Spectrum, Sept. 2018

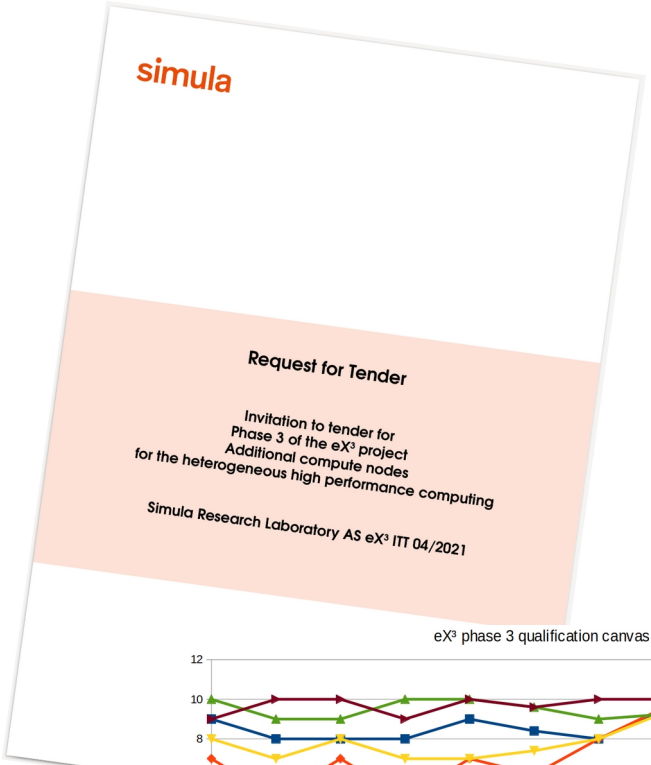
Objectives of eX³

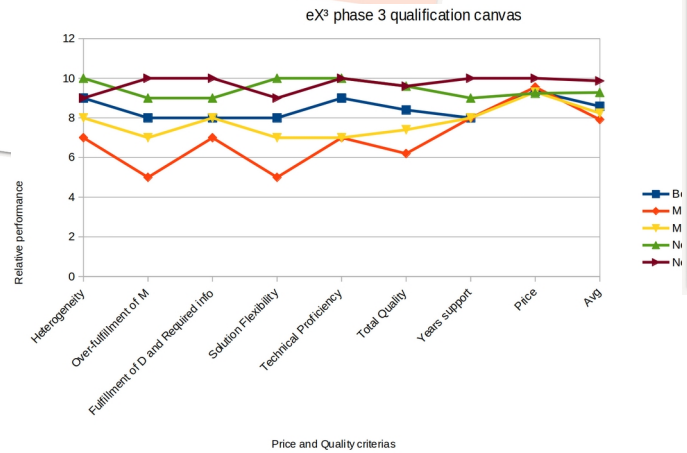
- To be a heterogeneous top-bin playground for students (MSc and Ph.D.) and researchers/postdocs.
- Allow them to test port code, test hypothesis on various processing elements
- Top-bin SKU, hardware which will be in the next generation of academic supercomputers
- (GPCPU (Arm/x86_64), GPGPU (V100/A100/Mi100), IPU/NPU/HPU (GC-200, HL205), neuromorphic hardware (Brainchip)

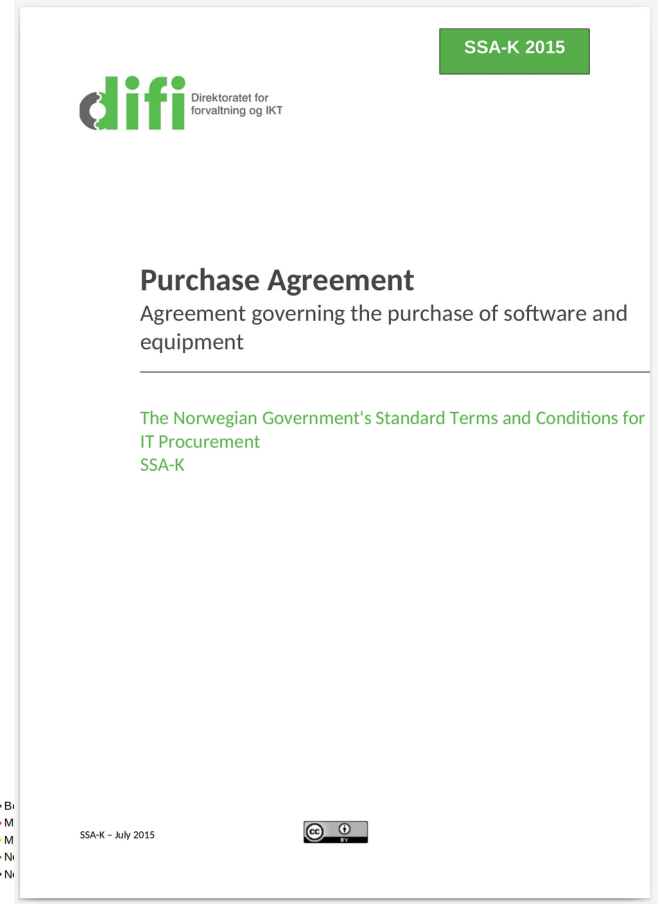
- https://www.simula.no/news/ex3-nutshell-new-feature-article
- https://www.simula.no/news/simulas-ex3-infrastructure-part-esfri-roadmap-2021
- https://www.simula.no/news/ex3-infrastructure-brings-state-art-ai-compute-research-community
- https://www.simula.no/feature-preparing-norway-next-generation-supercomputers

An Experimental Infrastructure for Exploration of Exascale Computing









The eX³ consortium has partners representing the full HPC landscape

HPC Researchers

Simula UiB

SIMULA@BI

UNIVERSITAS BERGENSIS

UNIVERSITY OF BERGEN

UNIVERSITY OF TROMSØ

NTNU

Norwegian University of Science and Technology

TECHNISCHE UNIVERSITÄT BERLIN

BMU

KOÇ ÜNİVERSİTESİ

Høgskolen i Østfold

HPC Management

UNINETT

Sigma2

HPC Users

BigInsight

SIRIUS

MICROCARD

SPARCITY

Meteorologisk institutt

HPC Industry

Dolphin INTERCONNECT SOLUTIONS

NUMASCALE

GRAPHCORE


NVIDIA

eX³ is an Experimental Infrastructure for Exploration of Exascale Computing

System stability

eX³ is an *experimental cluster:*

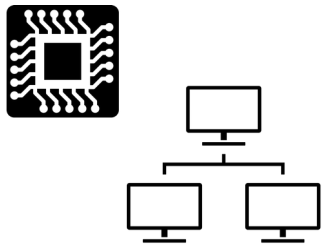
- can be **reconfigured and changed at any time**,
- can be **reserved for specific users** (for short periods)

Therefore:

- you **cannot expect the same uptime** as other systems,
- you **cannot store data or results for longer periods**.

https://www.ex3.simula.no/

Beginner’s guide to eX3



1. Hardware overview

2. Accessing eX3

slurm
workload manager

3. Getting started

[LAN by Stefan Traistaru from the Noun Project], [Microchip By Márcio Duarte, from the Noun Project], [DEC VT100 terminal by Gorthmog, CC BY-SA 4.0, via Wikimedia Commons]

A **highly heterogeneous cluster** with different computing hardware, network interconnects and storage solutions

CPUs:

- AMD Epyc (*Naples, Rome, Milan*)
- Intel Xeon (*Skylake Platinum/Gold/Silver, Icelake*)
- ARM Cavium ThunderX2
- HiSilicon Kunpeng

Accelerators:

- NVIDIA V100, A40 and A100 GPUs
- AMD Vega20, Instinct Mi100 & Mi210
- Graphcore M2k GC200 IPUs
- Brainchip Akida AKD1000
- Intel Habana Gaudi HL205

Interconnect:

- 200 Gbps Infiniband HDR network
  - Ib1net (stable, BeeGFS+MPI)
  - Ib2net (AR/CC research: slowq/rome16q)
- Dolphin Interconnect PCIe Gen3 NTB
- 10/25/100 Gbps Ethernet network
- 1Gbps «ME» network (BMC,iDRAC,ILO,++)

Storage:

- Up to 1 or 2 TB RAM per node
- BeeGFS parallel file system (ib0,eno0,dis0)
- >4TB NVMe/SSD scratch storage

eX3 has more than 40 nodes with lots of different hardware

Compute nodes	Hostnames	Slurm partitions
2x AMD Epyc 7601, 2TB RAM, AMD Vega20 GPU, 4TB NVMe	`n001–n003`	`defq, amdgpuq`
2x AMD Epyc 7601, 2TB RAM, AMD Instinct Mi100 GPU, 4TB NVMe	`n004`	`defq, mi100q`
2x Cavium ThunderX2 CN9980, 1TB RAM, 5TB SSD	`n005–n008`	`armq`
2x HiSilicon Kunpeng 920-6426, 1TB RAM, 4TB SSD	`n009–n012`	`huaq, a40q`
2x AMD Epyc 7763, 2TB RAM, NVIDIA A100 GPUs, 4TB NVMe	`n013–n014`	`milanq, a100q`
2x AMD Epyc 7763, 2TB RAM, AMD Instinct Mi210 GPUs, 4TB NVMe	`n015–n016`	`milanq, mi210q`
2x AMD Epyc 7413	`n017–n020`	`fpgaq`
2x Intel Xeon Platinum 8186, 16x NVIDIA V100 GPUs	`g001`	`dgx2q`
2x AMD Epyc 7763, 2TB RAM, 8x NVIDIA A100 GPUs	`g002`	`hgx2q`
2x Intel Xeon SP 8360Y, 2TB RAM, 8x Habana Gaudi HL205	`h001`	`habanaq`
2x Intel Xeon Gold 6130	`srl-login1,srl-adm1, srl-mds1,srl-mds2`	`xeongold16q`
1x Intel Xeon Silver 4112	`n041–n048`	`slowq`
1x AMD Epyc 7302P, 256GB RAM	`n049–n060`	`rome16q`
2x AMD Epyc 7302P, 512GB RAM, Xilinx Alveo U250/U280	`n061–n064`	`fpga32q`
Graphcore IPUPOD64, 64x GC200 IPUs	`ipu-pod64-server1`	`ipuq`

Beginner’s guide to eX³

Getting access to eX³

Start by requesting access to eX³: https://www.ex3.simula.no/access

Steps:

1. Request access at: *https://www.ex3.simula.no/access*.
2. You will receive an email from the system administrator with your username and password
3. Use an *SSH client* to login and change your password
    - Use, for example, OpenSSH on Linux, PuTTY on Windows or the builtin client on MacOS
4. Set up Public Key authentication
    - See, for example, the following tutorial on connecting to GitHub with SSH using Public Key authentication: https://docs.github.com/en/github/authenticating-to-github/connecting-to-github-with-ssh

      .

Login using SSH

1. Add the following to the SSH configuration file of your user on your own Linux or Mac PC (`~/.ssh/config`) :

```
Host ex3
  User <username>
  HostName dnat.simula.no
  Port 60441
```

2. Connect to the eX3 login node and change password:

```
~ $ ssh ex3
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 4.15.0-122-generic x86_64)

 General information:  http://srl-ex3.simula.no/dokuwiki/doku.php

[...]

 News and planned maintanence for cluster
 ****************************************************************
[...]

Last login: Tue Jul  6 14:33:39 2021 from 84.210.158.35
james@srl-login1:~$ passwd
(current) LDAP Password:
New password:
```

- You can use VPN instead of connecting to `dnat.simula.no`, as explained here:

  https://intranet.simula.no/accessing-simula-hosts-remote-location

- Windows users should configure their PuTTY client, as explained here:

  https://www.ssh.com/academy/ssh/putty/windows

Login using SSH and Public Key authentication

Public Key authentication allows secure login without requiring a password:

- Login to eX3 and add your **public key** to the file `~/.ssh/authorized_keys`:

```
# ~/.ssh/authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAACAQC2D04CXw9AkEgcfv+3sc1Hda2DbLYzW00CDuZLogYp1MYI6o/
ZFmbR7DOJFZ3dTYrfX+fzF55Tl4MmhoI4FxwacANPvWgZqeJEtDtkBzGPwR4Q0dJL5eFVoCT368Q19N4jv8uvl1BUFW/
JFGdM+OSKWf1zkW1dwCPEQltcT9dU09zCJwcLsFWkMXwJ+thFSSM1YTp9tgD0slHdWMpvONvLaCGsZmX/
3IOyknrQWaMmzpF3muOvkHhEl7XwzVQCAVcaLIEuWSz1aoLOLQ0BbdHOz8ZS8UtXelz3L8Qedij0zQhcwu/
lfM1Rtj6tmnPHnEh5JPbdGvInhfKfYozwsj39FcKDmPpr3ZkiDPv0vhc7+yKpU3Ml12K2Owsd9f4ueqn4DBy8UtAW47yp2nRTj9nWZ+H
nzAirMvPkgf3bThJSh5DZzNlJuQ8A9FhyCUVGCKgx6i32ngh+JyuY5Fn0DhQGwXzBQExhIl6qXaedP4iuu02XoZpkOY9TxQL+elNyYJq
IldTfCJSruoeyW9lWqkeZt9FRV6uB+77XQIK0yjBt3nuKY3Ex7W/swzir/
AMdIr9kH9j4QE1uwQzIrJ6dG7FZkdOs7sSiu3/1n3YAffwcZxAmy/jLWfdffWji/FvnUGH/
gPNzRwbwF35JOGoubIidcAJ6UC34PgbGG4pq4bkZAQ== openpgp:0xB429505
7ecdsa-sha2-nistp256
AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBECAIlBfBLDfzw9HuwJrW76jMU5wux4xCGH5fELKW6w6xTNgxl6E
Ba20qIyRp8JVm0/iLEF41vKWT6tAFdBxcBA= james@g001
```

- For further details, see, for example, the following tutorial on connecting to GitHub with SSH using Public Key authentication: https://docs.github.com/en/github/authenticating-to-github/connecting-to-github-with-ssh.

Beginner’s guide to eX³

Getting started on eX³:

Software

Basic examples: Compiling and running a C application, and using a Python interpreter

C example:

```
james@srl-login1:~$ cat hello.c
/* An example of a basic C application. */
#include <stdio.h>

int main(void) {
  printf("Hello, world!\n");
  return 0;
}
james@srl-login1:~$ gcc –o hello hello.c
james@srl-login1:~$ ./hello
Hello, world!
```

Python example:

```
james@srl-login1:~$ python3
Python 3.6.9 (default, Jan 26 2021, 15:33:00)
[GCC 8.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> print('Hello, World!')
Hello, World!
>>> <Ctrl+d> to exit
```

- You may compile your code on the login node, but you should only run VERY SHORT programs for testing purposes. All other programs should be submitted to eX3 using the Slurm queueing system (to be described later).

Use prebuilt software with **Environment Modules**

Environment modules is a commonly used system for managing software libraries and applications on HPC clusters:

- Loading a *module* changes a user's environment to make the relevant software package available
- Software can be installed once and shared by several users
- Different versions of the same software can be maintained

Basic commands:

- **`module avail`** – List modules that are installed on the system and which can be loaded
- **`module list`** – Display modules that are currently loaded
- **`module load`** – Load a module by modifying the user's environment
- **`module unload`** – Unload a module by resetting the user's environment
- For more details, see https://modules.readthedocs.io/en/latest/

Example: Using a different C compiler

```
james@srl-login1:~$ module use /cm/shared/ex3-modules/0.6.1/modulefiles
james@srl-login1:~$ module load intel/compiler/64/2019/19.0.5
james@srl-login1:~$ icc --version
icc (ICC) 19.0.5.281 20190815
Copyright (C) 1985-2019 Intel Corporation.  All rights reserved.

james@srl-login1:~$ icc –o hello hello.c
james@srl-login1:~$ ./hello
Hello, World!
```

Many other compilers are also available—check the list of available modules.

Compilers and compiler suites (amd64/x86_64)

```
torel@srl-login1:~$ module avail gcc
------------------------------------ /cm/shared/ex3-modules/0.6.1/modulefiles
------------------------------------
gcc-8.4.0  gcc-9.2.0  gcc-10.1.0

---------------------------------------------- /cm/local/modulefiles
----------------------------------------------
gcc/8.2.0  gcc/64/4.0.0rc5

---------------------------------------------- /cm/shared/modulefiles
----------------------------------------------
gcc/9.1.1  gcc/10.0.0-20190609  gcc/10.3.0           gcc/cu116/11.3.0  gcc/cu117/12.2.0  gcc9/9.3.0
gcc/9.2.0  gcc/10.1.0           gcc/11.1.0           gcc/cu116/12.1.0  gcc5/5.5.0
gcc/9.3.0  gcc/10.2.0           gcc/11.2.0(default)  gcc/cu117/11.3.0  gcc8/8.4.0
```

```
torel@srl-login1:~$ module avail intel
---------------------------------------------- /cm/shared/modulefiles
----------------------------------------------
intel-cluster-runtime/ia32/2019.6(default)              intel/oneapi/2021.2/intelfpgadpcpp/2021.2.0
intel-cluster-runtime/ia32/2020.2                       intel/oneapi/2021.2/intelfpgadpcpp/latest
intel-cluster-runtime/intel64(default)                  intel/oneapi/2021.2/itac/2021.2.0
intel-cluster-runtime/intel64/2019.6(default)           intel/oneapi/2021.2/itac/latest
intel-cluster-runtime/intel64/2020.2                    intel/oneapi/2021.2/mkl/2021.2.0
intel/compiler/32/2019/19.0.5                           intel/oneapi/2021.2/mkl/latest
.
```

Compilers and compiler suites (amd64/x86_64)

```
torel@srl-login1:~$ module avail nvidia/nvhpc
------------------------------------------- /cm/shared/modulefiles
-------------------------------------------
nvidia/nvhpc-byo-compiler/20.11          nvidia/nvhpc-nompi/20.11          nvidia/nvhpc/20.11
nvidia/nvhpc-byo-compiler/21.3           nvidia/nvhpc-nompi/21.3           nvidia/nvhpc/21.3
nvidia/nvhpc-byo-compiler/21.9           nvidia/nvhpc-nompi/21.9           nvidia/nvhpc/21.9
nvidia/nvhpc-byo-compiler/22.3(default)  nvidia/nvhpc-nompi/22.3(default)  nvidia/nvhpc/22.3(default)
torel@srl-login1:~$
```

```
torel@srl-login1:~$ module avail gcc/cu11
------------------------------------------- /cm/shared/modulefiles
-------------------------------------------
gcc/cu116/11.3.0  gcc/cu116/12.1.0  gcc/cu117/11.3.0  gcc/cu117/12.2.0
```

```
torel@srl-login1:~$ module avail clang/cu11

                          Coming soon!
```

Compilers and compiler suites (arm64)

```
torel@srl-login1:~$ srun -p armq -N 1 -n 256 --pty /bin/bash
srun: job 396523 queued and waiting for resources
srun: job 396523 has been allocated resources

torel@n005:~$ uname -ar
Linux n005 5.4.0-126-generic #142~18.04.1-Ubuntu SMP Thu Sep 1 16:27:32 UTC 2022 aarch64 aarch64 aarch64 GNU/Linux
torel@n005:~$

torel@n005:~$ module avail arm
------------------------------------------- /cm/shared/modulefiles
-------------------------------------------
arm/aor/gcc/2015_11_17  arm/armclang/21.0    arm/armclang/22.0  arm/armpl/21.0.0_gcc-10.2  arm/gcc/8.2.0
arm/armclang/19.3       arm/armclang/21.1.1  arm/ArmIE/19.2     arm/forge/21.0.2
```

```
torel@n005:~$ module avail gcc
.
.
------------------------------------------- /cm/shared/modulefiles
-------------------------------------------
gcc/9.1.1  gcc/9.2.0  gcc/9.3.0  gcc/10.1.0  gcc/10.2.0  gcc/10.3.0(default)  gcc/11.1.0  gcc/11.2.0
torel@n005:~$
```

```
torel@srl-login1:~$ srun -p a40q -N 1 -n 128 --pty /bin/bash
.
torel@n009:~$ module avail nvidia/nvhpc
```

Example: Loading pre-installed Python modules (Fenics)

```
james@srl-login1:~$ module use /cm/shared/ex3-modules/0.6.1/modulefiles
james@srl-login1:~$ module load python-3.7.4 python-numpy-1.19.2
Loading python-numpy-1.19.2
  Loading requirement: bzip2-1.0.8 xz-5.2.5 ncurses-6.1 readline-8.0
libffi-3.2.1 openssl-1.1.1c sqlite-3.31.1 python-3.7.4
    libgfortran-5.0.0 libstdcxx-6.0.28 openblas-0.3.12 lapack-3.9.0
fftw-3.3.8
james@srl-login1:~$ python3
Python 3.7.4 (default, Apr  1 2021, 09:45:00)
[GCC 7.5.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import numpy
>>>
```

Example: Creating python virtual environments

```
torel@srl-login1:~$ which python2 python3.6 python3.7 python3.8
python3.9 python3.10 python3.11
/usr/bin/python2
/usr/bin/python3.6
/usr/bin/python3.7
/usr/bin/python3.8
/usr/bin/python3.9
/usr/bin/python3.10
/usr/bin/python3.11
torel@srl-login1:~$
```

Example: Creating python virtual environments

```
torel@srl-login1:~$ python3.11 -m venv D1/mypy311env

torel@srl-login1:~$ source D1/mypy311env/bin/activate

(mypy311env) torel@srl-login1:~$ which python3
/home/torel/D1/mypy311env/bin/python3
(mypy311env) torel@srl-login1:~$ python3 -V
Python 3.11.0rc2
(mypy311env) torel@srl-login1:~$ pip list
Package    Version
---------- -------
pip        22.2.2
setuptools 63.2.0
(mypy311env) torel@srl-login1:~$ pip install numpy
Collecting numpy
  Using cached numpy-1.23.3-cp311-cp311-
manylinux_2_17_x86_64.manylinux2014_x86_64.whl (17.1 MB)
Installing collected packages: numpy
Successfully installed numpy-1.23.3
(mypy311env) torel@srl-login1:~$ deactivate
```

You may still need to install your own software...

Need help installing new software?

- Our eX3 system administrator, Tore Larsen, can sometimes help to install new software. Please send an email to: ex3-helpdesk@simula.no.

- We have also collected a lot of software installation scripts in a GitHub repository: https://github.com/jamtrott/ex3modules.

Beginner’s guide to eX³

Getting started on eX³:

Scheduling jobs

eX³ uses the Slurm[1] workload manager for allocating resources and scheduling jobs on the cluster

Parallel and distributed computing terminology:

- **Process:** An instance of a computer program with its own memory address space
- **Distributed memory:** Multiple processors, where each processor has its own *private* memory
- **Shared memory:** Multiple processors sharing a single memory (typically within a single node)
- **Message passing interface (MPI):** A standardised programming model for distributed memory parallel computing

Slurm Terminology:

- **Node:** A machine belonging to a cluster
- **Task:** A process that is assigned to a node by the Slurm workload manager. There can be ***multiple tasks per node***, but each task is a *separate process*.
- **CPU:** A CPU core on a node with multiple cores. There can be ***multiple CPUs per task***, for programs that use *multithreading*.
- **Partition:** A subset/grouping of nodes with same config, also called queue

[1] https://www.schedmd.com/

eX3 uses the Slurm workload manager for allocating resources and scheduling jobs on the cluster

Basic Slurm commands:

- **`salloc`** – Allocate resources, such as CPUs, GPUs, or entire compute nodes
- **`srun`** – Execute an application using allocated resources
- **`sbatch`** – Submit a *batch script* to execute commands when resources become available
- **`scancel`** – Free allocated resources or halt a running job
- **`squeue`** – Show current and queued allocations for all users
- **`sinfo`** – Show current partitions on systems

Documentation:

- For more details, see https://slurm.schedmd.com/tutorials.html
  - Quick start guide: https://slurm.schedmd.com/quickstart.html
  - Introduction to Slurm: http://www.youtube.com/watch?v=NH_Fb7X6Db0&feature=relmfu

eX3 uses the Slurm workload manager for allocating resources and scheduling jobs on the cluster

```
torel@srl-login1:~$ module avail slurm

torel@srl-login1:~$ module load slurm/20.02.7
```

Example: Use *sbatch* to submit a job on AMD Epyc (*defq*)

```
 1 | #!/bin/bash
 2 | # Basic usage:
 3 | #
 4 | #  $ sbatch hello.sbatch
 5 | #
 6 | # The job is queued and starts as soon as resources are available. The
 7 | # script is then executed on one of the allocated tasks, and standard
 8 | # output and standard error streams will be redirected to files that
 9 | # are prefixed by the job ID and job name. Commands prefixed with
10 | # `srun' are executed on every task acquired by the job allocation.
11 | #
12 | # The sbatch options below allocate a single task on a single node,
13 | # using a single CPU core with a one-hour time limit. To override
14 | # these defaults, you can also supply sbatch options on the command
15 | # line. For example:
16 | #
17 | #  $ sbatch --cpus-per-task=32 --time=02:00:00 hello.sbatch
18 |
19 | #SBATCH --job-name="hello"
20 | #SBATCH --partition=defq
21 | #SBATCH --time=0-01:00:00
22 | #SBATCH --nodes=1
23 | #SBATCH --ntasks=1
24 | #SBATCH --cpus-per-task=1
25 | #SBATCH --output=%j-%x-stdout.txt
26 | #SBATCH --error=%j-%x-stderr.txt
27 |
28 | echo ”1. Hello from $(hostname)"      # This command is run by ONE task
29 | srun echo ”2. Hello from $(hostname)" # This command is run by EVERY task
```

1. These are *shell scripts*, so you can use ordinary Unix shell programming techniques.

2. Lines beginning with

   `#SBATCH <OPTION>`

   can be used to provide options to the Slurm workload manager.

3. Use **`srun`** to execute a command on every task that has been allocated for the job.

4. Standard output and error streams are redirected to files in the current working directory.

Example: Use *sbatch* to submit a job on AMD Epyc (*defq*)

```
james@srl-login1:~$ sbatch --partition armq hello.sbatch
Submitted batch job 167194

james@srl-login1:~$ ls
167194-hello-stderr.txt    167194-hello-stdout.txt    hello.sbatch

james@srl-login1:~$ cat 167194-hello-stdout.txt
1. Hello from n006
2. Hello from n006

james@srl-login1:~$ sbatch -p armq --nodes=2 --ntasks=2 hello.sbatch
Submitted batch job 167195

james@srl-login1:~$ cat 167195-hello-stdout.txt
1. Hello from n006
2. Hello from n006
2. Hello from n006
```

1. These are *shell scripts*, so you can use ordinary Unix shell programming techniques.

2. Lines beginning with

   #SBATCH <*OPTION*>

   can be used to provide options to the Slurm workload manager.

3. Use **srun** to execute a command on every task that has been allocated for the job.

4. Standard output and error streams are redirected to files in the current working directory.

Beginner’s guide to eX³

Getting started on eX³:

Best practices

Best practices for allocating resources on eX³

Please follow these guidelines for resource allocation:

1. Always allocate resources using Slurm—otherwise, you risk interfering with the work of other users
2. Do not let allocated resources stand idle or go unused—others could use those resources instead
3. Do not run CPU- or memory-demanding workloads on the login node
4. Set a reasonable *time limit* when allocating resources
   - use the `--time` option with `salloc`/`srun`/`sbatch`
5. Only use the CPU cores or GPUs that you have allocated
   - For example, do not spawn more threads than the number of CPU cores that were allocated.
6. Do not login to compute nodes with `ssh`.
   - Interactive jobs can be run with `srun --pty`, but this is rarely needed.

Managing files and storage

Please follow these guidelines for storing data on eX3:

- **`/global/D1`** is a global, parallel file system (BeeGFS), which ***should be used for most storage***.
  - Every user has a home directory at **`/global/D1/homes/<user>`**
  - For data related to a project or shared by multiple users, project directories can be created under **`/global/D1/projects`**. Contact the system administrator (ex3-helpdesk@simula.no) if you would like to have a project directory.
- **`/home/<user>`**, the user's home directory, is a network file system with ***limited capacity*** and should only be used for very small amounts of data (e.g., source code, but *not* large data sets).
- **`/work`** is a fast, local storage on each node, only to be used for temporary data.
  - Create your own folder, e.g. «mkdir -p /work/$USER» in sbatch script
- **`/work/docker`** - Docker images are stored on the local scratch on nodes. You could test if docker image exists or not, and if not install it on the fly through your sbatch script.
- **`/tmp`** is a on localdisk, relatively small and should not be used for TEMPDIR, TMPDIR, etc.

Managing files and storage

Transfering files to eX³:

Using SSH tunnel:

$ **rsync -avrz -e "ssh -p 60441" <sourcedir> torel@dnat.simula.no:D1/**

If you are logged on with VIA VPN or on-premise:

$ **rsync -avrz <sourcedir> torel@srl-login1.ex3.simula.no:D1/**

Note! With MobaXterm you can will get an Explorer view of remote filesystem.

MacOSX is analogous to Linux/Unix.

Where can I get help?

Got questions?

- If you have questions, first take a look at the eX3 wiki: http://wiki.ex3.simula.no/
- If you cannot find the answer to your question there, please contact the eX3 system administrator, Tore Larsen, by email to ex3-helpdesk@simula.no.

The eX³ project is hosted by Simula Research Laboratory and funded by the Research Council of Norway

Whenever you publish research based on the eX3 infrastructure, please notify us by email to ex3-contact@simula.no.

Also, please acknowledge eX3 in your paper:

The research presented in this paper has benefited from the Experimental Infrastructure for Exploration of Exascale Computing (eX3), which is financially supported by the Research Council of Norway under contract 270053.

https://www.ex3.simula.no/publications

The Research Council of Norway

With funding from
The Research Council of Norway

Other eX³ resources

Other resources such as

eX³-project-consortium-presentation
Graphcore 101

are available please contact me torel@simula.no

If anyone would like to contribute in creating a new Wiki/Sphinx dokuserver, please contact me.

The Research Council of Norway

With funding from
The Research Council of Norway