CBS COPENHAGEN BUSINESS SCHOOL
HANDELSHØJSKOLEN

HIGH PERFORMANCE COMPUTING AT CBS

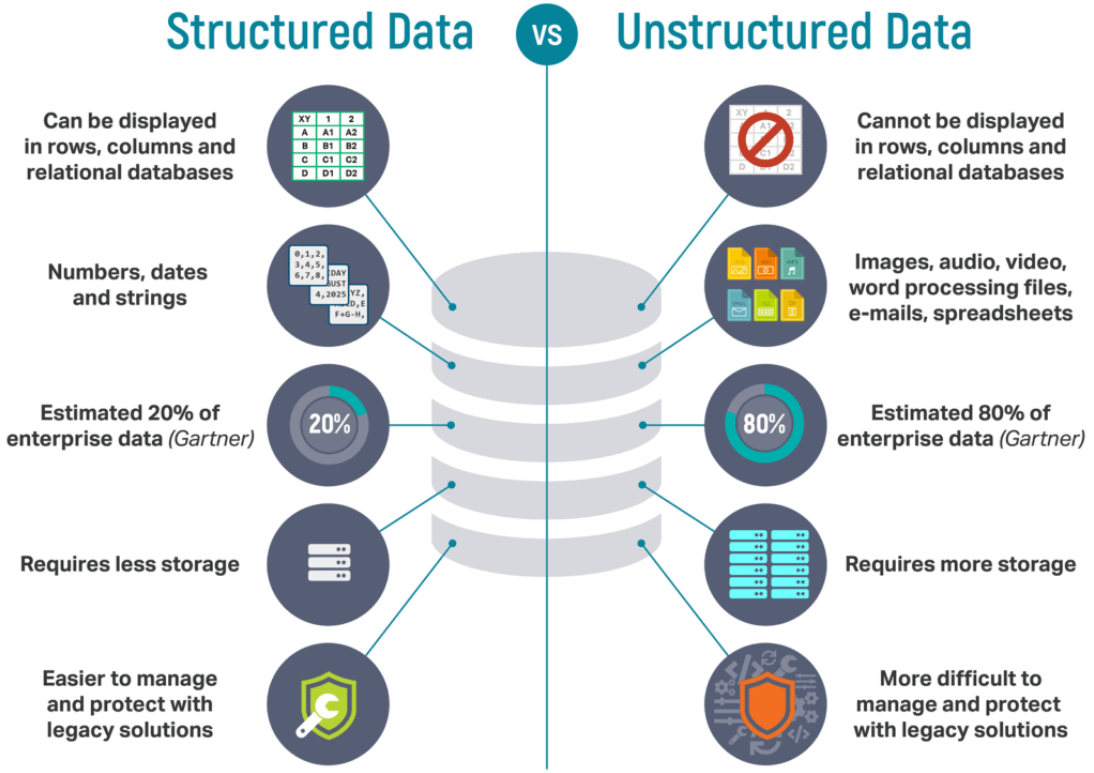
New cloud computing possibilities for researchers & students

Kristoffer Gulmark Poulsen & Lars Nondal
Research Data Management Support
CBS Library

EQUIS ACCREDITED · AACSB ACCREDITED · ASSOCIATION AMBA ACCREDITED · CEMS · PIM

About You?

1. What **type and size** of data do you work with?
2. What programming languages do you work in (e.g. R, Python..)?
3. Are you familiar with parallel programming?
4. Are you familiar with high performance computing?



Use Cases from CBS

HPC might be useful when:

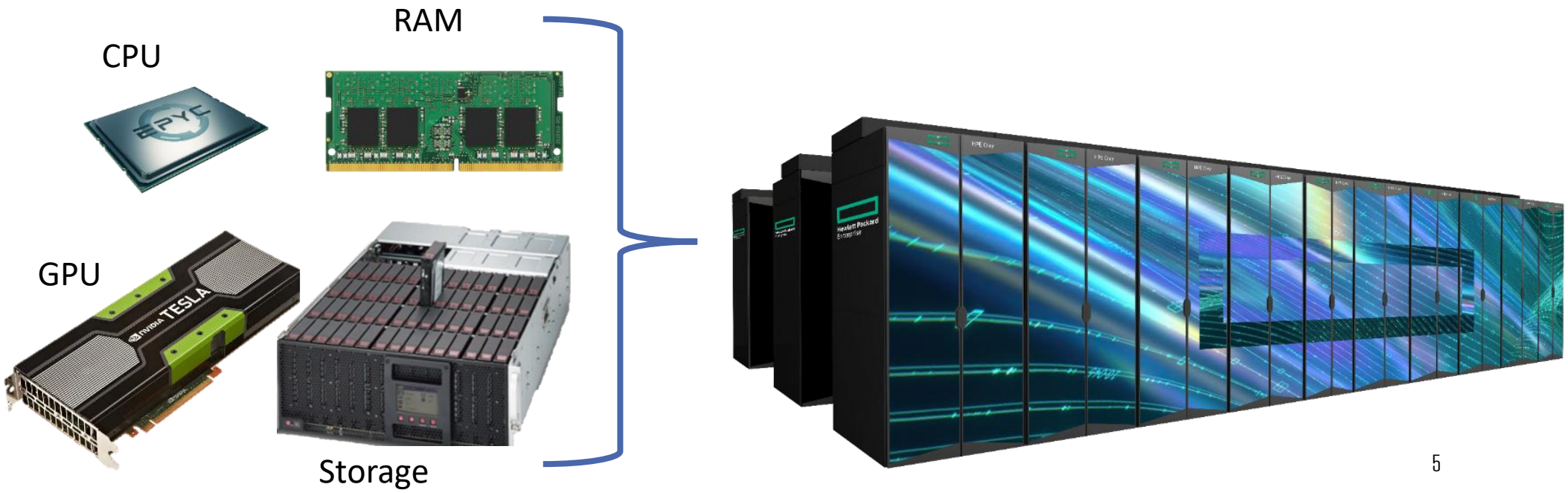
- Applying ML/AI
- Running other simulation and resampling techniques
- Working with large datasets
- My laptop runs out of memory
- My workflow is running very slow

CBS

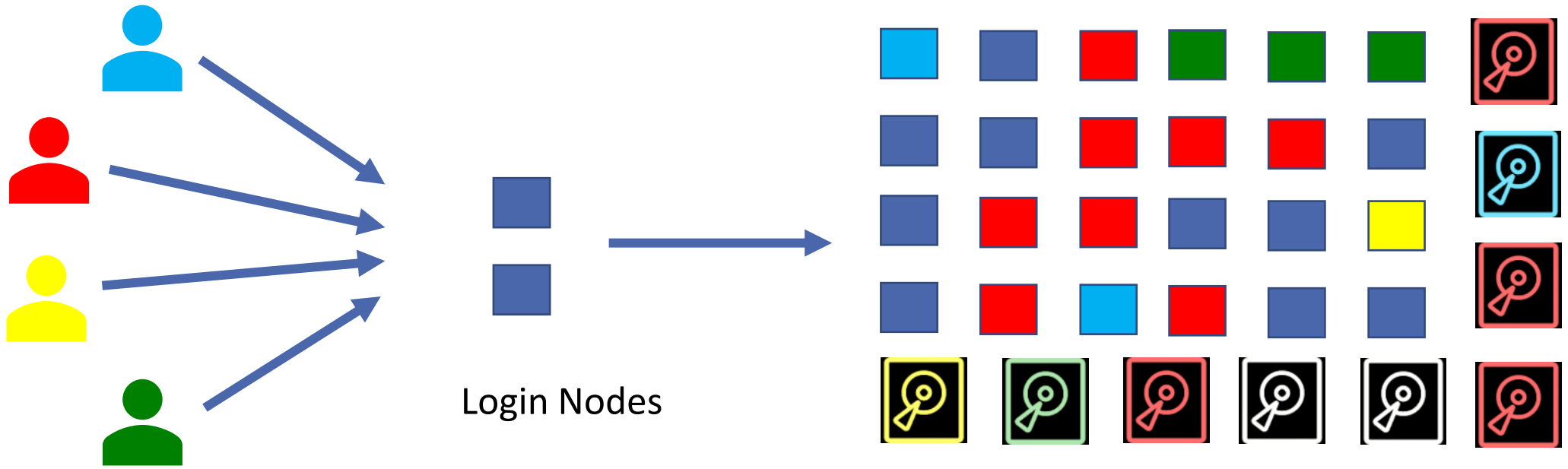
What is High Performance Computing (supercomputer)?

Hardware

- **Core**: Processing unit on a single machine.
- **Node**: A single machine.
- **Cluster**: Network of multiple nodes.

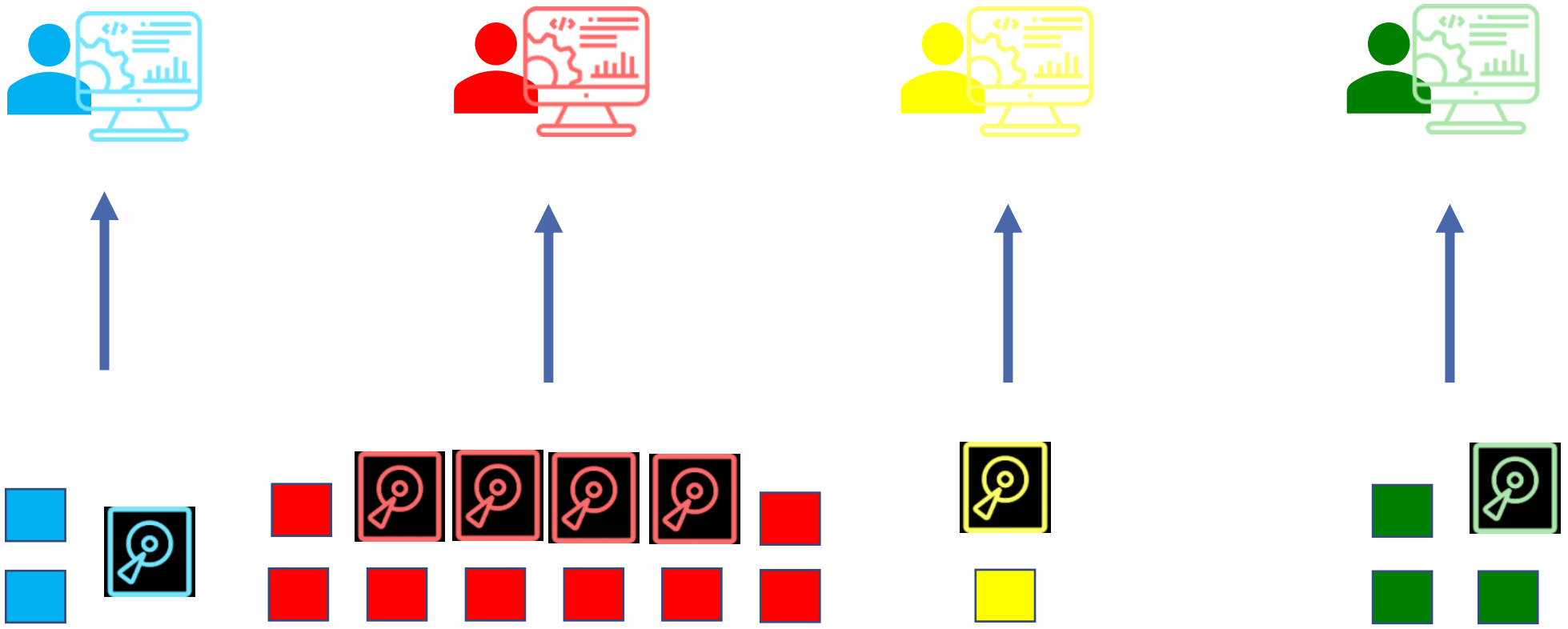


Accessing an HPC...



CBS

Accessing an HPC...

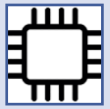


National HPC facilities

- Collaboration between Universities and DeiC (Danish e-Infrastructure Cooperation)

Type 1 (SDU, AAU)

Interactive HPC



Type 2 (AU,KU & DTU)

Throughput HPC

Type 3 (SDU)

Large Memory HPC

Type 5 (EuroHPC Consortium)

LUMI Capability HPC

https://www.deic.dk/en/supercomputing/national-hpc-facilities

CBS

Type 1: Interactive HPC

Cloud-based (HPC) systems (e.g. similar to google colab, amazon aws)

User friendly with Graphical User Interface (GUI).

Lots of preinstalled software (R, Python, Stata & Matlab)

Collaborative projects – work & share files with others

GDPR-Compliant

Access with university credentials from https://cloud.sdu.dk

- xxx@student.cbs.dk
- xxx@cbs.dk
- 1000 DKK Free credit.

Type 1 (SDU, AAU)

Interactive HPC

Type 1: SDU

- **CPU** resources
- GUI based
- Wide range of applications
- Slurm and Spark Cluster

Type 1 (SDU, AAU)

Interactive HPC

Name	vCPU	Memory (GB)	GPU	Price
DeiC Interactive HPC (SDU): u1-standard				
u1-standard-1	1 (Intel Xeon Gold 6130)	6	None	0,07 DKK/hour
u1-standard-2	2 (Intel Xeon Gold 6130)	12	None	0,16 DKK/hour
u1-standard-4	4 (Intel Xeon Gold 6130)	24	None	0,33 DKK/hour
u1-standard-8	8 (Intel Xeon Gold 6130)	48	None	0,67 DKK/hour
u1-standard-16	16 (Intel Xeon Gold 6130)	96	None	1,36 DKK/hour
u1-standard-32	32 (Intel Xeon Gold 6130)	192	None	2,74 DKK/hour
u1-standard-64	64 (Intel Xeon Gold 6130)	384	None	5,49 DKK/hour

CBS

Support at CBS

Local CBS support

- Lars Nondal & Kristoffer Gulmark Poulsen
- Contact: rdm@cbs.dk or directly to Kristoffer (kgp.lib@cbs.dk)

User support: Advising and granting resources, technical problems.

Consultation: Code development etc.

Teaching: "High Performance Computing", "HPC & Parallel Programming in R and Python" and "Train your ML/AI Model on GPUs".

Documentation and Tutorials: https://cbs-hpc.github.io/

CBS

CBS COPENHAGEN BUSINESS SCHOOL
HANDELSHØJSKOLEN

PARALLEL PROGRAMMING

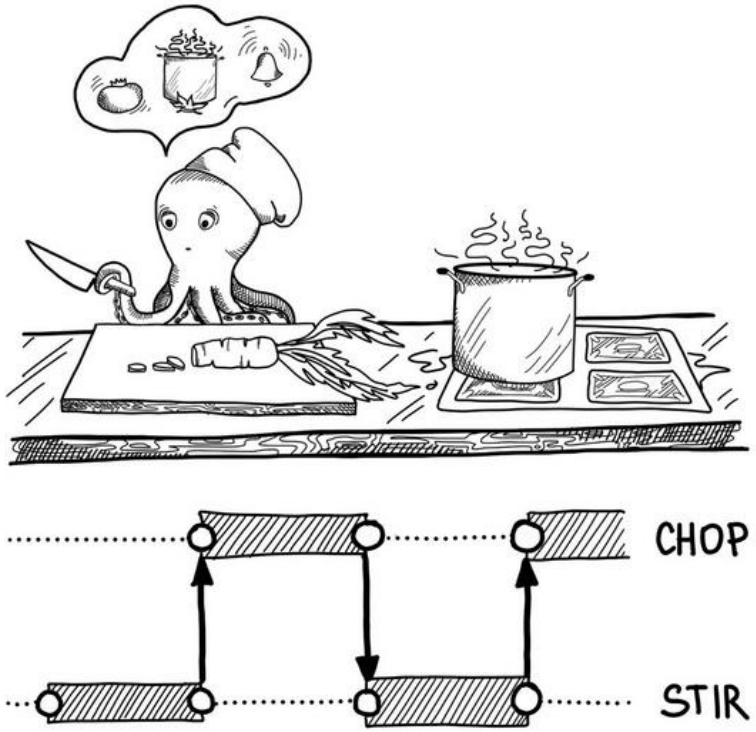
Kristoffer Gulmark Poulsen & Lars Nondal
CBS

EQUIS ACCREDITED
AACSB ACCREDITED
ASSOCIATION AMBA ACCREDITED
CEMS
PIM

Working on Laptop vs HPC

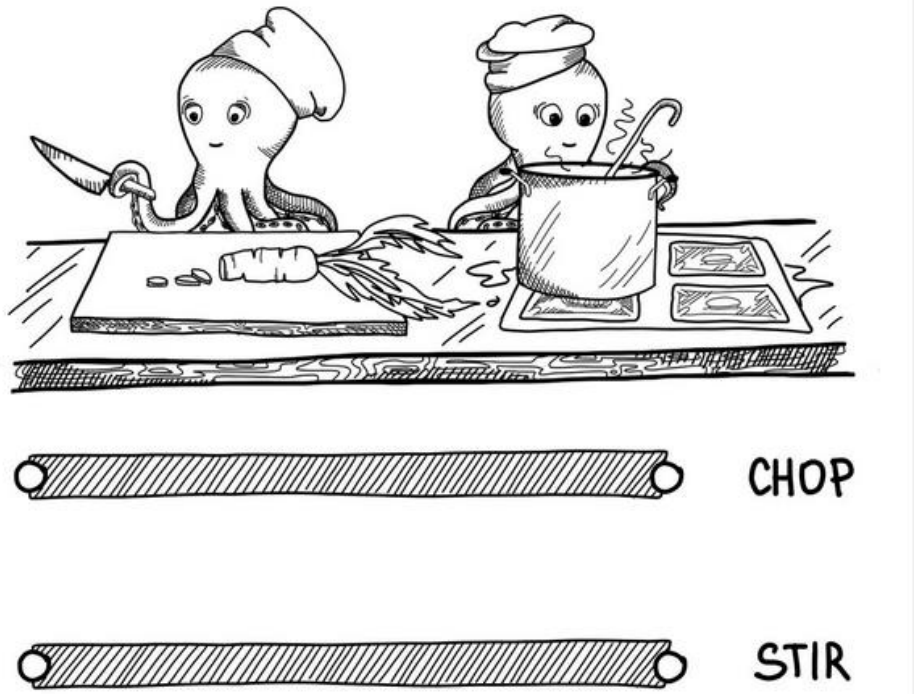
Sequential Computing

- Single core processor
- Multiple tasks which runs overlapping but **not** at same time
- Synchronous tasks



Parallel Computing

- Multi-core processor
- Multiple tasks which runs overlapping.
- Synchronous/Asynchronous



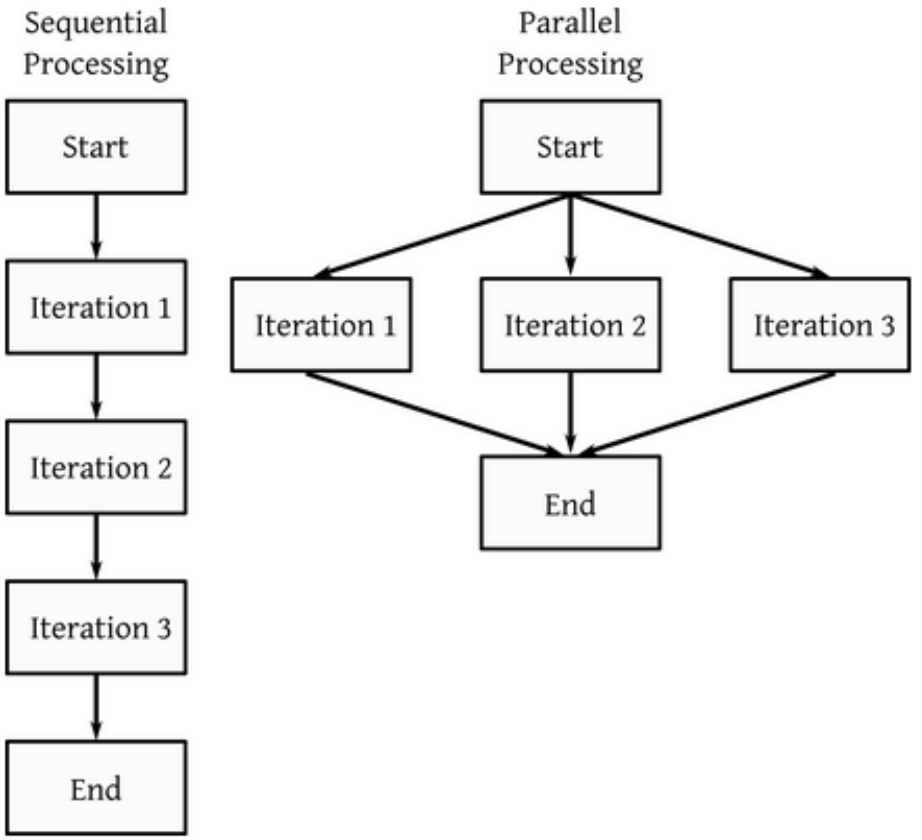
CBS

Parallel Programming

Single Instruction, Multiple Data (SIMD) - single thread/processor where same each processing unit performs the same instruction on different data. **Used in Vectorization**.

Shared Memory Parallelism (SMP) work is divided between **multiple threads/processes** running on a **single machine.**

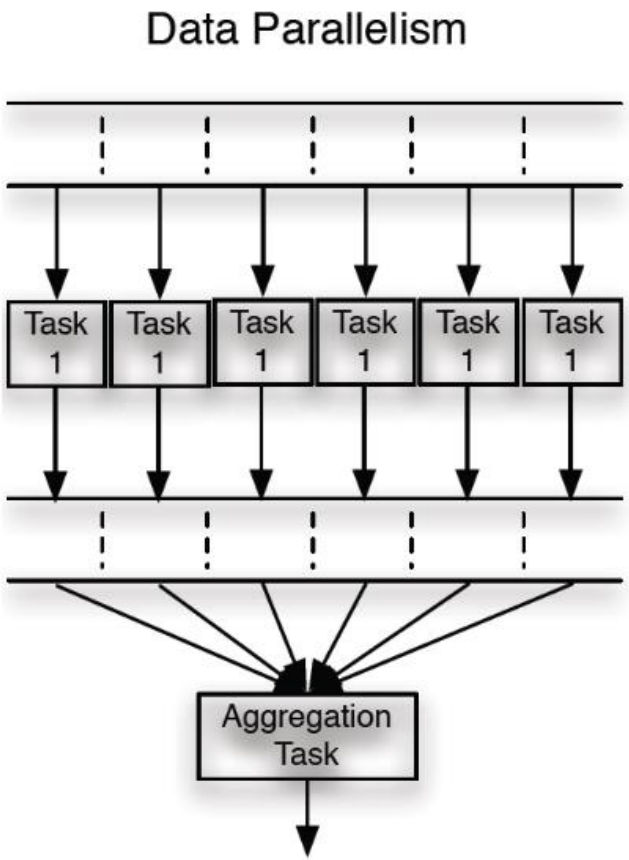
Distributed Memory Parallelism work is divided between **multiple machines** with its own private memory.

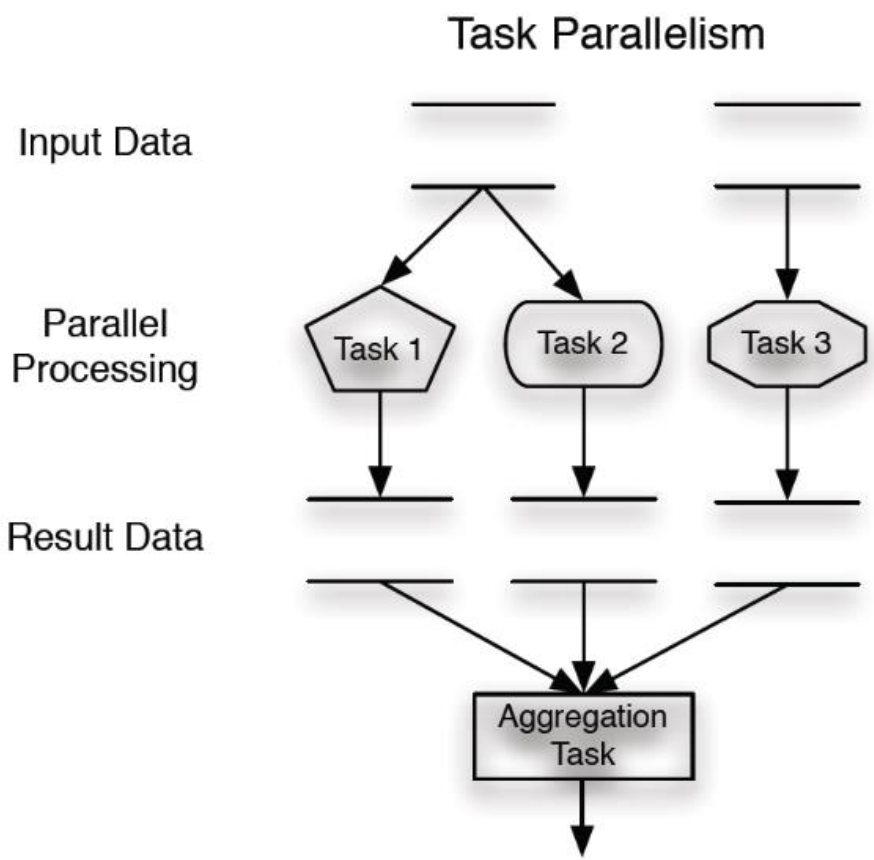


Parallel Programming

Data vs Task Parallelism

Multi-Threading vs Multi-Processing



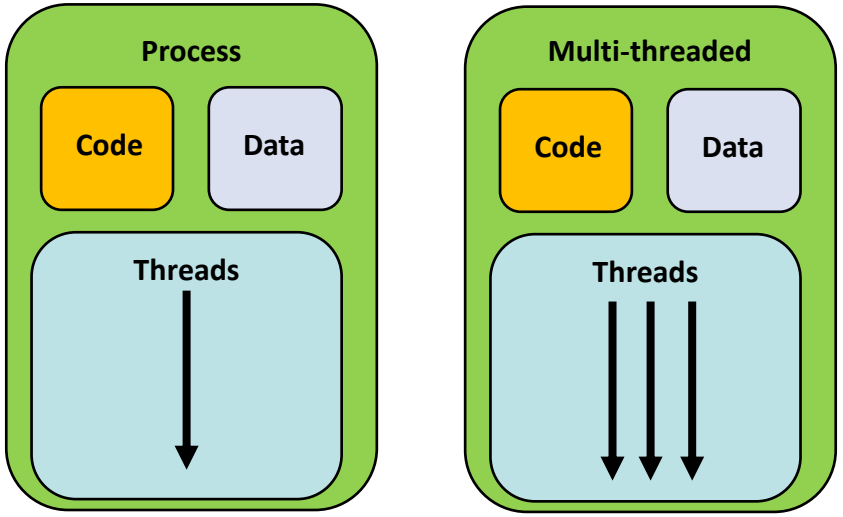


Multi-Threading

Threads are multiple paths of execution within a single process.

- Appears as a single process.
- Not **hyperthreading**: a single core appears as two cores.

Python and **R** are examples of single-threaded programming languages.




```
top - 15:12:02 up 2 days, 54 min,  0 users,  load average: 6.42, 6.45, 6.45
Tasks:  10 total,   1 running,   9 sleeping,   0 stopped,   0 zombie
%Cpu(s): 11.0 us,  0.3 sy,  0.0 ni, 88.7 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem : 385583.7 total, 193583.0 free, 102124.0 used,  89876.6 buff/cache
MiB Swap:   8192.0 total,   4461.5 free,   3730.5 used. 280235.0 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
  243 ucloud    20   0 3970780 962704  74288 S 278.1   0.2   0:44.50 rsession
  202 rstudio+  20   0  182200  18268  14724 S   0.3   0.0   0:01.00 rserver
    1 ucloud    20   0    6896   3428   3196 S   0.0   0.0   0:00.05 start-rstu+
    7 root      20   0   10420   4920   4376 S   0.0   0.0   0:00.00 sudo
    8 root      20   0     200      4      0 S   0.0   0.0   0:00.01 s6-svscan
   37 root      20   0     200      4      0 S   0.0   0.0   0:00.00 s6-supervi+
  198 root      20   0     200      4      0 S   0.0   0.0   0:00.00 s6-supervi+
  265 ucloud    20   0    2492    580    512 S   0.0   0.0   0:00.01 sh
  271 ucloud    20   0    8168   4904   3408 S   0.0   0.0   0:00.01 bash
  273 ucloud    20   0   10032   3824   3316 R   0.0   0.0   0:00.12 top
```

CBS

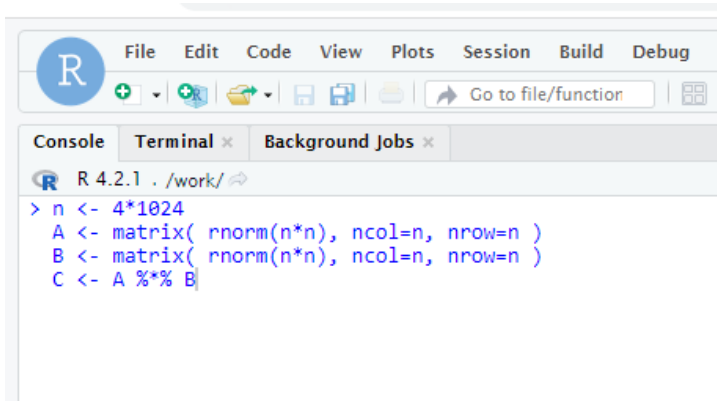
Multi-Threading & SIMD in Python and R

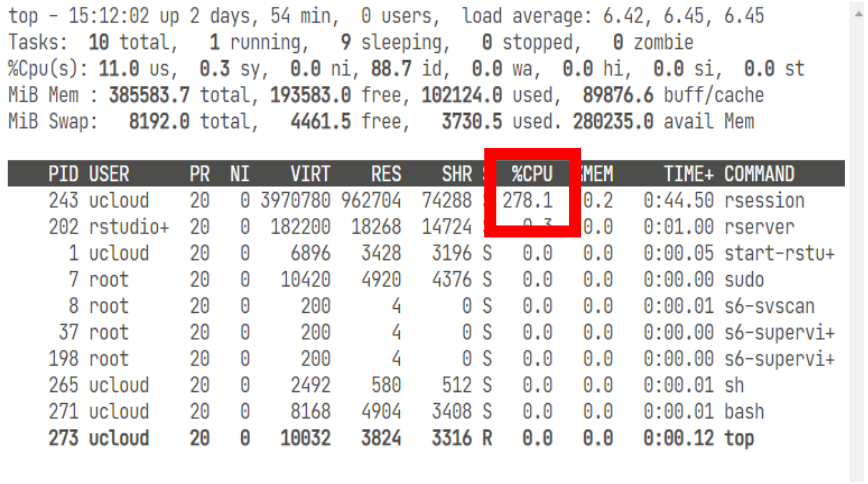
Multi-Threading and **SIMD** is achieved through external libraries written in other languages (e.g. C, C++, Fortran) that run multi-threaded.

Basic Linear Algebra Subprograms (BLAS) – Allows vectorized calculations in R and Python.

Python e.g. NumPy and Pandas.

R e.g. some built-in functions.





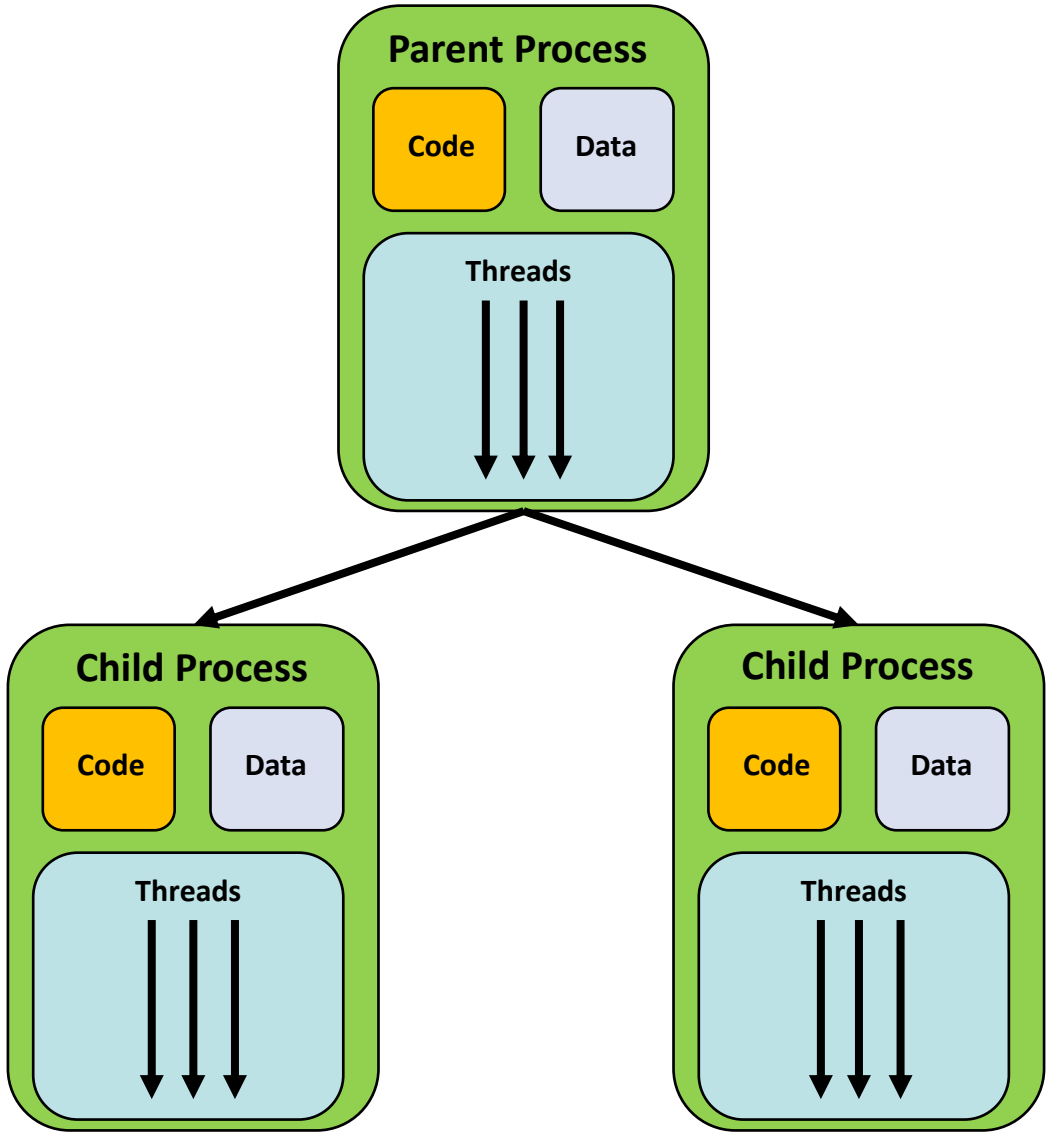
CBS

Multi-Processing

R packages: ***parallel***, ***doParallel***, ***future*** and ***Tidymodels***…

Python libraries: ***multiprocessing***, ***threading***, ***Joblib, Dask and Ray…***

Other Frameworks***: Tensorflow, Torch and Apache Spark…***



CBS

CBS COPENHAGEN BUSINESS SCHOOL
HANDELSHØJSKOLEN

UCLOUD - TYPE-1 INTERACTIVE HPC

Kristoffer Gulmark Poulsen & Lars Nondal
CBS

EQUIS ACCREDITED AACSB ACCREDITED ASSOCIATION AMBA ACCREDITED CEMS PIM

Getting Started

Tutorials and Documentation https://cbs-hpc.github.io/

Getting Started with HPC (UCloud)

Use Conda to manage R-packages and Python-libraries

Batch Processing on Ucloud

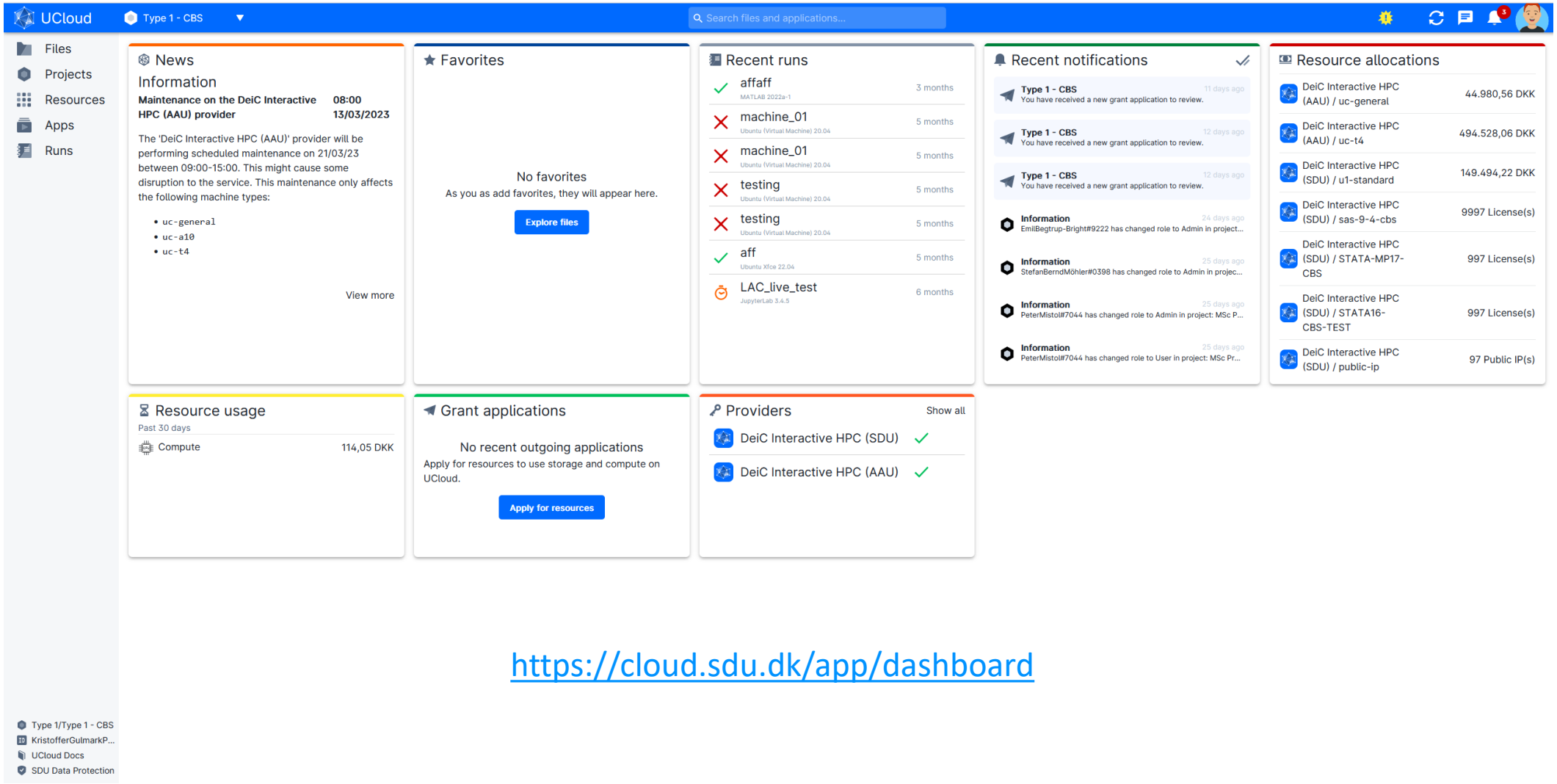
Rsync - Large data transfer to UCloud

Synchronization on UCloud

Tutorials and Documentation

CBS

UCloud Dashboard



https://cloud.sdu.dk/app/dashboard

CBS

User Pit Falls

Wait in queue when starting job- at “rush hours” you might have to wait in queue.

The job stopped while I was working - Remember to set enough hours for the job!

My results disappeared - Remember to work in the right folder!

Remember to stop the application after use!

CBS

CBS COPENHAGEN BUSINESS SCHOOL
HANDELSHØJSKOLEN

QUESTIONS ?

Kristoffer Gulmark Poulsen & Lars Nondal
CBS

EQUIS ACCREDITED · AACSB ACCREDITED · ASSOCIATION AMBA ACCREDITED · CEMS · PIM