HPC at UNIGE

Get familiar with the Baobab cluster

Yann Sagon

Plan

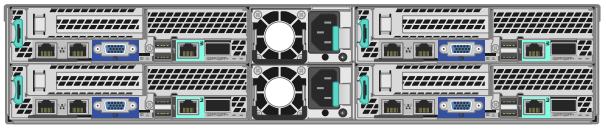
- Infrastructure
- Resources
- Procedure
- Good practices
- Demo

Infrastructure

What is a HPC cluster?

- A very big computer?
- A very fast computer?
- A way to compute quickly?
- Many computers?

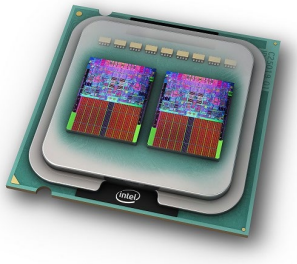
The Baobab Cluster (CPU nodes)



4 nœud (nodes) par chassis

2 cpus physiques par nodes

Intel i7 2 – 4 coeurs



8-14 cœurs (cores) par cpus

4Go de mémoire par coeur

CPU
Central
Processing
Units

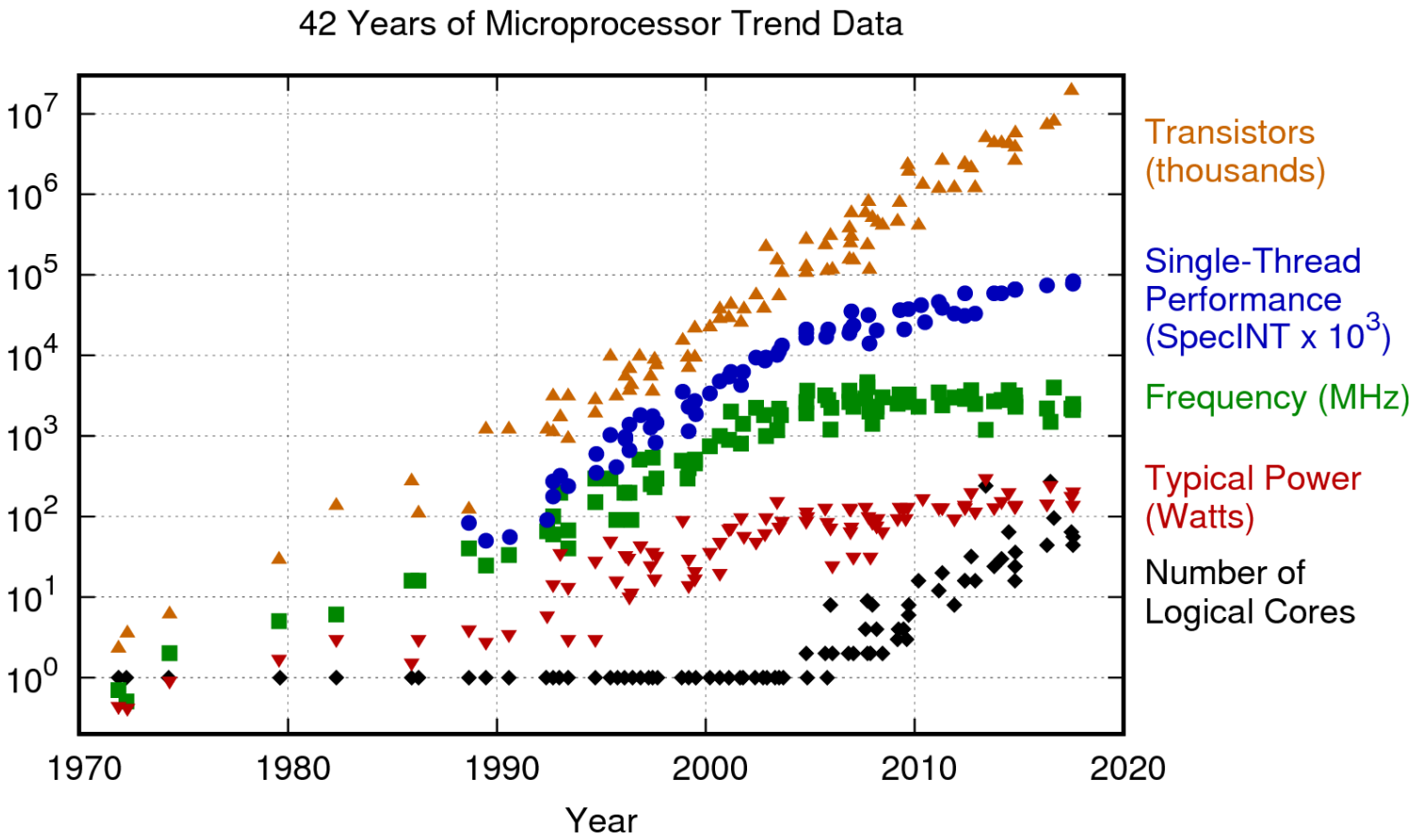
- In modern HPC: 1 CPU = 1 core
- Baobab: ~3700 cores and 19TB RAM

The Baobab Cluster (GPGPU nodes)

GPGPU
General-purpose
Computing on
Graphical
Processing
Units

- Providers: mainly NVIDIA with CUDA
- Two types of GPGPU available on Baobab:
  - Single precision: TitanXp 3840 cores / 12GB RAM > Purpose: TensorFlow
  - Double Precisision: P100 3584 cores / 16GB RAM > Purpose: Numerical modelisation (meteo, chemical, physical)

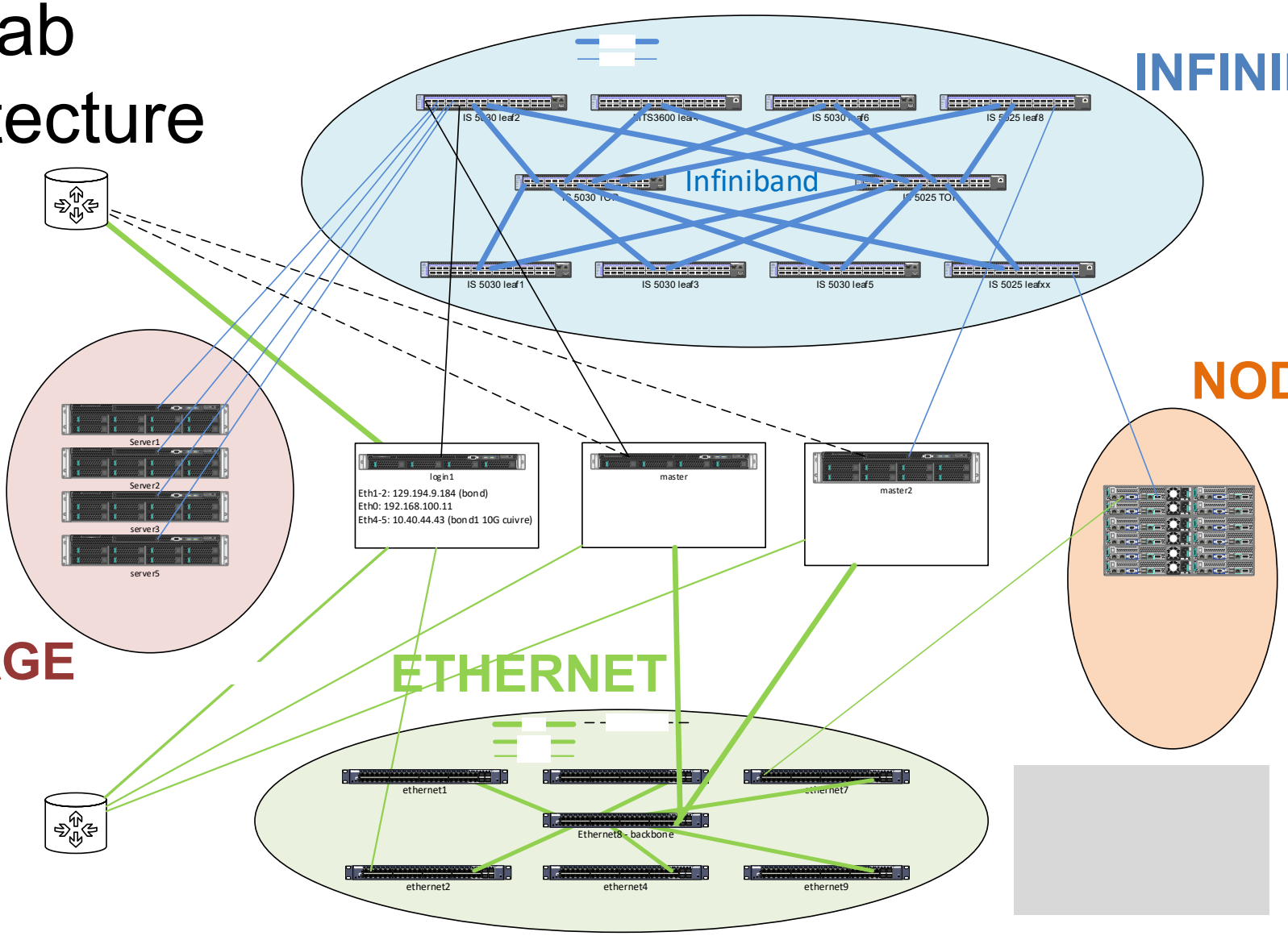
Evolution of CPUs over time



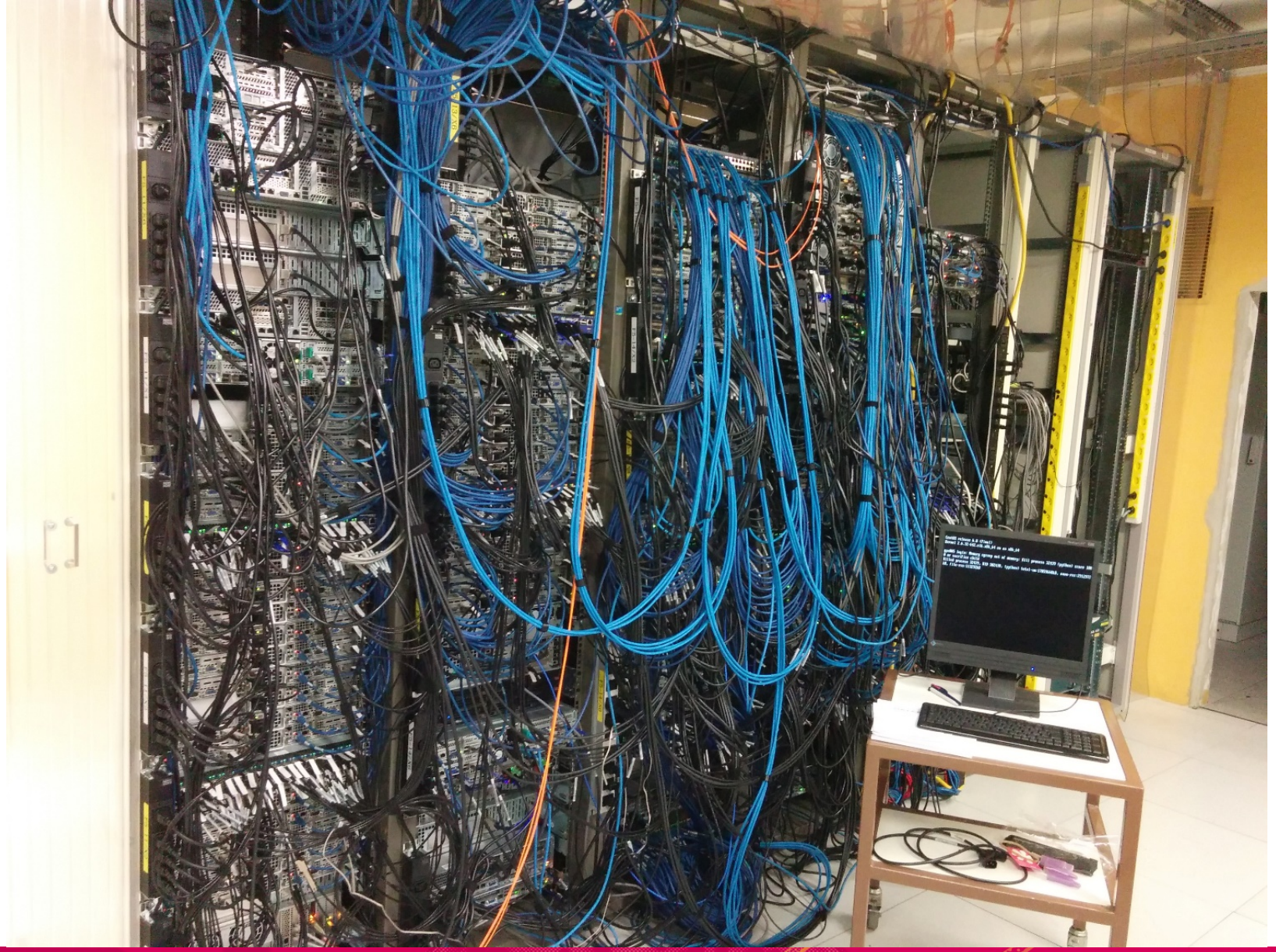
Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten
New plot and data collected for 2010-2017 by K. Rupp

Baobab architecture

NASAC
Network
Acces
Server
ACademic



Baobab front



Baobab back

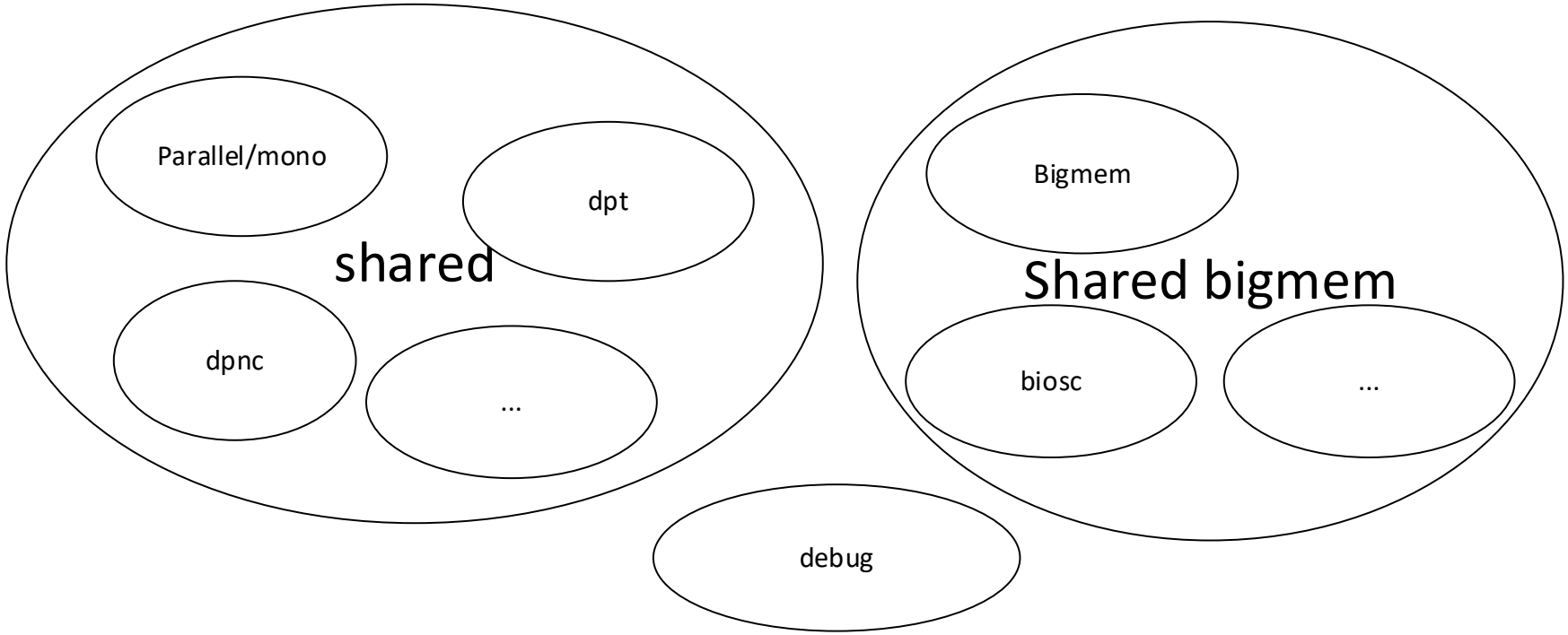
Resources

Baobab Resources

- Compute nodes (CPUs, GPGPUs …)
- Memory (MB, GB)
- Execution time (min, hours, days)
- Software licenses (Matlab, stata…)
- Storage space (home directory, scratch space, NAS)

Partitions type : private and

Partition = nodes grouping



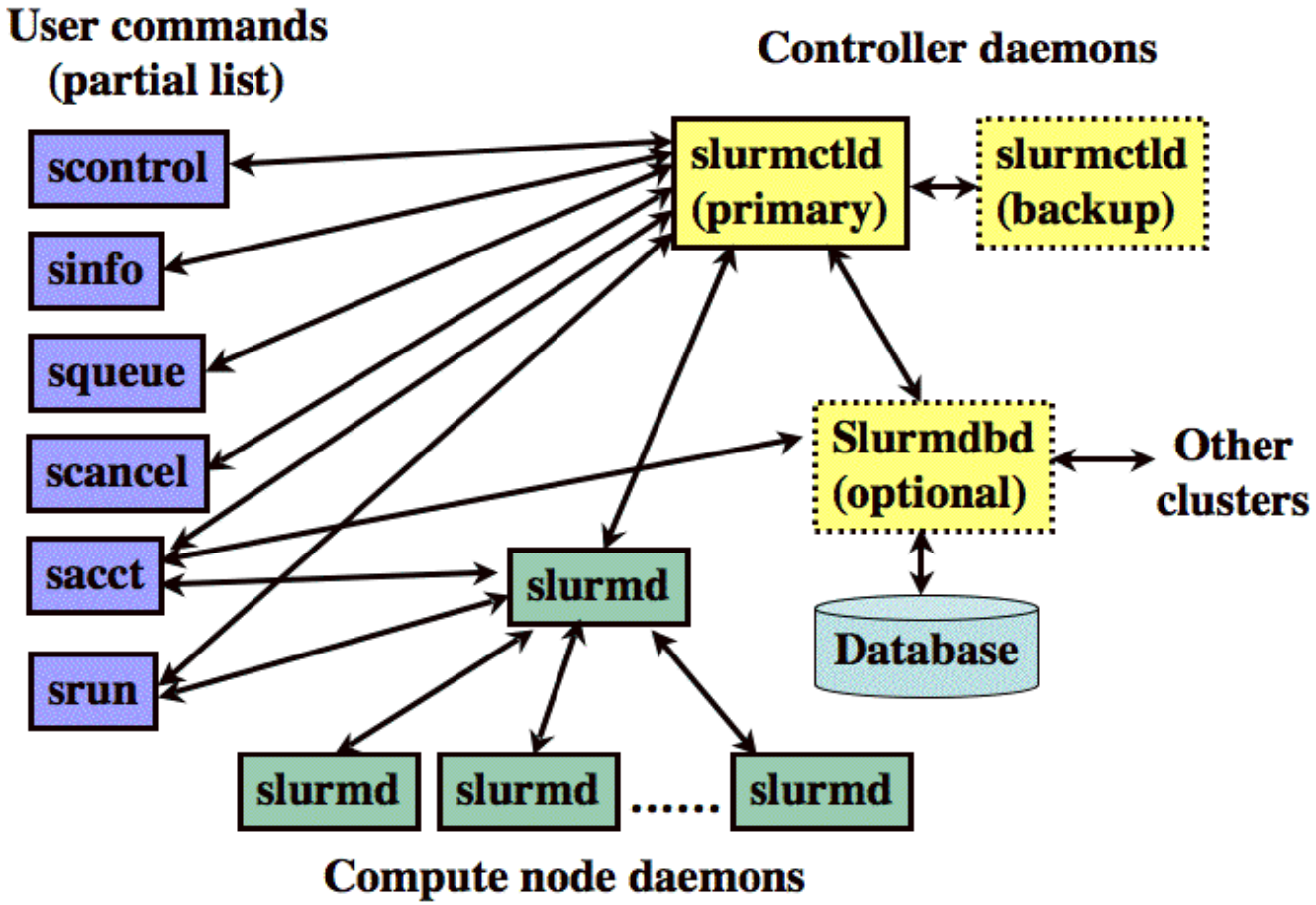
Partitions

Choose the correct partition type

- Debug? > debug
- Runtime less than 12h00? > mono-shared
- Between 12h00 and 4 days? > mono
- More than 96GB RAM? > shared-bigmem

Resources management with SLURM

SLURM
Simple
Linux
Utility for
Resource
Management



Resources management

- Job priority in the waiting queue
- Resources allocation and reservation
- Job execution
- Job archive

Slurm Documentation
https://slurm.schedmd.com

Job Priority

Four factors to prioritize jobs

- Fair Share: The highest is the past usage, the lowest is the priority. **SLURM forgets history !** (50% each 2 weeks)
- Job Age: Priority increases with the waiting time in the queue
- Job Size: Big jobs are favored
- Partition Owner: Private partitions give higher priority to their owner

Procedure

Baobab procedure

Registration and connection to Baobab

1 - Ask for an account on Baobab (once)

2 - Connect to Baobab

Job preparation

3 - Transfer data and/or software to Baobab

4 - Write a script to submit your job to Baobab

Job execution

5 - Select the software to run (module)

6 - Check the correct execution of the job

7 - Run the job

Job completion

8 – See the job state

9 - Fetch the results

1 - Ask for an account

> Support-SI (Request form)

2 - Connection to Baobab

> https://baobabmaster.unige.ch

- Textual connection through SSH (putty, terminal)
- Graphical session using x2go

3 - Transfer data to Baobab

- Transfer data using SFTP or SCP using for example FileZilla

>http://baobabmaster.unige.ch/enduser/src/enduser/access.html#file-transfer

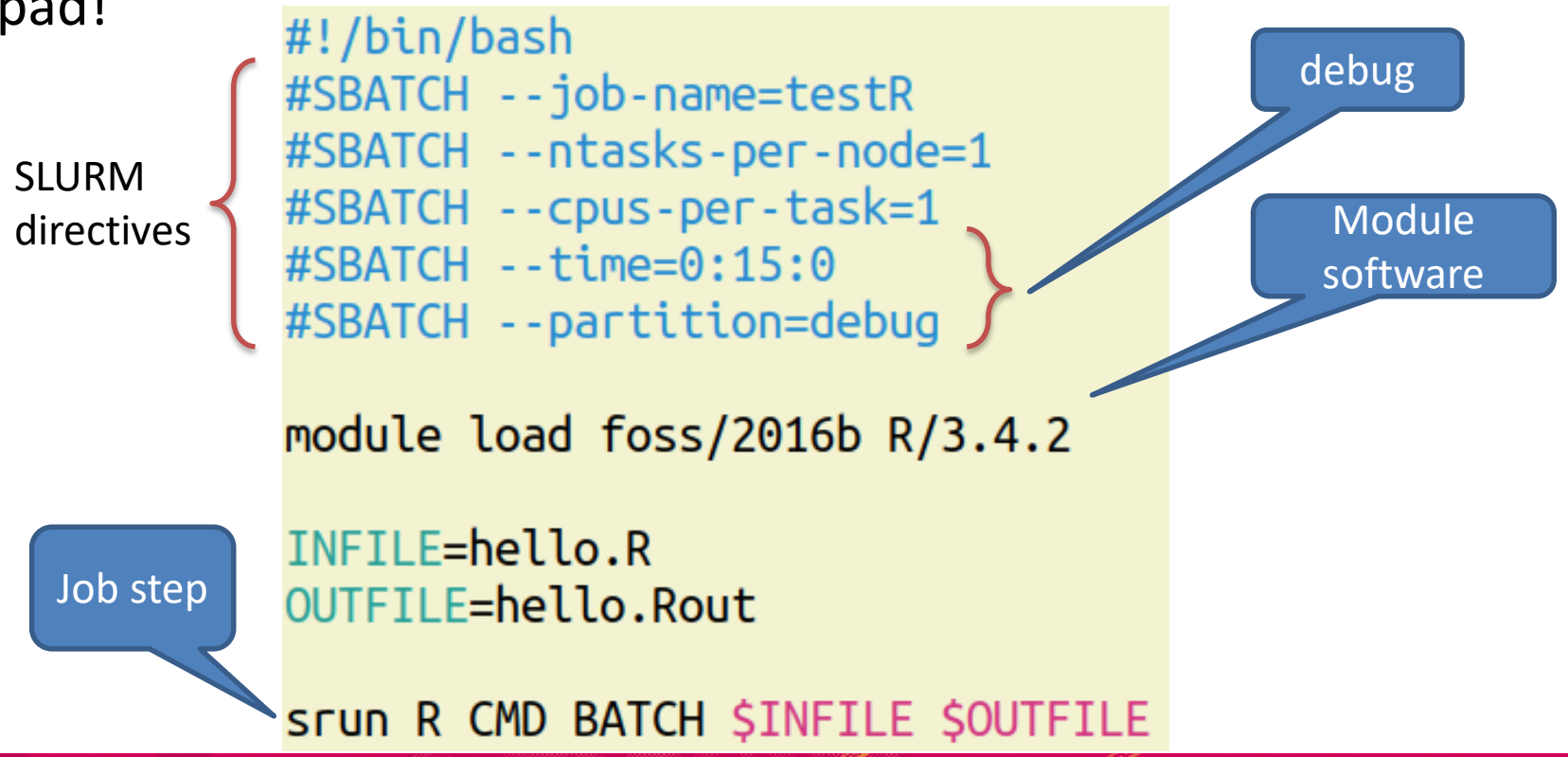
4 - Write a script to submit your job

sbatch script

- Request needed resources
  - Number of CPUs > 1 to 32
  - Needed memory > 3 Go per core by default
  - License software (Optional)
  - Execution time > max 4 days
  - Choose partition > debug, mono or mono-shared
- Select the software
- Specify input and output data
- Specify log files (the program output)

4 - sbatch script - Example

- Use a text editor on Baobab (nano, gedit etc).
- If editing from windows, be sure that you use notepad++, not notepad!



5 - Select the software to run (module)

5.1 - Search the needed software

5.2 - See the details of a specific version (if more than one available)

5.3 - Load the dependencies and the software

5.1 - Module spider

```
[sagon@login1 ~] $ module spider R

----------------------------------------------------------------------------
  R:
----------------------------------------------------------------------------
    Description:
      R is a free software environment for statistical computing and graphics.

     Versions:
        R/3.2.3
        R/3.3.1
        R/3.3.2
        R/3.4.2
     Other possible modules matches:
        BioPerl  Bismark  Blender  CoordgenLibs  DISCOVARdenovo  FreeCT  FreeSurfer  ...

----------------------------------------------------------------------------
  To find other possible module matches execute:

      $ module -r spider '.*R.*'

----------------------------------------------------------------------------
  For detailed information about a specific "R" module (including how to load the modules)
 use the module's full name.
  For example:

     $ module spider R/3.4.2
----------------------------------------------------------------------------
```

Search for R versions

Available R versions

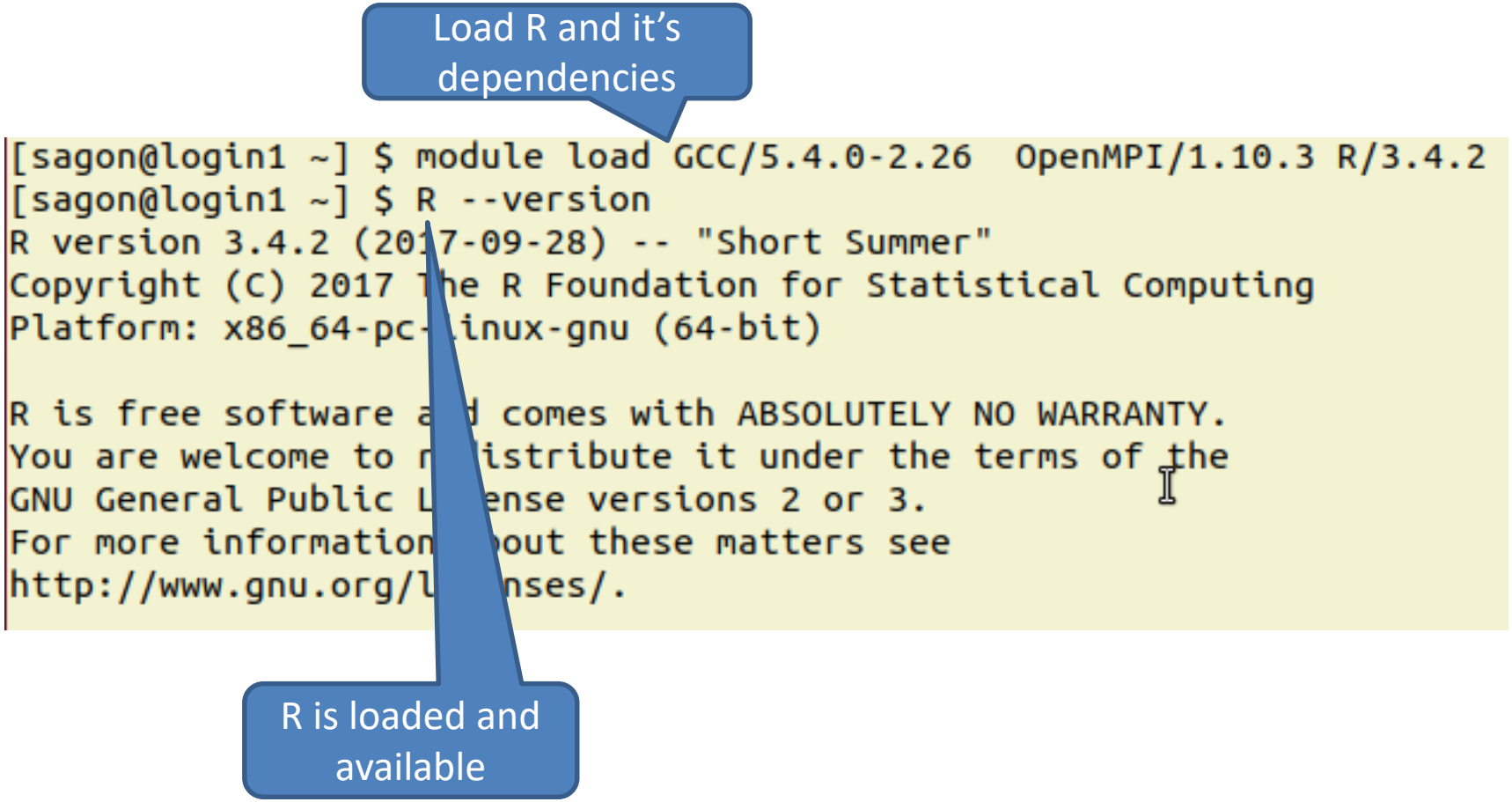
5.2 - Module details of a specific version

```
[sagon@login1 ~] $ module spider R/3.4.2

----------------------------------------------------------------------------
  R: R/3.4.2
----------------------------------------------------------------------------
    Description:
      R is a free software environment for statistical computing and graphics.


    You will need to load all module(s) on any one of the lines below before the "R/3.4.2"
 module is available to load.

      GCC/5.4.0-2.26  OpenMPI/1.10.3

    Help:

      Description
      ===========
      R is a free software environment for statistical computing and graphics.


      More information
      ================
       - Homepage: http://www.r-project.org/


      Included extensions
      ===================
      abc-2.1, abc.data-1.0, abind-1.4-3, acepack-1.3-3.3, adabag-4.1, ade4-1.7-4,
      adegenet-2.0.1, adephylo-1.1-6, ADGofTest-0.3, akima-0.5-12,
      AlgDesign-1.1-7.3, animation-2.4, ape-3.5, arm-1.8-6, assertthat-0.1,
```

Show detail of R version 3.4.2

Dependencies (pick one line if many shown)

5.3 - Module load

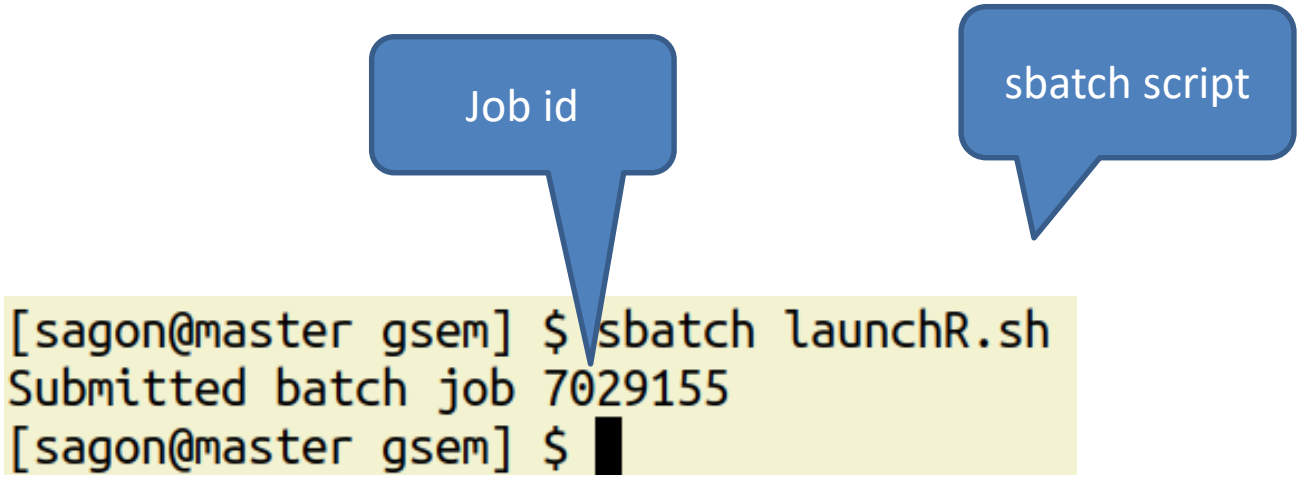


6 - Check correct execution of the job

- Submit your job into the debug partition
- Check if everything is working fine
  - logs
  - result files
  - memory usage
  - CPU usage

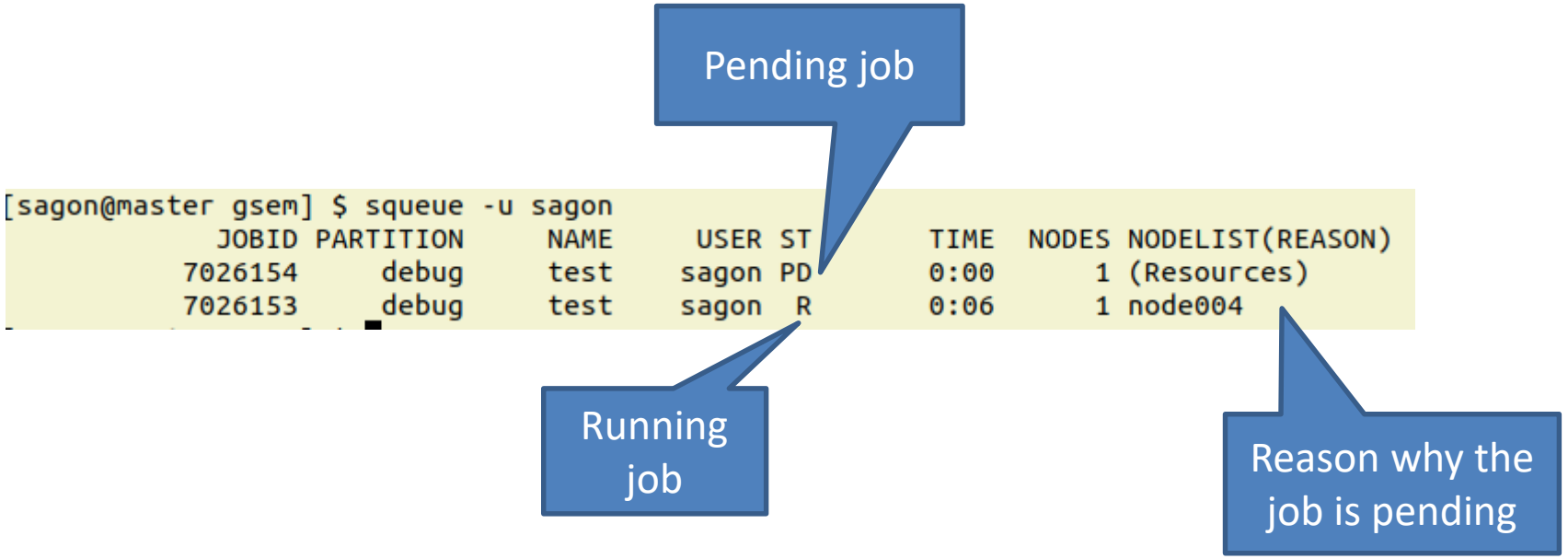
7 - Run the job

- The job goes into the waiting queue



8 - See the job state

- Job state > $ squeue –j job_id
- See all your jobs: $ squeue –u your_username



8 - See job state

- See the job priority > $ sprio
- See more details > $ scontrol show job job_id
- See output files > (slurm-nnnnn.out and or slurm-nnnn.err) with tail, nano, vim

If you did a mistake, don't let the job running !

- Cancel the job > $ scancel job_id

9 - Fetch the result

- Copy the results with FileZilla
- Delete the unneeded temporary files

Good practices

Analyze job behavior

- When running > $ sstat

```
[sagon@master gsem] $ sstat --format=AveCPU,MaxRSS,Nodelist,MaxDiskRead -j 7031229
    AveCPU     MaxRSS              Nodelist  MaxDiskRead
---------- ---------- -------------------- ------------
 00:00.000      1560K              node004        0.05M
```

Memory used

- When finished > $ sacct

Job step 0

```
[sagon@master gsem] $ sacct --format=Start,AveCPU,MaxRSS,JobID,NodeList,ReqMem --units=G -j 7031229
              Start     AveCPU     MaxRSS        JobID        NodeList     ReqMem
------------------- ---------- ---------- ------------ --------------- ----------
2018-04-12T13:44:32                        7031229              node004     2.93Gc
2018-04-12T13:44:32   00:00:00      0.01G 7031229.bat+              node004     2.93Gc
2018-04-12T13:44:34   00:00:00      0.00G 7031229.0                 node004     2.93Gc
```

Multithread, Distributed

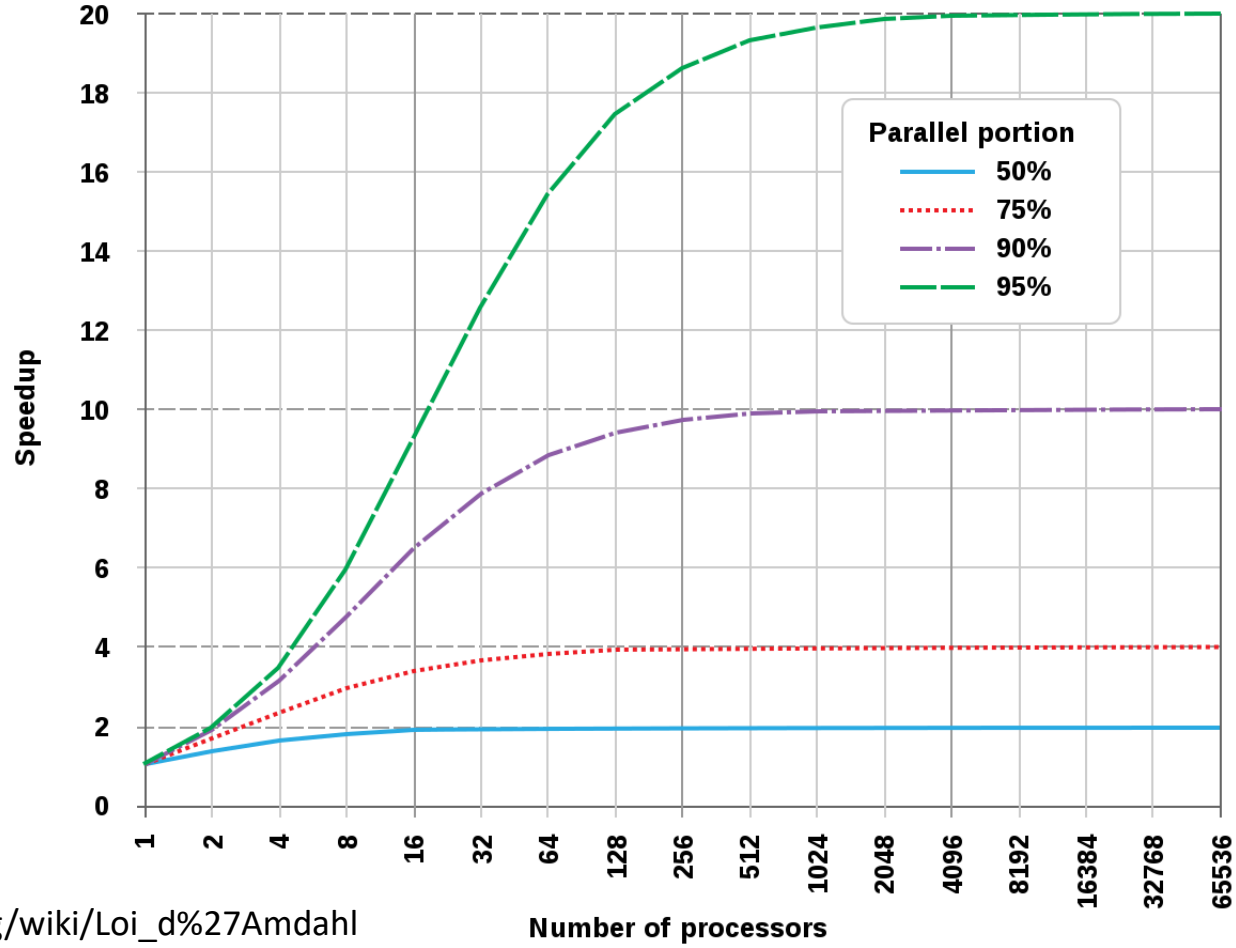
- Does your software benefit from multiple CPUs?
- **Hint:** multicore, multithread in the documentation
- Does your software benefit from multiple CPUs on more than one compute node?
- **Hint:** openmpi in the documentation
- How many cores?

Parallel computing

- The Amdahl's law
- Speedup

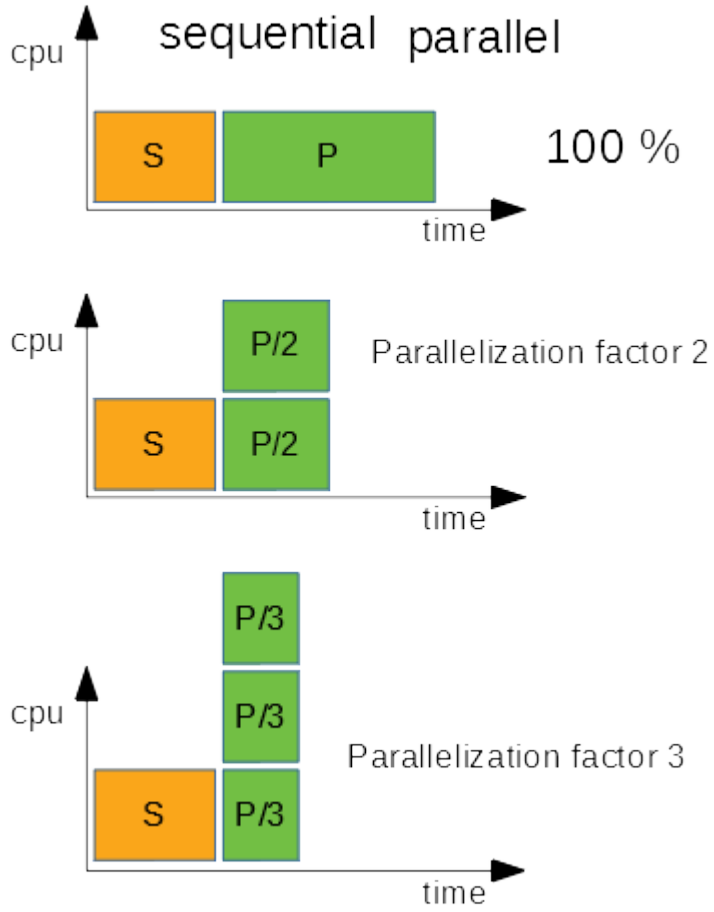
One woman takes 9 months to give birth Does 9 women give birth to one child in 1 month? Not everything is parallelizable!

The Amdahl's Law



https://fr.wikipedia.org/wiki/Loi_d%27Amdahl

The Amdahl's law illustrated



- Speedup is limited by the «S» block
- Measure run time with 1, 2, 4, 8, 16 cores
- Stick with something reasonable (don't double cpu for 5% gain!)

Multithread example using R

- R only use one core by default
- Use extra packages (mc, parallel etc)
- Specify the number of cores to be used

Multithread example using R

Load the parallel package

```
library(foreach)
library(doParallel)

# set the number of cores in the sbatch script
registerDoParallel(cores=Sys.getenv("SLURM_CPUS_PER_TASK"))

# print the number of workers
getDoParWorkers()

trials <- 100000
x <- iris[which(iris[,5] != "setosa"), c(1,5)]


# parallel execution
system.time({
  r <- foreach(icount(trials), .combine=rbind) %dopar% {
    ind <- sample(100, 100, replace=TRUE)
    result1 <- glm(x[ind,2]~x[ind,1], family=binomial(logit))
    coefficients(result1)
  }
})

# sequential execution
system.time({
  r <- foreach(icount(trials), .combine=rbind) %do% {
    ind <- sample(100, 100, replace=TRUE)
    result1 <- glm(x[ind,2]~x[ind,1], family=binomial(logit))
    coefficients(result1)
  }
})
```

Embarrassingly parallel

- Same job launched on different dataset: use a Job array!

```
#!/bin/sh
#SBATCH --job-name=test
#SBATCH --ntasks-per-node=1
#SBATCH --cpus-per-task=1
#SBATCH --time=00:15:00
#SBATCH --partition=debug
#SBATCH --output myjob-%A_%a.out

srun echo "I'm task_id " ${SLURM_ARRAY_TASK_ID} " on node " $(hostname)
```

Embarrassingly parallel

- Run 3 similar jobs

```
[sagon@master gsem] $ sbatch --array=1-3 test.sh
Submitted batch job 7028896
[sagon@master gsem] $ ls -la
total 3
drwxr-xr-x 2 sagon unige   4 Apr 12 13:19 .
drwxr-xr-x 9 sagon unige  11 Apr 12 12:05 ..
-rw-r--r-- 1 sagon unige  41 Apr 12 13:19 myjob-7028896_1.out
-rw-r--r-- 1 sagon unige  41 Apr 12 13:19 myjob-7028896_2.out
-rw-r--r-- 1 sagon unige  41 Apr 12 13:19 myjob-7028896_3.out
```

```
[sagon@master gsem] $ cat myjob-7028896_1.out
I'm task_id  1  on node  node004.cluster
[sagon@master gsem] $
```

Useful links

- Baobab welcome page
  > http://baobab.unige.ch
- Baobab documentation > **direct links**
  > http://baobabmaster.unige.ch/enduser/src/enduser/enduser.html
  - **Connection**
    > http://baobabmaster.unige.ch/enduser/src/enduser/access.html#connection
  - **Files transfer**
    > http://baobabmaster.unige.ch/enduser/src/enduser/access.html#file-transfer
  - **Modules**
    > http://baobabmaster.unige.ch/enduser/src/enduser/enduser.html#end-user-modules
  - **Slurm** > http://baobabmaster.unige.ch/slurmdoc/overview.html
  - **sbatch** > http://baobabmaster.unige.ch/enduser/src/enduser/submit.html#batch-mode

HPC Support

- **To get an access to Baobab** > Support-SI (request form)
- **To install a software** > Support-SI (request form)
- **To ask for help and advice** > hpc@unige.ch

Join us on our new web forum : HPCcommunity@UNIGE
> hpc-community.unige.ch
(an account is automaticaly created when logging on the first time)

Your support staff

> Yann Sagon and Luca Capello

Thanks for your attention !