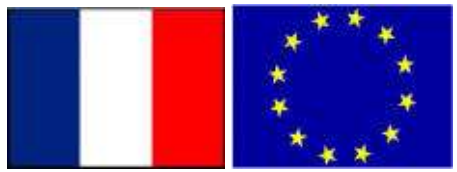
CentraleSupélec



SG6 - HPC

Using machines of CentraleSupelec DCE for TD2-3/Lab-2

Stéphane Vialle

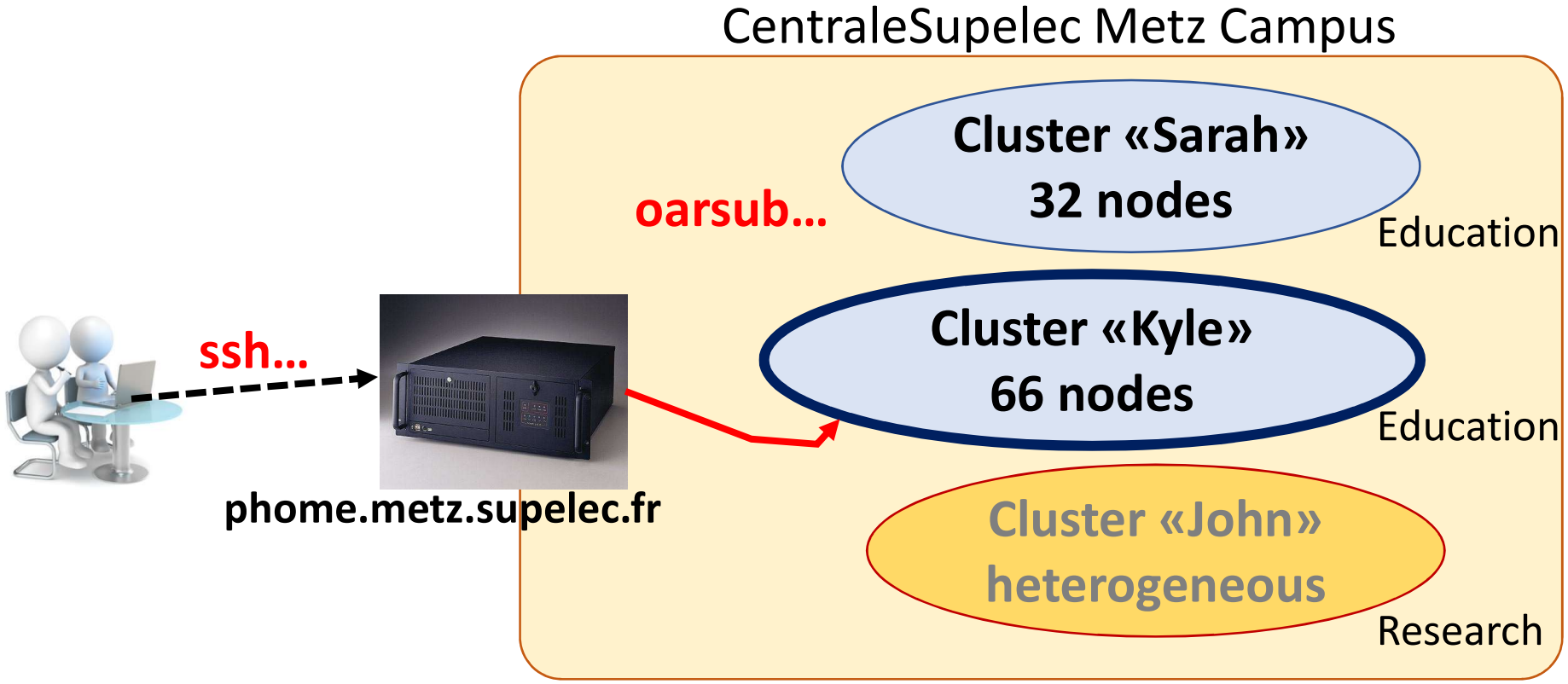
CentraleSupélec université PARIS-SACLAY ÉCOLE DOCTORALE Sciences et technologies de l'information et de la communication (STIC) RISEGrid Research Institute for Smarter Electric Grids Grand Est ALSACE CHAMPAGNE-ARDENNE LORRAINE

CentraleSupélec

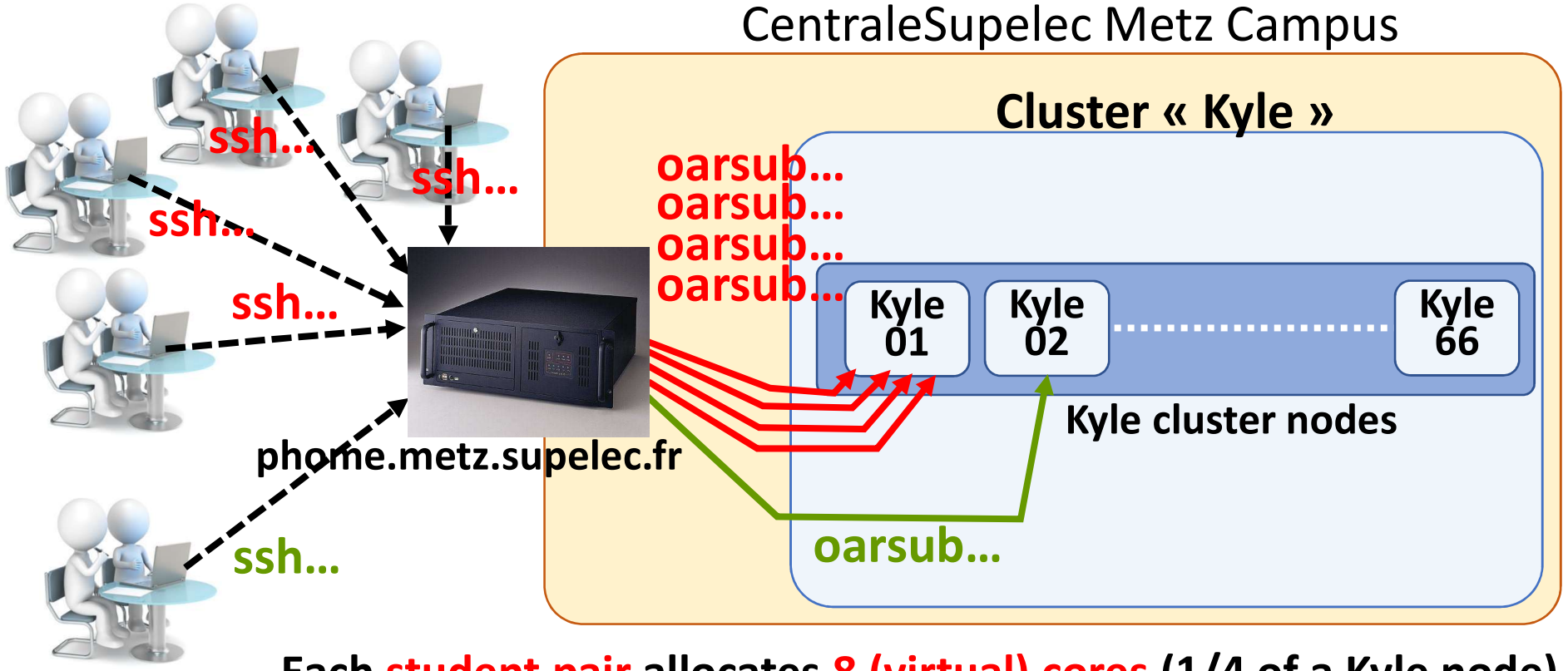
Using machines of CentraleSupelec DCE for Lab-2

- **DCE architecture and access**
  → Connection
- **Application installation & test**
  → File copy & unzip & compilation & small size local executions
- **Large size run**
  → Batch mode & MPI on cluster & shell script

DCE CPU clusters



Configuration for Distributed MPI TP



Each student pair allocates 8 (virtual) cores (1/4 of a Kyle node)

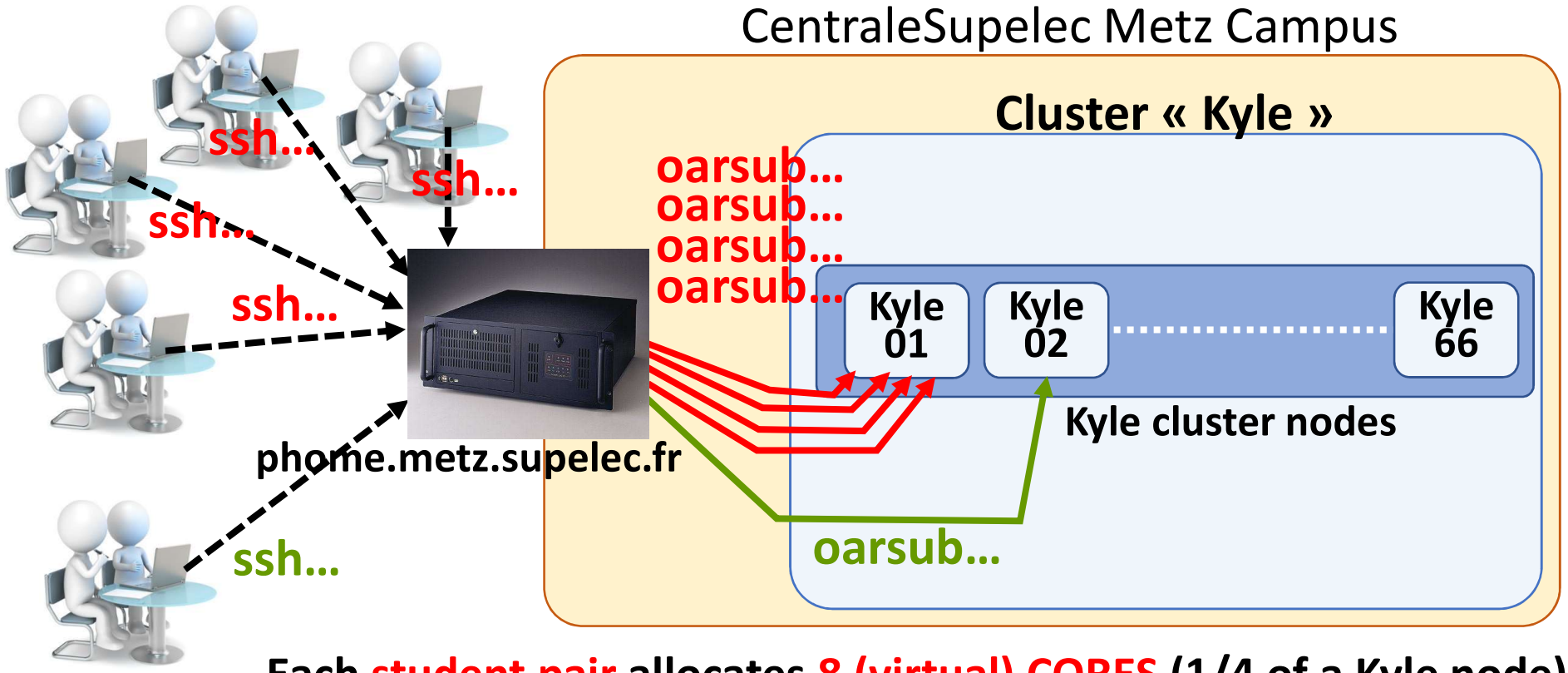
Linux → REUSE YOUR Lab-1 ACCOUNT

Group k: `ssh phome.metz.supelec.fr –l cpucs1k_n`

From cpucs1k_1 up to cpucs1k_19

`oarsub –p "cluster='kyle'" -l core=8,walltime=4:00:00 -I`

Configuration for Distributed MPI TP



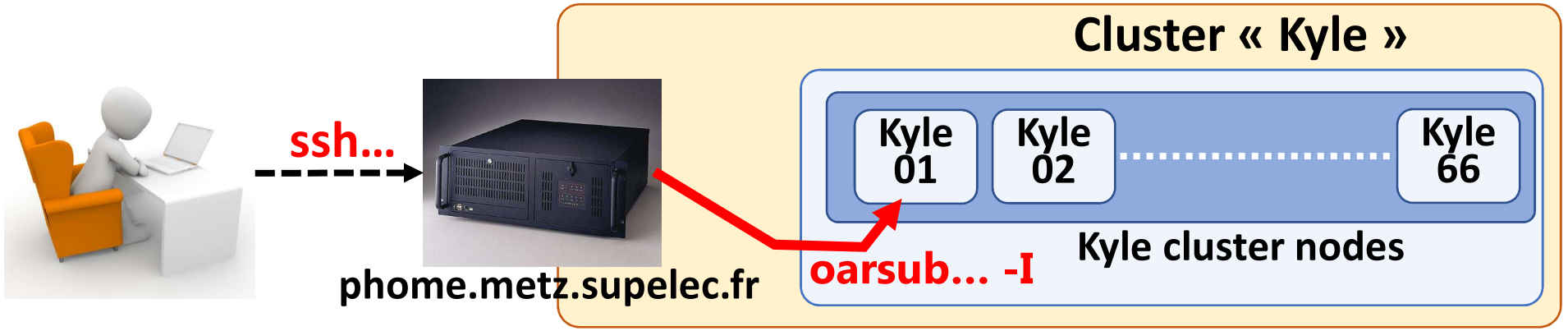
Each student pair allocates 8 (virtual) CORES (1/4 of a Kyle node)

Windows → REUSE YOUR Lab-1 ACCOUNT

Group k: **PUTTY → connect to phome.metz.supelec.fr –l cpucs1k_*n***

oarsub –p "cluster='kyle'" -l core=8,walltime=4:00:00 -I

Linux useful commands



To know on which machine you are:

hostname → « sammy »: from *phome*

(name of phome inside our campus network)

hostname → « kyle*xx* » from *kyle01* up to *kyle66*

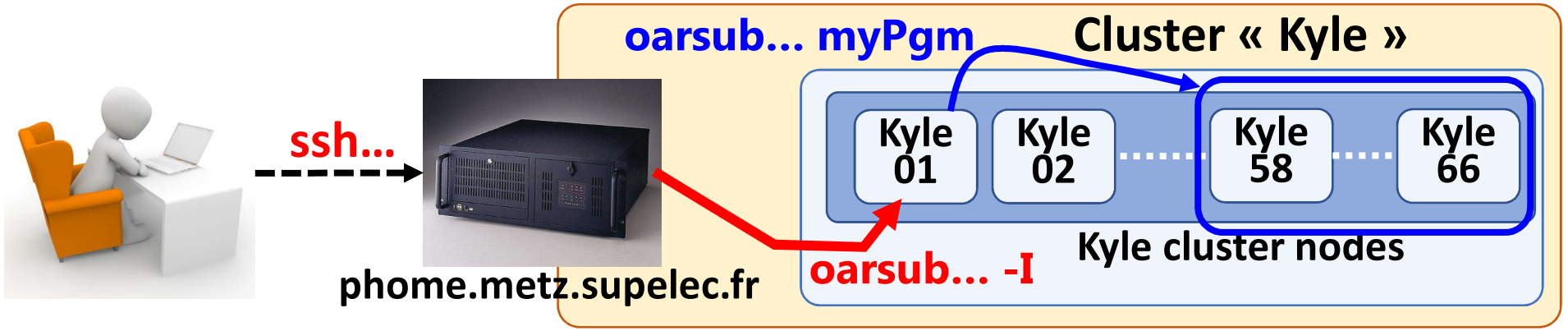
Rmk : if *hostname* returns « sammy » after an oarsub …

… you are still logged on sammy (frontal machine) !!

… your oarsub command has failed…

→ re-try !

Linux useful commands



To submit and manage *OAR jobs/sessions* on clusters:

oarsub ... -I → requires an *interactive session* (interactive job)

oarsub ... myPgm → submits a *batch job* in a job queue:
will be run when resources will be available

You can submit a new batch job on kyle nodes
from your interactive session on kyle*xx* node

oarstat → lists running and waiting jobs/session on the clusters

oardel *jobId* → can remove your job/session from the OAR queue

CentraleSupélec

Using machines of CentraleSupelec DCE for Lab-2

- **DCE architecture and access**
  → Connection
- **Application installation & test**
  → File copy & unzip & compilation & small size local executions
- **Large size run**
  → Batch mode & MPI on cluster & shell script

Copy & Unzip

pwd *Print Working Directory*

→Your should be in your home directory:

Ex: */usr/users/cpucs11/cpucs11_1*

cp ~vialle/tmp/SG6-TD2-P1-MatrixProduct.zip . *Copy a file*

→Your home directory contains a new file:

SG6-TD2-P1-MatrixProduct.zip

do not forget the « . »

unzip SG6-TD2-P1-MatrixProduct.zip *Unzip a file*

→Your home directory contains a new sub-directory:

SG6-TD2-P1-MatrixProduct/

cd SG6-TD2-P1-MatrixProduct *Change Directory*

→ Enter into *SG6-TD2-P1-MatrixProduct/*

cd ..

→ go in the parent/upper directory

ls *List (current directory)*

→ **Makefile**.c init.c **kernel.c** main.c

Compilation

ls *List (current directory)*

→ **Makefile**.c init.c kernel.c **main.c**

more Makefile *Print the containt of the file 'Makefile'*

```
CC=mpicc                          #MPI compiler      →
CFLAGS= -O3 …… -fopenmp #OpenMP enabled →
…
all:
    $(CC) $(CFLAGS) $(LDFLAGS) -o $(EXECNAME)……
```

Implementation is distributed with MPI (see *main.c*)

Computing kernels are multithreaded (see *kernel.c*)

make *Compile the source files according to the 'Makefile' rules*

ls *List (current directory)*

→ Makefile.c **MatixProduct** init.c kernel.c main.c

2 – Application installation & test

1st small size local execution

ls *List (current directory)*

→ Makefile.c **MatixProduct** init.c kernel.c main.c

mpirun –np 1 ./MatrixProduct -h

Local interactive run:

- *1-process run*
- *on your current kylexx machine*
- *In your interactive session*

→ Print the application options

```
MatrixProduct usage:
        [-lc <nb of lines and columns> (default 1024)]
        [-klc <nb of kilo lines and columns> (default 1)]
        [-k <Kernel Id> (default 0)]
        [-h]: print this help
        [-nt <number of threads> (default 1)]
```

2nd small size local execution

ls *List (current directory)*

→ Makefile.c **MatixProduct** init.c kernel.c main.c

mpirun –np 1 ./MatrixProduct –klc 1 –k 1 –nt 4 *Local interactive run*

→ Compute a 1024x1024 matrix product, with **1** MPI process & with **4** OpenMP threads in the process (4 computing *tasks*)

```
Product of two square matrixes of 1024x1024 doubles:
- Number of MPI processes: 1
- Number of OpenMP threads per process: 4
- Kernel Id: 1

Parallel computation starts...
- Results:
      PE0: C[0][1023] = 366503002112.000000
      PE0: C[512][512] = 281841223335936.000000
      PE0: C[1023][0] = 562216956854272.000000
- Performances:
      PE0: Elapsed time of the loop = 0.03 (s)
      PE0: Speed of the loop        = 84.28 (Gflops)
```

3rd small size local execution

ls *List (current directory)*

→ Makefile.c **MatixProduct** init.c kernel.c main.c

mpirun –np 4 ./MatrixProduct –klc 1 –k 1 –nt 1 *Local interactive run*

→ Compute a 1024x1024 matrix product, with **4** MPI processes & with **1** OpenMP threads in each process (4 computing *tasks*)

Product of two square matrixes of 1024x1024 doubles:
- Number of MPI processes: 4
- Number of OpenMP threads per process: 1
- Kernel Id: 1

Parallel computation starts...
- Results:
  PE3: C[0][1023] = 366503002112.000000
  PE2: C[512][512] = 281841223335936.000000
  PE0: C[1023][0] = 562216956854272.000000
- Performances:
  PE0: Elapsed time of the loop = 0.03 (s)
  PE0: Speed of the loop = **77.14** (Gflops)

Be careful: performance variations are important from one run to another. **Too small pb!**

CentraleSupélec

Using machines of CentraleSupelec DCE for Lab-2

- **DCE architecture and access**
  → Connection
- **Application installation & test**
  → File copy & unzip & compilation & small size local executions
- **Large size run**
  → Batch mode & MPI on cluster & shell script

OAR batch mode principle

oarsub -p "cluster='kyle'" -l nodes=1 '……..'

Submit a job in a job queue, requiring 1 node

As soon as 1 node will be available:

it will be allocated for the job,

and the job will be launched

2 output files will be generated by OAR (one reporting errors, one reporting std output)

The batch command must be between two ' '

pwd

/usr/…../SG6-TD2-P1-MatrixProduct

In this exercise you must be in the src file directory

oarsub -p "cluster='kyle'" -l nodes=1 'mpirun -np 1 ./MatrixProduct -klc 8 -k 1 -nt 8'

```
================================================================================
..............
================================================================================
[ADMISSION RULE] Set default walltime to 7200.
[ADMISSION RULE] Modify resource description with type constraints
[ADMISSION RULE] cluster deja defini
OAR_JOB_ID=619374
```

Command will be run in the same directory than the oarsub call

ls –l

```
......
-rw-r--r-- 1 vialle ims      0 Dec 23 16:24 OAR.619374.stderr
-rw-r--r-- 1 vialle ims    428 Dec 23 16:24 OAR.619374.stdout
```

Error file is empty (0 bytes): perfect

Stores the *screen output* of the pgm

OAR batch mode principle

ls –l

```
.....
-rw-r--r-- 1 vialle ims      0 Dec 23 16:24 OAR.619374.stderr
-rw-r--r-- 1 vialle ims    428 Dec 23 16:24 OAR.619374.stdout
```

Stores the *screen output* of the pgm

more OAR.619374.stdout

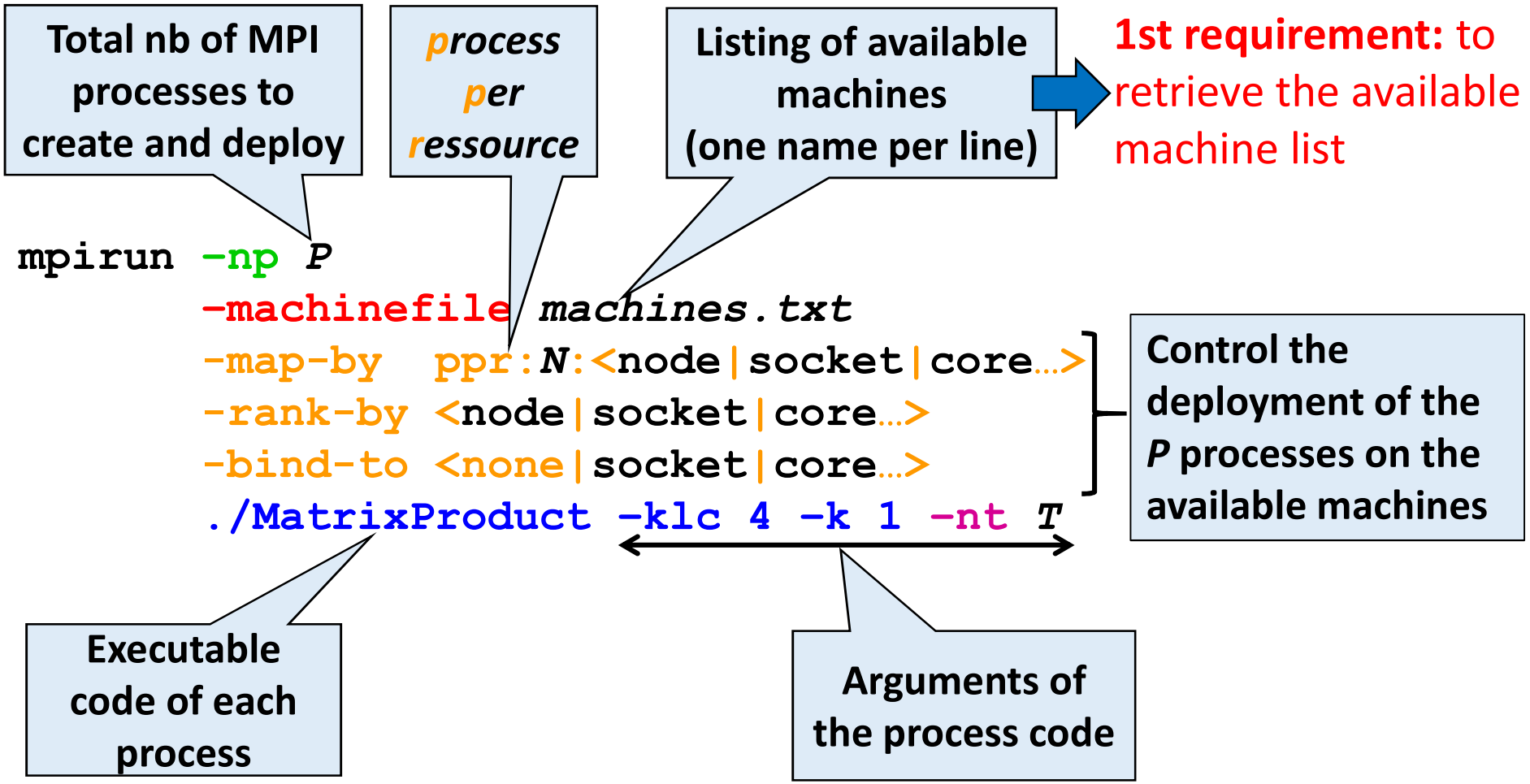
```
Product of two square matrixes of 8192x8192 doubles:
- Number of MPI processes: 1
- Number of OpenMP threads per process: 8
- Kernel Id: 1

Parallel computation starts...
- Results:
      PE0: C[0][8191] = 1501199786311680.000000
      PE0: C[4096][4096] = 922487287022669048.000000
      PE0: C[8191][0] = 1844374130745409536.000000
- Performances:
      PE0: Elapsed time of the loop = 6.10 (s)
      PE0: Speed of the loop        = 180.29 (Gflops)
```

Screen output in interactive mode

OAR.xxxxx.stdout file in batch mode

MPI execution on cluster

Complete *mpirun* syntax (when deploying on a PC cluster)



MPI execution on cluster

List of nodes allocated by OAR

The **OAR_NODEFILE** environment variable is the path to a temporary file listing all the allocated « resources »

```
oarsub -p "cluster='kyle'" –l core=8 –I
```

```
more  $OAR_NODEFILE
```

kyle01
kyle01
kyle01
kyle01
kyle01
kyle01
kyle01
kyle01

Each virtual core is listed like a resource!

-u: unique
Remove duplicated resources

```
sort -u  $OAR_NODEFILE
```

kyle01

```
sort -u  -o machines.txt  $OAR_NODEFILE
```

→ Generate the "machines.txt" file, listing once each allocated node

Shell script for the batch mode

A *shell script* to aggregate several operations in a batch command

Ex: oarsub -p "cluster='kyle'" -l nodes=4 *'./myrun 8 16'*

myrun shell script main steps:

- *Generate the machines.txt file*
- mpirun –np **$1** –machinefile machines.txt …. ./MatrixProduct –klc **$2** –k 1 –nt 8
- *Print extra information*
- *Remove the machines.txt file*

➡ A first *myrun* shell script is supplied with the source files

→ **Have a look at the *myrun* shell script**

➡ ***myrun*** **must be a Linux and executable file**

→ **dos2unix myrun**

→ **chmod 700 myrun**

CentraleSupélec

Using machines of CentraleSupelec DCE for Lab-2

Appendix

- **Available editors**

CentraleSupélec

Appendix

Available editors

ls *List (current directory)*

→ Makefile.c **MatixProduct** init.c kernel.c main.c

2 alphanumerical editors are available on Kyle nodes:

- *nano* : for beginners
- *vi* (or *vim*) : if you know their commands…

nano kernel.c *Edit the kernel.c source file*

Rmk : commands are listed on bottom of the window

« ^ » means « Ctrl »

Save before to exit the editor !!

When you have modified your source files:

→ you need to recompile your application

→ enter make

CentraleSupélec

Using machines of CentraleSupelec DCE for TD2-3/Lab-2

End