Introduction to High Performance Computing

Sérgio Almeida - DyBHo 256667 - 2012

High Performance Computing

High Performance Computers?

High Performance Code?

Not quite.

High Performance Analysis

High Performance Planning

High Performance Workflow

High Performance Results

1. Know your tools
2. Know your needs
3. Parallel programming
4. Meet Baltasar

Know your tools

Meet your best friend - bash

```
#!/bin/bash
# This is sample.sh sample script

VARIABLE="very valuable string"

echo $VARIABLE
```

```
~$ bash sample.sh
very valuable string
~$
```

"Be the human, not the robot"

```
#!/bin/bash
# This is batch.sh sample script
WORK_DIR="~/myRuns"
INPUT_DIR="input"
cd $WORK_DIR
for FILE in $WORK_DIR/$INPUT_DIR/*
do
    echo "Running with $FILE"; ./run $FILE
done
```

```
~$ bash batch.sh
Running with data1.in
Running with data2.in
Running with data3.in
...
```

Libraries

"There's a lib for that"

Library version matters

Compilers

Compiler versions matters

Compiler flags matters

Optimization flags matters

Compilation process matters

Still compiling by hand?

Script it with a Makefile

```
# Sample Makefile
CC=gcc
CFLAGS = -O2
CLIBS = -fopenmp -lmath

default:
	$(CC) $(CFLAGS) $(CLIBS) -o code code.c
clean:
	rm -rf code
```

```
~/my_code$ make
gcc -O2 -fopenmp -o code code.c
~/my_code$ make clean
rm -rf code
~/my_code$
```

Why Makefile?

Keep up with the evolution

High perfomance today...

is low performance tomorrow.

Keep up with the evolution

Keep up with the high performance

Working remotely

Working remotely with SSH

Public key authentication

```
~$ ssh-keygen -f ~/.ssh/mykey
Generating public/private rsa key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in ~/.
ssh/mykey.
Your public key has been saved in ~/.ssh/mykey.
pub.
```

Public key generation

```
~$ ssh-keygen -f ~/.ssh/mykey
Generating public/private rsa key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in ~/.
ssh/mykey.
Your public key has been saved in ~/.ssh/mykey.
pub.
```

```
~$ ssh baltasar.ist.utl.pt
Enter passphrase for key '~/.ssh/my_key':
baltasar ~$
```

Remote X11

```
~$ ssh -XC baltasar.ist.utl.pt
baltasar ~$
```

```
~$ ssh -XC baltasar.ist.utl.pt
baltasar ~$ mathematica
```

```
~$ ssh -XC baltasar.ist.utl.pt
baltasar ~$ mathematica
```

Yup, it opens locally.

```
~$ ssh -XC baltasar.ist.utl.pt
baltasar ~$ mathematica
```

Yup, it opens locally.
In MS Windows too.

```
~$ ssh -XC baltasar.ist.utl.pt
baltasar ~$ mathematica
```

Yup, it opens locally.
In MS Windows too.
With a few tweaks.

Remote screen

```
~$ ssh baltasar.ist.utl.pt
baltasar ~$
```

```
~$ ssh baltasar.ist.utl.pt
baltasar ~$ screen
```

```
baltasar ~$
```

A screen opens

```
baltasar ~$ echo "What a cool screen!"
```

I do my hard work

```
baltasar ~$ echo "What a cool screen!"
What a cool screen!
baltasar ~$
```

I do my hard work

```
baltasar ~$ echo "What a cool screen!"
What a cool screen!
baltasar ~$
```

And I'm out to come back tomorrow

```
baltasar ~$ echo "What a cool screen!"
What a cool screen!
baltasar ~$
```

ctrl+a then d - suspends the screen

```
baltasar ~$ screen
[detached from 2424.pts-7.baltasar]
baltasar ~$
```

ctrl+a then d - suspends the screen

```
baltasar ~$ screen
[detached from 2424.pts-7.baltasar]
baltasar ~$
```

ctrl+a then d - suspends the screen
ctrl+a then esc - screen scroll mode

```
baltasar ~$ screen
[detached from 2424.pts-7.baltasar]
baltasar ~$
```

ctrl+a then d - suspends the screen
ctrl+a then esc - screen scroll mode
that's it. really.

```
baltasar ~$ screen -list
There are screens on:
      2424.pty1.baltasar    (Detached)
```

List your screens

```
baltasar ~$ screen -r 2424
```

Resume your screens

```
# while inside a screen...
baltasar ~$ exit
```

Close a screen

Know your needs

The Computer

The Processor/Core

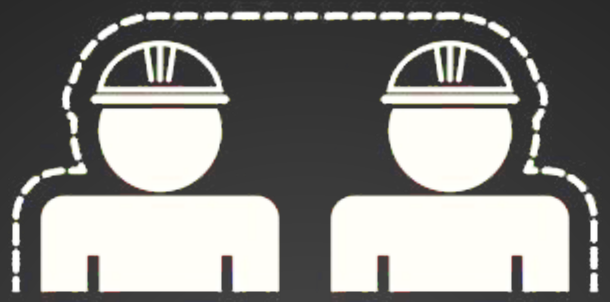
The Memory

The Storage

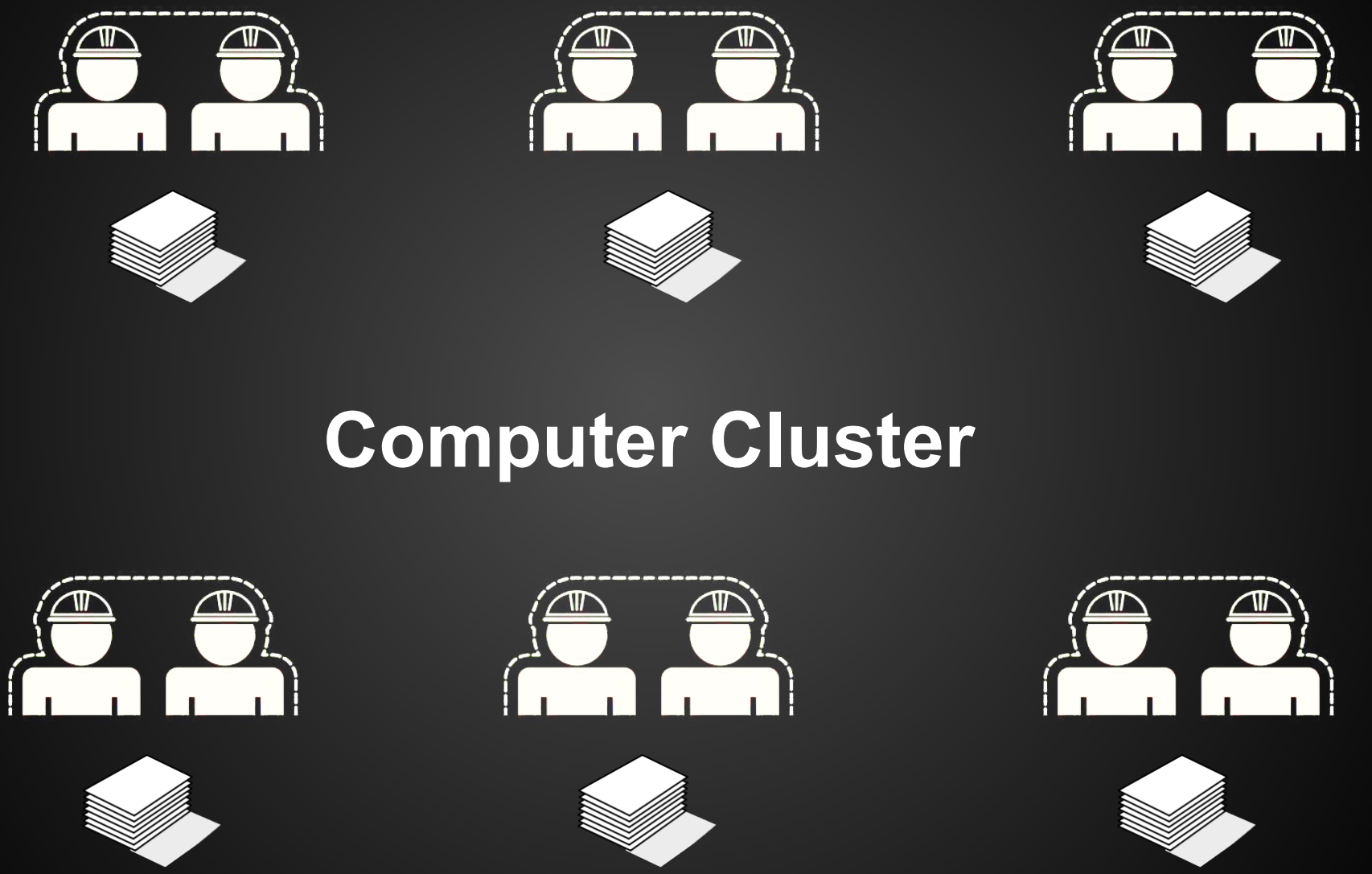
Inside the Computer

Single-Core Computer

Single-Core Computer

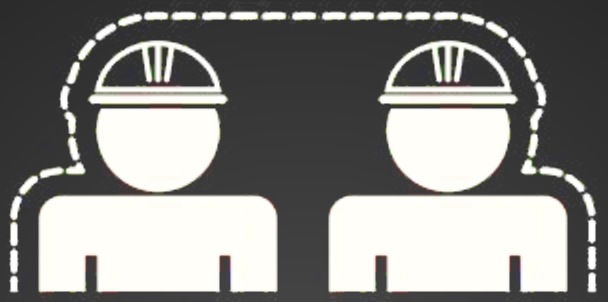


Multi-Core Computer

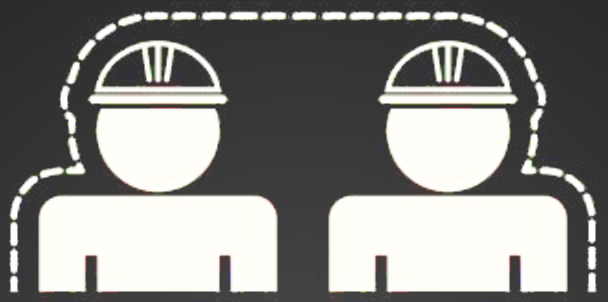


Computer Cluster

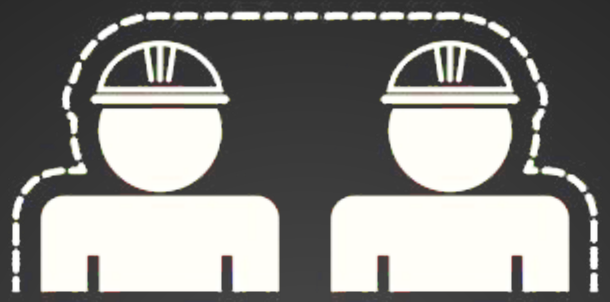
Parallel Programming



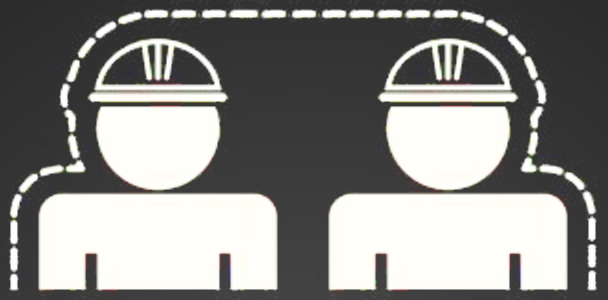
Multi-Core Programming



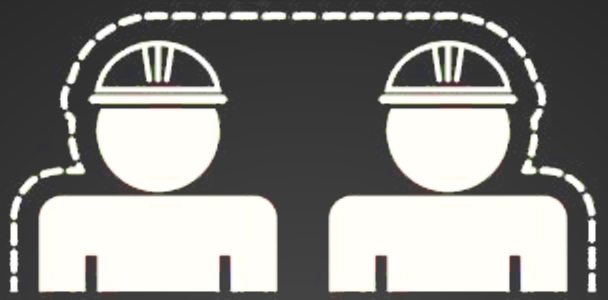
Multi-Core Programming



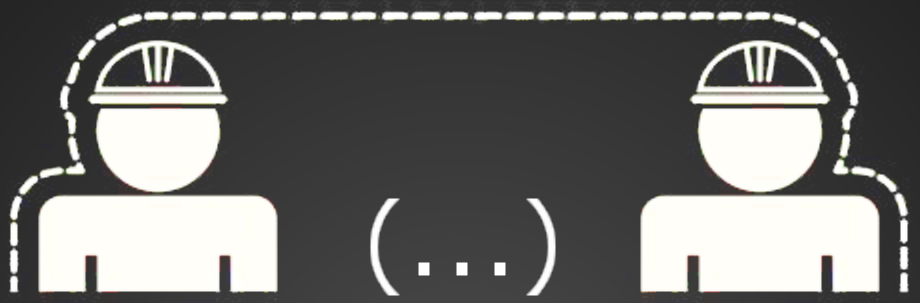
Multi-Core Programming



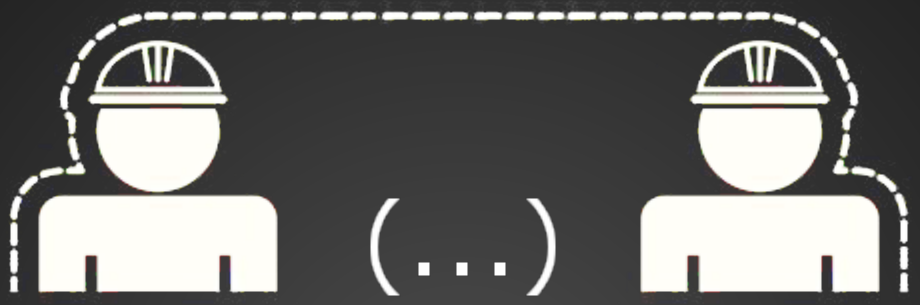
Threads



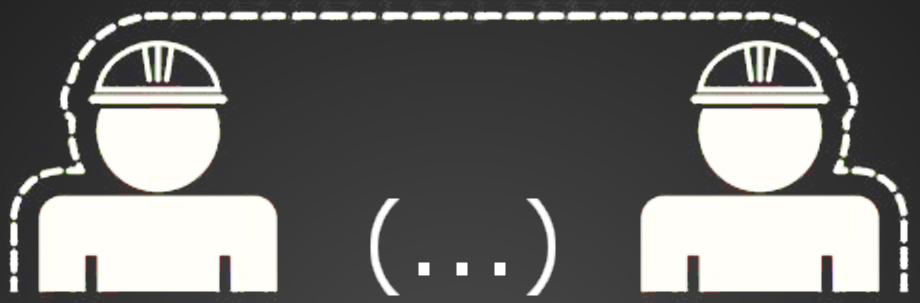
OpenMP



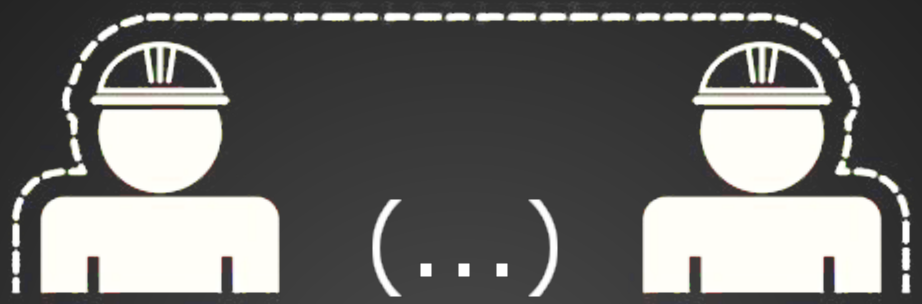
More Threads?



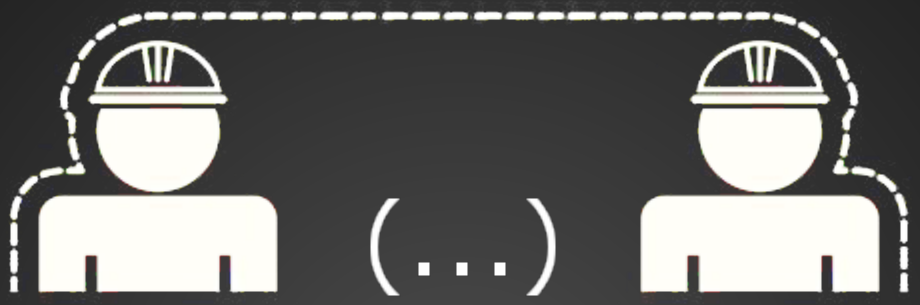
More work done?



Not quite. Why?

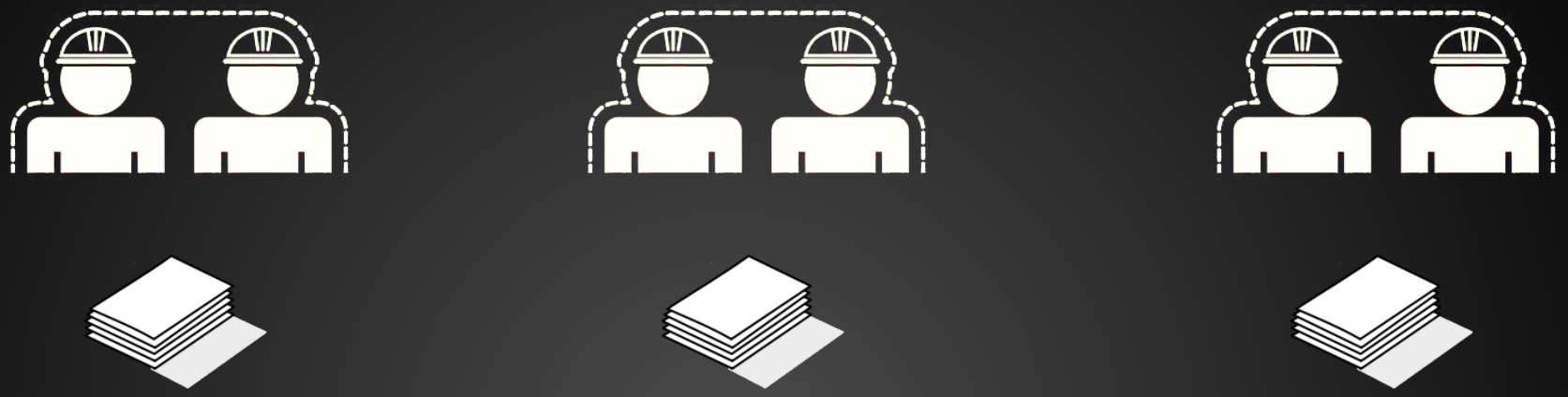


Not quite. Why?

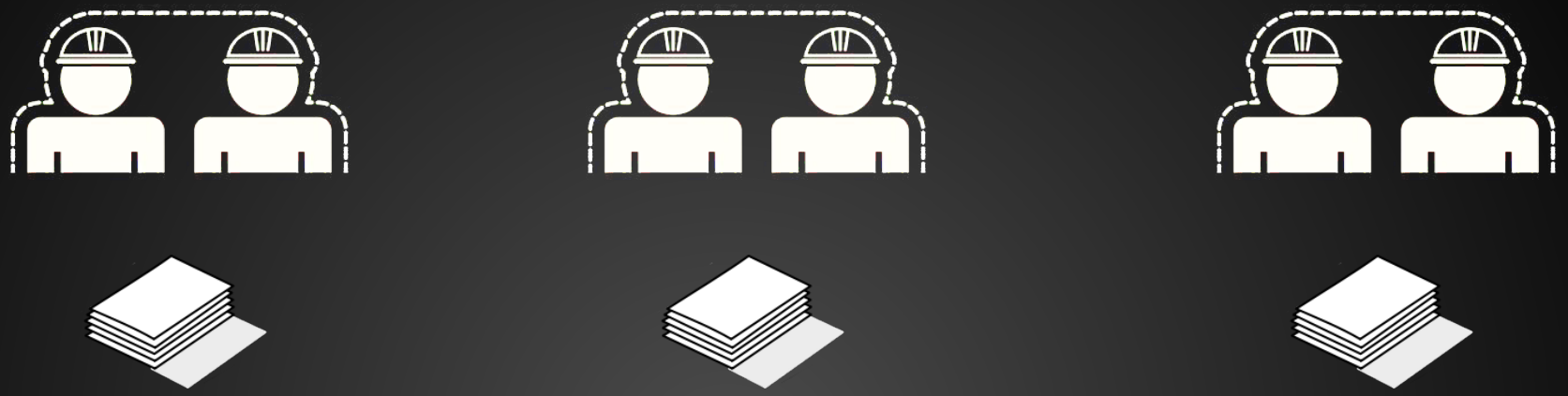


CUDA / OpenCL

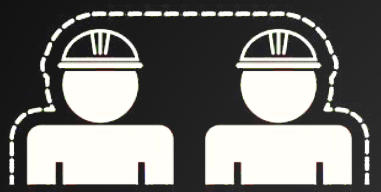
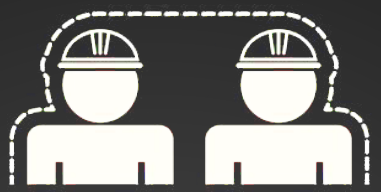
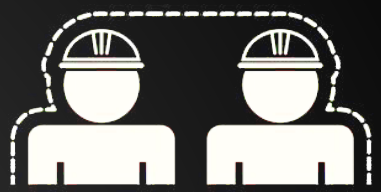
GPU Computing



Computer Cluster Programming

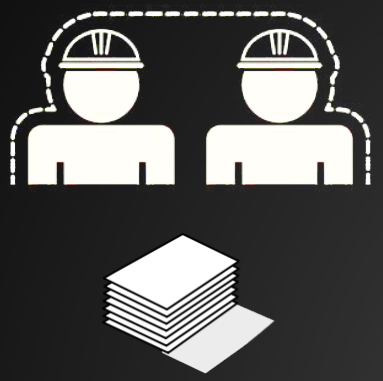


MPI - Message Passing Interface

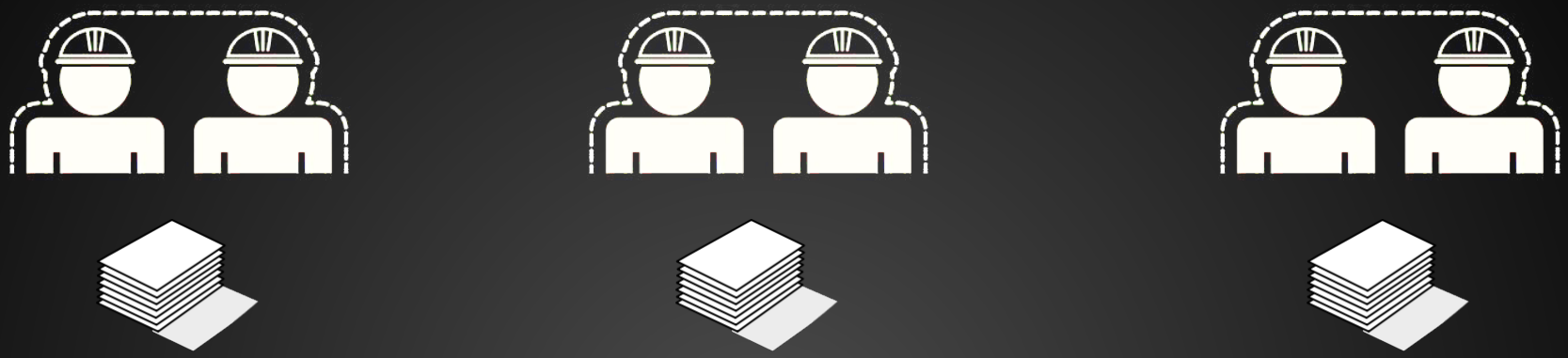

MPICH2 - Baltasar's favorite flavor

Meet Baltasar

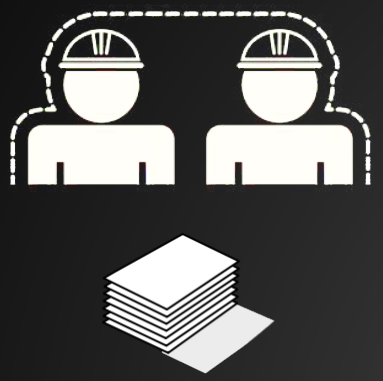


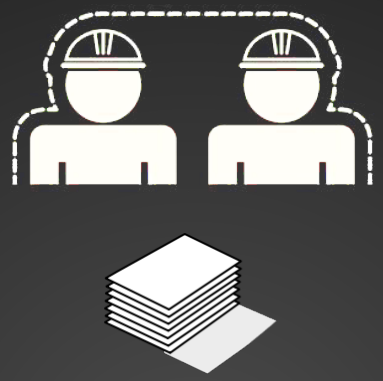
Baltasar Cluster

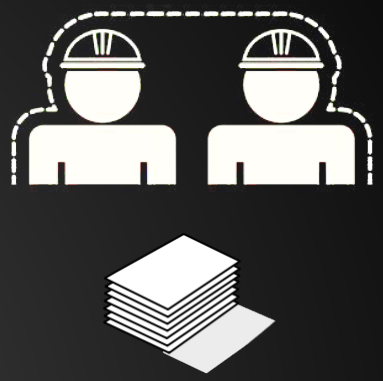
Baltasar Cluster



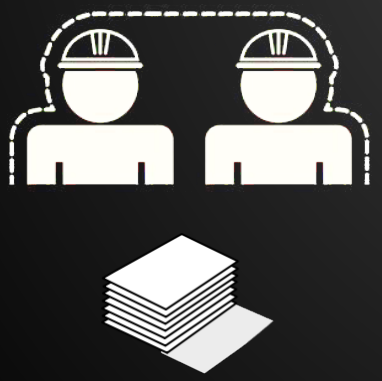
Baltasar Cluster

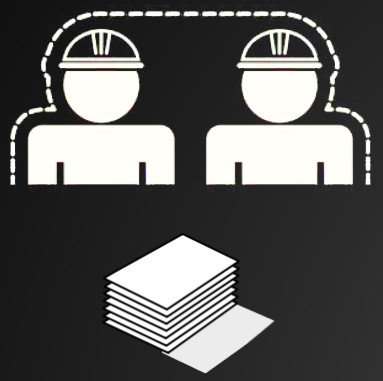
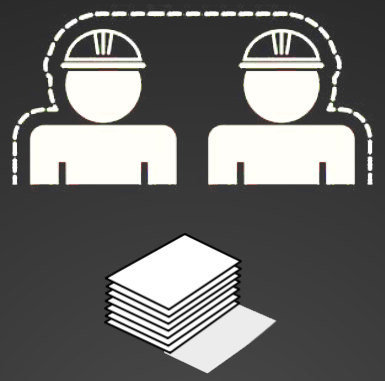
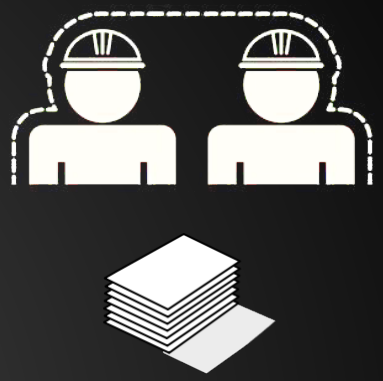




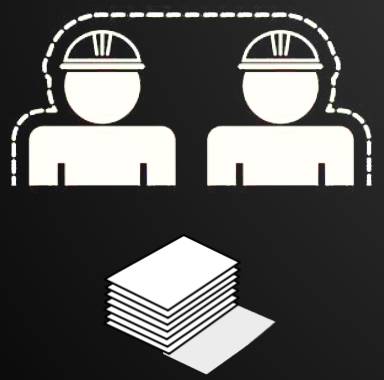
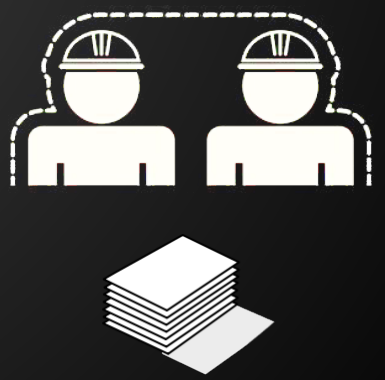


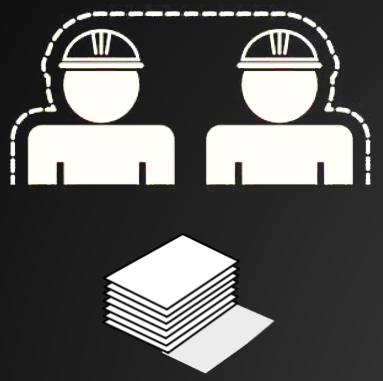
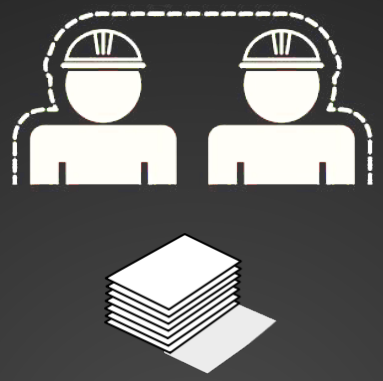
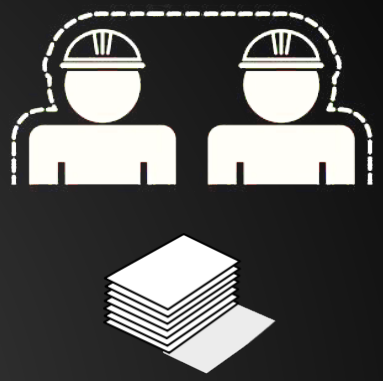
Baltasar Cluster



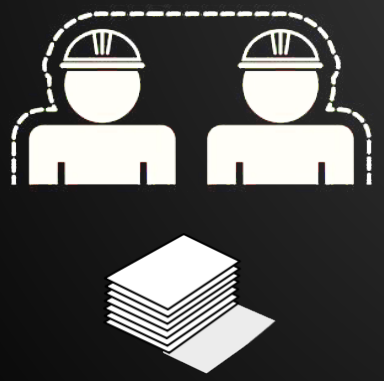
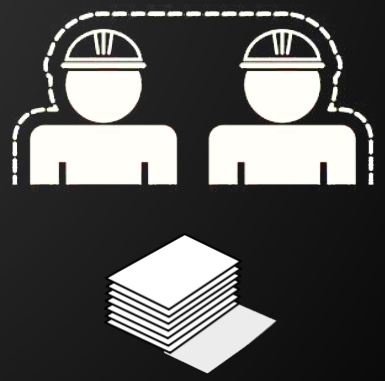




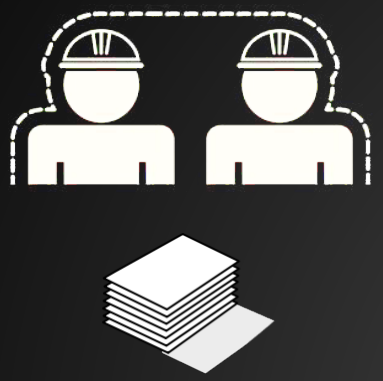
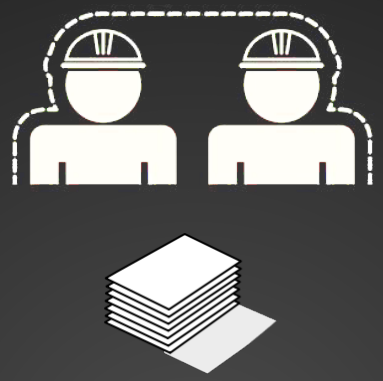
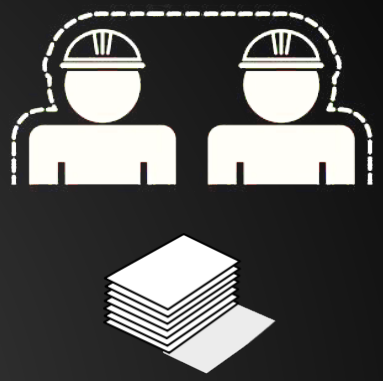
5 Computers

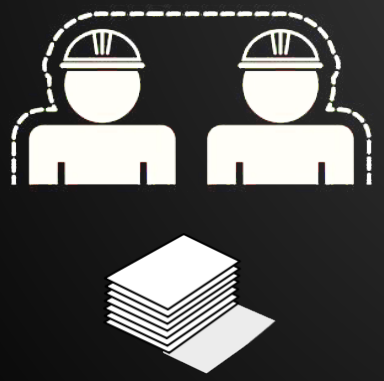
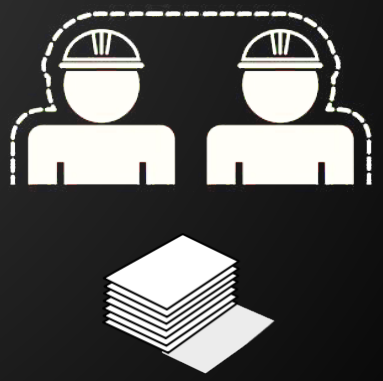




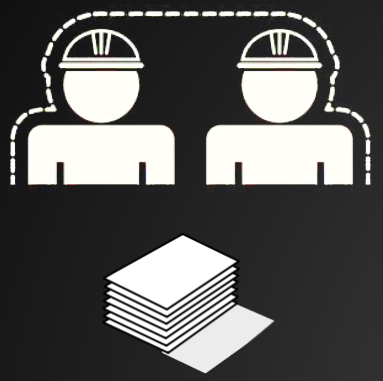
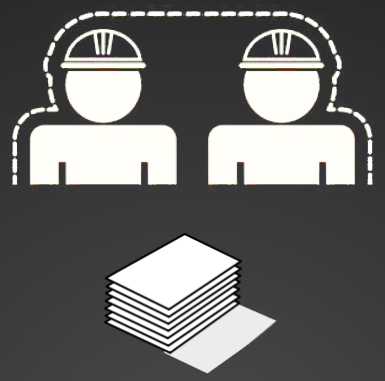
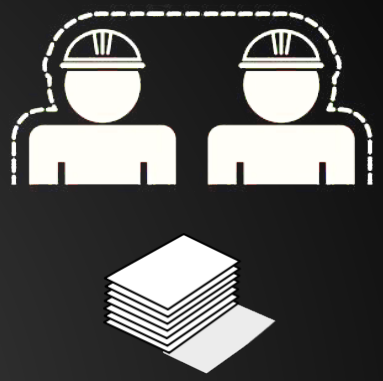
48 Processors per Computer

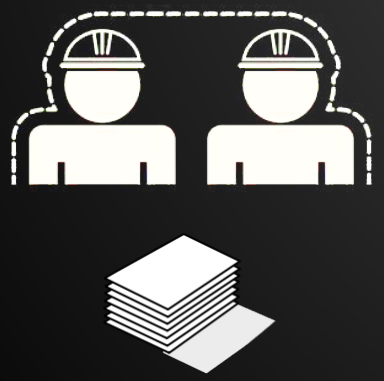
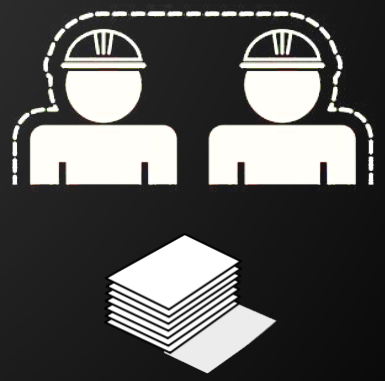




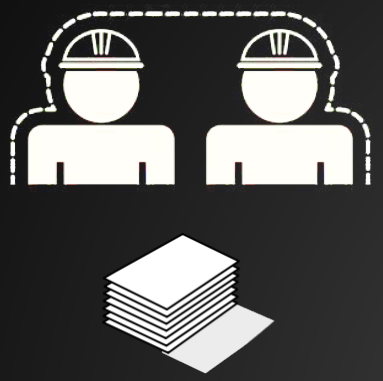
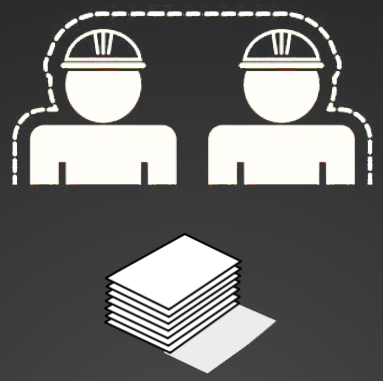
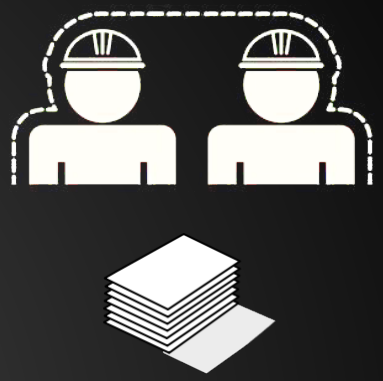
256GB Memory per Computer

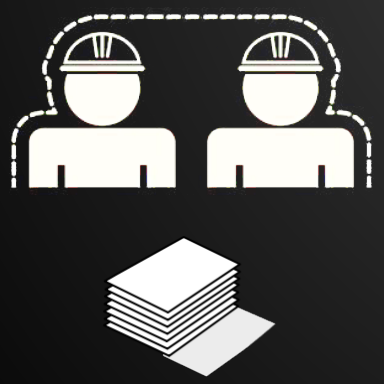




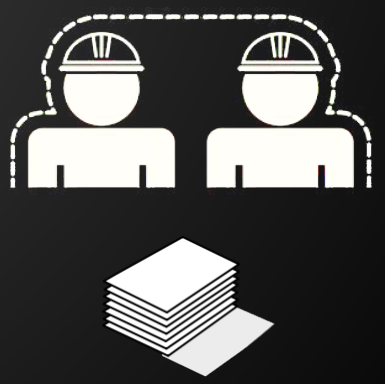
Lightning fast 10Gb Network

Lightning fast shared storage





High Performance Computing

We did our part

Now its up to you

Analyse your program

Estimate resources

Is parallelization needed?

Ready to run

PBS Queues

What is a job?

A job is a PBS script yet to be ran.

A job is a PBS script yet to be a run.

A job turns into a run when the PBS queue decides to run it. Easy.

Writing your PBS Scripts

The Job/The PBS Script

```
#!/bin/bash
#PBS -o /home/username/output.log
#PBS -S /bin/bash
#PBS -l walltime=12:00:00
#PBS -l nodes=1:ppn=12
#PBS -l mem=48GB

PARAMETERS="data1.in"
RUNPATH=/home/username/
cd $RUNPATH

mpiexec.osc ./program $PARAMETERS
```

OSC mpiexec

"Be the human, not the robot"

PBS Script generation

```
baltasar ~/runs/one/$
```

Submitting your jobs

```
baltasar ~/runs/one/$ qsub one.pbs
1337.baltasar.ist.utl.pt
baltasar ~/runs/one/$
```

Done

Is it running?

```
baltasar ~/runs/one/$ qstat -a
baltasar.ist.utl.pt:
1337.baltasar.is    user  batch one.pbs  59468   1
48 128gb 00:00 E 00:30
```

qstat tells you the state of your jobs

Are there free slots to run my job?

```
baltasar ~/runs/one/$ pbstop
Usage Totals: 0/240 Procs, 0/5 Nodes, 0/0 Jobs
Node States:  3 free
(...)
```

Cluster overview with pbstop

```
baltasar ~/runs/one/$ pbsnodes
(...)
```

Individual Node state

Checkpointing

Limitations

Questions?