BSCTools
Hands-On

Judit Giménez

Barcelona Supercomputing Center

Installing Paraver

Install Paraver in your laptop

- Download the Paraver binaries from leftaru to your machine
- Built-in packages:
  - wxparaver-4.5.8-linux-x86_32.tar.gz
  - wxparaver-4.5.8-linux-x86_64.tar.gz wxparaver-4.5.8-linux_fc6-x86_64.tar-gz
  - wxparaver-4.5.8-mac.zip
  - wxparaver-4.5.8-win.zip

Example: Linux 64 bits:

```
scp <uid>@leftraru.nlhpc.cl:~instructor02/paraver/binaries/wxparaver-4.5.8-linux-x86_64.tar.gz .
```

Install Paraver in your laptop (II)

- Uncompress into your home directory

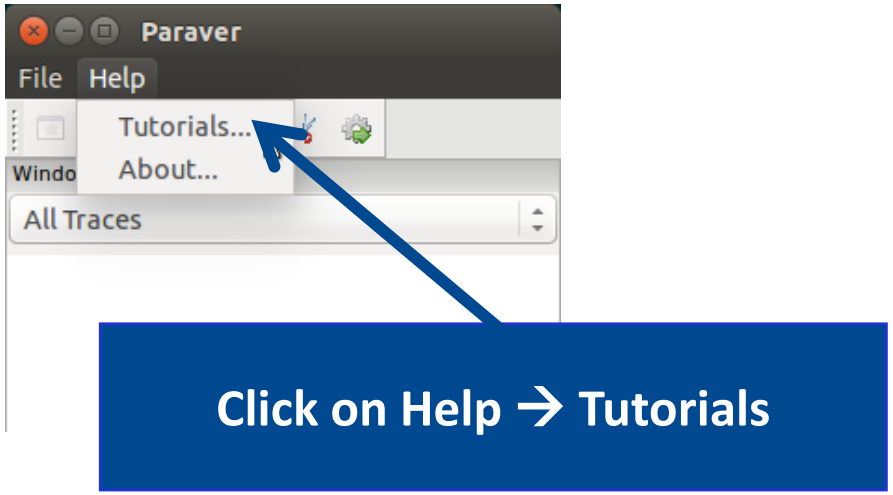
```
> tar xvfz wxparaver-4.5.6-linux-x86_64.tar.gz -C $HOME
> cd $HOME
> ln -s wxparaver-4.5.6-linux-x86_64 paraver
```

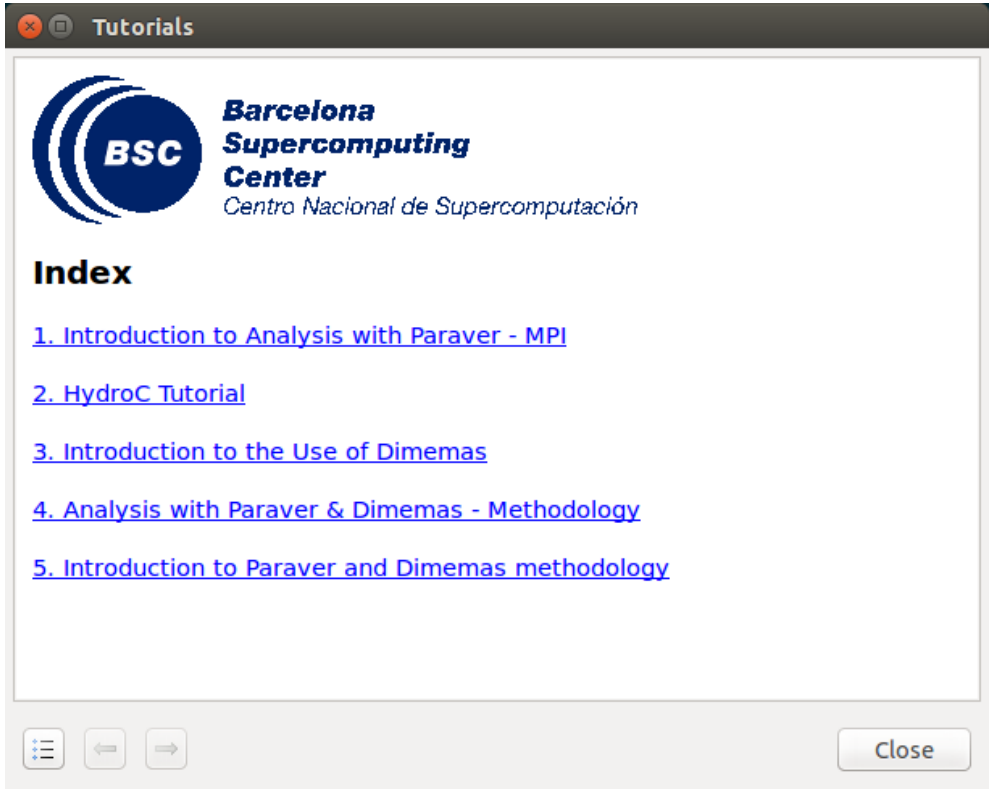
- Download Paraver tutorials and uncompress into the Paraver directory

```
> scp <uid>@leftraru.nlhpc.cl:~instructor02/paraver/paraver-tutorials-
20150526.tar.gz .
> tar xvfz paraver-tutorials-20150526.tar.gz -C $HOME/paraver
```

Launch Paraver

```
> $HOME/paraver/bin/wxparaver
```





Getting a trace with Extrae

Extrae features

- Parallel programming models
  - MPI, OpenMP, pthreads, OmpSs, CUDA, OpenCL, Intel MIC…
- Performance Counters
  - Using PAPI and PMAPI interfaces
- Link to source code
  - Callstack at MPI routines
  - OpenMP outlined routines and their containers
  - Selected user functions
- Periodic samples
- User events (Extrae API)

No need to recompile / relink!

How does Extrae work?

- Symbol substitution through LD_PRELOAD
  - Specific libraries for each combination of runtimes
    - MPI
    - OpenMP
    - OpenMP+MPI
    - …

- Dynamic instrumentation (not yet available at leftraru)
  - Based on DynInst (developed by U.Wisconsin/U.Maryland)
    - Instrumentation in memory
    - Binary rewriting

- Alternatives
  - Static link (i.e., PMPI, Extrae API)

How to use Extrae?

1. Adapt the job submission script

2. [Optional] Tune the Extrae XML configuration file
   - Examples distributed with Extrae at $EXTRAE_HOME/share/example

3. Run it!

- For further reference check the **Extrae User Guide:**
  - Also distributed with Extrae at $EXTRAE_HOME/share/doc

  http://www.bsc.es/computer-sciences/performance-tools/documentation

Log in to leftraru and copy the example to your home directory

```
> ssh <uid>@leftraru.nlhpc.cl

> cp -r ~instructor02/paraver_handson $HOME

> cd $HOME/paraver_handson/aplic/lulesh
```

Example: Adapt the job script to load Extrae with LD_PRELOAD

run.sh

```
#!/bin/bash
#SBATCH --job-name=lulesh
#SBATCH --partition=slims
#SBATCH -n 27
#SBATCH --ntasks-per-node=14
#SBATCH --output=lulesh_%j.out
#SBATCH --error=lulesh_%j.err
#SBATCH --exclusive=user
#SBATCH --mail-type=ALL

module load intel impi

srun $TRACE ./lulesh2.0 -i 10 -p -s 65
```

Example: Adapt the job script to load Extrae with LD_PRELOAD

job.sh

```
#!/bin/bash
#SBATCH --job-name=lulesh
#SBATCH --partition=slims
#SBATCH -n 27
#SBATCH --ntasks-per-node=14
#SBATCH --output=lulesh_%j.out
#SBATCH --error=lulesh_%j.err
#SBATCH --exclusive=user
#SBATCH --mail-type=ALL

module load intel impi

module load extrae

TRACE=./extrae/trace.sh
srun $TRACE ./lulesh2.0 -i 10 -p -s 65
$EXTRAE_HOME/bin/mpi2prv -f TRACE.mpits -o
      lulesh2.0.prv
```

extrae/trace.sh

```
#!/bin/bash

# Configure Extrae
export EXTRAE_CONFIG_FILE=extrae/extrae-config.xml

# Load the tracing library (C/Fortran)
export LD_PRELOAD=${EXTRAE_HOME}/lib/libmpitrace.so
#export LD_PRELOAD=${EXTRAE_HOME}/lib/libmpitracef.so

# Run the program
$*
```

LD_PRELOAD library selection

- Choose depending on the application type

Library	Serial	MPI	OpenMP	pthread	CUDA
libseqtrace	✓				
libmpitrace[f][1]		✓			
libomptrace			✓		
libpttrace				✓	
libcudatrace					✓
libompitrace[f] [1]		✓	✓		
libptmpitrace[f] [1]		✓		✓	
libcudampitrace[f] [1]		✓			✓

[1] include suffix "f" in Fortran codes

Extrae XML configuration: extrae_config.xml

```
<mpi enabled="yes">
  <counters enabled="yes" />
</mpi>

<openmp enabled="yes">
  <locks enabled="no" />
  <counters enabled="yes" />
</openmp>

<pthread enabled="no">
  <locks enabled="no" />
  <counters enabled="yes" />
</pthread>

<callers enabled="yes">
  <mpi enabled="yes">1-3</mpi>
  <sampling enabled="no">1-5</sampling>
</callers>
```

Trace MPI calls + HW counters

Trace call-stack events @ MPI calls

Extrae XML configuration: extrae_config.xml (II)

```
<counters enabled="yes">
  <cpu enabled="yes" starting-set-distribution="cyclic">
    <set enabled="yes" changeat-time="500000us" domain="all">
   PAPI_TOT_INS,PAPI_TOT_CYC,PAPI_L1_DCM,PAPI_L2_DCM,PAPI_L3_TCM,PAPI_BR_CN,PAPI_BR_UCN,PAPI_BR_MSP
      </set>
      <set enabled="yes" changeat-time="500000us" domain="all">
   PAPI_TOT_INS,PAPI_TOT_CYC,PAPI_VEC_SP,PAPI_VEC_DP,RESOURCE_STALLS:SB,RESOURCE_STALLS:ROB,RESOURCE_STALLS:RS
       </set>
       <set enabled="yes" changeat-time="500000us" domain="all">
   PAPI_TOT_INS,PAPI_TOT_CYC,PAPI_FP_INS,RESOURCE_STALLS
        </set>
   </cpu>

  <network enabled="no" />

  <resource-usage enabled="no" />

  <memory-usage enabled="no" />
</counters>
```

Define which HW counters are measured

Extrae XML configuration: extrae_config.xml (III)

```
<buffer enabled="yes">
  <size enabled="yes">500000</size>
  <circular enabled="no" />
</buffer>

<sampling enabled="no" type="default" period="50m" variability="10m" />

<merge enabled="no"
       synchronization="default"
       tree-fan-out="16"
       max-memory="512"
       joint-states="yes"
       keep-mpits="yes"
       sort-addresses="yes"
       overwrite="yes"
>
  $TRACE_NAME$
</merge>
```

Trace buffer size

Enable sampling

Merge intermediate files into Paraver trace

Run it!

- Submit your job @leftraru.nlhpc.cl

```
> cd $HOME/paraver-handson/aplic/lulesh
> sbatch job.sh
```

- Copy the resulting trace to your laptop and load it with Paraver @your laptop

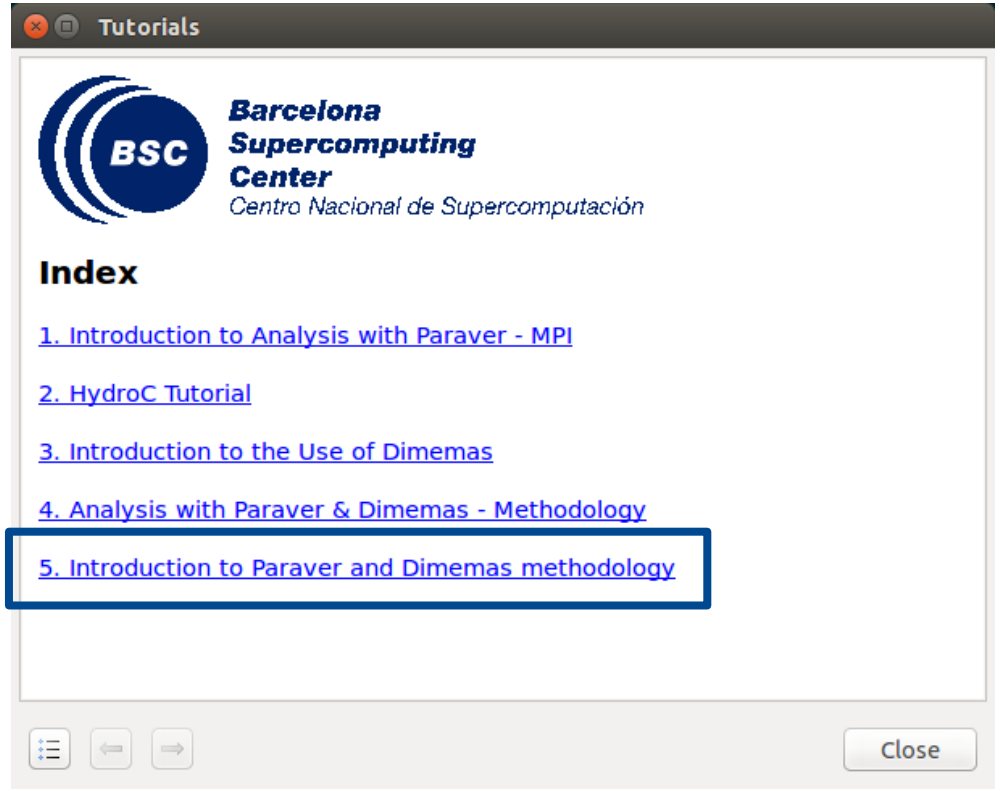
```
> scp <uid>@leftraru.nlhpc.cl:paraver_handson/aplic/lulesh/lulesh2.0.* $HOME
```

- Load the trace with Paraver @your laptop

```
> $HOME/paraver/bin/wxparaver $HOME/lulesh.2.0.prv
```

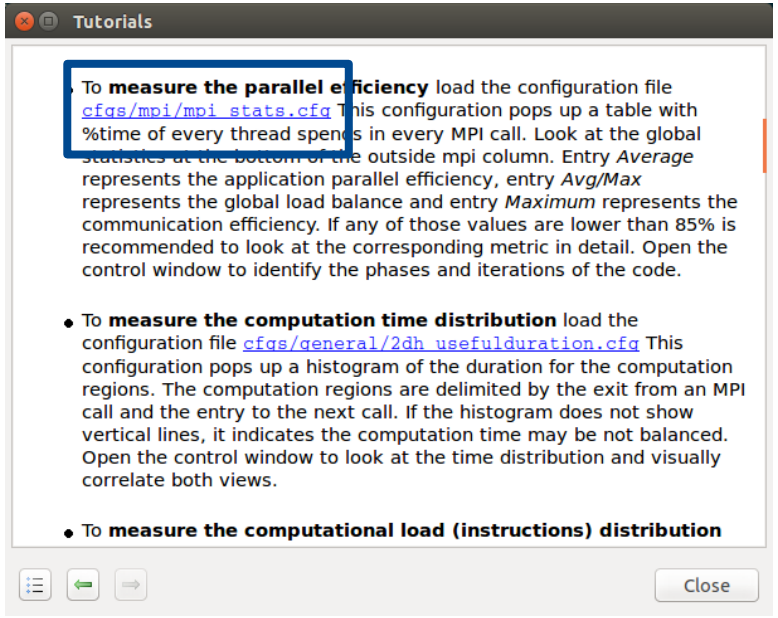
First steps of analysis

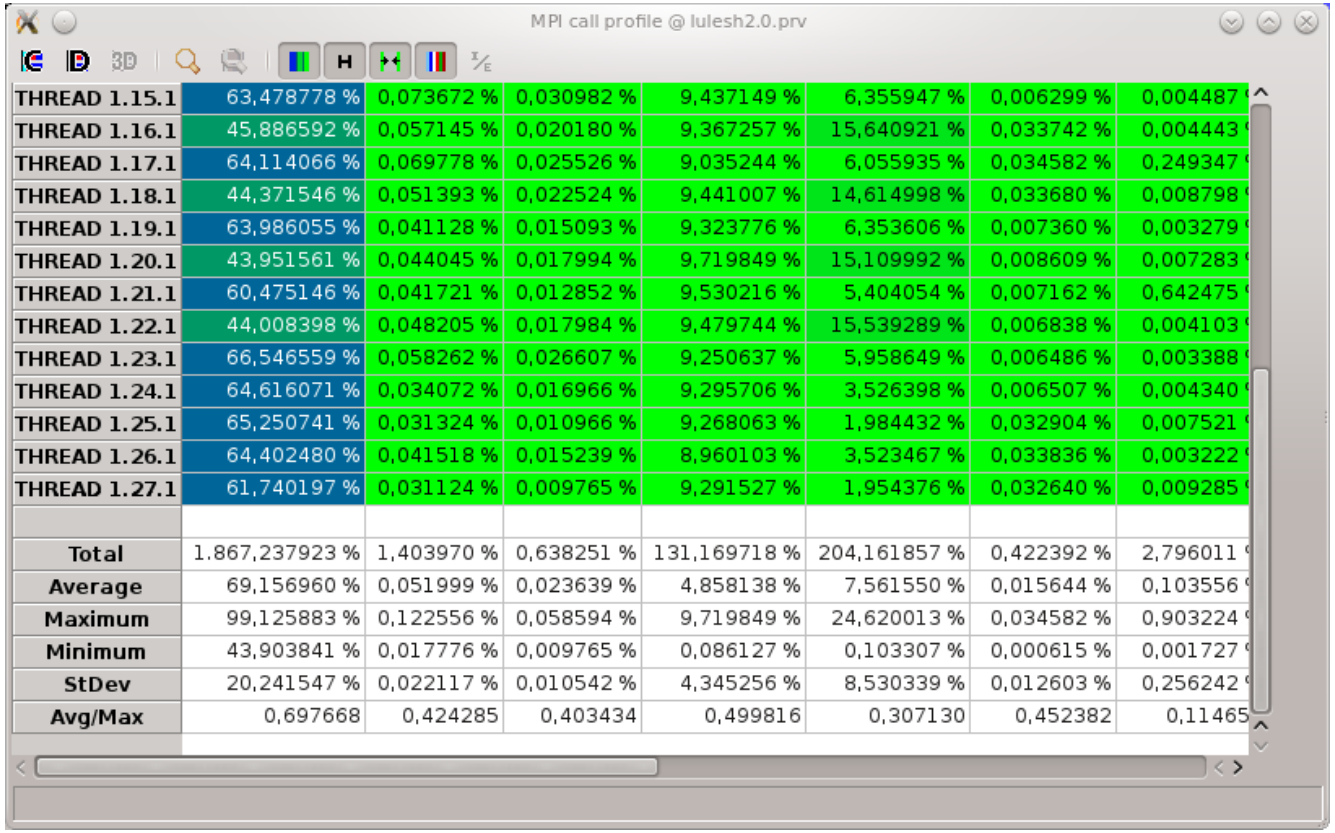
- Open Tutorial #5
  - Help → Tutorials



Measure the parallel efficiency

- Click on the “mpi_stats.cfg”

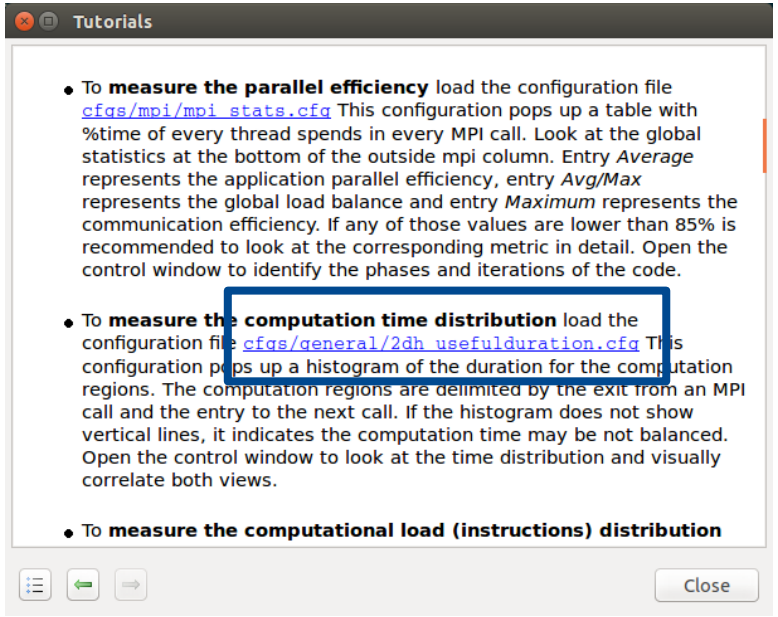


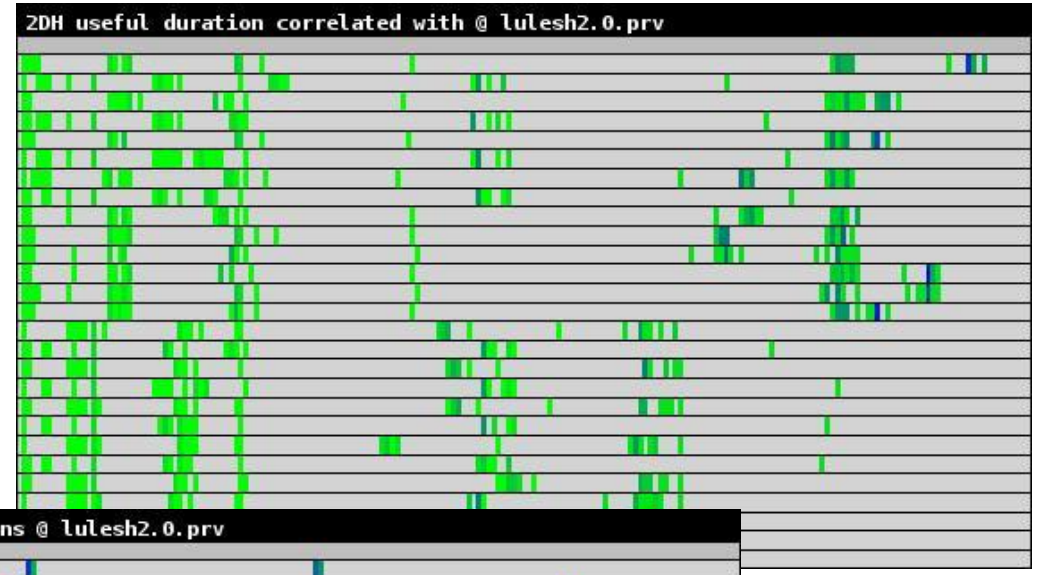


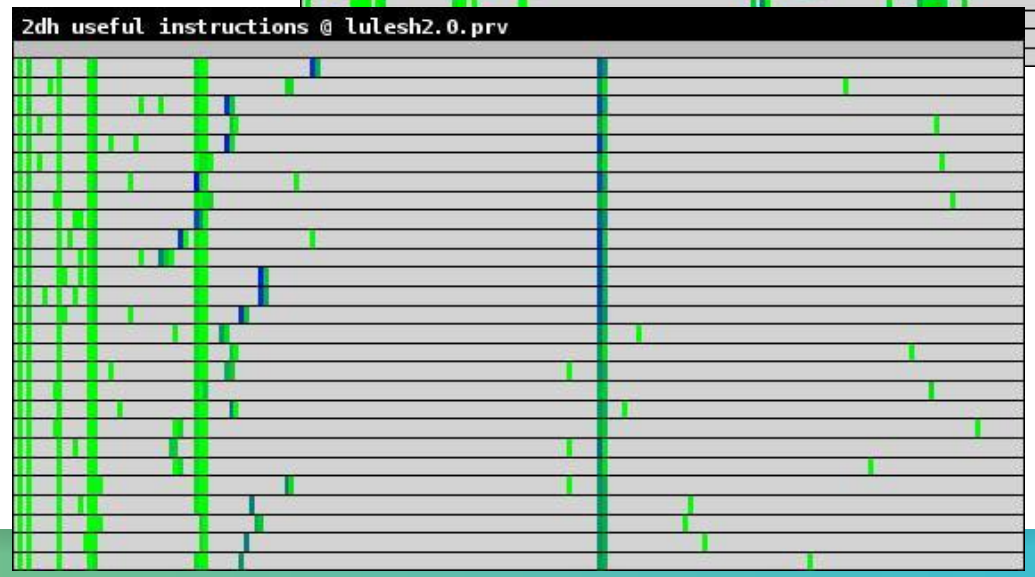
THREAD 1.15.1	63,478778 %	0,073672 %	0,030982 %	9,437149 %	6,355947 %	0,006299 %	0,004487 %
THREAD 1.16.1	45,886592 %	0,057145 %	0,020180 %	9,367257 %	15,640921 %	0,033742 %	0,004443 %
THREAD 1.17.1	64,114066 %	0,069778 %	0,025526 %	9,035244 %	6,055935 %	0,034582 %	0,249347 %
THREAD 1.18.1	44,371546 %	0,051393 %	0,022524 %	9,441007 %	14,614998 %	0,033680 %	0,008798 %
THREAD 1.19.1	63,986055 %	0,041128 %	0,015093 %	9,323776 %	6,353606 %	0,007360 %	0,003279 %
THREAD 1.20.1	43,951561 %	0,044045 %	0,017994 %	9,719849 %	15,109992 %	0,008609 %	0,007283 %
THREAD 1.21.1	60,475146 %	0,041721 %	0,012852 %	9,530216 %	5,404054 %	0,007162 %	0,642475 %
THREAD 1.22.1	44,008398 %	0,048205 %	0,017984 %	9,479744 %	15,539289 %	0,006838 %	0,004103 %
THREAD 1.23.1	66,546559 %	0,058262 %	0,026607 %	9,250637 %	5,958649 %	0,006486 %	0,003388 %
THREAD 1.24.1	64,616071 %	0,034072 %	0,016966 %	9,295706 %	3,526398 %	0,006507 %	0,004340 %
THREAD 1.25.1	65,250741 %	0,031324 %	0,010966 %	9,268063 %	1,984432 %	0,032904 %	0,007521 %
THREAD 1.26.1	64,402480 %	0,041518 %	0,015239 %	8,960103 %	3,523467 %	0,033836 %	0,003222 %
THREAD 1.27.1	61,740197 %	0,031124 %	0,009765 %	9,291527 %	1,954376 %	0,032640 %	0,009285 %
Total	1.867,237923 %	1,403970 %	0,638251 %	131,169718 %	204,161857 %	0,422392 %	2,796011 %
Average	69,156960 %	0,051999 %	0,023639 %	4,858138 %	7,561550 %	0,015644 %	0,103556 %
Maximum	99,125883 %	0,122556 %	0,058594 %	9,719849 %	24,620013 %	0,034582 %	0,903224 %
Minimum	43,903841 %	0,017776 %	0,009765 %	0,086127 %	0,103307 %	0,000615 %	0,001727 %
StDev	20,241547 %	0,022117 %	0,010542 %	4,345256 %	8,530339 %	0,012603 %	0,256242 %
Avg/Max	0,697668	0,424285	0,403434	0,499816	0,307130	0,452382	0,11465

Measure the computation time distribution

- Click on the “2dh_usefulduration.cfg”







…and the “2dh_useful_instructions.cfg”

Compare with other configurations

- Sockets more balanced (max 7 tasks per socket)
  - If you want to get the trace…

```
> cd $HOME/paraver_handson/aplic/lulesh
> sbatch job2.sh
```

  - It is already generated at
    `~instructor02/paraver_handson2/traces/lulesh/lulesh2.0_ts.prv`

  - To copy all the traces…

```
➢ cp –r ~instructor02/paraver_handson2 $HOME
```

Use clustering to analyse and compare both runs

- Run clustering

```
> cd $HOME/paraver-handson/aplic/lulesh
➢ module load clustering
➢ $HOME/paraver-handson/bin/clusterize.sh lulesh2.0
```

- Look at the results @leftaru

```
> gnuplot *ot
```

- Look at the results @your laptop

```
➢ scp <uid>@leftraru.nlhpc.cl:paraver_handson/aplic/lulesh/*clustered* $HOME
➢ $HOME/paraver/bin/wxparaver $HOME/lulesh.2.0_clustered.prv
```

- It is already generated at `$HOME/paraver_handson2/traces/lulesh`

Looking at the clusters

- Click on the “cluster_id.cfg” and “2d_usefulduration_vs_cluster.cfg”