

## BSCTools Hands-On

---

---

Judit Giménez, Jesús Labarta  
Barcelona Supercomputing Center

---

---

## Getting a trace with Extrae

## Extrae features

---

- Parallel programming model
  - MPI, OpenMP<sup>(\*)</sup>, pthreads, OmpSs, CUDA, OpenCL...Java, Python
- 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)
- Platforms: Intel, Cray, BlueGene, Fujitsu Sparc, Intel MIC, ARM, Android

**No need to  
recompile / relink!**

(\*) GNU, Inttel, OMPT

## 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)



**Recommended**

## 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 with instrumentation
    - 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 pi and copy the example to your home directory

---

```
> ssh -X <uid>@pi.ircpi.kobe-u.ac.jp  
> cp -r /home/S11505/shared/tutorial/ntchem-mini $HOME  
> cd $HOME/ntchem-mini/run_h2o
```

# Adapt the job script to load Extrae with LD\_PRELOAD

---

## PIcomputer.sh

```
#!/bin/bash -x
#
#PJM -o "h2o_10_rimp2.out"
#PJM -e "h2o_10_rimp2.err"
#PJM --rsc-list "rscgrp=small"
#PJM --rsc-list "node=2"
#PJM --rsc-list "elapsed=0:10:00"
##PJM --mpi "proc=32"
#PJM -j
#

export FLIB_FASTOMP=FALSE
export FLIB_CNTL_BARRIER_ERR=FALSE
export OMP_NUM_THREADS=16

mpiexec ./rimp2.exe_debug
```

# Adapt the job script to load Extrae with LD\_PRELOAD

## Picomputer\_extrae.sh

```
#!/bin/bash -x
#
#PJM -o "h2o_10_rimp2.out"
#PJM -e "h2o_10_rimp2.err"
#PJM --rsc-list "rscgrp=small"
#PJM --rsc-list "node=2"
#PJM --rsc-list "elapse=0:10:00"
##PJM --mpi "proc=32"
#PJM -j
#
xospastop

export FLIB_FASTOMP=FALSE
export FLIB_CNTL_BARRIER_ERR=FALSE
export OMP_NUM_THREADS=16

source extrae/trace.sh
export TRACE=t_ntchem_h2o.prv

mpiexec ./rimp2.exe_debug
```

## extrae/trace.sh

```
#!/bin/bash
export EXTRAE_HOME=/home/S11505/shared/tools/BSC/extrae-3.3.0/

export EXTRAE_CONFIG_FILE=extrae/extrae.xml

export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:
/home/S11505/judit/dependencies/binutils-2.24/lib

export LD_PRELOAD=$EXTRAE_HOME/lib/libmpitracef.so:libpapi.so
```



## LD\_PRELOAD library selection

- Choose depending on the application type

| Library                         | Serial | MPI | OpenMP | pthread | CUDA |
|---------------------------------|--------|-----|--------|---------|------|
| libseqtrace                     | ✓      |     |        |         |      |
| libmpitrace[f] <sup>1</sup>     |        | ✓   |        |         |      |
| libompitrace                    |        |     | ✓      |         |      |
| libpttrace                      |        |     |        | ✓       |      |
| libcudatrace                    |        |     |        |         | ✓    |
| libompitrace[f] <sup>1</sup>    |        | ✓   | ✓      |         |      |
| libptmpitrace[f] <sup>1</sup>   |        | ✓   |        | ✓       |      |
| libcudampitrace[f] <sup>1</sup> |        | ✓   |        |         | ✓    |

<sup>1</sup> include suffix "f" in Fortran codes

# Extrac XML configuration: extrac\_config.xml

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

Trace MPI calls + HW counters

```
<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 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_TCM,PAPI_BR_MSP,PAPI_FP_INS
    </set>
    <set enabled="yes" changeat-time="500000us" domain="all">
      PAPI_TOT_INS,PAPI_TOT_CYC,PAPI_SR_INS,PAPI_LD_INS
    </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>
```

Trace buffer size

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

```
<merge enabled="yes"  
  synchronization="default"  
  tree-fan-out="16"  
  max-memory="512"  
  joint-states="yes"  
  keep-mpits="yes"  
  sort-addresses="yes"  
  overwrite="yes"
```

Merge intermediate  
files into Paraver  
trace

```
>  
  $TRACE$  
</merge>
```

# Run with instrumentation

---

- Submit your job

@pi.ircpi.kobe-u.ac.jp

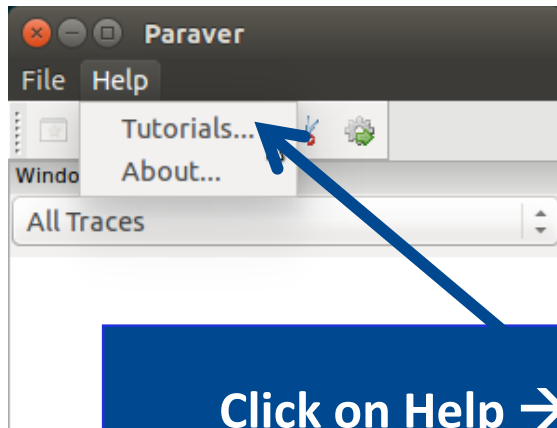
```
> cd $HOME/ntchem-mini/run_h2o  
> pjsub PIcomputer_extrae.sh
```

- Load the trace with Paraver

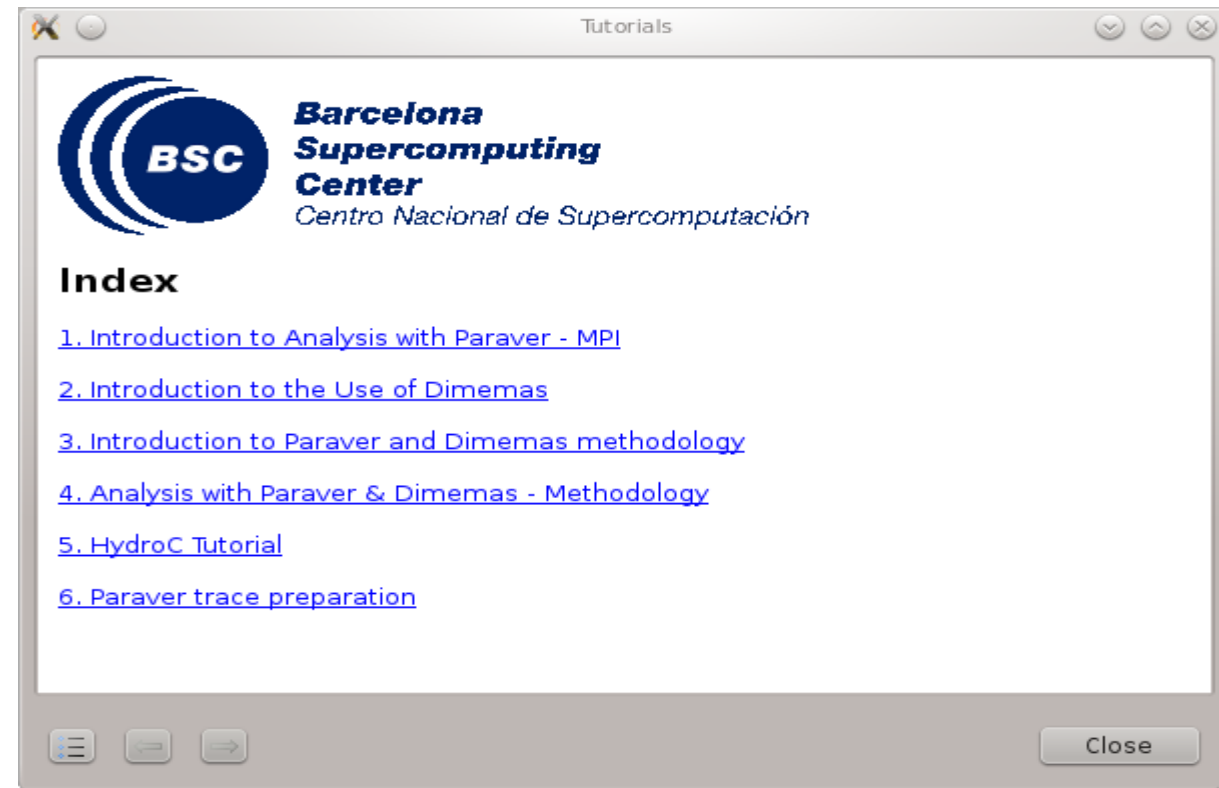
```
> wxparaver t_nchem_h2o.prv
```

# Launch Paraver

```
> source /home/S11505/shared/tools/setup.sh  
> wxparaver
```



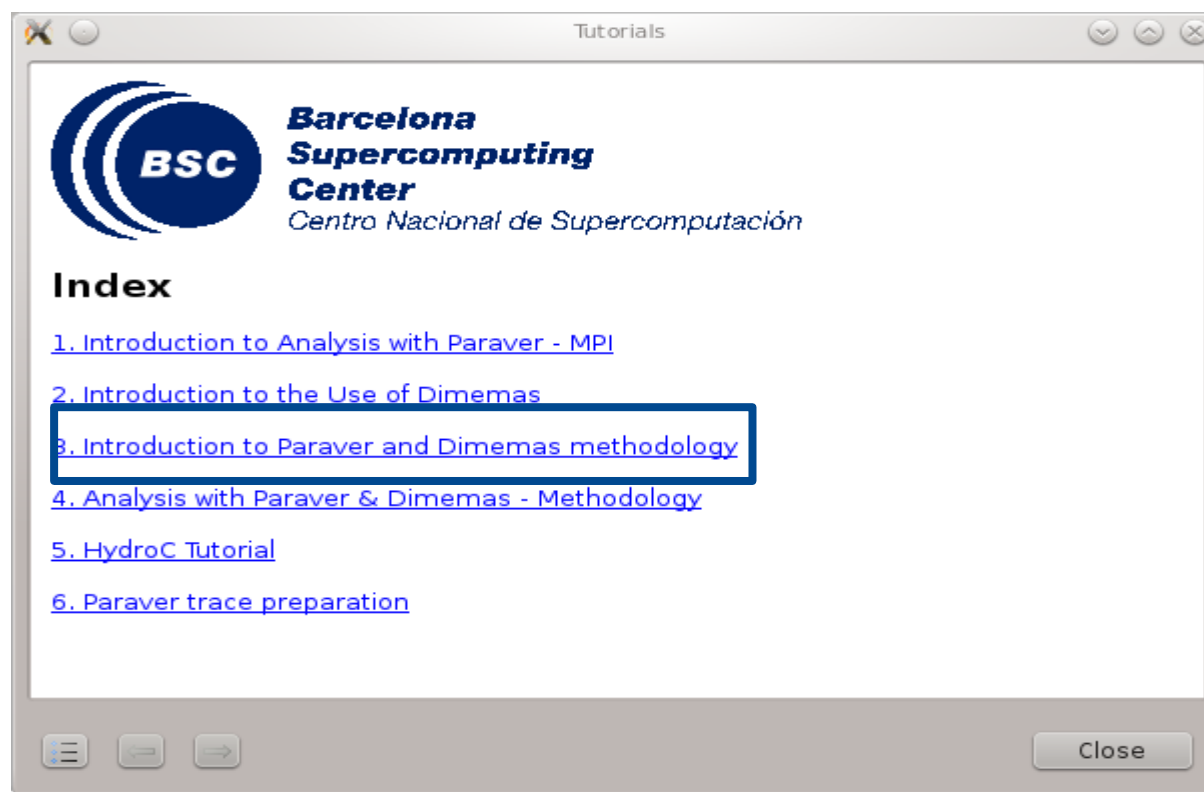
Click on Help → Tutorials



# First steps of analysis

---

- Open Tutorial #3
  - Help → Tutorials



# Measure the parallel efficiency

- Click on the “mpi\_stats.cfg”

**Tutorials**

To **measure the parallel efficiency** load the configuration file [cfigs/mpi/mpi\\_stats.cfg](#). This configuration pops up a table with %time of every thread spends in every MPI call. Look at the global statistics at the bottom of the outside mpi column. Entry *Average* represents the application parallel efficiency, entry *Avg/Max* represents the global load balance and entry *Maximum* represents the communication efficiency. If any of those values are lower than 85% is recommended to look at the corresponding metric in detail. Open the control window to identify the phases and iterations of the code.

- To **measure the computation time distribution** load the configuration file [cfigs/general/2dh\\_usefulduration.cfg](#). This configuration pops up a histogram of the duration for the computation regions. The computation regions are delimited by the exit from an MPI call and the entry to the next call. If the histogram does not show vertical lines, it indicates the computation time may be not balanced. Open the control window to look at the time distribution and visually correlate both views.
- To **measure the computational load (instructions) distribution**

Close

MPI call profile @ t\_ntchem\_h2o.prv.gz

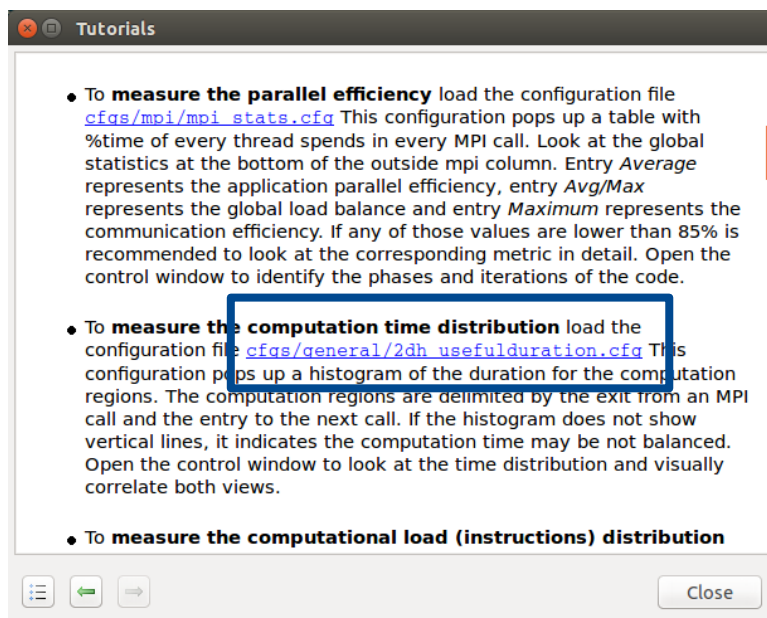
|                | Outside MPI  | MPI_Isend  | MPI_Irecv  | MPI_Wait    | MPI_Bcast  | MPI_Barrier | MPI_Allreduce |
|----------------|--------------|------------|------------|-------------|------------|-------------|---------------|
| THREAD 1.1.1   | 98,263954 %  | 0,024848 % | 0,009296 % | 0,093815 %  | 0,001019 % | 0,002378 %  | 1,592869 %    |
| THREAD 1.2.1   | 85,404495 %  | 0,023767 % | 0,009267 % | 10,205695 % | 0,015961 % | 0,053610 %  | 4,277246 %    |
| <b>Total</b>   | 183,668448 % | 0,048615 % | 0,018564 % | 10,299509 % | 0,016979 % | 0,055988 %  | 5,870115 %    |
| <b>Average</b> | 91,834224 %  | 0,024308 % | 0,009282 % | 5,149755 %  | 0,008490 % | 0,027994 %  | 2,935058 %    |
| <b>Maximum</b> | 98,263954 %  | 0,024848 % | 0,009296 % | 10,205695 % | 0,015961 % | 0,053610 %  | 4,277246 %    |
| <b>Minimum</b> | 85,404495 %  | 0,023767 % | 0,009267 % | 0,093815 %  | 0,001019 % | 0,002378 %  | 1,592869 %    |
| <b>StDev</b>   | 6,429730 %   | 0,000540 % | 0,000015 % | 5,055940 %  | 0,007471 % | 0,025616 %  | 1,342189 %    |
| <b>Avg/Max</b> | 0,934567     | 0,978260   | 0,998425   | 0,504596    | 0,531911   | 0,522178    | 0,686203      |

Outside MPI



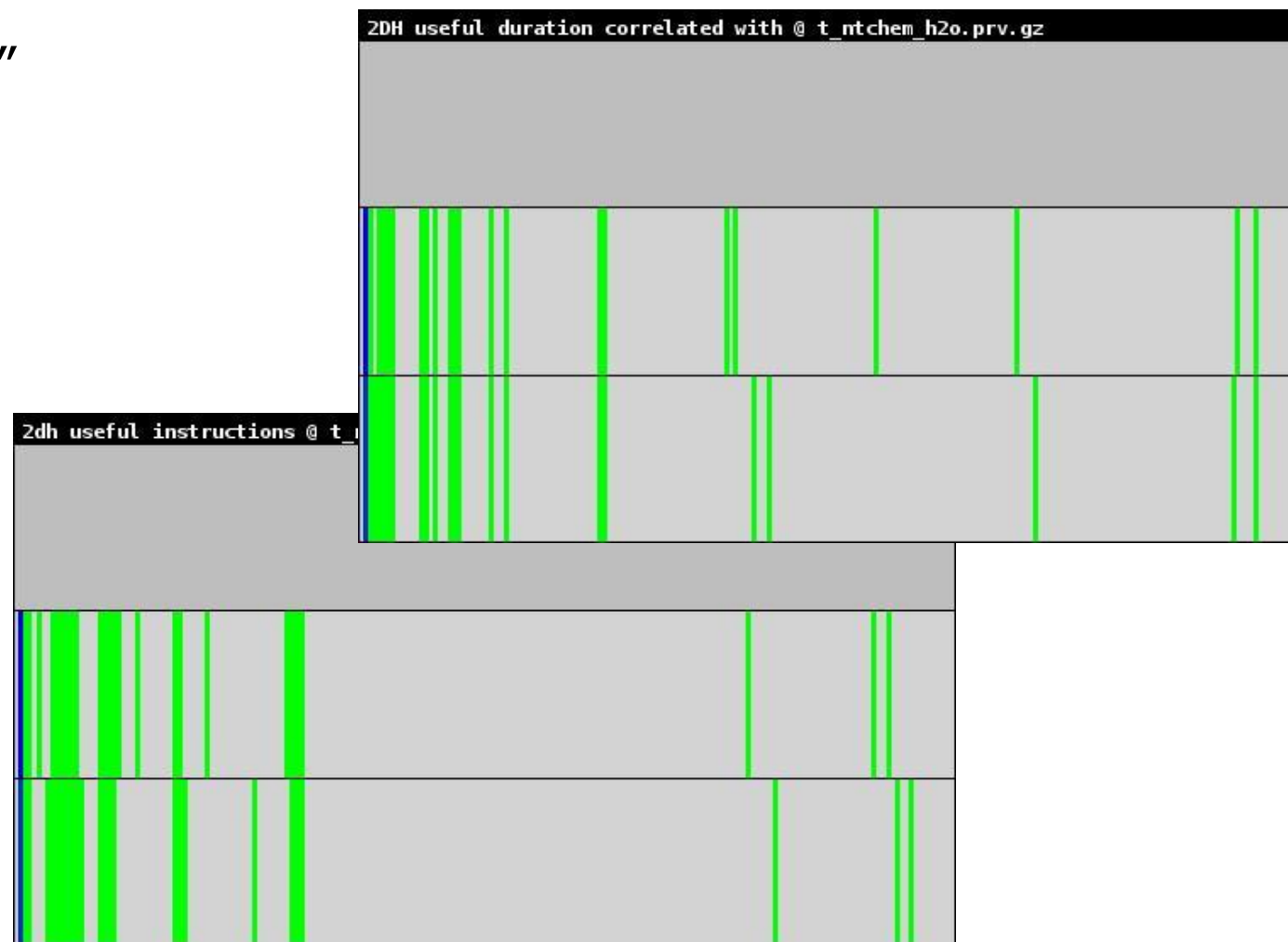
# Measure the computation time distribution

- Click on the "2dh\_usefulduration.cfg"



...and the "2dh\_useful\_instructions.cfg"

Zoom on the relevant (dark) area



## Compare with other configurations

---

- Taxol test case (f.i. 32 nodes)
  - If you want to get the trace...

```
> cd $HOME/ntchem_mini/run_taxol  
> pjsub PIcomputer_extrae.sh
```

- It is already generated at

```
➤ cd $HOME/ntchem_mini/traces/taxol
```

## Use clustering to analyse and compare both runs

---

- Run clustering

```
> cd $HOME/ntchem_mini/run_h2o  
➤ clusterize.sh t_ntchem_h2o
```

- Look at the results

```
> gnuplot t_ntchem_h2o_clustered.IPC.PAPI_TOT_INS.gnuplot
```

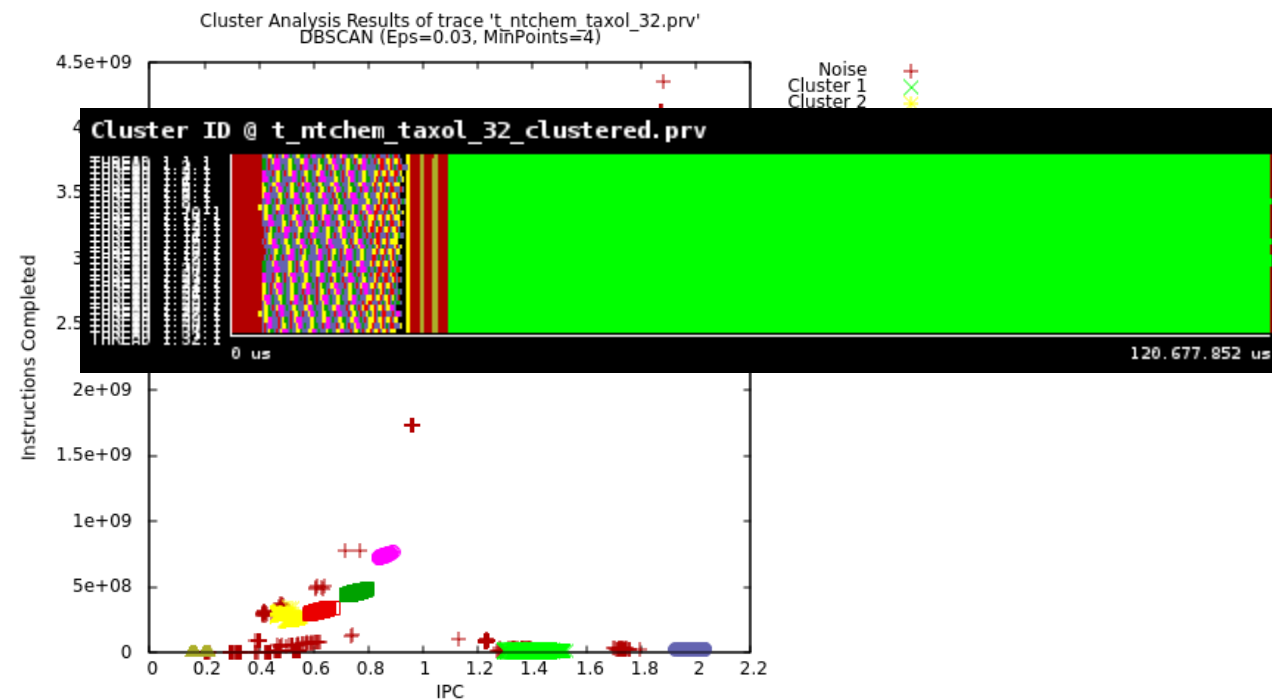
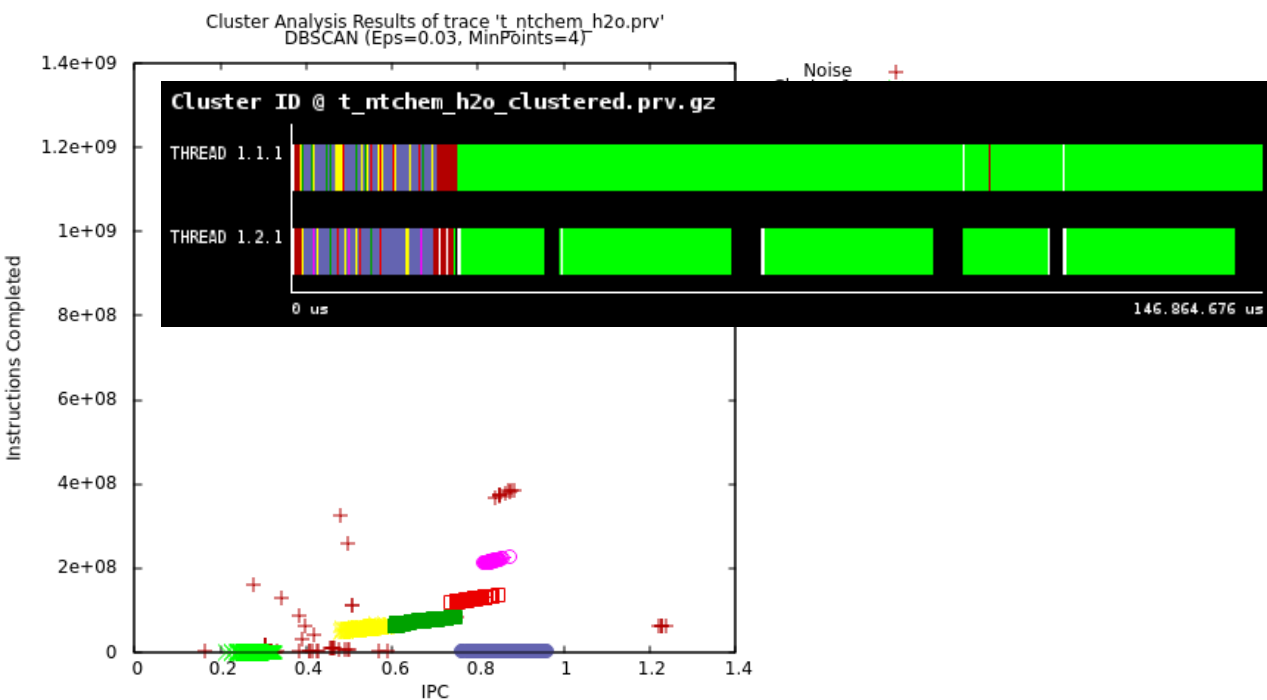
- Look at the resulting trace

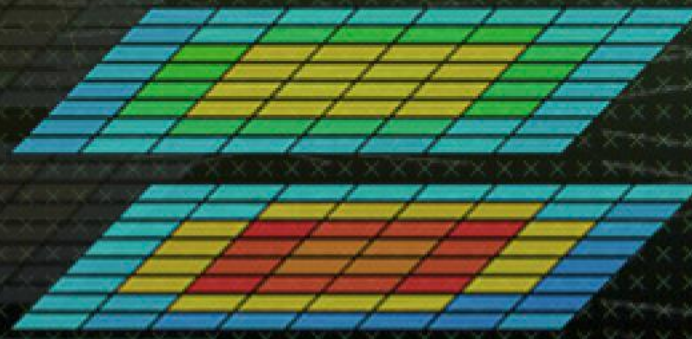
```
➤ wxparaver $HOME/ntchem_mini/run_h2o/t_ntchem_h2o_clustered.prv
```

- It is already generated at `$HOME/ntchem_mini/traces/h2o`

# Looking at the clusters

- Click on the "cluster\_id.cfg"





# Installing Paraver

## Install Paraver in your laptop

---

- Download the Paraver binaries from leftaru to your machine
- Built-in packages:
  - wxparaver-4.6.0-linux-x86\_32.tar.gz
  - wxparaver-4.6.0-linux-x86\_64.tar.gz wxparaver-4.6.0-linux\_fc6-x86\_64.tar-gz
  - wxparaver-4.6.0-mac.zip
  - wxparaver-4.6.0-win.zip

Example: Linux 64 bits:

```
scp <uid>@pi.ircpi.kobe-u.ac.jp:/home/S11505/shared/tools/BSC/paraver_pkgs/wxparaver-4.5.8-linux-x86_64.tar.gz .
```

## Install Paraver in your laptop (II)

---

- Uncompress into your home directory

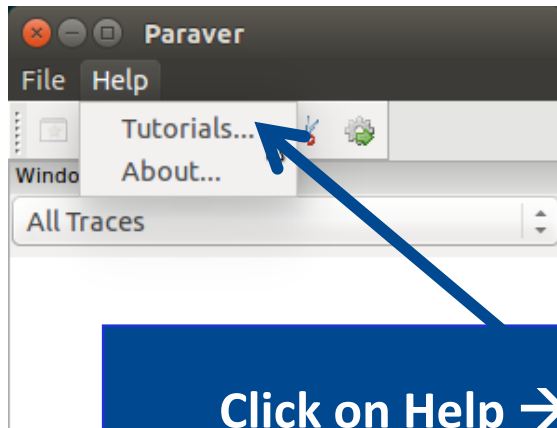
```
> tar xvfz wxparaver-4.5.6-linux-x86_64.tar.gz -C $HOME  
> cd $HOME  
> ln -s wxparaver-4.5.6-linux-x86_64 paraver
```

- Download Paraver tutorials and uncompress into the Paraver directory

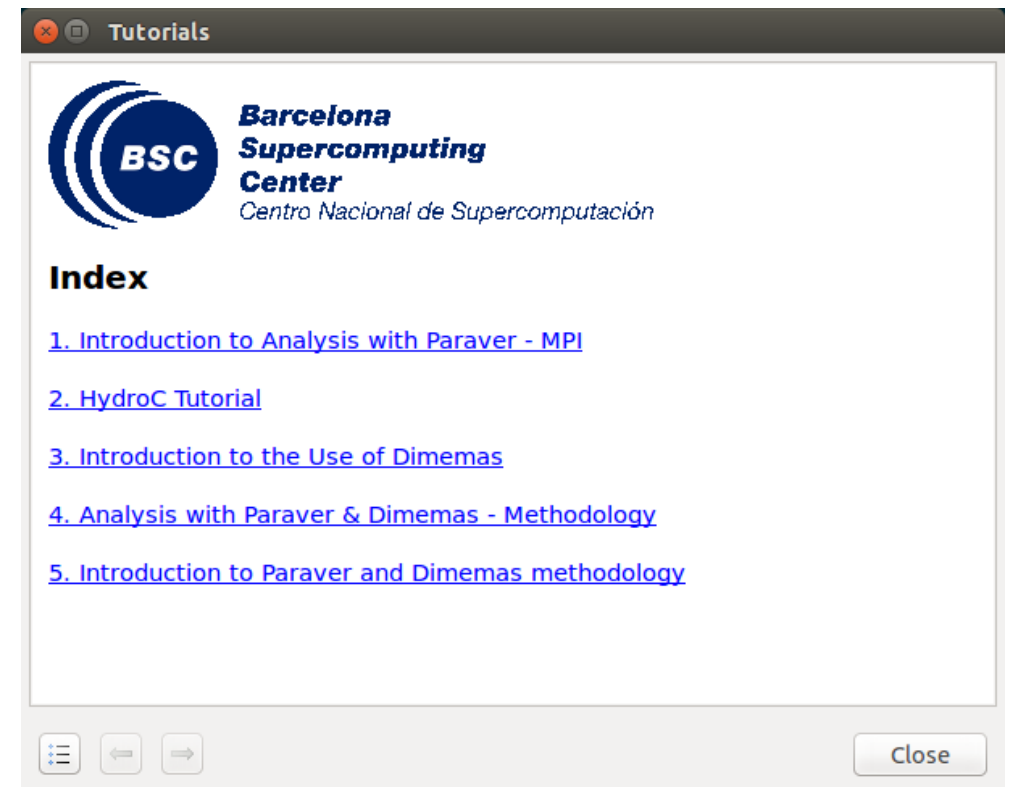
```
> scp <uid>@pi.ircpi.kobe-u.ac.jp:/home/S11505/shared/tools/BSC/paraver_pkgs/paraver-tutorials-2015026.tar.gz .  
> tar xvfz paraver-tutorials-20150526.tar.gz -C $HOME/paraver
```

# Launch Paraver

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



Click on Help → Tutorials





# Working with paraver locally

---

- Submit your job

@pi.ircpi.kobe-u.ac.jp

```
> cd $HOME/ntchem-mini.taxol/run  
> pjsub PIconputer_extrae.sh
```

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

**@your laptop**

```
> scp <uid>@pi.ircpi.kobe-u.ac.jp:$HOME/ntchem-mini/run_h2o/t_ntchem* $HOME
```

- Load the trace with Paraver

**@your laptop**

```
> $HOME/paraver/bin/wxparaver $HOME/t_ntchem*prv*
```