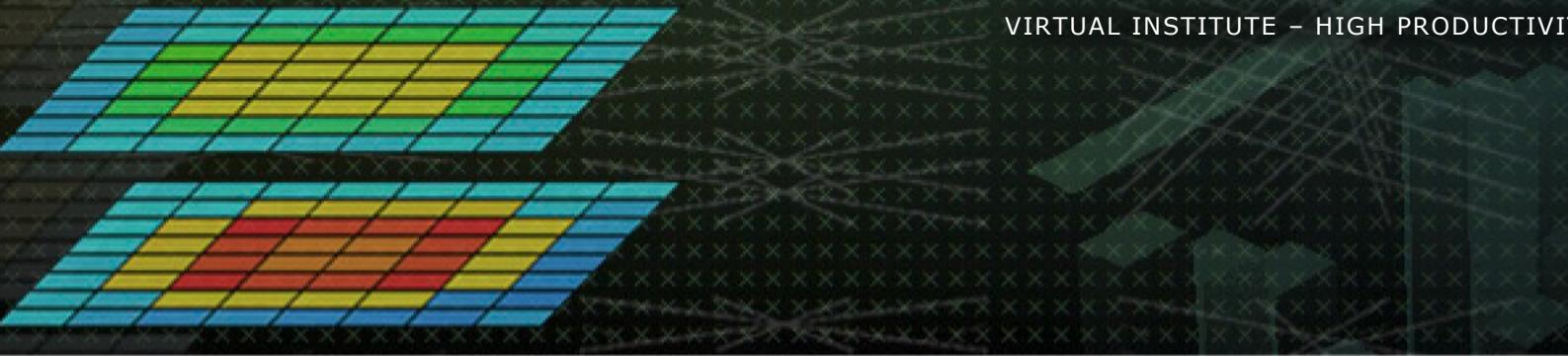


BSC Tools Hands-On

Lau Mercadal, Judit Giménez
(tools@bsc.es)
Barcelona Supercomputing Center



Getting a trace with Extrae

Extrae Features

- Platforms
 - Intel, Cray, BlueGene, Intel MIC, ARM, Android, Fujitsu Sparc, ...
- Parallel programming model
 - MPI, OpenMP, pthreads, OmpSs, CUDA, OpenCL, Java, Python, ...
- Performance counters
 - Using PAPI interface
- Link to source code
 - Callstack at MPI routines
 - OpenMP outlined routines
 - Selected user functions
- Periodic samples
- User events (Extrae API)



No need to
recompile
or relink!

Extrae Overheads

	Average values	Isambard XC50
Event	150 – 200ns	238ns
Event + PAPI	750 – 1000ns	1705ns
Event + callstack (1 level)	1μs	1854ns
Event + callstack (6 levels)	2μs	5700ns

How does Extrae work?

- Symbol substitution through LD_PRELOAD
 - Specific libraries for each combination of runtimes
 - MPI, OpenMP, MPI+OpenMP, ...
- Dynamic instrumentation
 - Based on DynInst (developed by U.Wisconsin/U.Maryland)
 - Instrumentation in memory
 - Binary rewriting
- Static link (i.e., PMPI, Extrae API)



Using Extrae in 3 steps

1. Adapt your job submission script

- Append Extrae loader script

2. Configure what to trace

- Modify extrae.xml

3. Run it!

- For further reference check the [Extrae User Guide](#)
 - <https://tools.bsc.es> -> Documentation -> Tools manuals
 - Also distributed with Extrae in \$EXTRAE_HOME/share/doc

Using Extrae in Isambard-XCI

```
laptop> ssh isambard-xci
xcil00> cp -r /home/ri-emercada/tools-material $HOME
xcil00> ls $HOME/tools-material
apps/
clustering/
extrae/
slides/
traces/
```

Step 1: Append loader script

```
xcil00> vi $HOME/tools-material/exrae/job.pbs
```

```
#!/usr/bin/env bash

#PBS -N "lulesh2.0_27p"
#PBS -l select=1
#PBS -l walltime=00:10:00
#PBS -q arm

cd $PBS_O_WORKDIR

export OMP_NUM_THREADS=1

aprun -n 27 ../apps/lulesh2.0 -i 10 -s 65 -p
```

Request
resources

Run the program

Step 1: Append loader script

```
xcil00> vi $HOME/tools-material/exrae/job.pbs
```

```
#!/usr/bin/env bash

#PBS -N "lulesh2.0_27p"
#PBS -l select=1
#PBS -l walltime=00:10:00
#PBS -q arm

cd $PBS_O_WORKDIR

export OMP_NUM_THREADS=1
export TRACE_NAME=lulesh2.0_27p.prv

aprun -n 27 ./trace.sh ../apps/lulesh2.0 -i 10 -s 65
```

Name of the resulting trace

Activate Extrae during the run

Step 1: Append loader script

```
xcil00> vi $HOME/tools-material/exrae/trace.sh
```

```
#!/usr/bin/env bash

#PBS -N "lulesh2.0_27p"
#PBS -l select=1
#PBS -l walltime=00:10:00
#PBS -q arm

cd $PBS_O_WORKDIR

export OMP_NUM_THREADS=1
export TRACE_NAME=lulesh2.0_27p.prv

aprun -n 27 ./trace.sh ./apps/lulesh2.0 -i 10 -s 65
```

```
#!/usr/bin/env bash

# Configure Exrae
export EXRAE_HOME=.../exrae/3.6.1/cray-mpich_7.7.6
export EXRAE_CONFIG_FILE=./exrae.xml

# Load the tracing library
export LD_PRELOAD=${EXRAE_HOME}/lib/libmpitrace.so

# Run the program
$*
```

What to trace

Type of application

Step 1: Append loader script

- Choose depending on 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 Append "f" suffix for Fortran codes

Step 3: Run it!

- Submit your job & check status

```
xcil00> cd $HOME/tools-material/exrae  
xcil00> qsub job.pbs  
xcil00> qstat -u $USER
```

- Once finished the trace will be in the same folder
 - lulesh2.0_27p.{pcf,prv,row}
- Any issue?
 - Traces already generated in \$HOME/tools-material/traces

Step 2: Configure what to trace

```
xcil00> vi $HOME/tools-material/extrاء/extrاء.xml
```

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

**Trace MPI calls
(What is the program doing?)**

```
<openmp enabled="no">
  <clocks enabled="no" />
  <counters enabled="yes" />
</openmp>
```

**Trace the call-stack
(Where in the code?)**

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

Compile with debug (-g)

Step 2: Configure what to trace

```
xcil00> vi $HOME/tools-material/extrae/extrae.xml
```

```
<counters enabled="yes">
  <cpu enabled="yes" starting-set-distribution="1">
    <set enabled="yes" domain="all" changeat-time="500000us">
      PAPI_TOT_INS,PAPI_TOT_CYC,PAPI_L1_DCM,PAPI_L2_DCM,PAPI_BR_INS,PAPI_SR_INS,PAPI_LD_INS
    </set>
    <set enabled="yes" domain="all" changeat-time="500000us">
      PAPI_TOT_INS,PAPI_TOT_CYC, PAPI_FP_INS
    </set>
  </cpu>
  <network enabled="no" />
  <resource-usage enabled="no" />
  <memory-usage enabled="no" />
</counters>
```

Select which HW counters
are measured
(How's the machine doing?)

Step 2: Configure what to trace

```
xcil00> vi $HOME/tools-material/extrاء/extrاء.xml
```

```
<buffer enabled="yes">
  <size enabled="yes">5000000</size> ←
  <circular enabled="no" />
</buffer>
```

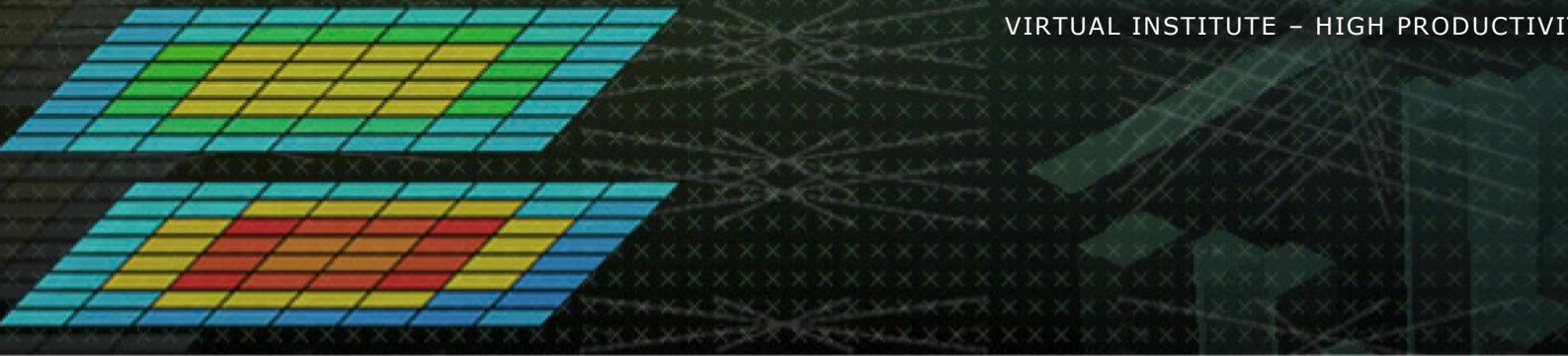
**Trace buffer size
(Flush / memory trade-off)**

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

**Enable periodic sampling
(More details)**

```
<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>
```

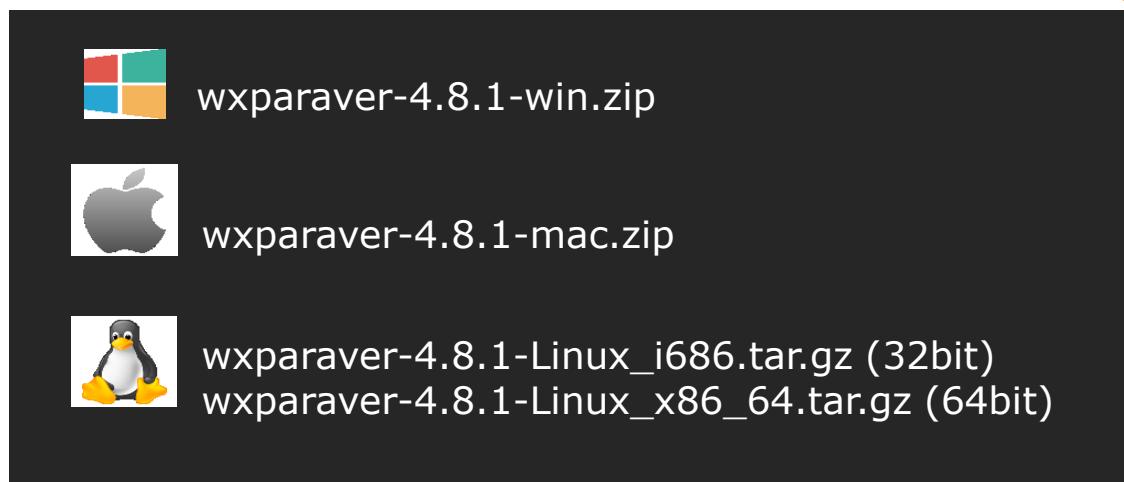
**Automatic post-processing
to generate
the Paraver trace**



Installing Paraver & First analysis steps

Install Paraver in your laptop

- Download a binary for your OS
 - <https://tools.bsc.es/downloads>

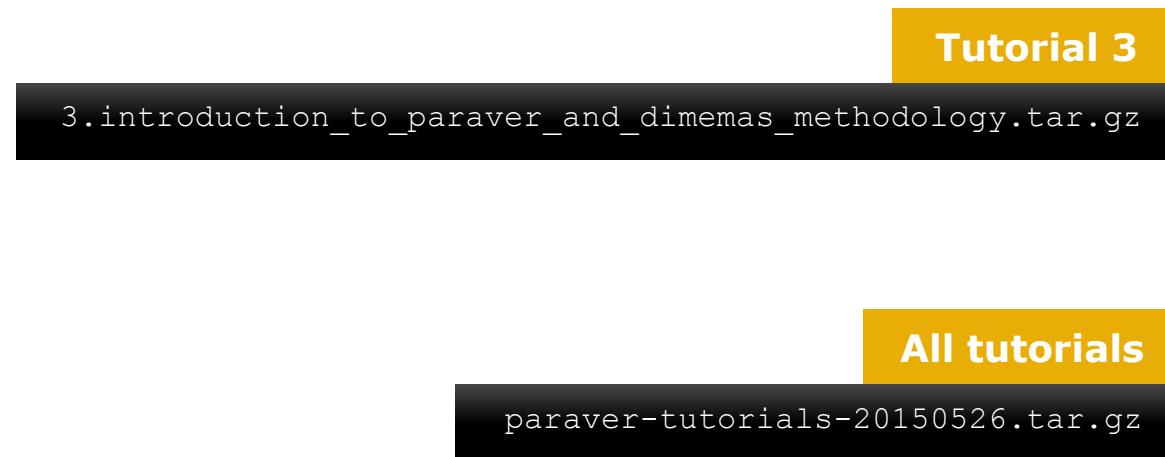


- Also available in Isambard-XCI
 - /home/ri-emercada/tools-packages

The screenshot shows the BSC Tools Downloads page. The Paraver section is highlighted with a yellow box and arrow. It includes links for "Get PARAVER" (Windows, Mac, Linux, 32bit, 64bit) and "Get CLUSTERING", "Get SPECTRAL", "Get TRACKING", "Get BASIC ANALYSIS", and "Get FOLDING".

Install Paraver tutorials

- Download tutorials archive
 - <https://tools.bsc.es/paraver-tutorials>



The screenshot shows a web browser displaying the Paraver documentation page. The URL in the address bar is "news@tools:~ > Paraver 4.7.2 available". The page title is "Home » Documentation » Paraver tutorials". A text block explains that seven tutorials can be opened with wxParaver versions newer than 4.3.0. A list of tutorials is provided, with the third item, "Introduction to Paraver and Dimemas methodology", highlighted with a yellow box. At the bottom, there is a note about downloading all tutorials together in ".tar.gz" or ".zip" format.

Tutorial	Description
Paraver introduction (MPI)	Start here to familiarize with Paraver basic commands and the first steps of a performance analysis.
Dimemas introduction	The basic steps to learn how to configure and run the Dimemas simulator and to start looking at the results.
Introduction to Paraver and Dimemas methodology	This tutorial presents different ways to analyze a MPI application through well-known rules, their diagnosis and how they impact on your exploration (no traces included).
Methodology	This tutorial shows some examples of the analysis that can be done using the provided configuration files.
Tutorial on HydroC analysis (MPI, Dimemas, CUDA)	One example of performance analysis of the MPI application Hydro and further simulations with Dimemas.
Trace preparation	Look at this tutorial to select a representative region for a large trace that cannot be loaded into memory.
Trace alignment tutorial.	If you identify some unexpected unalignment or backwards communications, use this tutorial to learn how to correct shifts between processors.

- Also available in Isambard-XCI
 - `/home/ri-emercada/tools-packages`

Uncompress, rename & move

- Paraver

```
laptop> tar xf wxparaver-4.8.1-Linux_x86_64.tar.bz2  
laptop> mv wxparaver-4.8.1-Linux_x86_64 paraver
```

- Tutorials

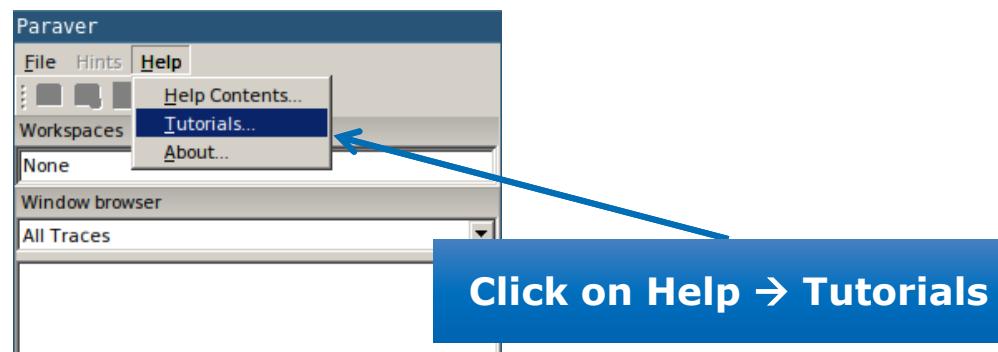
```
laptop> tar xf paraver-tutorials-20150526.tar.bz2  
laptop> mv paraver-tutorials-20150526 paraver/tutorials
```

Check that everything works

- Start Paraver

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

- Check the tutorials are available



A screenshot of a web browser displaying the 'Tutorials' page of the Barcelona Supercomputing Center (BSC) website. The page features the BSC logo and the text 'Barcelona Supercomputing Center' and 'Centro Nacional de Supercomputación'. Below this is a section titled 'Index' containing a numbered list of six tutorials:

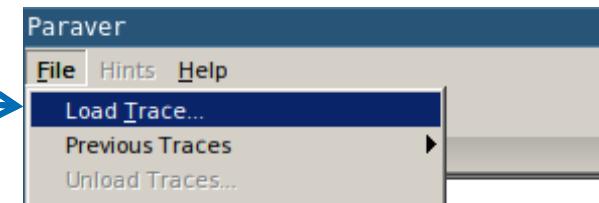
- [1. Introduction to Analysis with Paraver - MPI](#)
- [2. Introduction to the Use of Dimemas](#)
- [3. Introduction to Paraver and Dimemas methodology](#)
- [4. Analysis with Paraver & Dimemas - Methodology](#)
- [5. HydroC Tutorial](#)
- [6. Paraver trace preparation](#)

First steps of the analysis

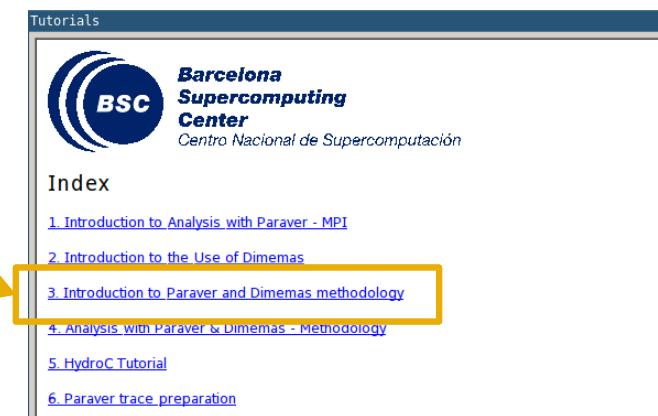
- Copy the trace to your laptop

```
laptop> scp isambard-xci:$HOME/tools-material/exrae/lulesh2.0_27p.{prv,pcf,row} ./
```

- Load the trace
 - File -> Load trace -> Browse to .prv file

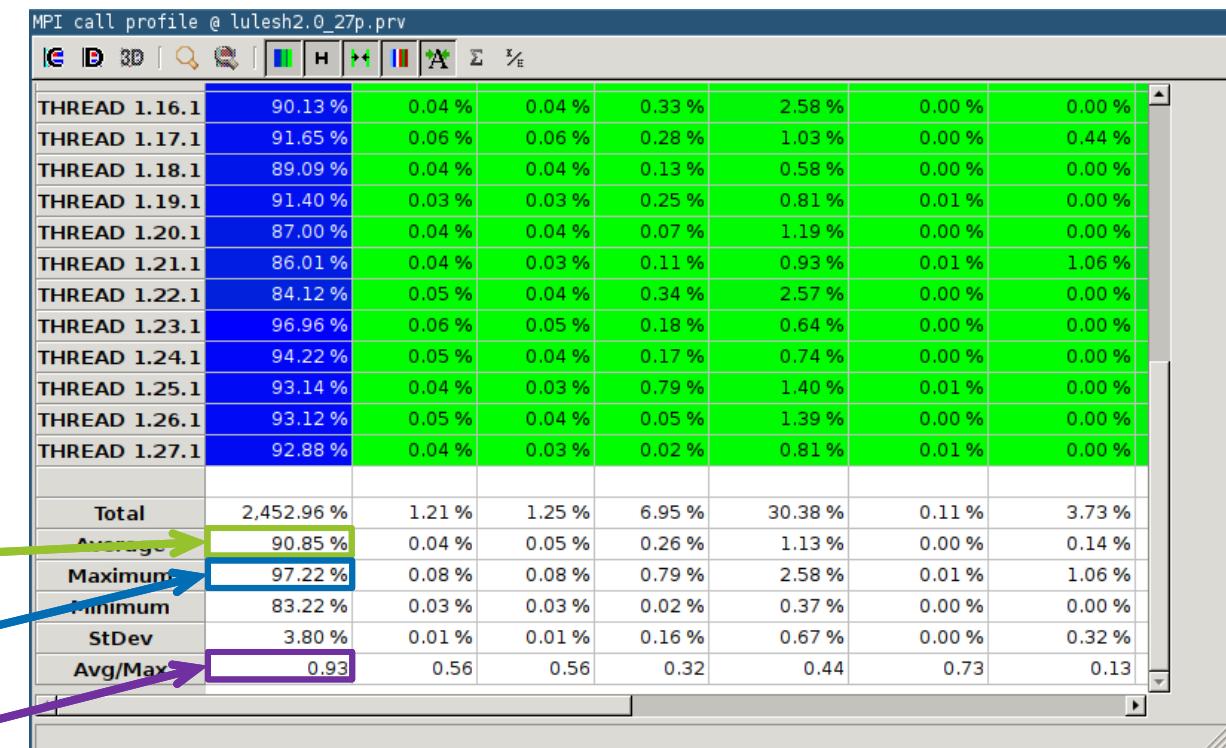
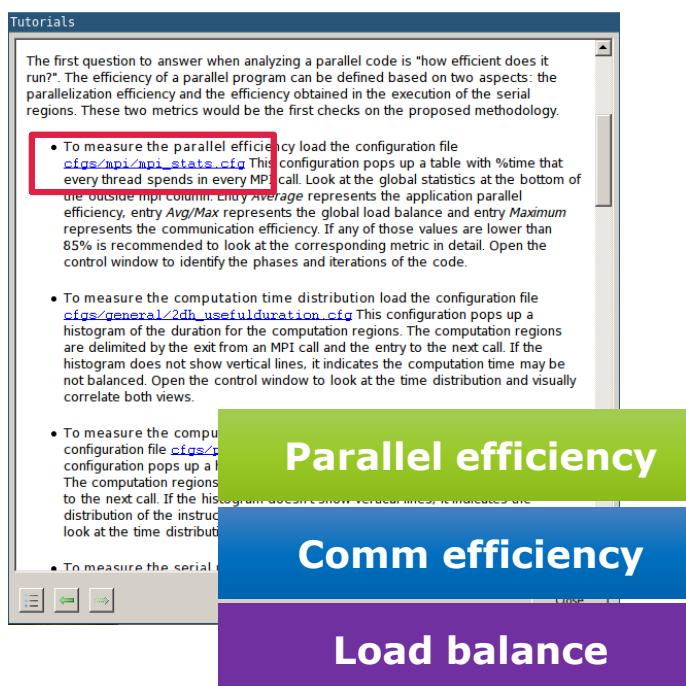


- Follow tutorial #3
 - Help -> Tutorials



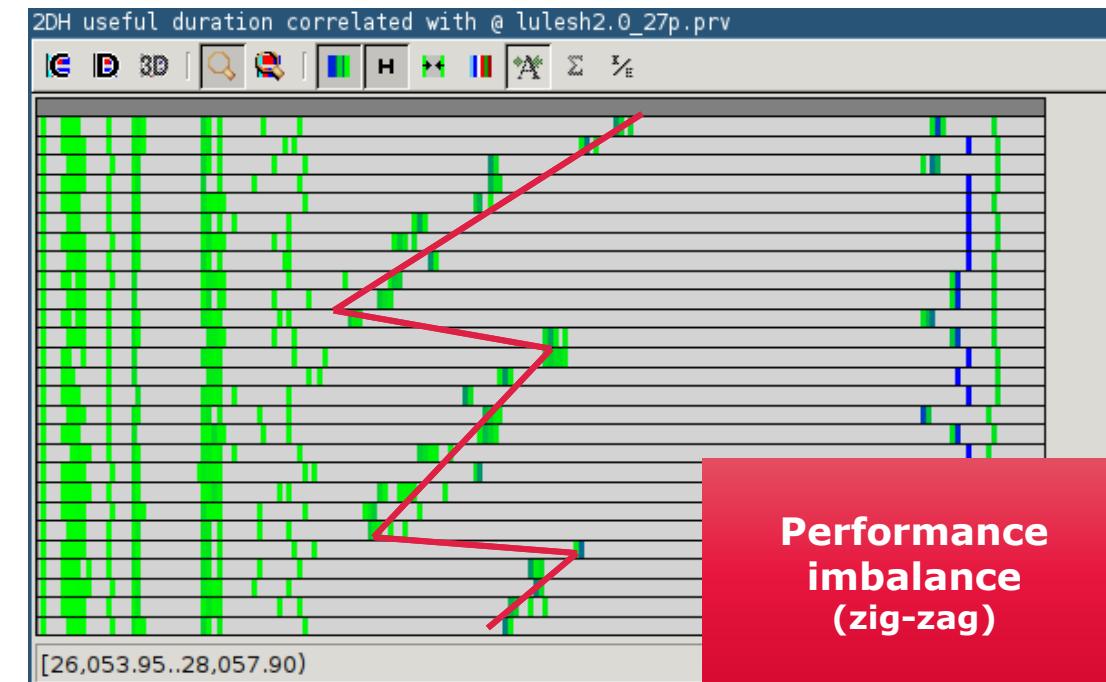
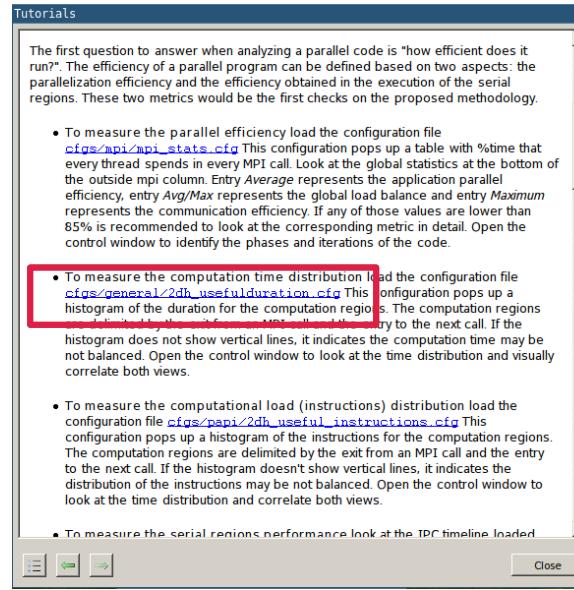
Measure the parallel efficiency

- Click on `mpi_stats.cfg`



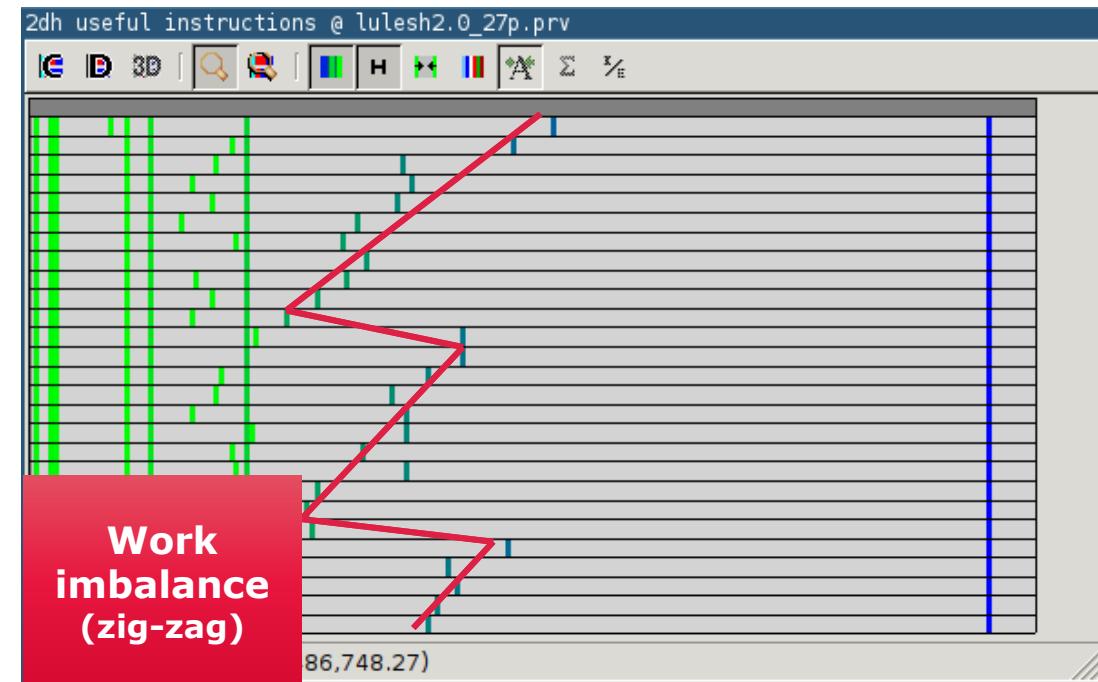
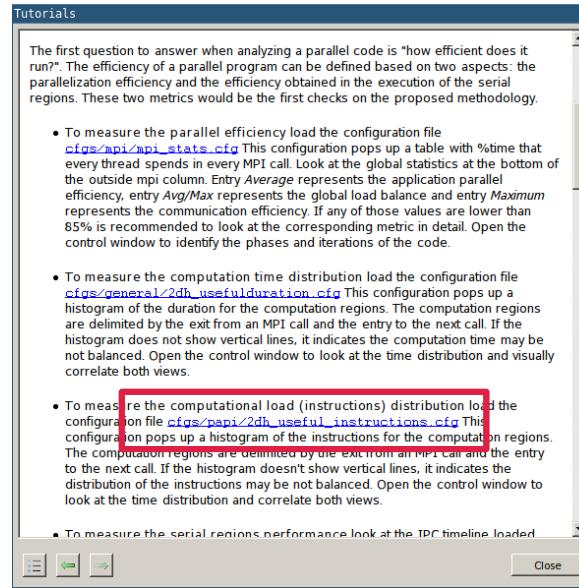
Computation time and work distribution

- Click on `2dh_usefulduration.cfg` (2nd link) → Shows **time computing**



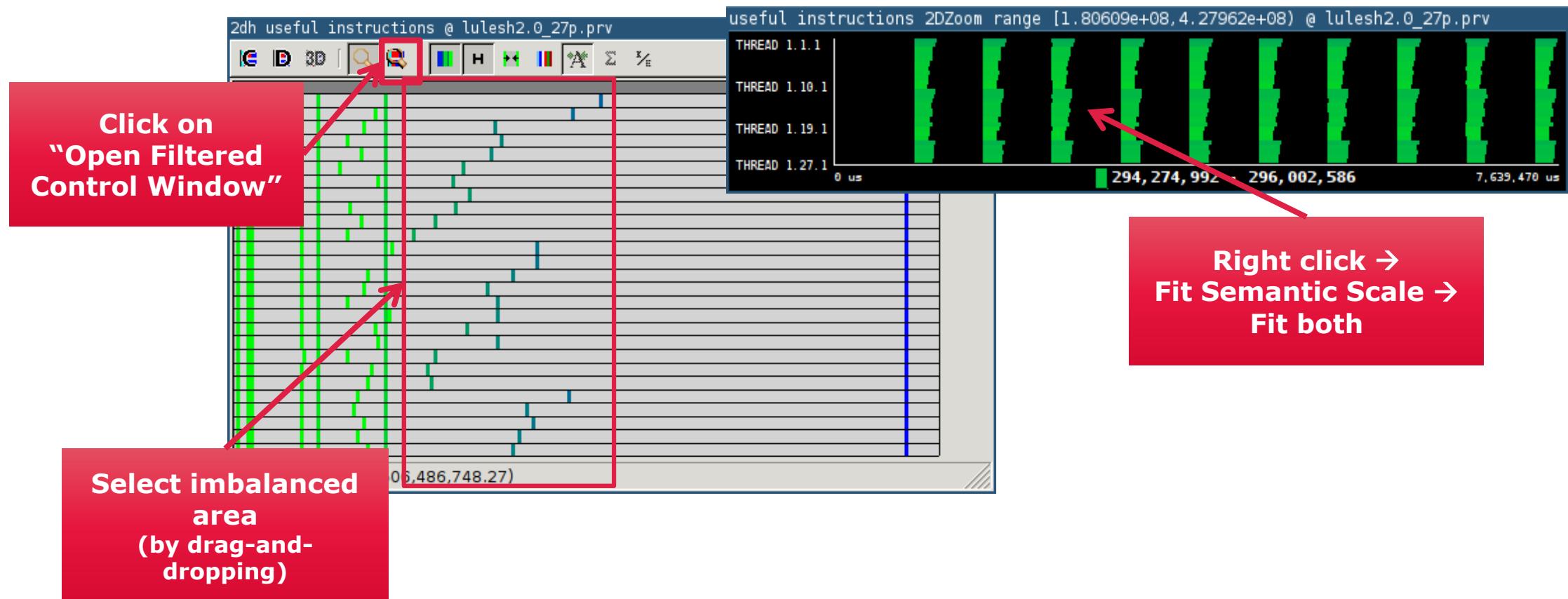
Computation time and work distribution

- ... and `2dh_useful_instructions.cfg` (3rd link) → Shows **amount of work**



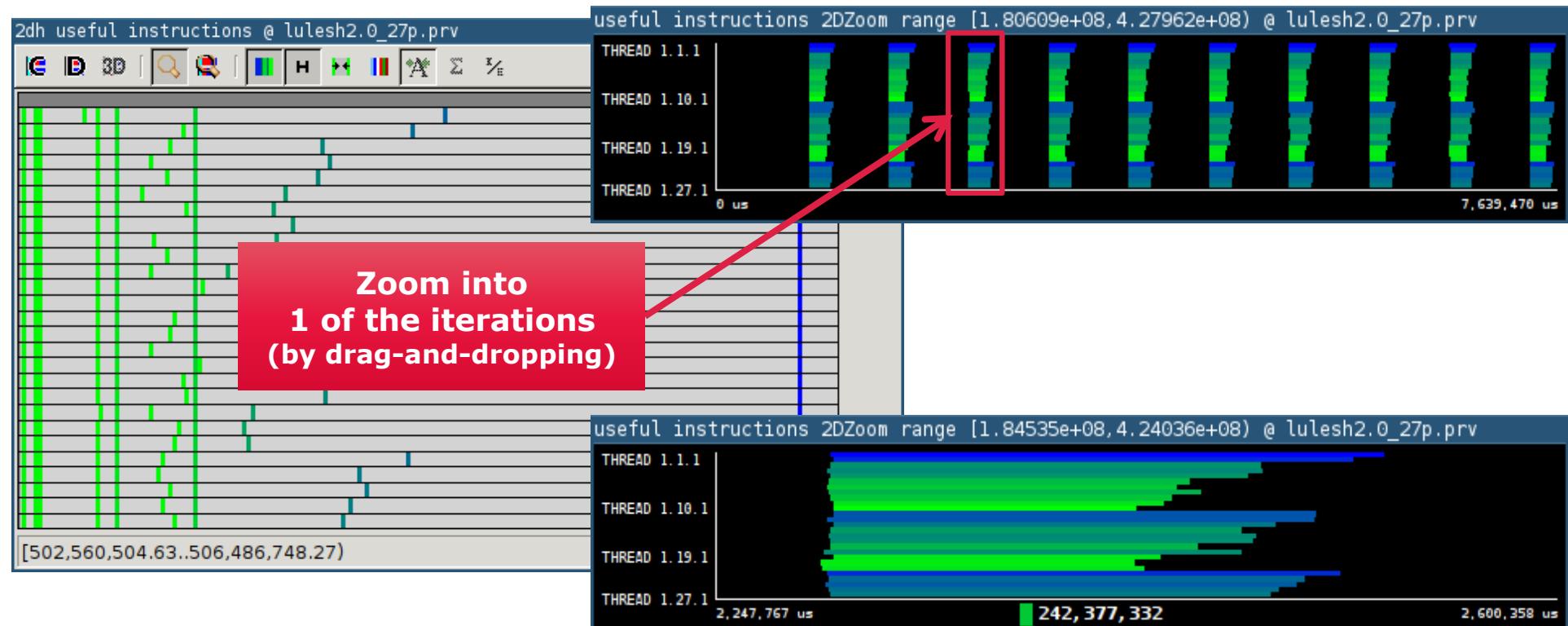
Where does this happen?

- Go from tables to timelines



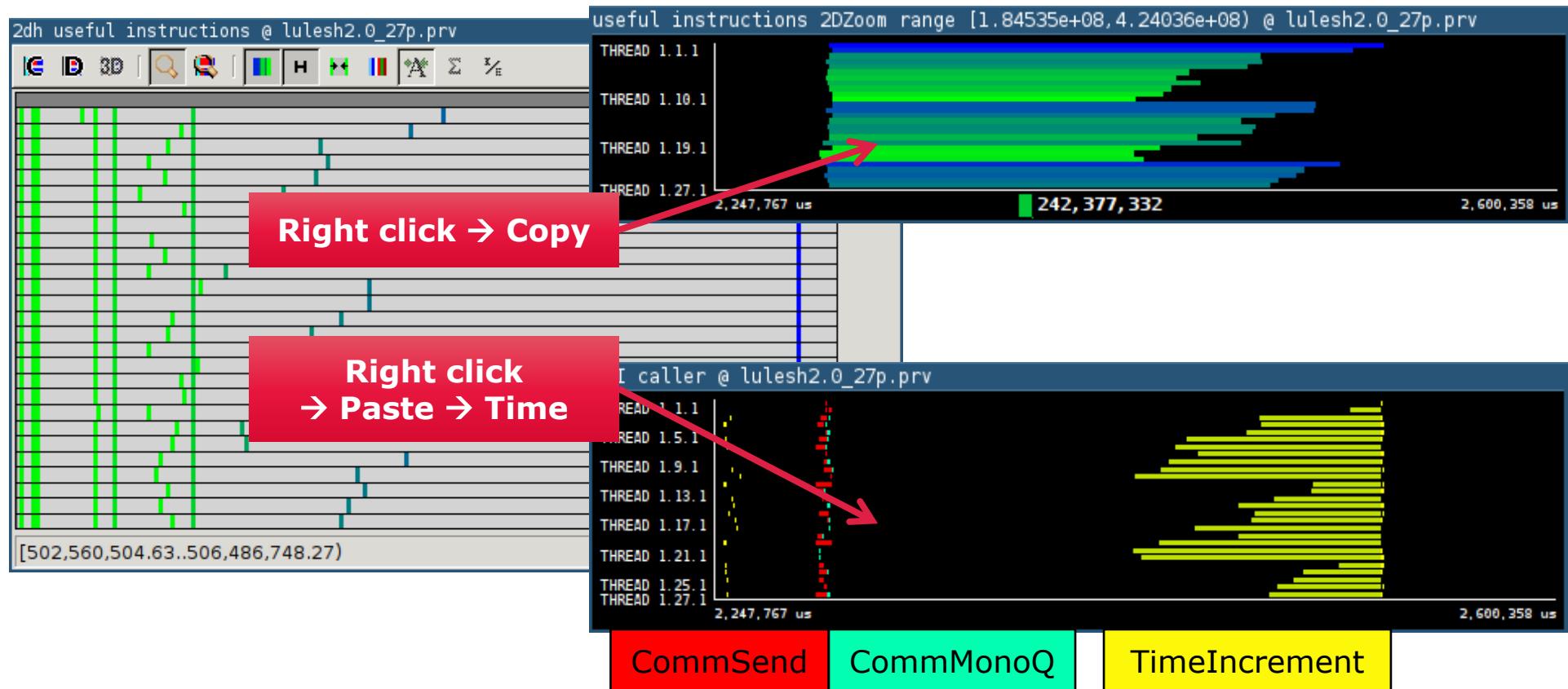
Where does this happen

- **Slow** & **Fast** at the same time → **Imbalance**

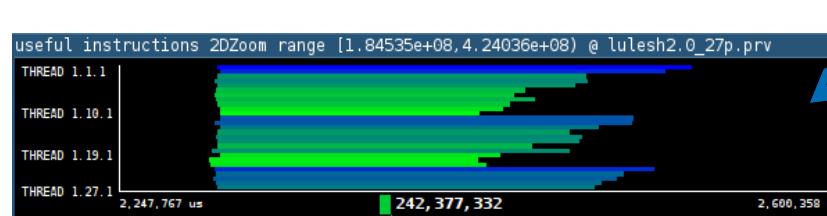


Where does this happen

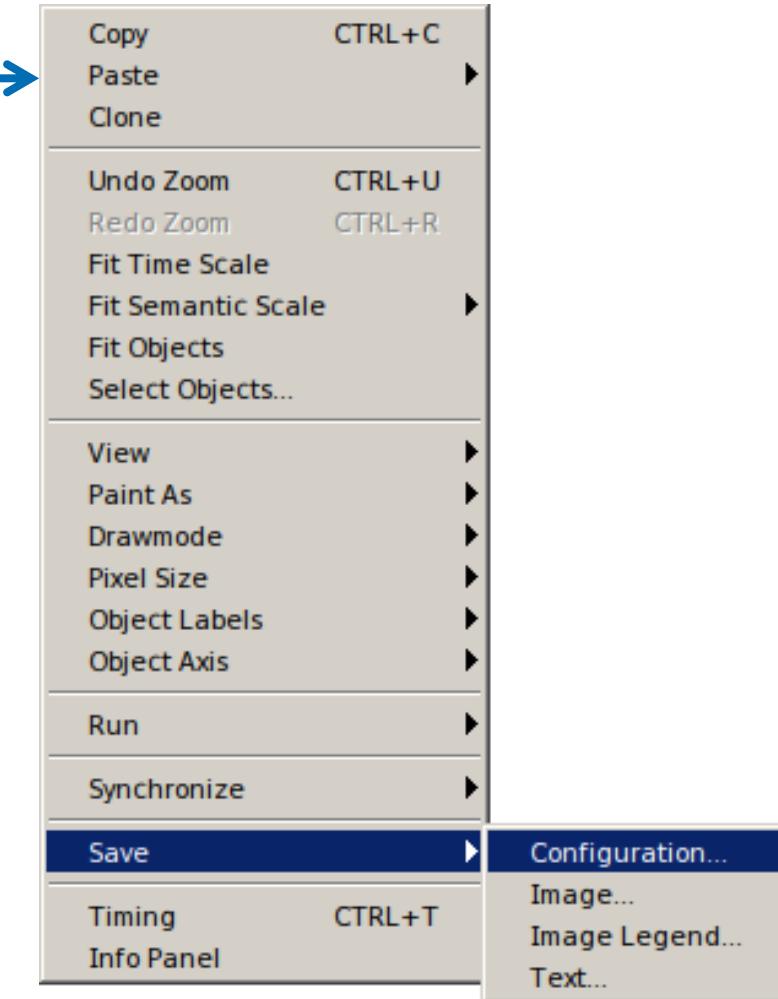
- Hints → Callers → Caller function



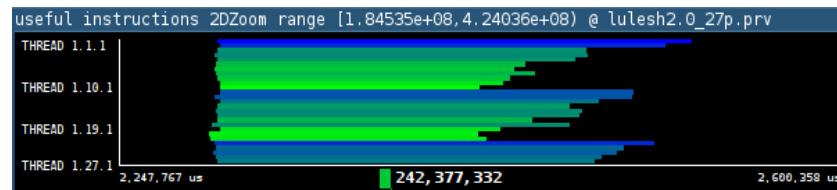
Save CFG's (2 methods)



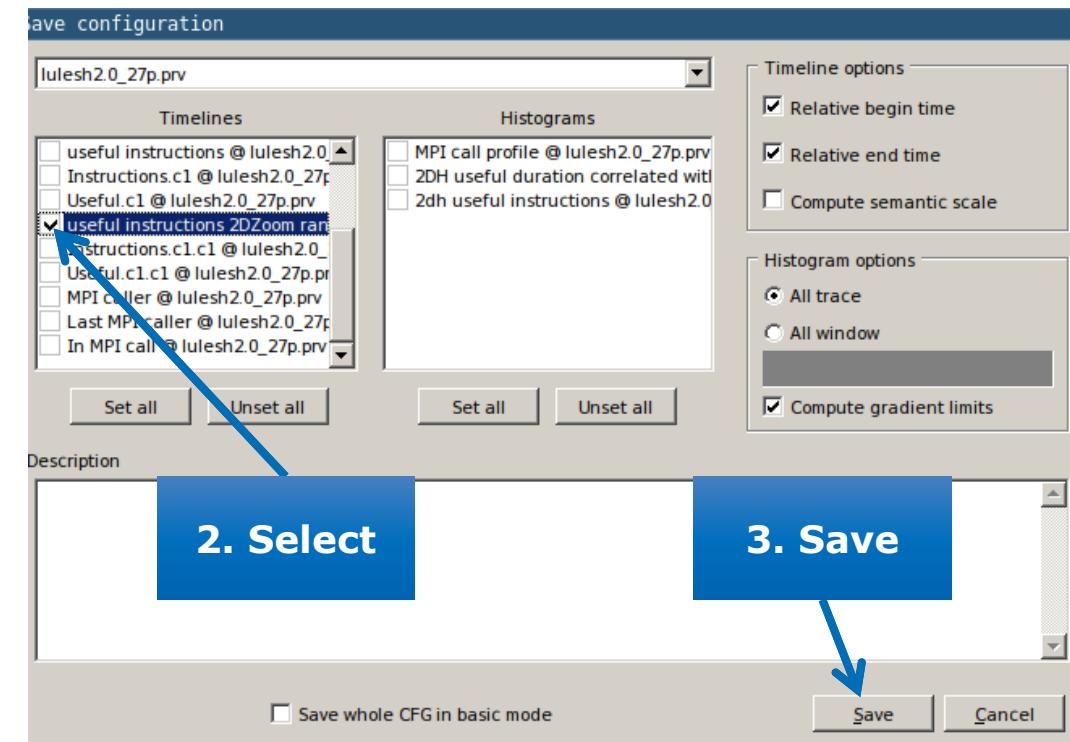
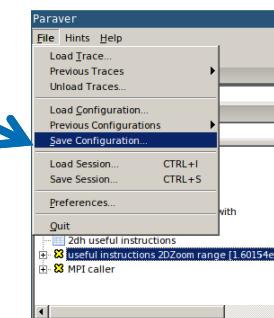
Right click
on
timeline



Save CFG's (2 methods)

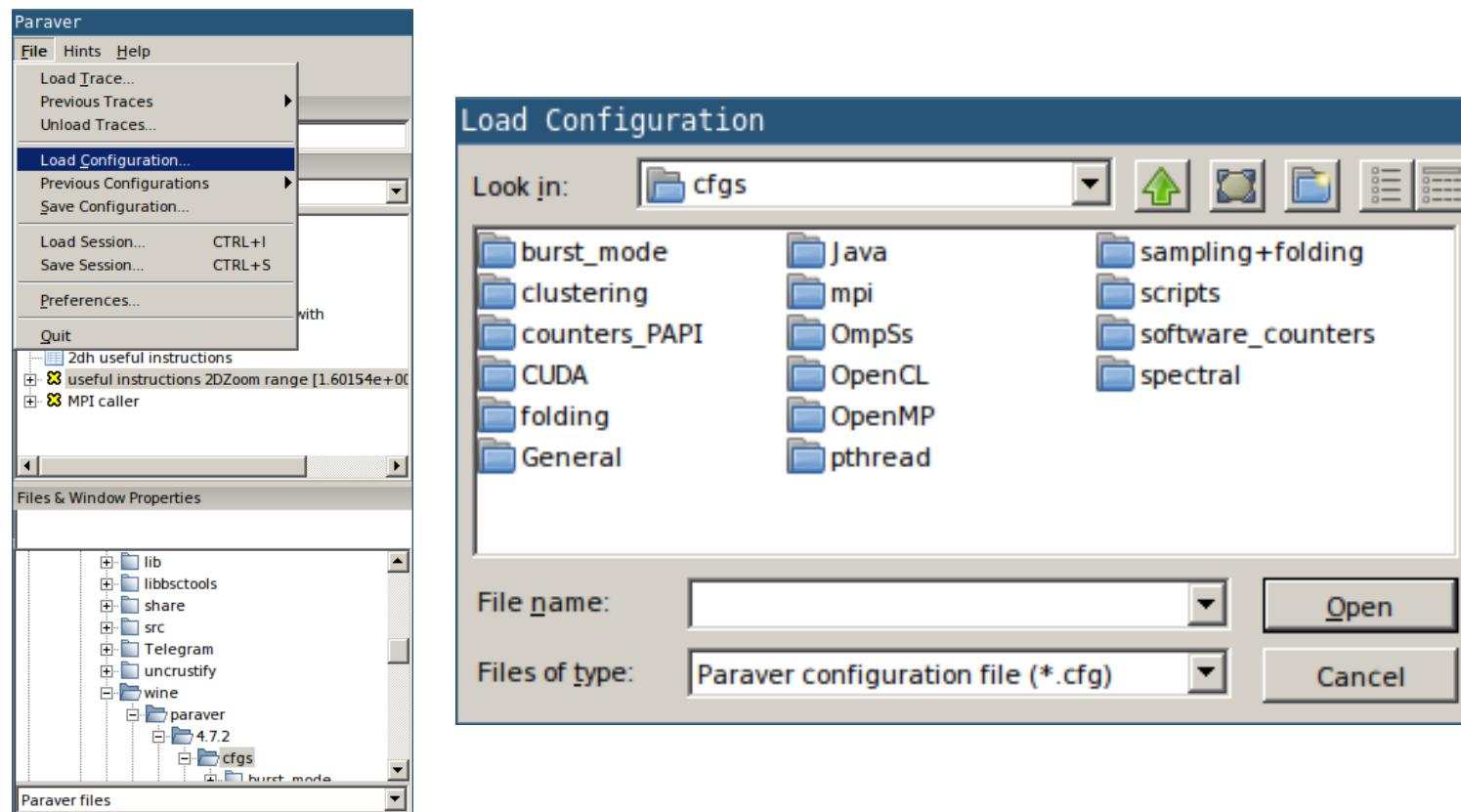


1. Main
Paraver
window



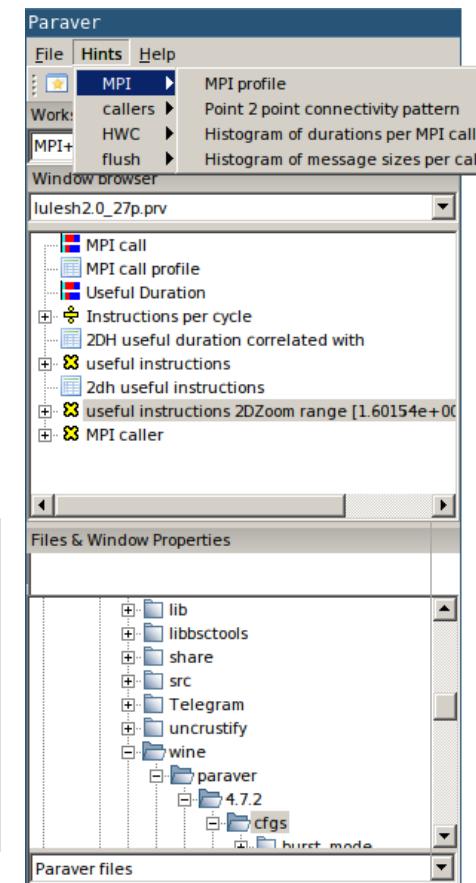
CFG's distribution

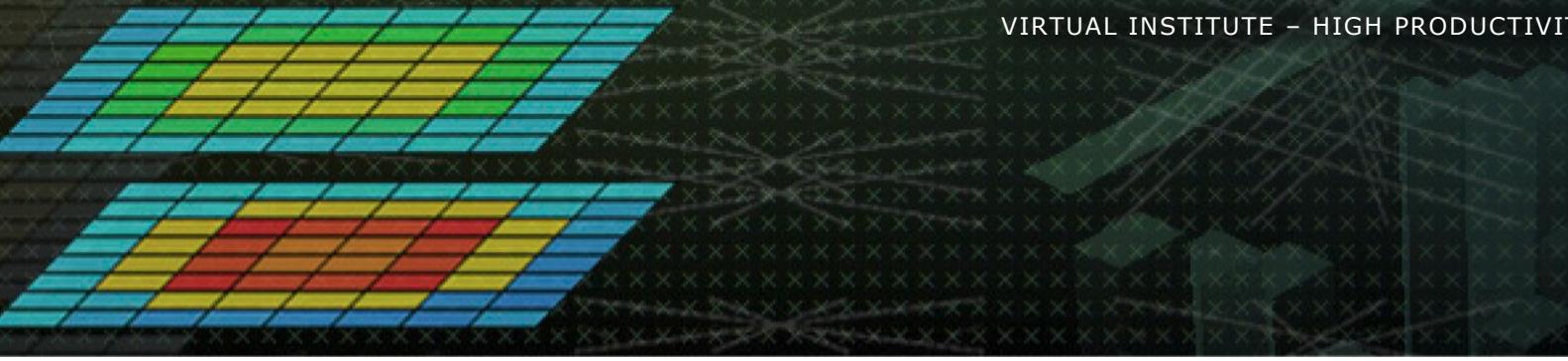
- Paraver comes with many more included CFG's



Hints: a good place to start!

- Paraver suggests CFG's based on the information present in the trace





Cluster-based analysis

Install Clustering in your laptop

- Download a binary for your OS
 - <https://tools.bsc.es/downloads>

```
laptop> tar xf clusteringsuite-2.6.8-Linux_x86_64.tar.bz2  
laptop> mv clusteringsuite-2.6.8-Linux_x86_64 clustering
```

- Also available in Isambard-XCI
 - /home/ri-emercada/tools-packages

Use clustering analysis

- Run clustering

```
laptop> cd $HOME/tools-material/clustering  
laptop> $HOME/clustering/bin/BurstClustering \  
      -d cluster.xml \  
      -i ../extrae/lulesh2.0_27p.prv \  
      -o lulesh2.0_27p_clustered.prv
```

- If you didn't get your own trace, use a prepared one from:

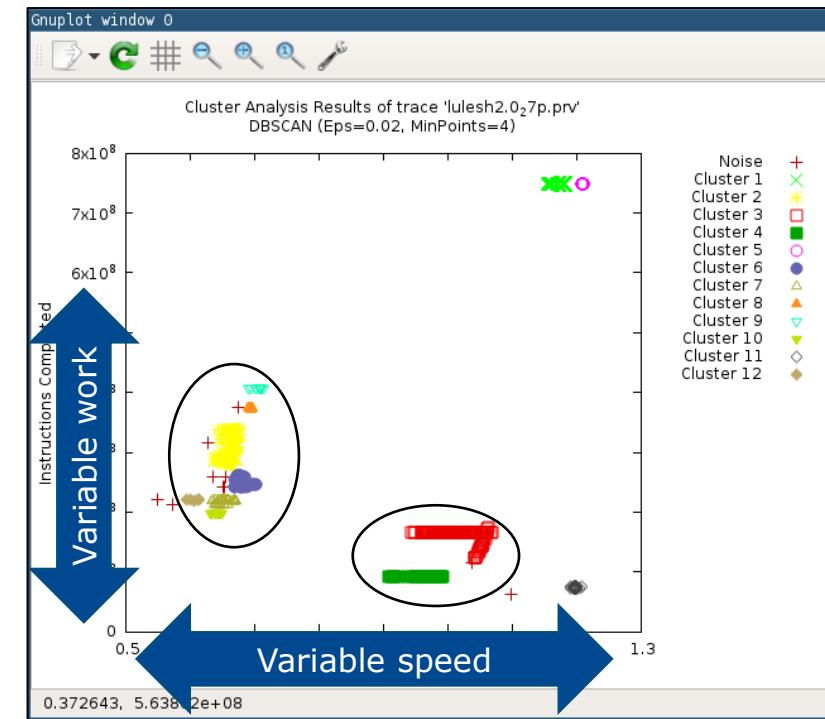
```
xcil00> ls $HOME/tools-material/traces/lulesh2.0_27p.prv
```

Cluster-based analysis

- Check the resulting scatter plot

```
laptop> gnuplot lulesh2.0_27p_clustered.IPC.PAPI_TOT_INS.gnuplot
```

- Identify main computing trends
- Work (Y) vs. Speed (X)
- Look at the clusters shape
 - Variability in both axes indicate **potential imbalances**



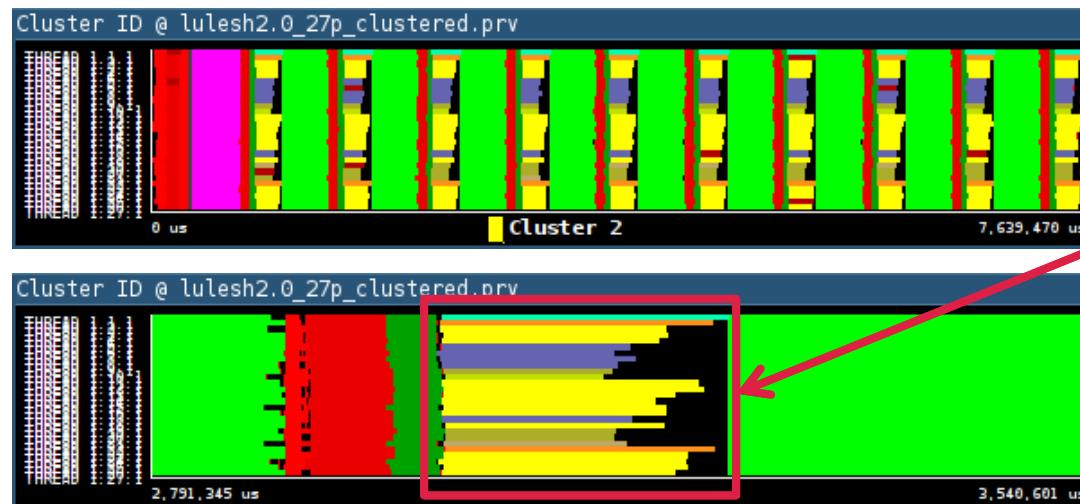
Correlating scatter plot and time distribution

- Open the clustered trace with Paraver and look at it

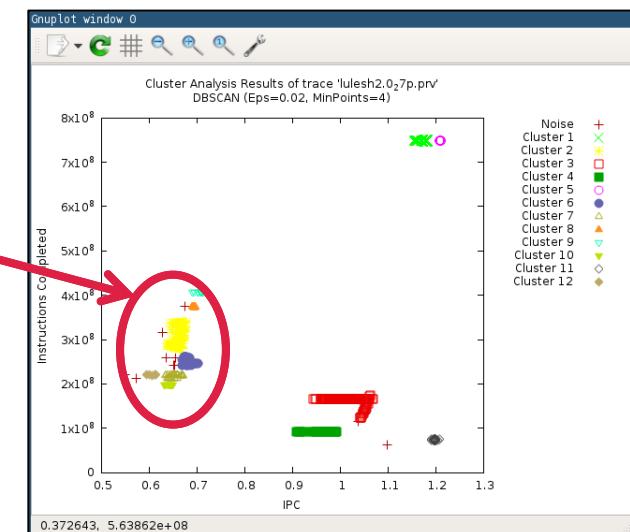
```
laptop> $HOME/paraver/bin/wxparaver <path-to>/lulesh_27p_clustered.prv
```

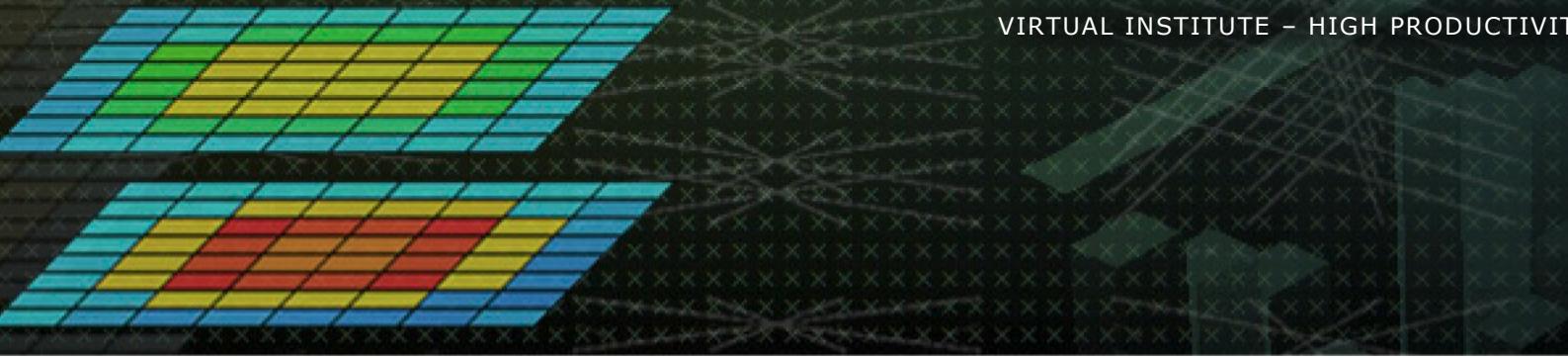
- Display the distribution of clusters over time

- File → Load configuration → \$HOME/paraver/cfgs/clustering/clusterID_window.cfg



Variable work / speed + Simultaneously @ different processes = Imbalances





BSC Tools Hands-On

Lau Mercadal, Judit Giménez
(tools@bsc.es)
Barcelona Supercomputing Center
