Performance Tools Hands-On

PAT – Oct/2016

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

tools@bsc.es

Extrae features

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

No need to recompile / relink!

Extrae overheads

	Average values	Mistral
Event	150 – 200 ns	167 ns
Event + PAPI	750 ns – 1 us	4.7 us
Event + callstack (1 level)	600 ns	626 ns
Event + callstack (6 levels)	1.9 us	2 us

How does Extrae work?

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

- Dynamic instrumentation
  - Based on Dyninst (developed by U.Wisconsin/U.Maryland)
    - Instrumentation in memory
    - Binary rewriting

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

Using Extrae in 3 steps

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 Mistral

@ your laptop

```
> ssh -Y <USER>@mistral.dkrz.de
```

- The following directory in your home folder contains all the examples:

@ mistral.dkrz.de

```
> cp ~k203109/tools-material $HOME

> ls -l $HOME/tools-material
   … clustering/
   … extrae/
   … paraver/
   … slides/
   … traces/
```

Here you have a copy of this slides.
You can copy them to your laptop
or open remotely with:
> evince slides/Tools-Hands-On.pdf

Installing ICON

```
> cd $HOME
> tar xfzv icon-dev.tgz
> cd icon-dev
> module add intel/15.0.6
> module add mxm/3.3.3002
> module add fca/2.5.2393
> module add bullxmpi_mlx/bullxmpi_mlx-1.2.8.3
> ./configure --with-fortran=intel
> ./build_command
> ./make_runscripts atm_amip
```

Step 1: Adapt the job script to load Extrae (LD_PRELOAD)

@ mistral.dkrz.de

```
> cd $HOME/icon-dev/run
> cp exp.atm_amip.run exp.atm_amip.run.extrae
> vi exp.atm_amip.run.extrae
```

exp.atm_amip.run.extrae

```
#SBATCH --acount =kg0166
#SBATCH --job-name=exp.atm_amip.run
#SBATCH --workdir=/home/dkrz/k203109/icon-dev/run
#SBATCH --nodes=4
#SBATCH --threads-per-core=2
#SBATCH --output=LOG.exp.atm_amip.run.%j.o
#SBATCH --error=LOG.exp.atm_amip.run.%j.e
#SBATCH --exclusive
#SBATCH --time=00:30:00

…
  end_date="1979-01-01T23:50:00Z"
…
${START} ${MODEL}

…
```

Line 147: Reduce the simulation time

Line 820 (srun app)

Step 1: Adapt the job script to load Extrae (LD_PRELOAD)

- Copy Extrae files to experiment directory

@ mistral.dkrz.de

```
> cp $HOME/tools-material/extrae/* $HOME/icon-dev/run
```

exp.atm_amip.run.extrae

```
#SBATCH --acount=kg0166
#SBATCH --job-name=exp.atm_amip.run
#SBATCH --workdir=/home/dkrz/k203109/icon-dev/run
#SBATCH --nodes=4
#SBATCH --threads-per-core=2
#SBATCH --output=LOG.exp.atm_amip.run.%j.o
#SBATCH --error=LOG.exp.atm_amip.run.%j.e
#SBATCH --exclusive
#SBATCH --time=00:30:00
…
TRACE=${basedir}/run/trace.sh
EXTRAE_CONFIG=${basedir}/run/extrae.xml
cp ${TRACE} .
cp ${EXTRAE_CONFIG} .
${START} ${TRACE} ${MODEL}
…
```

Step 1: Adapt the job script to load Extrae (LD_PRELOAD)

@ mistral.dkrz.de

```
> vi $HOME/icon-dev/run/trace.sh
```

exp.atm_amip.run.extrae

```
#SBATCH --acount =kg0166
#SBATCH --job-name=exp.atm_amip.run
#SBATCH --workdir=/home/dkrz/k203109/icon-dev/run
#SBATCH --nodes=4
#SBATCH --threads-per-core=2
#SBATCH --output=LOG.exp.atm_amip.run.%j.o
#SBATCH --error=LOG.exp.atm_amip.run.%j.e
#SBATCH --exclusive
#SBATCH --time=00:30:00
…
TRACE=${basedir}/run/trace.sh
EXTRAE_CONFIG=${basedir}/run/extrae.xml
cp ${TRACE} .
cp ${EXTRAE_CONFIG} .
${START} ${TRACE} ${MODEL}
…
```

trace.sh

```
#!/bin/bash

# Configure Extrae
export EXTRAE_HOME=/sw/…/extrae-3.4.1-bullxmpi-intel14
source ${EXTRAE_HOME}/etc/extrae.sh

export TRACE_NAME=icon.prv
export EXTRAE_CONFIG_FILE=extrae.xml

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

# Run the program
$*
```

Step 1: Which tracing library?

- 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

Step 3: Run it!

- Submit your job

@ mistral.dkrz.de

```
> cd $HOME/icon-dev/run

> sbatch exp.atm_amip.run.extrae
```

- Easy! ☺

Step 2: Extrae XML configuration

@ mistral.dkrz.de

```
> vi $HOME/icon-dev/run/extrae.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

Step 2: Extrae XML configuration (II)

@ mistral.dkrz.de

```
> vi $HOME/icon-dev/run/extrae.xml
```

```
<counters enabled="yes">
  <cpu enabled="yes" starting-set-distribution="cyclic">
    <set enabled="yes" domain="all" changeat-time="500000us">
      PAPI_TOT_INS, PAPI_TOT_CYC, PAPI_L1_DCM, PAPI_L3_TCM,
      PAPI_BR_INS, PAPI_BR_MSP
    </set>
    <set enabled="yes" domain="all" changeat-time="500000us">
      PAPI_TOT_INS, PAPI_TOT_CYC, PAPI_SR_INS, PAPI_LD_INS,
      RESOURCES_STALLS:RS, RESOURCE_STALLS:SB
    </set>
    <set enabled="yes" domain="all" changeat-time="500000us">
      PAPI_TOT_INS, PAPI_TOT_CYC, RESOURCE_STALLS:ROB, PAPI_L2_DCM
    </set>
    <set enabled="yes" domain="all" changeat-time="500000us">
      ...
    </set>
    <set enabled="yes" domain="all" changeat-time="500000us">
      ...
    </set>
  </cpu>
  <network enabled="no" />
  <resource-usage enabled="no" />
  <memory-usage enabled="no" />
</counters>
```

Select which HW counters are measured

Step 2: Extrae XML configuration (III)

@ mistral.dkrz.de

```
> vi $HOME/icon-dev/run/extrae.xml
```

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

<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"
>
  $TRACE_NAME$
</merge>
```

Trace buffer size

Enable sampling

Merge intermediate files into Paraver trace

All done! Check your resulting trace

- Once finished (check with “squeue”) you will have the trace (3 files):

@ mistral.dkrz.de

```
> cd $HOME/icon-dev/experiments/atm_amip
> ls -l
   ...
   icon.pcf
   icon.prv
   icon.row
```

- Compress the trace (takes a little while)

@ mistral.dkrz.de

```
> gzip icon.prv
```

- Any trouble? Traces already generated here:

@ mistral.dkrz.de

```
> ls $HOME/tools-material/traces
```

Installing Paraver

- Download the Paraver binaries to your laptop

@ your laptop

```
> scp <USER>@mistral.dkrz.de:tools-packages/<VERSION> $HOME
```

Pick your version

Linux 64 bits	`wxparaver-4.6.2-linux-x86_64.tar.gz`
Linux 32 bits	`wxparaver-4.6.2-linux-x86_32.tar.gz`
Mac	`wxparaver-4.6.2-mac.zip`
Windows	`wxparaver-4.6.2-win.zip`

Installing Paraver (II)

- Uncompress the package into your home directory

@ your laptop

```
> tar xvfz wxparaver-4.6.2-linux-x86_64.tar.gz -C $HOME

> ln -s $HOME/wxparaver-4.6.2-linux-x86_64 $HOME/paraver
```

- Download Paraver tutorials and uncompress into the Paraver directory

@ your laptop

```
> scp <USER>@mistral.dkrz.de:
    tools-packages/paraver-tutorials-20150526.tar.gz $HOME

> tar xvfz $HOME/paraver-tutorials-20150526.tar.gz -C $HOME/paraver
```

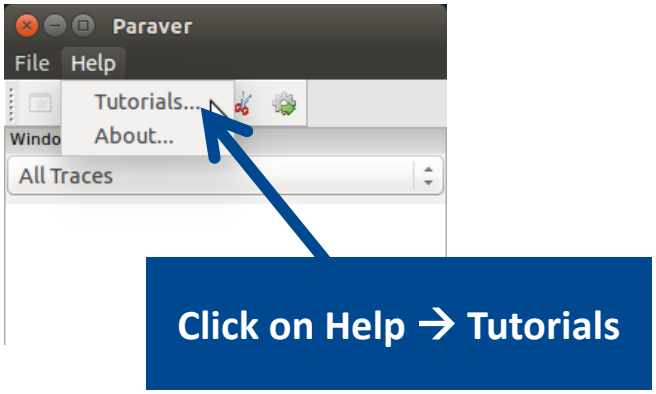
Check that everything works

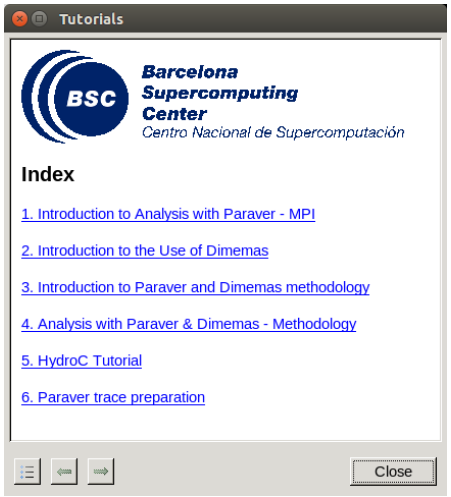
- Start Paraver

@ your laptop

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

- Check that tutorials are available





- Trouble installing locally? Remote open from Mistral

@ mistral.dkrz.de

```
> ssh -Y <USER>@mistral.dkrz.de
> cd /sw/rhel6-x64/analysis-tools/wxparaver-4.6.1/bin
> ./wxparaver
```

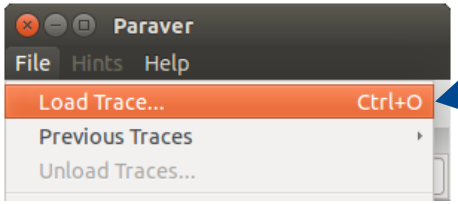
First steps of analysis

- Copy the trace to your laptop

@ your laptop

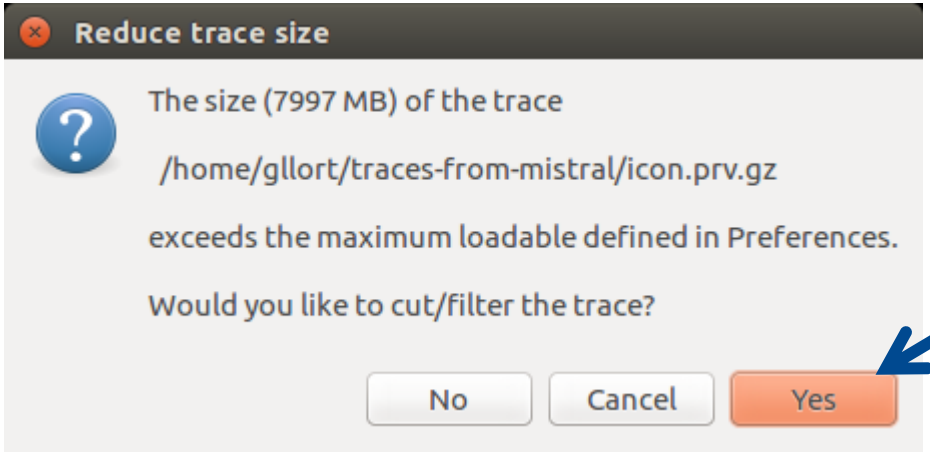
```
> scp <USER>@mistral.dkrz.de:icon-dev/experiments/atm_amip/icon.* $HOME
```

- Load the trace with Paraver



Click on File → Load Trace → Browse to "icon.prv.gz"

- Trace is big: Filter it

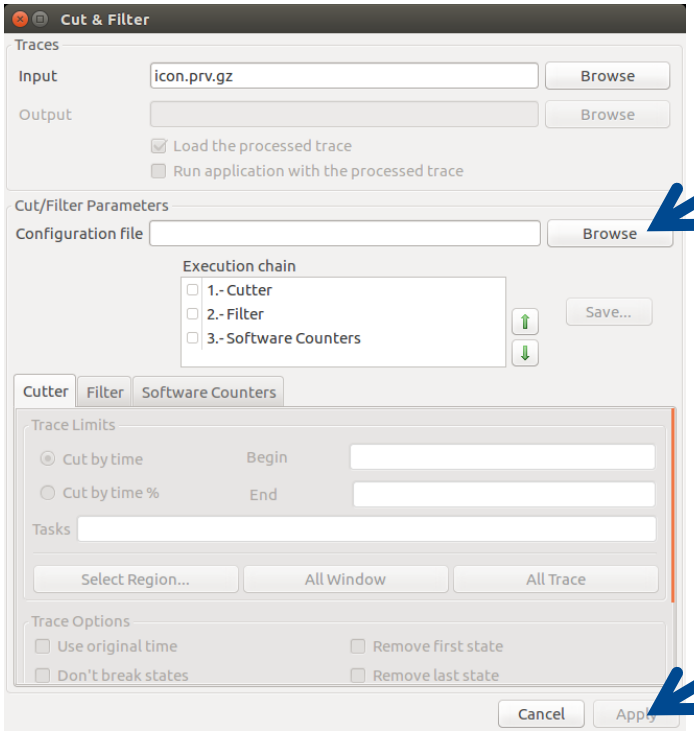


Click on "Yes"

Filter the trace

- What to filter?
  - Keep only long computations and flushing events
  - Copy this configuration from Mistral

```
> scp <USER>@mistral.dkrz.de:tools-material/paraver/filter.xml $HOME
```

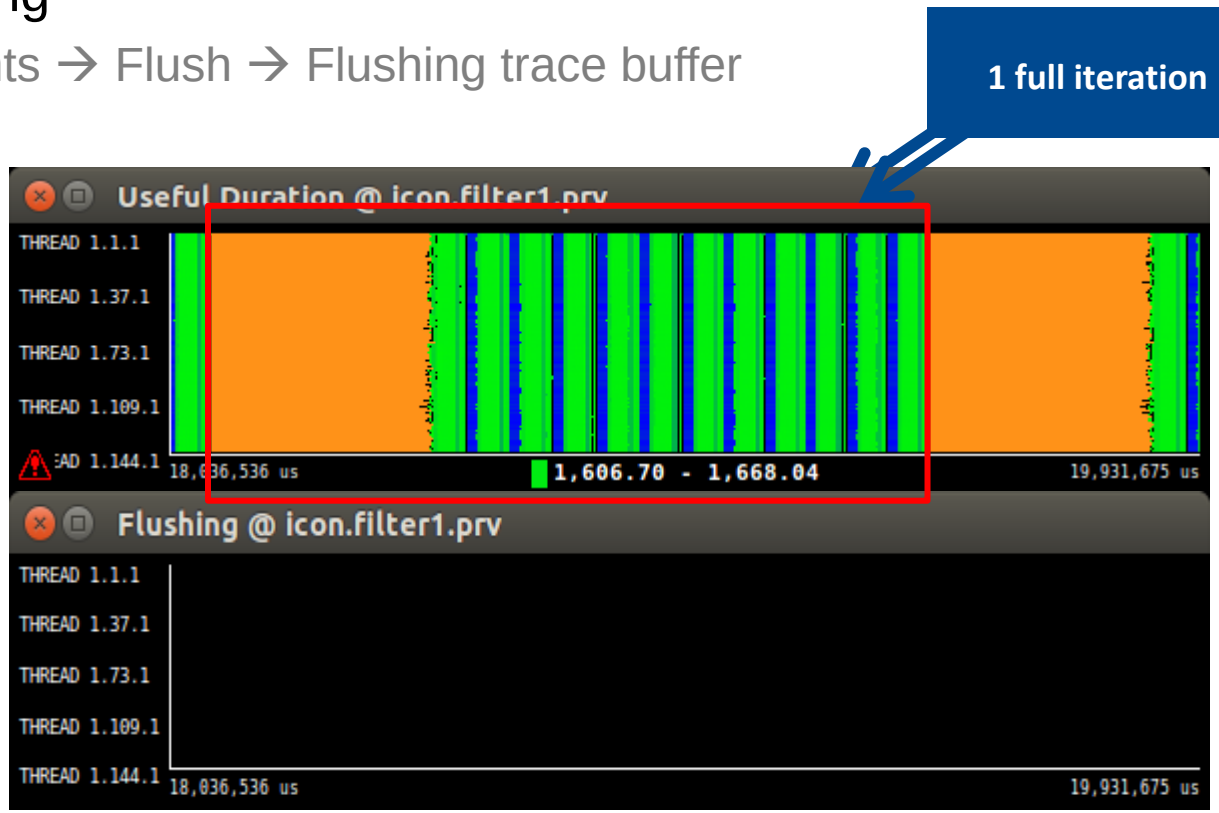


1. Click on “Browse”
2. Select “filter.xml”

3. Click on “Apply”

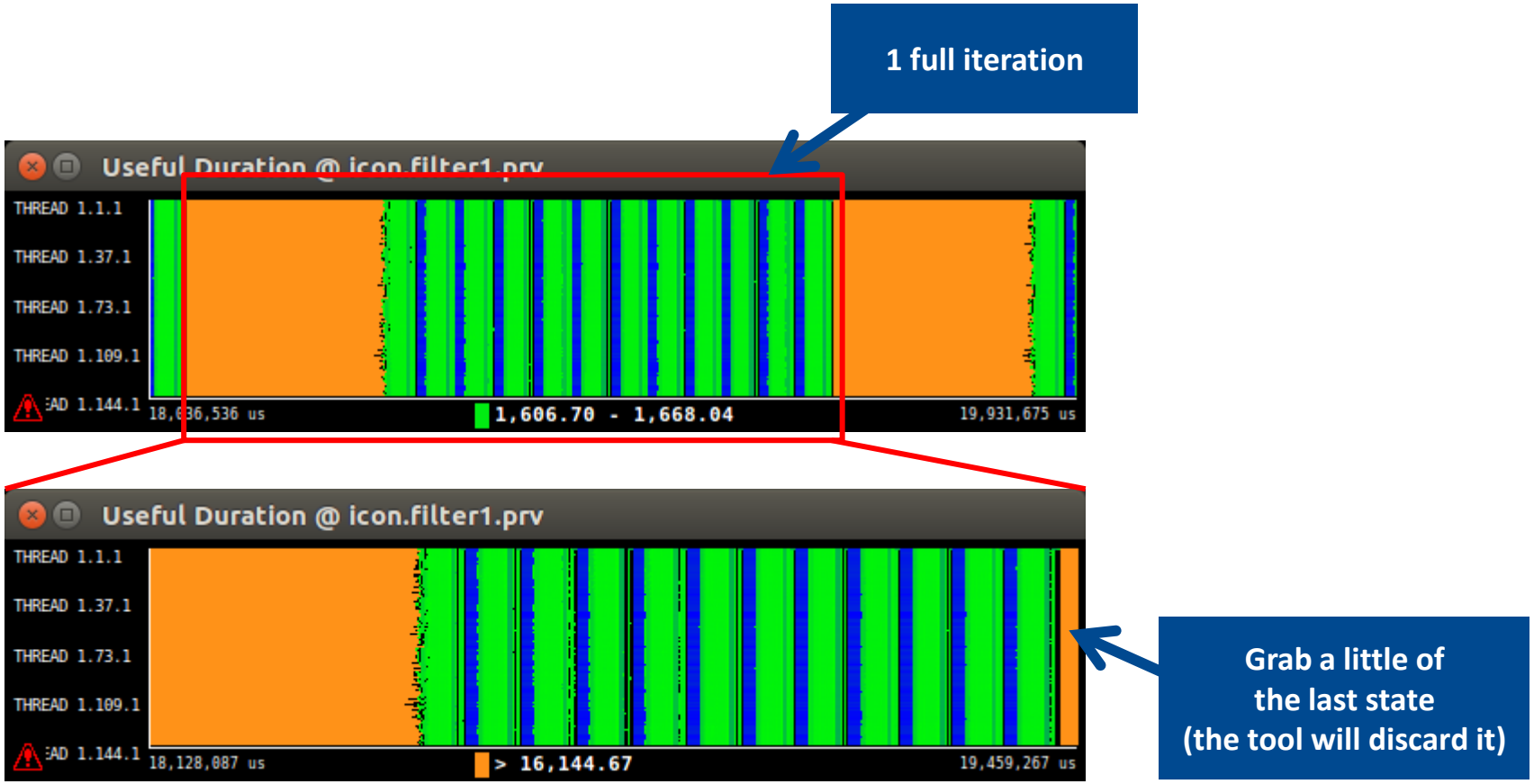
Find a representative region

- Load views:
  - Useful duration
    - File → Load Configuration → “cfgs/General/views/useful_duration.cfg”
  - Flushing
    - Hints → Flush → Flushing trace buffer



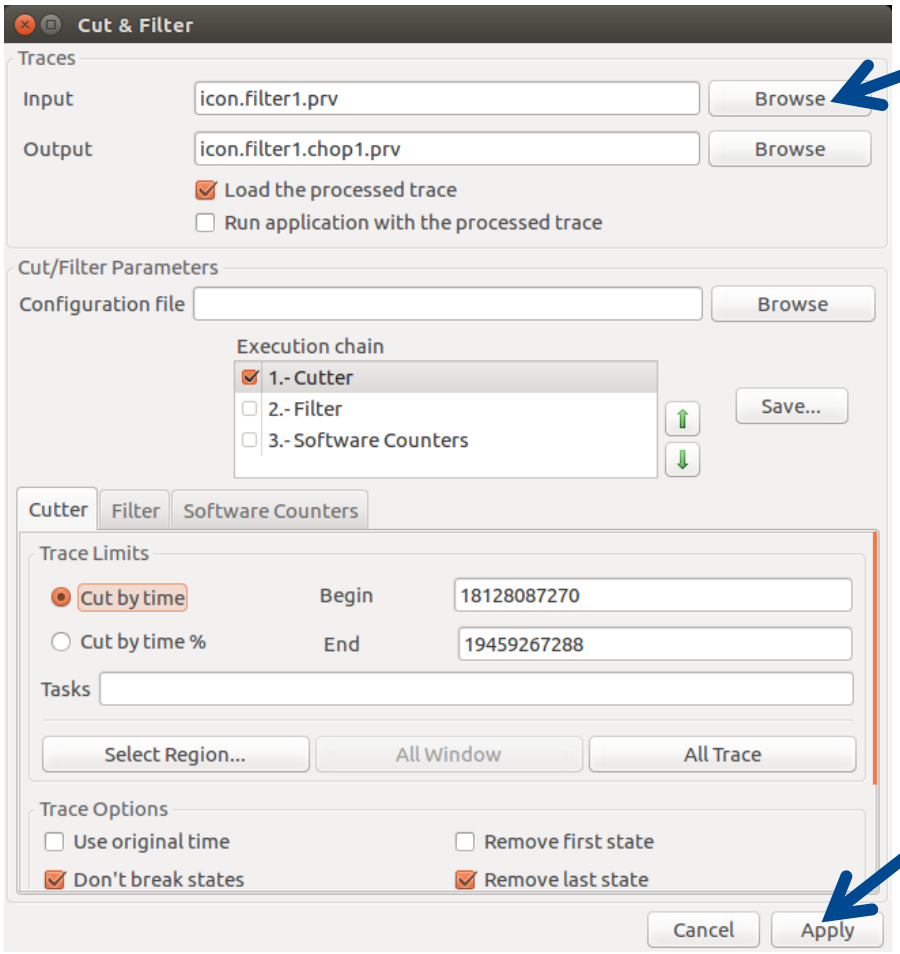
Cut the trace (I)

- Zoom the time interval to cut in the timeline → 1 iteration



Cut the trace (II)

- Get a subtrace that contains all events only for this iteration
  - Right click → Run → Cutter

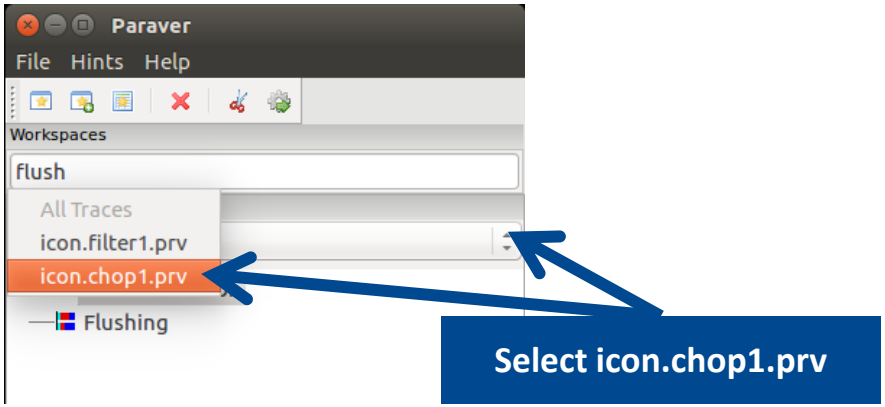


1. Click on "Browse" and select the original (big) trace: icon.prv.gz

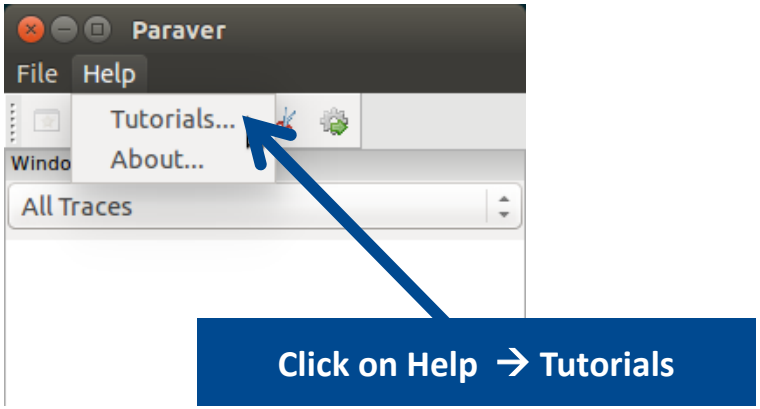
2. Click on "Apply"

First steps of analysis

- Select the generated cut

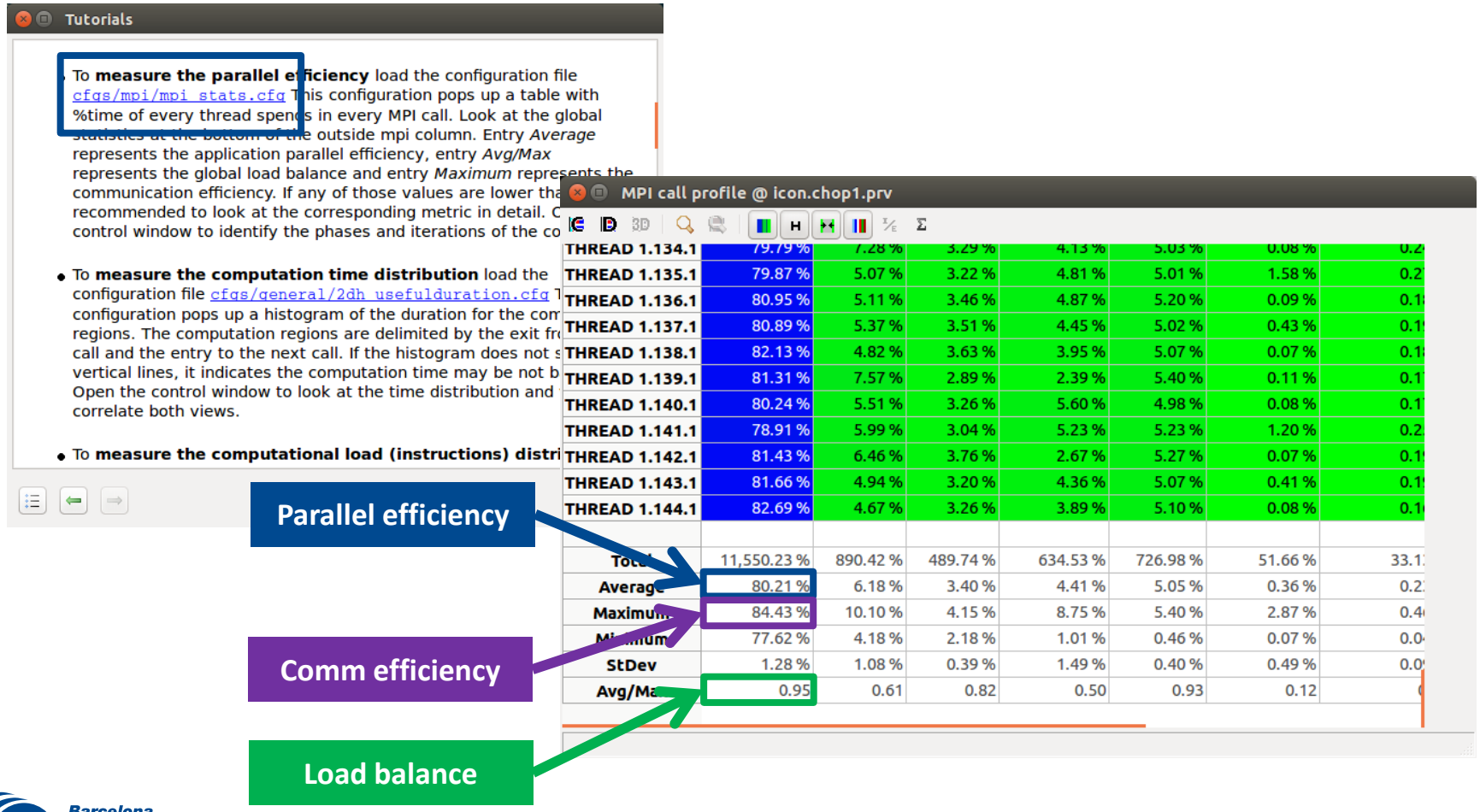


- Follow Tutorial #3

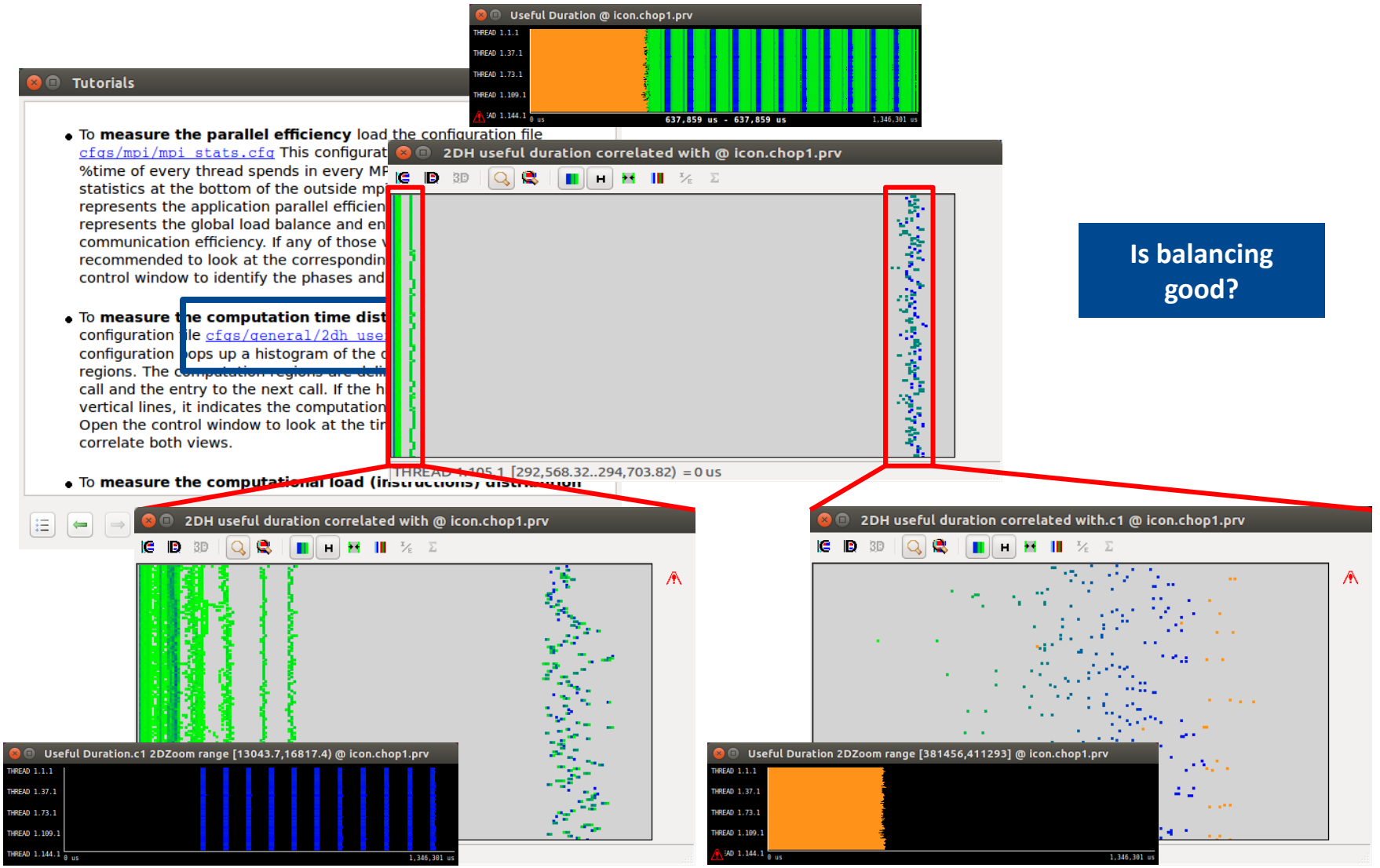


Measure the parallel efficiency

- Click on “mpi_stats.cfg”
  - Check the Average for the column labeled “Outside MPI”



Measure the computation time distribution



Cluster-based analysis

- Run the clustering tool on the trace you have generated
  - To avoid copying the cut trace back to Mistral, use a prepared cut

@ mistral.dkrz.de

```
> cd $HOME/tools-material/clustering

> ./clusterize.sh ../traces/icon.chop1.prv.gz
```

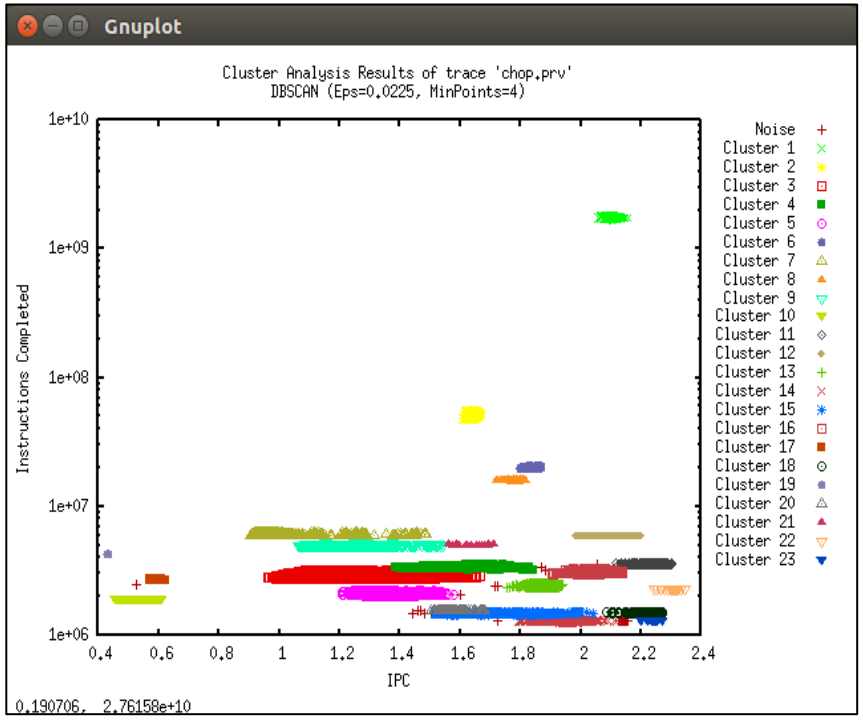
Cluster-based analysis (II)

- Check the clustering scatter plot

@ mistral.dkrz.de

```
> gnuplot icon.chop1.clustered.IPC.PAPI_TOT_INS.gnuplot
```

  - Press “L” (once) to switch to logarithmic scale



- Identify main computing trends
- Work (Y) vs. Performance (X)
- See the horizontal clusters?
  - Large IPC variability
  - Indicate potential imbalances

Cluster-based analysis (III)

Check the clustered trace

- Copy the clustered trace to your laptop

@ your laptop

```
> scp <USER>@mistral.dkrz.de:tools-material/clustering/icon.chop1.clustered.* $HOME
```

- Load with Paraver

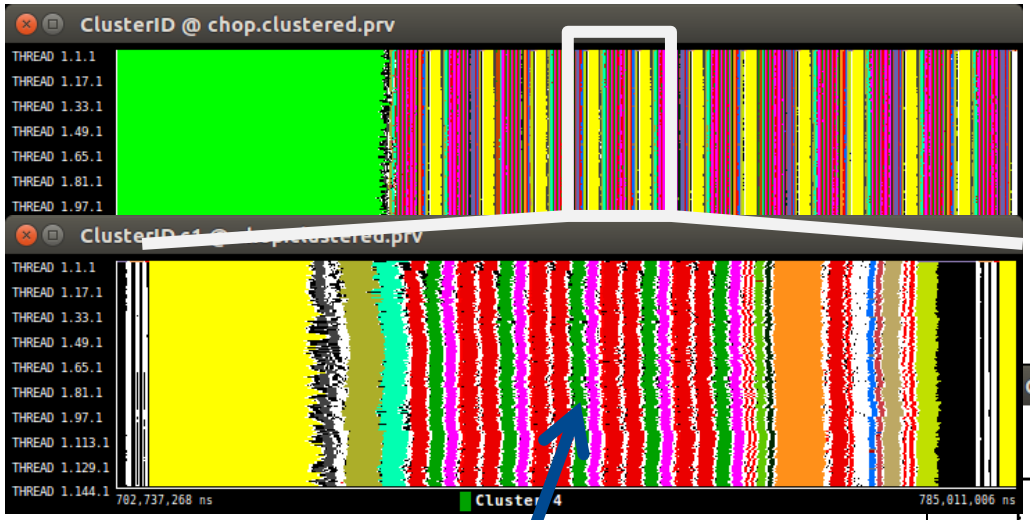
@ your laptop

```
> $HOME/paraver/bin/wxparaver $HOME/icon.chop1.clustered.prv.gz
```

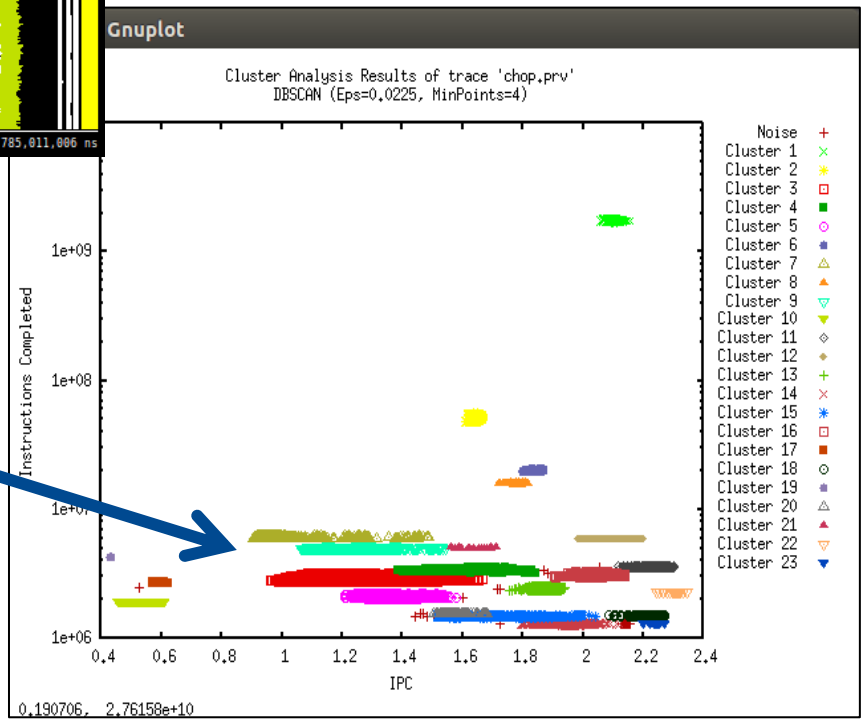
- Display the distribution of clusters over time
  - File → Load configuration
  - Select: $HOME/paraver/cfgs/clustering/clusterID_window.cfg

Cluster-based analysis (IV)

- Correlate scatter plots & timelines to detect imbalances



Constant work
+
Variable performance
+
Simultaneously @ different processes
=
Imbalances




Thank you!