BSC Tools
Extrae & Paraver
Hands-On

Germán Llort, Judit Giménez

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 sampling
- User events (Extrae API)

No need to recompile / relink!

Extrae overheads

	Average values	Stampede2 (KNL)	Stampede2 (SKX)
Event	150 – 200 ns	450 ns	110 ns
Event + PAPI	750 – 1000 ns	6.3 us	1000 ns
Event + callstack (1 level)	600 ns	8.7 us	2.3 us
Event + callstack (6 levels)	1.9 us	20 us	5.6 us

How does Extrae work?

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

Recommended

- 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

2. **Configure** what to trace
   - XML configuration file
   - Example configurations at $EXTRAE_HOME/share/example

3. **Run** it!

- For further reference check the **Extrae User Guide:**
  - https://tools.bsc.es/doc/html/extrae/index.html
  - Also distributed with Extrae at $EXTRAE_HOME/share/doc

Login to Stampede2 and copy the examples

```
laptop> ssh -Y <USER>@stampede2.tacc.xsede.org

stampede2> cp -r ~tg856590/tools-material $WORK

stampede2> ls $WORK/tools-material
    ... apps/
    ... clustering/
    ... extrae/
    ... slides/
    ... traces/
```

Here you have a copy of this slides

Step 1: Adapt the job script to load Extrae with LD_PRELOAD

```
stampede2> vi $WORK/tools-material/extrae/run_64p_knl.sh
```

```
#!/bin/bash
#SBATCH -J lulesh-knl
#SBATCH -o lulesh-knl.%j.out
#SBATCH -e lulesh-knl.%j.err
#SBATCH -N 2
#SBATCH -n 64
#SBATCH -t 00:05:00
#SBATCH -p normal

module load papi

export TRACE_NAME=lulesh_64p_knl.prv

ibrun ./trace_knl.sh ../apps/lulesh2.0 -i 10 -s 65 -p
```

Request resources

Setup environment

Output name (optional)

Activate Extrae

Step 1: Adapt the job script to load Extrae with LD_PRELOAD

```
stampede2> vi $WORK/tools-material/extrae/trace_knl.sh
```

```
#!/bin/bash
#SBATCH -J lulesh-knl
#SBATCH -o lulesh-knl.%j.out
#SBATCH -e lulesh-knl.%j.err
#SBATCH -N 2
#SBATCH -n 64
#SBATCH -t 00:05:00
#SBATCH -p normal

module load papi

export TRACE_NAME=lulesh_64p_knl.prv

ibrun ./trace_knl.sh ../apps/lulesh2.0 -i 10 -s 65 -p
```

```
#!/bin/bash

# Configure Extrae
export EXTRAE_HOME=<…point to the install folder…>
export EXTRAE_CONFIG_FILE=./extrae_knl.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
$*
```

Select "what to trace"

Select your type of application

Step 1: Which tracing library?

- Choose depending on the application type

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

*** include suffix “f” for Fortran apps**

Step 3: Run it!

- Submit your job

```
stampede2> cd $WORK/tools-material/extrae

stampede2> sbatch run_64p_knl.sh
```

- Once finished the trace will be in the same folder: lulesh_64p_knl.{pcf,prv,row} **(3 files)**
  - Check the status of your job with: squeue –u $USER
- Any issue?
  - Already generated at $WORK/tools-material/traces

Step 2: Extrae XML configuration

```
stampede2> vi $WORK/tools-material/extrae/extrae_knl.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 the MPI calls
(What's the program doing?)

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

Compile with debug! (-g)

Step 2: Extrae XML configuration (II)

```
<counters enabled="yes">
  <cpu enabled="yes" starting-set-distribution=“1">
    <set enabled="yes" changeat-time="500000us" domain="all“>
      PAPI_TOT_INS, PAPI_TOT_CYC, PAPI_L1_DCM, PAPI_BR_INS
    </set>
    <set enabled="yes" changeat-time="500000us" domain="all">
      PAPI_TOT_INS, PAPI_TOT_CYC, PAPI_BR_MSP, PAPI_LD_INS
    </set>
    <set … /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: Extrae XML configuration (III)

```
<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
(Flush/memory trade-off)

Enable sampling
(Want more details?)

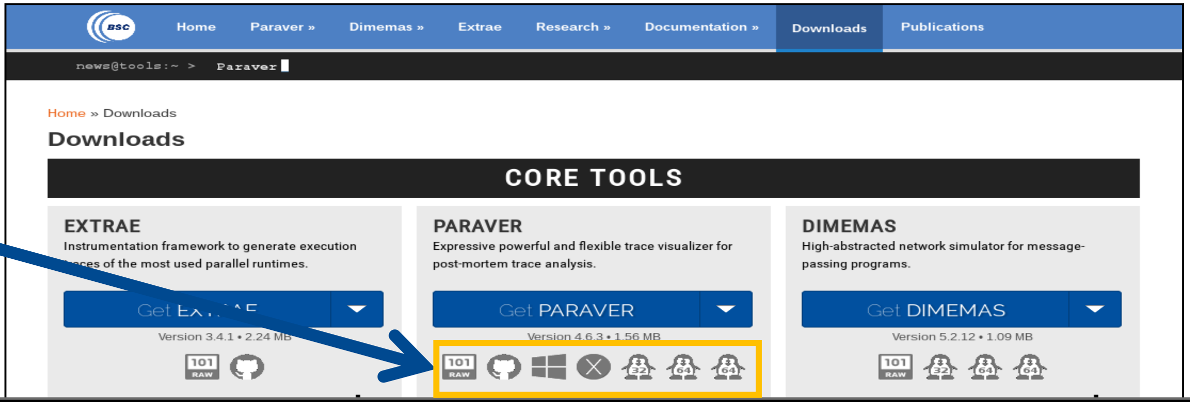
Automatic post-processing to generate the Paraver trace

Installing Paraver
&
First analysis steps

Install Paraver in your laptop

- Download from http://tools.bsc.es/downloads

Pick your version



- Also available @Stampede2
  - ~tg856590/tools-packages

wxparaver-4.8.1-win.zip

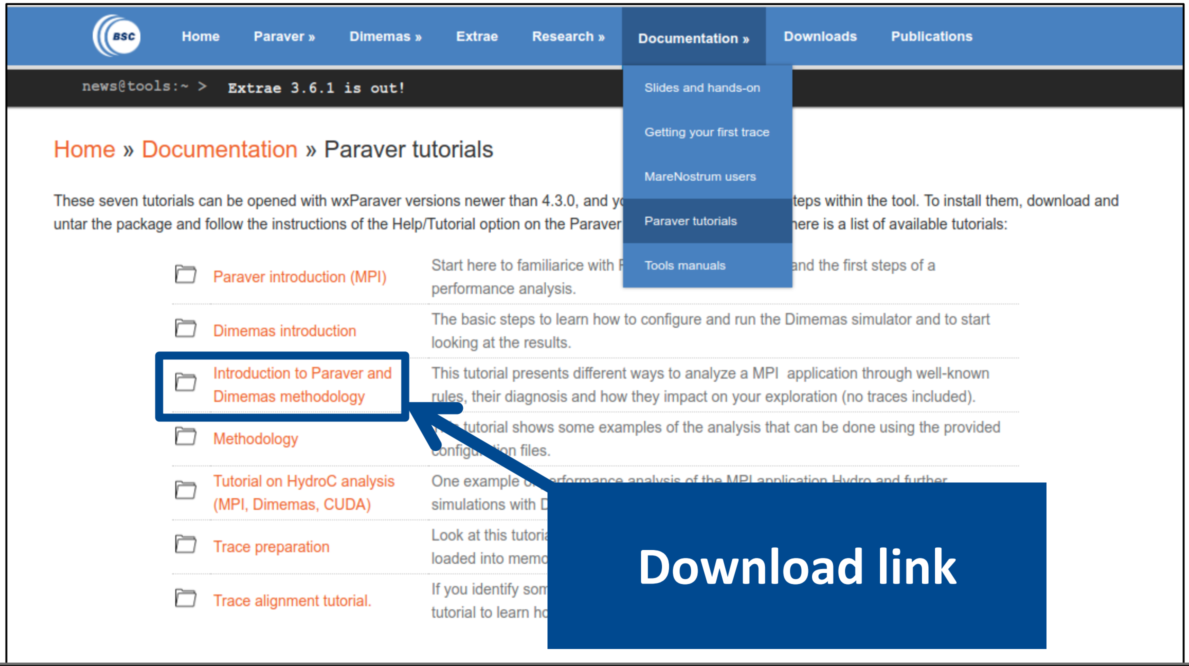
wxparaver-4.8.1-mac.zip

wxparaver-4.8.1-Linux_i686.tar.gz (32-bits)
wxparaver-4.8.1-Linux_x86_64.tar.gz (64-bits)

```
laptop> scp <USER>@stampede2.tacc.xsede.org:~tg856590/tools-packages/<PACKAGE> $HOME
```

Install Paraver (II)

- Download tutorials:
  - Documentation → Paraver tutorials → Introduction to Paraver and Dimemas methodology



- Also available @Stampede2
  - ~tg856590/tools-packages

```
laptop> scp <USER>@stampede2.tacc.xsede.org:~tg856590/tools-packages/3.* $HOME
```

Install tutorial

- Uncompress both packages
- Move the tutorial folder (3.Introduction*) inside "paraver" /tutorials" folder

?

```
laptop> tar xf wxparaver-4.8.1-linux-x86_64.tar.gz
laptop> mv wxparaver-4.8.1-linux-x86_64 paraver
laptop> mkdir paraver/tutorials
laptop> tar xf 3.introduction*
laptop> mv 3.Introduction* paraver/tutorials
laptop> ls -l paraver/tutorials
drwxrwxr-x … 3.Introduction_to_Paraver_and_Dimemas_methodology
```

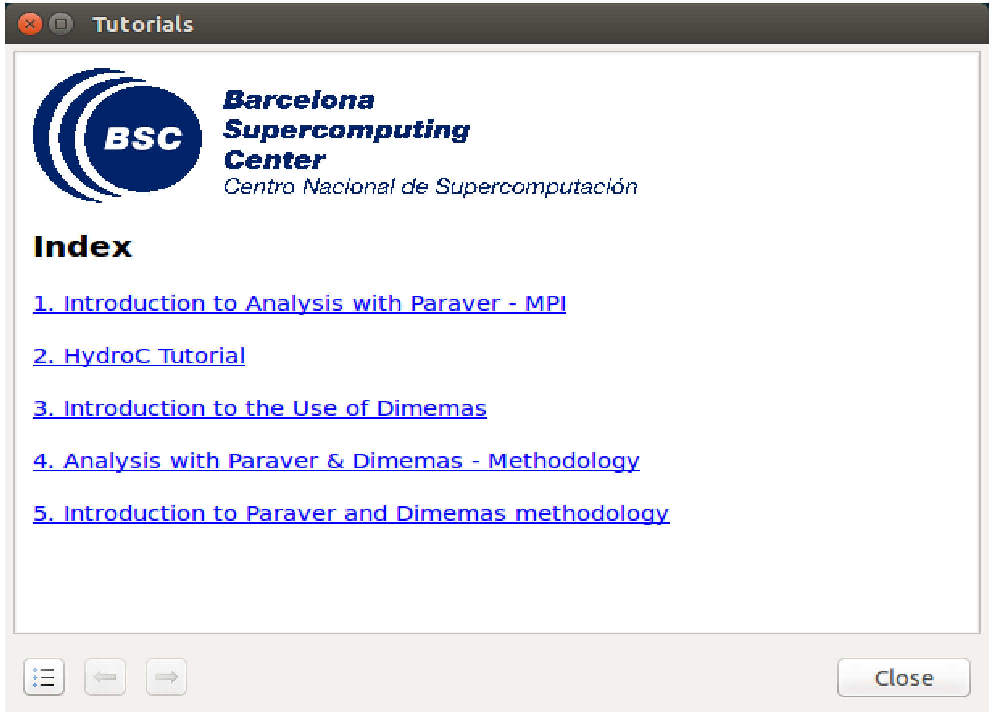
? Right click on wxparaver.app → Show Package Contents → Contents → Resources

Check that everything works

- Start Paraver

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

- Check that tutorials are available



- Remotely available in Stampede2

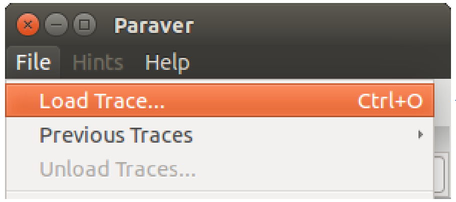
```
laptop> ssh -Y <USER>@stampede2.tacc.xsede.org
stampede2> ~tg856590/tools/paraver/bin/wxparaver &
```

First steps of analysis

- Copy the trace to your laptop (All 3 files: *.prv, *.pcf, *.row)

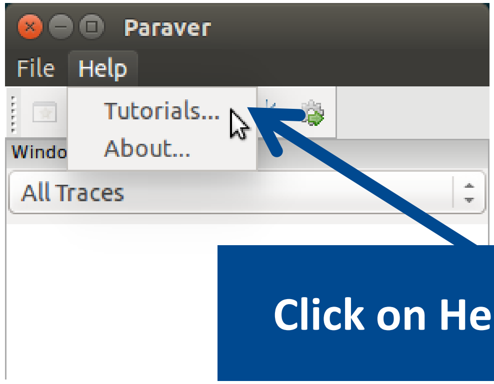
```
laptop> scp <USER>@stampede2.tacc.xsede.org:$WORK/tools-material/extrae/lulesh_64p_knl.* ./
```

- Load the trace

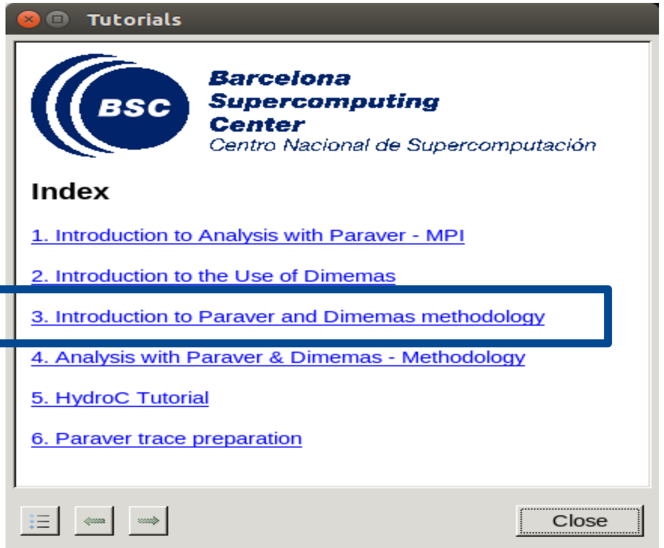


Click on File → Load Trace → Browse to the *.prv file

- Follow Tutorial #3
  - Introduction to Paraver and Dimemas methodology

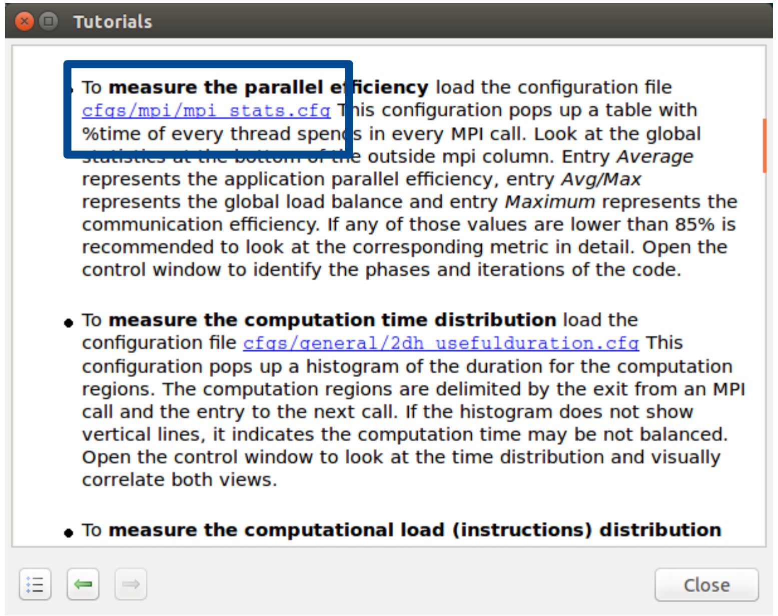


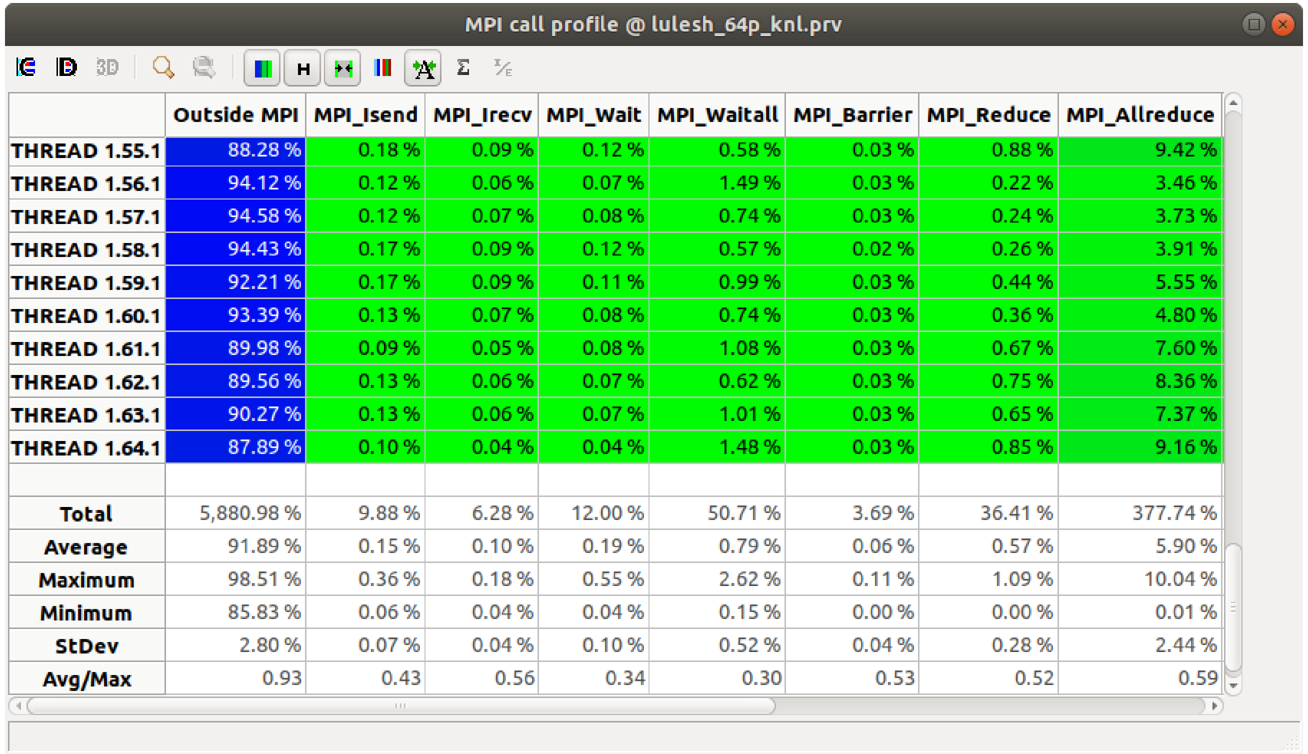
Click on Help → Tutorials



Measure the parallel efficiency

- Click on the “mpi_stats.cfg”



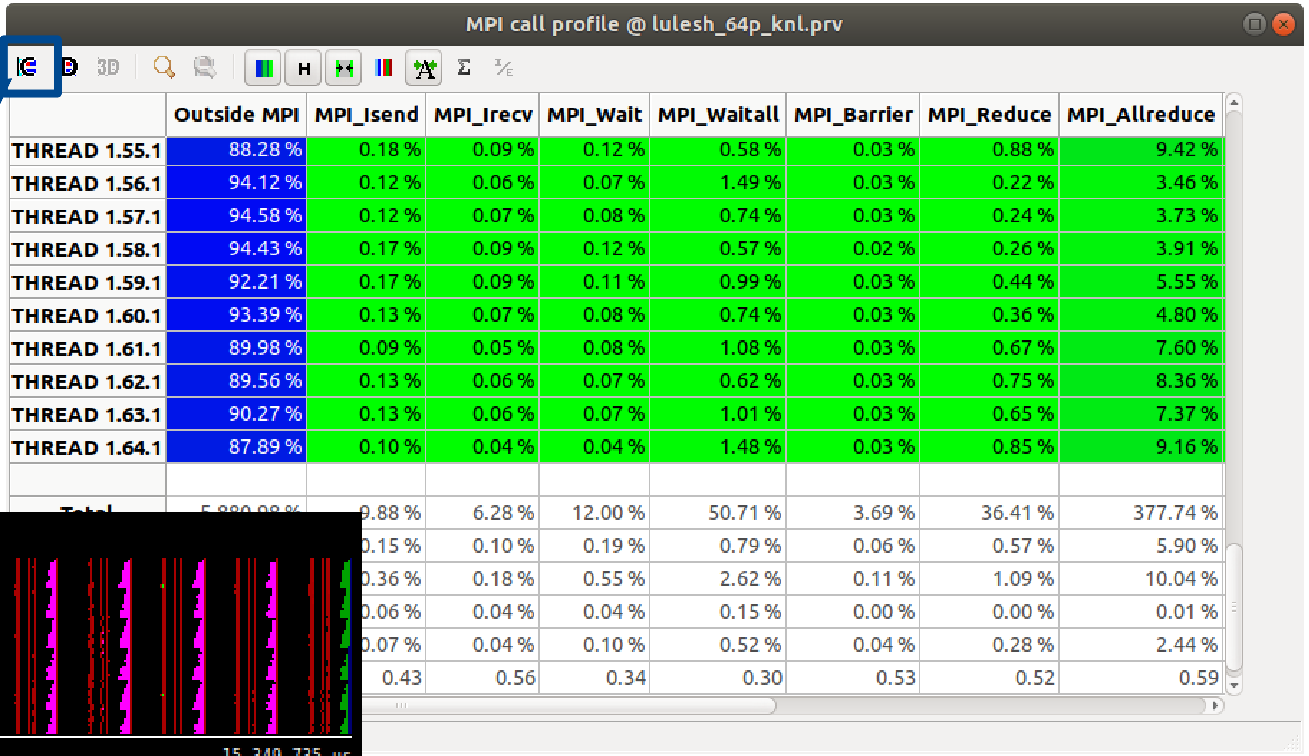

MPI call profile @ lulesh_64p_knl.prv

	Outside MPI	MPI_Isend	MPI_Irecv	MPI_Wait	MPI_Waitall	MPI_Barrier	MPI_Reduce	MPI_Allreduce
THREAD 1.55.1	88.28 %	0.18 %	0.09 %	0.12 %	0.58 %	0.03 %	0.88 %	9.42 %
THREAD 1.56.1	94.12 %	0.12 %	0.06 %	0.07 %	1.49 %	0.03 %	0.22 %	3.46 %
THREAD 1.57.1	94.58 %	0.12 %	0.07 %	0.08 %	0.74 %	0.03 %	0.24 %	3.73 %
THREAD 1.58.1	94.43 %	0.17 %	0.09 %	0.12 %	0.57 %	0.02 %	0.26 %	3.91 %
THREAD 1.59.1	92.21 %	0.17 %	0.09 %	0.11 %	0.99 %	0.03 %	0.44 %	5.55 %
THREAD 1.60.1	93.39 %	0.13 %	0.07 %	0.08 %	0.74 %	0.03 %	0.36 %	4.80 %
THREAD 1.61.1	89.98 %	0.09 %	0.05 %	0.08 %	1.08 %	0.03 %	0.67 %	7.60 %
THREAD 1.62.1	89.56 %	0.13 %	0.06 %	0.07 %	0.62 %	0.03 %	0.75 %	8.36 %
THREAD 1.63.1	90.27 %	0.13 %	0.06 %	0.07 %	1.01 %	0.03 %	0.65 %	7.37 %
THREAD 1.64.1	87.89 %	0.10 %	0.04 %	0.04 %	1.48 %	0.03 %	0.85 %	9.16 %
Total	5,880.98 %	9.88 %	6.28 %	12.00 %	50.71 %	3.69 %	36.41 %	377.74 %
Average	91.89 %	0.15 %	0.10 %	0.19 %	0.79 %	0.06 %	0.57 %	5.90 %
Maximum	98.51 %	0.36 %	0.18 %	0.55 %	2.62 %	0.11 %	1.09 %	10.04 %
Minimum	85.83 %	0.06 %	0.04 %	0.04 %	0.15 %	0.00 %	0.00 %	0.01 %
StDev	2.80 %	0.07 %	0.04 %	0.10 %	0.52 %	0.04 %	0.28 %	2.44 %
Avg/Max	0.93	0.43	0.56	0.34	0.30	0.53	0.52	0.59

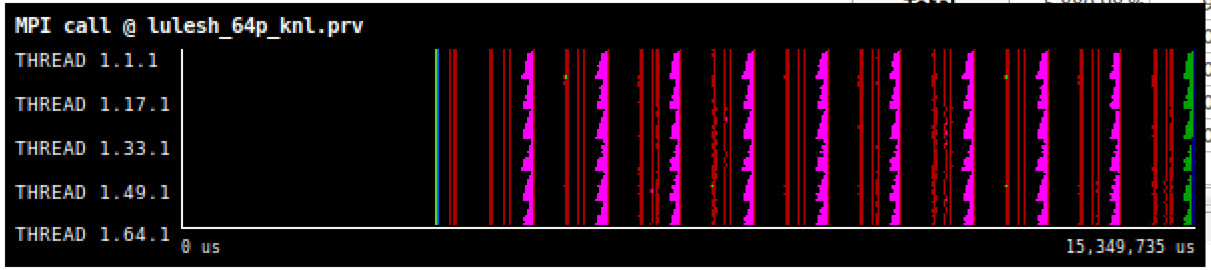
Measure the parallel efficiency

- Click on the "mpi_stats.cfg"

Click on
"Open Control Window"



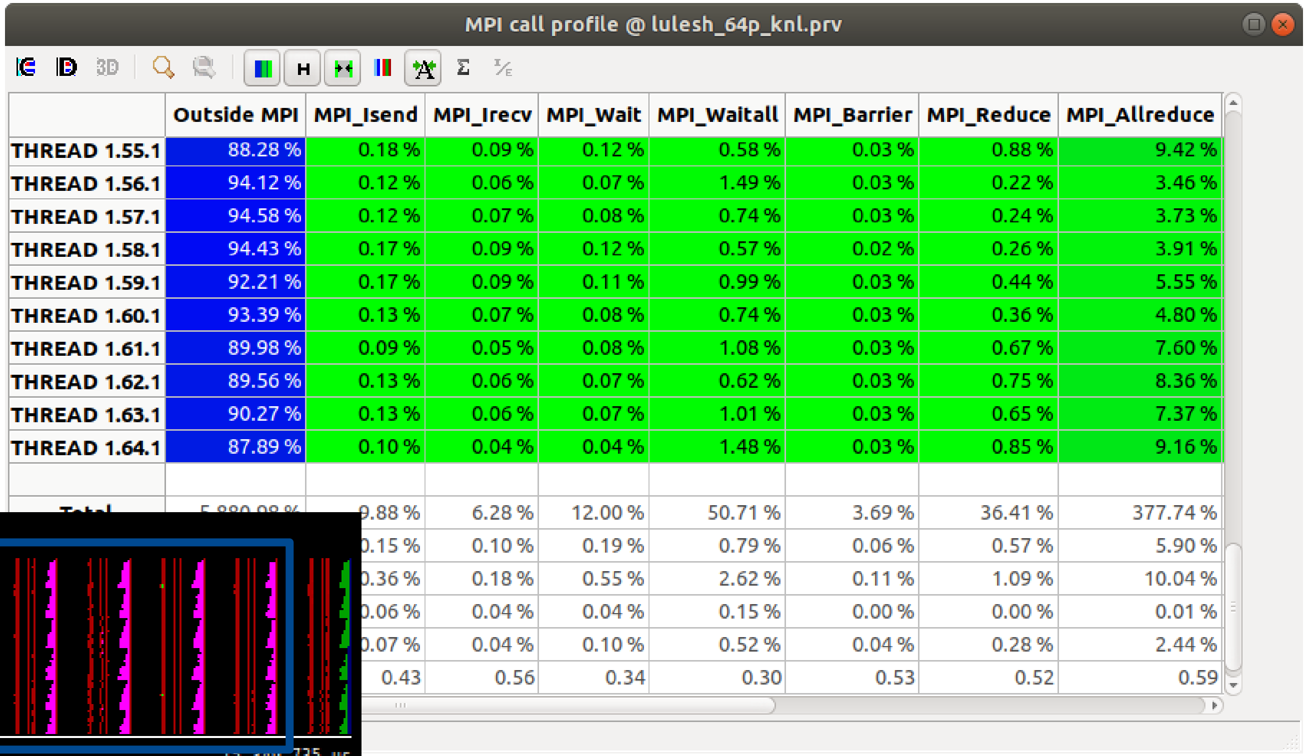
	Outside MPI	MPI_Isend	MPI_Irecv	MPI_Wait	MPI_Waitall	MPI_Barrier	MPI_Reduce	MPI_Allreduce
THREAD 1.55.1	88.28 %	0.18 %	0.09 %	0.12 %	0.58 %	0.03 %	0.88 %	9.42 %
THREAD 1.56.1	94.12 %	0.12 %	0.06 %	0.07 %	1.49 %	0.03 %	0.22 %	3.46 %
THREAD 1.57.1	94.58 %	0.12 %	0.07 %	0.08 %	0.74 %	0.03 %	0.24 %	3.73 %
THREAD 1.58.1	94.43 %	0.17 %	0.09 %	0.12 %	0.57 %	0.02 %	0.26 %	3.91 %
THREAD 1.59.1	92.21 %	0.17 %	0.09 %	0.11 %	0.99 %	0.03 %	0.44 %	5.55 %
THREAD 1.60.1	93.39 %	0.13 %	0.07 %	0.08 %	0.74 %	0.03 %	0.36 %	4.80 %
THREAD 1.61.1	89.98 %	0.09 %	0.05 %	0.08 %	1.08 %	0.03 %	0.67 %	7.60 %
THREAD 1.62.1	89.56 %	0.13 %	0.06 %	0.07 %	0.62 %	0.03 %	0.75 %	8.36 %
THREAD 1.63.1	90.27 %	0.13 %	0.06 %	0.07 %	1.01 %	0.03 %	0.65 %	7.37 %
THREAD 1.64.1	87.89 %	0.10 %	0.04 %	0.04 %	1.48 %	0.03 %	0.85 %	9.16 %
[illegible]	[illegible]	0.88 %	6.28 %	12.00 %	50.71 %	3.69 %	36.41 %	377.74 %
		0.15 %	0.10 %	0.19 %	0.79 %	0.06 %	0.57 %	5.90 %
		0.36 %	0.18 %	0.55 %	2.62 %	0.11 %	1.09 %	10.04 %
		0.06 %	0.04 %	0.04 %	0.15 %	0.00 %	0.00 %	0.01 %
		0.07 %	0.04 %	0.10 %	0.52 %	0.04 %	0.28 %	2.44 %
		0.43	0.56	0.34	0.30	0.53	0.52	0.59



Measure the parallel efficiency

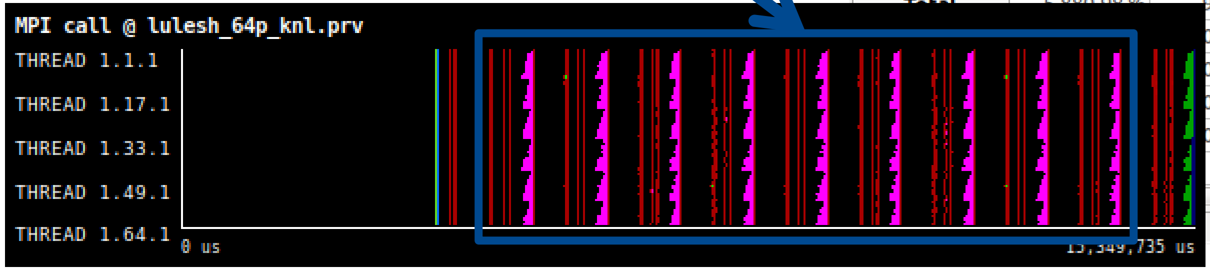
- Click on the “mpi_stats.cfg”

Zoom to skip initialization / finalization phases (drag & drop)



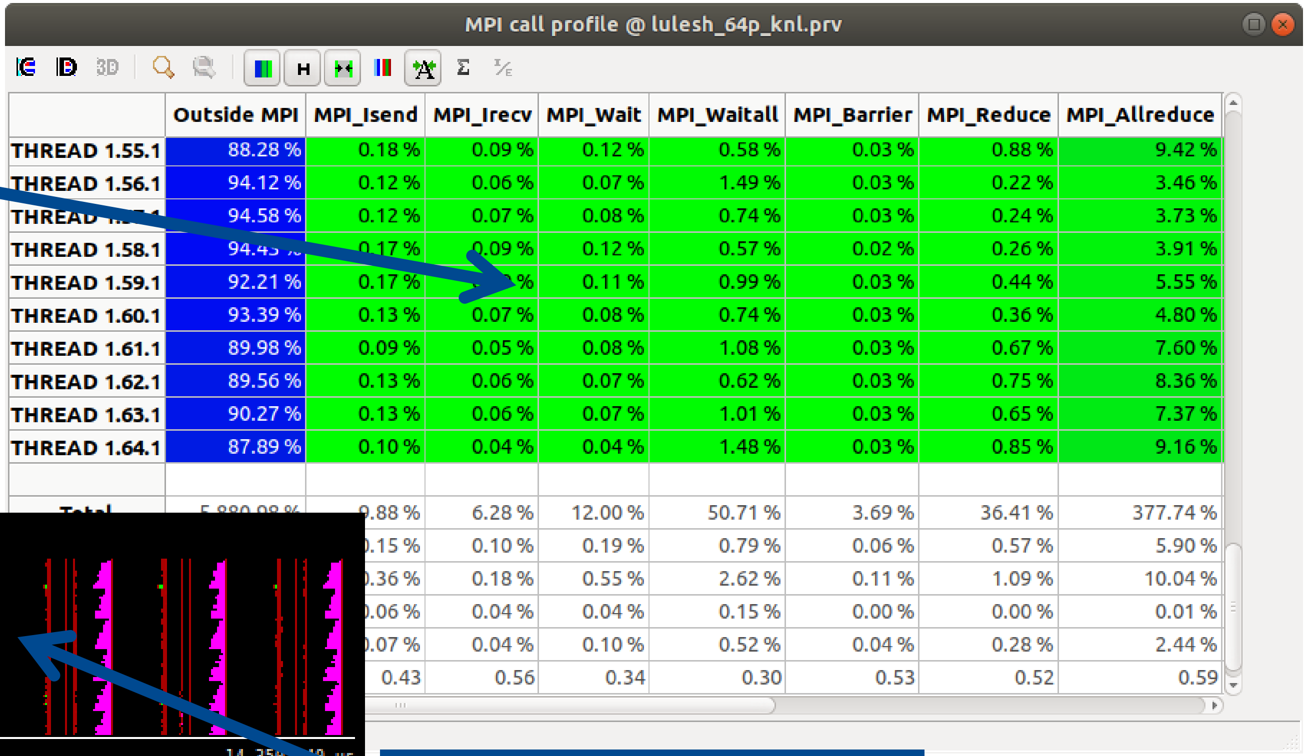
MPI call profile @ lulesh_64p_knl.prv

	Outside MPI	MPI_Isend	MPI_Irecv	MPI_Wait	MPI_Waitall	MPI_Barrier	MPI_Reduce	MPI_Allreduce
THREAD 1.55.1	88.28 %	0.18 %	0.09 %	0.12 %	0.58 %	0.03 %	0.88 %	9.42 %
THREAD 1.56.1	94.12 %	0.12 %	0.06 %	0.07 %	1.49 %	0.03 %	0.22 %	3.46 %
THREAD 1.57.1	94.58 %	0.12 %	0.07 %	0.08 %	0.74 %	0.03 %	0.24 %	3.73 %
THREAD 1.58.1	94.43 %	0.17 %	0.09 %	0.12 %	0.57 %	0.02 %	0.26 %	3.91 %
THREAD 1.59.1	92.21 %	0.17 %	0.09 %	0.11 %	0.99 %	0.03 %	0.44 %	5.55 %
THREAD 1.60.1	93.39 %	0.13 %	0.07 %	0.08 %	0.74 %	0.03 %	0.36 %	4.80 %
THREAD 1.61.1	89.98 %	0.09 %	0.05 %	0.08 %	1.08 %	0.03 %	0.67 %	7.60 %
THREAD 1.62.1	89.56 %	0.13 %	0.06 %	0.07 %	0.62 %	0.03 %	0.75 %	8.36 %
THREAD 1.63.1	90.27 %	0.13 %	0.06 %	0.07 %	1.01 %	0.03 %	0.65 %	7.37 %
THREAD 1.64.1	87.89 %	0.10 %	0.04 %	0.04 %	1.48 %	0.03 %	0.85 %	9.16 %
Total	[illegible]	0.88 %	6.28 %	12.00 %	50.71 %	3.69 %	36.41 %	377.74 %
[illegible]	[illegible]	0.15 %	0.10 %	0.19 %	0.79 %	0.06 %	0.57 %	5.90 %
[illegible]	[illegible]	0.36 %	0.18 %	0.55 %	2.62 %	0.11 %	1.09 %	10.04 %
[illegible]	[illegible]	0.06 %	0.04 %	0.04 %	0.15 %	0.00 %	0.00 %	0.01 %
[illegible]	[illegible]	0.07 %	0.04 %	0.10 %	0.52 %	0.04 %	0.28 %	2.44 %
[illegible]	[illegible]	0.43	0.56	0.34	0.30	0.53	0.52	0.59



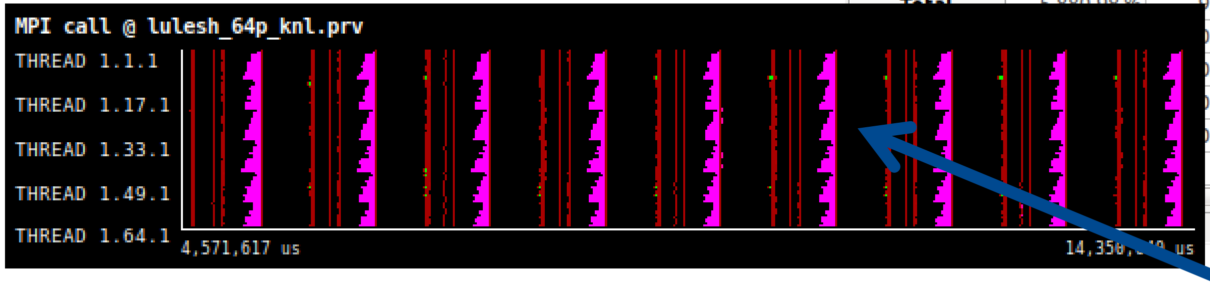
Measure the parallel efficiency

- Click on the "mpi_stats.cfg"



	Outside MPI	MPI_Isend	MPI_Irecv	MPI_Wait	MPI_Waitall	MPI_Barrier	MPI_Reduce	MPI_Allreduce
THREAD 1.55.1	88.28 %	0.18 %	0.09 %	0.12 %	0.58 %	0.03 %	0.88 %	9.42 %
THREAD 1.56.1	94.12 %	0.12 %	0.06 %	0.07 %	1.49 %	0.03 %	0.22 %	3.46 %
THREAD 1.57.1	94.58 %	0.12 %	0.07 %	0.08 %	0.74 %	0.03 %	0.24 %	3.73 %
THREAD 1.58.1	94.43 %	0.17 %	0.09 %	0.12 %	0.57 %	0.02 %	0.26 %	3.91 %
THREAD 1.59.1	92.21 %	0.17 %	[illegible]	0.11 %	0.99 %	0.03 %	0.44 %	5.55 %
THREAD 1.60.1	93.39 %	0.13 %	0.07 %	0.08 %	0.74 %	0.03 %	0.36 %	4.80 %
THREAD 1.61.1	89.98 %	0.09 %	0.05 %	0.08 %	1.08 %	0.03 %	0.67 %	7.60 %
THREAD 1.62.1	89.56 %	0.13 %	0.06 %	0.07 %	0.62 %	0.03 %	0.75 %	8.36 %
THREAD 1.63.1	90.27 %	0.13 %	0.06 %	0.07 %	1.01 %	0.03 %	0.65 %	7.37 %
THREAD 1.64.1	87.89 %	0.10 %	0.04 %	0.04 %	1.48 %	0.03 %	0.85 %	9.16 %
Total	5,880.98 %	.88 %	6.28 %	12.00 %	50.71 %	3.69 %	36.41 %	377.74 %
		.15 %	0.10 %	0.19 %	0.79 %	0.06 %	0.57 %	5.90 %
		.36 %	0.18 %	0.55 %	2.62 %	0.11 %	1.09 %	10.04 %
		.06 %	0.04 %	0.04 %	0.15 %	0.00 %	0.00 %	0.01 %
		.07 %	0.04 %	0.10 %	0.52 %	0.04 %	0.28 %	2.44 %
		0.43	0.56	0.34	0.30	0.53	0.52	0.59

2. Right click → Paste → Time



1. Right click → Copy

Measure the parallel efficiency

- Click on the “mpi_stats.cfg”

MPI call profile @ lulesh_64p_knl.prv

	Outside MPI	MPI_Isend	MPI_Irecv	MPI_Wait	MPI_Waitall	MPI_Allreduce	MPI_Comm_rank
THREAD 1.55.1	83.88 %	0.23 %	0.11 %	0.15 %	0.81 %	14.74 %	0.07 %
THREAD 1.56.1	92.06 %	0.16 %	0.07 %	0.09 %	2.12 %	5.41 %	0.07 %
THREAD 1.57.1	92.73 %	0.16 %	0.08 %	0.11 %	1.03 %	5.83 %	0.07 %
THREAD 1.58.1	92.52 %	0.23 %	0.11 %	0.15 %	0.81 %	6.12 %	0.07 %
THREAD 1.59.1	89.40 %	0.22 %	0.11 %	0.14 %	1.38 %	8.68 %	0.06 %
THREAD 1.60.1	91.04 %	0.17 %	0.08 %	0.10 %	1.03 %	7.52 %	0.07 %
THREAD 1.61.1	86.25 %	0.12 %	0.05 %	0.11 %	1.53 %	11.89 %	0.06 %
THREAD 1.62.1	85.66 %	0.17 %	0.07 %	0.09 %	0.87 %	13.09 %	0.07 %
THREAD 1.63.1	86.64 %	0.17 %	0.07 %	0.10 %	1.43 %	11.53 %	0.07 %
THREAD 1.64.1	83.37 %	0.13 %	0.04 %	0.05 %	2.01 %	14.33 %	0.07 %
Total	5,698.23 %	12.04 %	7.73 %	15.76 %	70.74 %	591.10 %	4.41 %
Average	89.03 %	0.19 %	0.12 %	0.25 %	1.11 %	9.24 %	0.07 %
Maximum	98.36 %	0.33 %	0.23 %	0.76 %	3.72 %	15.70 %	0.08 %
Minimum	80.66 %	0.07 %	0.04 %	0.05 %	0.21 %	0.02 %	0.06 %
StDev	3.94 %	0.07 %	0.05 %	0.14 %	0.72 %	3.82 %	0.01 %
Avg/Max	0.91	0.57	0.53	0.32	0.30	0.59	0.81

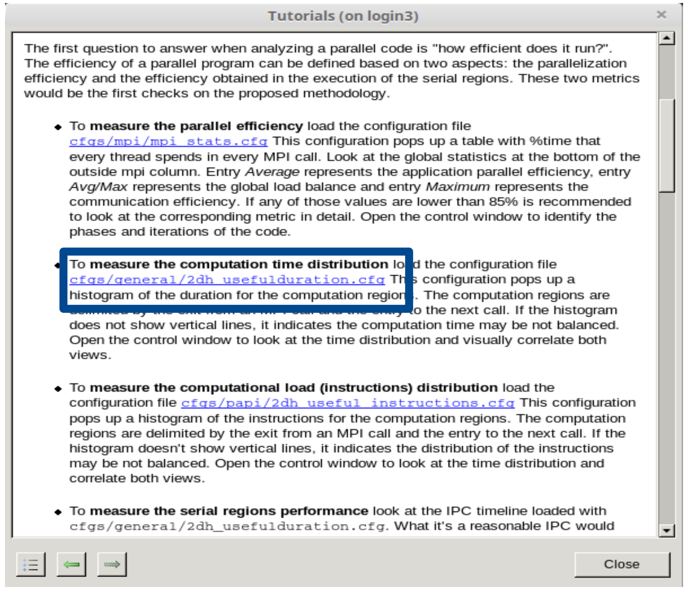
Parallel efficiency
(89% computing)

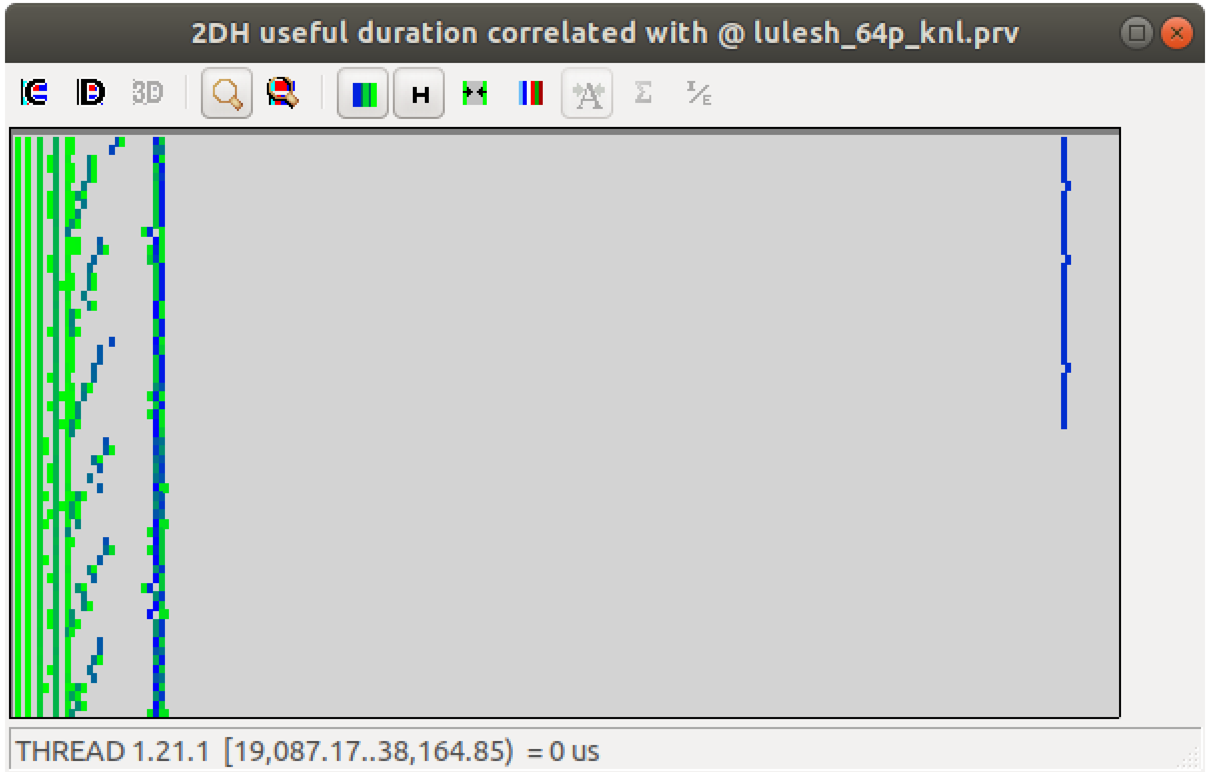
Comm efficiency
(<2% communicating)

Load balance
(9% process variability)

Computation time and work distribution

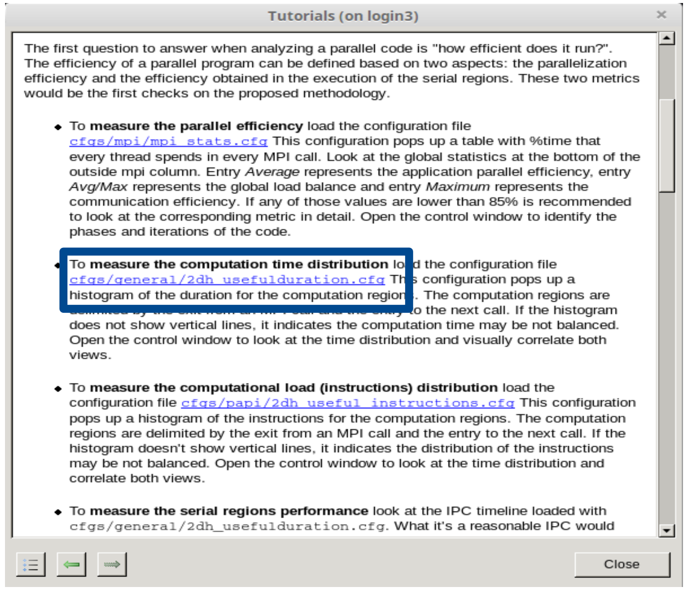
- Click on "2dh_usefulduration.cfg" (2nd link) ➔ Shows **time computing**

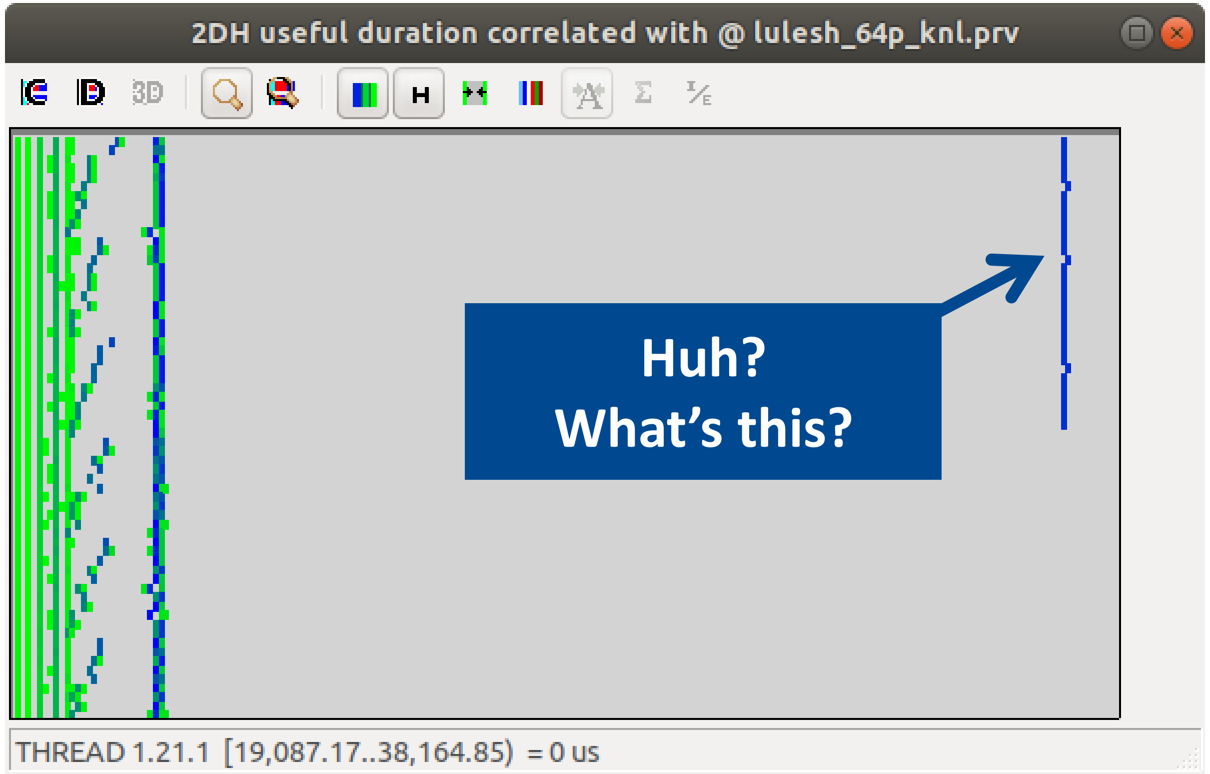




Computation time and work distribution

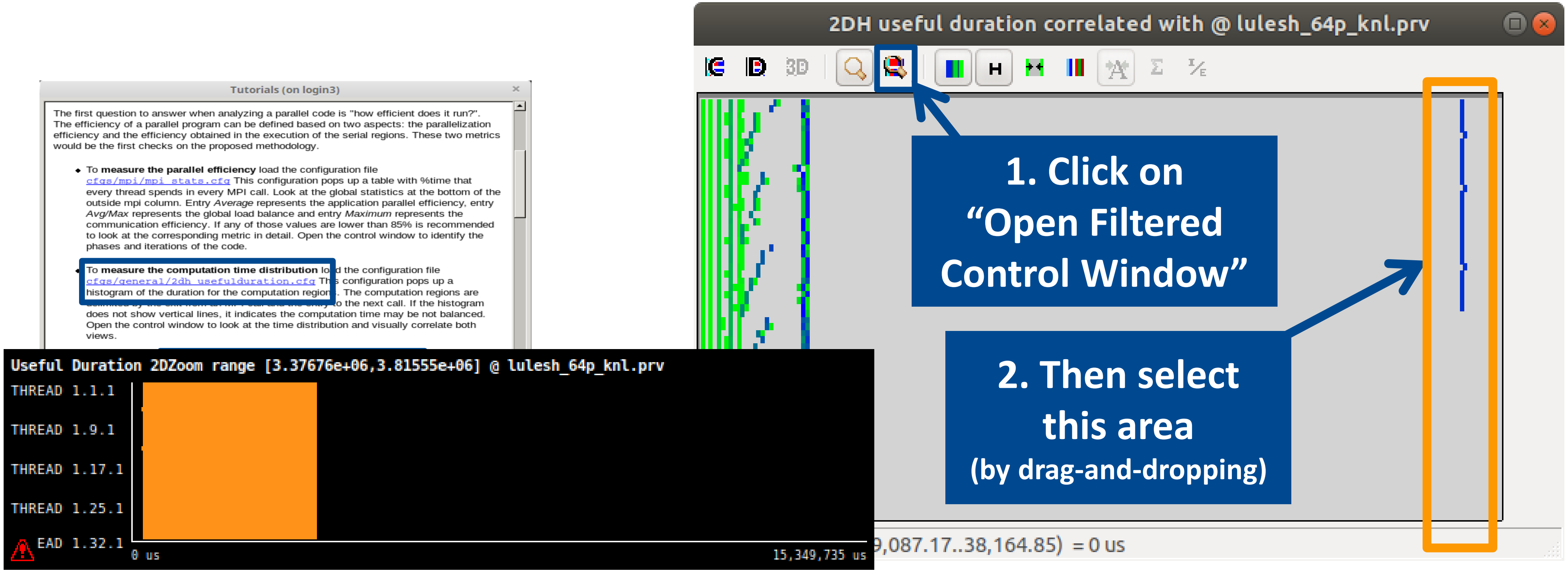
- Click on “2dh_usefulduration.cfg” (2nd link) ➔ Shows **time computing**





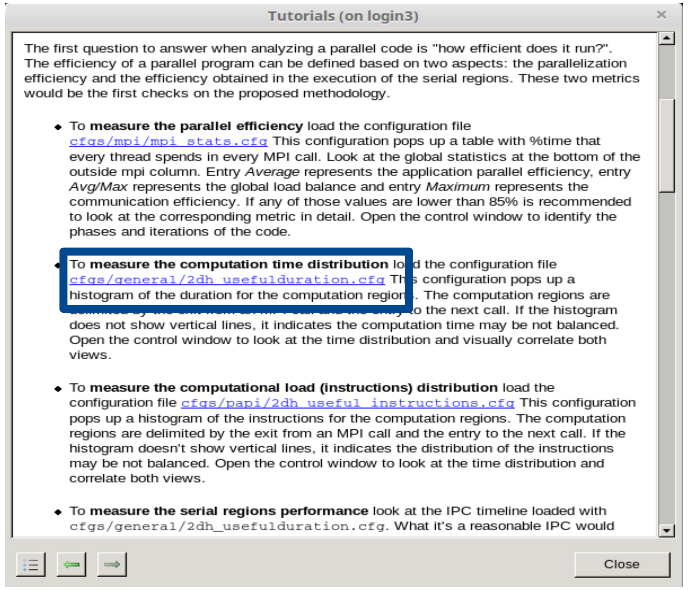
Computation time and work distribution

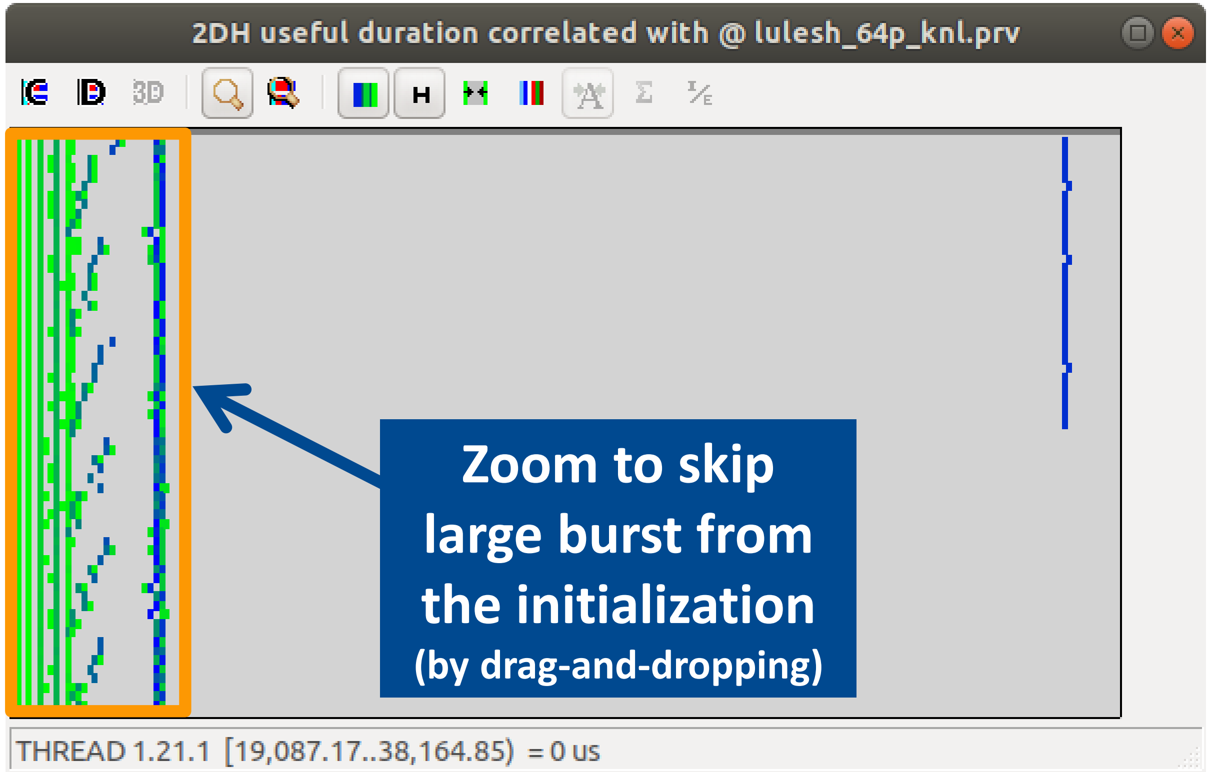
- Click on “2dh_usefulduration.cfg” (2nd link) ➔ Shows **time computing**



Computation time and work distribution

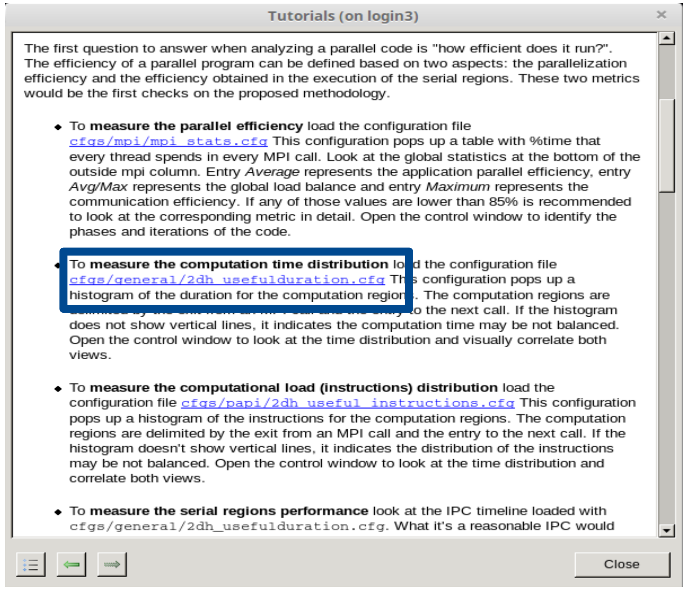
- Click on “2dh_usefulduration.cfg” (2nd link) ➔ Shows **time computing**

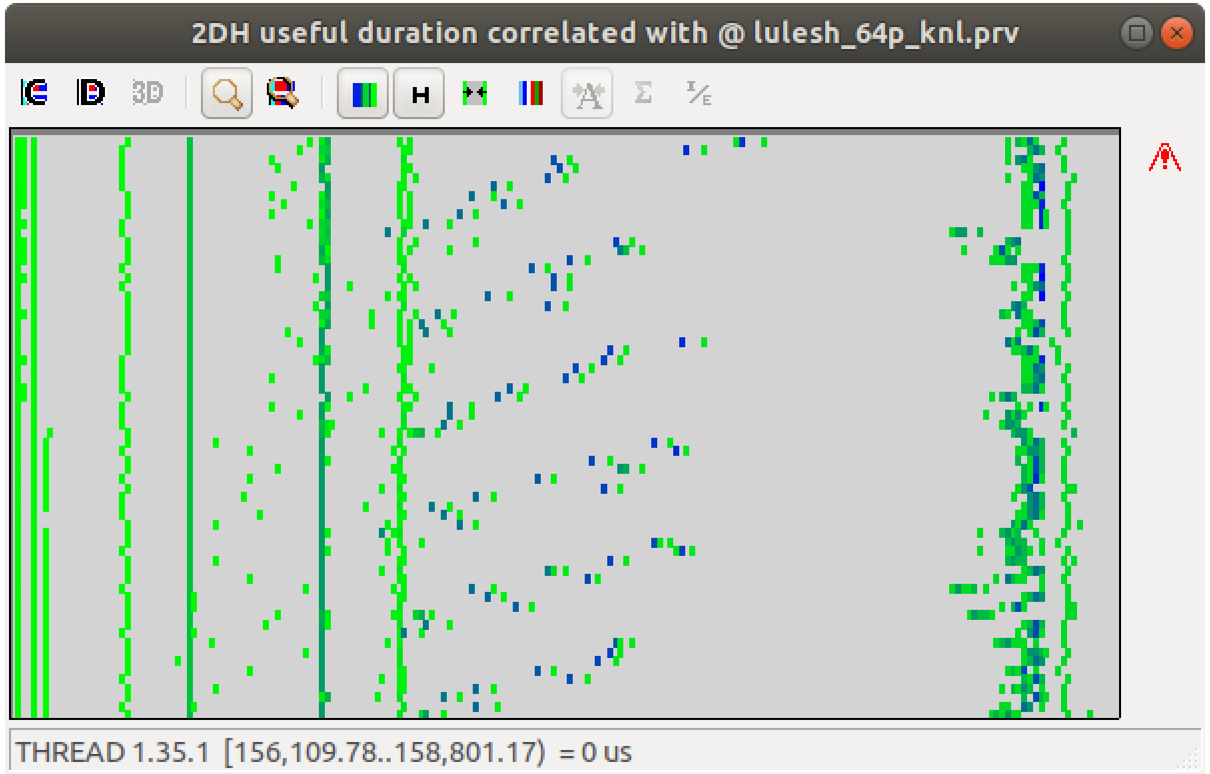




Computation time and work distribution

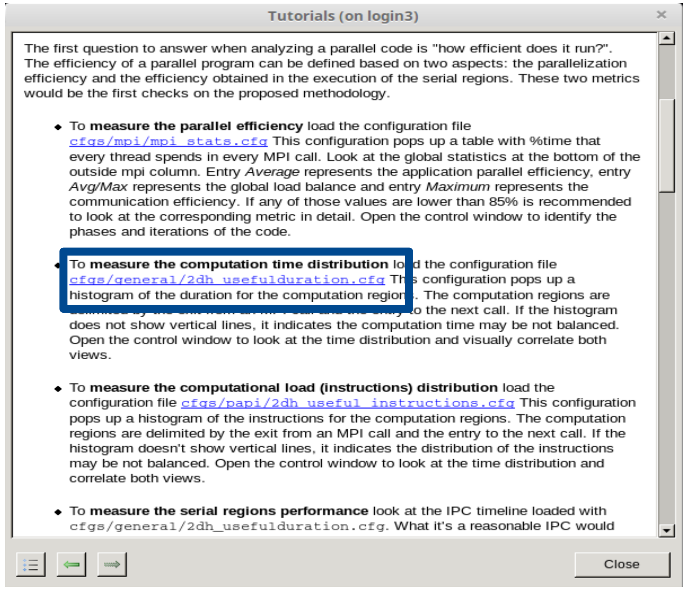
- Click on “2dh_usefulduration.cfg” (2nd link) ➔ Shows **time computing**

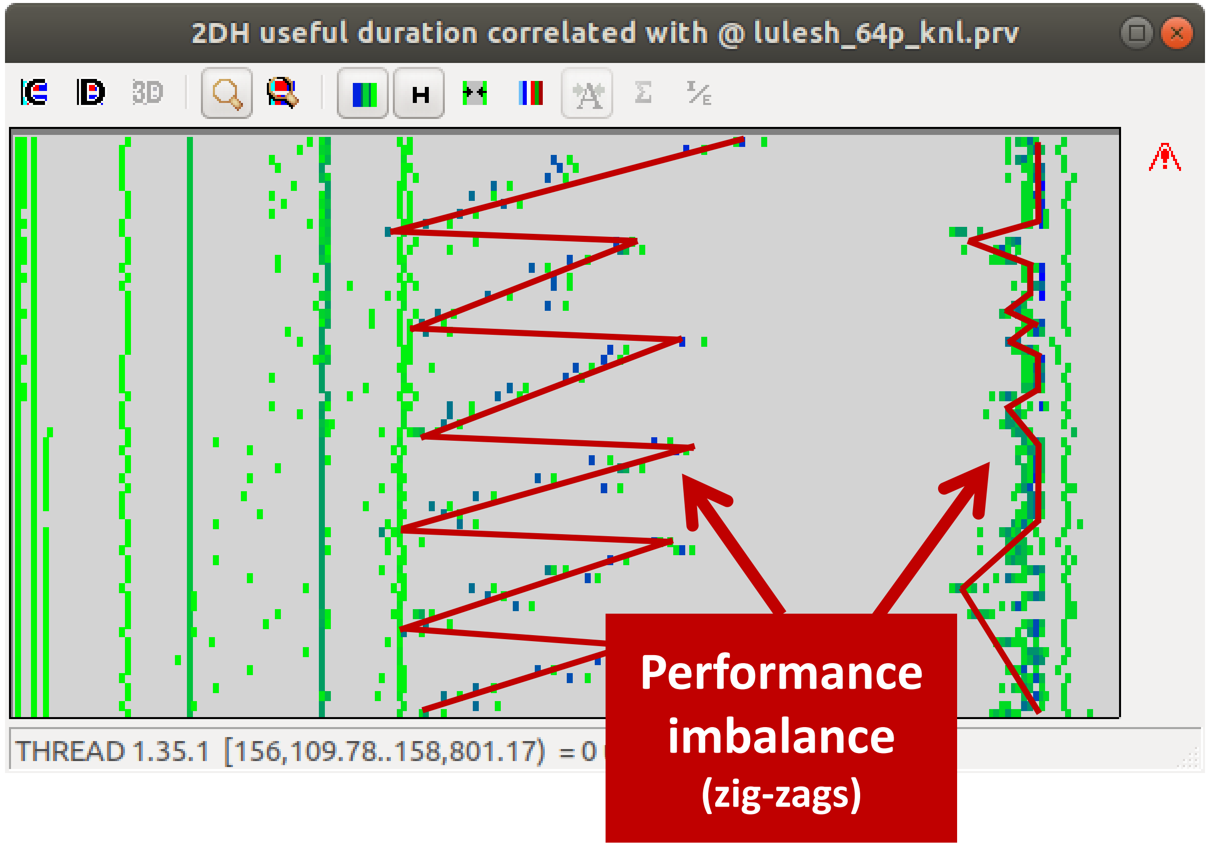




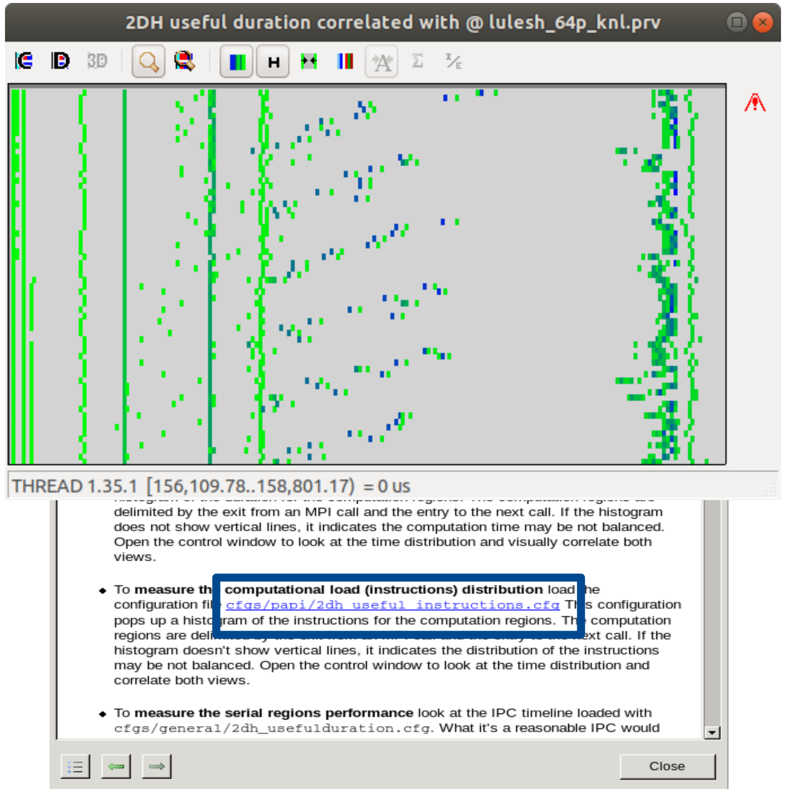
Computation time and work distribution

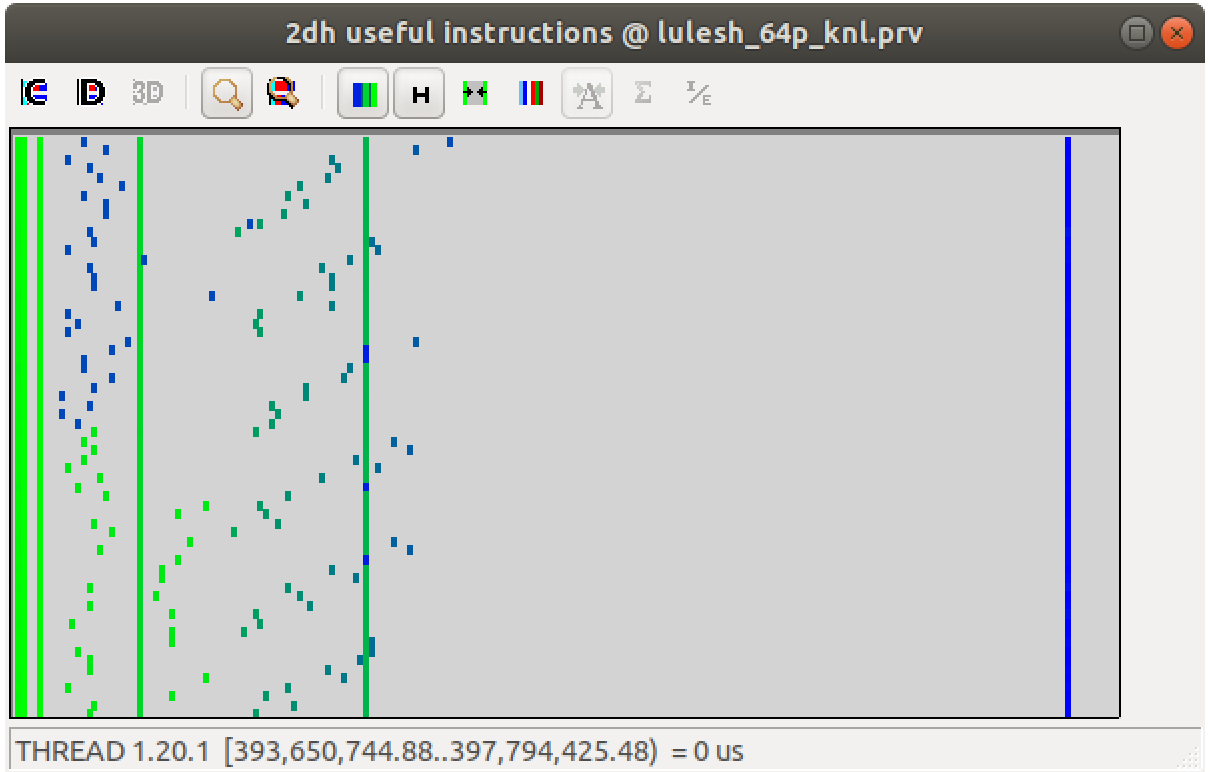
- Click on "2dh_usefulduration.cfg" (2nd link) ➔ Shows **time computing**





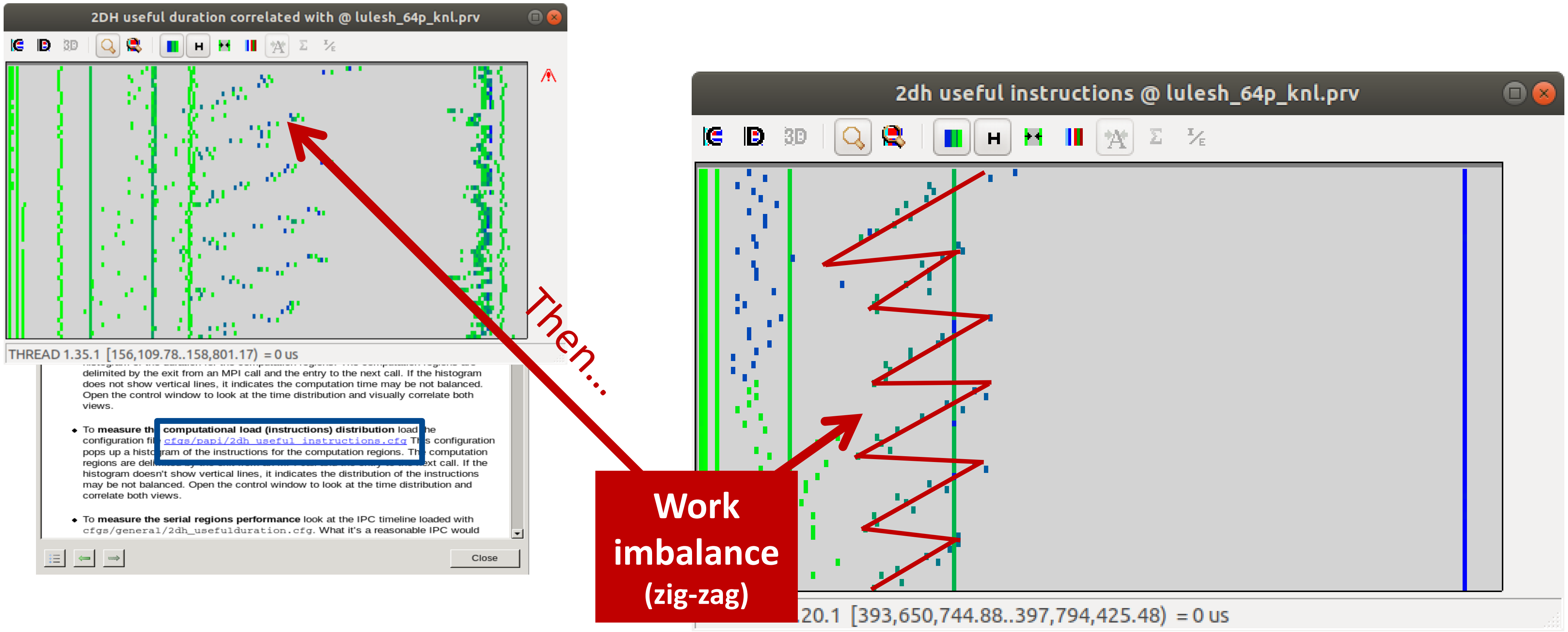
Computation time and work distribution





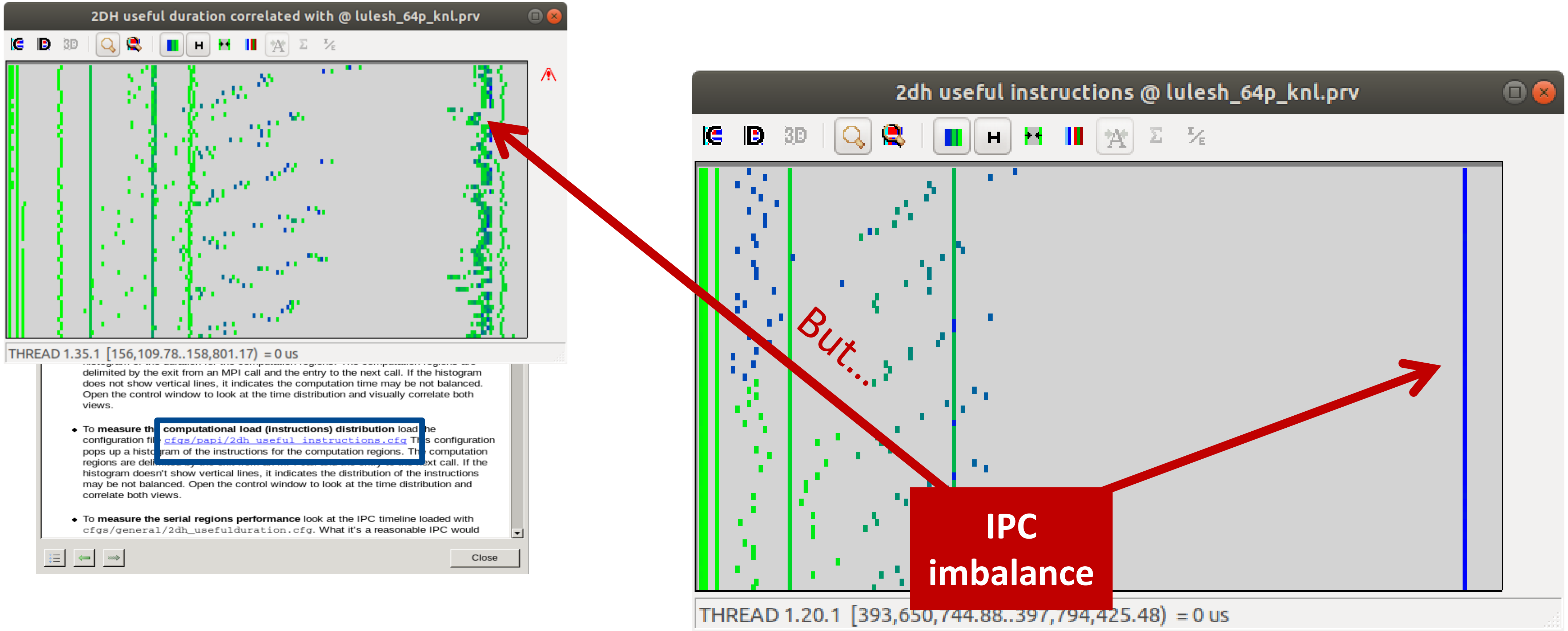
- Click on “2dh_useful_instructions.cfg” (3rd link) ➔ Shows **amount of work**

Computation time and work distribution



- Click on "2dh_useful_instructions.cfg" (3rd link) ➔ Shows **amount of work**

Computation time and work distribution



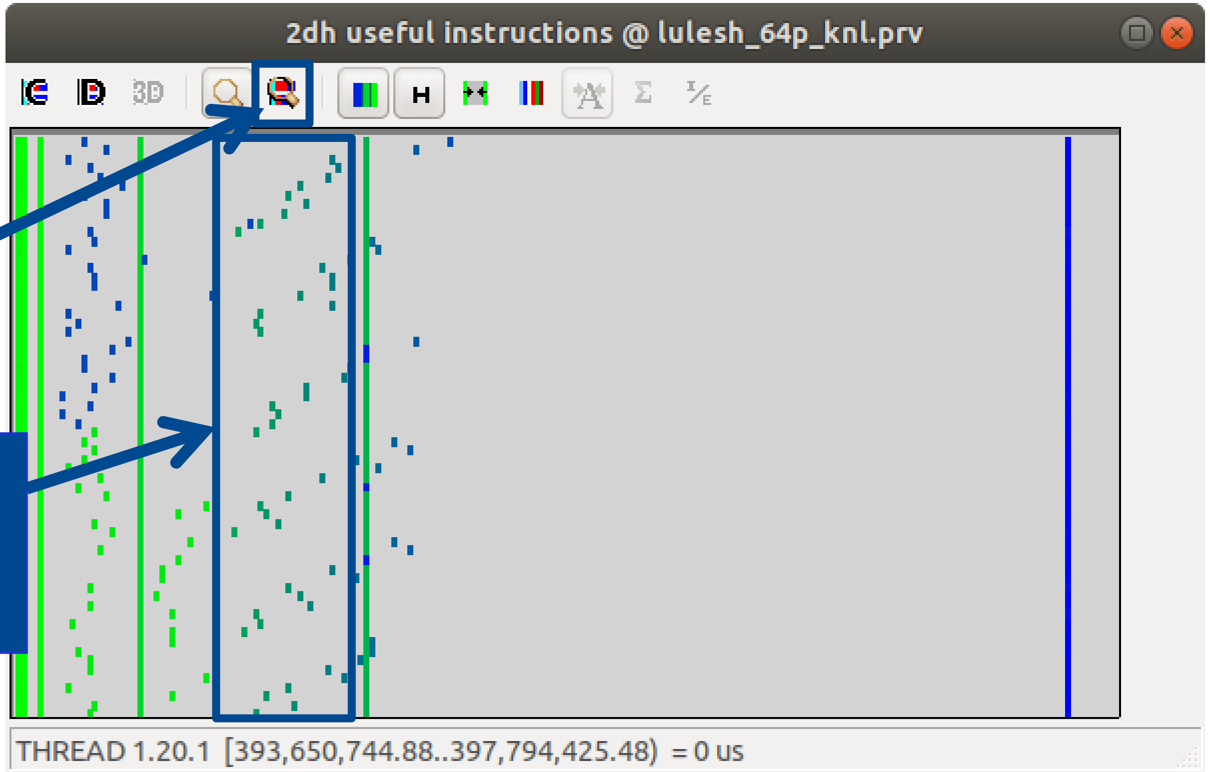
- Click on “2dh_useful_instructions.cfg” (3rd link) ➔ Shows **amount of work**

Where does this happen?

- Go from the table to the timeline

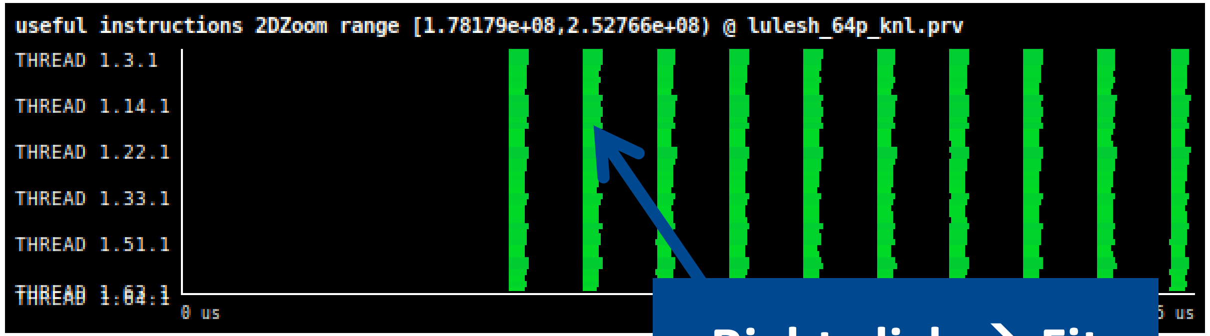
1. Click on "Open Filtered Control Window"

2. Select this area (by drag-and-dropping)



Where does this happen?

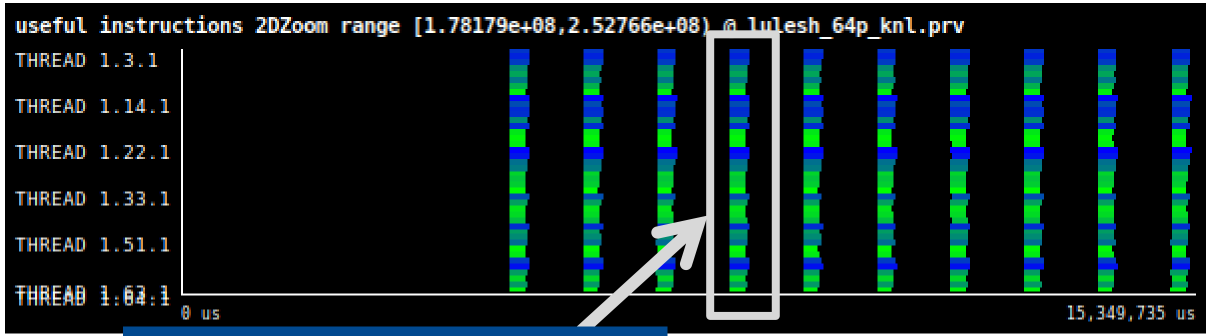
- Go from the table to the timeline



Right click → Fit Semantic Scale → Fit Both

Where does this happen?

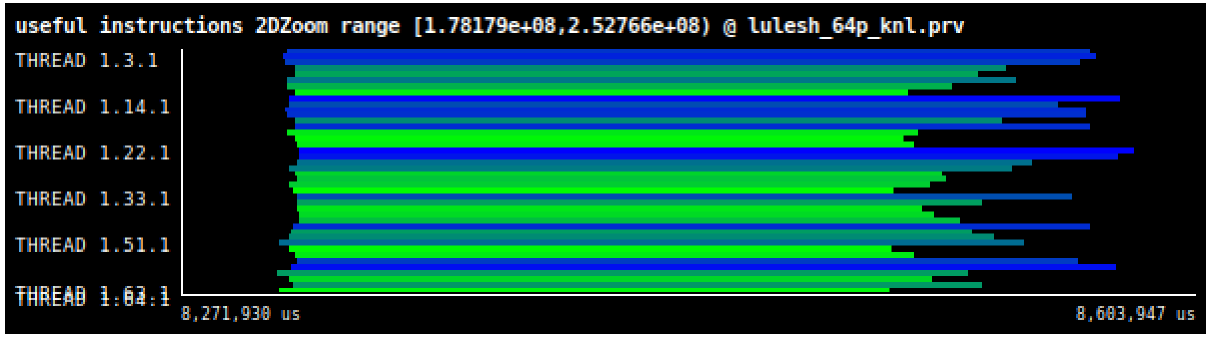
- Go from the table to the timeline



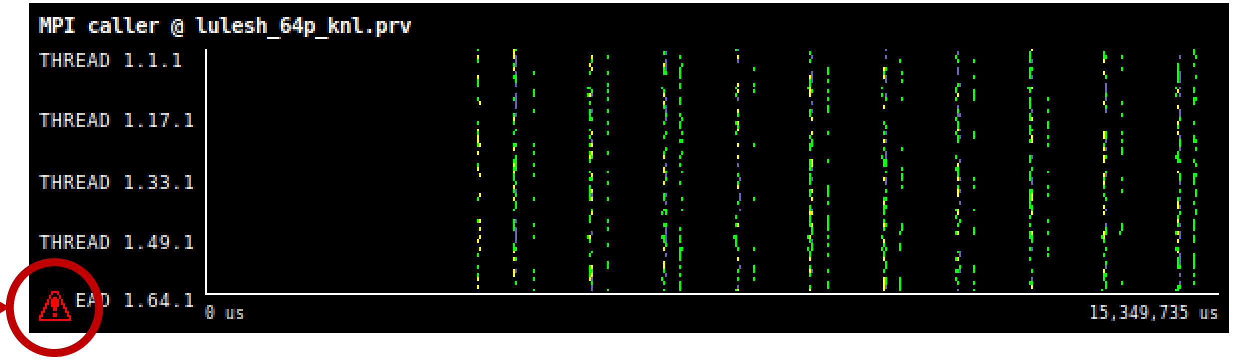
**Zoom into
1 of the iterations
(by drag-and-dropping)**

Where does this happen?

- Slow & Fast at the same time ➔ Imbalance



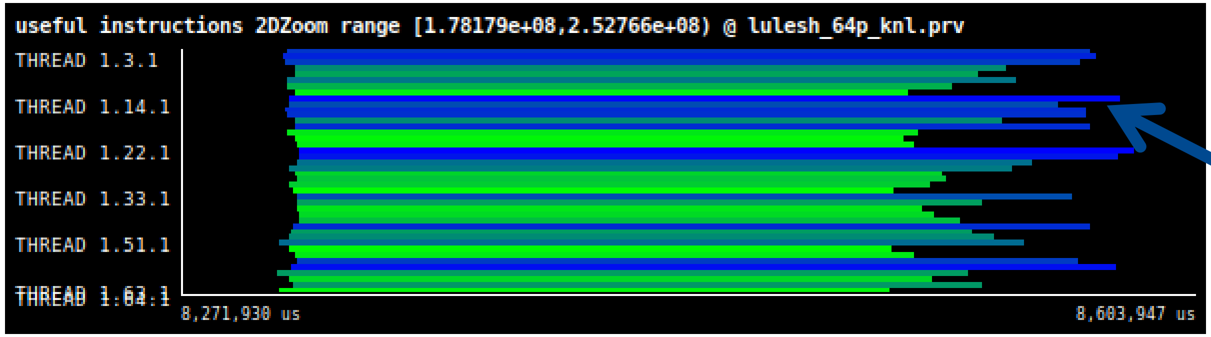
- Hints → Callers → Caller function



Hidden values (click to show)

Where does this happen?

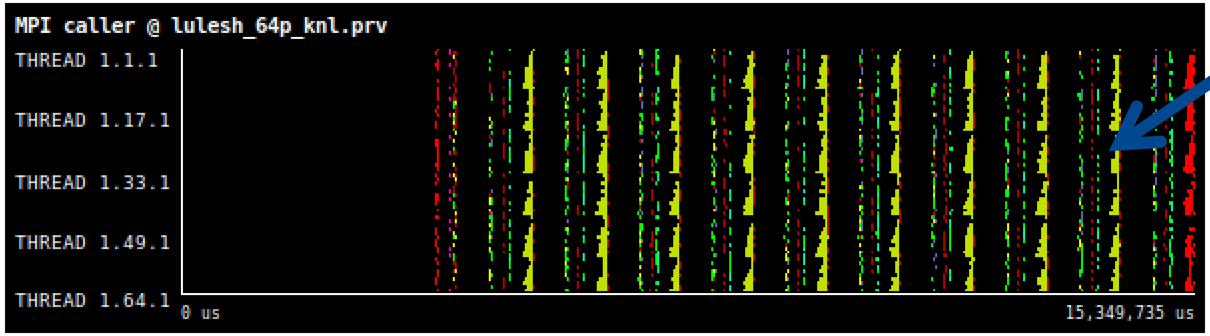
- **Slow** & **Fast** at the same time ➔ Imbalance



Right click → Copy

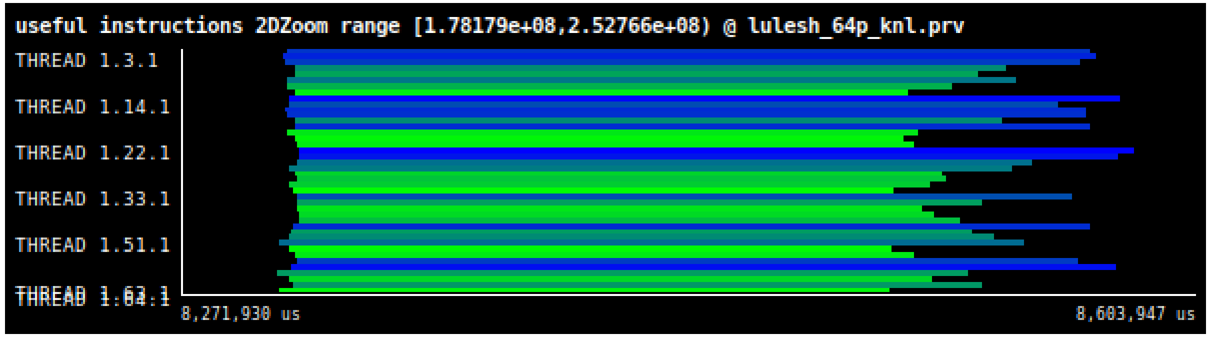
- Hints → Callers → Caller function

Right click → Paste → Time

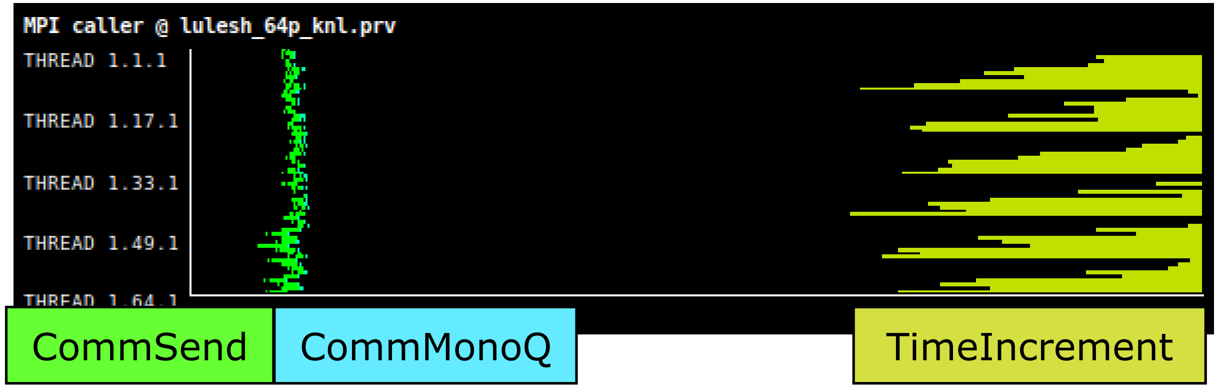


Where does this happen?

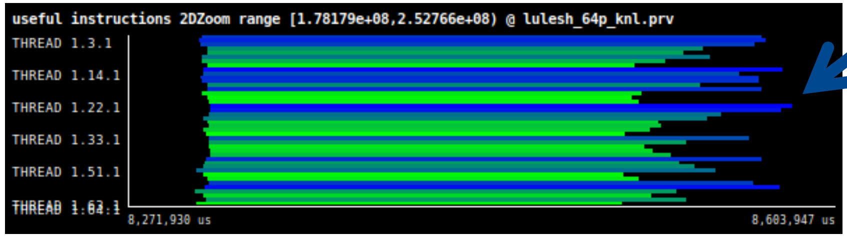
- Slow & Fast at the same time → Imbalance



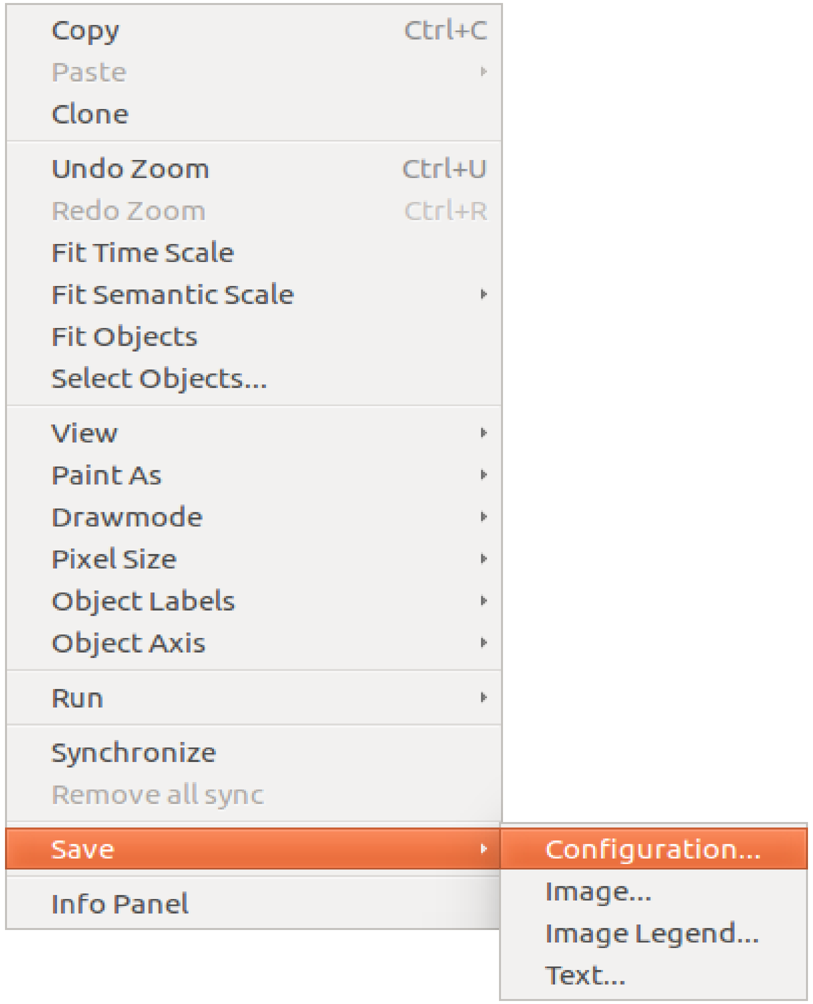
- Hints → Callers → Caller function



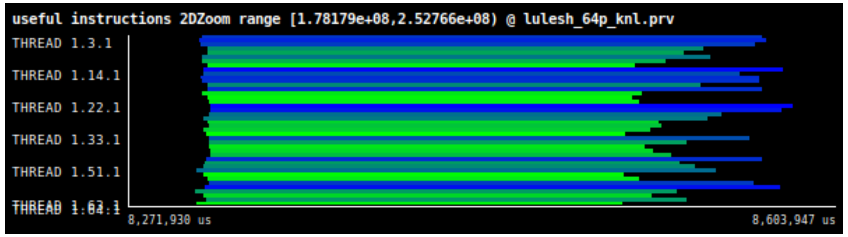
Save CFG’s (2 methods)



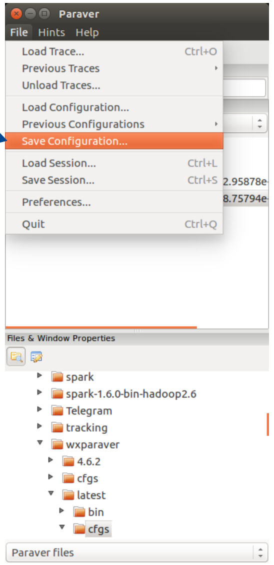
Right click on timeline

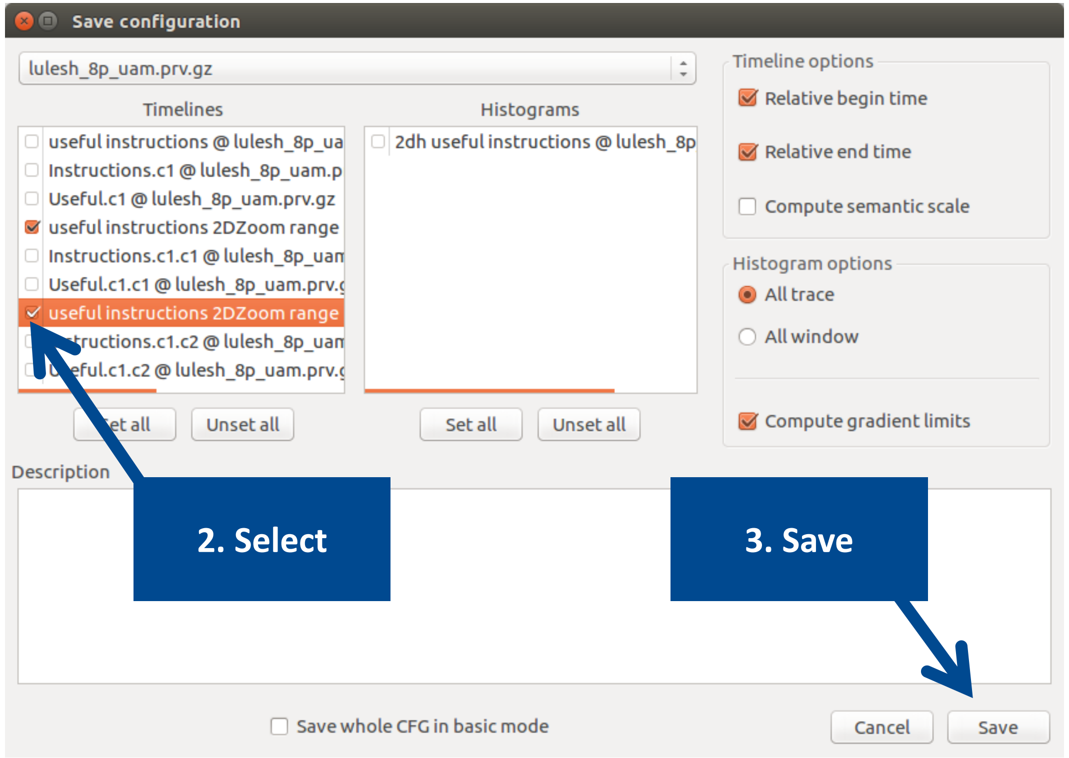


Save CFG's (2 methods)



1. Main Paraver window



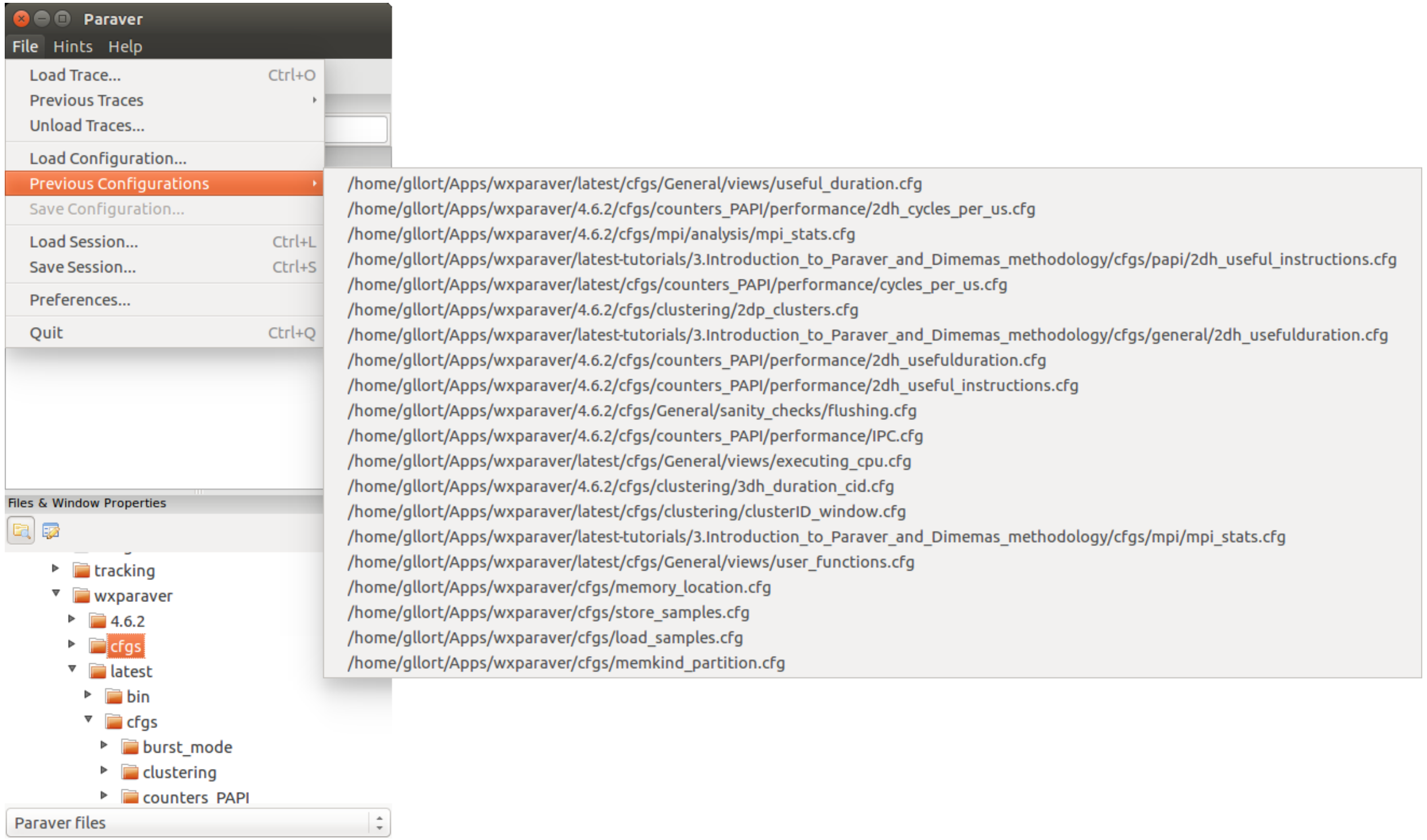


2. Select

3. Save

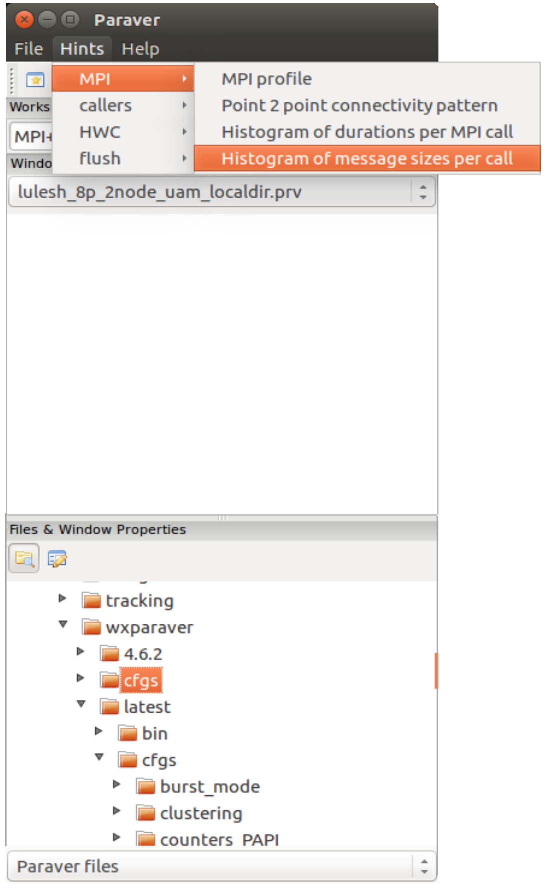
CFG's distribution

- 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

Use clustering analysis

- Run clustering

```
laptop> ssh -Y <USER>@stampede2.tacc.xsede.org

stampede2> cd $WORK/tools-material/clustering

stampede2> ~tg856590/tools/clustering/bin/BurstClustering
        -d cluster.xml
        -i ../extrae/lulesh_64p_knl.prv
        -o lulesh_64p_knl_clustered.prv
```

- If you didn't get your own trace, use as input (-i) a prepared one from:

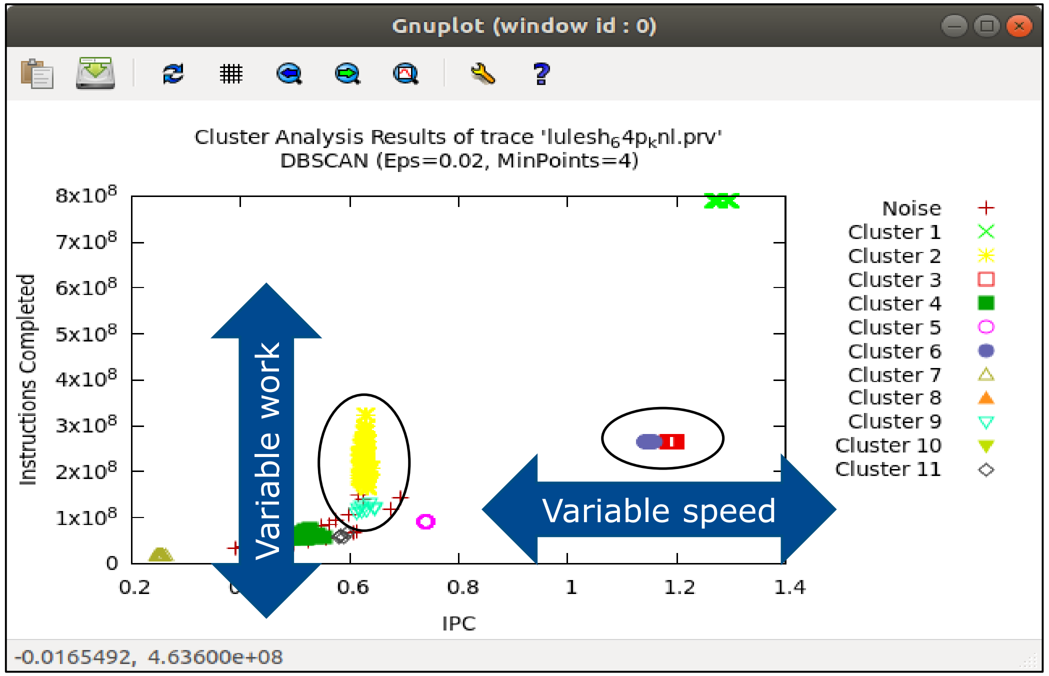
```
stampede2> ls $WORK/tools-material/traces/lulesh_64p_knl.prv
```

Cluster-based analysis

- Check the resulting scatter plot

```
stampede2> gnuplot lulesh_64p_knl_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**



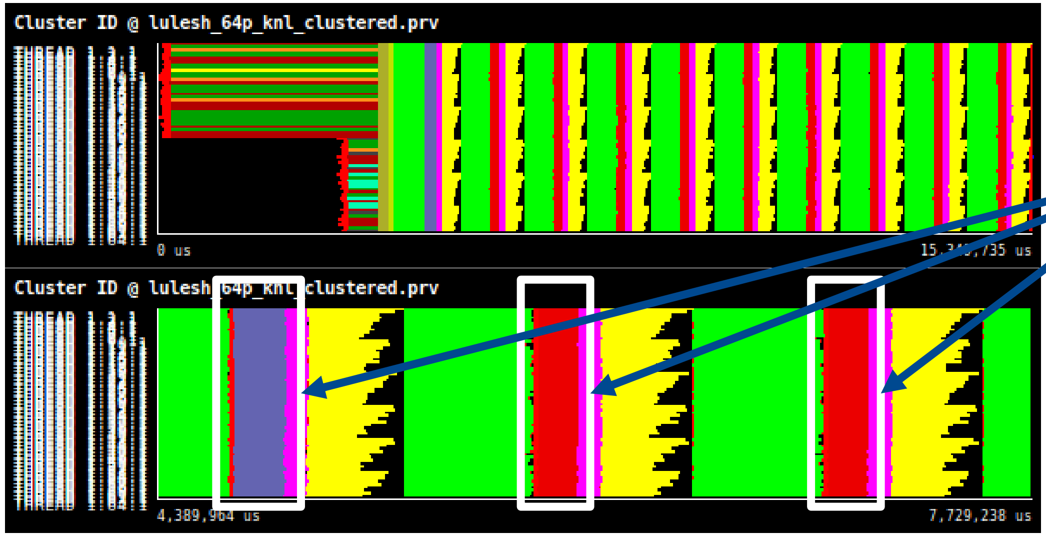
Correlating scatter plot and time distribution

- Copy the clustered trace to your laptop and look at it

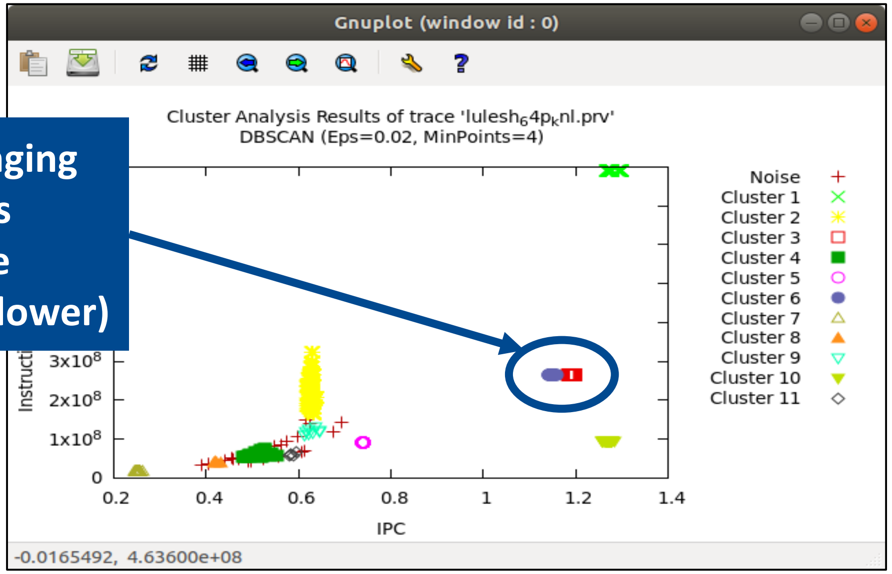
```
laptop> scp <USER>@stampede2.tacc.xsede.org:$WORK/tools-material/clustering/lulesh_64p_knl_clustered.* .

laptop> $HOME/paraver/bin/wxparaver lulesh_64p_knl_clustered.prv
```

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



Identify changing behaviors over time (1st iteration slower)



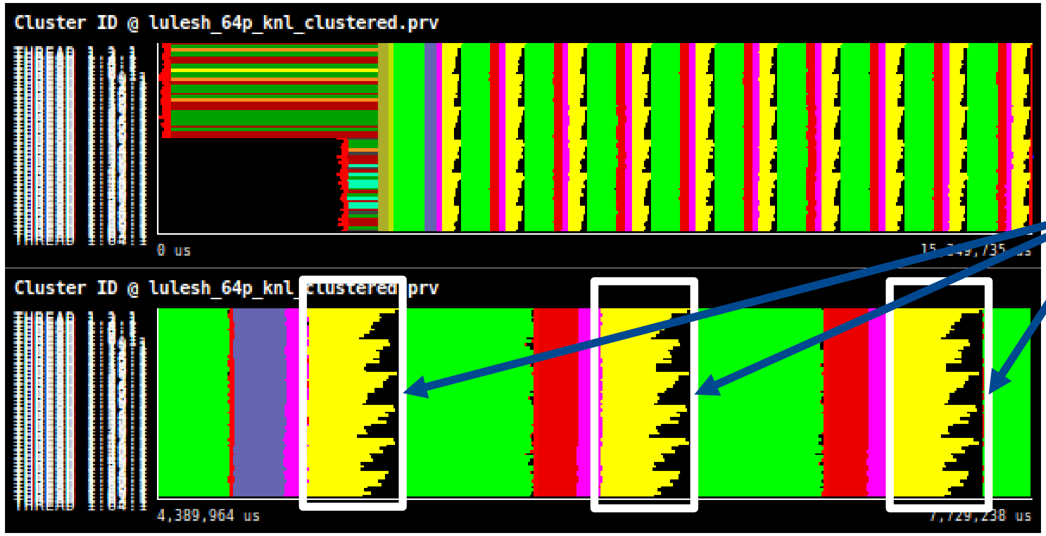
Correlating scatter plot and time distribution

- Copy the clustered trace to your laptop and look at it

```
laptop> scp <USER>@stampede2.tacc.xsede.org:$WORK/tools-material/clustering/lulesh_64p_knl_clustered.* .

laptop> $HOME/paraver/bin/wxparaver lulesh_64p_knl_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