Barcelona Supercomputing Center
Centro Nacional de Supercomputación

EXCELENCIA SEVERO OCHOA

esiwace
CENTRE OF EXCELLENCE IN SIMULATION OF WEATHER AND CLIMATE IN EUROPE

Computational Profiling Analysis for Climate and Weather

Mario C. Acosta and Xavier Yepes

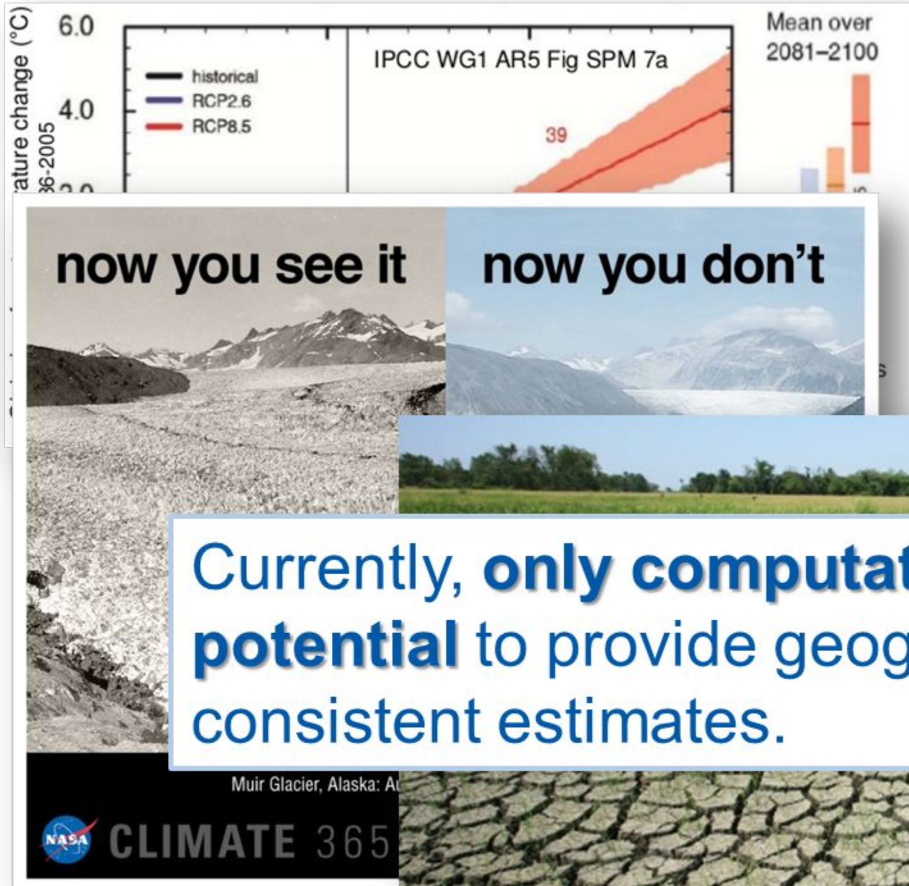
27/08/2020

Summer School on Effective HPC

Computational Profiling Analysis for Climate and Weather

- Objectives
  - Define performance analysis fundamentals (objectives, methods, metrics, hardware counters, etc.)
  - Define a methodology to study HPC performance for numerical models, know your enemy.
  - Describe the BSC performance analysis tools suite (Extrae, Paraver, Dimemas)
  - Interpret uses cases from Earth System Models (HARMONIE, IFS, NEMO, etc.) that illustrate how to identify and solve performance issues
  - Apply profiling techniques to identify performance bottlenecks in your code
  - Summarise typical performance problems
  - Discuss specific knowledge about performance analysis applied to earth system modelling

Introduction



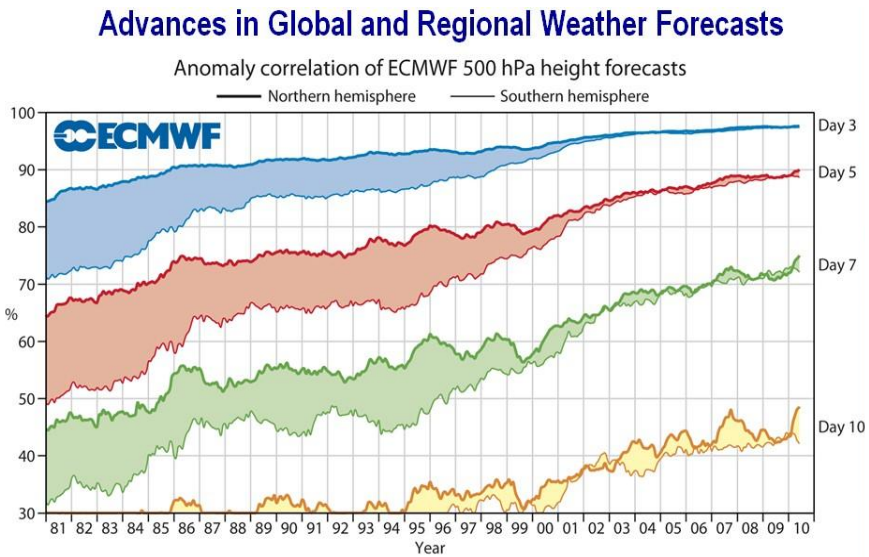
- Projections
- Impact analysis
- Adaptation to climate change.

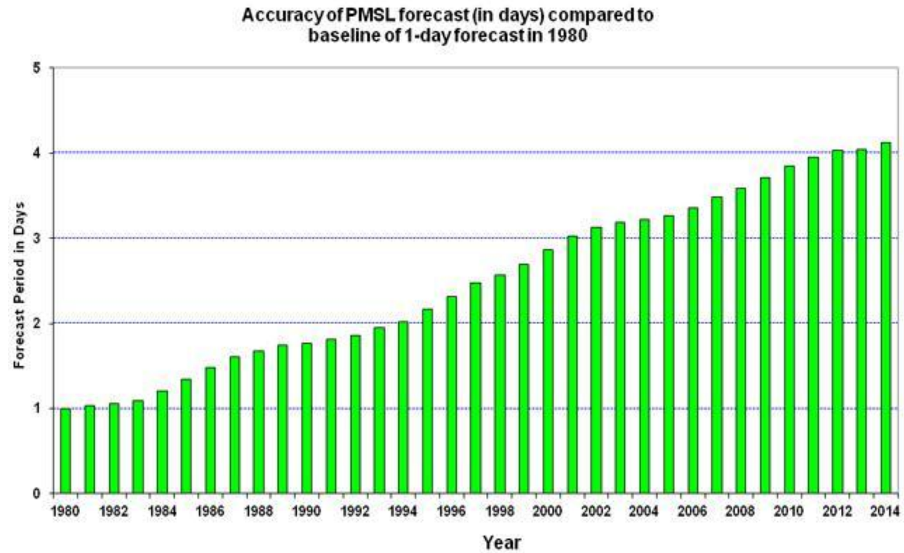
Currently, **only computational models have the potential** to provide geographically and physically consistent estimates.

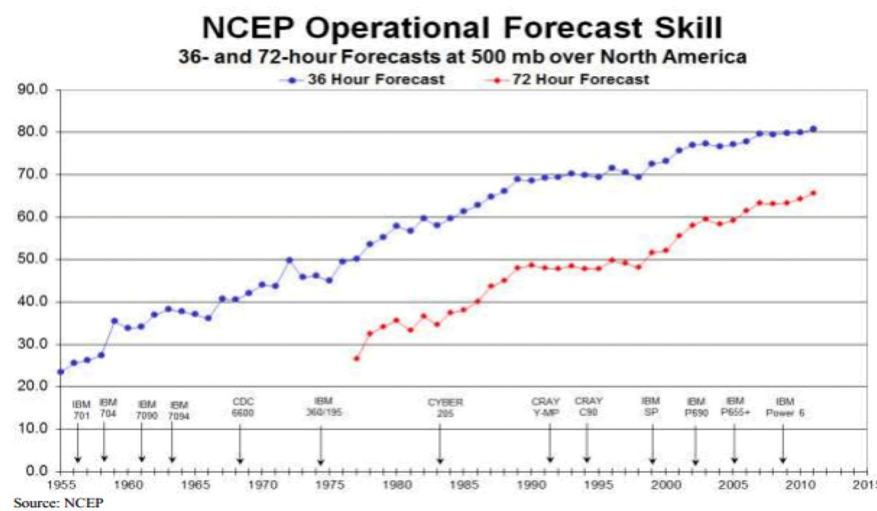
Barcelona Supercomputing Center
Centro Nacional de Supercomputación

esiwace
CENTRE OF EXCELLENCE IN SIMULATION OF WEATHER AND CLIMATE IN EUROPE

Introduction



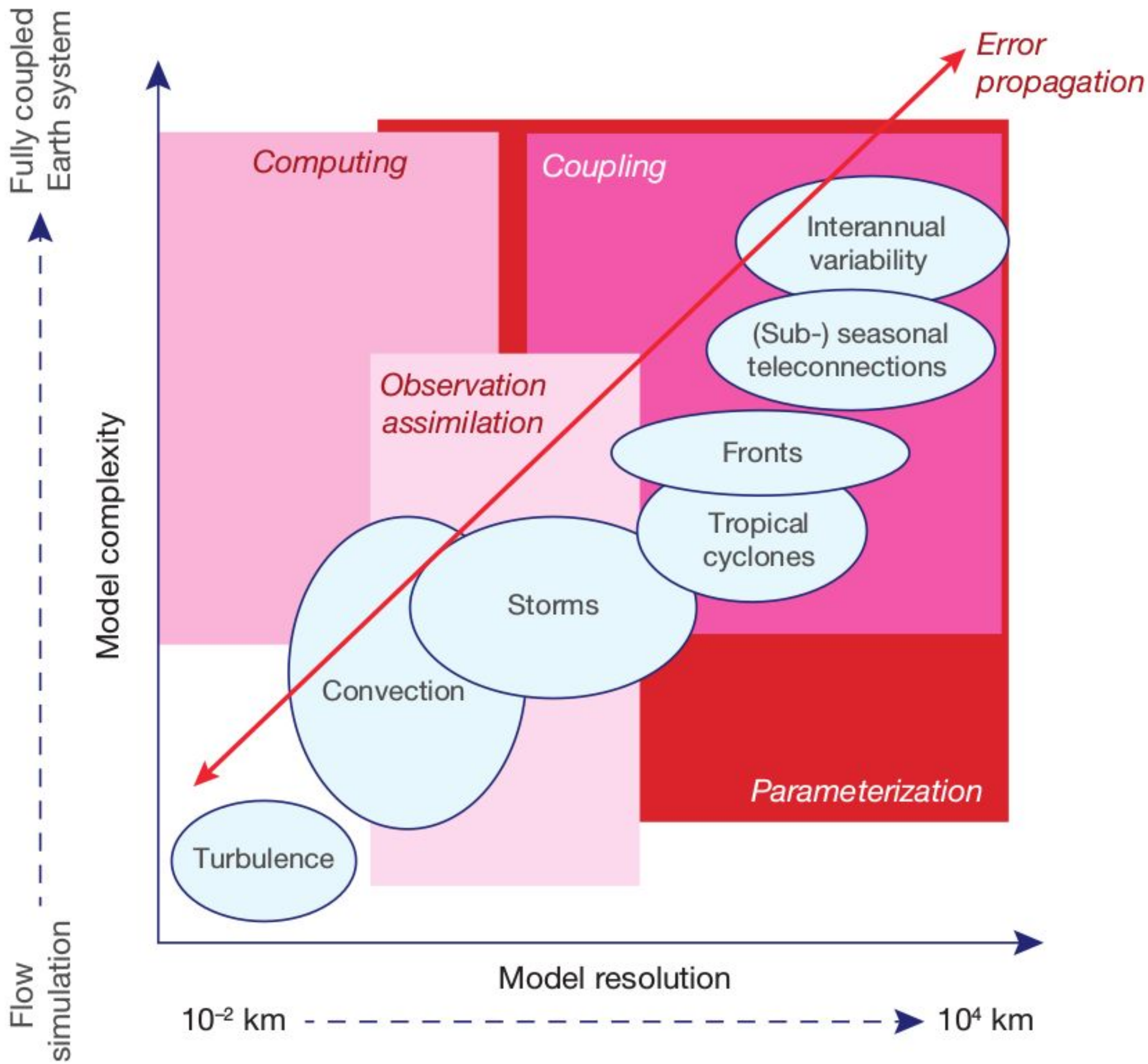




Barcelona Supercomputing Center
Centro Nacional de Supercomputación

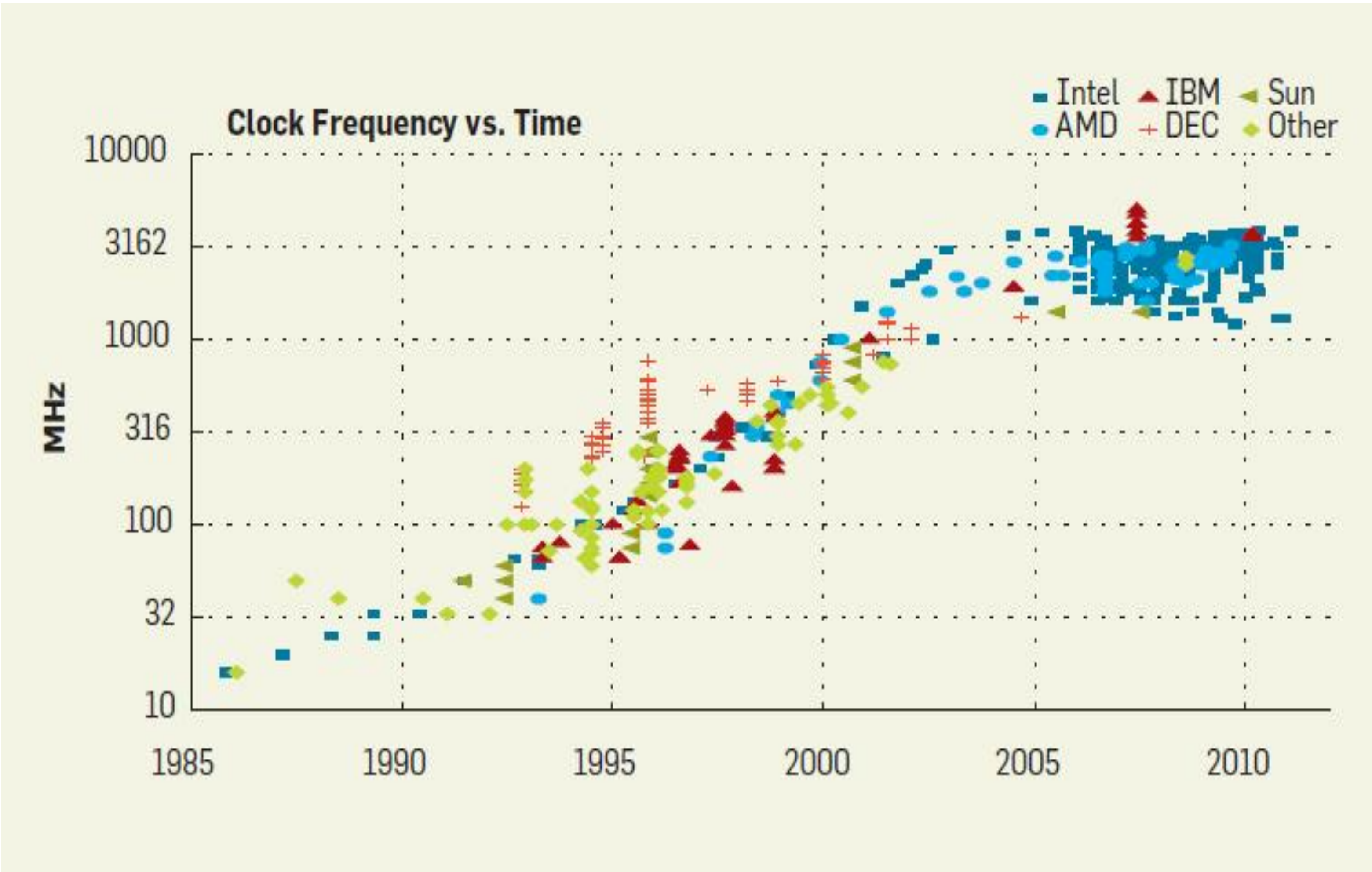
esiwace
CENTRE OF EXCELLENCE IN SIMULATION OF WEATHER AND CLIMATE IN EUROPE

Introduction



Introduction

Introduction



Barcelona Supercomputing Center
Centro Nacional de Supercomputación

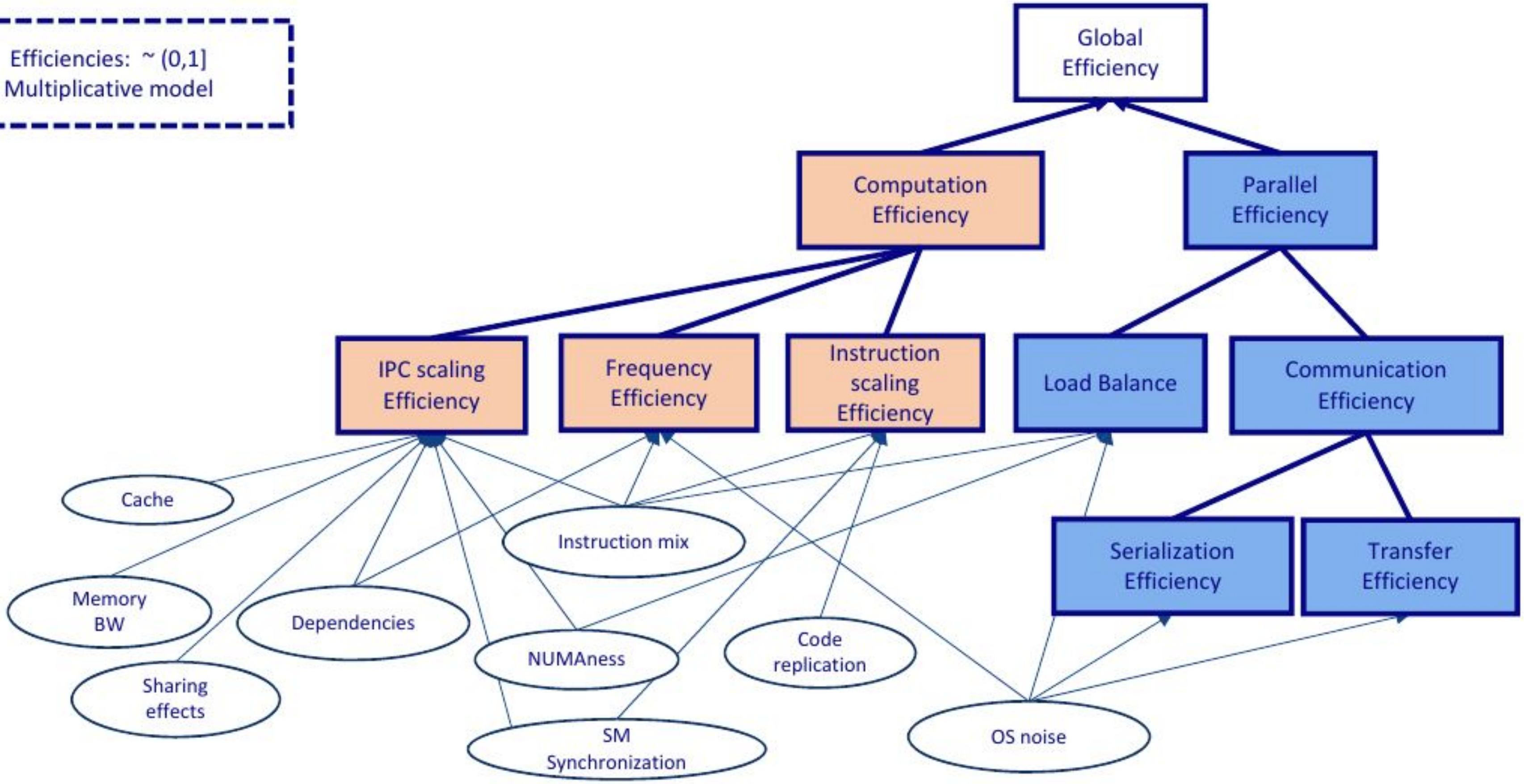
esiwace
CENTRE OF EXCELLENCE IN SIMULATION OF WEATHER AND CLIMATE IN EUROPE

Introduction

- To be able to use the computing power of modern supercomputers, applications must exploit parallelism.
- Parallelism produce overhead (extra computation and communications)
  - "*Overhead does not look a problem in my model*" → But if the needs increase (i.e. higher resolutions), a bad implementation will be a problem in some point.
  - We need a method to evaluate the parallelism efficiency of our computational models.
    - When the hardware change
    - When the number of resources change
    - When the model complexity change
    - When the resolution change
    - …

Introduction

Efficiencies: ~ (0,1]
Multiplicative model



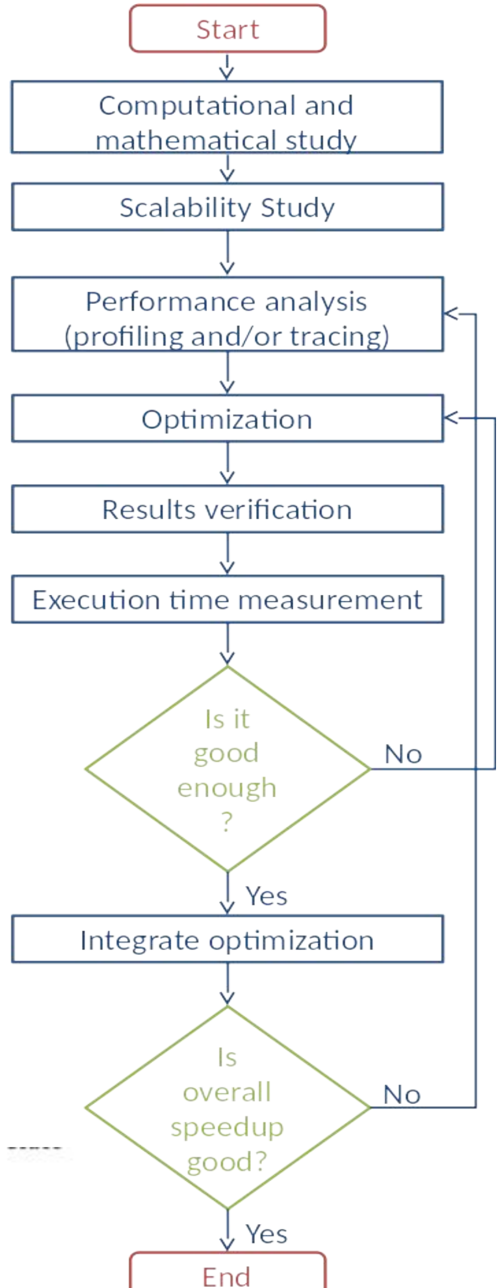
Introduction

- The necessary refactoring of numerical codes is given a lot of attention and is stirring a number of discussions.
  - Computational performance analysis and new optimizations are needed for actual numerical models.
  - Study new algorithms for the new generation of high performance platforms (path to exascale).
- Several European institutions and projects working together on the same direction (ESCAPE2, ESiWACE2, IS-ENES3, ETP4HPC…)

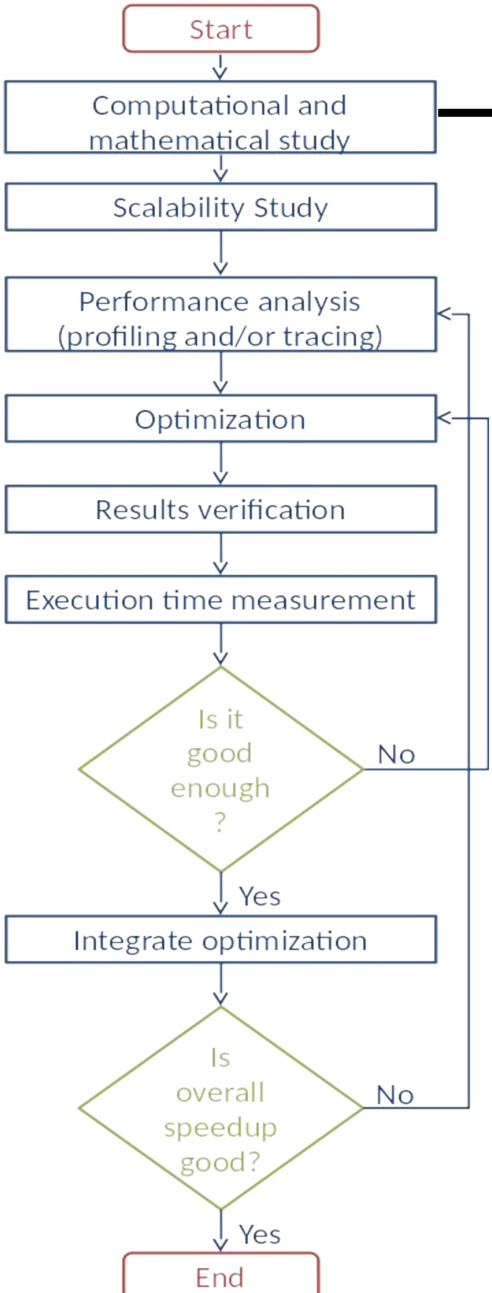
CES-Performance Team & ES Department

- Knowledge about the mathematical and computational side of Earth System Applications
- Knowledge about the specific needs in HPC of the Earth System Applications
- Researching about HPC methods specifically used for Earth System Applications

Methodology

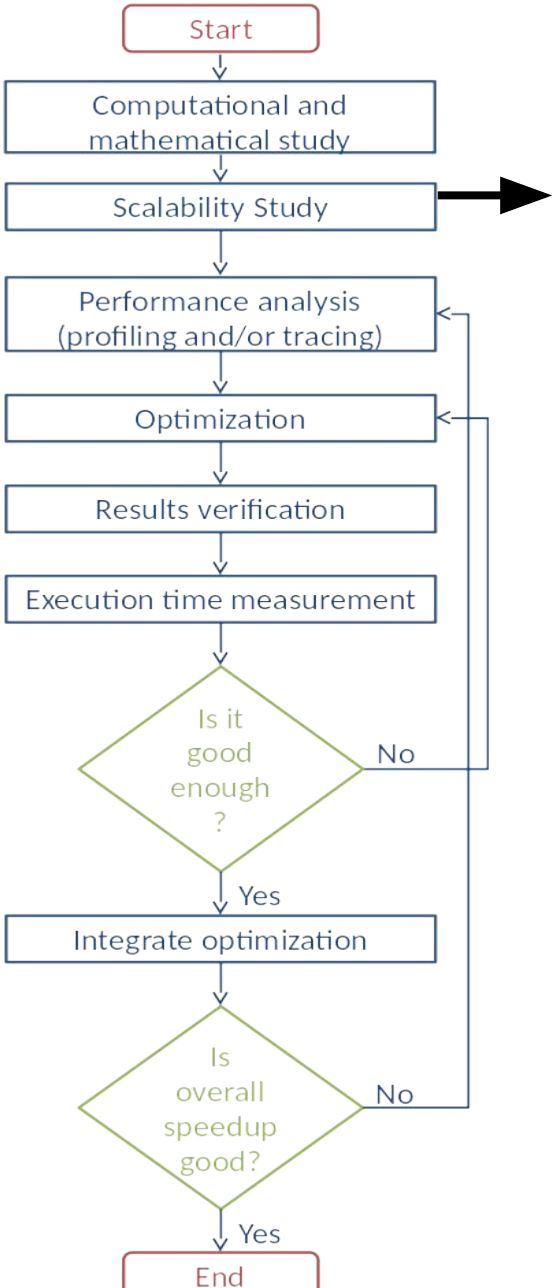


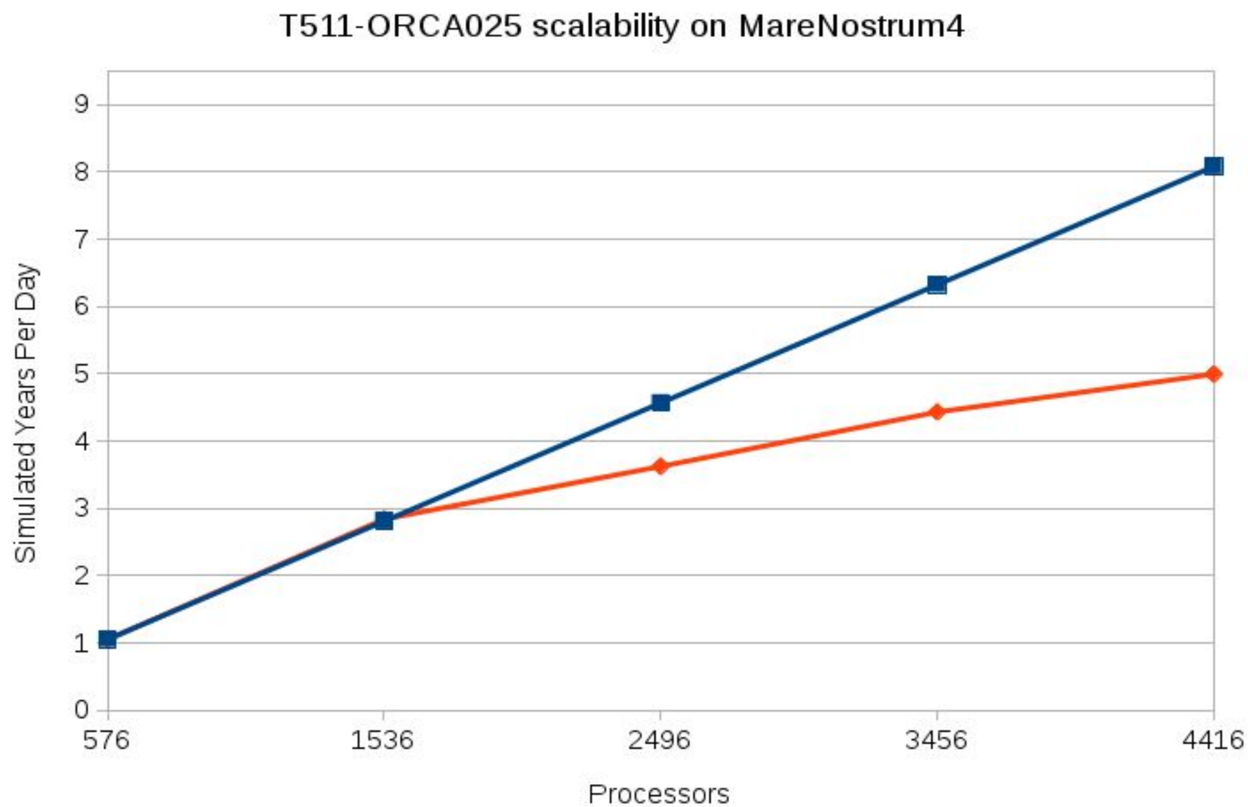
Methodology

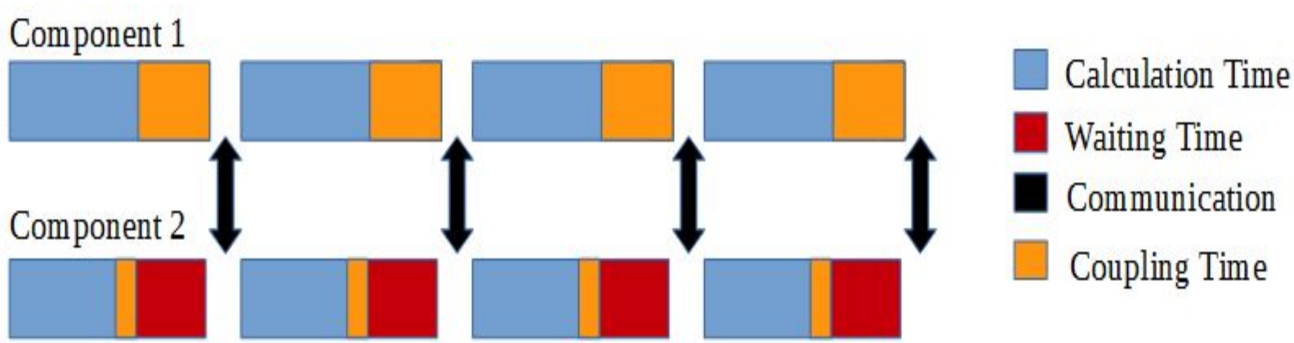


- Mathematical study
  - Some methods could be better than others
    - Discretization used (explicit, implicit, semi-implicit…)
    - Parallel adaptation (solvers, preconditioners…)
  - How to implement new algorithms for new architectures
- Computational study
  - Achieve load balance among components
  - Reduce overhead introduced by parallel applications
  - Assure that the computational algorithm takes advantage of the architecture

Methodology

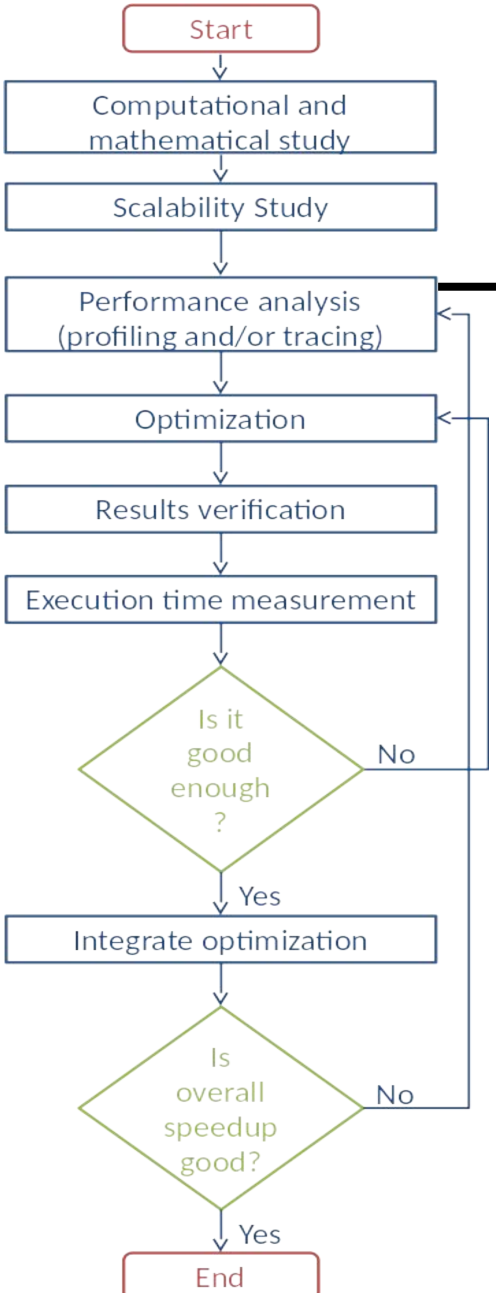




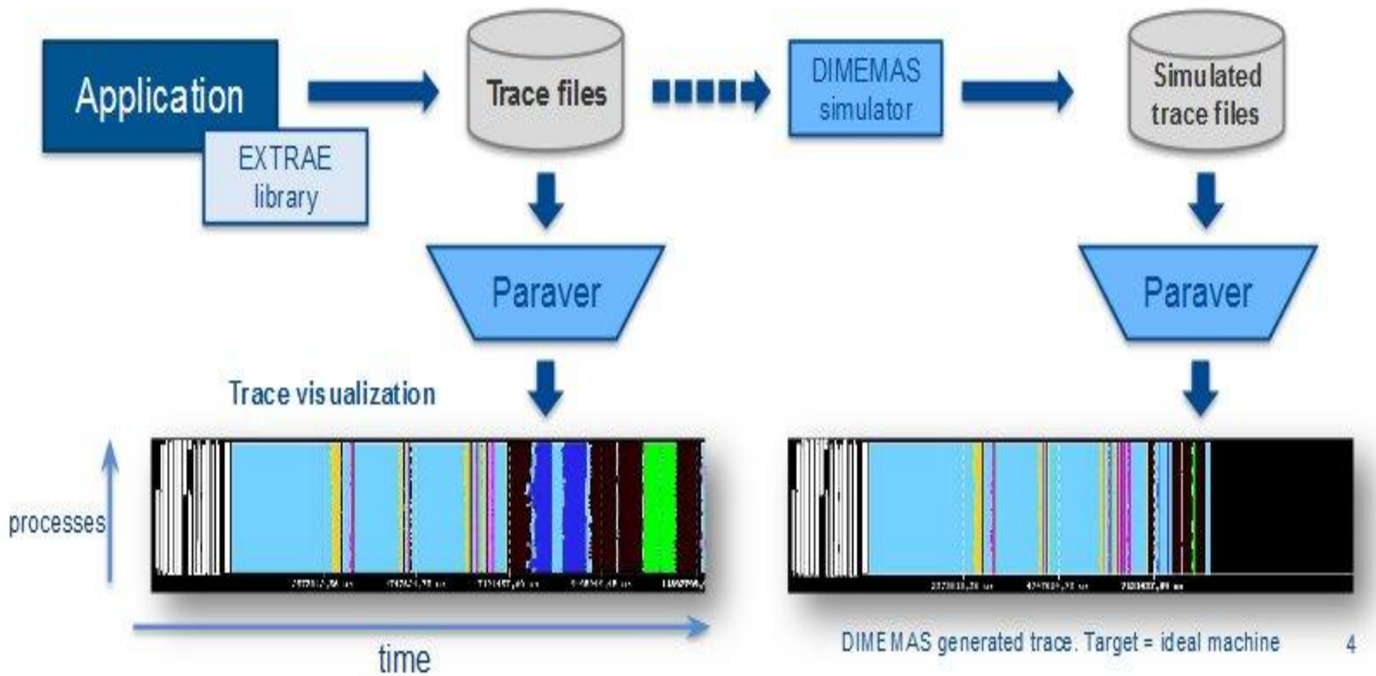


Possible load balance of coupled components of a Earth System Model

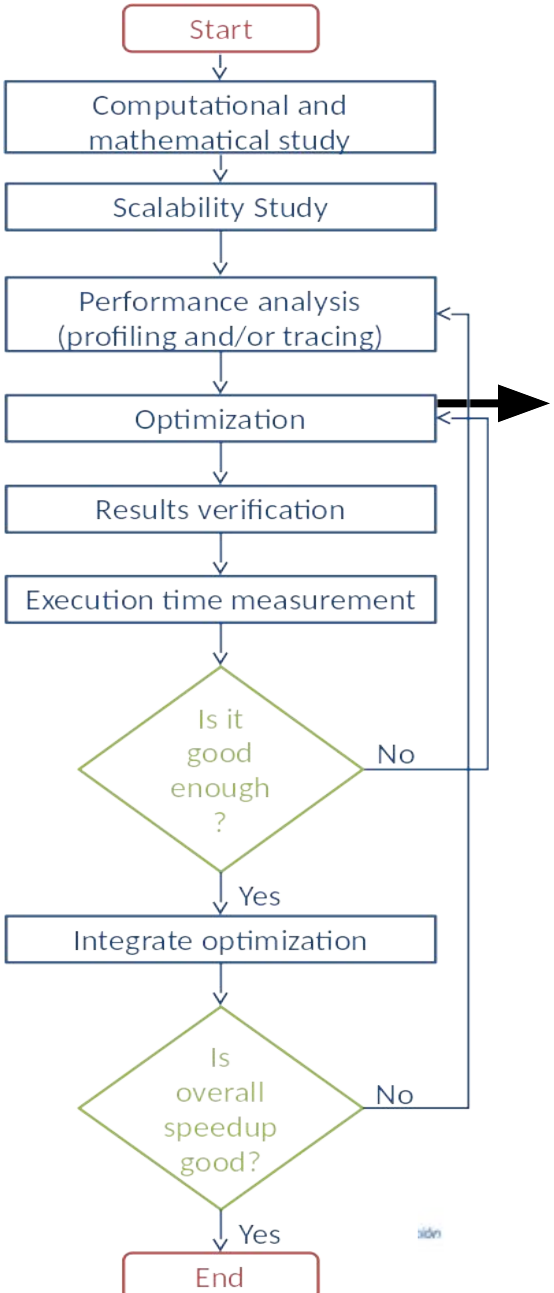
Methodology



- Since 1991
- Based on traces
- Open Source: http://www.bsc.es/paraver
- **Extrae**: Package that generates Paraver trace-files for a post-mortem analysis
- **Paraver**: Trace visualization and analysis browser
  - Includes trace manipulation: Filter, cut traces
- **Dimemas**: Message passing simulator



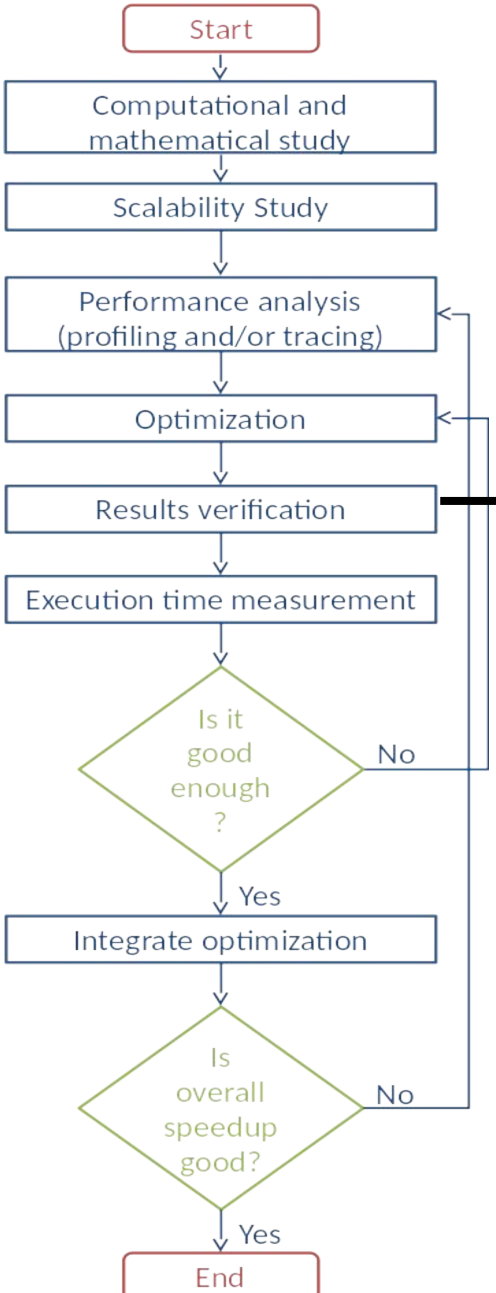
Methodology



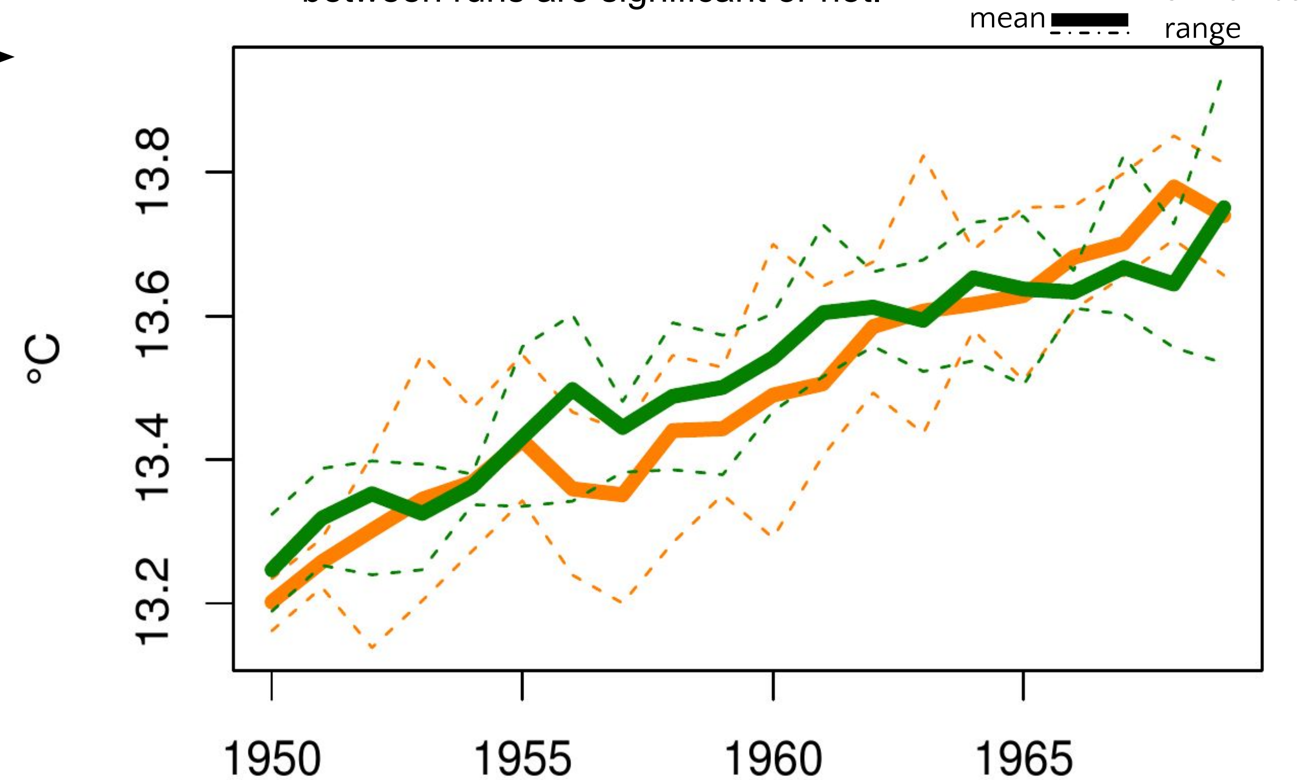
- Introducing optimizations
  - Improvement of the mathematical and/or computational algorithm
    - Apply scientific methods which are found in the literature
    - Improve the method with a new approach
  - Revolution: Create a new (and better) algorithm taking into account the research line followed

esiwace
CENTRE OF EXCELLENCE IN SIMULATION OF WEATHER AND CLIMATE IN EUROPE

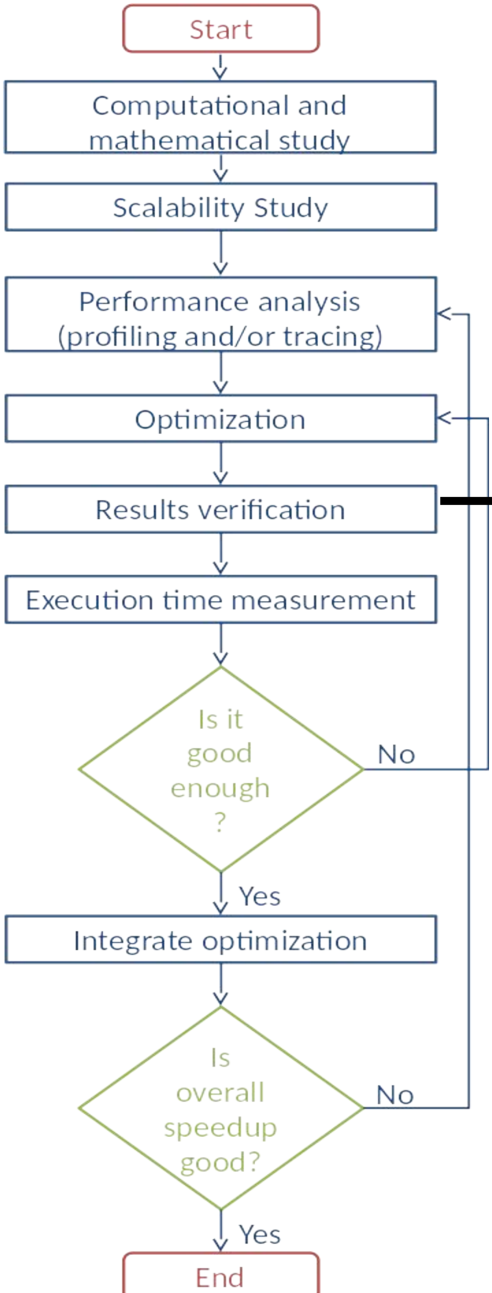
Methodology



- Reproducibility study
  - Evaluate if the accuracy and reproducibility of the model is similar using or not the optimizations proposed
  - Take into account the nature of climate models
    - How to evaluate, in parallel executions, if the differences between runs are significant or not.

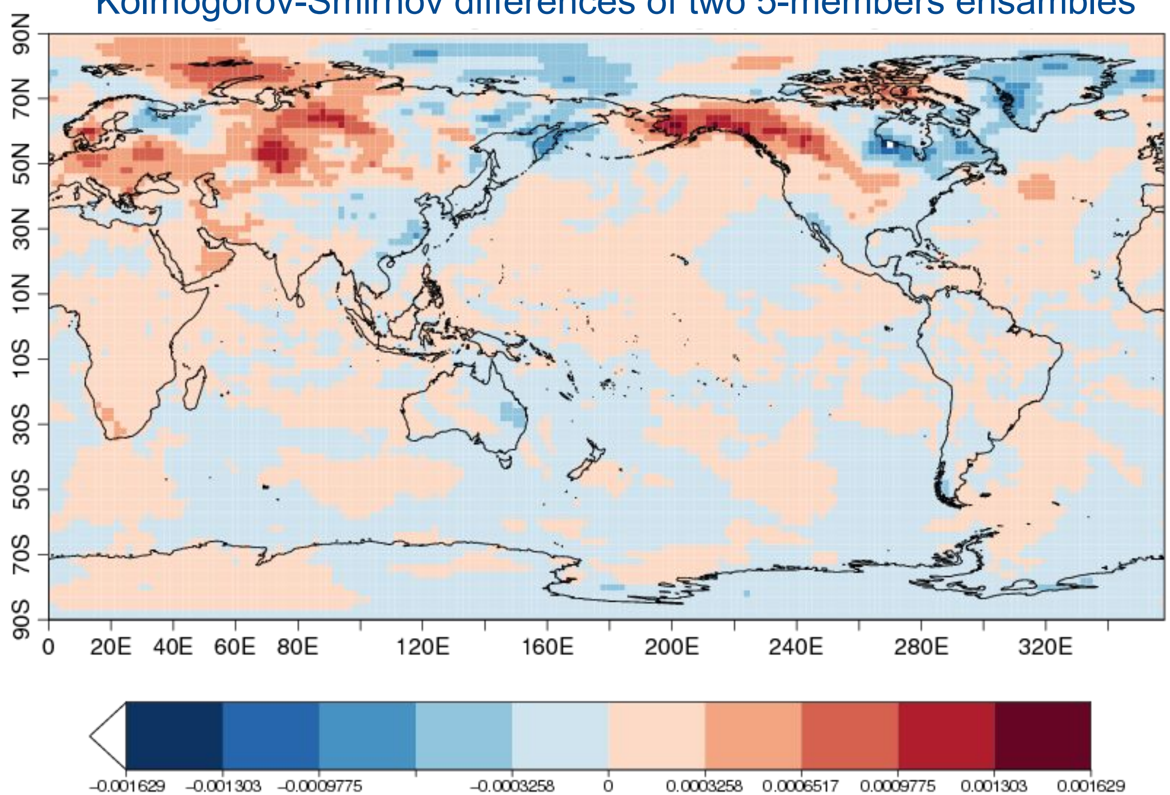


Methodology



- Reproducibility study
  - Evaluate if the accuracy and reproducibility of the model is similar using or not the optimizations proposed
  - Take into account the nature of climate models
    - How to evaluate, in parallel executions, if the differences between runs are significant or not.

Kolmogorov-Smirnov differences of two 5-members ensambles

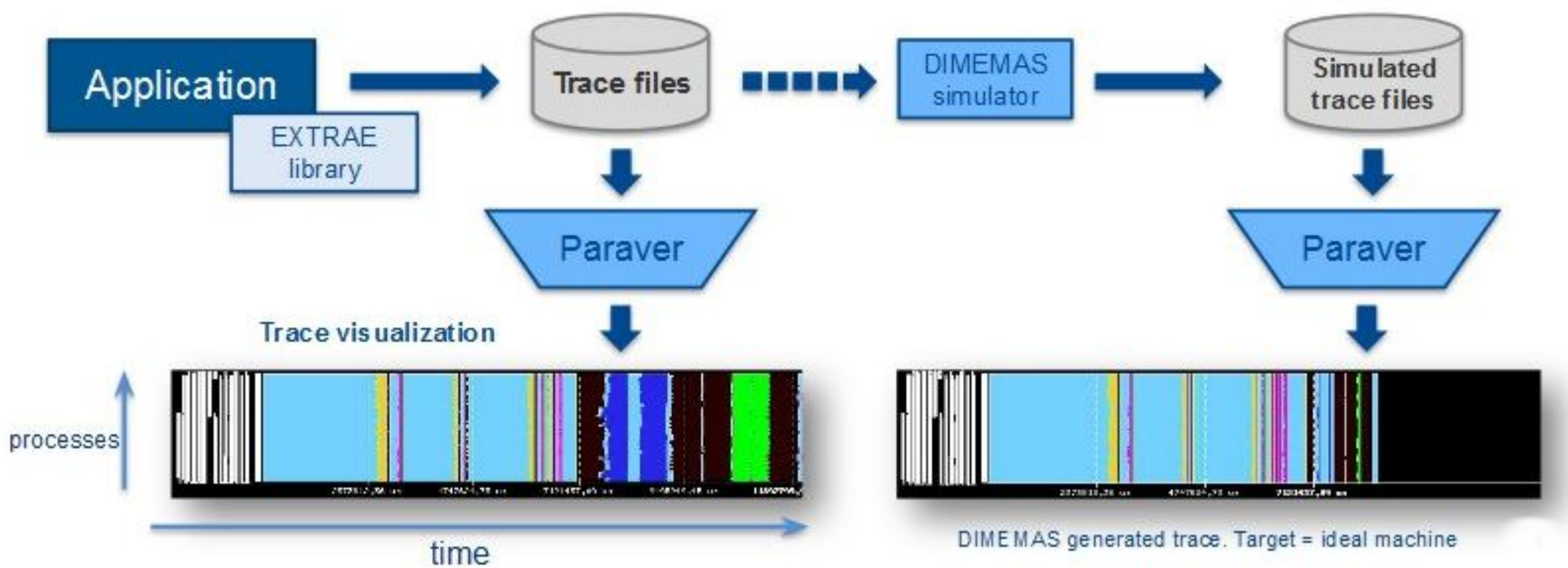


Profiling Analysis: BSC Tools

- BSC Tools
  - General description
  - Extrae
    - General description
    - How to use it
  - Paraver
    - General description
    - How to use it
    - Configurations available
  - Dimemas
    - General description
  - How to work with large traces
    - Filtering/Burst mode
    - Cutting

BSC Tools

- Since 1991
- Based on traces
- Open Source → http://www.bsc.es/paraver
- Extrae: Package that generates Paraver-trace files for a post-morten analysis
- Paraver: Trace visualization and analysis browser
- Dimemas: Message passing simulator
  - Include traces manipulation: Filter, cut traces...



BSC Tools

CORE TOOLS

EXTRAE
Instrumentation framework to generate execution traces of the most used parallel runtimes.

Get EXTRAE

Version 3.4.3 • 2.32 MB

PARAVER
Expressive powerful and flexible trace visualizer for post-mortem trace analysis.

Get PARAVER

Version 4.6.3 • 1.56 MB

DIMEMAS
High-abstracted network simulator for message-passing programs.

Get DIMEMAS

Version 5.3.0 • 0.93 MB

PERFORMANCE ANALYTICS

CLUSTERING
Automatically expose the main performance trends in applications' computation structure.

Get CLUSTERING

Version 2.6.6 • 1.97 MB

TRACKING
Analyze how the behavior of a parallel application evolves through different scenarios.

Get TRACKING

Version 2.6.5 • 1.88 MB

FOLDING
Combined instrumentation and sampling for instantaneous metric evolution with low overhead.

Get FOLDING

Version 1.3.1 • 12.67 MB

SPECTRAL
Signal processing techniques to select representative regions from Paraver traces.

Get SPECTRAL

Version 3.4.0 • 0.3 MB

BASIC ANALYSIS
Framework for automatic extraction of fundamental factors for Paraver traces.

Get BASIC ANALYSIS

Version 0.2 • 66.41 MB

https://tools.bsc.es/downloads

BSC Tools

Home » Documentation » Tutorial Guidelines

These six tutorials can be opened with wxParaver versions newer than 4.3.0, and you'll be able to follow the steps within the tool. To install them, download and untar the package and follow the instructions of the Help/Tutorial option on the Paraver main window. You can download them in a single package either in .tar.gz format (127 Mb) or .zip format (127 Mb).

- Paraver introduction (MPI) Start here to familiarice 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 unnalignement or backwards communications, use this tutorial to learn how to correct shifts between processors.

Methodology of analysis

MPI+OpenMP Performance Analysis tips

Tutorial slides

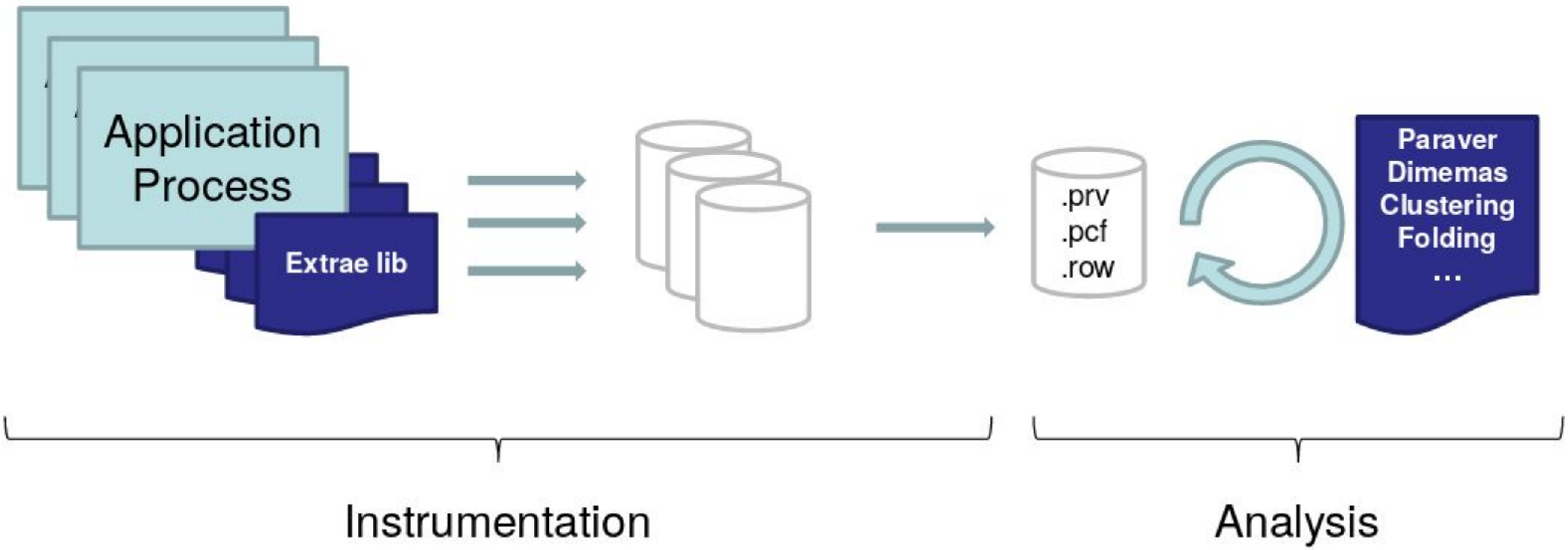
Introduction

Core tools	Advanced features
Paraver, Detailed material	Tools scalability
Dimemas	Clustering
Extrae	Sampling

https://tools.bsc.es/tutorial_guidelines

BSC Tools:Extrae

- Trace generation



BSC Tools:Extrae

- Trace Generation: Set Environment
  - Module load extrae
    - load extrae 3.X.0 (PATH, EXTRAE_DIR, EXTRAE_ROOT, EXTRAE_LIB)
  - Job script → trace-fortran.sh | trace-c.sh
  - Extrae config → extraeMPI.xml | extraeMPI+OMP.xml
  - Files modified for model→ run_parallel.sh

BSC Tools:Extrae

- Job script:trace-fortran.sh
  - Available loading extrae module

```
#!/bin/bash

#Workaround for tracing in MN3, make TMPDIR point to an existing dir
#if [ ! -z "${TMPDIR}" ]; then
#        export TMPDIR=$TMPDIR/extrae
#        mkdir -p $TMPDIR
#fi

EXTRAE_ROOT=/usr/local/apps/extrae

if [ -z "$EXTRAE_DIR" ]; then
    echo "ERROR: EXTRAE_DIR not set, maybe extrae module not loaded?"
    exit -1
fi

if [ -z "$OMP_TRACE" ]; then
    export LD_PRELOAD=${EXTRAE_DIR}/lib/libmpitracef.so
    if [ -z "$EXTRAE_CONFIG_FILE" ]; then
        export EXTRAE_CONFIG_FILE=${EXTRAE_ROOT}/xml/MPI/extrae.xml
    fi
else
    export LD_PRELOAD=${EXTRAE_DIR}/lib/libompitracef.so # For Fortran apps
    if [ -z "$EXTRAE_CONFIG_FILE" ]; then
        export EXTRAE_CONFIG_FILE=${EXTRAE_ROOT}/xml/MPI+OMP/extrae.xml
    fi
fi
```

True if openmp is used

Extrae config by default

BSC Tools:Extrae

- Extrae config:extrae.xml
  - Available using nama_CY43R1_IFS_traces branch

```
<?xml version='1.0'?>

<trace enabled="yes"
 home="/usr/local/apps/extrae/3.3.0rc/MPICH2-6.3.1"
 initial-mode="detail"
 type="paraver"
 xml-parser-id="Id: xml-parse.c 3893 2016-03-04 12:30:23Z harald $"
>
  <mpi enabled="yes">
    <counters enabled="yes" />
  </mpi>

  <openmp enabled="no">
    <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="yes">1-5</sampling>
    <dynamic-memory enabled="no">1-3</dynamic-memory>
  </callers>

  <user-functions enabled="yes" list="/perm/rd/nama/extrae/xml/MPI/functions_for_xml.txt" exclude-automatic-functions="no">
    <counters enabled="yes" />
  </user-functions>

  <counters enabled="yes">
    <cpu enabled="yes" starting-set-distribution="1">
      <set enabled="yes" domain="all" changeat-time="0">
        PAPI_TOT_INS,PAPI_TOT_CYC,PAPI_L1_DCM,PAPI_L2_DCM,PAPI_L3_TCM,PAPI_FP_INS,PAPI_BR_MSP,PAPI_FP_OPS
      </set>
      <set enabled="yes" domain="all" changeat-time="0">
        PAPI_TOT_INS,PAPI_TOT_CYC,PAPI_LD_INS,PAPI_SR_INS,PAPI_BR_UCN,PAPI_BR_CN,PAPI_VEC_SP,RESOURCE_STALLS
        <sampling enabled="no" period="1000000000">PAPI_TOT_CYC</sampling>
      </set>
    </cpu>
    <network enabled="no" />
    <resource-usage enabled="no" />
    <memory-usage enabled="no" />
  </counters>
```

Activate MPI tracing and emit hardware counters at MPI calls

Activate OpenMP tracing

Emit call stack information (number of levels) at acquisition point

Add instrumentation at specific user functions

PAPI counters used

BSC Tools:Extrae

- Extrae config:extrae.xml
  - Available using nama_CY43R1_IFS_traces branch

```
<storage enabled="no">
  <trace-prefix enabled="yes">TRACE</trace-prefix>
  <size enabled="no">5</size>
  <temporal-directory enabled="yes">/scratch</temporal-directory>
  <final-directory enabled="yes">/gpfs/scratch/bsc41/bsc41273</final-directory>
</storage>

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

<trace-control enabled="no">
  <file enabled="no" frequency="5M">/gpfs/scratch/bsc41/bsc41273/control</file>
  <global-ops enabled="no"></global-ops>
  <remote-control enabled="no">
    <signal enabled="no" which="USR1"/>
  </remote-control>
</trace-control>

<others enabled="yes">
  <minimum-time enabled="no">10M</minimum-time>
  <finalize-on-signal enabled="yes"
    SIGUSR1="no" SIGUSR2="no" SIGINT="yes"
    SIGQUIT="yes" SIGTERM="yes" SIGXCPU="yes"
    SIGFPE="yes" SIGSEGV="yes" SIGABRT="yes"
  />
  <flush-sampling-buffer-at-instrumentation-point enabled="yes" />
</others>

<bursts enabled="no">
  <threshold enabled="yes">500u</threshold>
  <mpi-statistics enabled="yes" />
</bursts>

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

<dynamic-memory enabled="no">
  <alloc enabled="yes" threshold="32768" />
  <free  enabled="yes" />
</dynamic-memory>

<input-output enabled="no" />

<merge enabled="yes"
  synchronization="default"
  tree-fan-out="16"
  max-memory="32768"
  joint-states="yes"
  keep-mpits="yes"
  sort-addresses="yes"
  overwrite="yes"
/>

</trace>
```

Emit computation burst of a minimal duration

Plus summarized MPI events

Merge individual traces automatically

BSC Tools:Extrae

- Files modified for run_parallel
  - If BSCTRACE=1 → Extrae is used

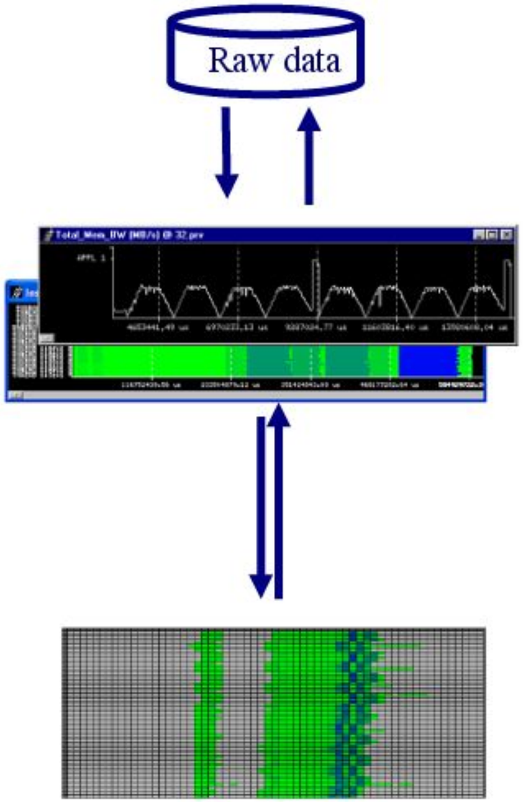
```
    if [[ %BSCTRACE:0% = 1 ]]; then
      command="aprun -cc cpu $marg -n $submit_total_tasks -N $submit_tasks_per_node $Sarg -j $submit_cpus_per_compute_unit -d $omp_num_threads $submit_force_numa_memory_affinity trace-fortran.sh $(whence $cm
d) $args"
    else
      command="aprun -cc cpu $marg -n $submit_total_tasks -N $submit_tasks_per_node $Sarg -j $submit_cpus_per_compute_unit -d $omp_num_threads $submit_force_numa_memory_affinity $(whence $cmd) $args"
    fi
  fi
```

Parameter to activate profiling

```
export EC_LD_LIBRARY_PATH=${PIFS_LD_LIBRARY_PATH:-$LIBS}
if [[ %BSCTRACE:0% = 1 ]]; then
  LOAD_MODULE extrae
  export EXTRAE_CONFIG_FILE=/perm/rd/nama/extrae/xml/MPI/extrae.xml
  export TRACEDIR=${WDIR}/bsctrace.${TASK}.${ECF_TRYNO}.$$
  export EC_LD_LIBRARY_PATH=${EC_LD_LIBRARY_PATH}:$EXTRAE_LIB
  if [[ %OMPTRACE:0% = 1 ]]; then
   export OMP_TRACE=1
  fi
  mkdir $TRACEDIR
fi
(export LD_LIBRARY_PATH=$EC_LD_LIBRARY_PATH ; eval $command)
if [[ %BSCTRACE:0% = 1 ]]; then
  mv *.prv *.pcf *.row $TRACEDIR
fi
```

Trace files generated

BSC Tools:Paraver



Timelines

2/3D tables (Statistics)

Trace visualization/analysis

+ trace manipulation

Goal = Flexibility

No semantics

Programmable

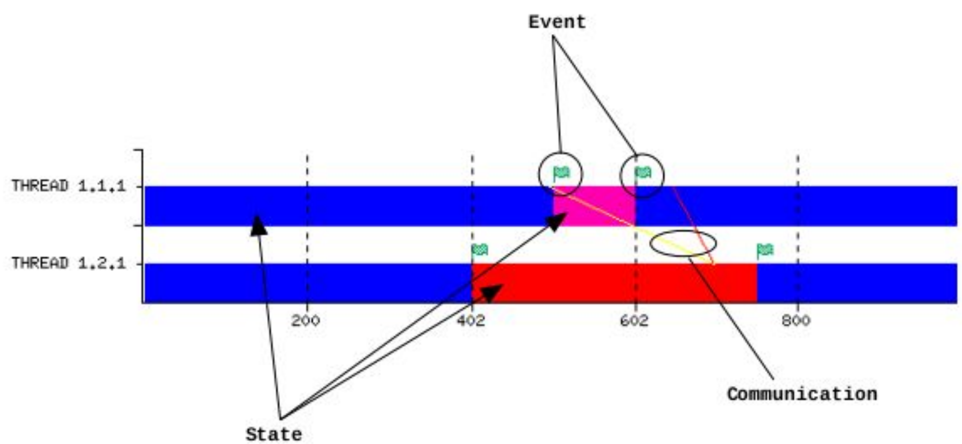
Comparative analyses

Multiple traces

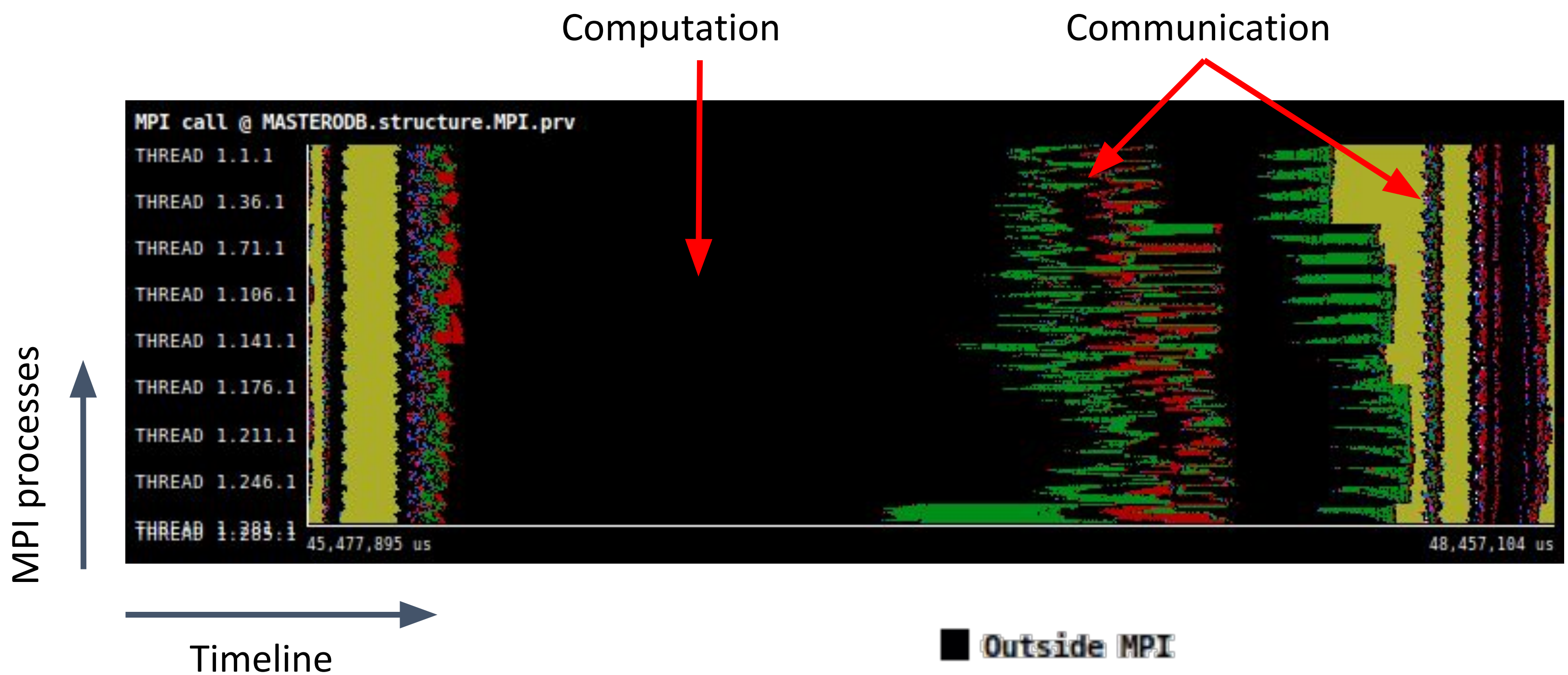
Synchronize scales

BSC Tools:Paraver

- **Paraver traces:** made up from records (timestamp + event or activity) of three different kind:
  - **State records:** intervals of thread status, i.e, waiting in a barrier (either MPI or OpenMP), waiting for a message, computing...
  - **Event records:** punctual event occurred in a given timestamp, as entry & exit points of user functions, MPI routines, OpenMP parallel regions...
  - **Communication records:** relationship between two objects, as communication between two processes (MPI), task movement among threads (OpenMP/OmpSs) or memory transfers (CUDA/OpenCL).



How a trace looks like: basic overview



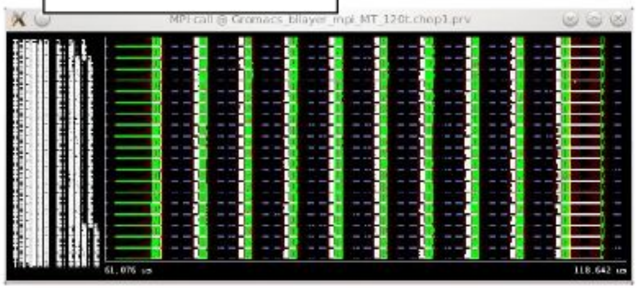
MPI call color legend

- Outside MPI
- MPI_Recv
- MPI_Isend
- MPI_Irecv
- MPI_Wait
- MPI_Alltoallv
- MPI_Comm_size
- MPI_Waitany

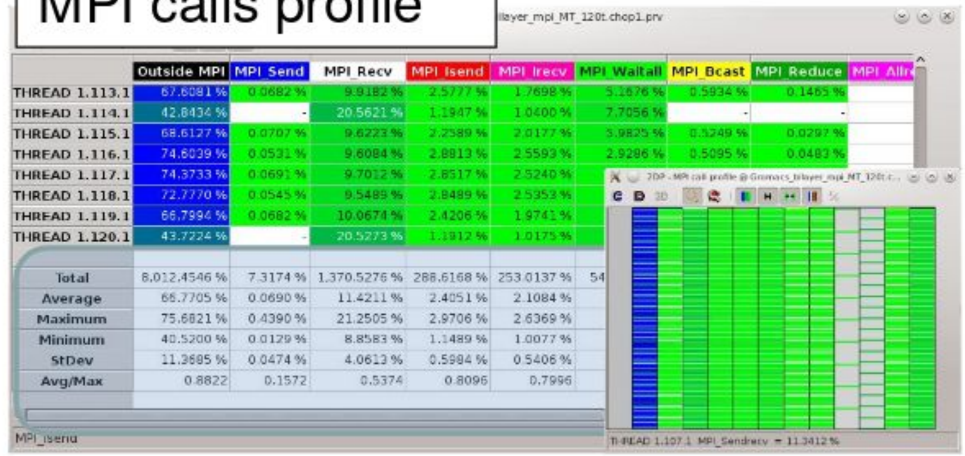
BSC Tools:Paraver

- From timelines to tables

MPI calls



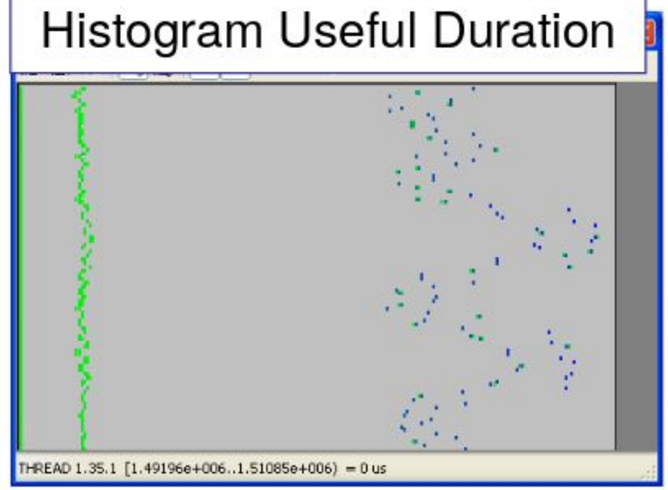
MPI calls profile



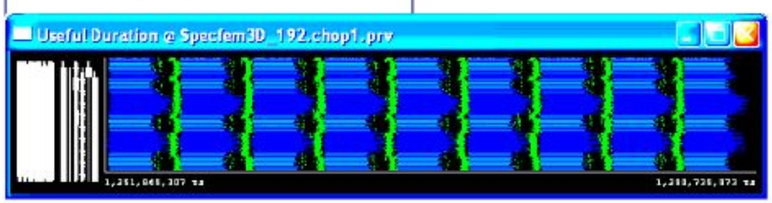
	Outside MPI	MPI_Send	MPI_Recv	MPI_Isend	MPI_Irecv	MPI_Waitall	MPI_Bcast	MPI_Reduce	MPI_Allr
THREAD 1.113.1	67.6081 %	0.0682 %	9.9182 %	2.5777 %	1.7698 %	5.1676 %	0.5934 %	0.1465 %	
THREAD 1.114.1	42.8434 %	-	20.5621 %	1.1947 %	1.0400 %	7.7056 %	-	-	
THREAD 1.115.1	68.6127 %	0.0707 %	9.6223 %	2.2589 %	2.0177 %	5.9825 %	0.5249 %	0.0297 %	
THREAD 1.116.1	74.6039 %	0.0531 %	9.6084 %	2.8813 %	2.5593 %	2.9286 %	0.5095 %	0.0483 %	
THREAD 1.117.1	74.3733 %	0.0691 %	9.7012 %	2.8517 %	2.5240 %				
THREAD 1.118.1	72.7770 %	0.0545 %	9.5489 %	2.8489 %	2.5353 %				
THREAD 1.119.1	66.7994 %	0.0682 %	10.0674 %	2.4206 %	1.9741 %				
THREAD 1.120.1	43.7224 %	-	20.5273 %	1.1912 %	1.0175 %				
Total	8,012.4546 %	7.3174 %	1,370.5276 %	288.6168 %	253.0137 %	54			
Average	66.7705 %	0.0690 %	11.4211 %	2.4051 %	2.1084 %				
Maximum	75.6821 %	0.4390 %	21.2505 %	2.9706 %	2.6369 %				
Minimum	40.5200 %	0.0129 %	8.8583 %	1.1489 %	1.0077 %				
StDev	11.3685 %	0.0474 %	4.0613 %	0.5984 %	0.5406 %				
Avg/Max	0.8822	0.1572	0.5374	0.8096	0.7996				

MPI_Isend

Histogram Useful Duration



Useful Duration



BSC Tools:Paraver

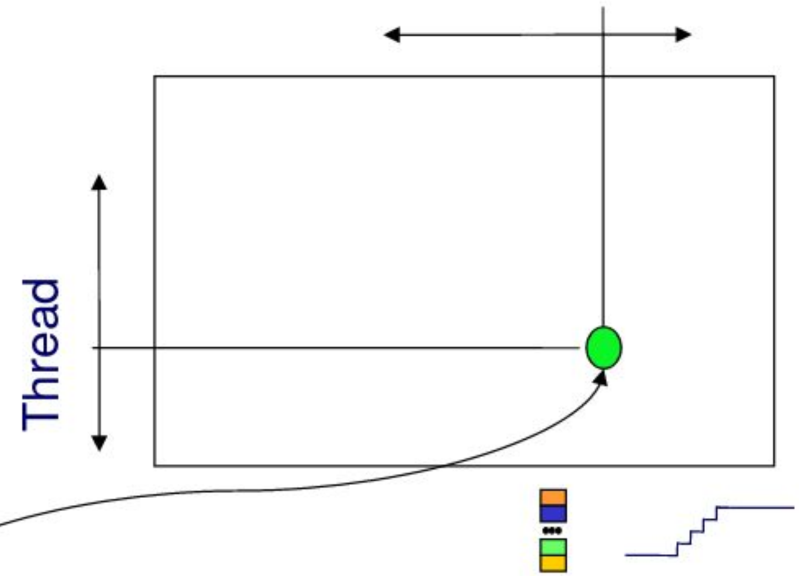
One columns per specific value of categorical **Control window**

MPI call, user function,…

MPI profile @ Iberia-128-CA.chop1.1it.shifted.prv

	End	MPI_Isend	MPI_Irecv	MPI_Wait	MPI_Allreduce	MPI_Com
THREAD 1.1.1	86,98 %	0,06 %	0,08 %	11,12 %	1,75 %	
THREAD 1.2.1	88,29 %	0,10 %	0,10 %	9,95 %	1,56 %	
THREAD 1.3.1	88,33 %	0,13 %	0,10 %	9,92 %	1,51 %	
THREAD 1.4.1	89,75 %	0,10 %	0,09 %	8,62 %	1,44 %	
THREAD 1.5.1	89,47 %	0,11 %	0,10 %	8,85 %	1,46 %	
THREAD 1.6.1	88,76 %	0,12 %	0,09 %	9,54 %	1,48 %	
THREAD 1.7.1	91,77 %	0,13 %	0,10 %	6,51 %	1,49 %	
THREAD 1.8.1	90,23 %	0,06 %	0,08 %	8,13 %	1,50 %	
THREAD 1.9.1	91,88 %	0,13 %	0,09 %	6,73 %	1,17 %	
THREAD 1.10.1	93,24 %	0,18 %	0,11 %	5,41 %	1,05 %	
THREAD 1.11.1	93,25 %	0,18 %	0,11 %	5,45 %	1,00 %	
THREAD 1.12.1	94,63 %	0,17 %	0,11 %	4,16 %	0,93 %	
THREAD 1.13.1	93,40 %	0,17 %	0,11 %	5,35 %	0,96 %	
THREAD 1.14.1	94,99 %	0,20 %	0,11 %	3,77 %	0,93 %	
THREAD 1.15.1	96,80 %	0,22 %	0,11 %	1,92 %	0,95 %	
THREAD 1.16.1	95,73 %	0,12 %	0,09 %	2,99 %	1,06 %	

MPI_Wait



Value/color is a statistic computed for the specific thread when control window had the value corresponding to the column

Relevant statistics:
Time, %time, #bursts, Avg. burst time
Average of **Data window**

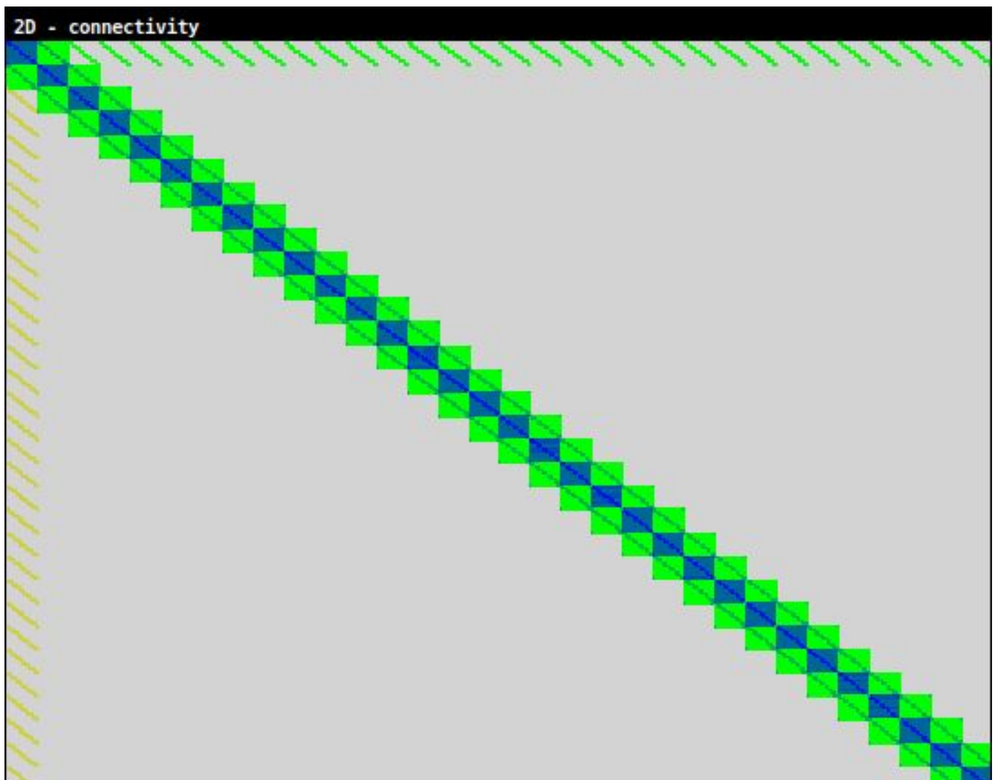
MPI calls and profile

- Different types of MPI functions are quantified
- In this case, only the MPI_Alltoallv and MPI_Waitany functions represent a significant amount of time with 14.65% and 9.29% respectively.

	Outside MPI	MPI_Recv	MPI_Isend	MPI_Irecv	MPI_Wait	MPI_Alltoallv	MPI_Comm_size	MPI_Waitany
Total	41,154.02 %	37.32 %	260.82 %	172.16 %	2,023.25 %	8,438.25 %	163.09 %	5,351.09 %
Average	71.45 %	0.07 %	0.45 %	0.30 %	3.51 %	14.65 %	0.28 %	9.29 %
Maximum	84.51 %	0.18 %	1.15 %	0.81 %	9.70 %	20.07 %	0.31 %	31.12 %
Minimum	51.79 %	0.00 %	0.13 %	0.12 %	0.49 %	8.59 %	0.23 %	1.31 %
StDev	4.91 %	0.04 %	0.18 %	0.14 %	1.79 %	3.07 %	0.01 %	4.33 %
Avg/Max	0.85	0.38	0.39	0.37	0.36	0.73	0.93	0.30

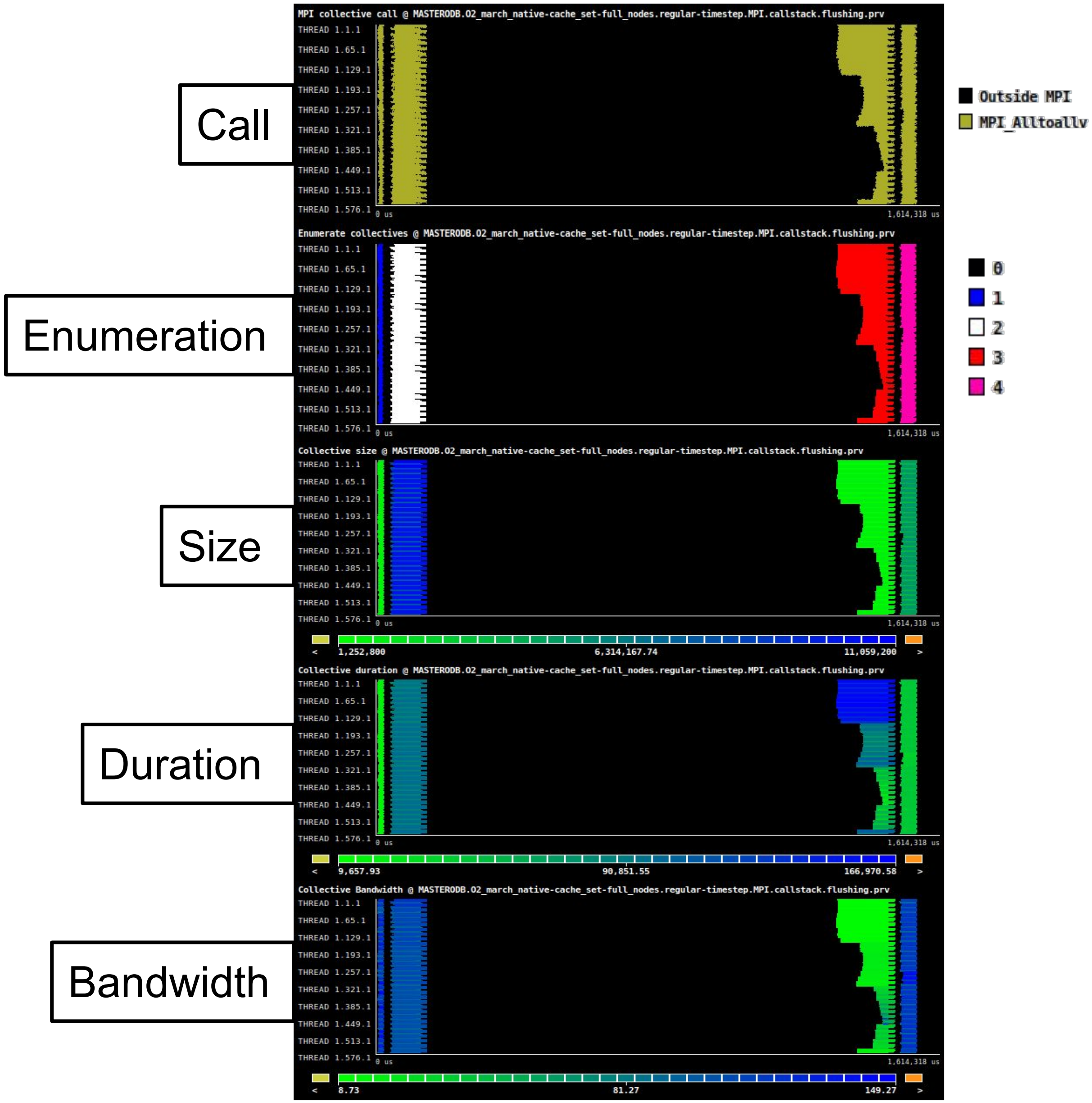
Point-to-point connectivity matrix

- It indicates who communicates with whom
- Almost all point-to-point communications are locally performed between MPI processes neighbours



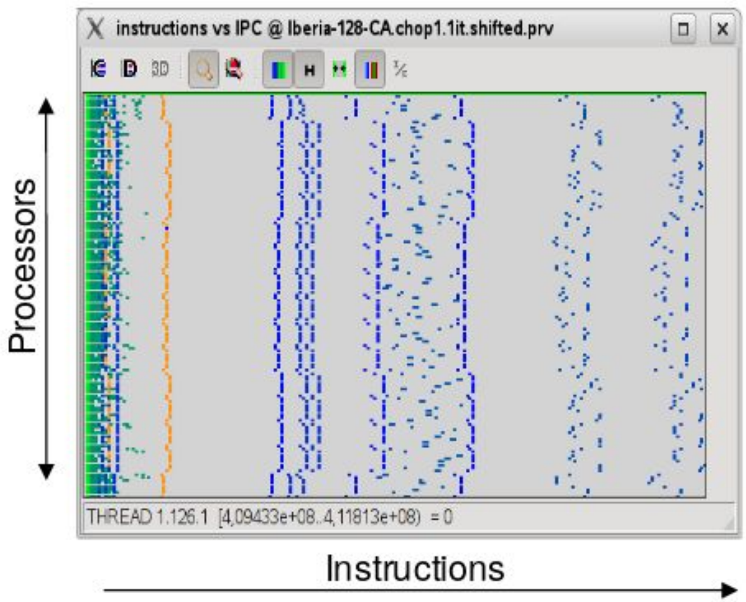
Collective communications

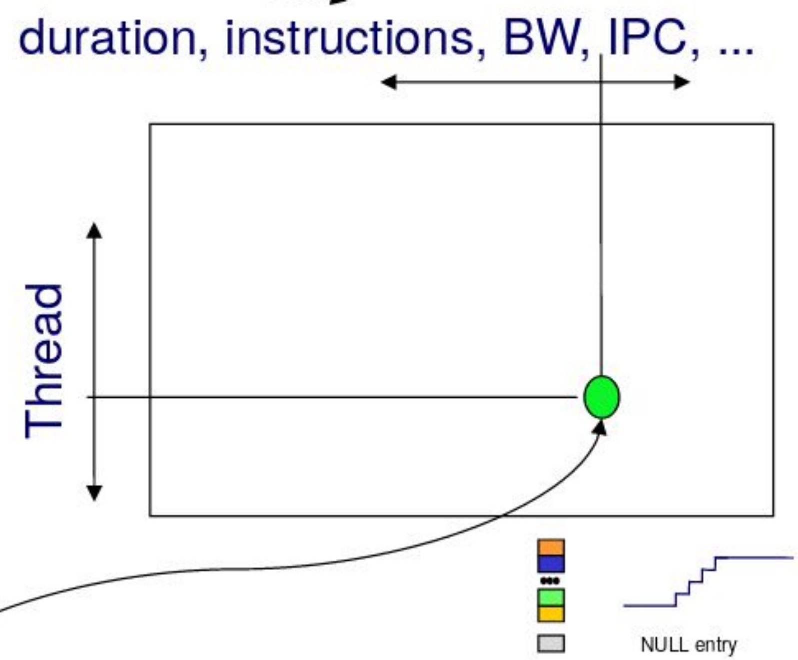
- Four calls to MPI_Alltoallv each time step
- The most significant in terms of size and duration is the second one



BSC Tools:Paraver

Columns correspond to bins of values of a numeric **Control window**



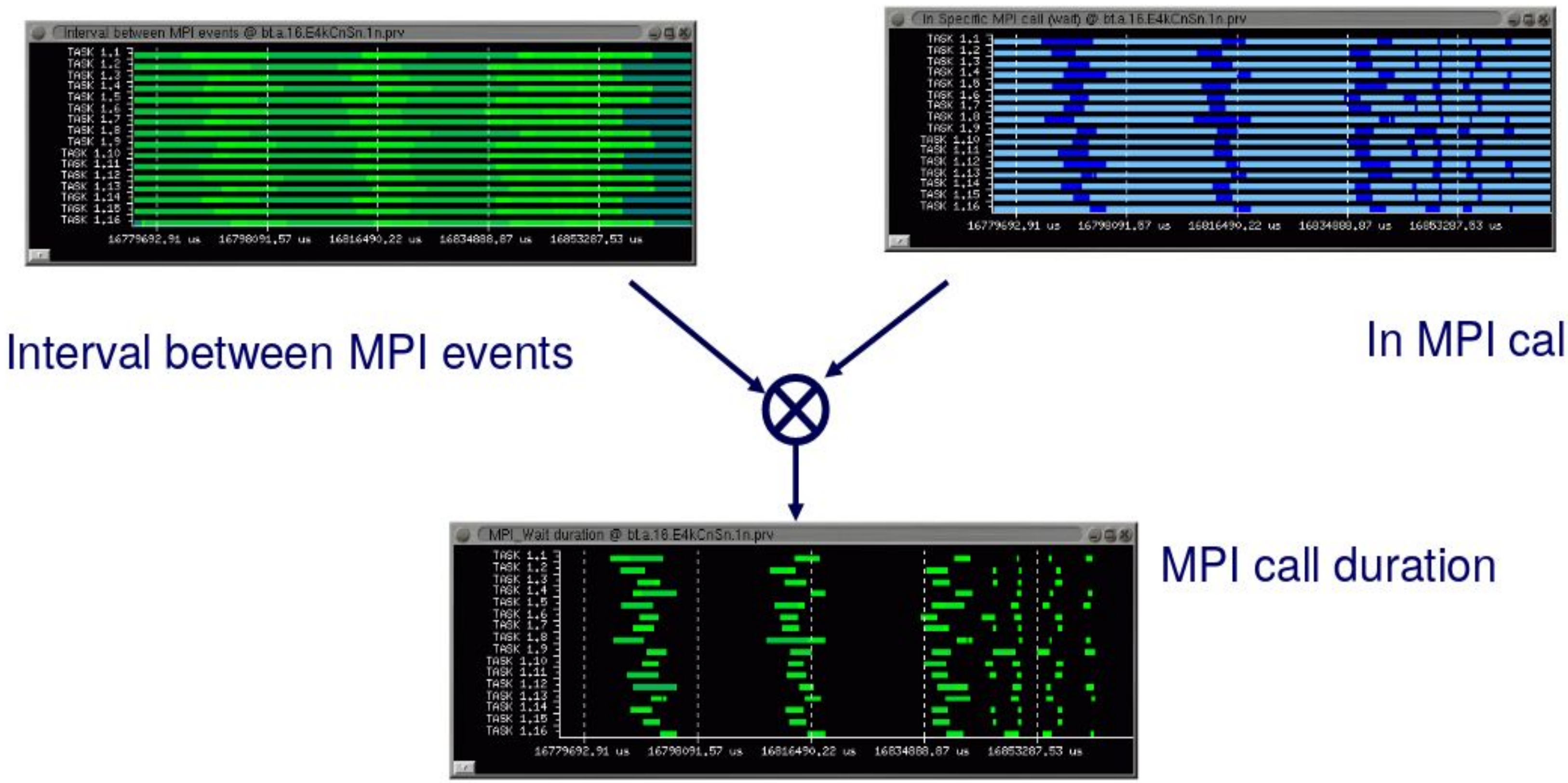


Value/color is a statistic computed for the specific thread
when control window had the value corresponding to the column

Relevant statistics:
Time, %time, #bursts, Avg. burst time
Average of **Data window**

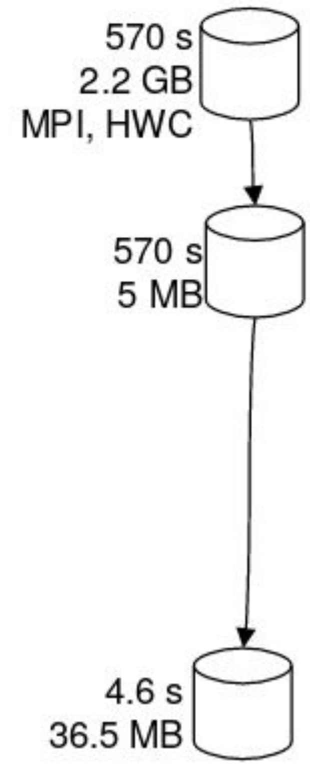
BSC Tools:Paraver

- Semantic functionality
  - Derived windows
    - Point wise operation
      - $S = \alpha * S^a \langle op \rangle \ \beta * S^b$
      - <op> : + , -, *, /, …



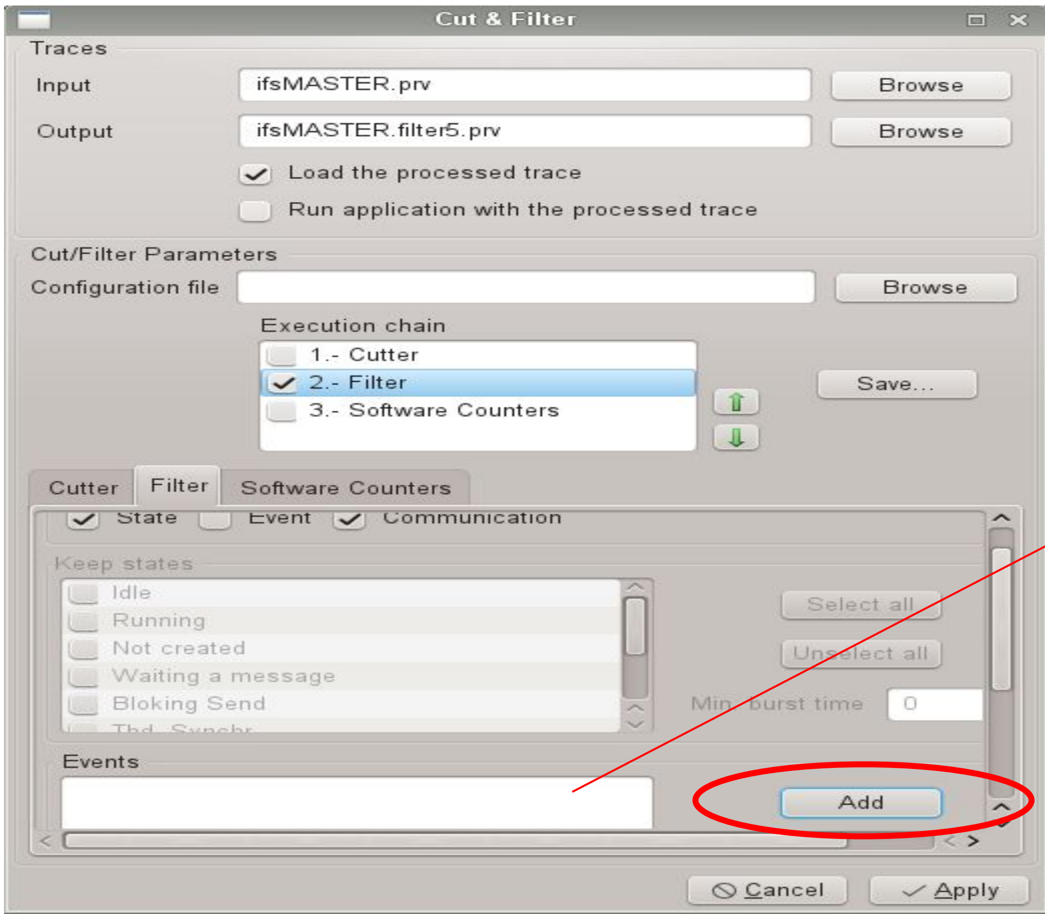
BSC Tools:Paraver

- Data handling capability
  - Original trace containing all the events
  - Filtering/Burst mode
    - Subset of records in original trace
    - By duration, time, value, event type
    - Trace filtered can be analysed in the same way
    - Also using burst mode from xml file
      - Save only computation bursts longer than a value
  - Cutting
    - All records in a given time interval
    - Only some processes



BSC Tools:Paraver

- Filtering
  - Filter original trace discarding most of the records, only keeping most relevant information (MPI events can be used for this purpose)

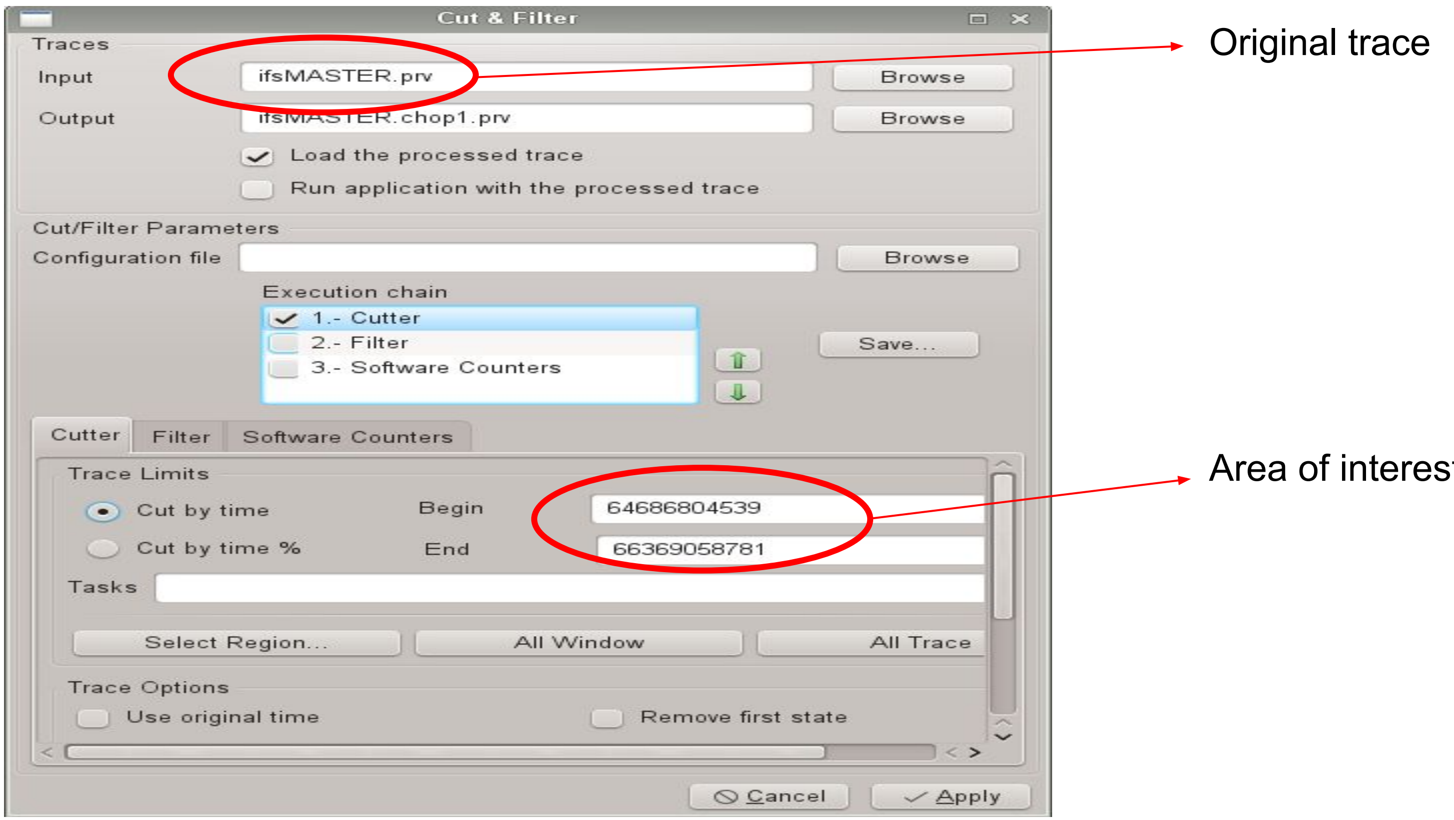


50000001
50000002
50000003 MPI events
50100001
50100002
50100003
50100004

BSC Tools:Paraver

- Cutting

- Cut original trace to obtain a fully detailed trace for the time interval considered representative or of interest
- Use filtered trace to know the area of interest (remember that input must be the original trace)
  - Right click → run → cutter



BSC Tools:Paraver

- Configurations for analysis (usr/local/apps/paraver/X.X.X/cfgs)
  - General
    - Including basic views (timelines) and analysis (2D/3D profiles)
  - Counters_PAPI
    - Hardware counters derived metrics
      - Program: related to algorithm/compilation (instructions, FP ops…)
      - Architecture:related to execution on specific architectures (cache misses…)
      - Performance: metrics reporting rating per time (MIPS, IPC…)
  - MPI → Views and analysis of MPI events
  - OpenMP → Views and analysis of OpenMP events
  - Complete Profile (general_cfgs)

BSC Tools:Dimemas

The impossible machine: BW = ∞, L = 0

- Actually describes/characterizes intrinsic application behavior
  - Load balance problems?
  - Dependence problems?

Profiling Methodology

- Area of study
- Deployment efficiency
- Benchmarking
- Profiling analysis
- Validation

Profiling Methodology

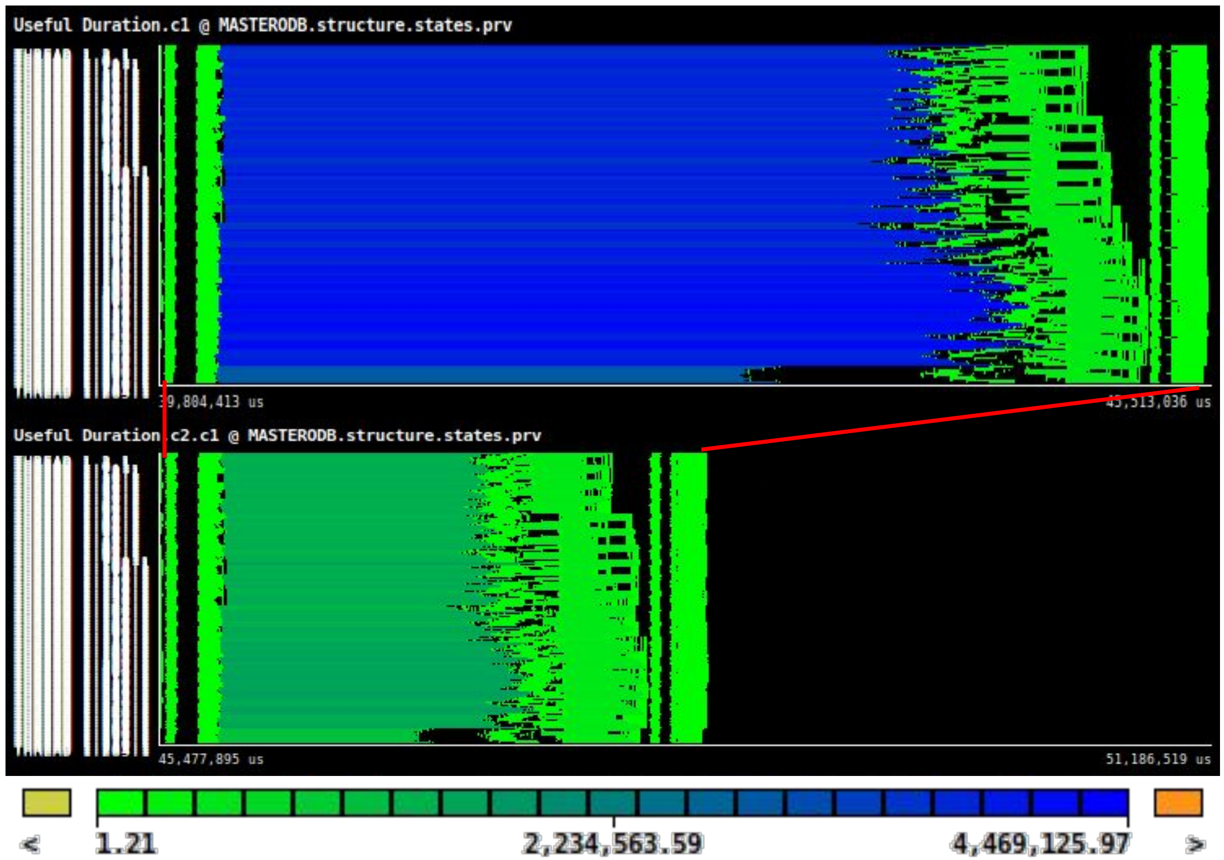
- **Area of study**
  - Configuration used (Operational, New algorithms, Global, Parallelization paradigm…)
  - Components activated and cyclic patterns
    - IO, ICE, Radiation, MPI, OpenMP
  - Area of study
    - 1 complete time step
- Deployment efficiency
- Benchmarking
- Profiling analysis
- Validation

Types of time step for the practical example

Time steps with radiation are much more expensive due to the extra computation in the grid-point part

Regular time step plus radiation

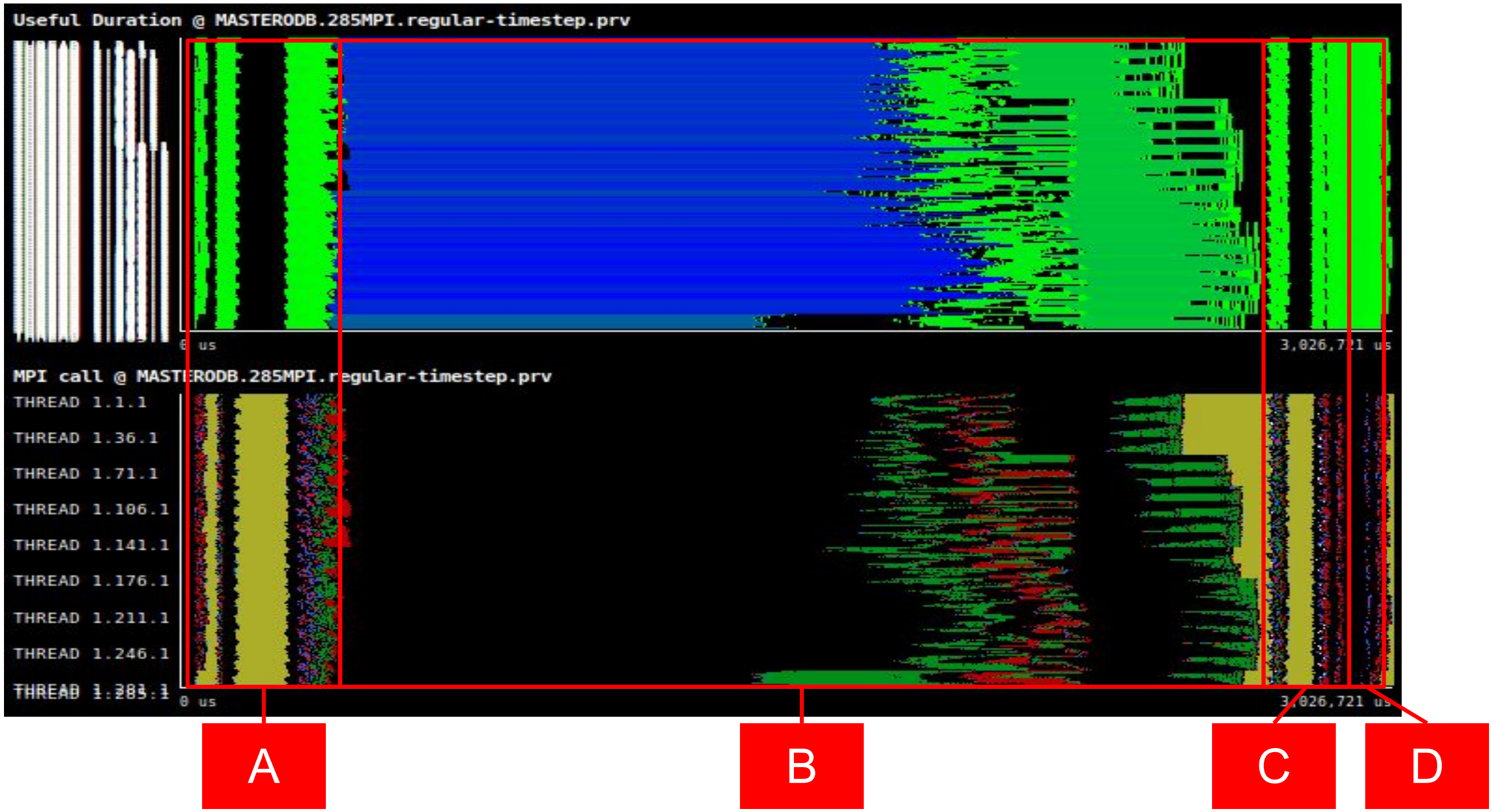
Regular time step



Barcelona Supercomputing Center
Centro Nacional de Supercomputación

esiwace
CENTRE OF EXCELLENCE IN SIMULATION OF WEATHER AND CLIMATE IN EUROPE

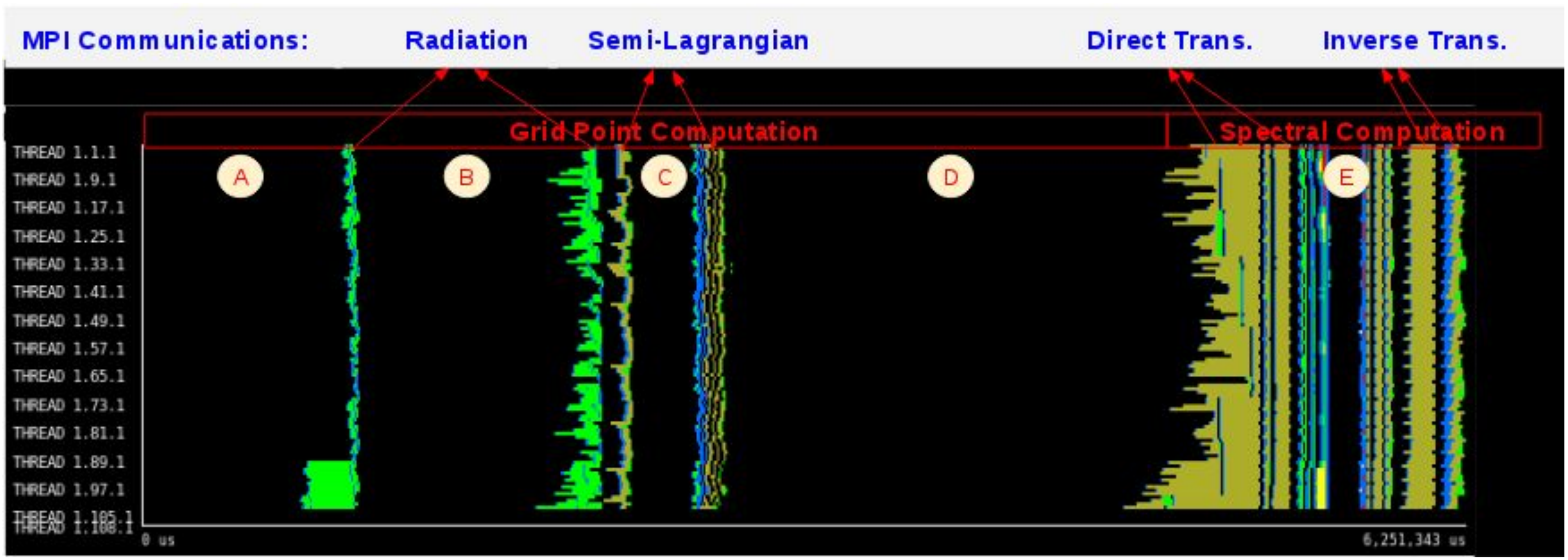
Structure of a regular time step



A - Inverse transformations
B - Grid-point computations
C - Direct transformations
D - Spectral computations

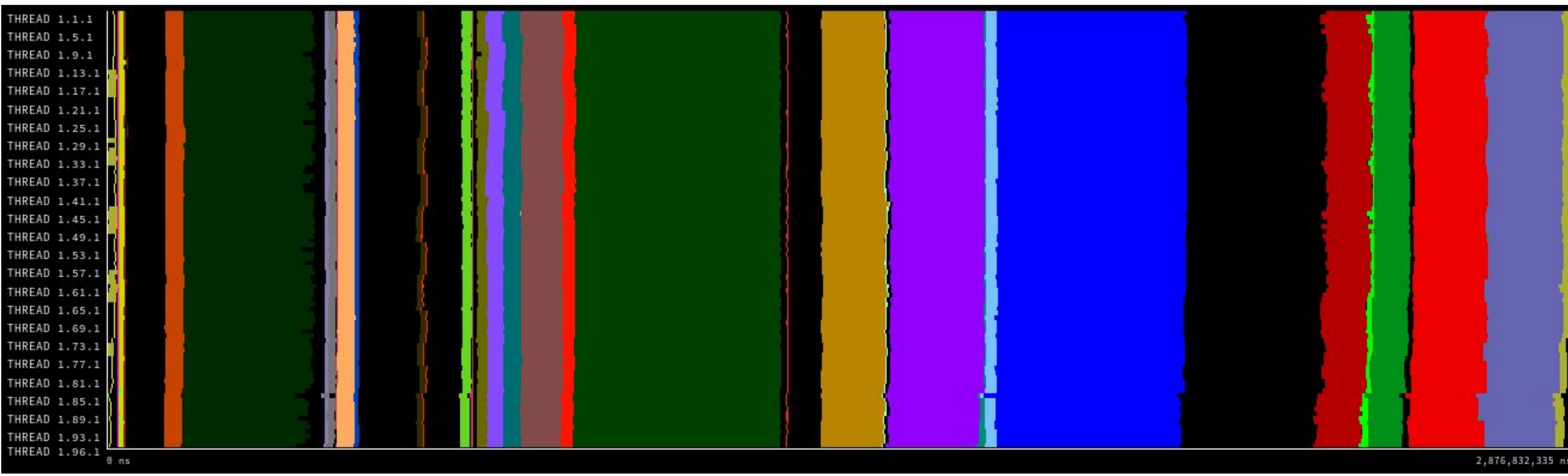
Profiling Methodology

- Area of study (IFS)
  - 24 hours of simulation, T511L137 on CCA (ECMWF)
  - Selected 1 time step: 104 MPI processes + 4 IO (No OpenMP)
  - Metrics collected for large areas of computation automatically

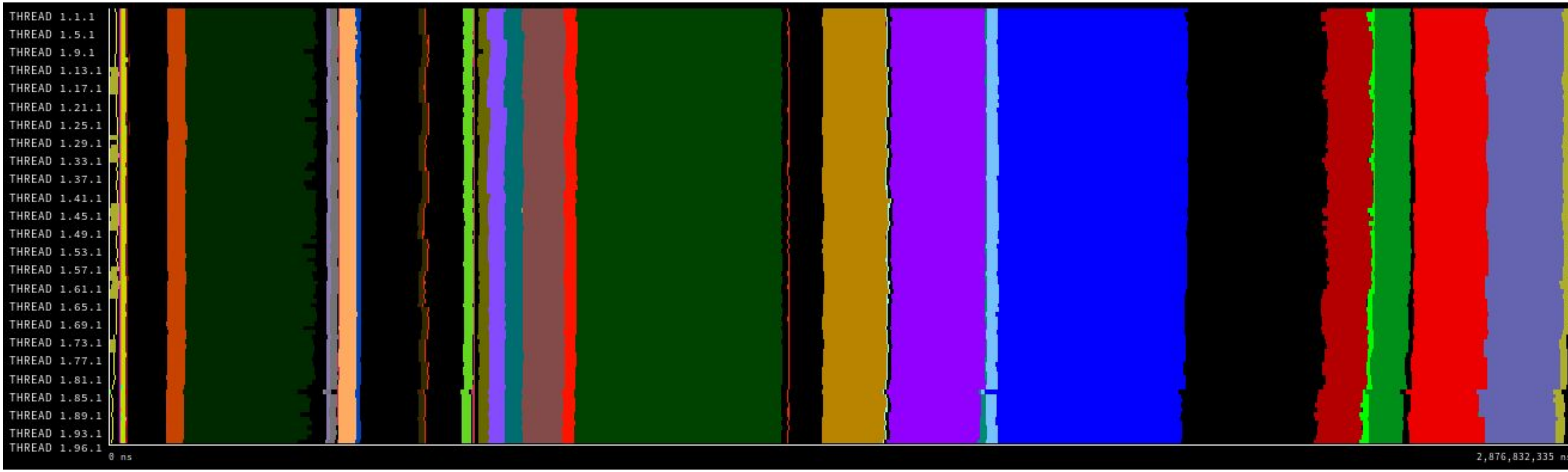


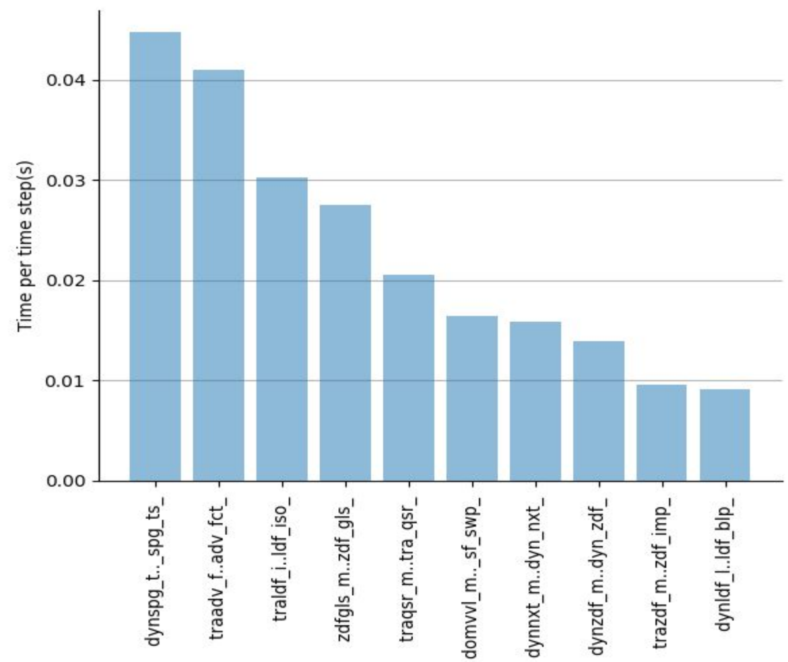
Profiling Methodology

- Area of study (NEMO)
  - 1 day of simulation, ORCA025L91 on MN4 (BSC)
  - Selected the fastest time step automatically
  - 1 time step: 72 MPI processes (No IO, No OpenMP, No SI3)
  - Metrics collected for User functions manually



Profiling Methodology





Barcelona Supercomputing Center
Centro Nacional de Supercomputación

esiwace
CENTRE OF EXCELLENCE IN SIMULATION OF WEATHER AND CLIMATE IN EUROPE

Profiling Methodology

- Area of study
- **Deployment efficiency**
  - Compilation flags
    - Comparing fp options (fast, precise, strict...) and optimization options (OX, vectorization, approximations...)
  - Checking external libraries compilation
  - Debug flags (-g, Optimization reports, -f-instrument-functions...)
- Benchmarking
- Profiling analysis
- Validation

Profiling Methodology

- Area of study
- Deployment efficiency
- **Benchmarking**
  - Basic Tests to collect Hardware metrics
    - Communications (Latency, Bandwidth, CPU, Parallel Efficiency…)
  - Weak and Strong scaling (MPI, OpenMP, Block processing and Hybrid sets)
  - Comparing optimizations (Double VS Single Precision...)
  - Extrae metrics collection and trace production
- Profiling analysis
- Validation

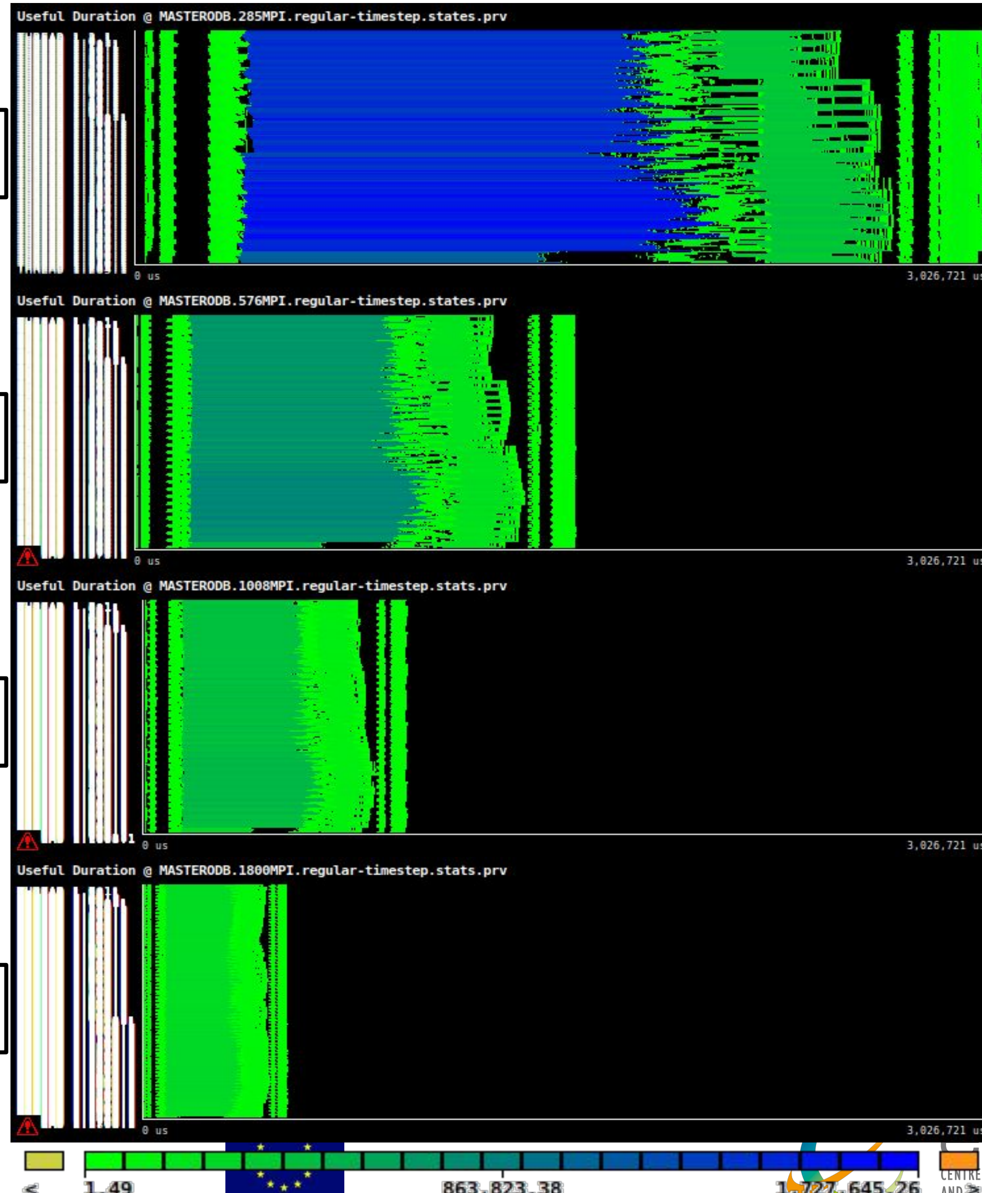
MPI strong scaling: trace views

285 MPI (8 nodes)

576 MPI (16 nodes)

1008 MPI (28 nodes)

1800 MPI (50 nodes)



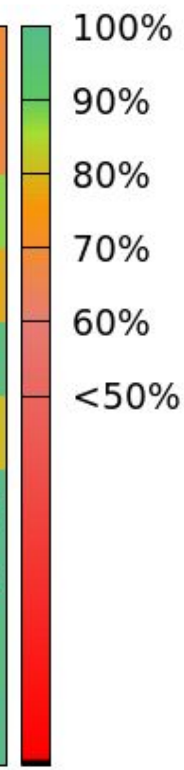
Barcelona Supercomputing Center
Centro Nacional de Supercomputación

CENTRE OF EXCELLENCE IN SIMULATION OF WEATHER AND CLIMATE IN EUROPE

Basic Analysis: MPI Strong Scaling

- Computation and parallel efficiency factors for MPI only:
  - Good computation scalability and serialization efficiency
  - Not very good load balance neither transfer efficiency

Global efficiency	75.34	72.04	68.04	69.82
-- Parallel efficiency	75.34	71.36	68.37	67.69
-- Load balance	88.05	85.38	87.76	86.87
-- Communication efficiency	85.56	83.58	77.91	77.93
-- Serialization efficiency	96.35	98.23	94.03	96.87
-- Transfer efficiency	88.81	85.09	82.85	80.44
-- Computation scalability	100.00	100.95	99.51	103.15
-- IPC scalability	100.00	102.06	101.86	106.12
-- Instruction scalability	100.00	99.24	98.00	97.55
-- Frequency scalability	100.00	99.67	99.69	99.64



Basic Analysis: Double P VS Single P

Overview of the collected raw data:

	108	108
Runtime (us)	110741508.76	71238767.9
Runtime (ideal)	105675625.64	68396939.23
Useful duration (average)	88427932.03	57382830.24
Useful duration (maximum)	94410288.2	61484222.58
Useful duration (total)	9196504931.3	5967814345.21
Useful duration (ideal, max)	94410288.2	61484222.58
Useful instructions (total)	26798422515714	23201423473963
Useful cycles (total)	21985000332874	14299301515415

Overview of the computed model factors:

	108	108
Parallel efficiency	79.85%	80.55%
Load balance	93.66%	93.33%
Communication efficiency	85.25%	86.31%
Serialization efficiency	89.34%	89.89%
Transfer efficiency	95.43%	96.01%
Computation scalability	100.00%	154.10%
Global efficiency	79.85%	124.13%
IPC scalability	100.00%	133.11%
Instruction scalability	100.00%	115.50%
Frequency scalability	100.00%	100.23%
Speedup	1.00	1.55
Average IPC	1.22	1.62
Average frequency (GHz)	2.39	2.40

Profiling Methodology

- Area of study
- Deployment efficiency
- Benchmarking
- **Profiling analysis**
  - MPI and OpenMP profile summary and Basic Analysis Tool
  - PAPI counters
  - MPI and OpenMP evaluation in detail
  - Clustering and Tracking Tools
  - Sampling and Folding Tools
  - Connection to the code
  - Dimemas Tool
- Validation

MPI Profile Summary

Parallel and Communication efficiency, Global load balance → less than 85%?

Parallel Efficiency

IFS

	Outside MPI	MPI_Send	MPI_Recv	MPI_Isend	MPI_Irecv	MPI_Wait	MPI_Barrier	MPI_Alltoallv	MPI_Gatherv	MPI_Comm_rank	MPI_Comm_size	MPI_Bsend	MPI_Waitany
Total	66,578.44 %	1.71 %	773.76 %	646.21 %	239.35 %	12,362.37 %	806.93 %	10,757.31 %	35.56 %	2.49 %	448.23 %	0.81 %	7,746.82 %
Average	66.31 %	0.00 %	0.77 %	0.64 %	0.24 %	12.31 %	0.80 %	10.71 %	0.04 %	0.00 %	0.45 %	0.81 %	7.72 %
Maximum	72.93 %	0.01 %	2.98 %	1.60 %	0.80 %	18.56 %	1.84 %	25.06 %	1.12 %	0.01 %	1.88 %	0.81 %	19.25 %
Minimum	57.05 %	0.00 %	0.01 %	0.08 %	0.07 %	3.11 %	0.00 %	5.25 %	0.00 %	0.00 %	0.16 %	0.81 %	0.31 %
StDev	2.03 %	0.00 %	0.57 %	0.36 %	0.06 %	2.52 %	0.41 %	3.57 %	0.12 %	0.00 %	0.10 %	0 %	3.18 %
Avg/Max	0.91	0.31	0.26	0.40	0.30	0.66	0.44	0.43	0.03	0.34	0.24	1	0.40

MPI_Send

Communication Efficiency

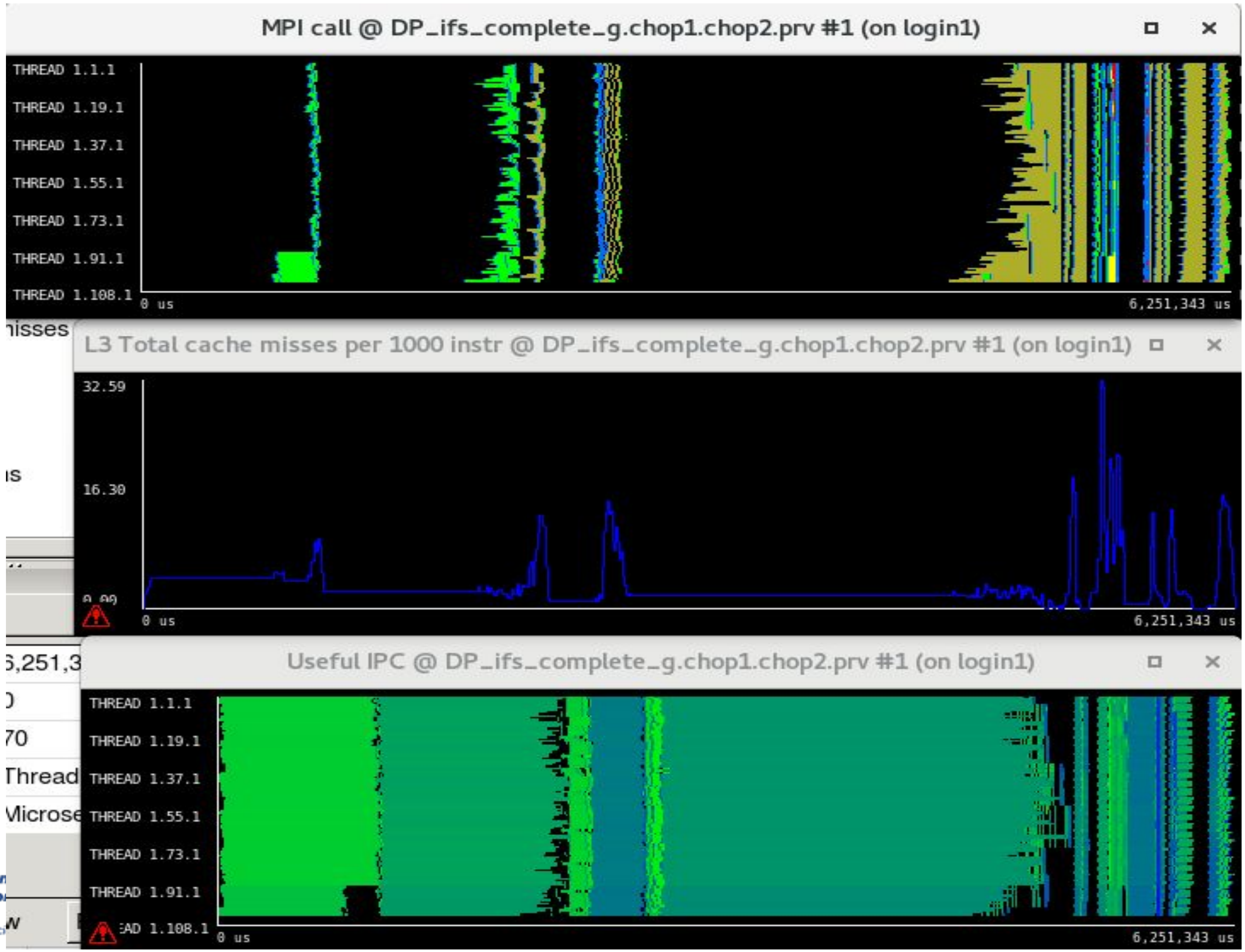
Global Load Balance

PAPI Counters

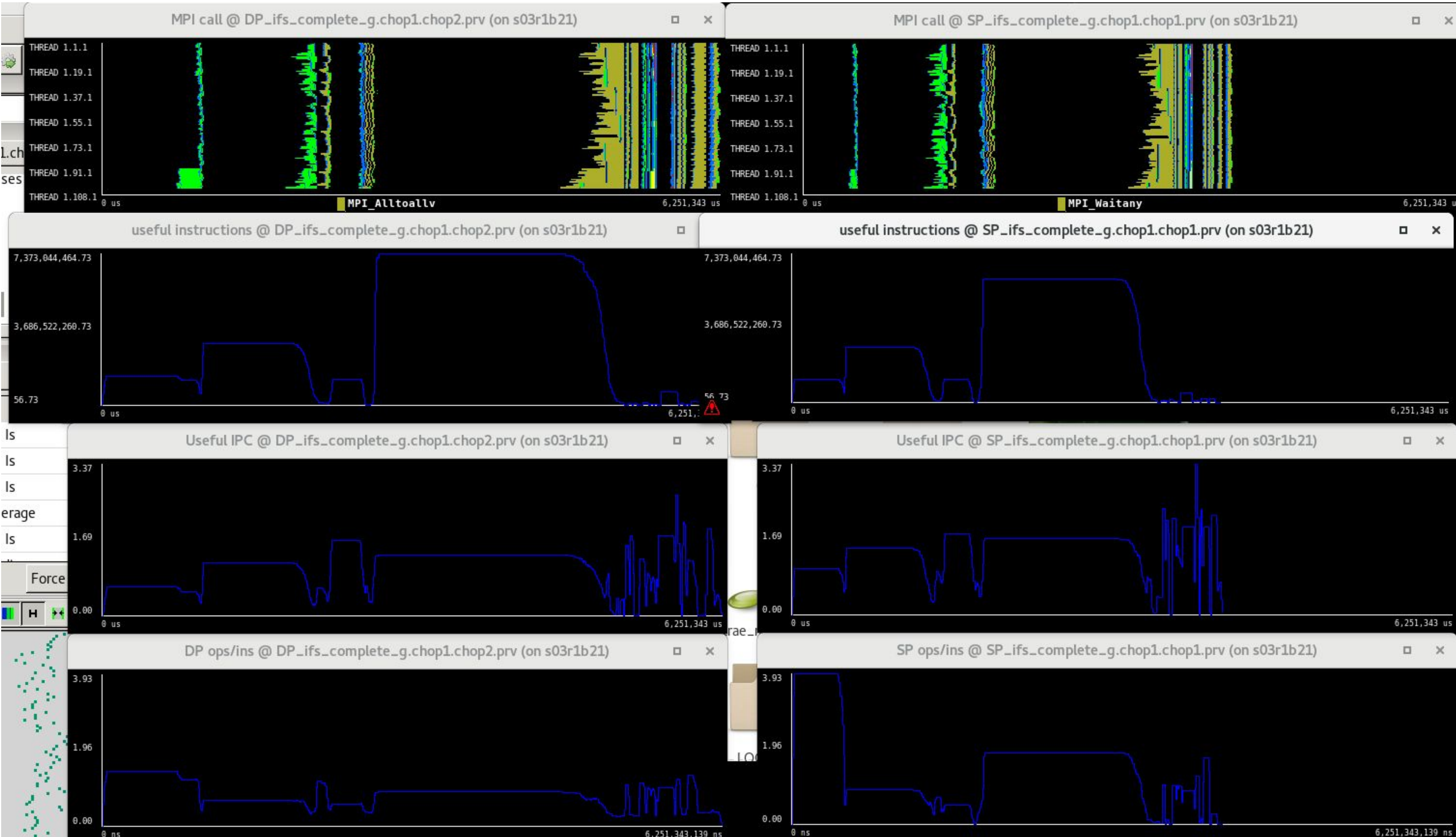
- PAPI counters collected during the execution
- Some of them are based on other native PAPI counters and derived from the base metrics

	Derived
Instructions	
Cycles	
Useful Duration	X
Useful Instructions	X
Useful IPC	X
Loads	
Stores	
L3/L2/L1_Total_Misses	
L3/L2/L1_MISS_RATIO	X
FP_OPS	
FP_TOT_INS	
INS_VEC	X

PAPI Counters



PAPI Counters

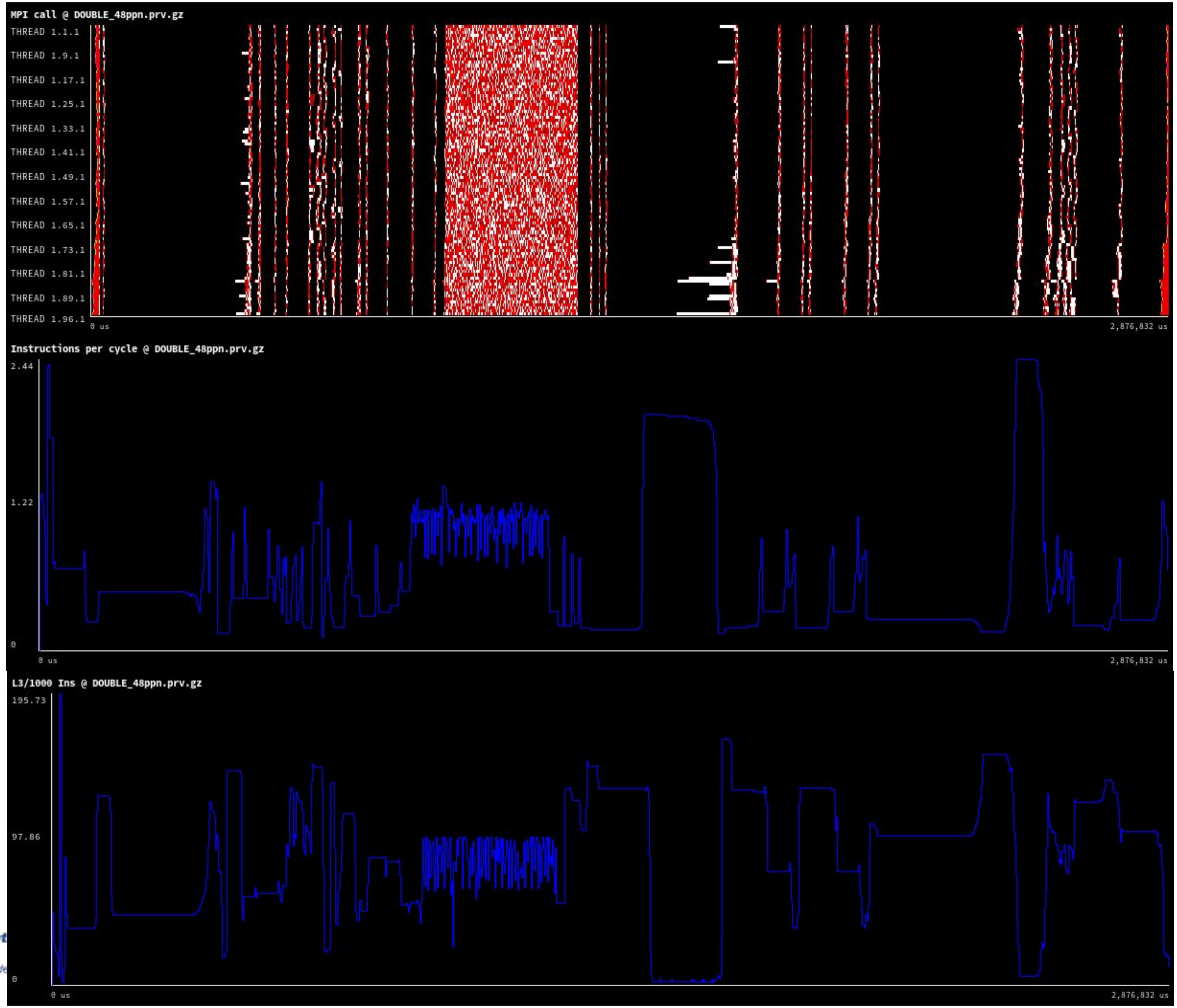


PAPI Counters

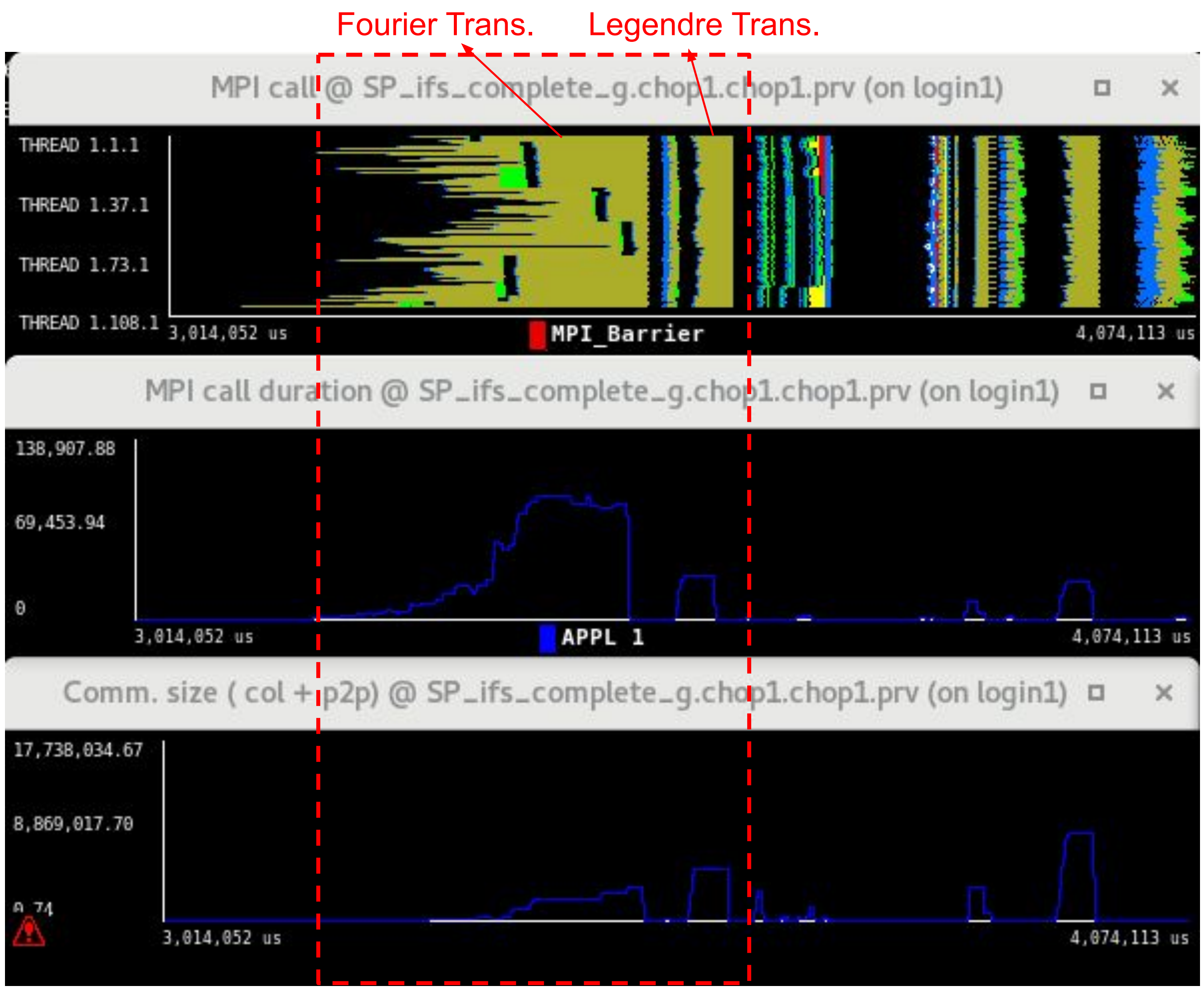
MPI Events

IPC

L1 Misses per 1000 INS

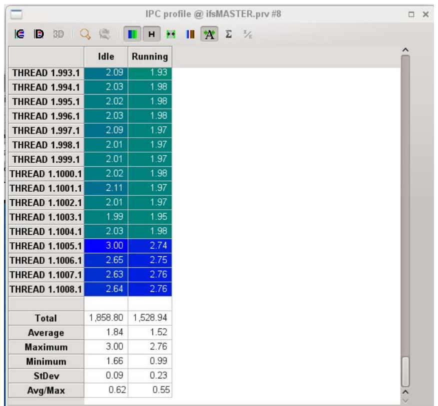


MPI evaluation



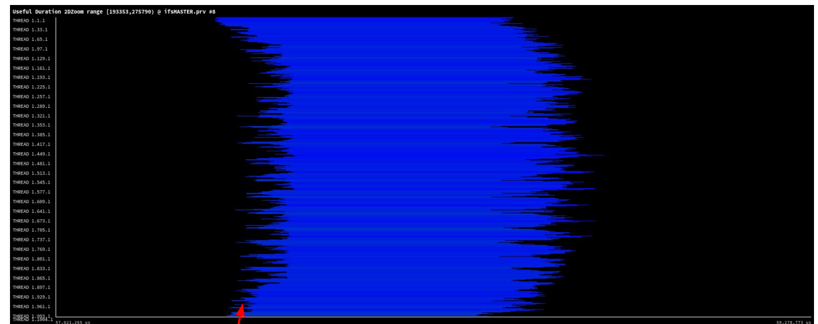
MPI evaluation

- IPC less than 1 for calculation areas?
- Are there load imbalance regions?



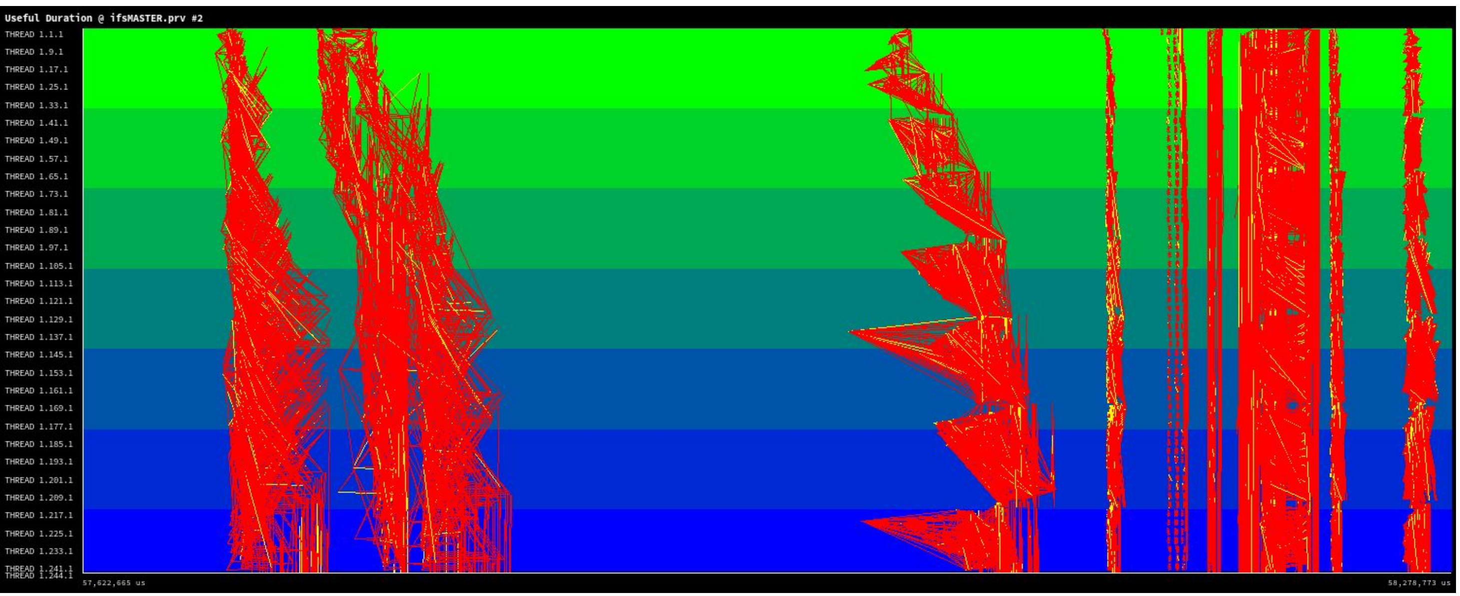
	Idle	Running
THREAD 1.993.1	2.09	1.93
THREAD 1.994.1	2.03	1.98
THREAD 1.995.1	2.02	1.98
THREAD 1.996.1	2.03	1.98
THREAD 1.997.1	2.09	1.97
THREAD 1.998.1	2.01	1.97
THREAD 1.999.1	2.01	1.97
THREAD 1.1000.1	2.02	1.98
THREAD 1.1001.1	2.11	1.97
THREAD 1.1002.1	2.01	1.97
THREAD 1.1003.1	1.99	1.95
THREAD 1.1004.1	2.03	1.98
THREAD 1.1005.1	3.00	2.74
THREAD 1.1006.1	2.65	2.75
THREAD 1.1007.1	2.63	2.76
THREAD 1.1008.1	2.64	2.76
Total	1,858.80	1,528.94
Average	1.84	1.52
Maximum	3.00	2.76
Minimum	1.66	0.99
StDev	0.09	0.23
Avg/Max	0.62	0.55

IPC_Profile



MPI evaluation

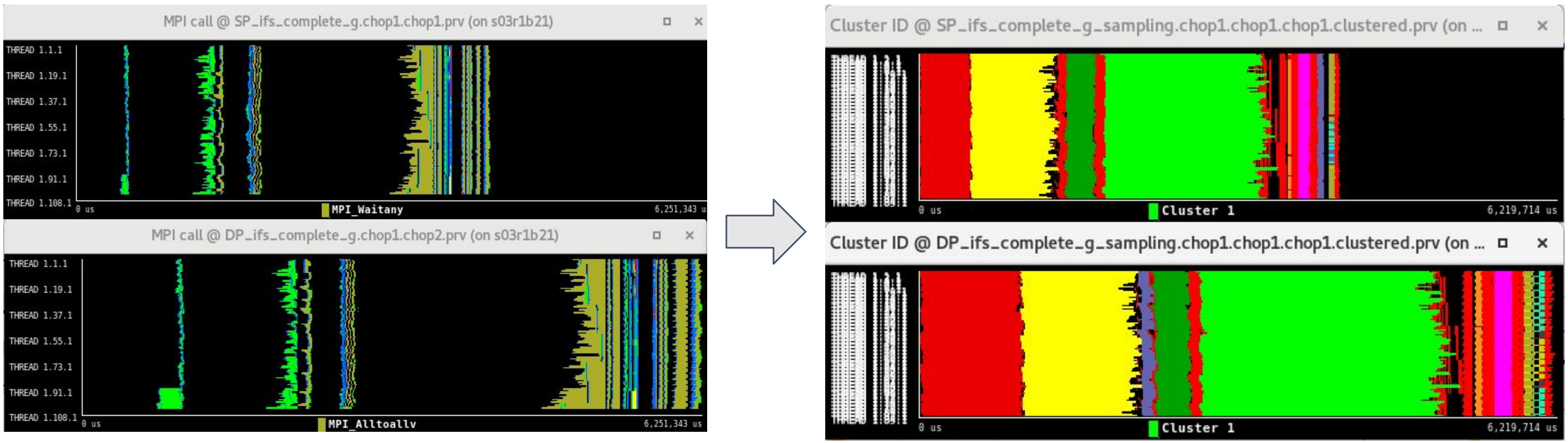
- Are MPI communications efficient according to the map affinity?



Affinity per node

Clustering Tool

Applying Clustering for an automatic profiling analysis

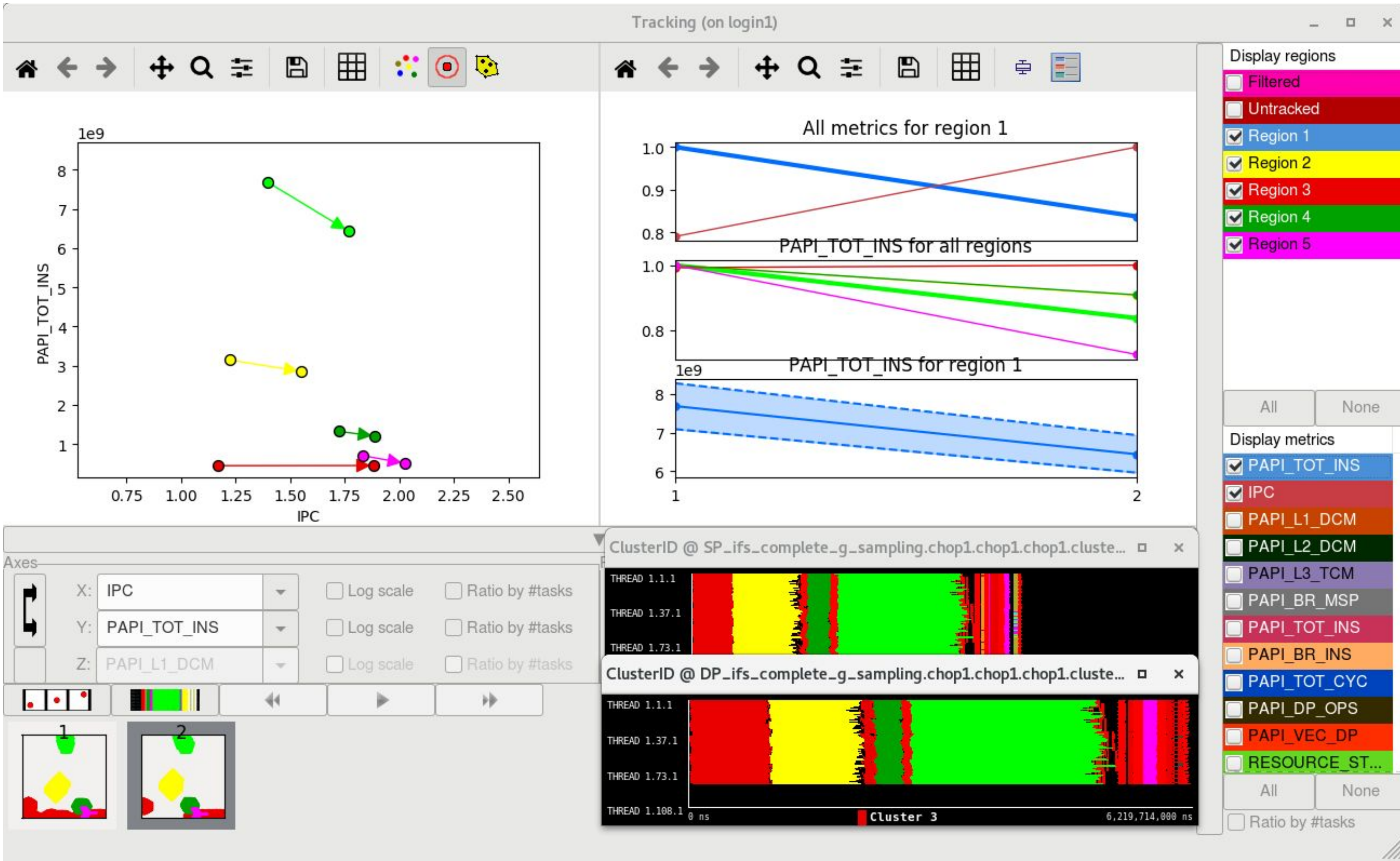


- Characterizes computing bursts that are similar and groups them into clusters
- Allows to study the behavior of the clusters separately, identify patterns, etc.

Tracking Tool

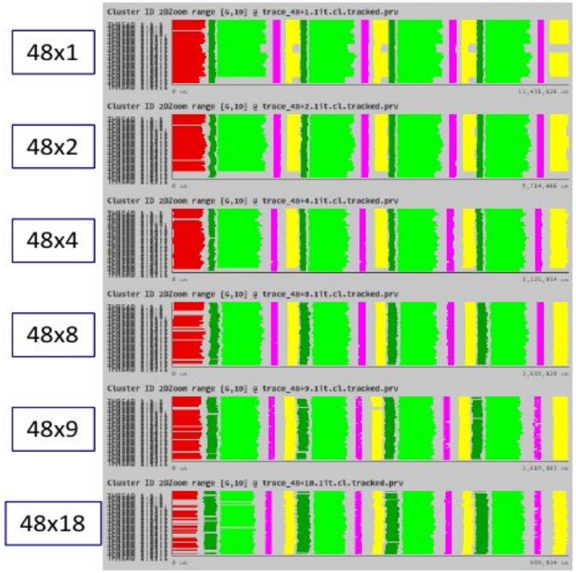
- A friendly way to quantify and visualize the evolution of the clusters among several traces
- The tool has 2 parts
  - Recognition algorithm of “who-is-who”, based on heuristics
  - A visualization GUI
- Examples analyzing multiple traces
  - Scaling number of MPI/OpenMP resources (64 – 128 – 256…)
  - Testing different microarchitecture features
  - Changing the problem size
  - Trying different compiler optimizations

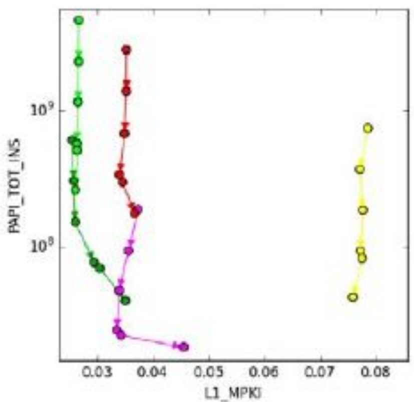
Tracking Tool

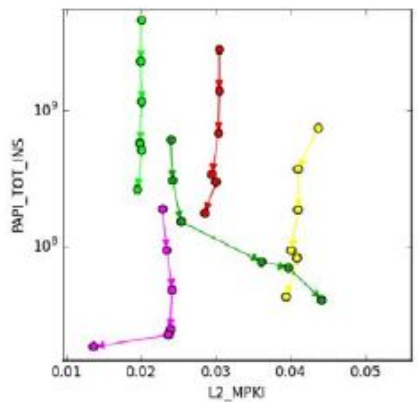


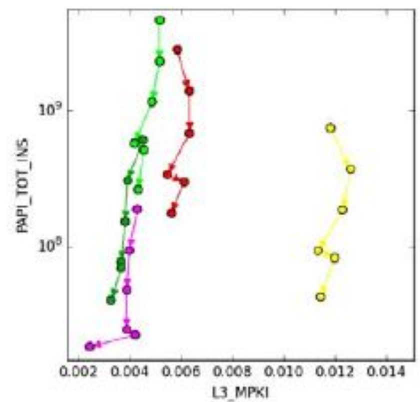
Tracking Tool

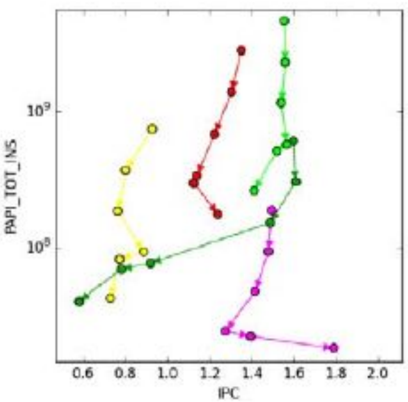
Tracking IFS MPI+OMP Strong Scaling

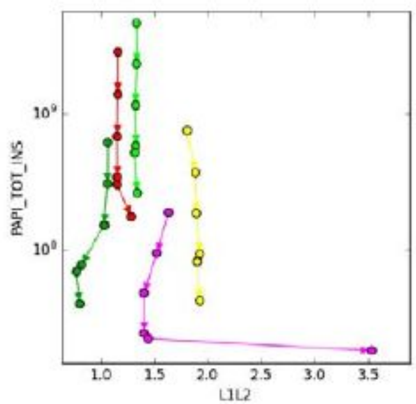


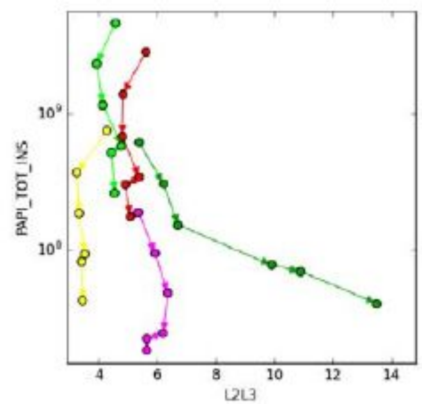








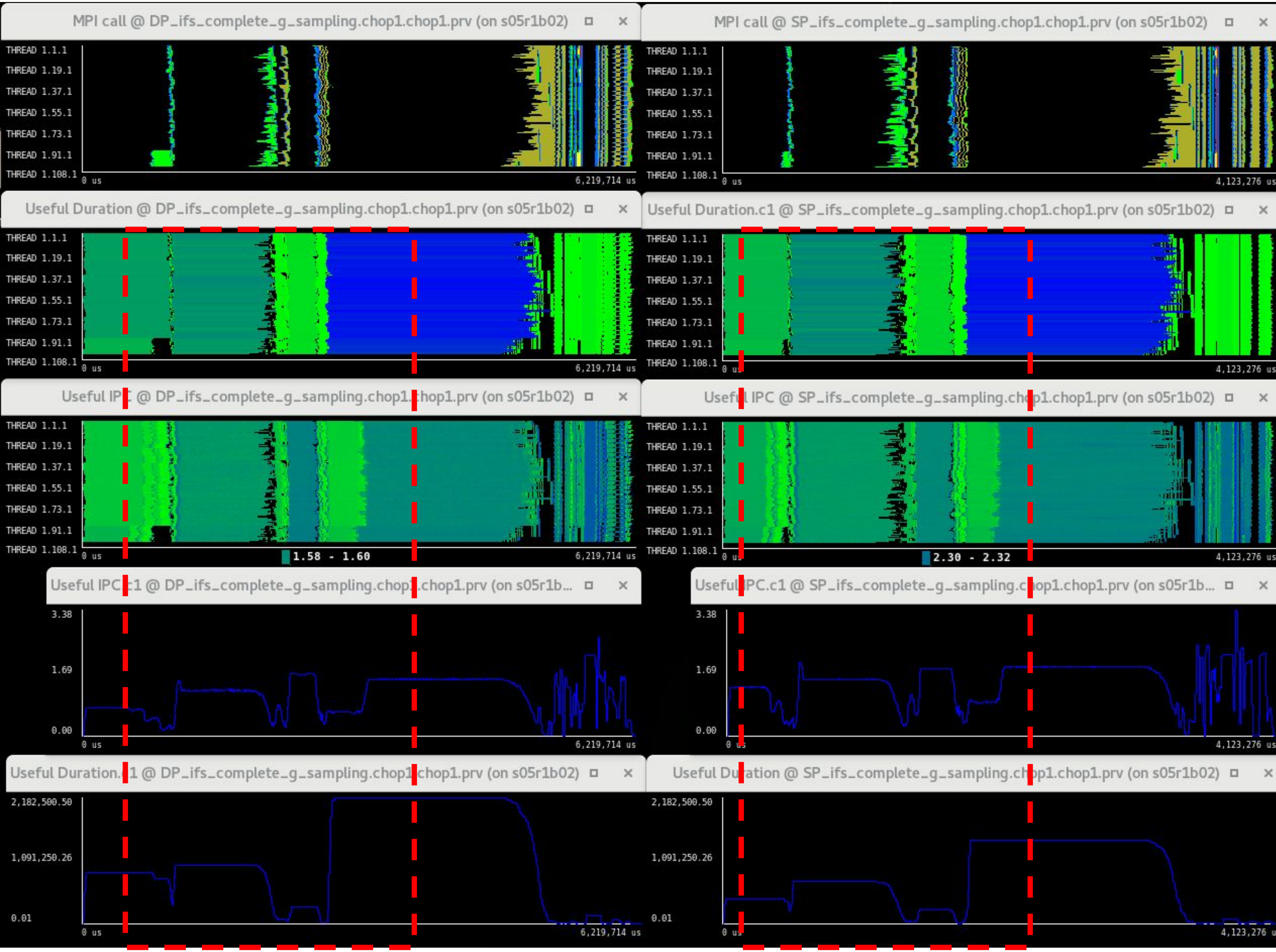




Sampling Tool

- Extrae can be configured to capture performance metrics on a periodic basis using alarm signals and specifying period and variability (10 and 2 respectively for IFS and NEMO tests).
- This means that we will capture samples every 10 ms with a random variability of 2 ms.

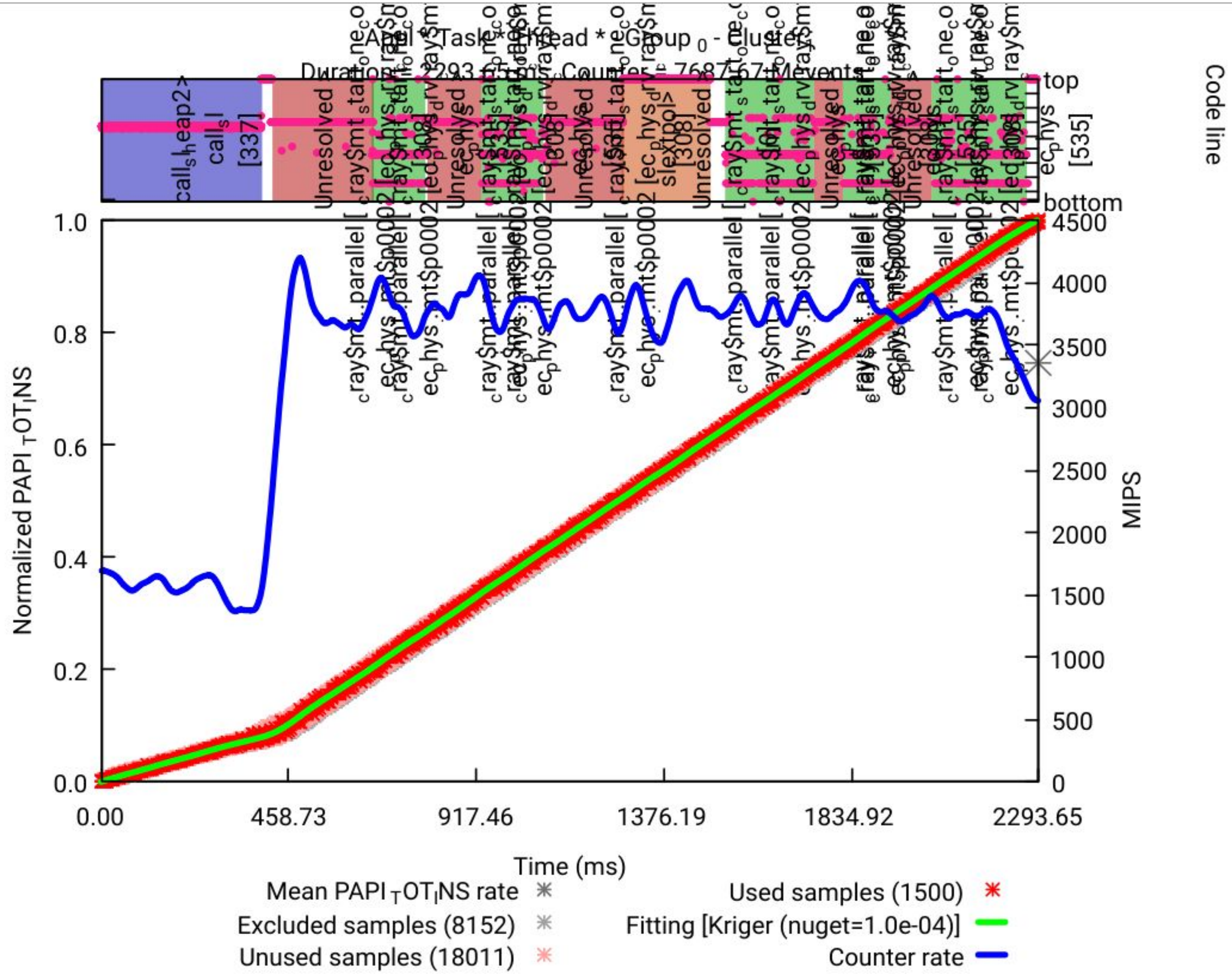
- Every sample contains processor performance counters (where every PAPI counter is referred at configured time) and callstack information.


MPI call @ DP_ifs_complete_g_sampling.chop1.chop1.prv (on s05r1b02)
MPI call @ SP_ifs_complete_g_sampling.chop1.chop1.prv (on s05r1b02)
THREAD 1.1.1
THREAD 1.19.1
THREAD 1.37.1
THREAD 1.55.1
THREAD 1.73.1
THREAD 1.91.1
THREAD 1.108.1
0 us
6,219,714 us
4,123,276 us
Useful Duration @ DP_ifs_complete_g_sampling.chop1.chop1.prv (on s05r1b02)
Useful Duration.c1 @ SP_ifs_complete_g_sampling.chop1.chop1.prv (on s05r1b02)
Useful IPC @ DP_ifs_complete_g_sampling.chop1.chop1.prv (on s05r1b02)
Useful IPC @ SP_ifs_complete_g_sampling.chop1.chop1.prv (on s05r1b02)
1.58 - 1.60
2.30 - 2.32
Useful IPC.c1 @ DP_ifs_complete_g_sampling.chop1.chop1.prv (on s05r1b...
Useful IPC.c1 @ SP_ifs_complete_g_sampling.chop1.chop1.prv (on s05r1b...
3.38
1.69
0.00
Useful Duration.c1 @ DP_ifs_complete_g_sampling.chop1.chop1.prv (on s05r1b02)
Useful Duration @ SP_ifs_complete_g_sampling.chop1.chop1.prv (on s05r1b02)
2,182,500.50
1,091,250.26
0.01

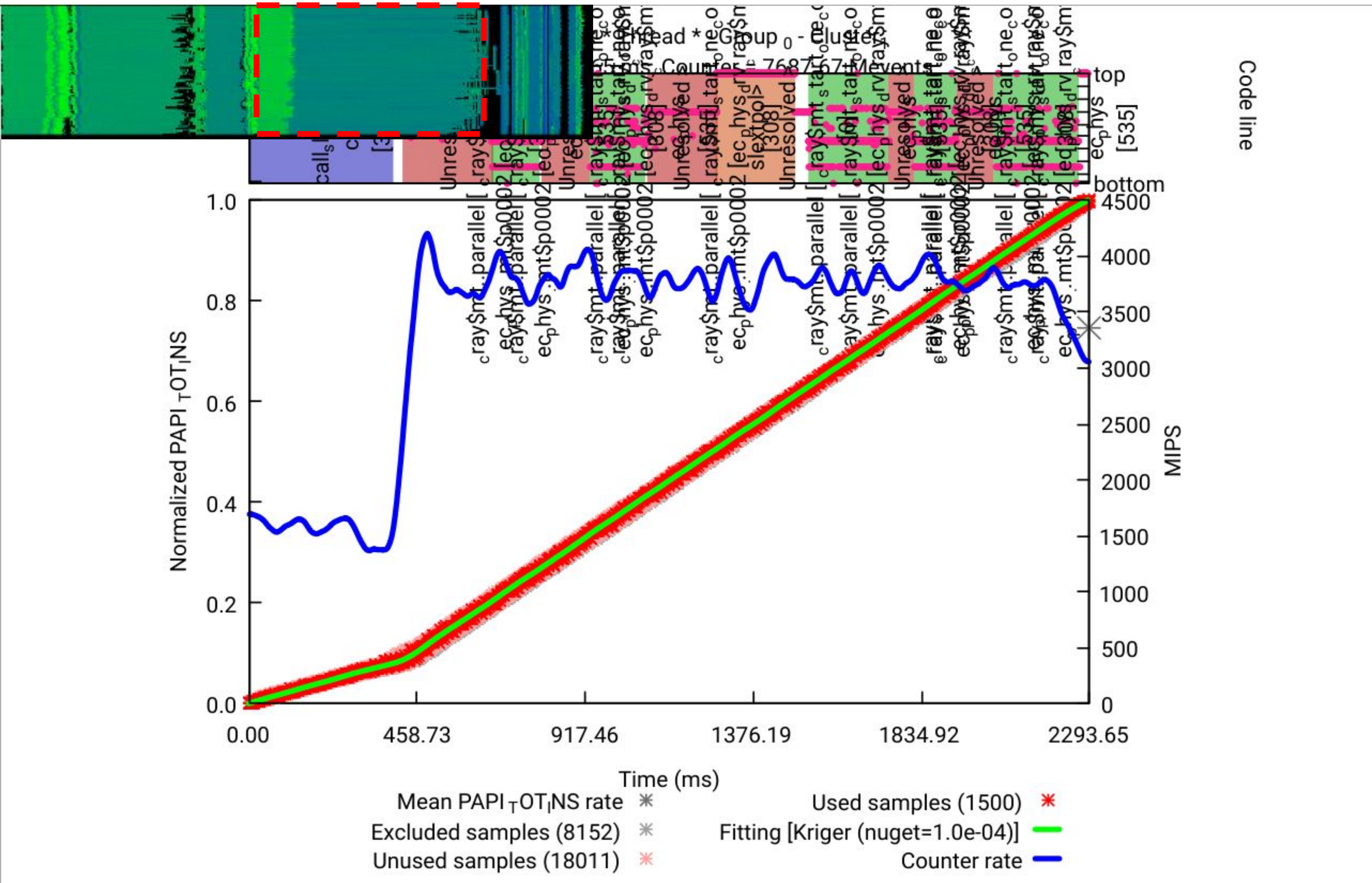
Folding Tool

- Combine instrumentation and sampling to provide **instantaneous performance metrics**, source code and memory references. This mechanism receives a trace-file and generates plots showing the fine evolution of the performance.
- The samples collected are gathered from scattered computing regions into a synthetic region by preserving their relative time within their original region so that the sampled information determines how the performance evolves within the region.
- The performance evolution is connected to source code and memory references at the same time.

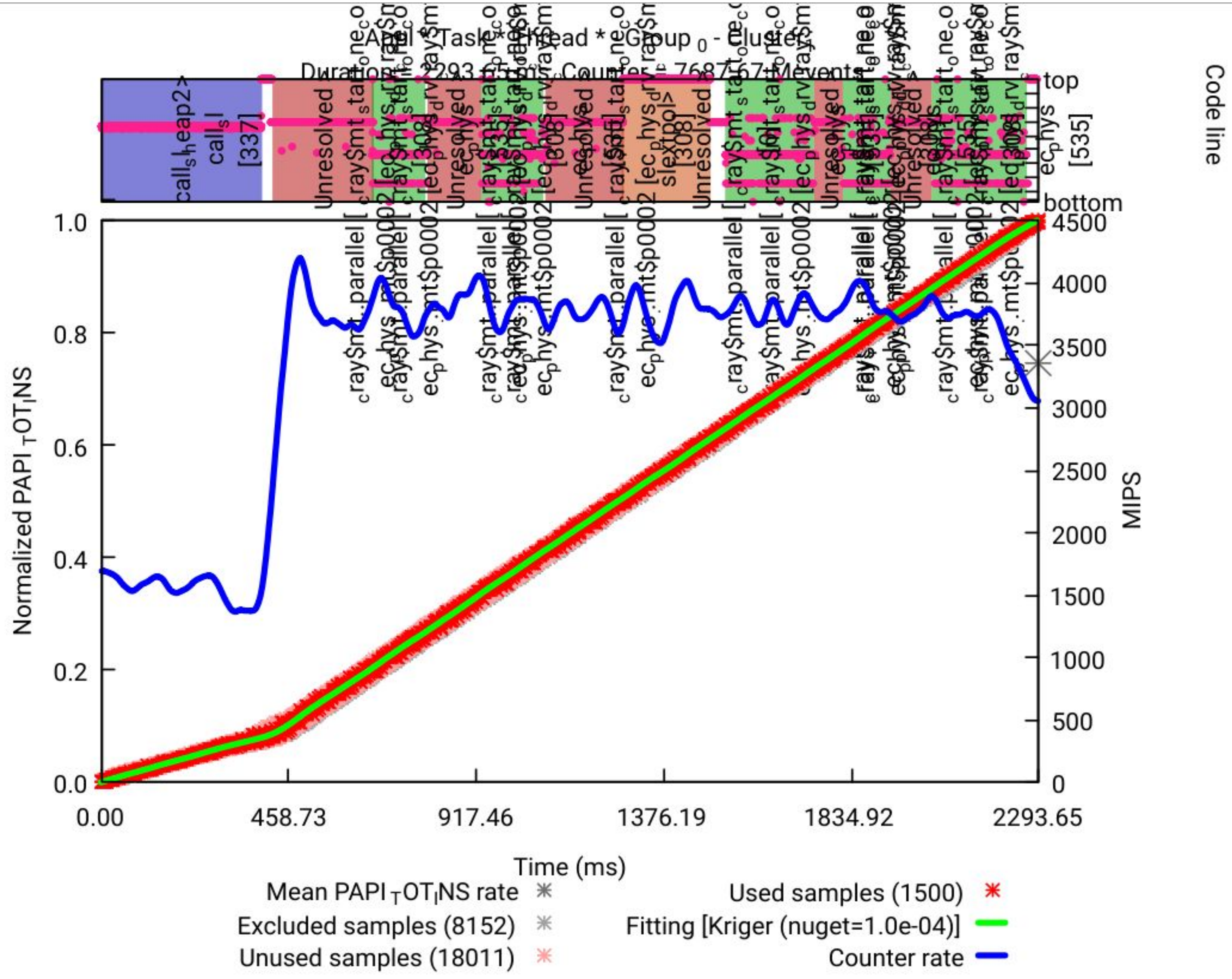
Folding Tool



Folding Tool



Folding Tool



Folding Tool

TOT_INS

TOT_CACHE_MISSES

Connection to the code

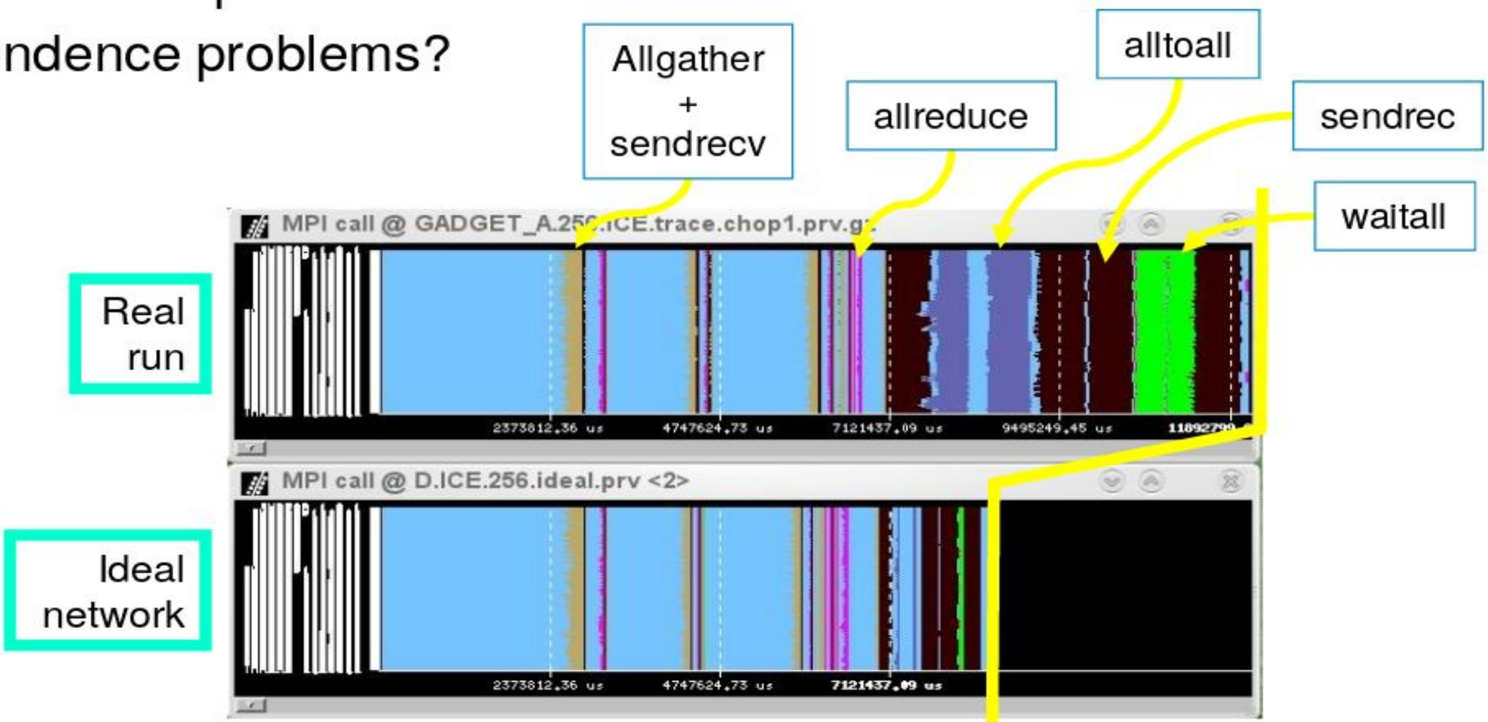
USER_FUNCTION_LINE

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

DIMEMAS Tool

The impossible machine: BW = ∞, L = 0

- Actually describes/characterizes intrinsic application behavior
  - Load balance problems?
  - Dependence problems?

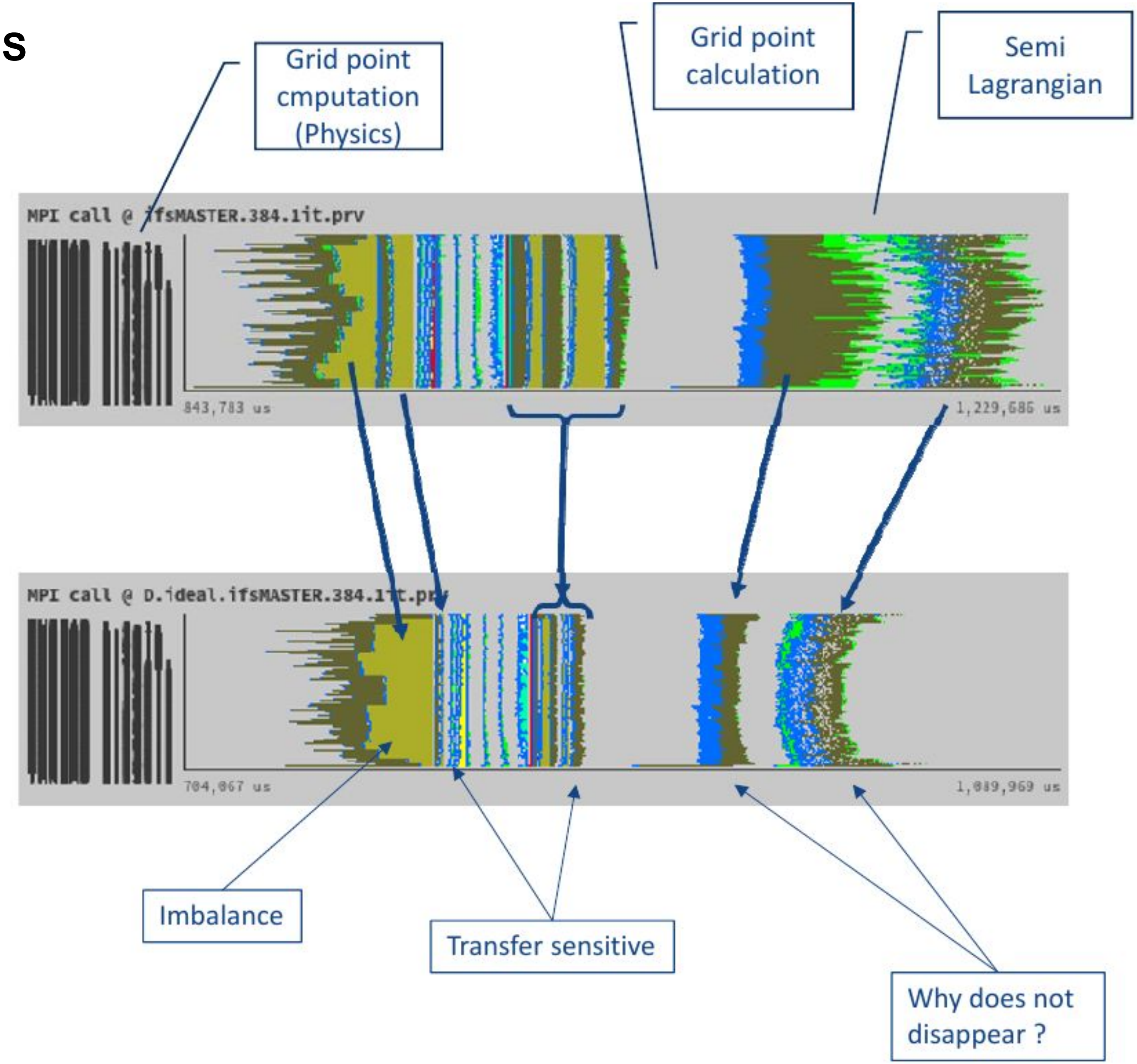


Impact on practical machines?

DIMEMAS Tool

Ideal Network for IFS execution

- Actual run
- Ideal network



Profiling Methodology

- Area of study
- Deployment efficiency
- Benchmarking
- Profiling analysis
- **Validation**
    - Reproducibility Test
    - Validation Test

Validation

Reproducibility Test: Are your results comparable to the EC-Earth community results?

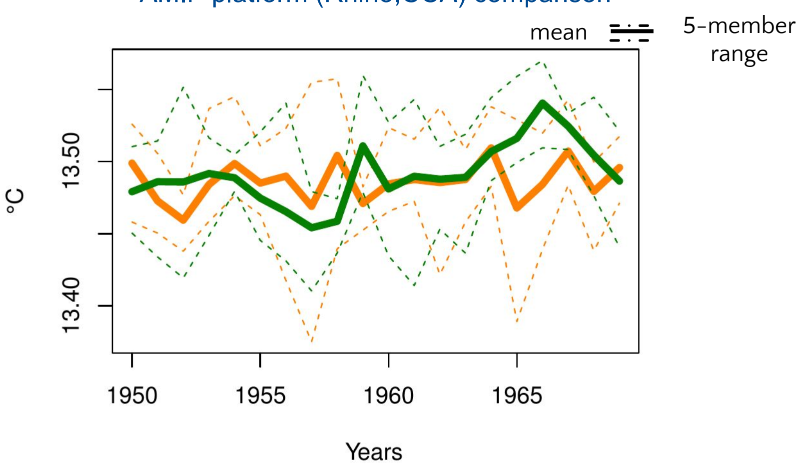
The Test proposed:

20-yr long, 5-member, Forcing Fixed Cmip and Amip simulations

- 20-yr long: Allows to look at impact of machine on mean state/bias (not possible in the case of 1-yr simulations)
- 5-member: Allows to measure differences due hardware as compared to internal variability
- Forcing Fixed: Working under stationary conditions removes possible dependence of hardware impact on the mean state
- Cmip and Amip simulations: Addresses the problem from a global point-of-view; suitable to give recommendations for CMIP6

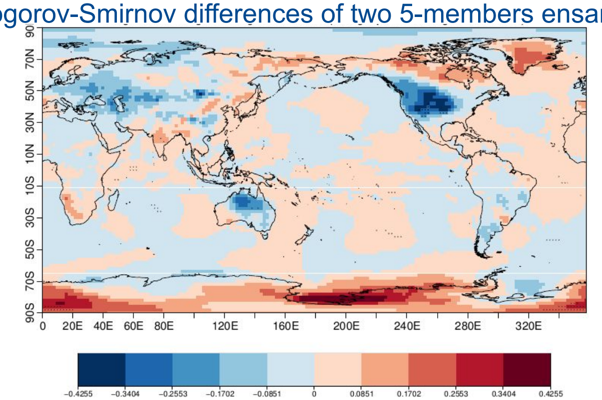
The results comparing platforms or configurations:

AMIP platform (Rhino;CCA) comparison



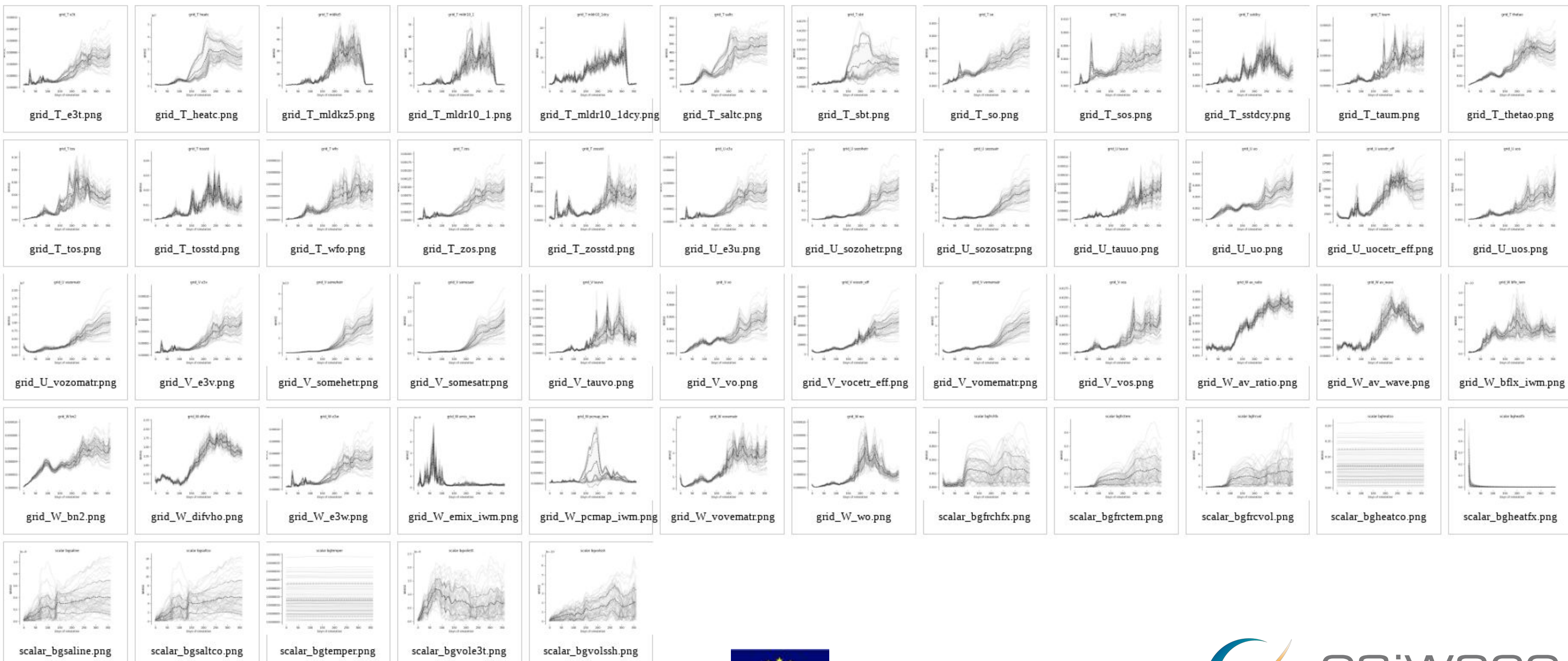
AMIP platform (Rhino;CCA) comparison

Kolmogorov-Smirnov differences of two 5-members ensambles



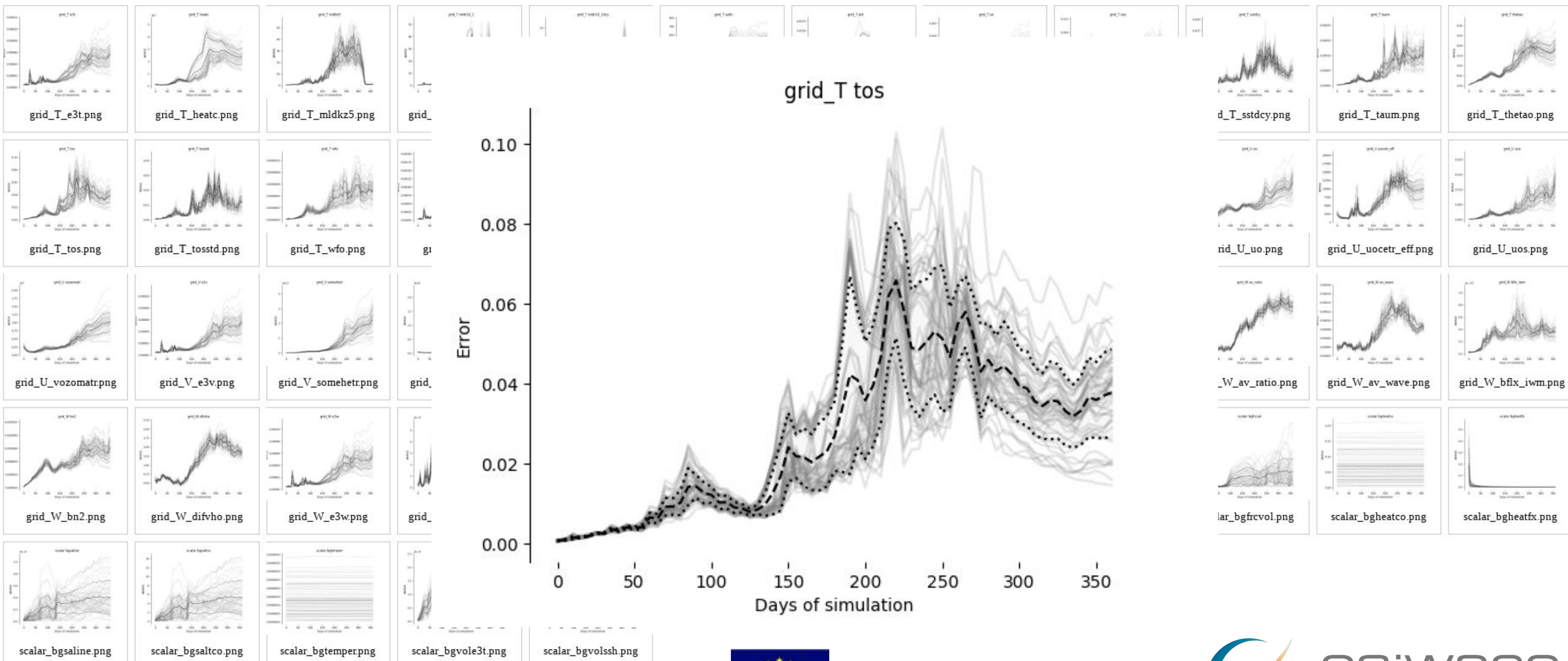
Validation Test (NEMO)

- Initial conditions perturbed with white noise in the 3D temperature field.
- Evaluating 53 output variables.



Validation Test (NEMO)

- Initial conditions perturbed with white noise in the 3D temperature field.
- Evaluating 53 output variables.

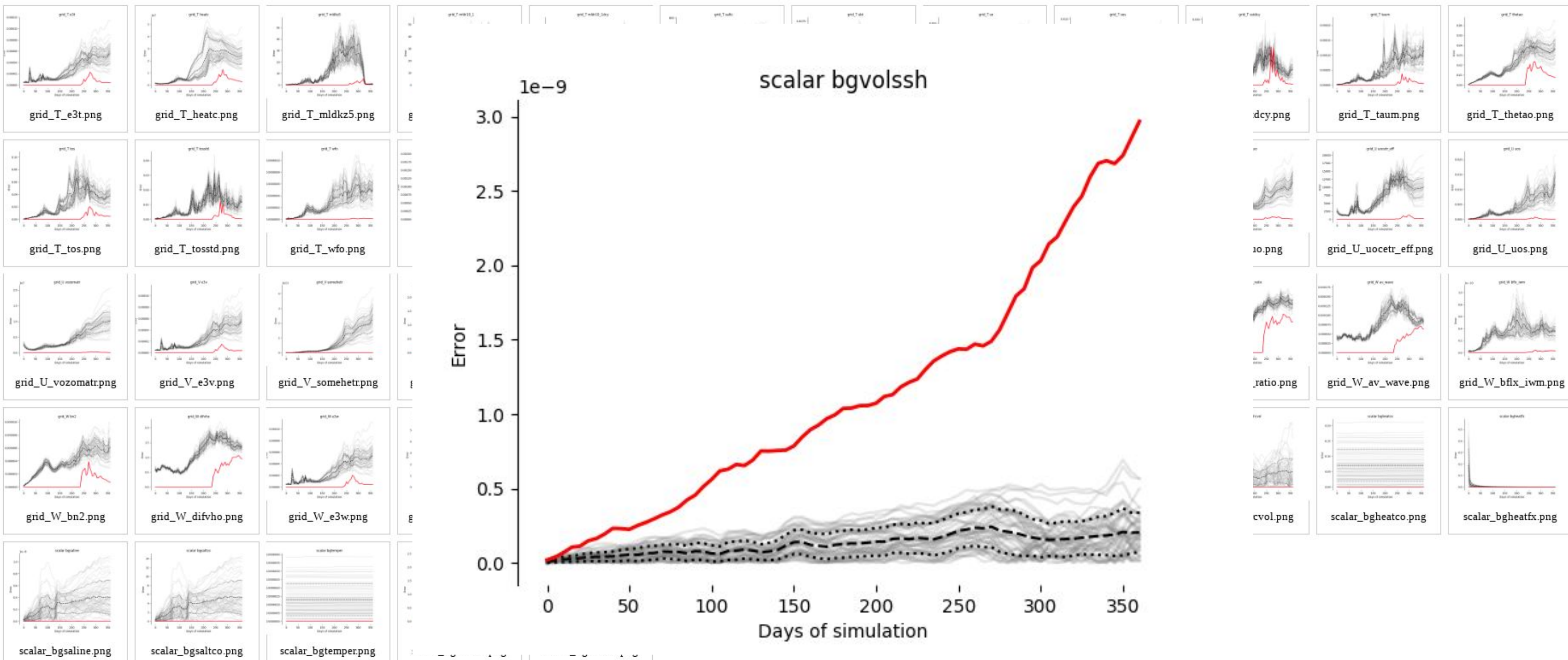


Supercomputing Center
Centro Nacional de Supercomputación

esiwace
CENTRE OF EXCELLENCE IN SIMULATION OF WEATHER AND CLIMATE IN EUROPE

Validation Test (NEMO)

Example: Compiling with -xHost



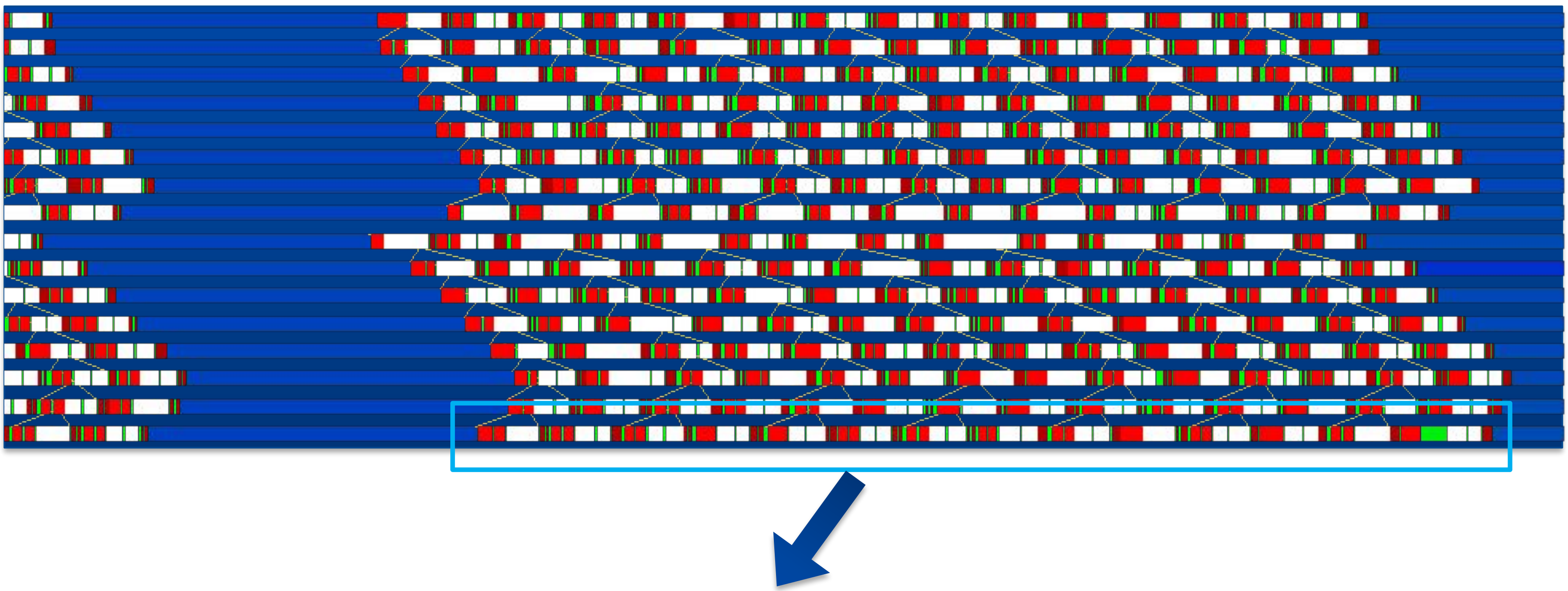
Examples

Function
Timelines

Barcelona
Supercomputing
Center
Centro Nacional de Supercomputación

esiwace
CENTRE OF EXCELLENCE IN SIMULATION OF WEATHER
AND CLIMATE IN EUROPE

Examples



Border Exchange

Examples

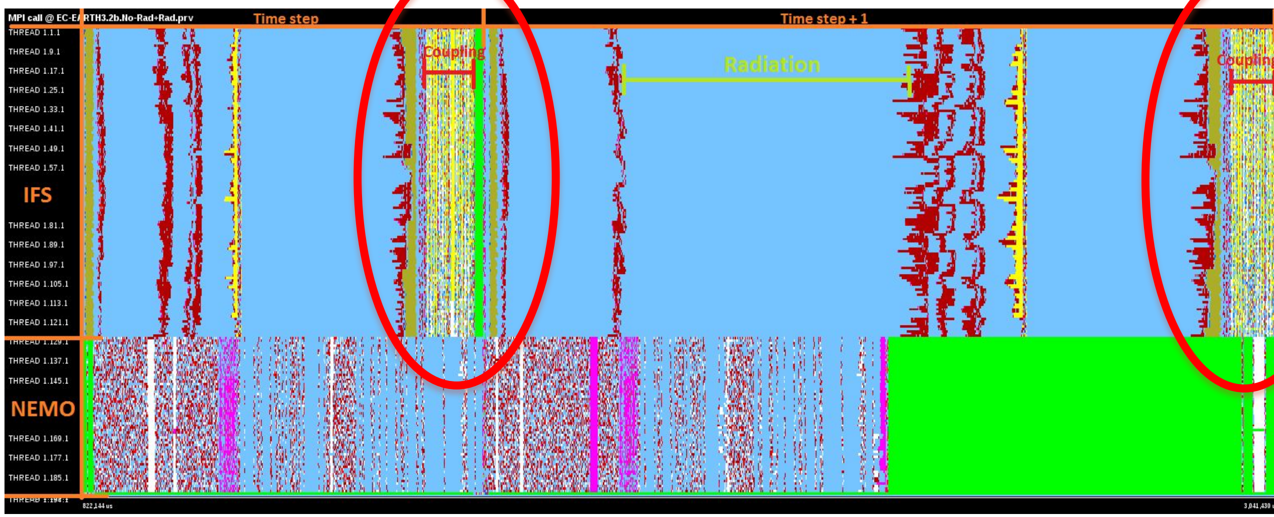
- Diagnostic for NEMO:
  - Scalability is constrained by:
    - 1) Algorithms with too much communication
    - 2) Sub-optimal implementation
- Actions taken
  - Improve communication implementation to reduce number of point-to-point messages
  - Reduce number of collectives

Examples

- First studies showed that IFS-NEMO coupling was not a big issue

IFS: 128 cores

NEMO: 128 cores



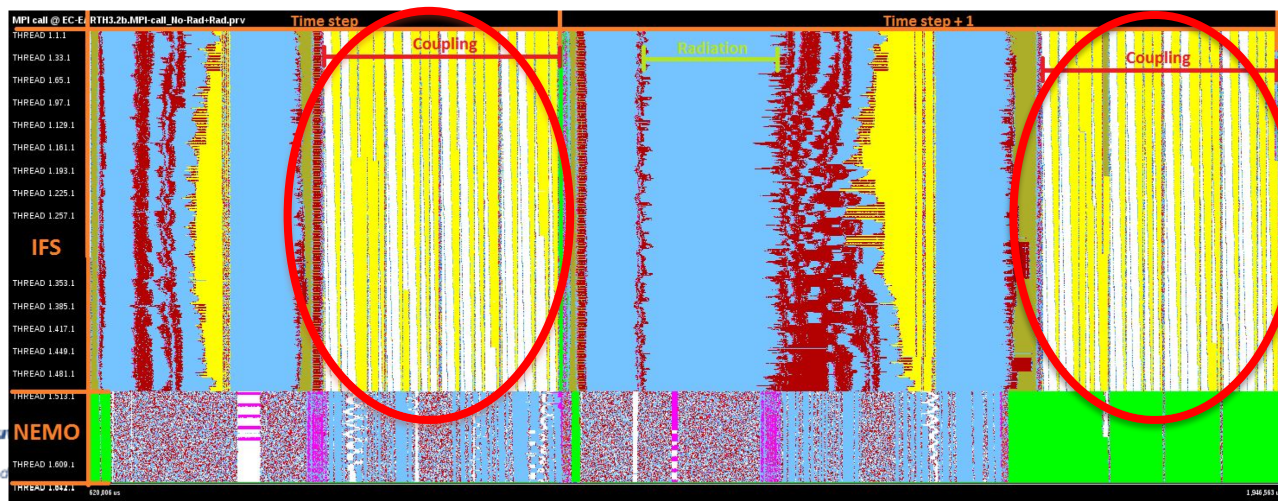
Coupling time is increased

TIME

- But it seems that it is when increasing number of cores

IFS: 512 cores

NEMO: 128 cores

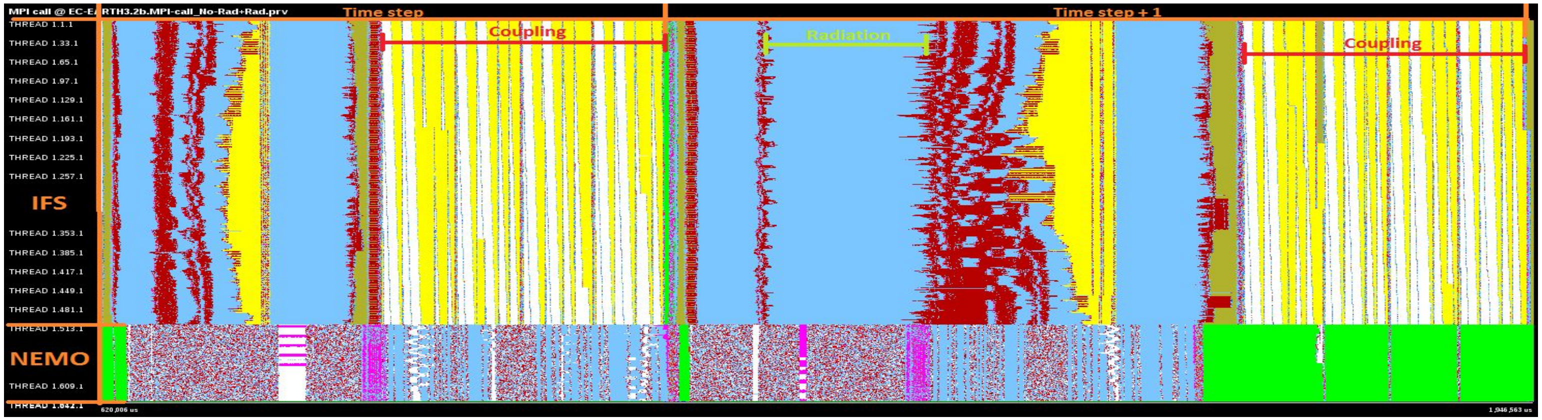


Barcelona Supercompu Center Centro Nacional d

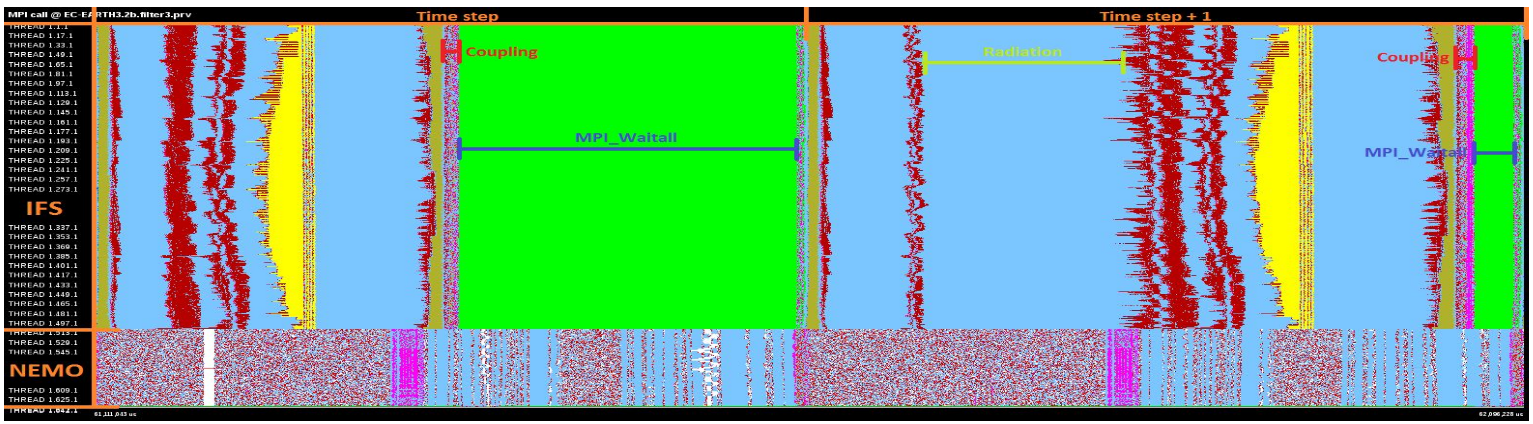
esiwace
ENTRE OF EXCELLENCE IN SIMULATION OF WEATHER
ND CLIMATE IN EUROPE

Examples

- BSC has been working successfully with the EC-Earth Technical Working Group to improve the execution of the model



- A success case: coupling field gathering and OPT option of OASIS coupler for global conservative transformations



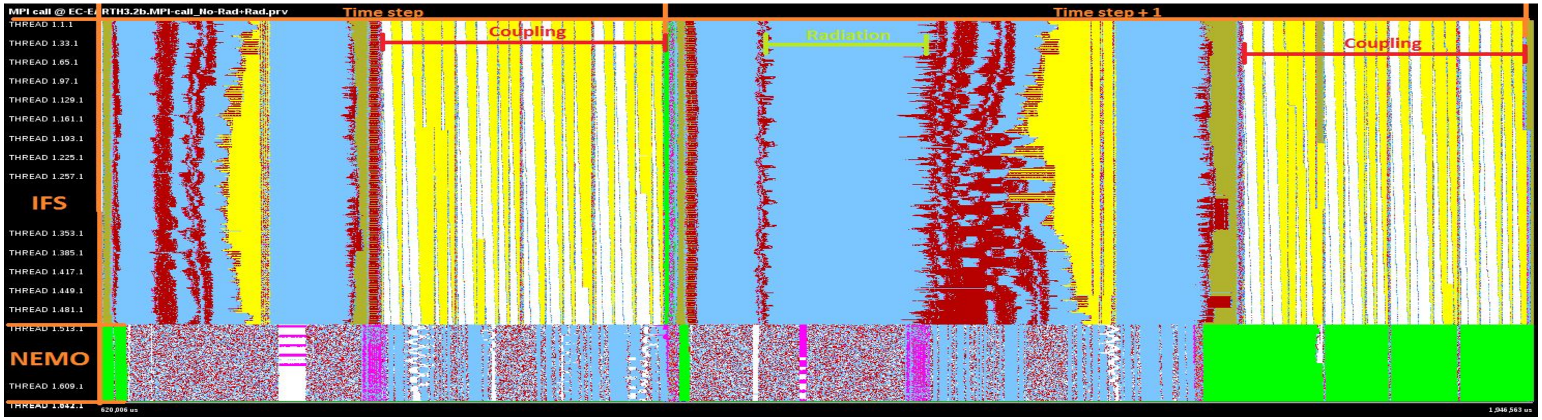
- With these optimizations, up to 90% improvement in coupling process can be achieved
- These improvements are now in trunk EC-Earth 3.2.2, substantially benefiting our CMIP6 simulations

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

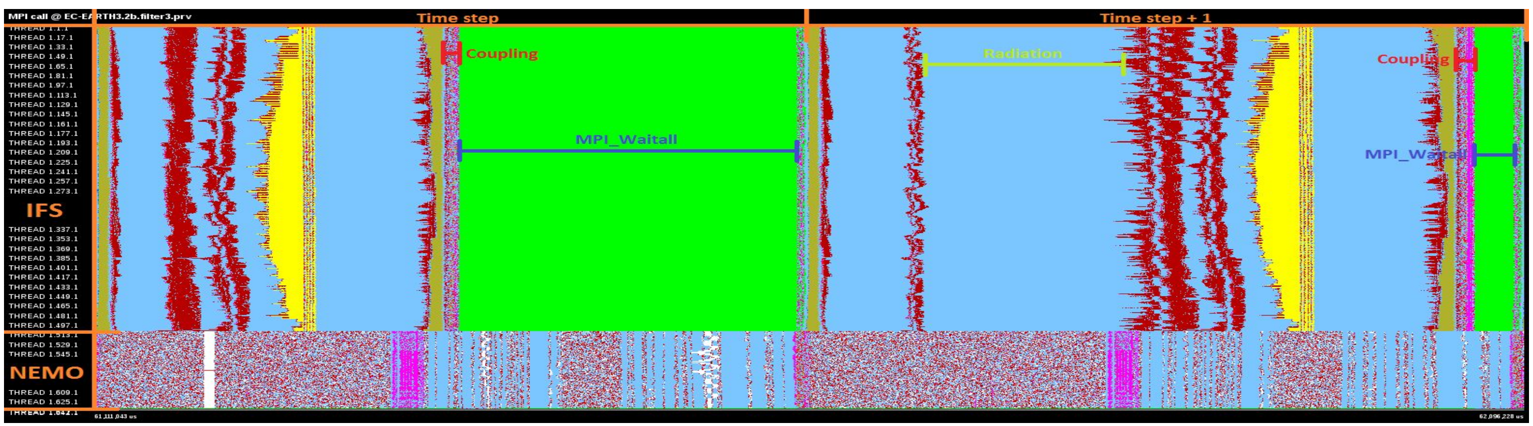
esiwace
CENTRE OF EXCELLENCE IN SIMULATION OF WEATHER AND CLIMATE IN EUROPE

Examples

- BSC has been working successfully with the EC-Earth Technical Working Group to improve the execution of the model



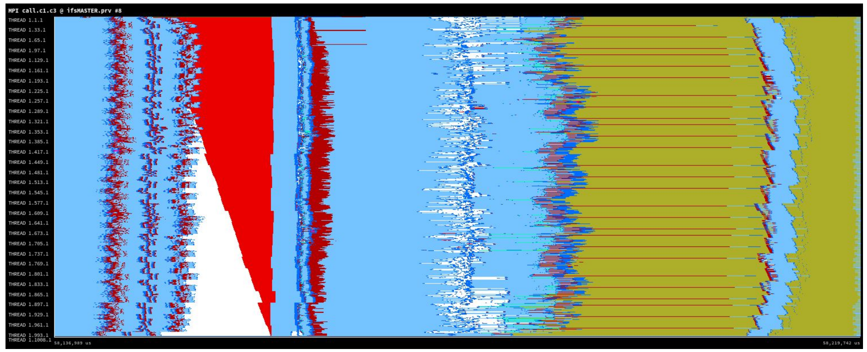
- A success case: coupling field gathering and OPT option of OASIS coupler for global conservative transformations



- With these optimizations, up to 90% improvement in coupling process can be achieved
- These improvements are now in trunk EC-Earth 3.2.2, substantially benefiting our CMIP6 simulations

Examples

- Synchronal point to point communication could be a bottleneck even for only one message from one master to hundreds of slaves
    - Sigcheck method



MPI call profile @ ifsMASTER.prv #8

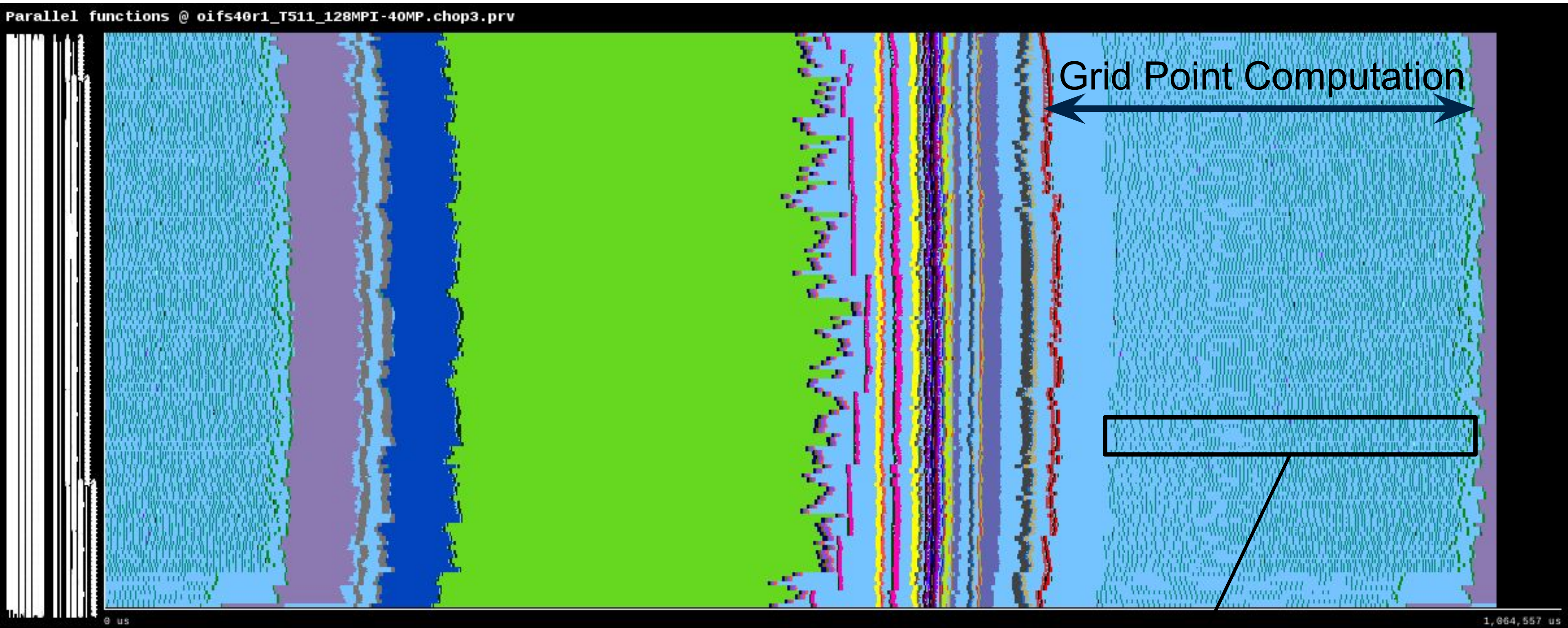
89 (sigcheck.F90, ifsMASTER)	95 (sigcheck.F90, ifsMASTER)	111 (sigcheck.F90, ifsMASTER)
0.81 %	442.74 %	806.93 %
0.81 %	0.44 %	0.80 %
0.81 %	1.73 %	1.84 %
0.81 %	0.00 %	0.00 %
0 %	0.45 %	0.41 %
1	0.25	0.44

- Using one asynchronal collective communication this time is reduced almost to 0

Examples

- Hybrid Test (128 MPI+4 OpenMP, Total: 512)

OpenMP Parallel Regions



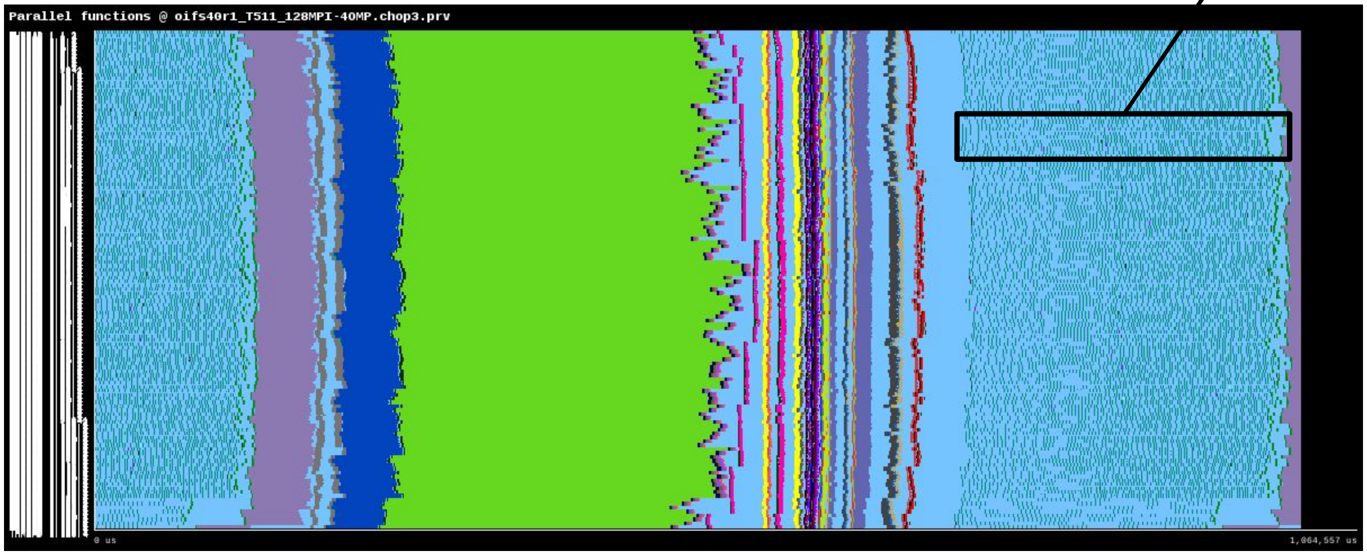
One Complete Time Step

Very small granularity of the OpenMP parallel Regions

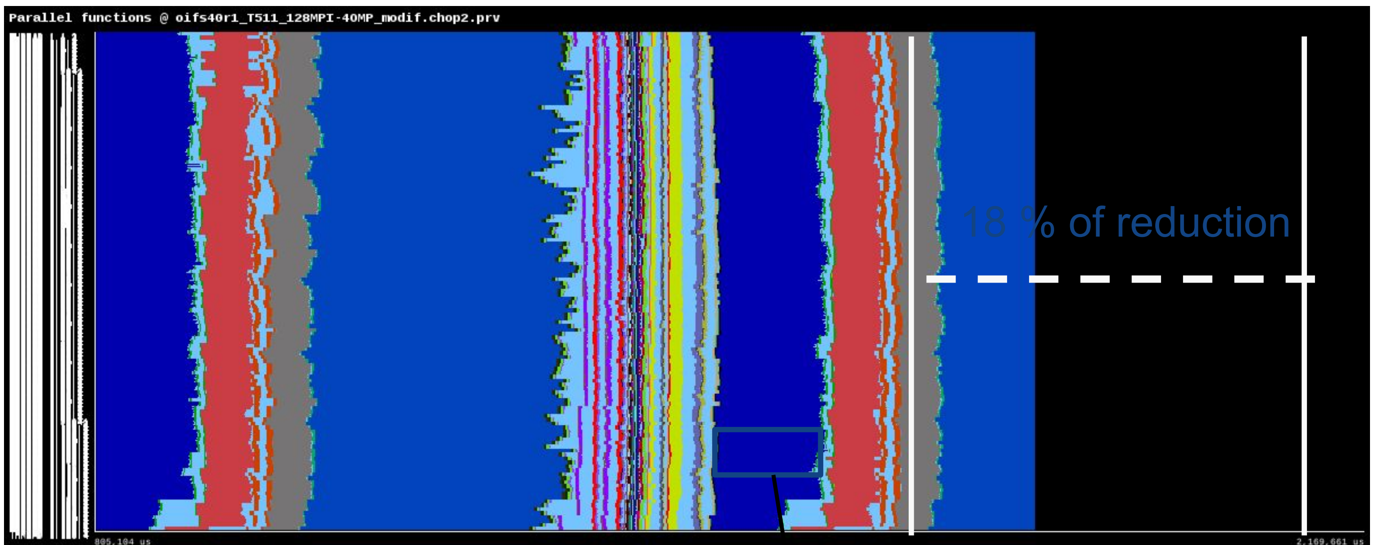
Examples

Small OpenMP parallel Regions

128 MPI processes and 4 OpenMP threads per process



128 MPI processes and 4 OpenMP threads per process



Only one coarse OpenMP parallel region

Barcelona
Supercomputing
Center
Centro Nacional de Supercomputación

EXCELENCIA
SEVERO
OCHOA

esiwace
CENTRE OF EXCELLENCE IN SIMULATION OF WEATHER AND CLIMATE IN EUROPE

Thank you

The research leading to these results has received funding from the EU H2020 Framework Programme under grant agreement H2020 GA 675191.

mario.acosta@bsc.es