The BSC Tools: Extrae and Paraver

Based on slides by Judit Gimenez, BSC

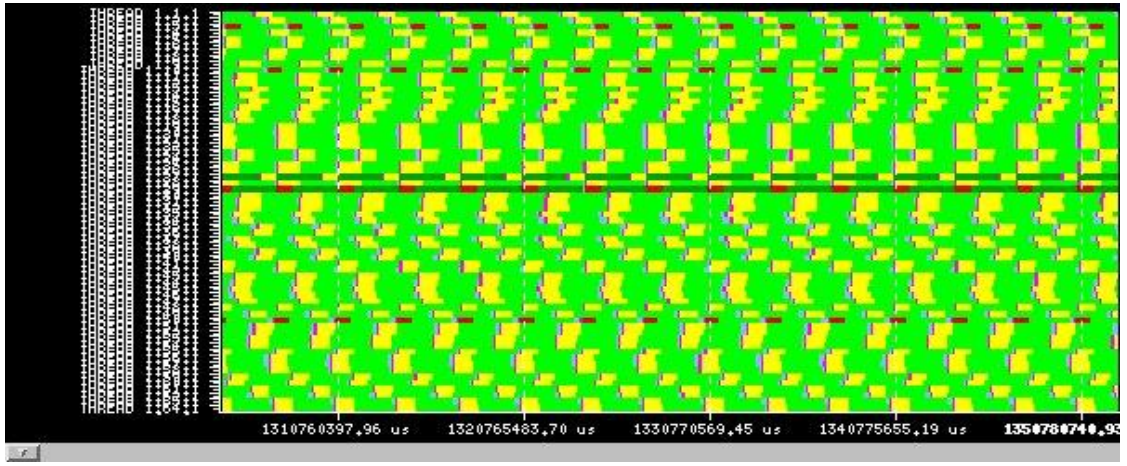
EU H2020 Centre of Excellence (CoE)

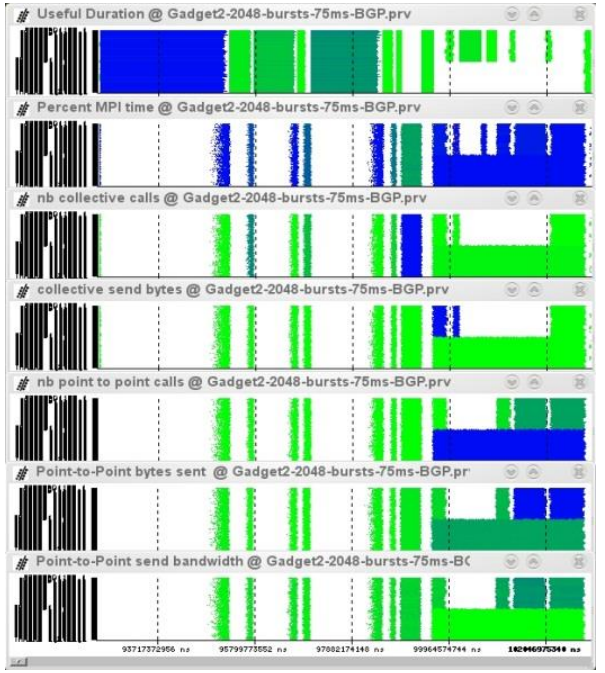
1 October 2015 – 31 March 2018

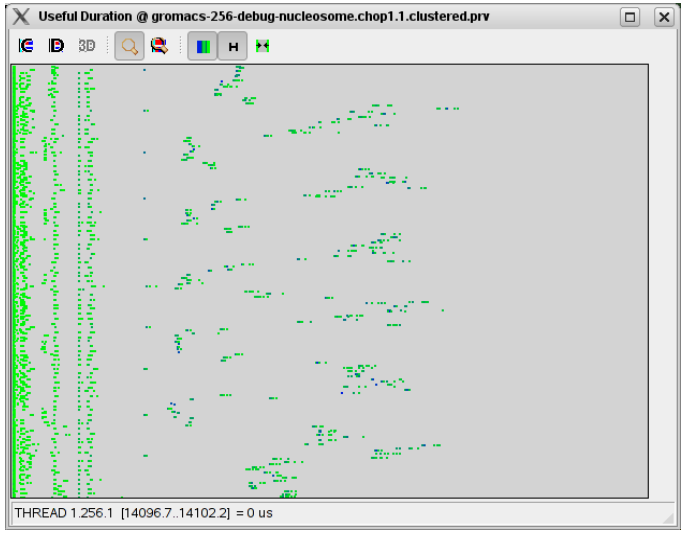
Grant Agreement No 676553

The BSC Tools

- Since 1991
- Based on traces
- Open Source
  - http://tools.bsc.es
- Core tools:
  - Extrae – instrumentation
  - Paraver (paramedir) – offline trace analysis
  - Dimemas – message passing simulator
- Focus
  - Detail, variability, flexibility
  - Behavioral structure vs. syntactic structure
  - Intelligence: Performance Analytics







Extrae

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!

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
- 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
  - https://tools.bsc.es/tools_manuals

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

train.sh

```
(...)

### Application path
PISVM=../../bin/pisvm-train

### Input path
TRAINDATA=../../input/sdap_area_all_training.el

### Extrae path
export EXTRAE_HOME=/homec/deep/deep83/tools/extrae/
export EXTRAE_WORK_DIR=/work/$USER/pisvm/romeraw

### Run the application
srun ./trace.sh $PISVM -D -o 1024 -q 512 -c 10000
                g 16 -t 2 -m 1024 -s 0 $TRAINDATA

### Generate the trace
export TRACE_NAME=pisvm-romeraw-train.prv
${EXTRAE_HOME}/bin/mpi2prv
   -f ${EXTRAE_WORK_DIR}/TRACE.mpits
   -o ${TRACE_NAME}
```

trace.sh

```
#!/bin/bash

### Load Extrae
source ${EXTRAE_HOME}/etc/extrae.sh
export EXTRAE_CONFIG_FILE=../../../../config/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
$*
```

Select tracing library

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 2: Extrae XML configuration

```
<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)

```
<counters enabled="yes">
  <cpu enabled="yes" starting-set-distribution="cyclic">
    <set enabled="yes" domain="all" changeat-time="0">
      PAPI_TOT_INS, PAPI_TOT_CYC, PAPI_L1_DCM, PAPI_L2_DCM
    </set>
    <set enabled="yes" domain="all" changeat-time="500000us">
      ...
    </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)

```
<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>
```

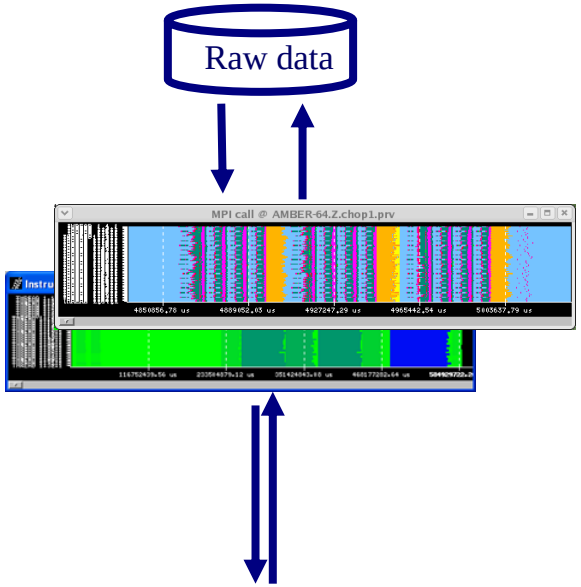
Trace buffer size

Collect more measurements with sampling

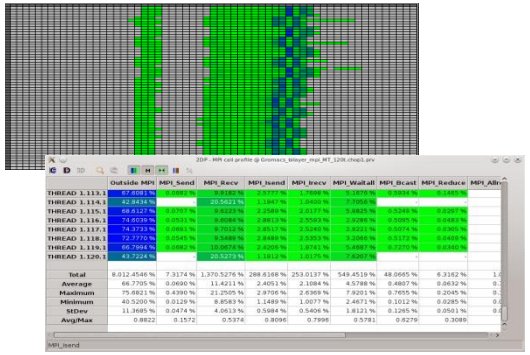
Merge intermediate files into Paraver trace

Paraver

Paraver – Performance data browser



Timelines



2/3D tables (Statistics)

Trace visualization/analysis

+ trace manipulation

Goal = Flexibility

No semantics

Programmable

Comparative analyses

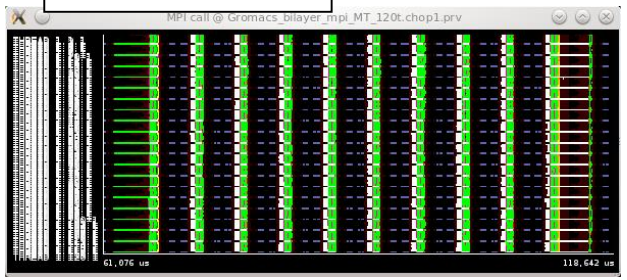
Multiple traces

Synchronize scales

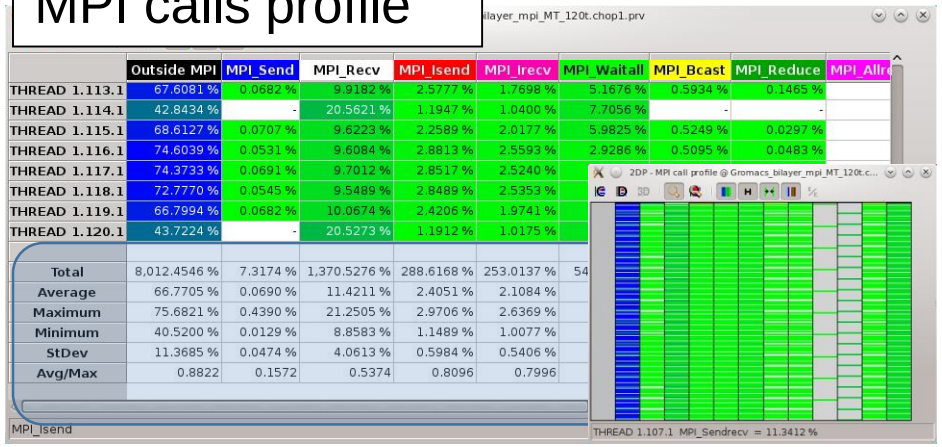
Tables: Profiles, histograms, correlations

- From timelines to tables

MPI calls

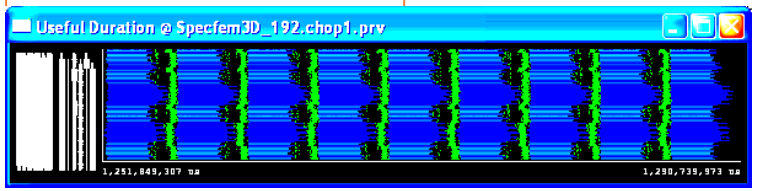


MPI calls profile

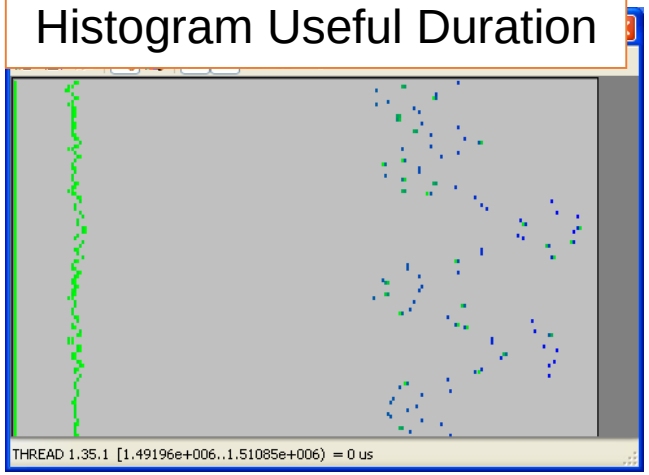


	Outside MPI	MPI_Send	MPI_Recv	MPI_Isend	MPI_Irecv	MPI_Waitall	MPI_Bcast	MPI_Reduce
THREAD 1.113.1	67.6081 %	0.0682 %	9.9182 %	2.5777 %	1.7698 %	5.1676 %	0.5934 %	0.1465 %
THREAD 1.114.1	42.8434 %	-	20.5621 %	1.1947 %	1.0400 %	7.7056 %	-	-
THREAD 1.115.1	68.6127 %	0.0707 %	9.6223 %	2.2589 %	2.0177 %	5.9825 %	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			

Useful Duration



Histogram Useful Duration

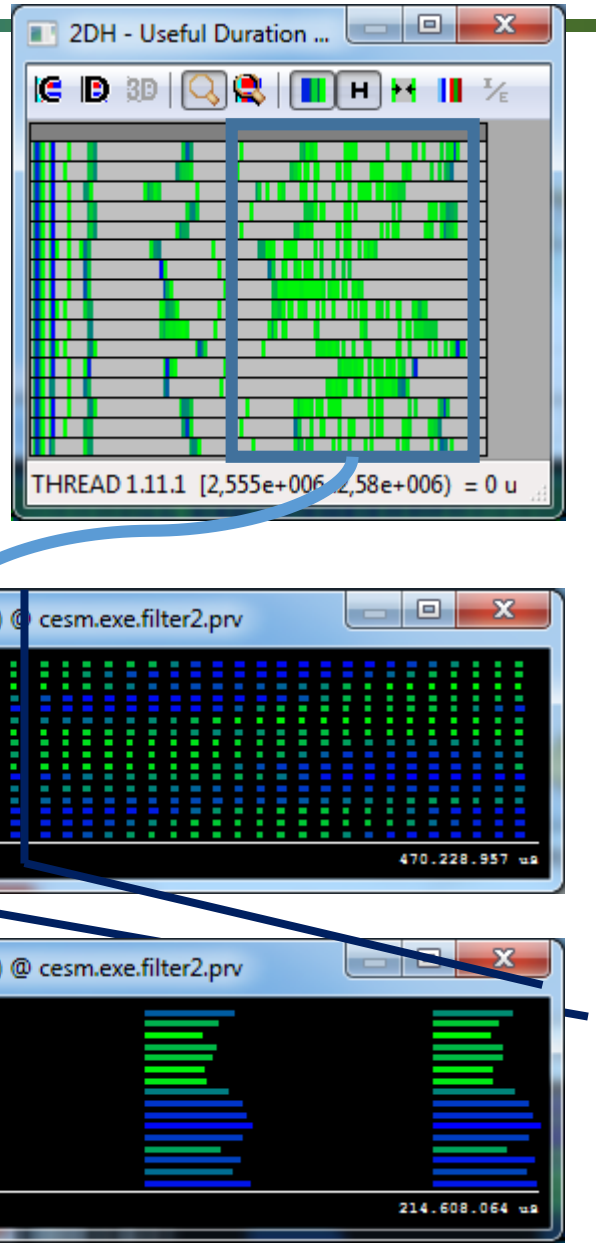


From tables to timelines

CESM: 16 processes, 2 simulated days

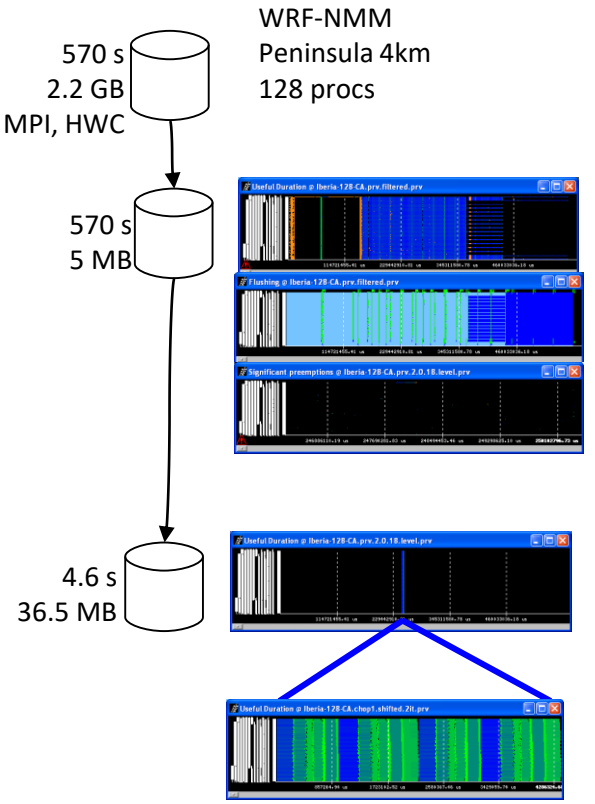
- Histogram useful computation duration shows high variability
- How is it distributed?

- Dynamic imbalance
  - In space and time
  - Day and night
  - Season ? ☺



Trace manipulation

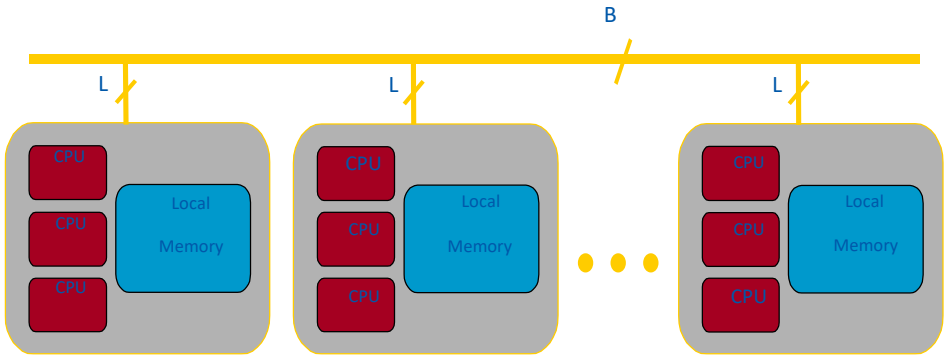
- Data handling/summarization capability
  - Filtering
    - Subset of records in original trace
    - By duration, type, value,…
    - Filtered trace is still a Paraver trace and can be analysed with the same cfgs (as long as the data required has been kept)
  - Cutting
    - All records in a given time interval
    - Only some processes
  - Software counters
    - Summarized values computed from those in the original trace emitted as new even types
    - #MPI calls, total hardware count,…

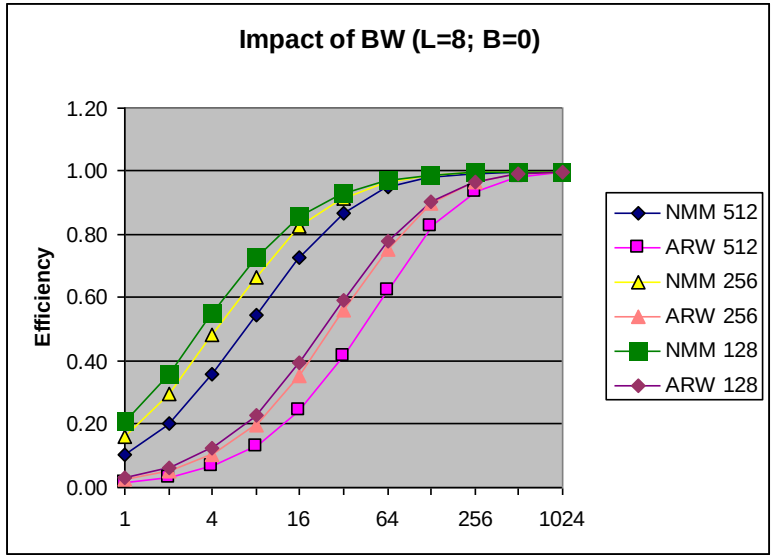


Dimemas

Dimemas: Coarse grain, Trace driven simulation

- Simulation: Highly non-linear model
  - MPI protocols, resource contention…
- Parametric sweeps
  - On abstract architectures
  - On application computational regions
- What-if analysis
  - Ideal machine (instantaneous network)
  - Estimating impact of ports to MPI+OpenMP/CUDA/…
  - Should I use asynchronous communications?
  - Are all parts equally sensitive to network?
- MPI sanity check
  - Modeling nominal
- Paraver – Dimemas tandem
  - Analysis and prediction
  - What-if from selected time window





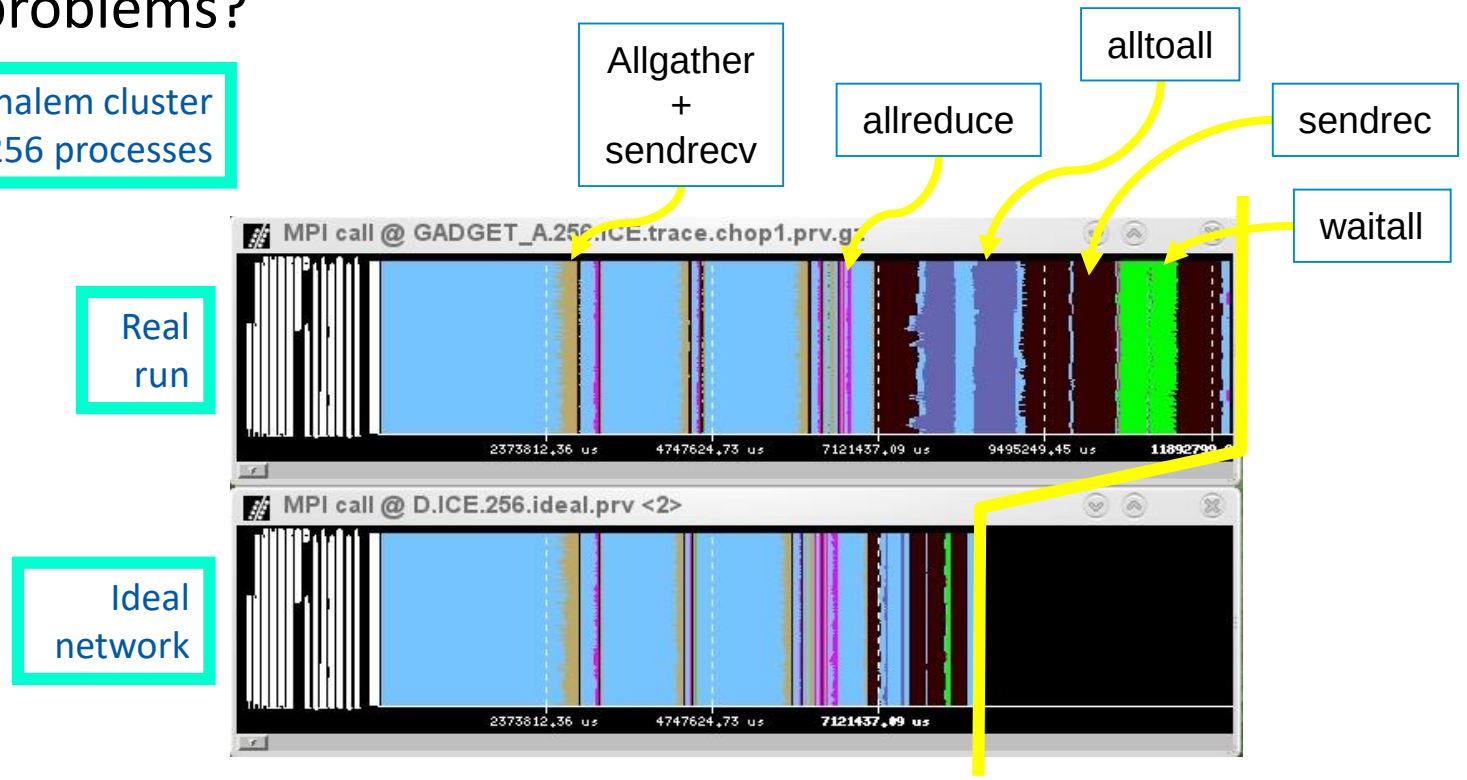
Detailed feedback on simulation (trace)

Ideal machine

The impossible machine: BW = ∞, L = 0

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

GADGET @ Nehalem cluster
256 processes



Performance Optimisation and Productivity

A Centre of Excellence in Computing Applications

Contact:

https://www.pop-coe.eu

mailto:pop@bsc.es

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 676553.