Understanding applications with Paraver and Dimemas

tools@bsc.es

Feb 11th 2014

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

German Research School for Simulation Sciences

RWTH Rheinisch-Westfälische Technische Hochschule Aachen

Technische Universität München

Universität Stuttgart

Université de Versailles Saint-Quentin-en-Yvelines

Jülich Forschungszentrum

Lawrence Livermore National Laboratory

Technische Universität Dresden

University of Oregon

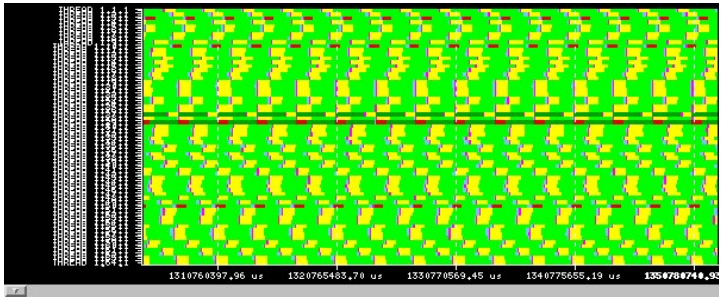
The University of Tennessee

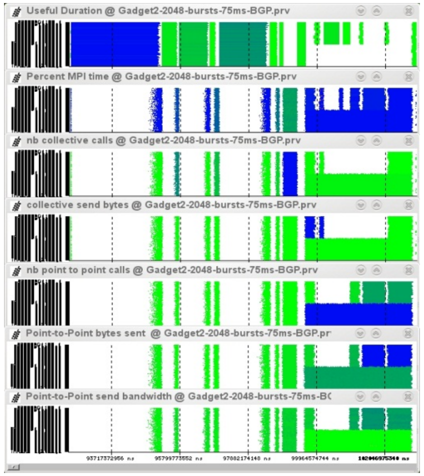
Tuesday Morning Agenda

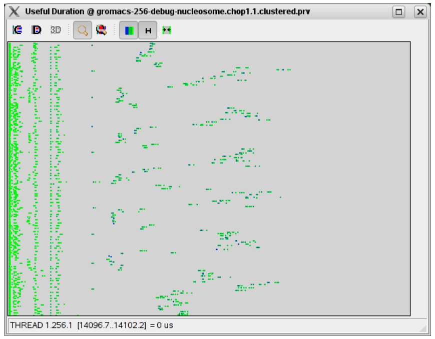
- Tools presentation
- Demo: CG-POP analysis
- Hands-on
  - Core tools: Extrae, Paraver, Dimemas
  - Analytics modules: Clustering, Tracking, Folding

Our Tools

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

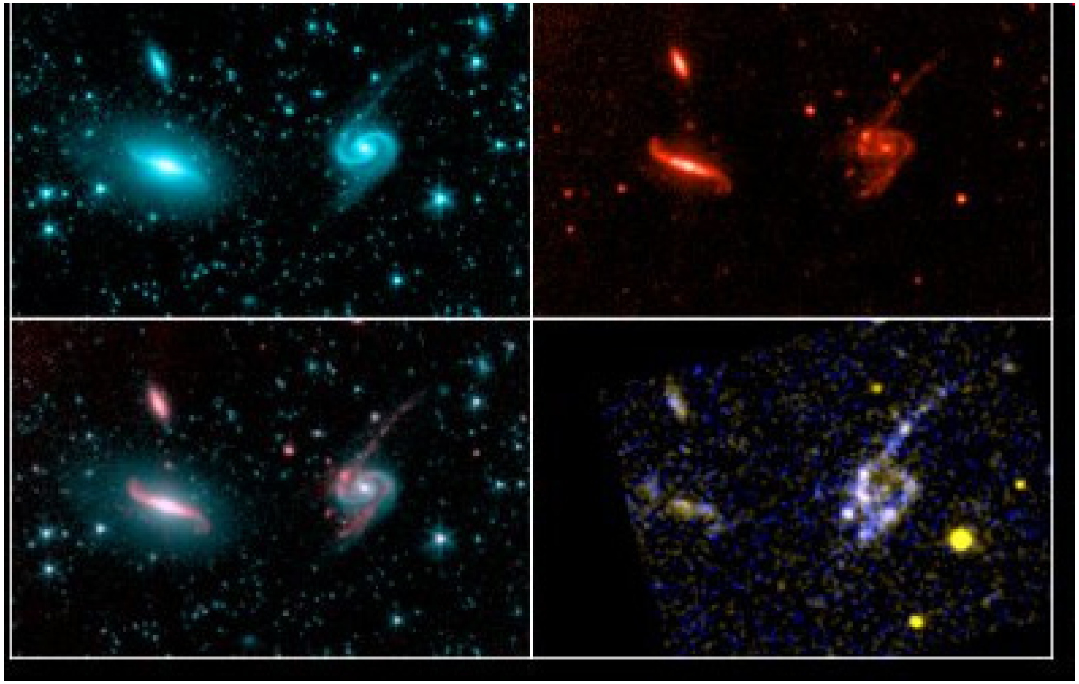



Useful Duration @ Gadget2-2048-bursts-75ms-BGP.prv
Percent MPI time @ Gadget2-2048-bursts-75ms-BGP.prv
nb collective calls @ Gadget2-2048-bursts-75ms-BGP.prv
collective send bytes @ Gadget2-2048-bursts-75ms-BGP.prv
nb point to point calls @ Gadget2-2048-bursts-75ms-BGP.prv
Point-to-Point bytes sent @ Gadget2-2048-bursts-75ms-BGP.pr
Point-to-Point send bandwidth @ Gadget2-2048-bursts-75ms-BC


Useful Duration @ gromacs-256-debug-nucleosome.chop1.1.clustered.prv
3D
THREAD 1.256.1 [14096.7..14102.2] = 0 us

Multispectral imaging

- Different looks at one reality
  - Different spectral bands (light sources and filters)
- Highlight different aspects
  - Can combine into false colored but highly informative images



Spreadsheets and browsers

- Display, manipulate data
  - Dynamic content
  - User defined operations

BSC-CNS - Mozilla Firefox

File Edit View History Bookmarks Tools Help
BSC-CNS
www.bsc.es
Google
CONTACT | QUICK LINKS | SITE MAP | LOG IN
Search
ABOUT BSC
COMPUTER SCIENCES
EARTH SCIENCES
LIFE SCIENCES
COMPUTER APPLICATIONS
MARENOSTRUM SUPPORT & SERVICES
EXCELENCIA

Centro Nacional de Supercomputación

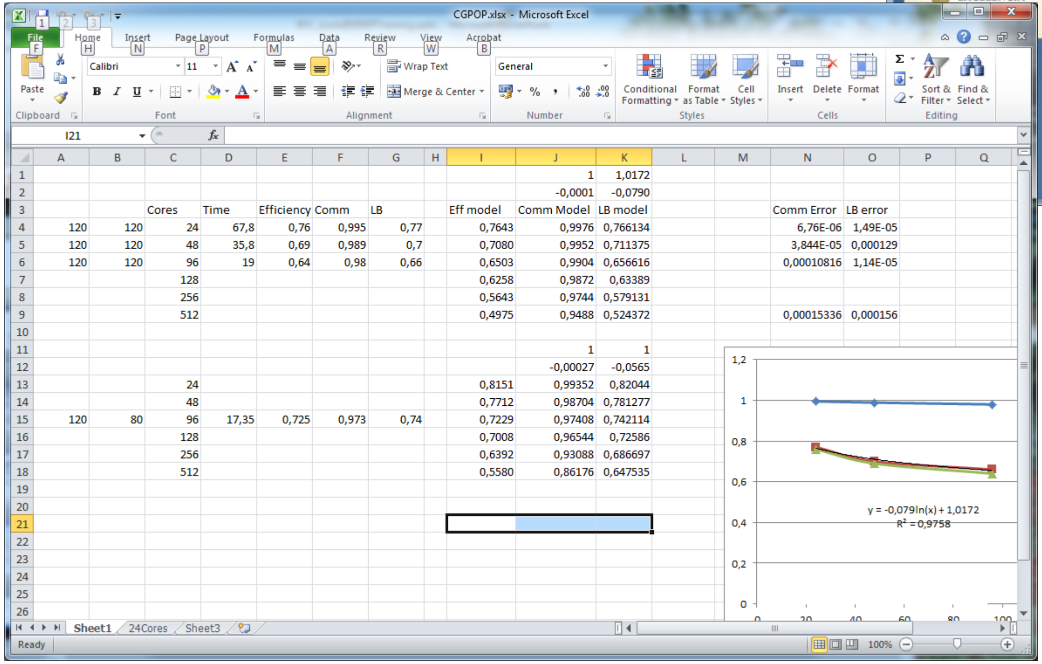
Barcelona Supercomputing Center

9 Nov 2012 - BSC
La nueva versión de MareNostrum multiplicará por 10 su capacidad de cálculo
La nueva máquina estará completamente instalada a principios de 2013. Una primera fase, el 70%, se ha situado en el 36 del mundo y el 12 de Europa según el ranking Top500.

20 Nov 2012 - BSC
HPC Services for Industry
This demo reel shows the simulation results of some of the research projects done at the Barcelona Supercomputing Center during 2012. The video is prepared by the BSC's Visualization team.

14 Nov 2012 - BSC
Premios internacionales para el proyecto Alya Red y su vídeo divulgativo
Los dos reconocimientos han tenido lugar durante la conferencia internacional SC12, la más importante en supercomputación, que se está celebrando esta semana en Salt Lake City

23 Nov 2012 - BSC
Una competición internacional avala la potencia de la bioinformática para el diagnóstico de enfermedades
El equipo liderado por David Rossell y Patrick Aloy del IRB Barcelona y la empresa Anaxomics Biotech, entre los mejores del mundo en pronosticar y clasificar pacientes en un "test a



CGPOP.xlsx - Microsoft Excel

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1										1	1,0172				
2										-0,0001	-0,0790				
3			Cores	Time	Efficiency	Comm	LB		Eff model	Comm Model	LB model			Comm Error	LB error
4	120	120	24	67,8	0,76	0,995	0,77		0,7643	0,9976	0,766134			6,76E-06	1,49E-05
5	120	120	48	35,8	0,69	0,989	0,7		0,7080	0,9952	0,711375			3,844E-05	0,000129
6	120	120	96	19	0,64	0,98	0,66		0,6503	0,9904	0,656616			0,00010816	1,14E-05
7			128						0,6258	0,9872	0,63389				
8			256						0,5643	0,9744	0,579131				
9			512						0,4975	0,9488	0,524372			0,00015336	0,000156
10															
11										1	1				
12										-0,00027	-0,0565				
13			24						0,8151	0,99352	0,82044				
14			48						0,7712	0,98704	0,781277				
15	120	80	96	17,35	0,725	0,973	0,74		0,7229	0,97408	0,742114				
16			128						0,7008	0,96544	0,72586				
17			256						0,6392	0,93088	0,686697				
18			512						0,5580	0,86176	0,647535				
19															
20															
21															
22															
23															
24															
25															
26															

$y = -0{,}079\ln(x) + 1{,}0172$
$R^2 = 0{,}9758$

A “different” view on performance analysis and tools

- **Behavioral structure** vs. syntactic structure
  - Algorithmic and performance
  - In space and time

- **Variability**
  - Multimodal distributions
  - Variability + synchronization → critical non linear effects

- **Flexibility** to let analyst navigate the captured data and gain as much **insight** as possible from as **few application runs** as possible.

Paraver

What is Paraver

- A browser …

…to manipulate (visualize, filter, cut, combine, …) ….

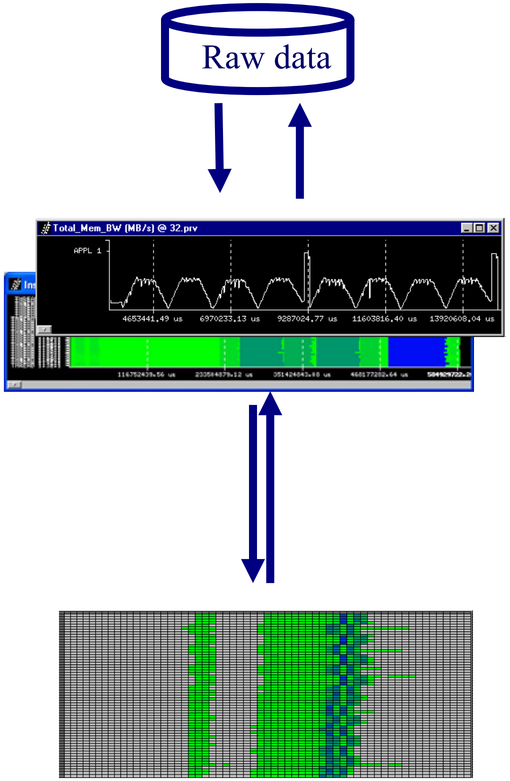
… sequences of time-stamped events …

… with a multispectral philosophy …

… and a mathematical foundation …

… that happens to be mainly used for **performance analysis**

Paraver – Performance data browser



Raw data

Timelines

2/3D tables (Statistics)

Trace visualization/analysis

+ trace manipulation

Goal = Flexibility

No semantics

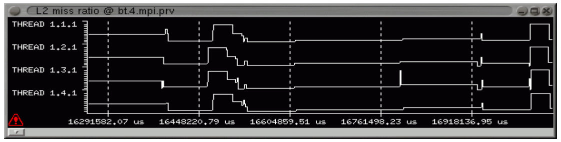
Programmable

Comparative analyses

Multiple traces

Synchronize scales

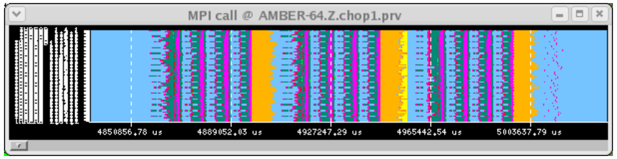
Timelines



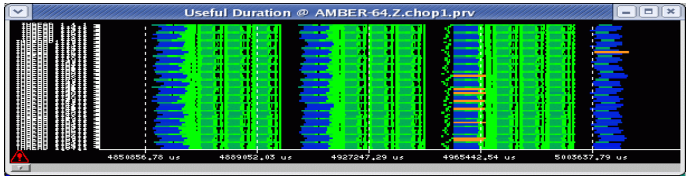
- Each window displays one view
  - **Piecewise constant** function of time

$$s(t) = S_i, i \in [t_i, t_{i+1})$$

- Types of functions
  - Categorical $S_i \in [0,\mathrm{n}] \subset N, \quad n <$
    - State, user function, outlined routine



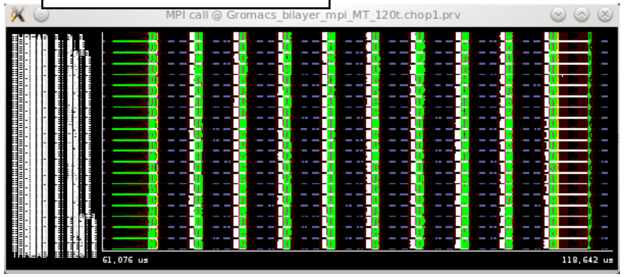
  - Logical $S_i \in \{0,1\}$
    - In specific user function, In MPI call, In long MPI call
  - Numerical $S_i \in R$
    - IPC, L2 miss ratio, Duration of MPI call, duration of computation burst



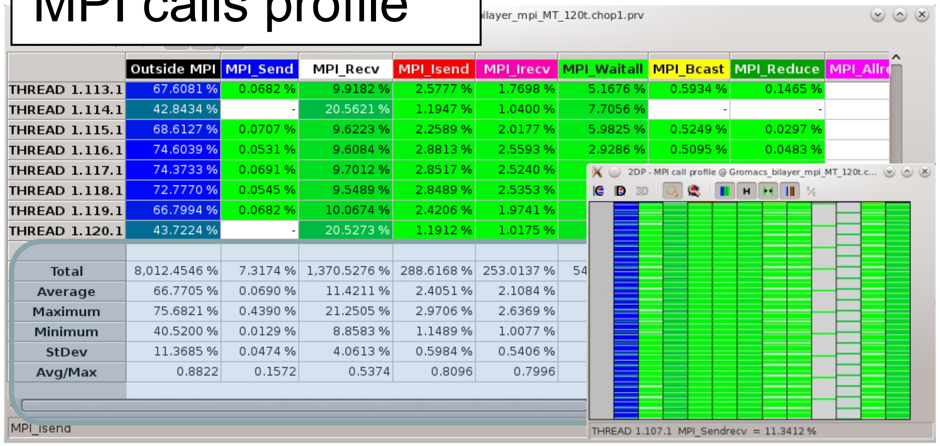
Tables: Profiles, histograms, correlations

- From timelines to tables

MPI calls



MPI calls profile

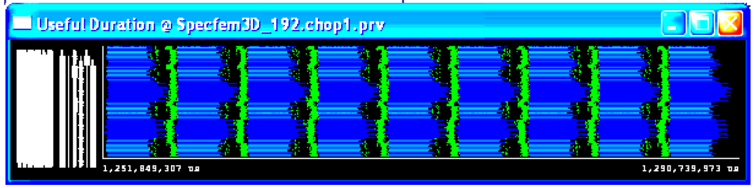

…ilayer_mpi_MT_120t.chop1.prv
2DP - MPI call profile @ Gromacs_bilayer_mpi_MT_120t.c…

	Outside MPI	MPI_Send	MPI_Recv	MPI_Isend	MPI_Irecv	MPI_Waitall	MPI_Bcast	MPI_Reduce	MPI_Allre
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				

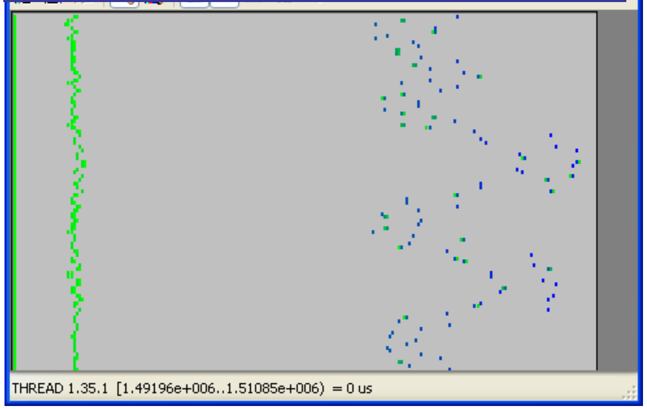
MPI_Isend

THREAD 1.107.1 MPI_Sendrecv = 11.3412 %

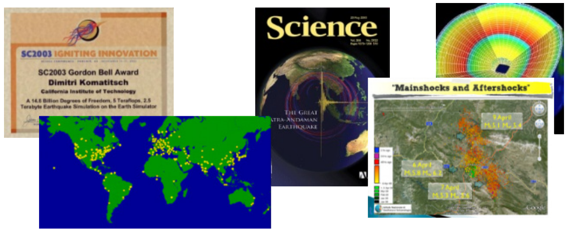
Useful Duration

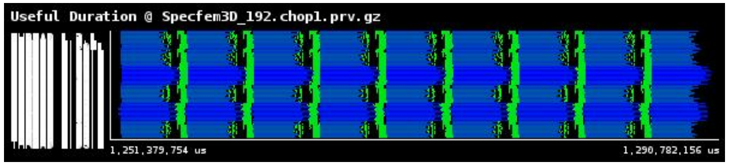


Histogram Useful Duration

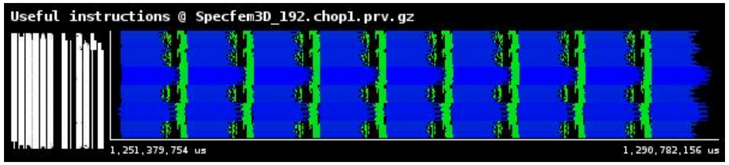

THREAD 1.35.1 [1.49196e+006..1.51085e+006) = 0 us

Analyzing variability through histograms and timelines

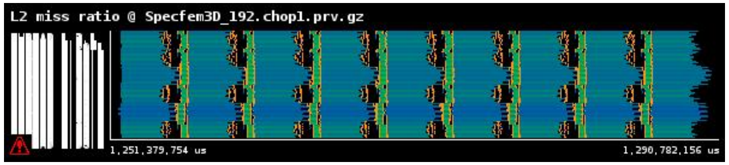

SC2003 IGNITING INNOVATION
SC2003 Gordon Bell Award
Dimitri Komatitsch
California Institute of Technology
Science
"Mainshocks and Aftershocks"


Useful Duration @ Specfem3D_192.chop1.prv.gz

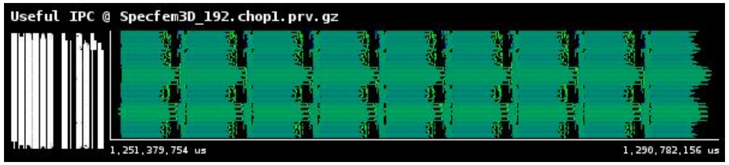
1,251,379,754 us 1,290,782,156 us


Useful instructions @ Specfem3D_192.chop1.prv.gz

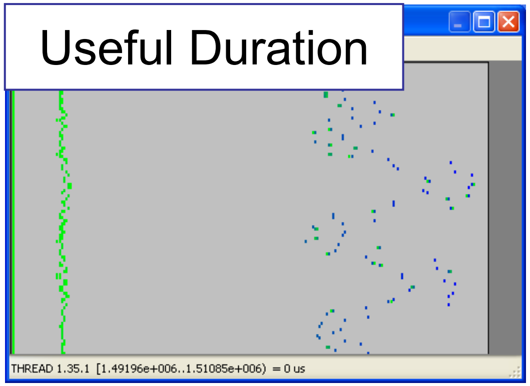
1,251,379,754 us 1,290,782,156 us


L2 miss ratio @ Specfem3D_192.chop1.prv.gz

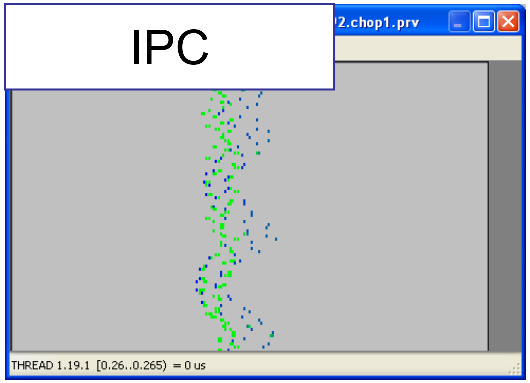
1,251,379,754 us 1,290,782,156 us


Useful IPC @ Specfem3D_192.chop1.prv.gz

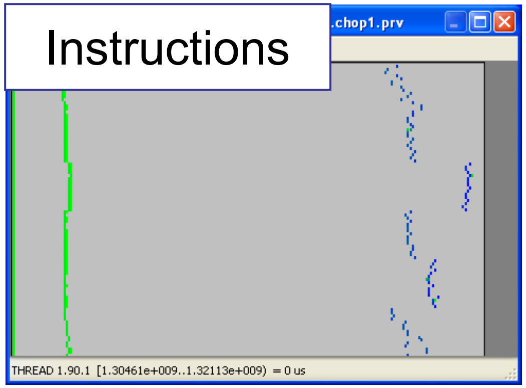
1,251,379,754 us 1,290,782,156 us


Useful Duration

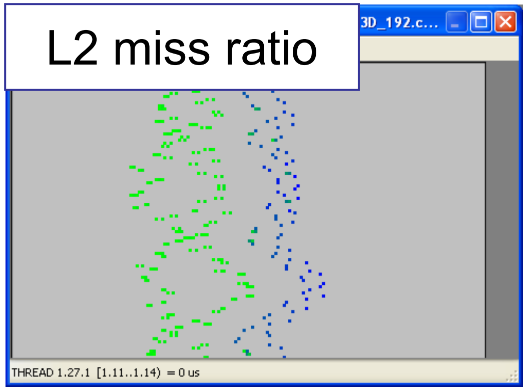
THREAD 1.35.1 [1.49196e+006..1.51085e+006) = 0 us


IPC

THREAD 1.19.1 [0.26..0.265) = 0 us


Instructions

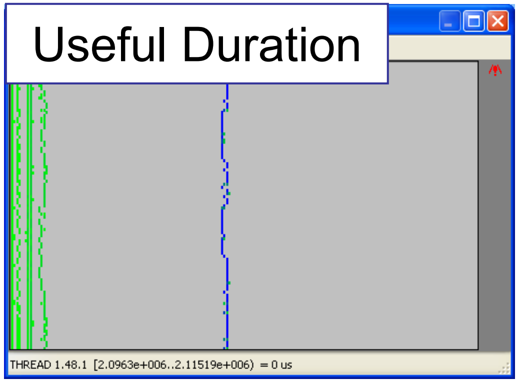
THREAD 1.90.1 [1.30461e+009..1.32113e+009) = 0 us


L2 miss ratio

THREAD 1.27.1 [1.11..1.14) = 0 us

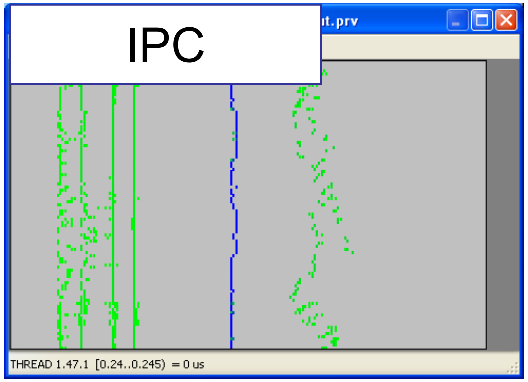
Analyzing variability through histograms and timelines

- By the way: six months later ….



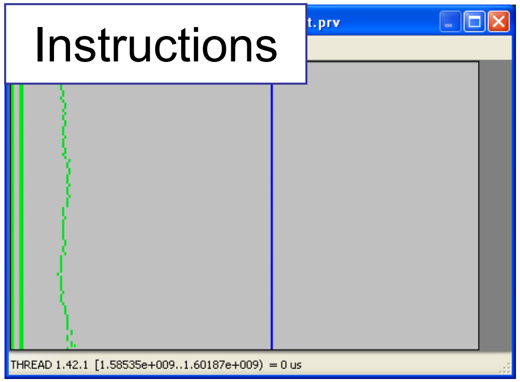
Useful Duration

```
THREAD 1.48.1  [2.0963e+006..2.11519e+006)  = 0 us
```



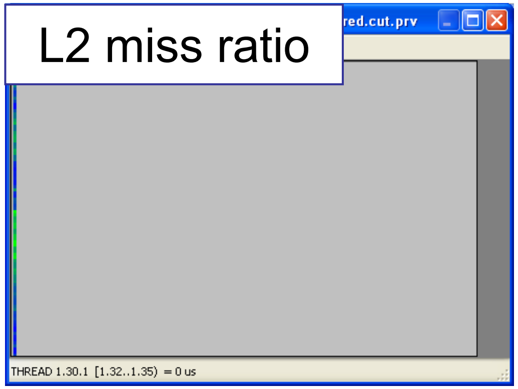
IPC

```
THREAD 1.47.1  [0.24..0.245)  = 0 us
```



Instructions

```
THREAD 1.42.1  [1.58535e+009..1.60187e+009)  = 0 us
```

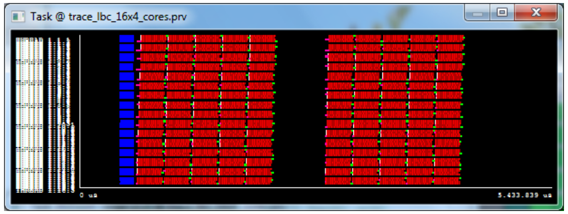


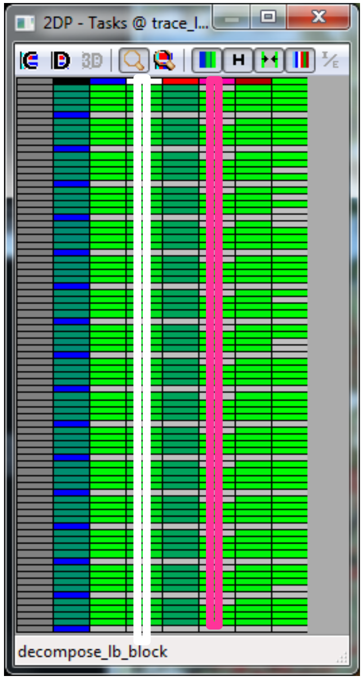
L2 miss ratio

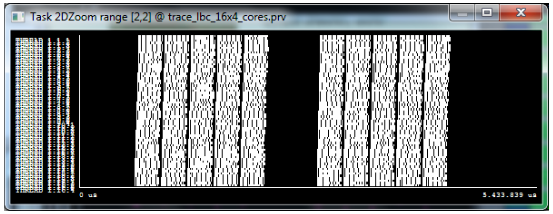
```
THREAD 1.30.1  [1.32..1.35)  = 0 us
```

From tables to timelines

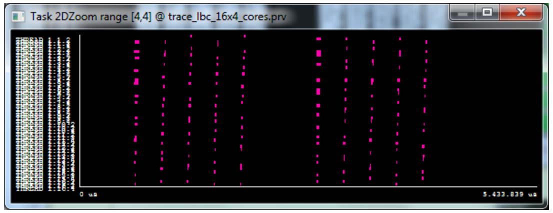
- Where in the timeline do the values in certain table columns appear?
  - ie. want to see the time distribution of a given routine?







Only showing when a given value happens

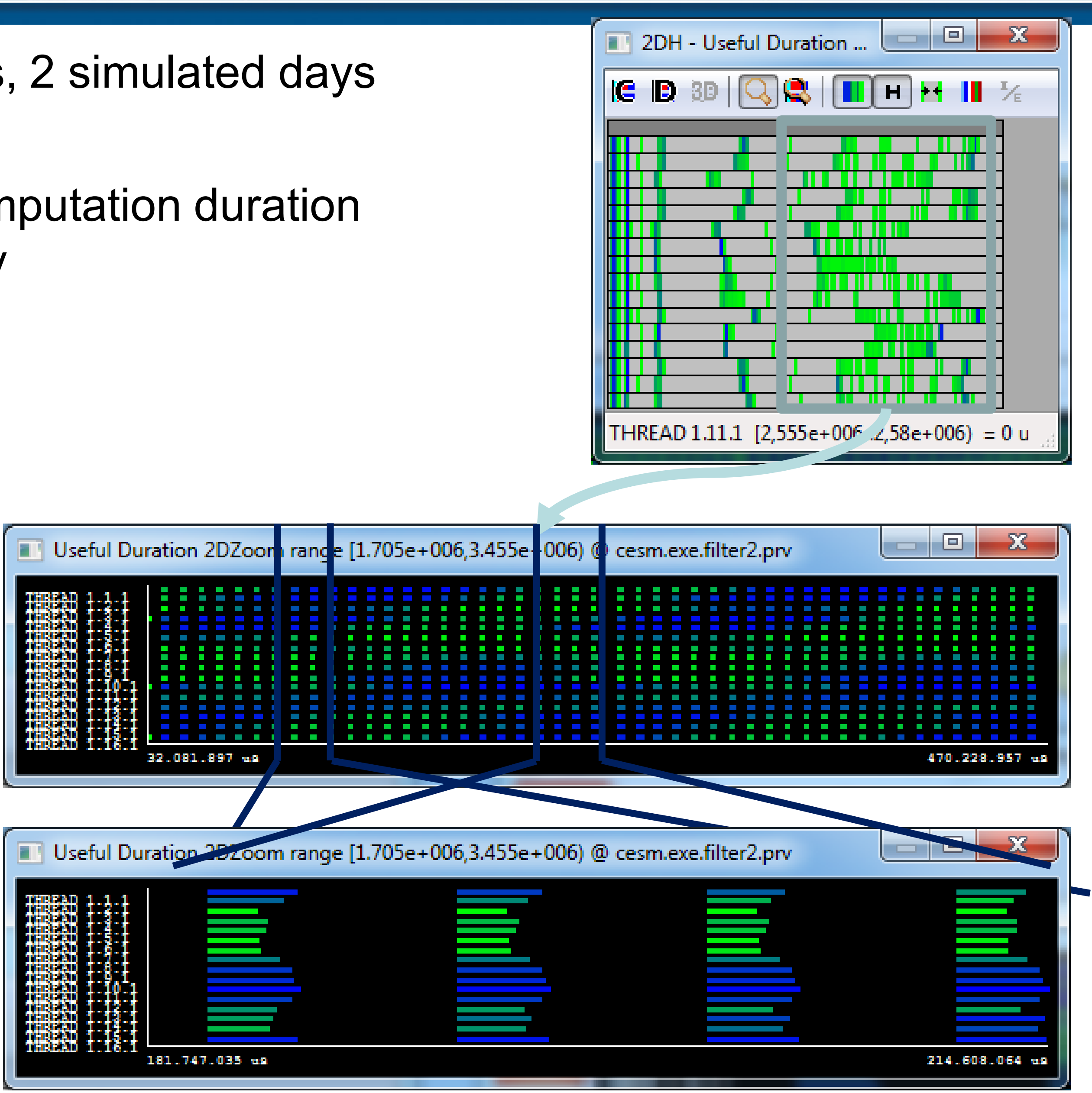


Variability … is everywhere

- CESM: 16 processes, 2 simulated days

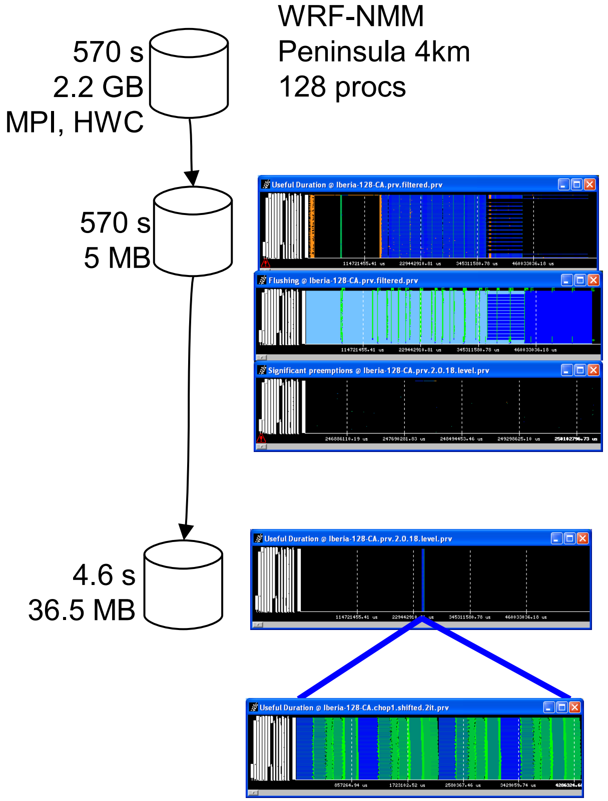
- Histogram useful computation duration shows high variability
- How is it distributed?

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


2DH - Useful Duration ...
THREAD 1.11.1 [2,555e+006,2,58e+006) = 0 u
Useful Duration 2DZoom range [1.705e+006,3.455e+006) @ cesm.exe.filter2.prv
THREAD 1.1.1 … THREAD 1.16.1
32.081.897 us
470.228.957 us
Useful Duration 2DZoom range [1.705e+006,3.455e+006) @ cesm.exe.filter2.prv
THREAD 1.1.1 … THREAD 1.16.1
181.747.035 us
214.608.064 us

Trace manipulation

- Data handling/summarization capability
  - Filtering
    - Subset of records in original trace
    - By duration, type, value,…
    - Filtered trace IS a paraver trace and can be analysed with the same cfgs (as long as needed data 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,…


WRF-NMM
Peninsula 4km
128 procs
570 s
2.2 GB
MPI, HWC
570 s
5 MB
4.6 s
36.5 MB
Useful Duration @ Iberia-128-CA.prv.filtered.prv
Flushing @ Iberia-128-CA.prv.filtered.prv
Significant preemptions @ Iberia-128-CA.prv.2.0.18.level.prv
Useful Duration @ Iberia-128-CA.prv.2.0.18.level.prv
Useful Duration @ Iberia-128-CA.chop1.shifted.2it.prv

See slides at end of presentation for details

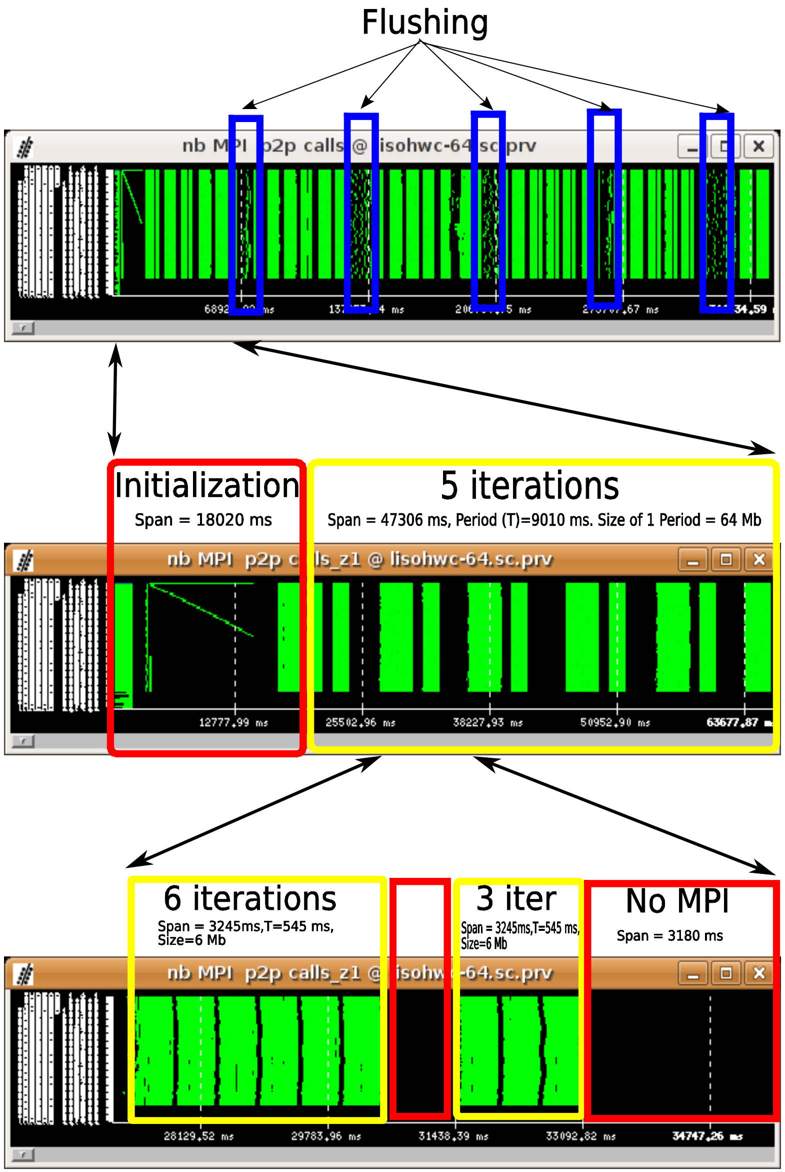
Performance Analytics

Spectral analysis

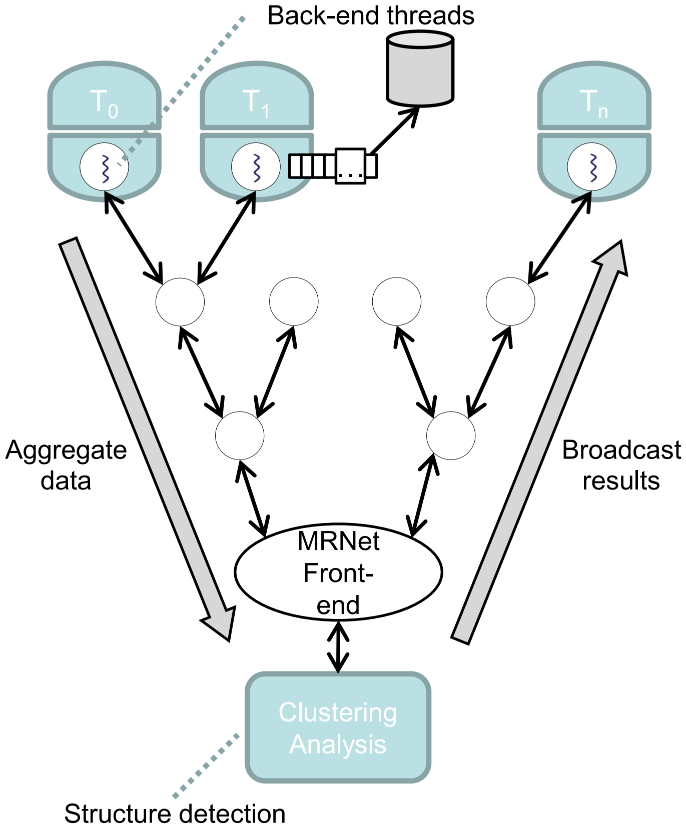
Signal processing applied to performance analysis

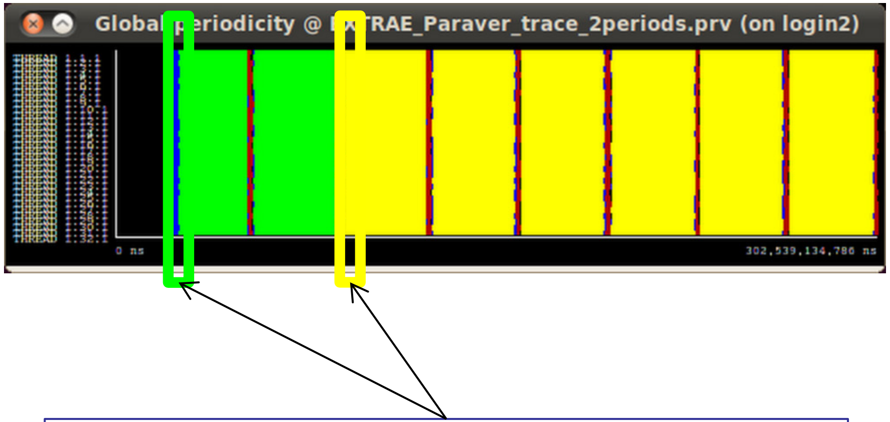
- Techniques
  - Mathematical morphology
    - clean up perturbed regions
  - Wavelet transform
    - identify coarse regions
  - Spectral analysis
    - detailed periodic pattern

- Useful
  - Identify structure (periodicity)
  - Reduce trace sizes
  - Increase precision of profiles (report non perturbed stats)


Flushing
nb MPI p2p calls @ lisohwc-64.sc.prv
Initialization
Span = 18020 ms
5 iterations
Span = 47306 ms, Period (T)=9010 ms. Size of 1 Period = 64 Mb
nb MPI p2p calls_z1 @ lisohwc-64.sc.prv
12777,99 ms
25502,96 ms
38227,93 ms
50952,90 ms
63677,87 ms
6 iterations
Span = 3245ms,T=545 ms, Size=6 Mb
3 iter
Span = 3245ms,T=545 ms, Size=6 Mb
No MPI
Span = 3180 ms
nb MPI p2p calls_z1 @ lisohwc-64.sc.prv
28129,52 ms
29783,96 ms
31438,39 ms
33092,82 ms
34747,26 ms

Scalability: online automatic interval selection




Global periodicity @ E TRAE_Paraver_trace_2periods.prv (on login2)

0 ns

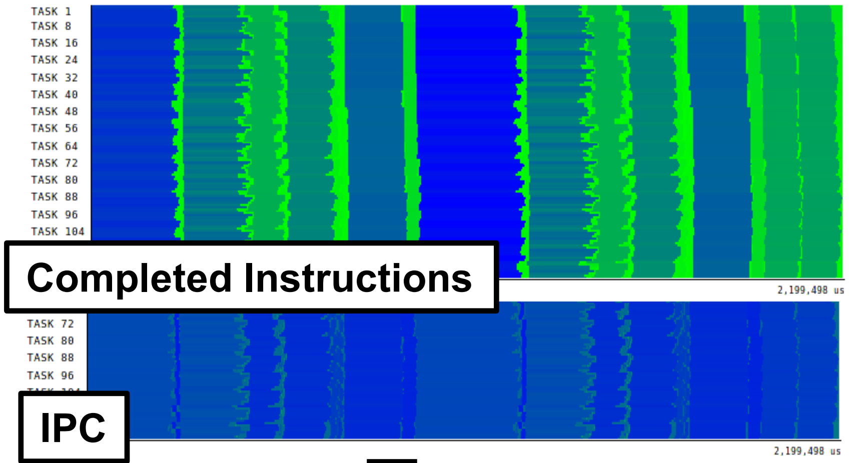
302,539,134,786 ns

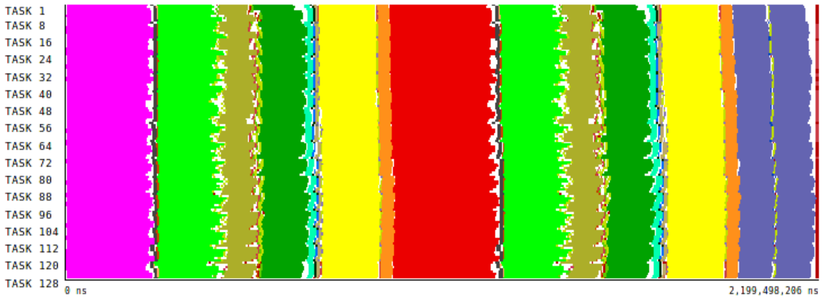
Detailed trace for only small interval

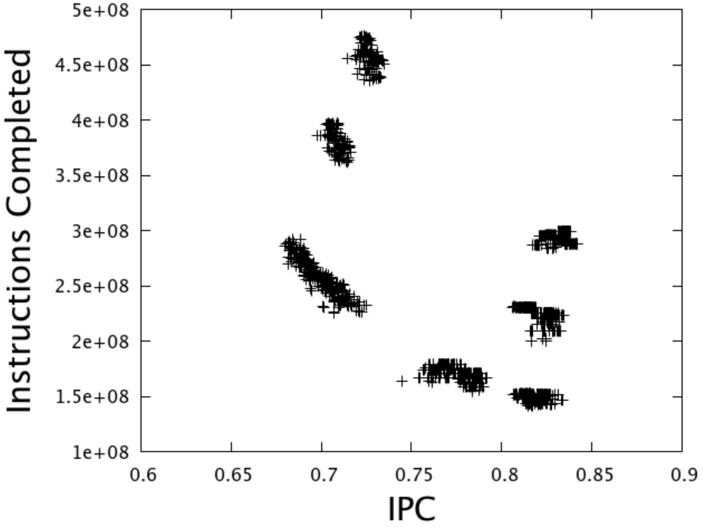
"Scalable tracing with dynamic levels of detail" ICPADS 2011

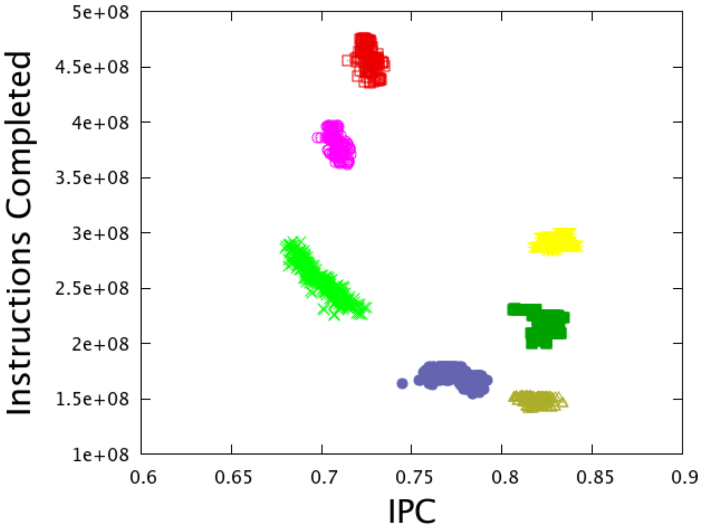
Clustering

Using Clustering to identify structure



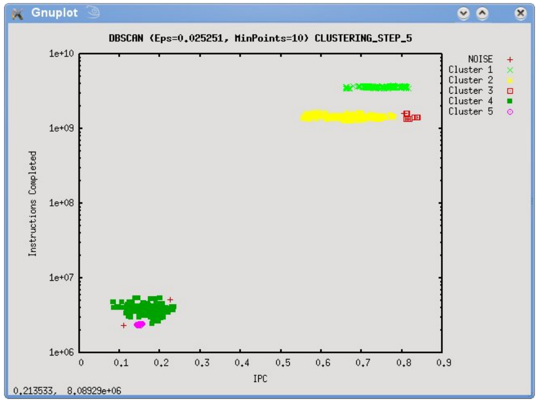




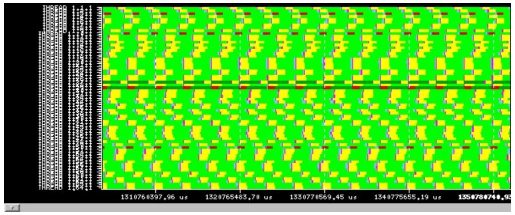


Automatic Detection of Parallel Applications Computation Phases. (IPDPS 2009)

Performance @ serial computation bursts

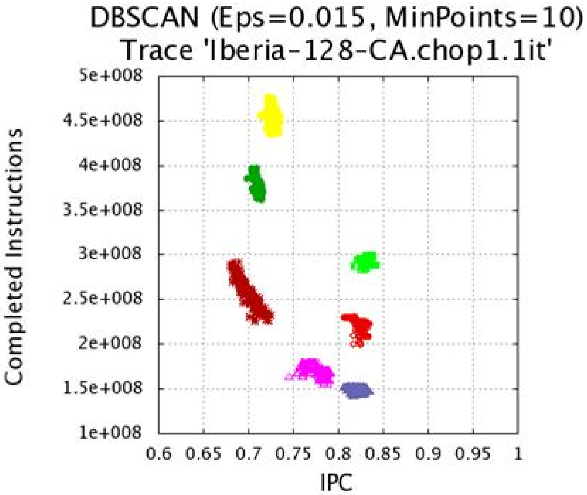


SPECFEM3D

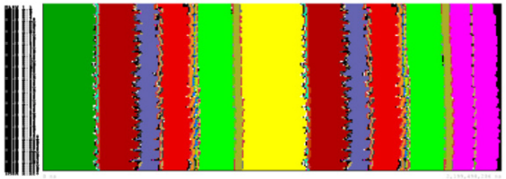


Asynchronous SPMD

Balanced #instr variability in IPC



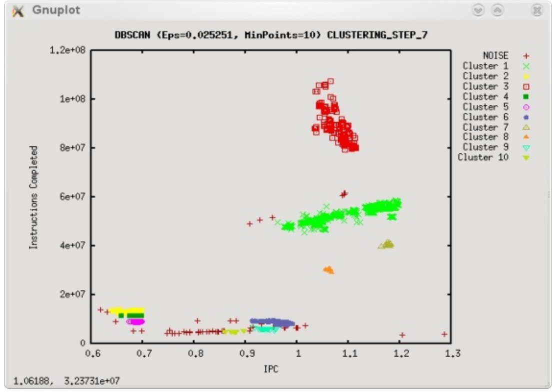
WRF 128 cores



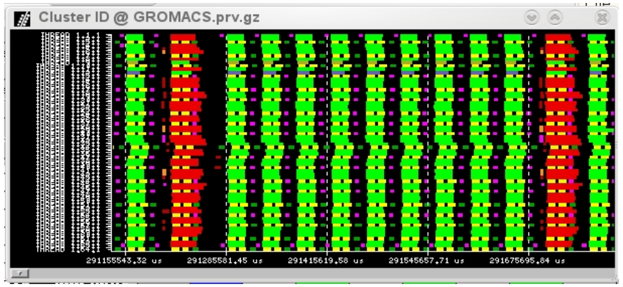
SPMD

Repeated substructure

Coupled imbalance



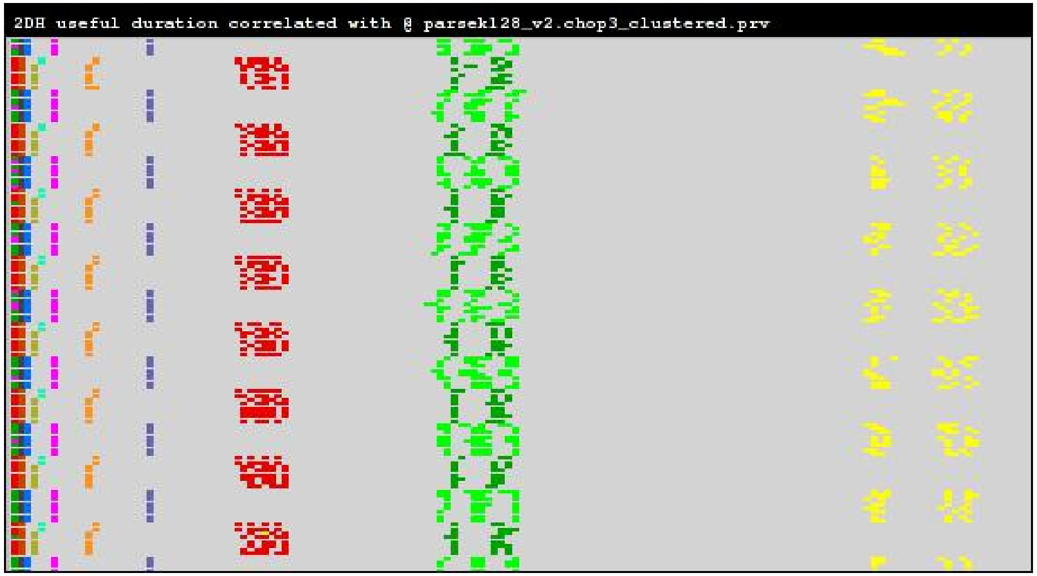
GROMACS



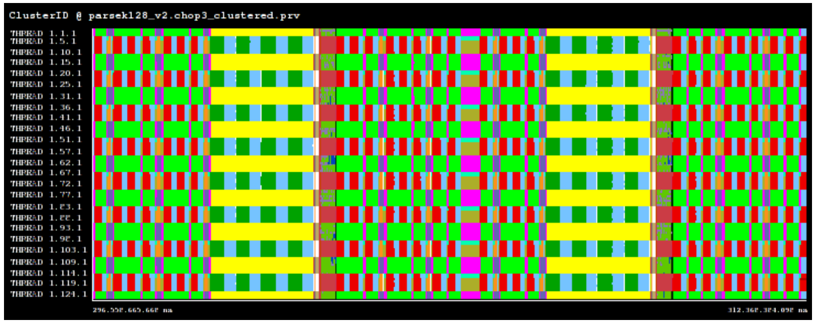
MPMD structure

Different coupled imbalance trends

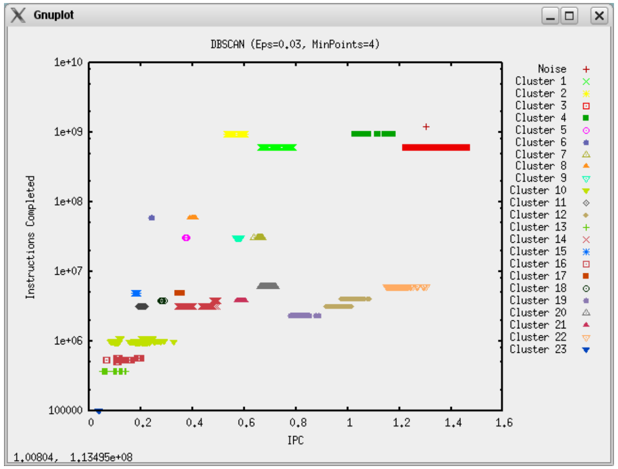
Using clusters with Paraver (PARSEK)

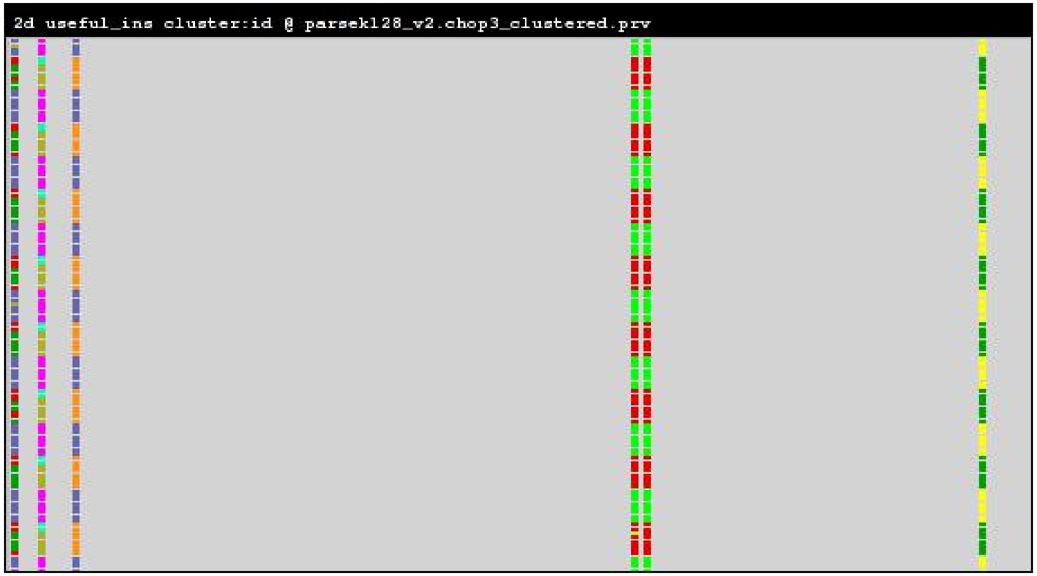

2DH useful duration correlated with @ parsek128_v2.chop3_clustered.prv

duration vs. cluster



instr. vs. cluster

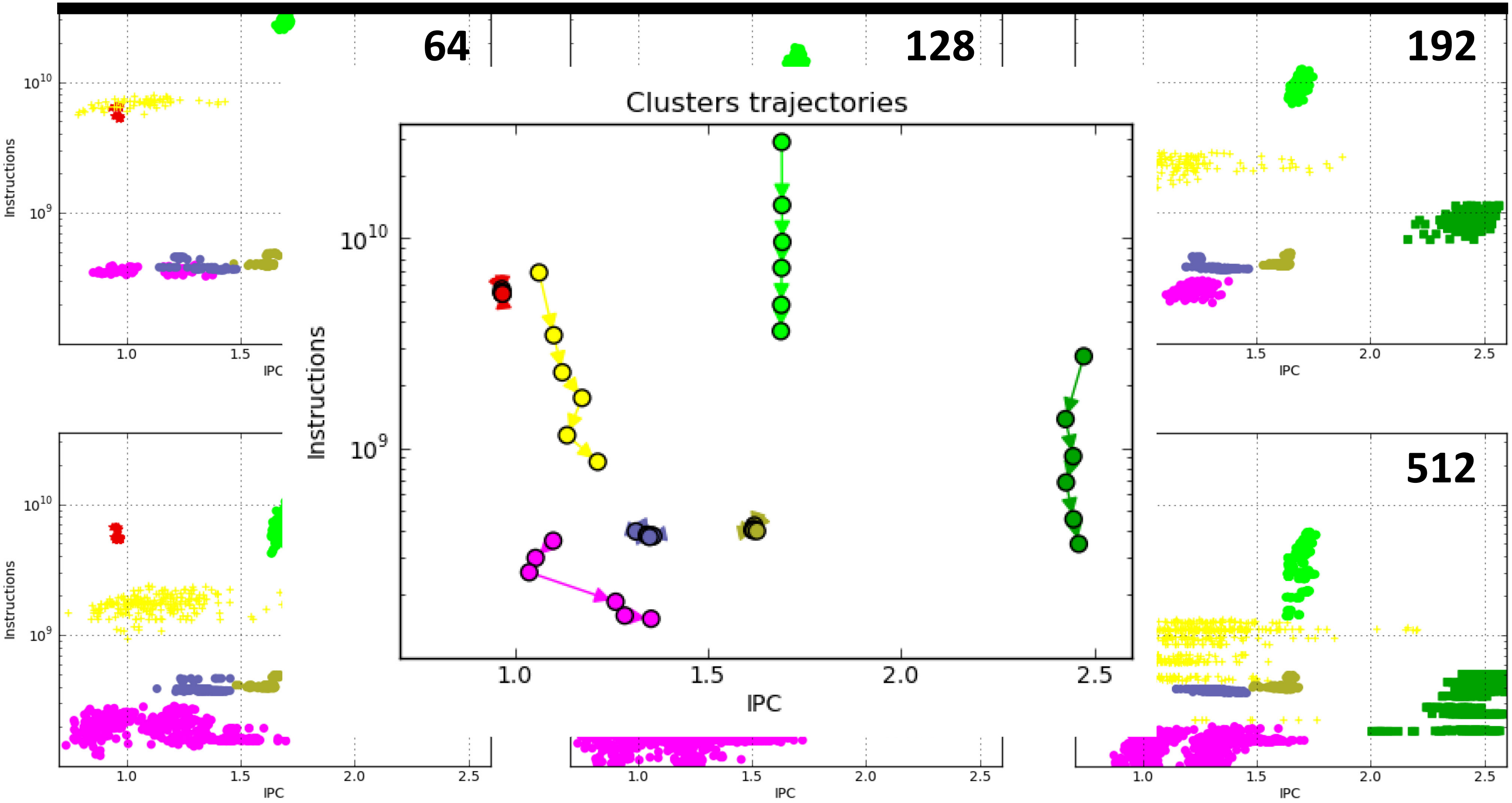




Tracking

Tracking scability through clustering

- OpenMX (strong scale from 64 to 512 tasks)

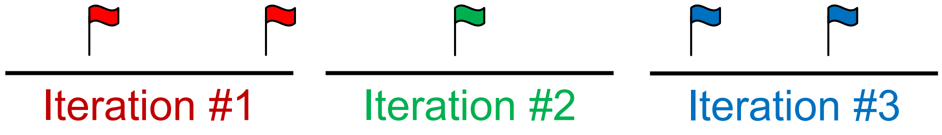


"On the usefulness of object tracking techniques in performance analysis " SC 2013

Folding

Folding: Detailed time evolution

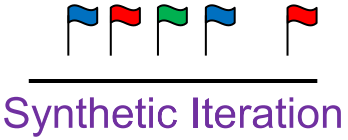
- Benefit from applications' repetitiveness



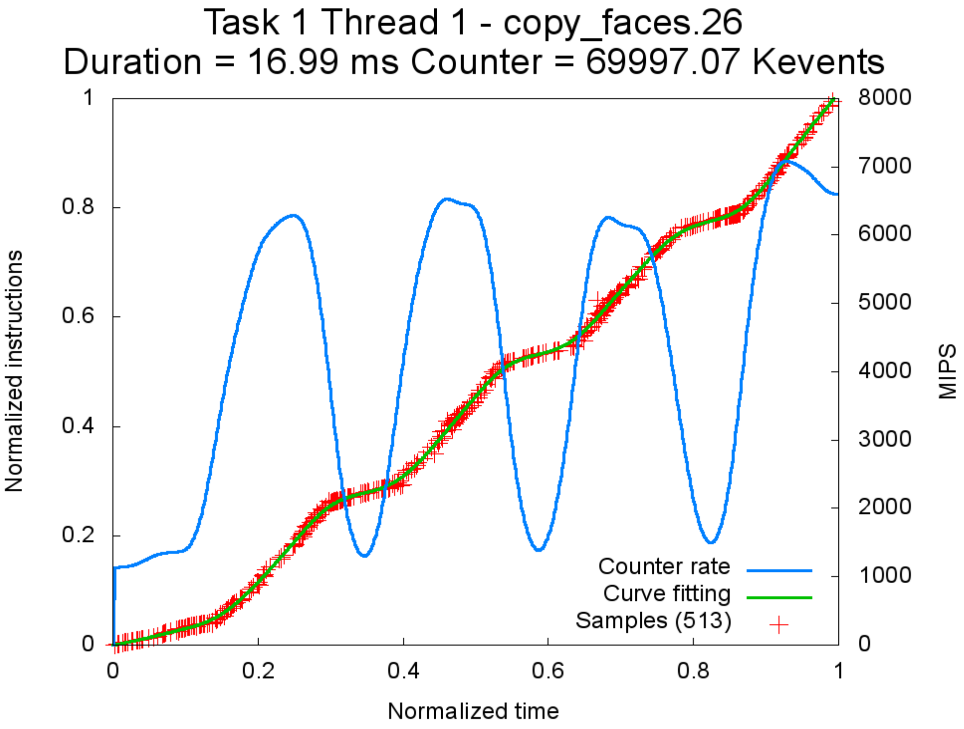
Iteration #1

Iteration #2

Iteration #3



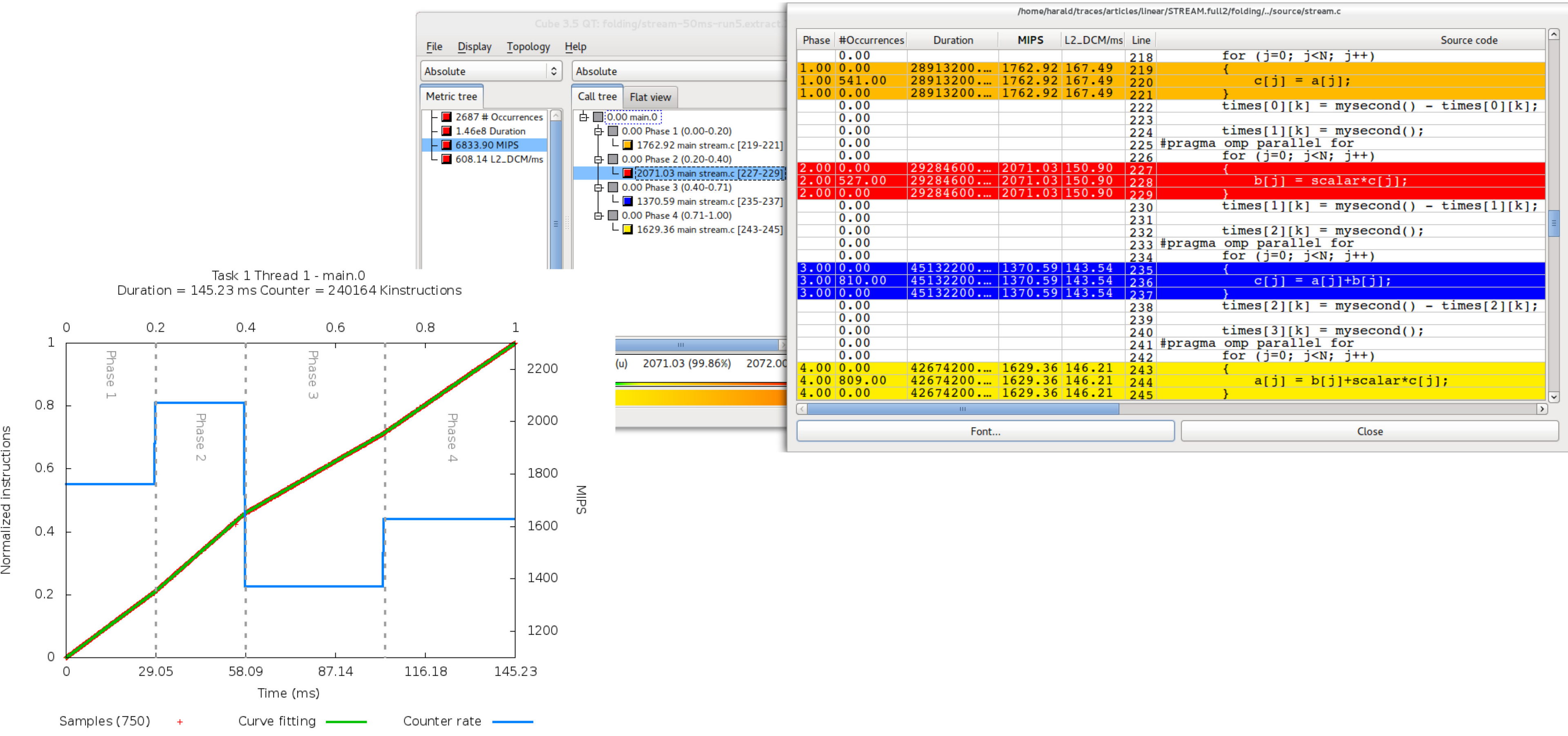
Synthetic Iteration



Unveiling Internal Evolution of Parallel Application Computation Phases (ICPP 2011)

Folding → profiles of rates and ratios

- Call-site sampling information is folded
  - Correlation between hwc and call-sites
  - GVIM/CUBE add-on to show performance within source code
    - Timeless but useful to point performance issues



Cube 3.5 QT: folding/stream-50ms-run5.extract

Metric tree:

```
├ 🟥 2687 # Occurrences
├ 🟥 1.46e8 Duration
├ 🟥 6833.90 MIPS
└ 🟥 608.14 L2_DCM/ms
```

Call tree:

```
⊟ ■ 0.00 main.0
  ⊟ ■ 0.00 Phase 1 (0.00-0.20)
    └ 🟧 1762.92 main stream.c [219-221]
  ⊟ ■ 0.00 Phase 2 (0.20-0.40)
    └ 🟥 2071.03 main stream.c [227-229]
  ⊟ ■ 0.00 Phase 3 (0.40-0.71)
    └ 🟦 1370.59 main stream.c [235-237]
  ⊟ ■ 0.00 Phase 4 (0.71-1.00)
    └ 🟨 1629.36 main stream.c [243-245]
```

/home/harald/traces/articles/linear/STREAM.full2/folding/../source/stream.c

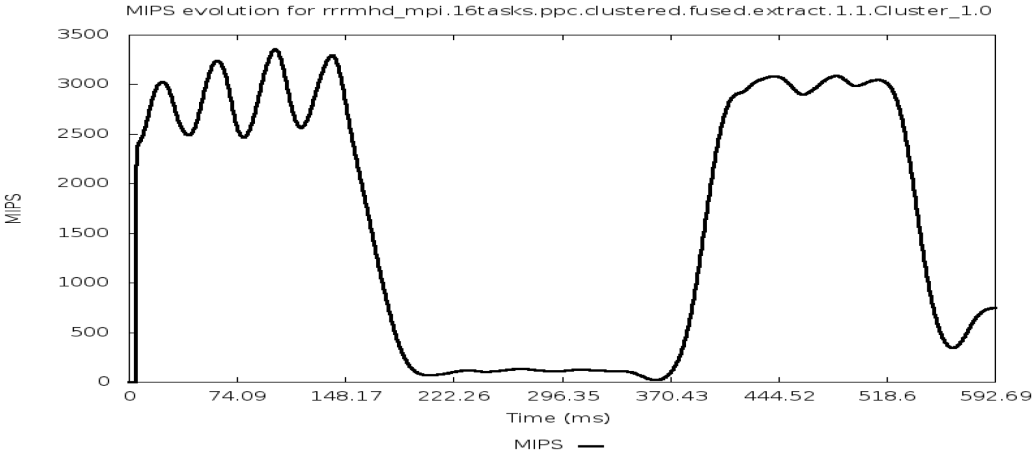
Phase	#Occurrences	Duration	MIPS	L2_DCM/ms	Line	Source code
	0.00				218	for (j=0; j<N; j++)
1.00	0.00	28913200....	1762.92	167.49	219	{
1.00	541.00	28913200....	1762.92	167.49	220	c[j] = a[j];
1.00	0.00	28913200....	1762.92	167.49	221	}
	0.00				222	times[0][k] = mysecond() - times[0][k];
	0.00				223	
	0.00				224	times[1][k] = mysecond();
	0.00				225	#pragma omp parallel for
	0.00				226	for (j=0; j<N; j++)
2.00	0.00	29284600....	2071.03	150.90	227	{
2.00	527.00	29284600....	2071.03	150.90	228	b[j] = scalar*c[j];
2.00	0.00	29284600....	2071.03	150.90	229	}
	0.00				230	times[1][k] = mysecond() - times[1][k];
	0.00				231	
	0.00				232	times[2][k] = mysecond();
	0.00				233	#pragma omp parallel for
	0.00				234	for (j=0; j<N; j++)
3.00	0.00	45132200....	1370.59	143.54	235	{
3.00	810.00	45132200....	1370.59	143.54	236	c[j] = a[j]+b[j];
3.00	0.00	45132200....	1370.59	143.54	237	}
	0.00				238	times[2][k] = mysecond() - times[2][k];
	0.00				239	
	0.00				240	times[3][k] = mysecond();
	0.00				241	#pragma omp parallel for
	0.00				242	for (j=0; j<N; j++)
4.00	0.00	42674200....	1629.36	146.21	243	{
4.00	809.00	42674200....	1629.36	146.21	244	a[j] = b[j]+scalar*c[j];
4.00	0.00	42674200....	1629.36	146.21	245	}

What is a good performance?

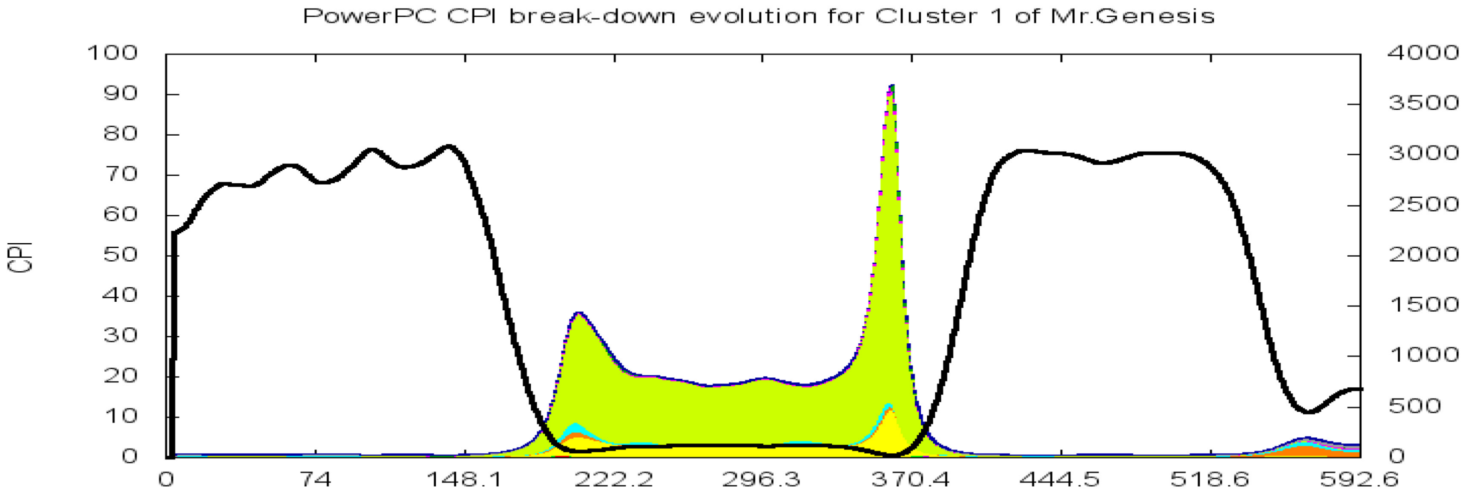
- Performance of a sequential region = 2000 MIPS

Is it good enough?

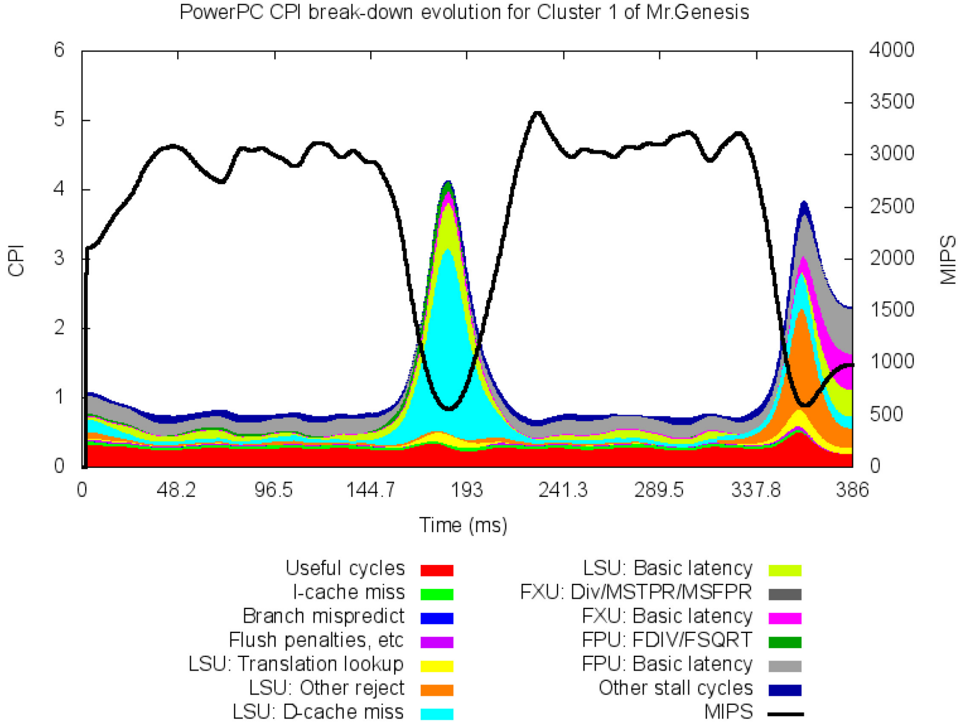
Is it easy to improve?



Instantaneous CPI stack



MRGENESIS

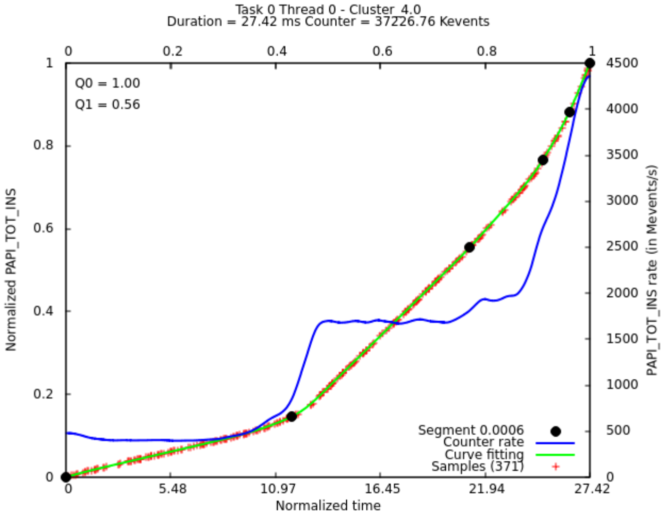


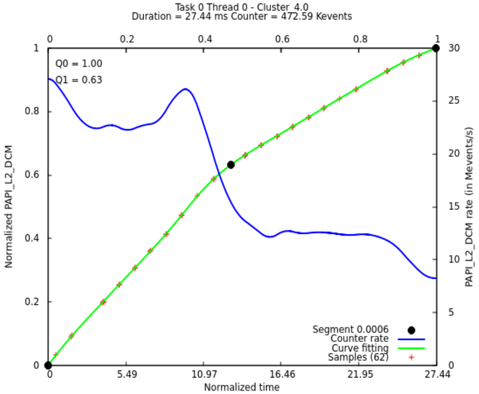
- Trivial fix.(loop interchange)
- Easy to locate?
- Next step?
- Availability of CPI stack models for production processors?
  - Provided by manufacturers?

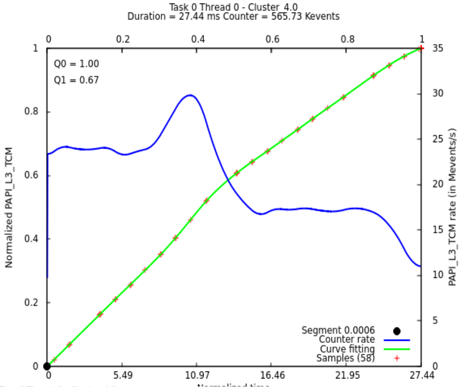
Correlating counters

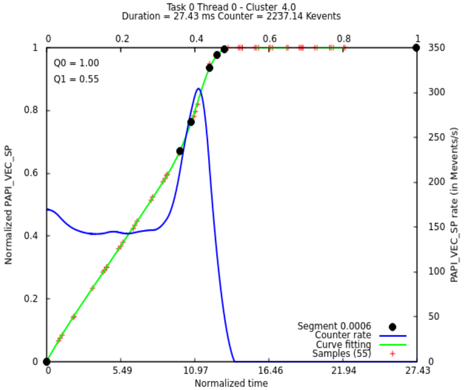
CG-POP

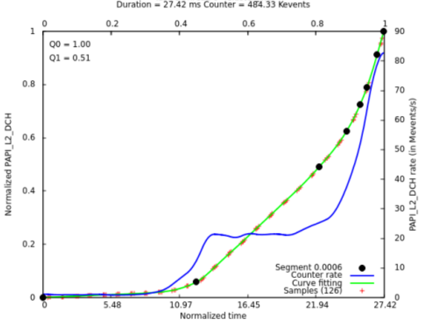
- Within a process
- 3 algorithmic phases
- Impact of multicore sharing







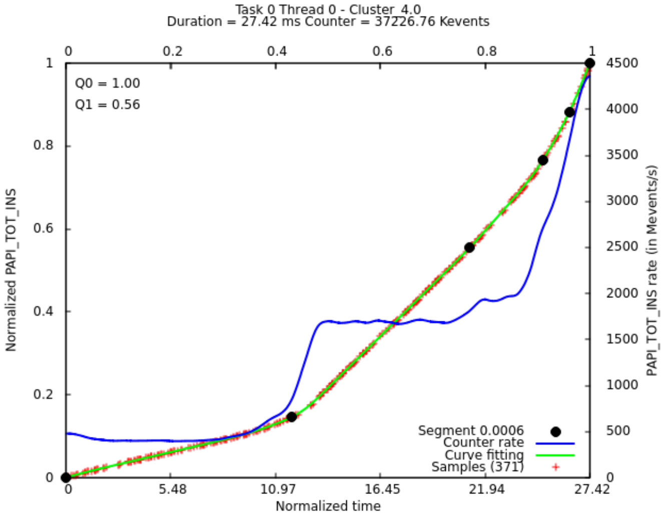


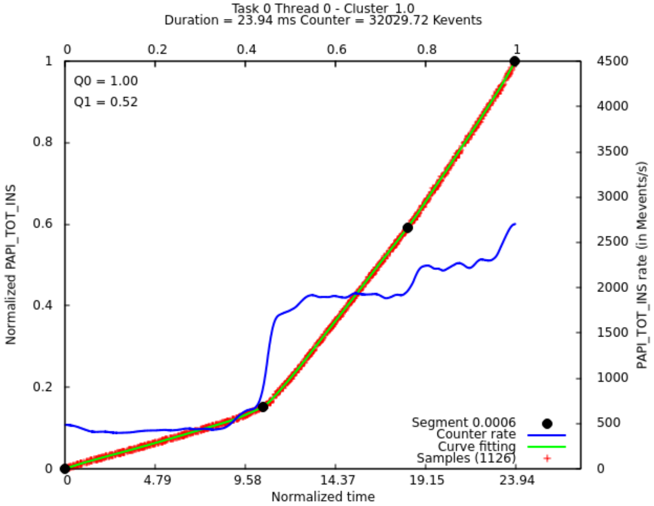


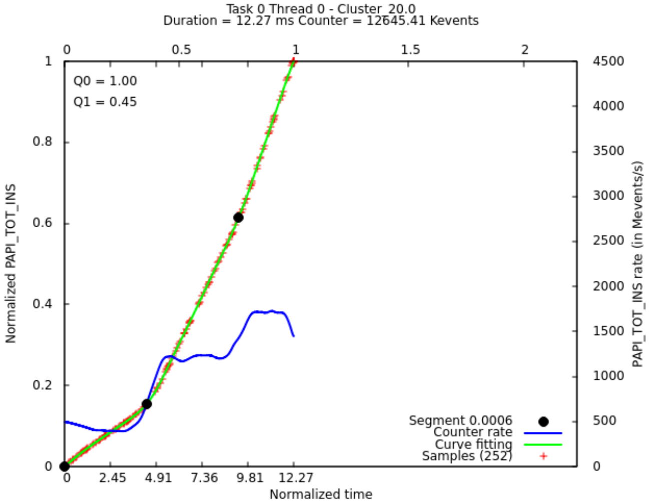
Correlating counters

CG-POP

- Between processes
- 3 Algorithmic phases
- Impact of multicore sharing



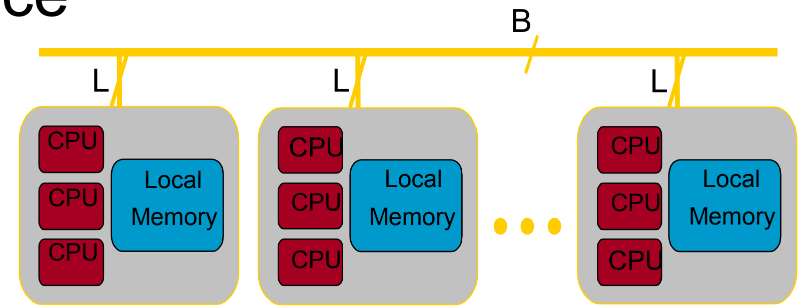




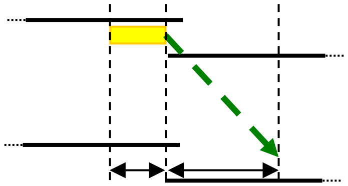
Dimemas

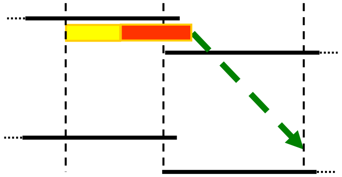
Dimemas: Coarse grain trace driven simulator

- Key factors influencing performance
  - Abstract architecture
  - Basic MPI protocols
  - No attempt to model details
- Objectives
  - Simple / general, Fast simulations
- Linear components
  - Point to point communication
  - Sequential processor performance (global CPU speed, per block/subroutine
- Non-linear components
  - Synchronization semantics (blocking receives, rendezvous)
  - Resources contention (CPU, links half/full duplex, busses)



$$T = \frac{MessageSize}{BW} + L$$



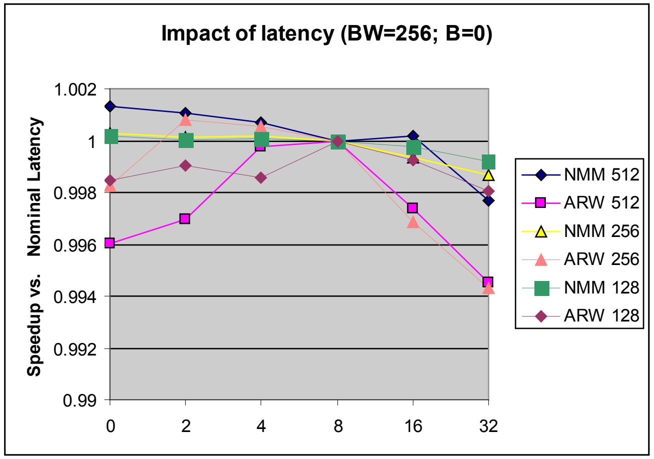


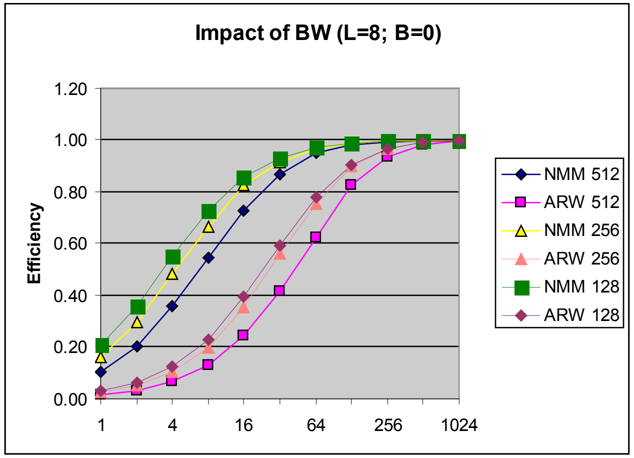
Dimemas vs. Paraver traces

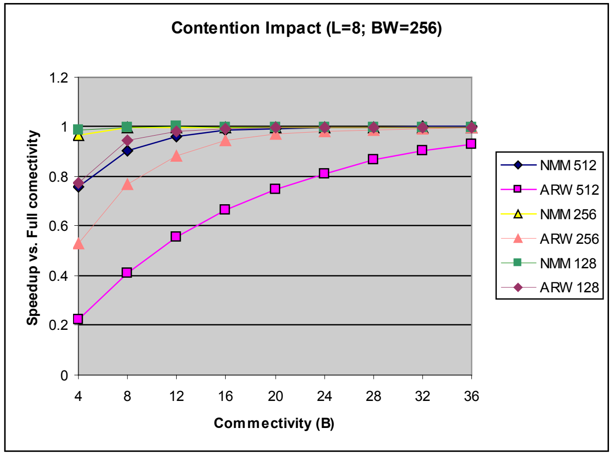
- Paraver trace: what happens when
  - Actual wall clock time of events
- Dimemas trace: sequence of resource demands
  - Duration of computation bursts
  - Type of communication, partners and bytes
- Can be generated from Paraver trace
  - prv2dim input.prv output.dim
- Dimemas generates as output a Paraver file of the simulated run

Parametric studies – network sensitivity

- WRF, Iberia 4Km, 4 procs/node
  - No sensitive to latency
  - NMM
    - BW – 256MB/s
    - 512 – sensitive to contention
  - ARW
    - BW - 1GB/s
    - Sensitive to contention



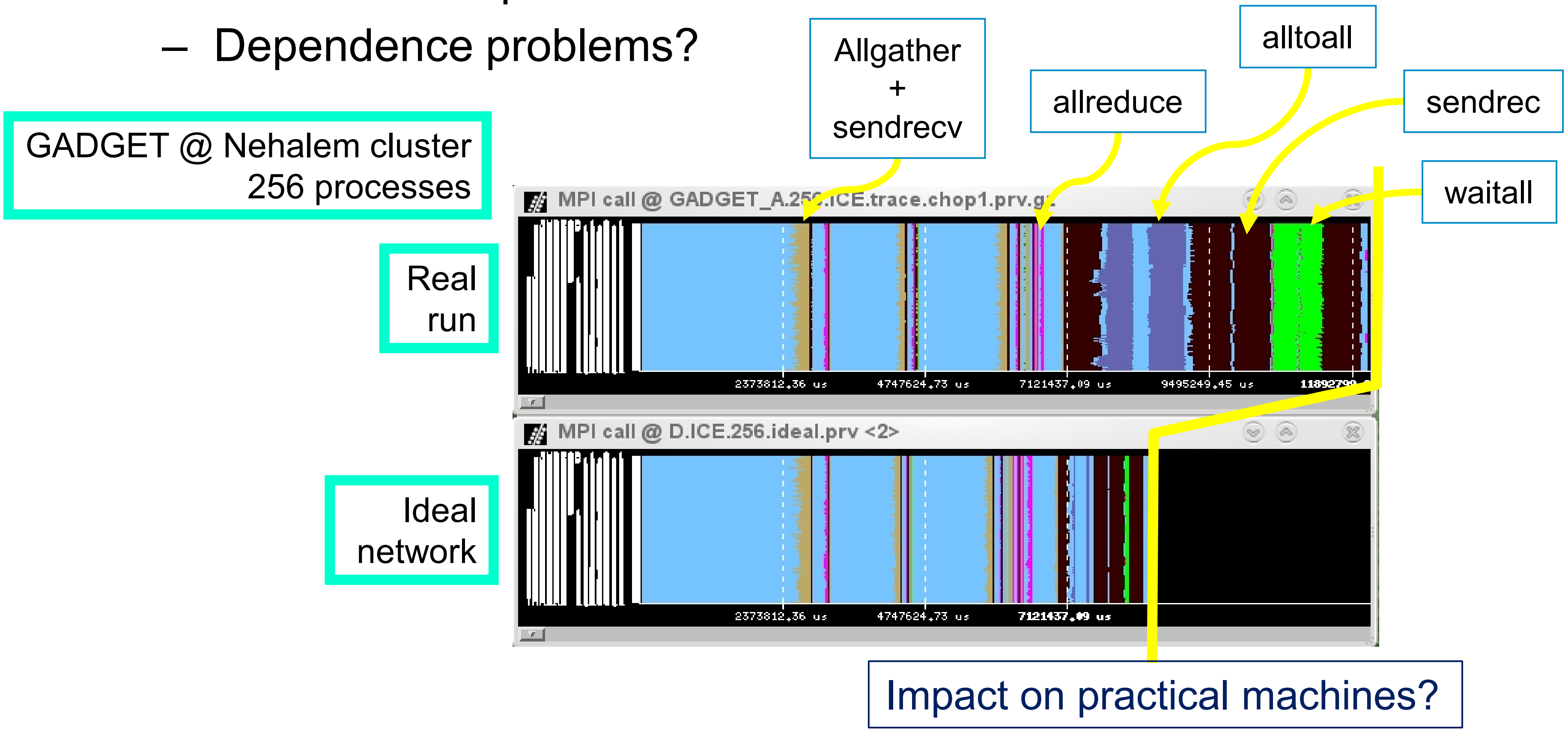




Ideal network machine (instantaneous communication)

The impossible machine: BW = ∞, L = 0

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



GADGET @ Nehalem cluster 256 processes

Real run

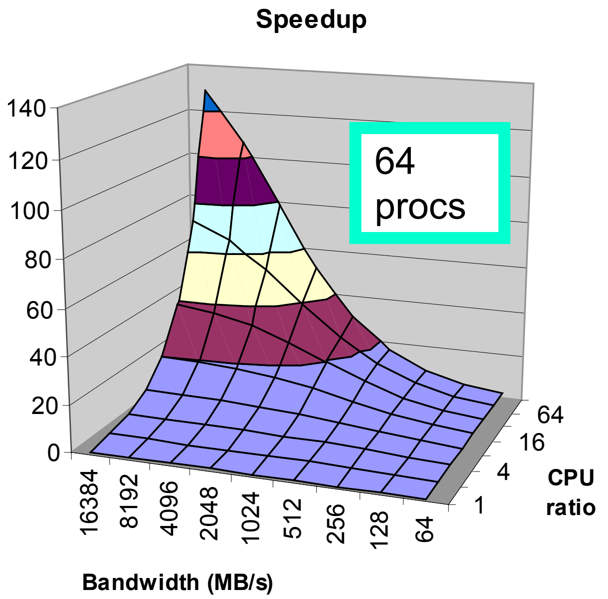
Ideal network

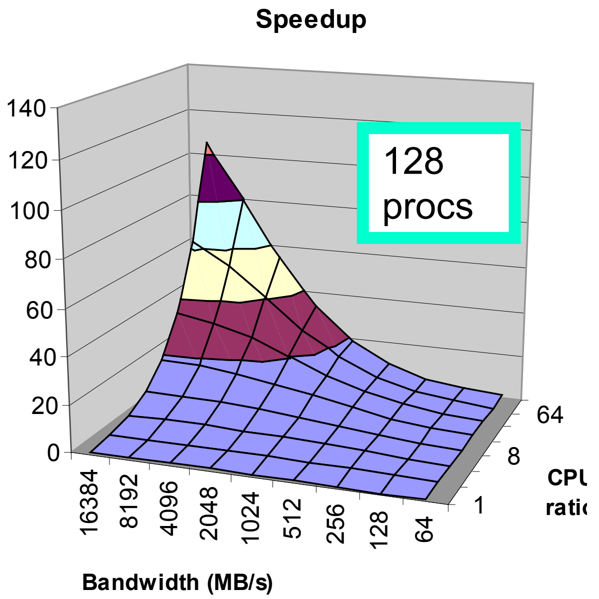
Impact on practical machines?

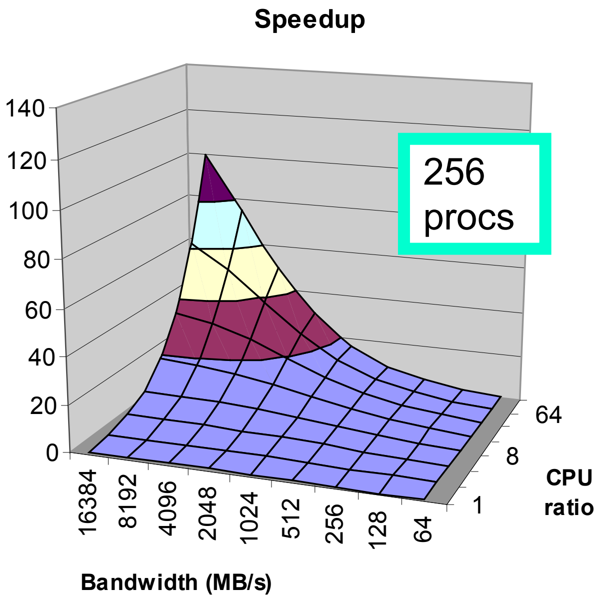
Impact of architectural parameters

- **Ideal speeding up ALL** the computation bursts by the CPUratio factor
  - The more processes the less speedup (higher impact of bandwidth limitations) !!

GADGET

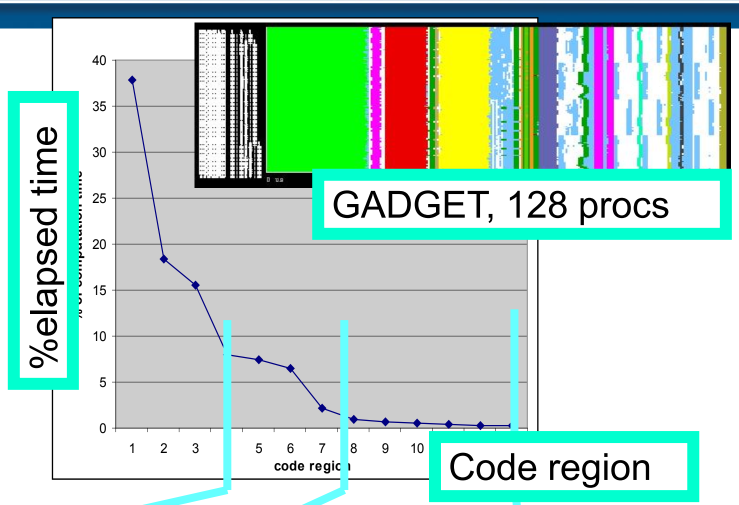






The potential of hybrid/accelerator parallelization

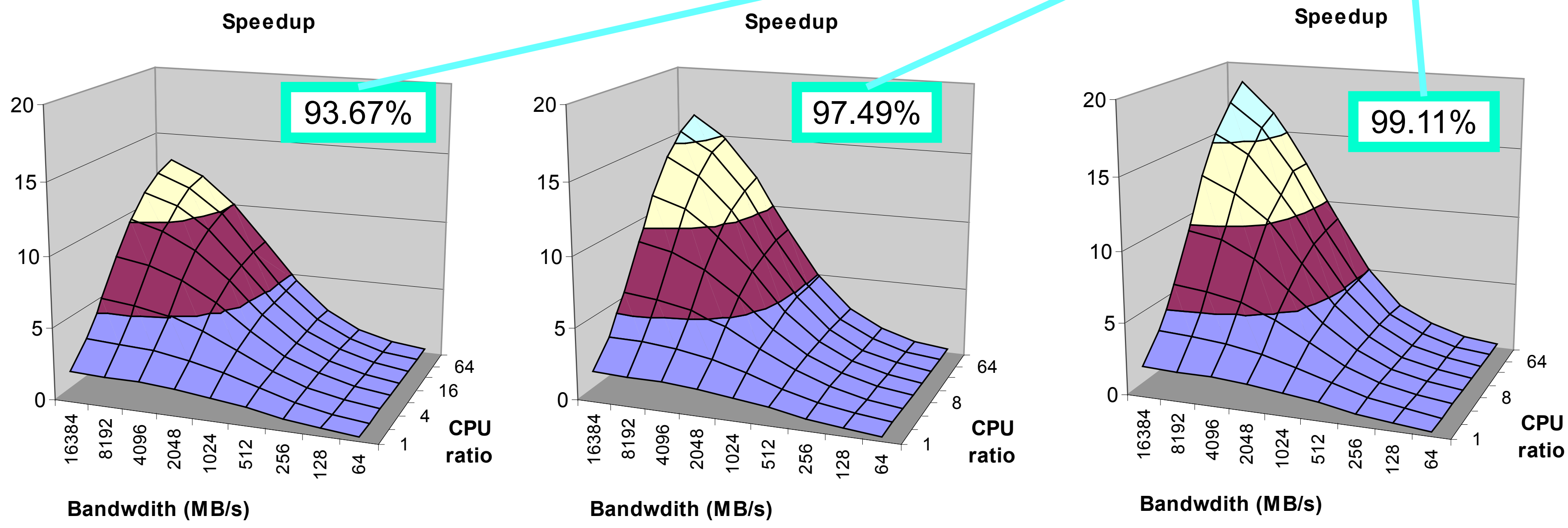
- **Hybrid parallelization**
  - Speedup SELECTED regions by the CPUratio factor
- We do need to overcome the **hybrid Amdahl's law**
  - **→ asynchrony + Load balancing mechanisms !!!**



%elapsed time

GADGET, 128 procs

Code region



93.67%

97.49%

99.11%

Methodology

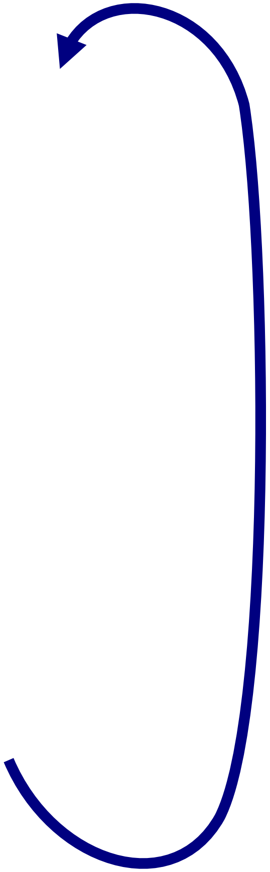
Performance analysis tools objective

Help generate hypotheses

Help validate hypotheses

Qualitatively

Quantitatively

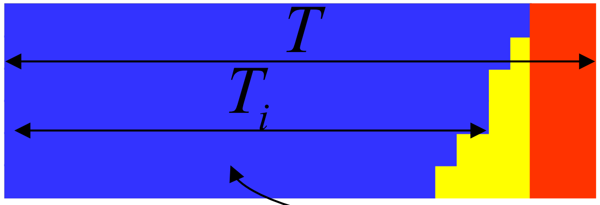


First steps

- Parallel efficiency – percentage of time invested on computation
  - Identify sources for “inefficiency”:
    - load balance
    - Communication /synchronization
- Serial efficiency – how far from peak performance?
  - IPC, correlate with other counters
- Scalability – code replication?
  - Total #instructions
- Behavioral structure? Variability?

Scaling model

$$\eta_{\parallel} = L \quad * B C \quad o \quad m$$



$$eff_i = \frac{T_i}{T}$$

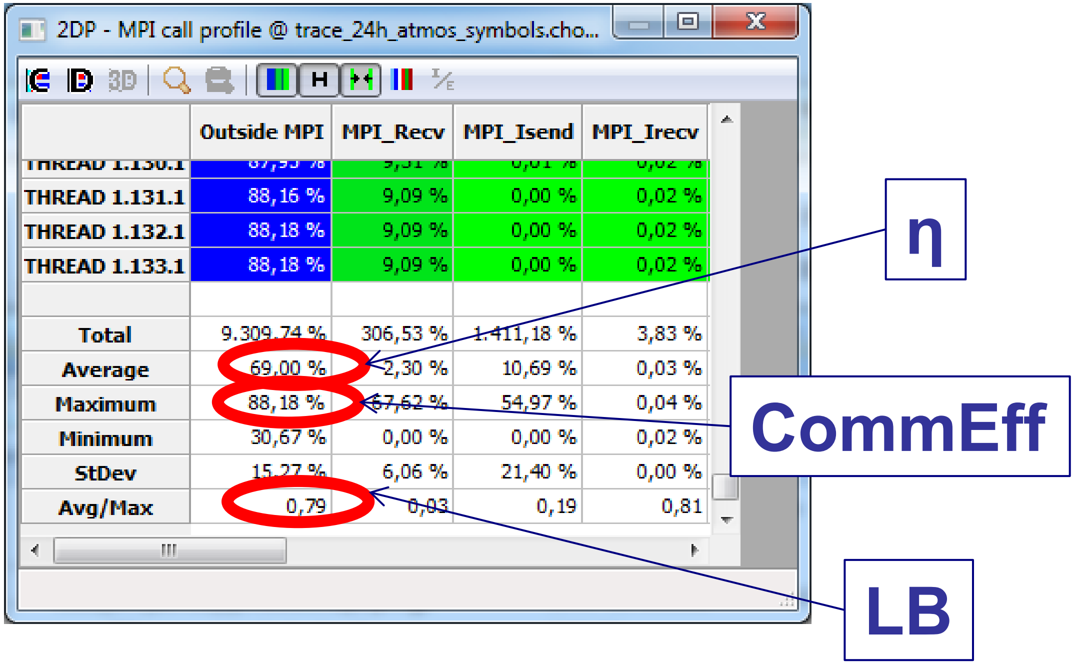
$$CommEff = \max(eff_i)$$

$$LB = \frac{\sum_{i=1}^{P} eff_i}{P * \max(eff_i)}$$

IPC

#instr

Directly from real execution metrics

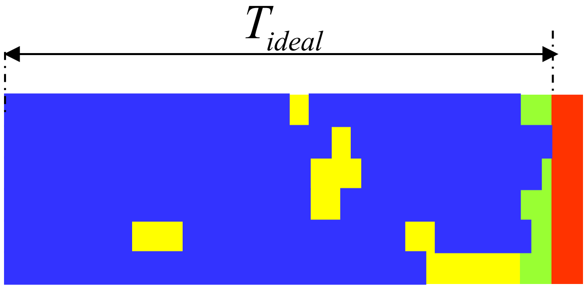


	Outside MPI	MPI_Recv	MPI_Isend	MPI_Irecv
THREAD 1.130.1	87,95 %	9,31 %	0,01 %	0,02 %
THREAD 1.131.1	88,16 %	9,09 %	0,00 %	0,02 %
THREAD 1.132.1	88,18 %	9,09 %	0,00 %	0,02 %
THREAD 1.133.1	88,18 %	9,09 %	0,00 %	0,02 %
Total	9.309,74 %	306,53 %	1.411,18 %	3,83 %
Average	69,00 %	2,30 %	10,69 %	0,03 %
Maximum	88,18 %	67,62 %	54,97 %	0,04 %
Minimum	30,67 %	0,00 %	0,00 %	0,02 %
StDev	15,27 %	6,06 %	21,40 %	0,00 %
Avg/Max	0,79	0,03	0,19	0,81

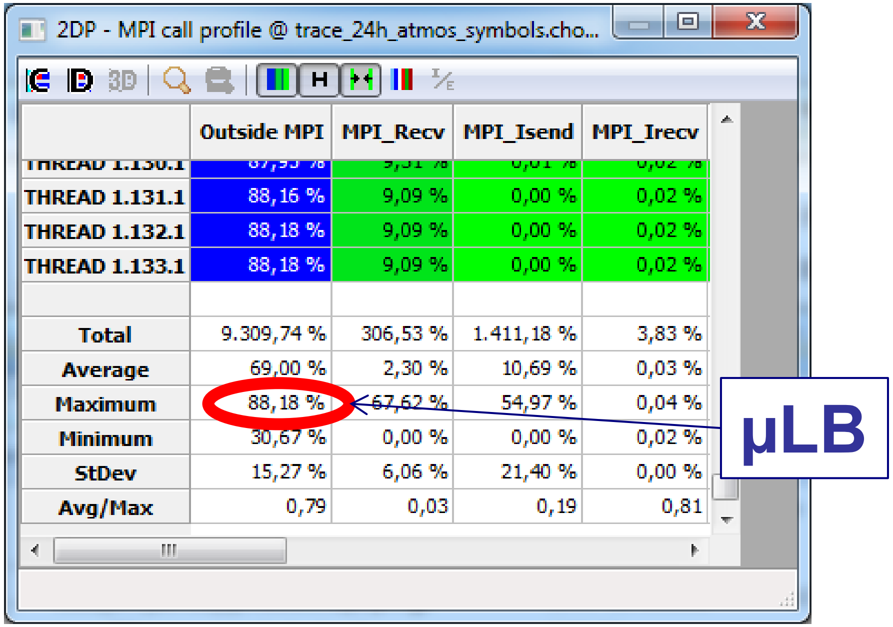
Scaling model

- Dimemas simulation with ideal target
  - Latency =0; BW = ∞

$$C \qquad o = \mu L m * T \;\; B \quad m r$$



Migrating/local load imbalance
Serialization

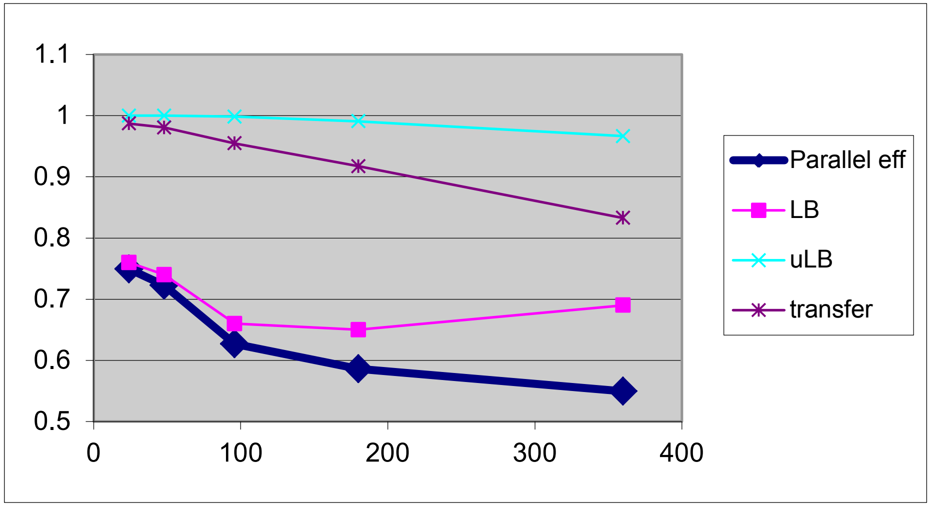


	Outside MPI	MPI_Recv	MPI_Isend	MPI_Irecv
THREAD 1.130.1	87,95 %	9,31 %	0,01 %	0,02 %
THREAD 1.131.1	88,16 %	9,09 %	0,00 %	0,02 %
THREAD 1.132.1	88,18 %	9,09 %	0,00 %	0,02 %
THREAD 1.133.1	88,18 %	9,09 %	0,00 %	0,02 %
Total	9.309,74 %	306,53 %	1.411,18 %	3,83 %
Average	69,00 %	2,30 %	10,69 %	0,03 %
Maximum	88,18 %	67,62 %	54,97 %	0,04 %
Minimum	30,67 %	0,00 %	0,00 %	0,02 %
StDev	15,27 %	6,06 %	21,40 %	0,00 %
Avg/Max	0,79	0,03	0,19	0,81

$$\mu L B = \frac{\max(T_i)}{T_{ideal}} \qquad Transfer = \frac{T_{ideal}}{T}$$

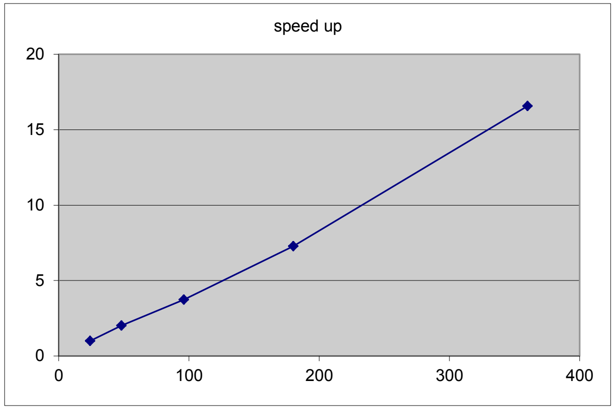
Modelling efficiency

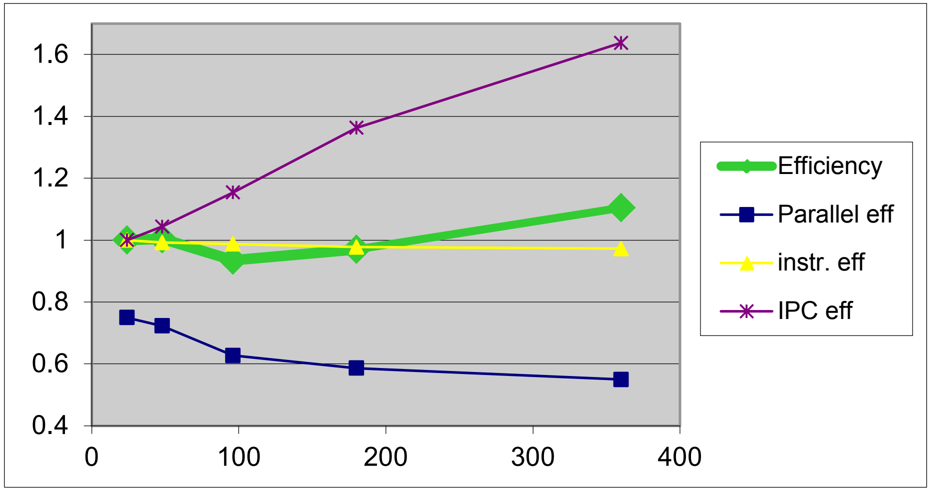
$$\eta_{\parallel} = L \quad * \mu B L \quad * T B \quad r$$



CG-POP mpi2s1D - 180x120

Good scalability !!
Should we be happy?





$$\eta = \eta_{\parallel} * \eta_{i\ n\ s\ t\ r} * \eta_{I\ P}$$

BSC Tools web site

- www.bsc.es/paraver
  - downloads
    - Sources / Binaries
    - Linux / windows / MAC
  - documentation
    - Training guides
    - Tutorial slides

- Getting started
  - Start wxparaver
  - Help → tutorials and follow instructions
  - Follow training guides
    - Paraver introduction (MPI): Navigation and basic understanding of Paraver operation

Short Paraver Demo

BSC tools hand-on

Paraver

- Install the Paraver binaries in your laptop
  - Binaries for linux x86, x86-64, windows & mac (not tested on Mavericks)

- Configure Paraver package
  - Set-up the tutorials

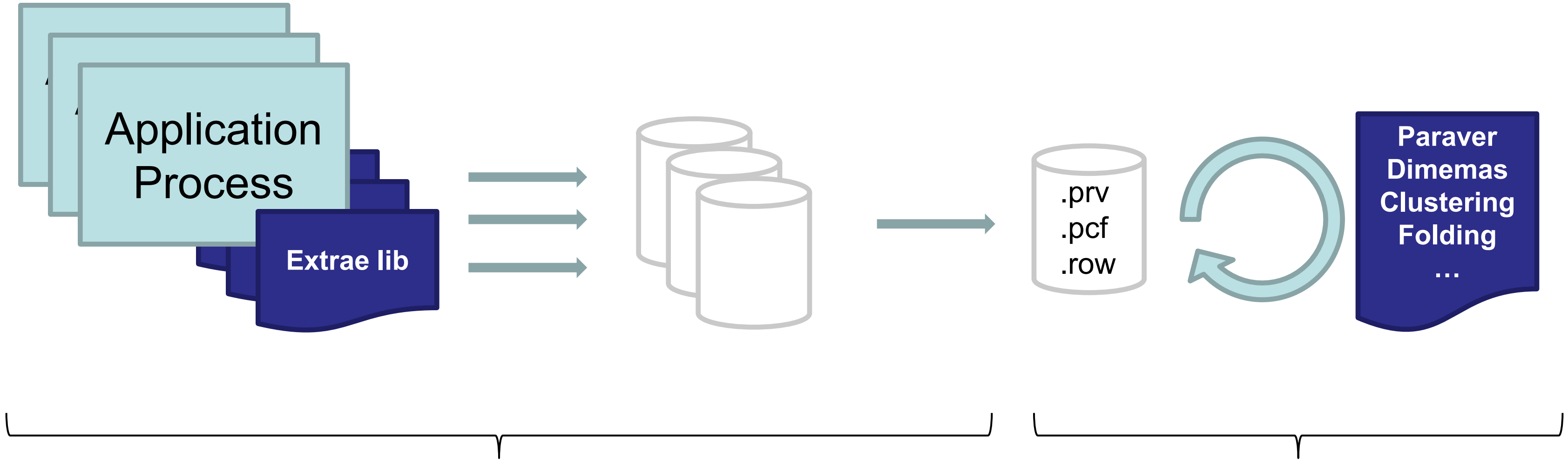
Directory structure

- Copy ~nct00001/gpfs_projects/tuesday_material into your ${HOME}
  - `cp -r /gpfs/projects/nct00/nct00001/tuesday_material ~`

- Contents of tuesday_material
  - bin/
    - Some BSC tools scripts to make your life easy
  - slides/
    - All the slides wrt BSC tools
  - packages/
    - Paraver binaries
  - tutorials/
    - Paraver tutorials
  - documentation/
    - Documentation related to the BSC tools
  - jobscripts/
    - Modified jobscript of the course applications
  - extrae/ | dimemas/ | clustering/ | folding/ | tracking/
    - Files for the different tools used in the Hands-On session

Extrae

Trace generation workflow



Instrumentation

Analysis

Obtaining traces (CGPOP)

- Build the CGPOP application (already done)
  - Copy from `apps/CGPOP`
  - Issue `./build` in your copy

- Generate trace-file for the CGPOP application
  - Change dir to `~/apps/CGPOP/jobscript/marenostrum3`
  - Copy `tuesday_material/jobscripts/cgpop.extrae.lsf`
  - Edit `cgpop.extrae.lsf` and uncomment module load bsctools & choose appropriate TRACE
    - Optionally choose a tracefile name
  - Submit the job
    - `bsub < cgpop.extrae.lsf`

Inside cgpop.extrae.lsf

```
#!/bin/bash

#BSUB -n 24
#BSUB -oo cgpop_%J.out
#BSUB -eo cgpop_%J.err
#BSUB -R"span[ptile=12]"
#BSUB -x # Exclusive use
#BSUB -J cgpop
#BSUB -W 00:10
#BSUB -U tools

# module load bsctools

# Override tracefile name using this environment variable
# export TRACE_NAME=cgpop.linux_icc.180x120.24tasks.prv

# Choose appropriate instrumentation type for your application
# export TRACE=~/tuesday_material/extrae/trace.mpi.c.sh # For C-based applications without sampling (MPI)
# export TRACE=~/tuesday_material/extrae/trace.mpi.sampling.c.sh # For C-based applications with sampling (MPI)
# export TRACE=~/tuesday_material/extrae/trace.mpi.f.sh # For Fortran-based applications without sampling (MPI)
# export TRACE=~/tuesday_material/extrae/trace.mpi.sampling.f.sh # For Fortran-based applications with sampling (MPI)

time mpirun -np 24 --npersocket 6 --bind-to-core ${TRACE} ../mpi2s1D/cgpop.linux_icc.180x120
```

Inside trace.*.sh

- What is inside the shellscript files?

```
#!/bin/bash

# Workaround for MN3
export TMPDIR=$TMPDIR/extrae
mkdir -p ${TMPDIR}

export EXTRAE_CONFIG_FILE=${HOME}/tuesday_material/extrae/extrae.xml
export LD_PRELOAD=${EXTRAE_HOME}/lib/libmpitrace.so

$@
```

Library alternatives

Library	Serial	MPI	OpenMP	pthread
libseqtrace	✓			
libmpitrace[f][1]		✓		
libomptrace			✓	
libpttrace				✓
libompitrace[f] [1]		✓	✓	
libptmpitrace[f] [1]		✓		✓

[1] for Fortran codes

Extrae configuration (I)

```
<counters enabled="yes">
  <cpu enabled="yes" starting-set-distribution="cyclic">
   <set enabled="yes" domain="all" changeat-time="100000us“>
PAPI_TOT_INS,PAPI_TOT_CYC,PAPI_L1_DCM,PAPI_L2_DCM,PAPI_L3_TCM,PAPI_FP_INS,PAPI_BR_MSP
      </set>
      <set enabled="yes" domain="all" changeat-time="100000us">
PAPI_TOT_INS,PAPI_TOT_CYC,PAPI_LD_INS,PAPI_SR_INS,RESOURCE_STALLS,PAPI_BR_UCN,PAPI_BR_CN,PAPI_VEC_SP
      </set>
      <set enabled="yes" domain="all" changeat-time="100000us">
PAPI_TOT_INS,PAPI_TOT_CYC,RESOURCE_STALLS:LB,RESOURCE_STALLS:RS,RESOURCE_STALLS:SB,RESOURCE_STALLS:ROB,PAPI_VEC_DP
     </set>
  </cpu>
  <network enabled="no" />
  <resource-usage enabled="no" />
  <memory-usage enabled="no" />
</counters>
```

Hardware counters

Extrae configuration (II)

```
<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>
                <gather-mpits enabled="no" />
</storage>
```

Storage options

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

Buffering options

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

MPI section

```
<sampling enabled="no" type="default" period="50m" variability="20m" />
```

Sampling

```
<callers enabled="yes">
                <mpi enabled="yes">1-3</mpi>
                <sampling enabled="yes">1-5</sampling>
</callers>
```

Callstack information

```
 <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 generation

Paraver

Using Paraver

- Analyse the results with Paraver

- Follow tutorials

- Paraver navigation
  - Load configuration files
  - Generate new views
  - Contextual menus
    - Synchronize between windows
    - Zoom & Fit time-scale
    - Draw mode
  - Save configuration files

Dimemas

Using Dimemas

- **Step 1**: CGPOP Paraver trace (chop) available at

```
cd ${HOME}/tuesday_material/dimemas/
```

- **Step 2**: Define Dimemas configuration
  - We supply a basic Dimemas configuration (MN3.cfg)

- **Step 3**: Execute Dimemas

```
${HOME}/tuesday_material/bin/dimemas-sim.sh <input_trace>
<dimemas_cfg>
```

- **Step 4**: Analyse the results with Paraver

Step 2 in detail

- Run the Dimemas GUI

```
> DimemasGUI
```

- Load MN3.cfg file

- Tune target machine parameters
  - Processors 10x faster
  - Set network bandwidth to ¼ to the original one

dimemas-sim.sh

```
[…]
#PARAVER_TRACE=${1}
PARAVER_TRACE=${1}.prv
DIMEMAS_TRACE=${1}.dim
DIMEMAS_CFG=${2}

[…]

# Translate from .prv to .dim
if [ ! -f ${DIMEMAS_TRACE} ]; then
  echo
  echo "Dimemas trace does not exist. Translating input Paraver trace"
  echo
  prv2dim ${PARAVER_TRACE} ${DIMEMAS_TRACE}
  echo
  echo "==============================================================================="
fi

# Simulate parameter -S 32K fixed by default
echo
echo "Executing Dimemas"
echo
Dimemas -t --dim ${DIMEMAS_TRACE} -S 32K -p ${OUTPUT_PARAVER_TRACE} ${DIMEMAS_CFG}

echo "==============================================================================="
```

Paraver to Dimemas trace translation (if required)

Dimemas execution

Dimemas outputs

- A Paraver trace of the simulation
  - Output trace name: **`<input_trace>.<cfg>.prv`**
- Run Paraver and compare with the original trace

Recommended steps

1. Follow guidelines for paraver basic navigation (Tut #1)

   Tracefile provided

2. Adapt scripts to instrument your application
3. Obtain a tracefile
4. Follow first steps of methodology guidelines (Tut #5)
   - Parallel efficiency
   - Distribution of computations
   - Instruction balance
   - …
5. Depending on the diagnosis use clustering, folding, dimemas…

Performance Analytics

Clustering

Running the cluster analysis

- **Step 1**: Previously generated trace SU3 trace available at:

```
cd ${HOME}/tuesday_material/clustering/
```

- **Step 2**: Tune the configuration xml
  - We supply a configuration *xml* (cluster.xml)
    - Completed Instructions vs. IPC
    - DBSCAN parameter Eps = 0.01, MinPoints = 10
    - Adjusted filters for this trace

- **Step 3**: execute the cluster analysis:

```
${HOME}/tuesday_material/bin/clusterize.sh
<trace_without_prv>
```

- To launch the clustering binary directly

```
BurstClustering -a -d <xml_file> -i <in_trace> -o
<out_trace>
```

Clustering definition XML

```
<clustering_definition use_duration="no" apply_log="yes"
normalize_data="yes" duration_filter=“1000" threshold_filter="0">

  <clustering_algorithm name="DBSCAN">
    <epsilon>.01</epsilon>
    <min_points>4</min_points>
  </clustering_algorithm>

  <clustering_parameters>

    <mixed_events apply_log="no" name="IPC" operation="/">
      <event_type_a>42000050</event_type_a>
      <event_type_b>42000059</event_type_b>
      <factor>1.0</factor>
    </mixed_events>

    <single_event apply_log="yes" name="PAPI_TOT_INS">
      <event_type>42000050</event_type>
<!--        <range_min>4e7</range_min> -->
      <factor>1.0</factor>
    </single_event>

  </clustering_parameters>
  […]
```

Duration Filters and Normalizations

DBSCAN parameters

Clustering dimensions using Paraver events

Clustering definition XML (cont.)

```
  [...]

 <extrapolation_parameters all_counters="yes"/>

 <output_plots all_plots="no">

   <plot_definition raw_metrics="yes">
     <x_metric title="IPC">IPC</x_metric>
     <y_metric title="Instructions Completed">PAPI_TOT_INS</y_metric>
   </plot_definition>

 </output_plots>

</clustering_definition>
```

Events to characterize the bursts

Definition of output plots. Must be events previously defined

Cluster analysis outputs

- GNUplot scripts

  ```
  gnuplot <input_trace>.clustered.[…].gnuplot
  ```

- A Paraver trace with clusters information
  - Output trace name: `<input_trace>.clustered.prv`

- Clusters statistics (including the extrapolation)
  - Statistics file name: `<input_trace>.clustered.clusters_info.csv`
  - You can visualize it with an editor or import it to a spreadsheet

Results evaluation

- What to do if…
  - … you find too many clusters?
    - Increase the value of Eps.
  - … there too many points (> 100K)
    - Increase the filters (duration / instructions)
    - You may need to cut the trace

Tracking

Using the ‘tracking’ tool

- **Step 1:** Previously clustered traces (SU3) available at:

```
cd ${HOME}/tuesday_material/tracking
```

- **Step 2**: Execute the tracking analysis:

```
${HOME}/tuesday_material/bin/track.sh <in_trace_1>
<in_trace_2> … <in_trace_N>
```

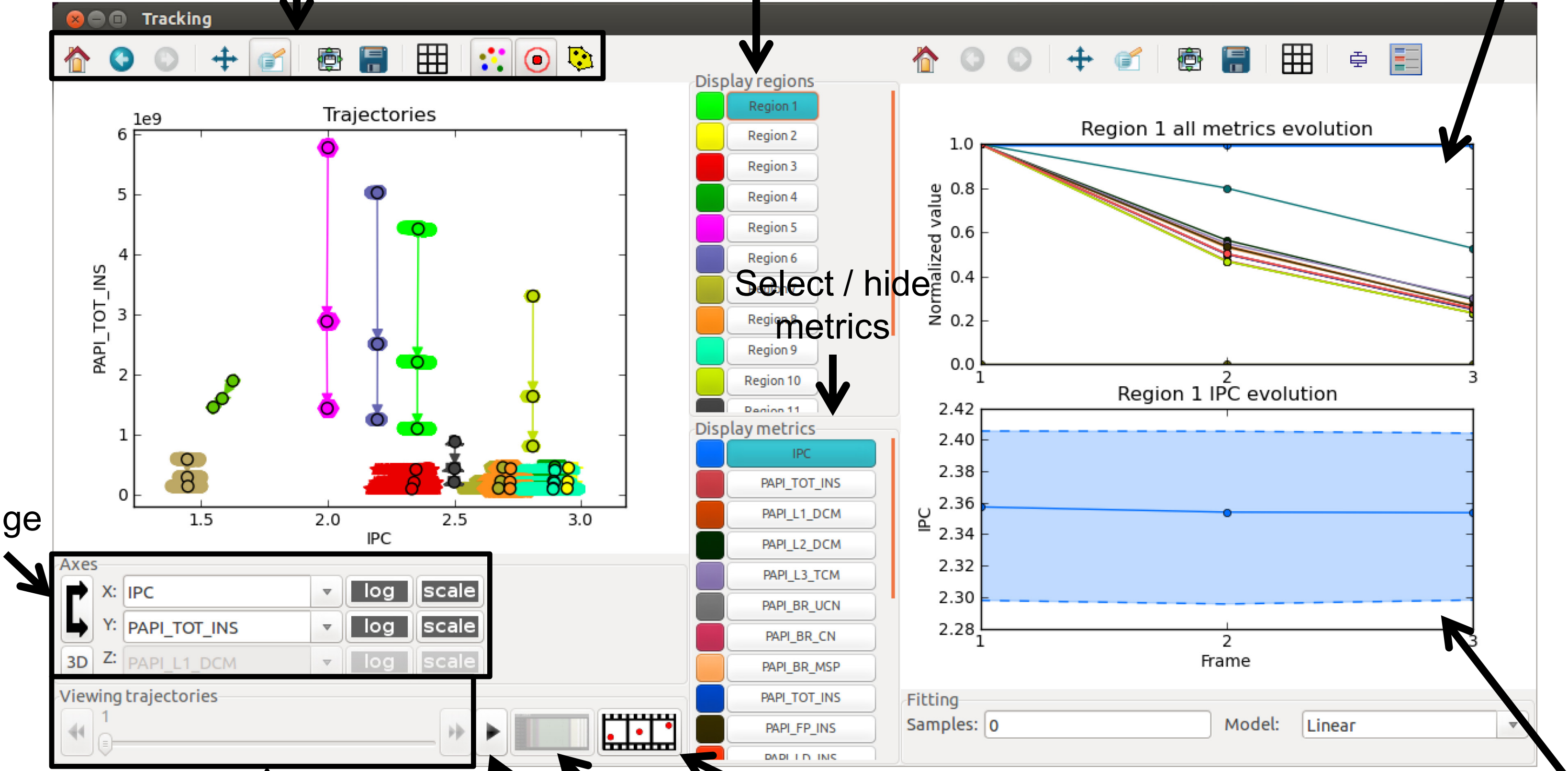
Tracking outputs

- **Step 3**: Load the results with the visualizer

```
xtrack TRACKING.RESULTS.xtrack
```

  - Other outputs
    - Scatter plots
      - gnuplot **<input_trace>.clustered.[…].gnuplot.scaled**
      - gnuplot **TRACKING.RESULTS.recolored.multiplot**
    - Paraver traces with clusters renamed
      - wxparaver **<input_trace>.clustered.tracked.prv**

The ‘xtrack’ visualizer



X: IPC

Y: PAPI_TOT_INS

Z: PAPI_L1_DCM

Samples: 0

Model: Linear

Manipulate the graph area

Select / hide clusters

Correlation of all metrics for selected cluster across experiments

Select / hide metrics

Change axes

Switch between frame/trajectory view

Open Paraver trace

Run timed animation

Average/dispersion for selected metric/cluster across experiments

Navigate to previous/next experiment

Refining tracking results

- What to do if…
  - … there's too many objects?
    - Filter clusters that represent low percentage of time
      - Add argument "-m <time-percentage>" (i.e. –m 5)
  - … over aggregation?
    - Turn off the tracking heuristic based on callstack.
      - Remove argument: "-c <callstack-depth>"
  - … the application is not SPMD?
    - Turn off the tracking heuristics based on alignment.
      - Add argument: "-a 1"

Folding

Folding

- **Step 1**: Previously clustered traces (SU3) with sampling available at:

```
cd ${HOME}/tuesday_material/folding
```

- **Step 2**: Apply folding

```
bsub < folding.lsf
```

- **Step 3**: Browse results

Browsing the results (for the brave)

- The folding generates plots for the combinations of
  - Cluster
  - Performance counter
    - Plus all counter slopes combined
    - Plus MIPS and remaining counters in terms of ctr/instruction
    - *Architecture impact*
    - *Stall distribution*
- For instance

```
# gnuplot -persist *Cluster_1.*PAPI_TOT_INS*gnuplot
# gnuplot -persist *Cluster_1.*stalldistribution*gnuplot
# gnuplot -persist *Cluster_3.*stalldistribution*gnuplot
# gnuplot -persist *Cluster_3.*architectureimpact*gnuplot
```

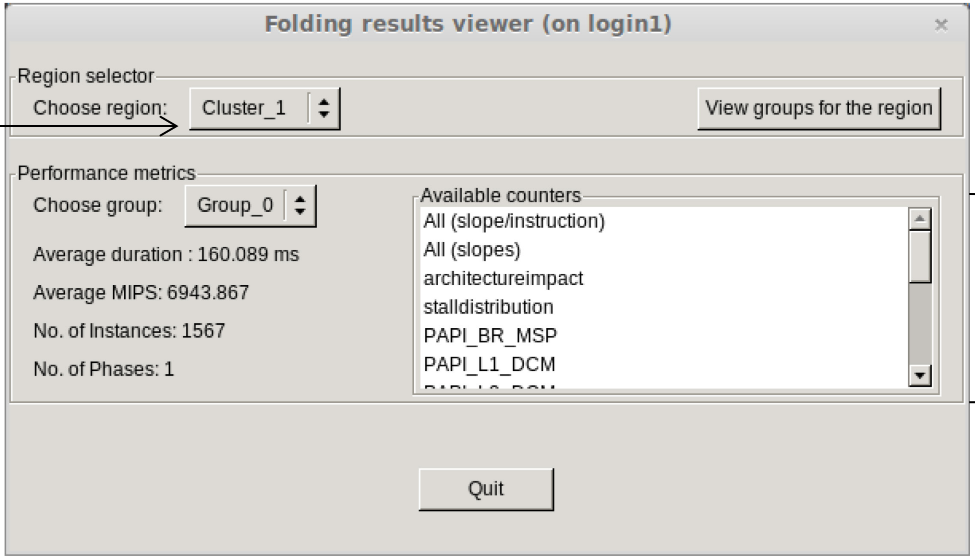
Browsing the results (for the practical)

- Execute in the directory results

```
# wxfolding-viewer *wxfolding
```

- In the GUI choose the cluster you want to analyze, and double click on the performance counter

Choose cluster to analyze



Folding results viewer (on login1)

Region selector

Choose region: Cluster_1

Performance metrics

Choose group: Group_0

Average duration : 160.089 ms

Average MIPS: 6943.867

No. of Instances: 1567

No. of Phases: 1

Available counters

All (slope/instruction)
All (slopes)
architectureimpact
stalldistribution
PAPI_BR_MSP
PAPI_L1_DCM

Choose metric or model

Barcelona
Supercomputing
Center
Centro Nacional de Supercomputación

THANKS

Detailed material

Semantic Module

Basic functions of time

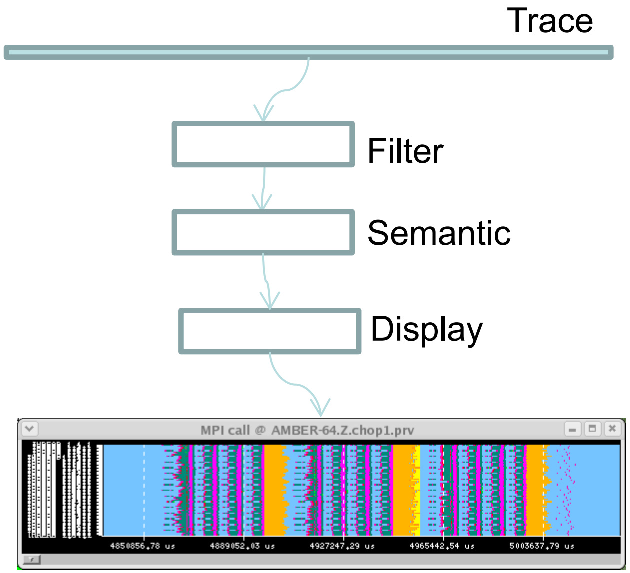
- The **filter module** presents a subset of the trace to the semantic module. Each thread th is described by
  - A sequence of events $Ev_i, i \in N$, states $St_i, i \in N$ and communications $C_i, i \in N$
  - For each event let $T(Ev_i)$ be its time and $V(Ev_i)$ its value
  - For each state let $T_s(St_i)$ be its start time $T_e(St_i)$ its stop time and $V(St_i)$ its value
  - For each Communication let $T_S(C_i)$ be its send time, $T_R(C_i)$ its receive time, $Sz(C_i)$ its size.
  - $Partner(C_i)$ and $Dir(C_i) \in \{send, recv\}$ identify the partner process and direction of the transfer

- **Semantic module** builds

$$s(t) = S(i), t \in [t_i, t_{i+1}), i \in \mathrm{N}$$

Function of time

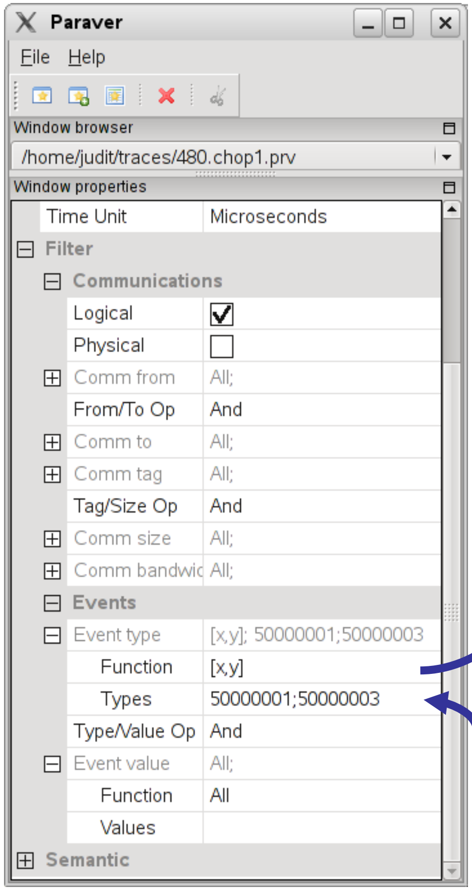
Series of values



Filter module

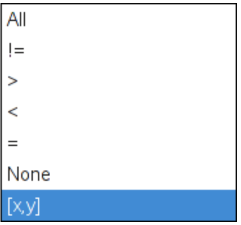
Communications that pass through the filter

Events that pass through the filter



/home/judit/traces/480.chop1.prv

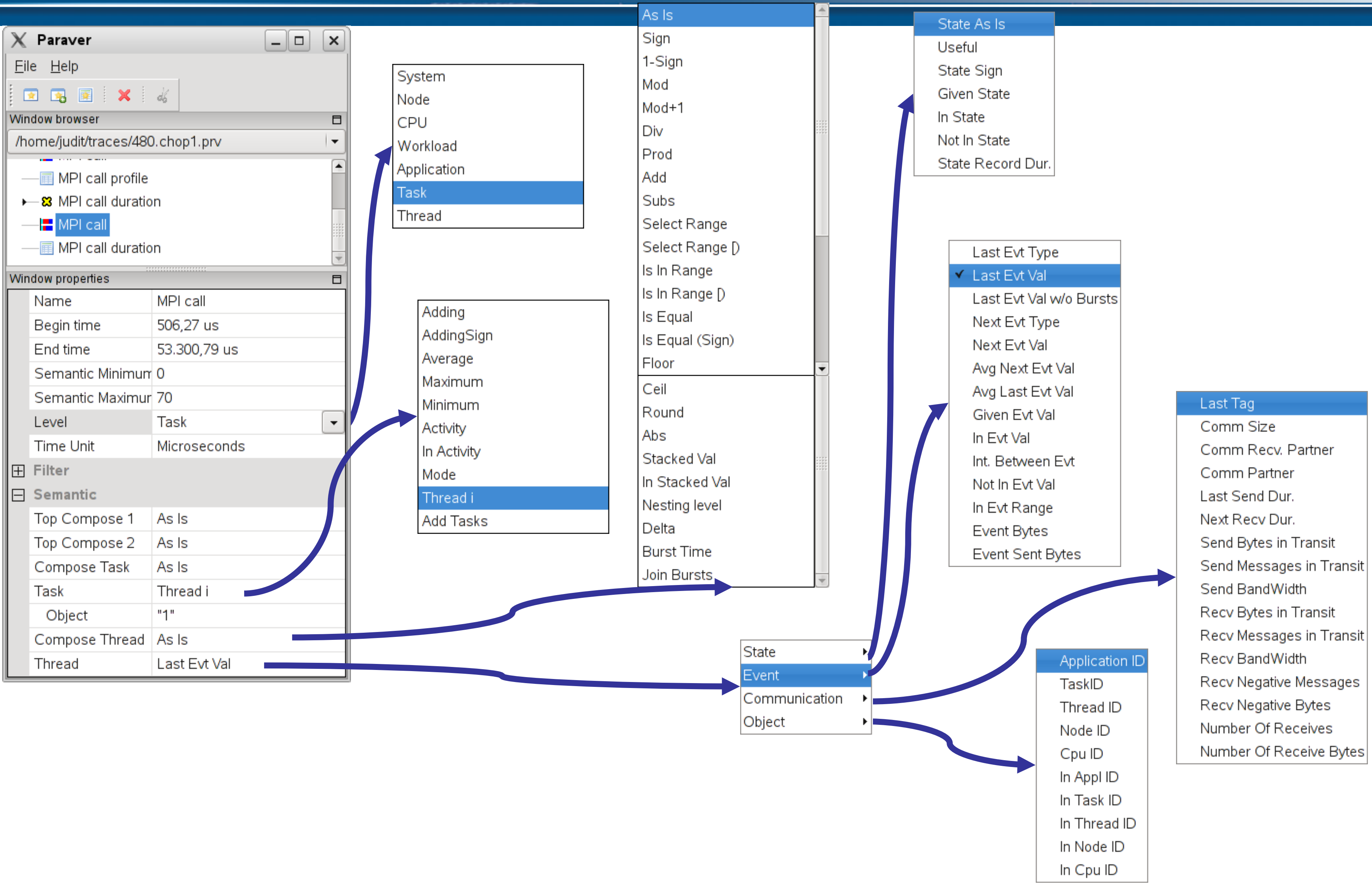
Time Unit	Microseconds
⊟ Filter	
⊟ Communications	
Logical	☑
Physical	□
⊞ Comm from	All;
From/To Op	And
⊞ Comm to	All;
⊞ Comm tag	All;
Tag/Size Op	And
⊞ Comm size	All;
⊞ Comm bandwic	All;
⊟ Events	
⊟ Event type	[x,y]; 50000001;50000003
Function	[x,y]
Types	50000001;50000003
Type/Value Op	And
⊟ Event value	All;
Function	All
Values	
⊞ Semantic	



- All
- !=
- >
- <
- =
- None
- [x,y]

Show list of event types

Semantic module: Control



Paraver

File Help

Window browser

/home/judit/traces/480.chop1.prv

- MPI call profile
- MPI call duration
- MPI call
- MPI call duration

Window properties

Name	MPI call
Begin time	506,27 us
End time	53.300,79 us
Semantic Minimum	0
Semantic Maximum	70
Level	Task
Time Unit	Microseconds
⊞ Filter	
⊟ Semantic	
Top Compose 1	As Is
Top Compose 2	As Is
Compose Task	As Is
Task	Thread i
Object	"1"
Compose Thread	As Is
Thread	Last Evt Val

Semantic module

- From Events to functions of time
  - Last event value $S(i) = V(Ev_i)$
  - Next event value $S(i) = V(Ev_{i+1})$
  - Average Next Event Value $S(i) = \frac{V(Ev_{i+1})}{T(Ev_{i+1}) - T(Ev_i)}$
  - Interval btw. Events $S(i) = T(Ev_{i+1}) - T(Ev_i)$

Semantic module

- From communication records to functions of time
  - Send Bytes

    $$s(t)=\sum_j Sz(C_j), j \mid (T_S(C_j)<t)\wedge(T_R(Cj)>t)\wedge(Dir(Cj)==send)$$

  - Send Bandwidth

    $$s(t)=\sum_j \frac{Sz(C_j)}{T_R(C_j)-T_S(C_j)}, j \mid (T_S(C_j)<t)\wedge(T_R(Cj)>t)\wedge(Dir(Cj)==send)$$

  - Msgs in transit

    $$s(t)=\sum_j sign(j), j \mid (T_S(C_j)<t)\wedge(T_R(Cj)>t)\wedge(Dir(Cj)==send)$$

  - Recv. Bandwidth

    $$s(t)=\sum_j \frac{Sz(C_j)}{T_R(C_j)-T_S(C_j)}, j \mid (T_S(C_j)<t)\wedge(T_R(Cj)>t)\wedge(Dir(C_j)==recv)$$

  - Rec. Negative Msgs

    $$s(t)=\sum_j sign(j), j \mid (T_R(C_j)<t)\wedge(T_S(Cj)>t)\wedge(Dir(Cj)==recv)$$

  - Comm. Partner

    $$s(t)=Partner(C_j), j \mid (T_S(C_j)<t)\wedge(T_R(Cj)>t)$$

  - Bytes btw. Events

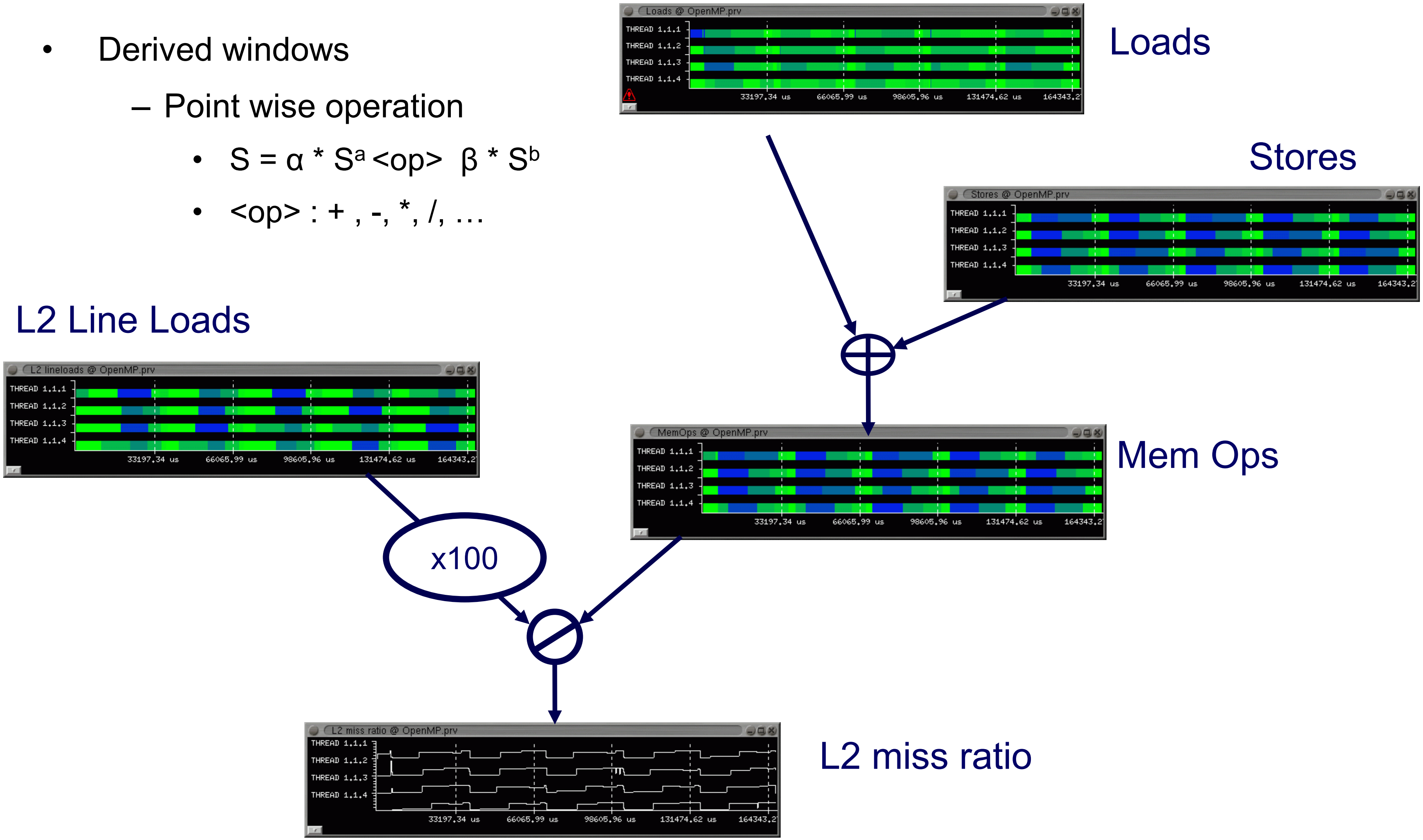
    $$S(i)=\sum_j Sz(C_j), j \mid T_S(C_j)\in\left[T(Ev_i),T(Ev_{i+1})\right)\vee T_R(C_j)\in[T(Ev_i),T(Ev_{i+1}))$$

Composition

- S’(t) = f(S(t)) S’ = f ° S
  - Sign $S'(t) = sign(S(t))$
  - 1-sign $S'(t) = 1 - sign(S(t))$
  - Select range $S'(t) = S(t) \in [a,b] ? S(t) : 0$
  - Sign ° Is equal $S'(t) = sign(S(t) = a ? S(t) : 0)$
  - Delta $S'(t) = S_{i+1} - S_i$
  - Stacked value

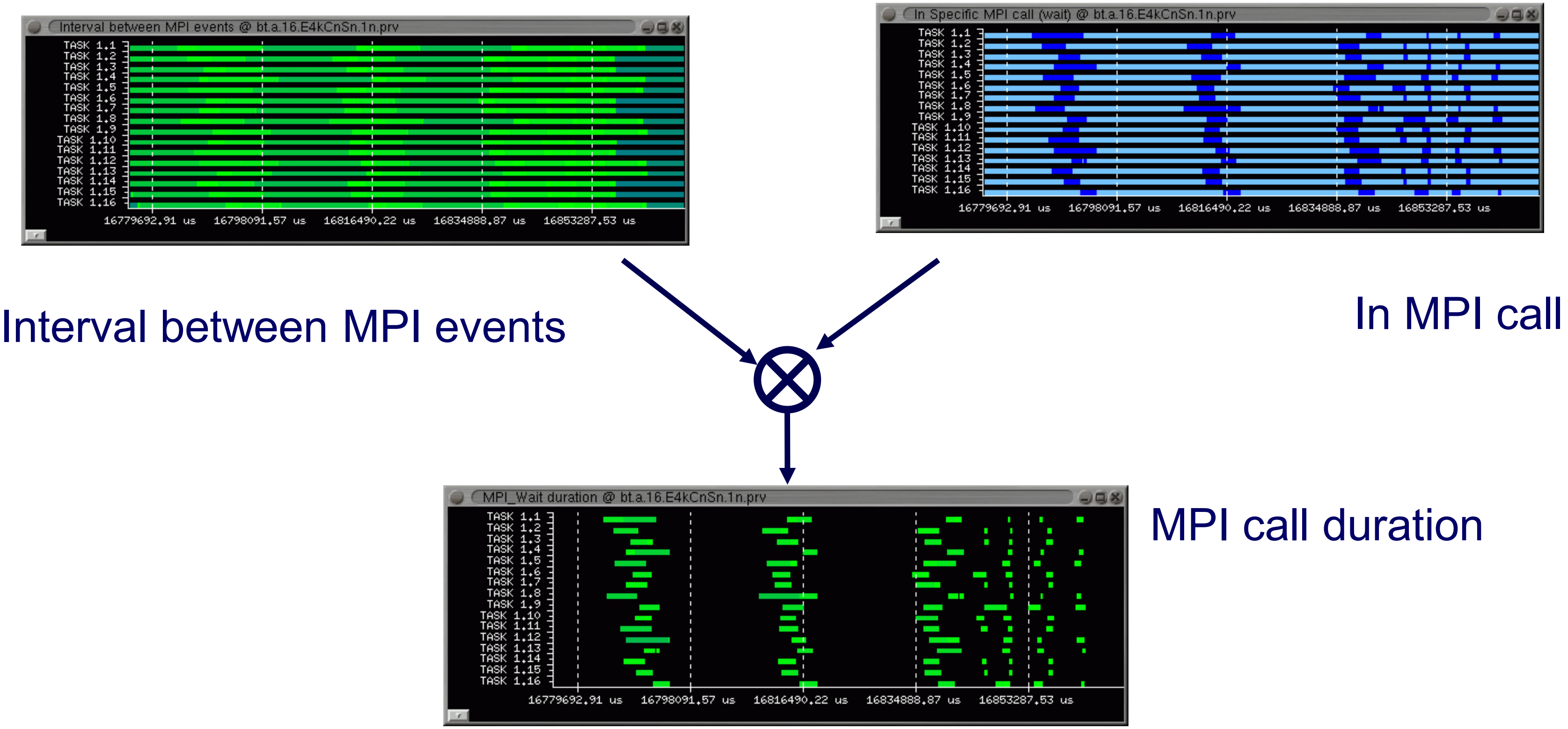
Semantic module

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



Semantic module

- Derived windows
  - Point wise operation
    - $S = \alpha * S^a \text{ <op> } \beta * S^b$
    - <op> : + , -, *, /, …

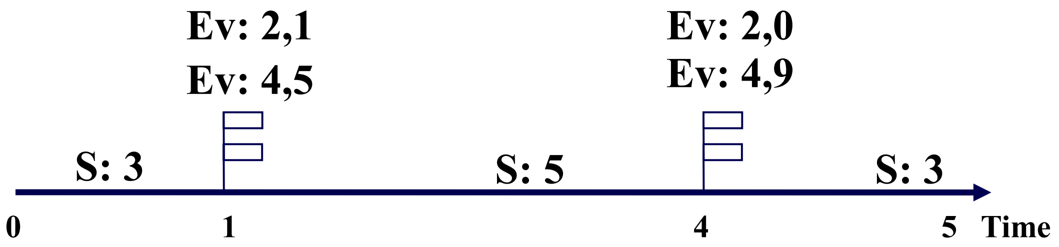


Interval between MPI events

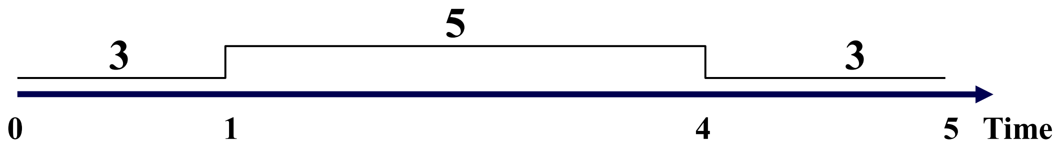
In MPI call

MPI call duration

Semantic module: Examples

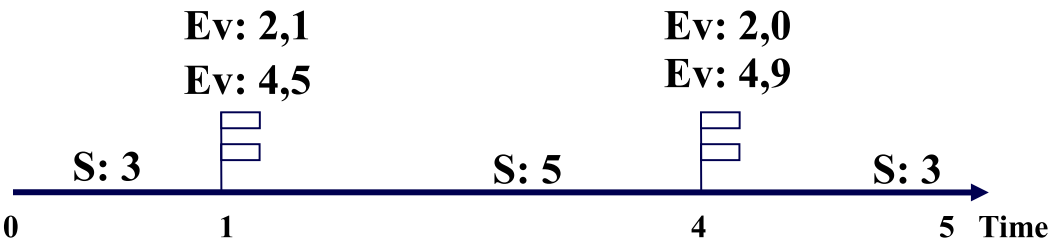


- Thread function: State as is

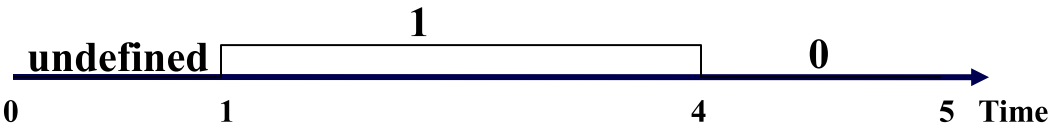


- Useful for
  - Global thread activity: computing, idle, fork/join, waiting,…..

Semantic module: Examples

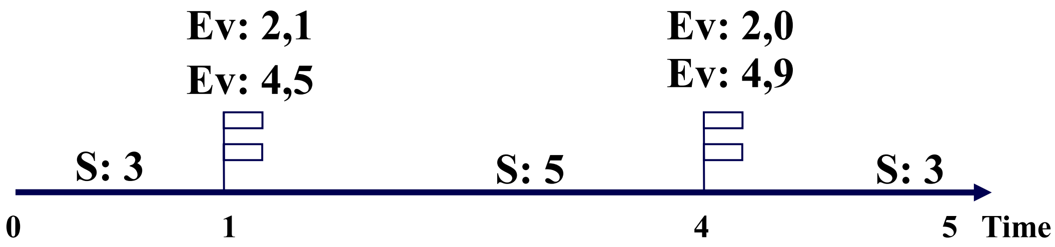


- Filter: type == 2
  - Thread function: Last event value

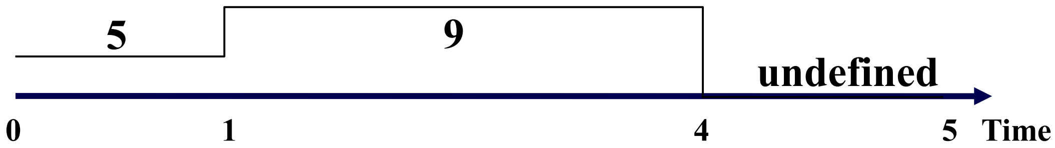


- Useful for
  - In parallel region
  - Mutual exclusion
  - Variable values: iteration,….

Semantic module: Examples

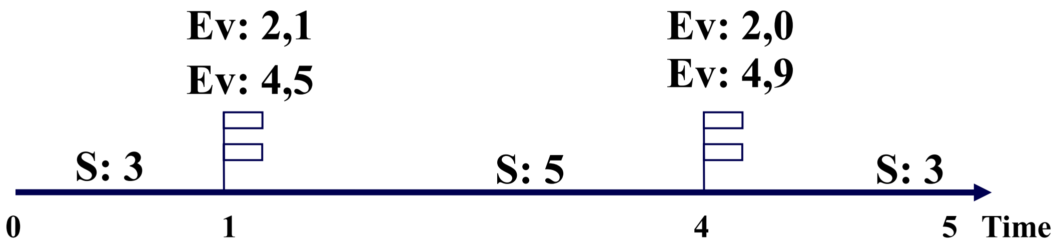


- Filter: type == 4
  - Thread function: Next event value

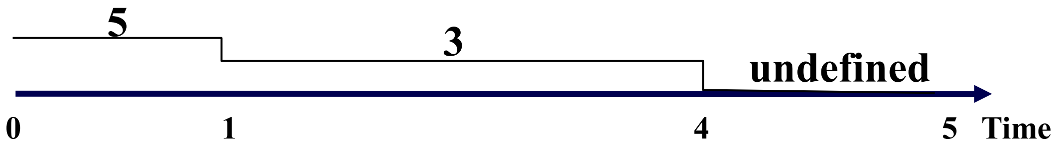


- Useful for
  - Hwc events (TLB, L1 misses,...) within interval

Semantic module: Examples

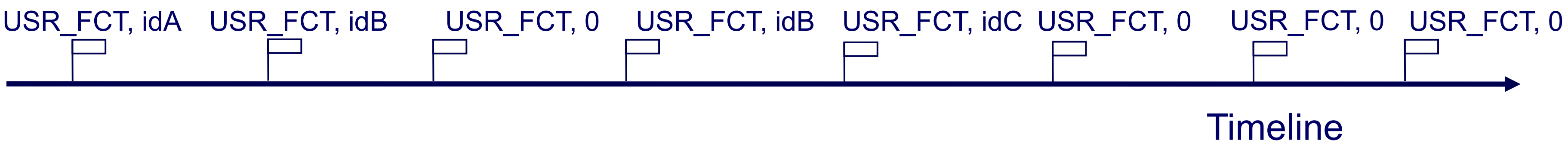


- Filter: type == 4
  - Thread function: Average next event value

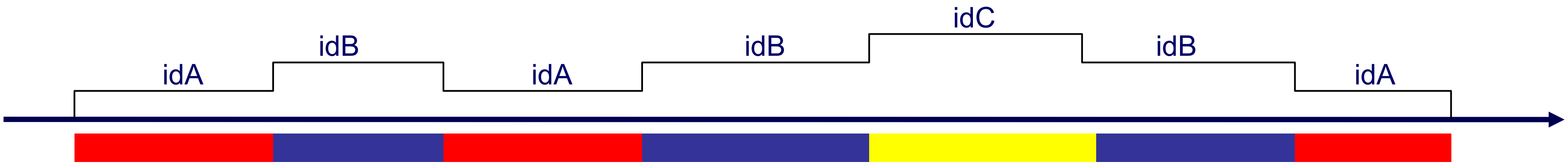


- Useful for
  - Hwc events (TLB, L1 misses,...) per time unit within interval

Semantic module: Examples



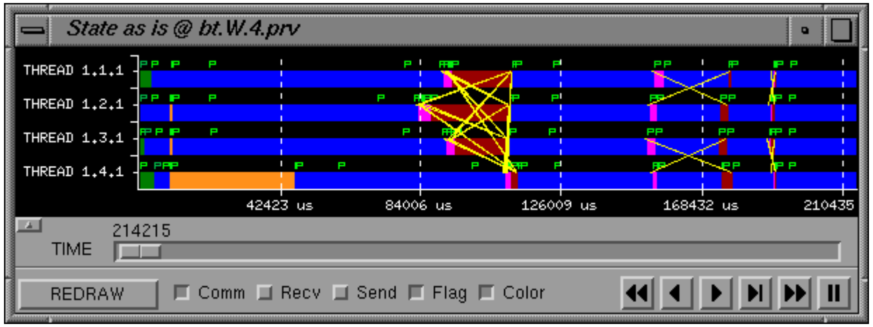
- Filter: type == USR_FCT
  Thread function: Last event value
  Compose: Stacked value

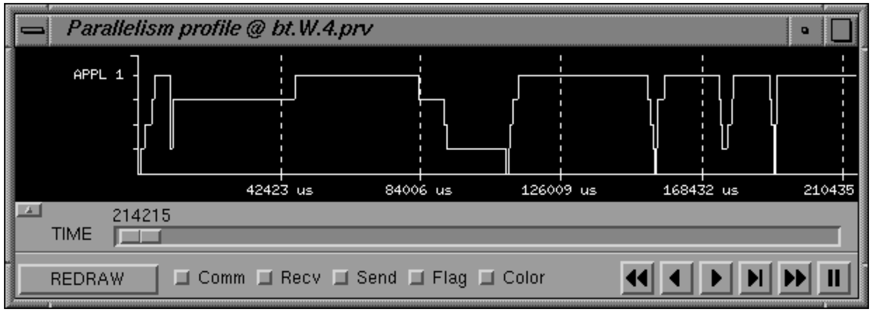


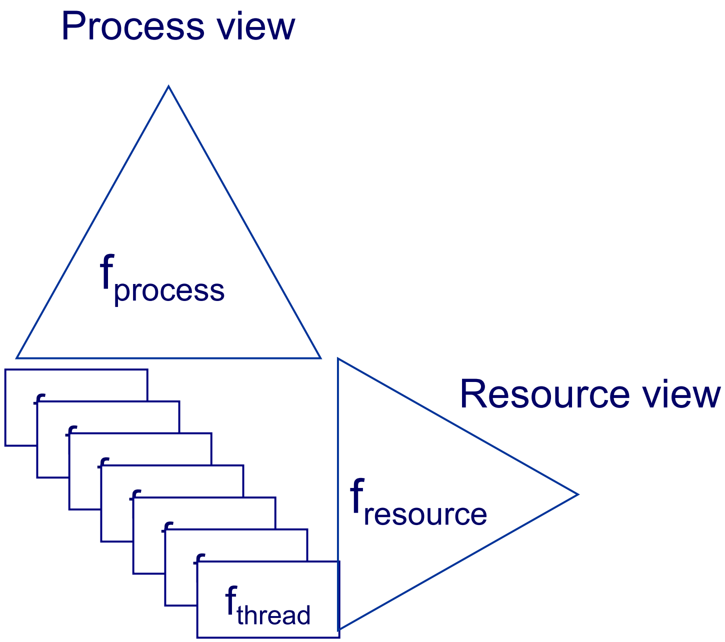
- Useful for
  - Routine

Semantic module perspective

- Process model
  - Thread, task, application, workload
- Resource model
  - CPU, node, system






Process view

$f_{process}$

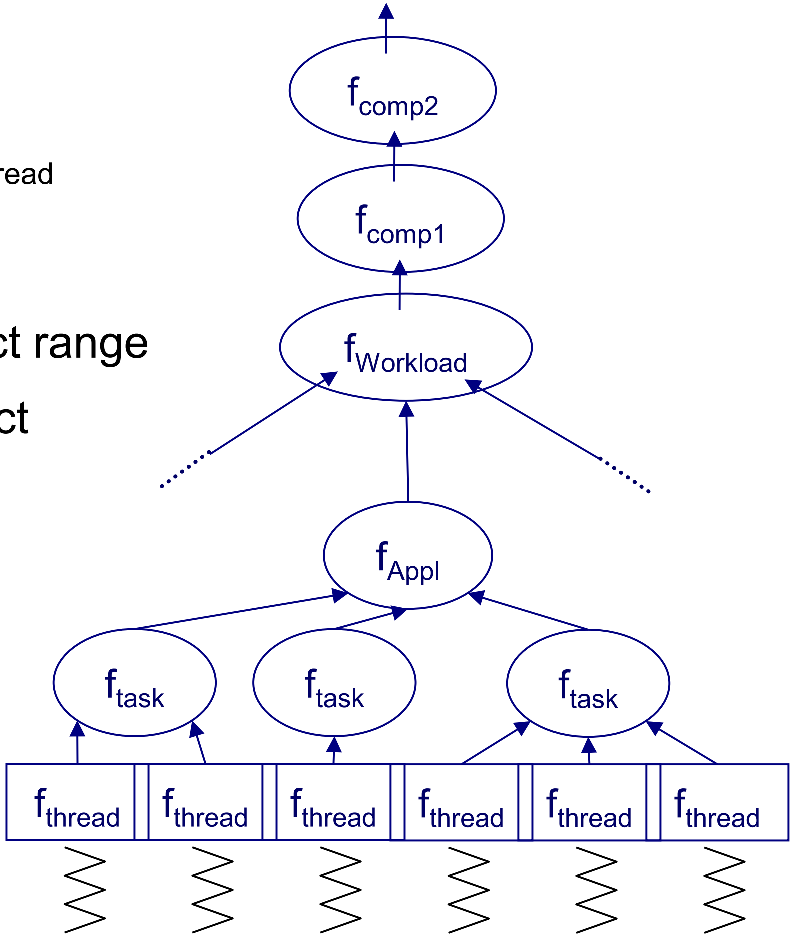
Resource view

$f_{resource}$

f_{thread}

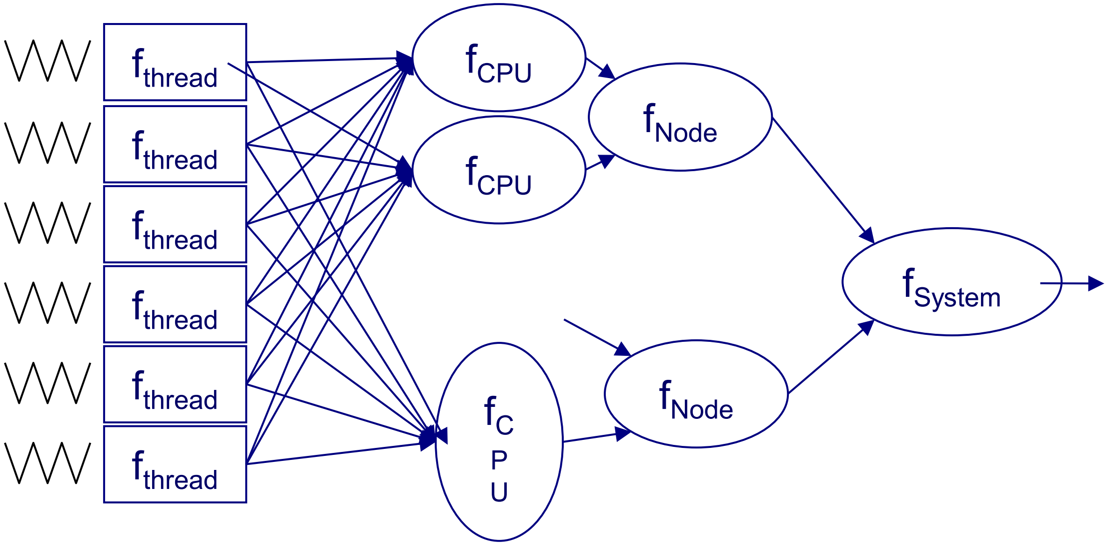
Process model perspective

- Semantic value: S(t)
- $S = f_{comp2} \circ f_{comp1} \circ f_{Workload} \circ f_{Application} \circ f_{task} \circ S_{thread}$
- Semantic functions
  - f_{comp2}, f_{comp1}: sign, mod, div, in range, select range
  - $f_{Application,}$ $f_{Workload}$: add, average, max, select
  - f_{task} : add, average, max, select
  - S_{thread}: in state, useful, given state,
  - last event value,
  - next event value,
  - average next event value
  - interval between events, …



Resource model perspective

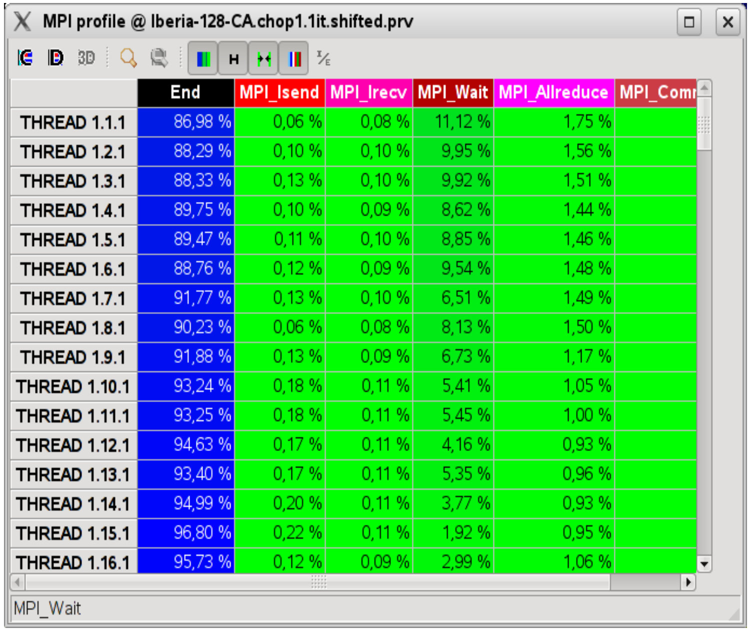
- $Sf_{resource} = f_{comp2} \circ f_{comp1} \circ f_{System} \circ f_{Node} \circ f_{CPU} \circ S_{thread}$
- Semantic functions
  - f_{System} : add, average, max, select
  - f_{Node} : add, average, max, select
  - f_{CPU} : active thread, select
  - S_{thread}: in state, useful, given state, next event value, thread_id



Analysis Module

How to read profiles

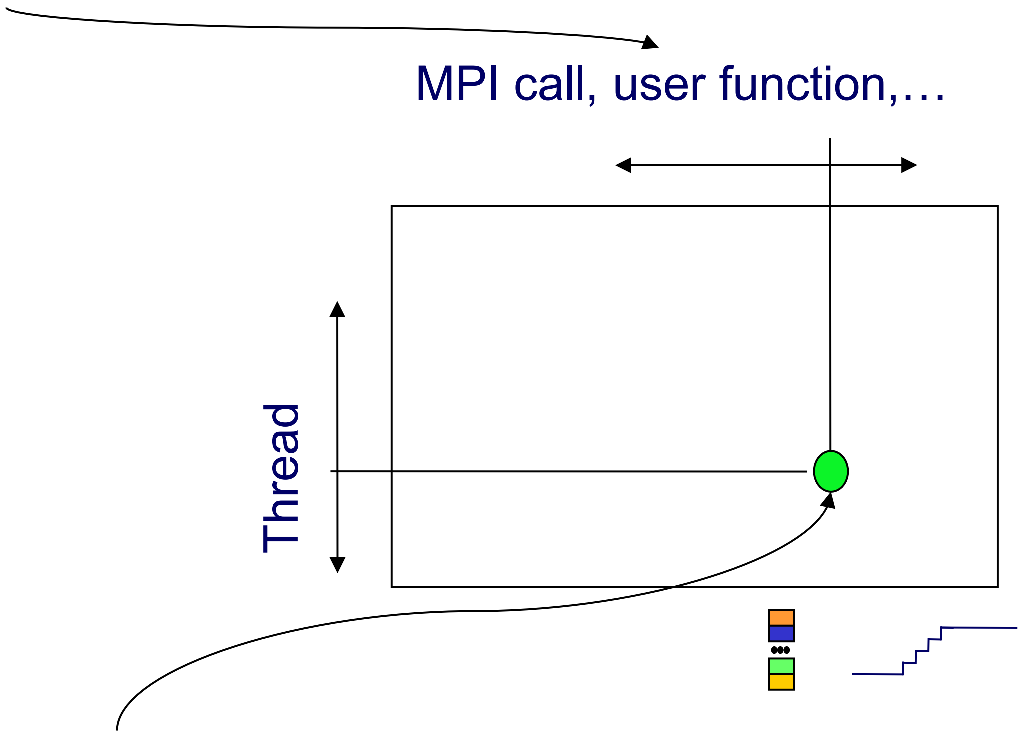
One columns per specific value of categorical **Control window**



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

	End	MPI_Isend	MPI_Irecv	MPI_Wait	MPI_Allreduce	MPI_Comm
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



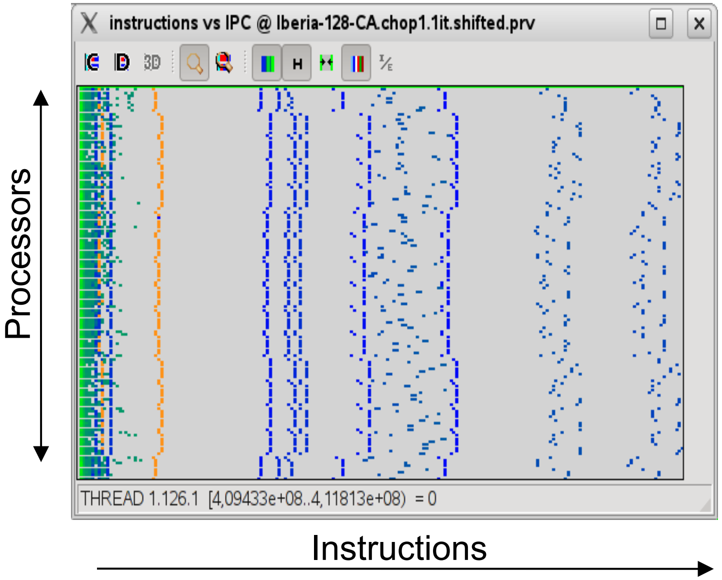
MPI call, user function,…

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

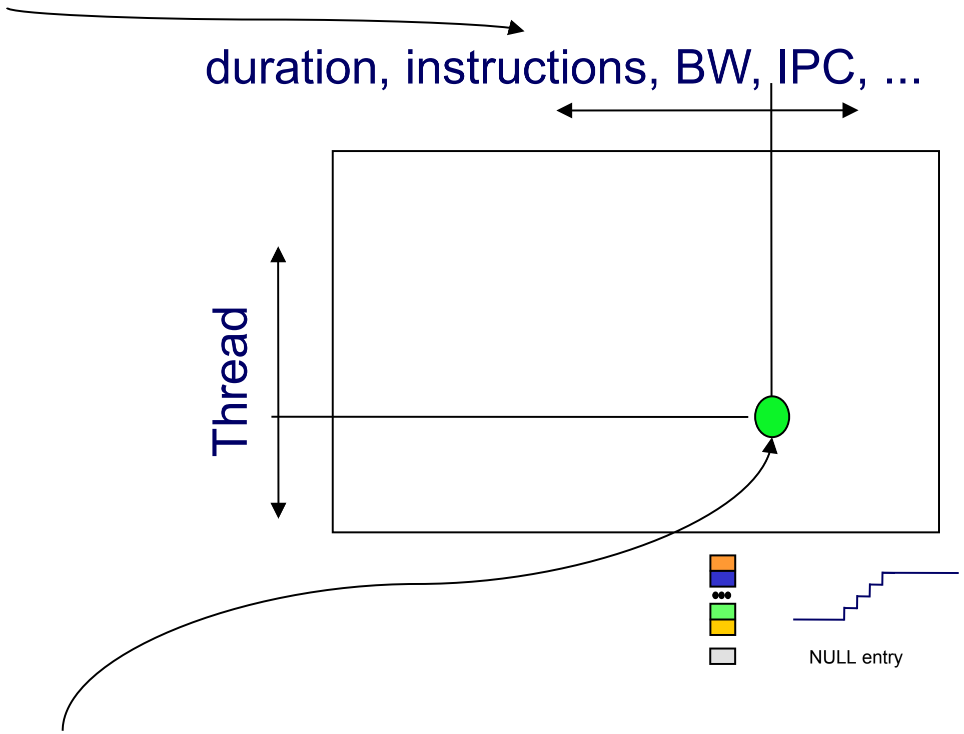
Relevant statistics:
Time, %time, #bursts, Avg. burst time
Average of **Data window**

How to read histograms

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


instructions vs IPC @ Iberia-128-CA.chop1.1it.shifted.prv

THREAD 1.126.1 [4,09433e+08..4,11813e+08) = 0


duration, instructions, BW, IPC, ...

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**

Tables

- Single flexible quantitative analysis mechanism
- Let
  - cw$_1$ and cw$_2$ two views we will call control views
  - dw a view we will call data window

For each window w

$$S^{w}_{th}(t) = S^{w}_{th}(i), t \in \left[t^{w}_{i}, t^{w}_{i+1}\right)$$

- For each control window we define a set of bins

$$bin^{cw}_{j} = \left[range^{cw}_{j}, range^{cw}_{j+1}\right) \qquad range^{cw}_{j+1} = range^{cw}_{j} + delta^{cw}$$

- And the discriminator functions

$$\delta^{cw}_{j}(t) = ((S^{cw}(t) \in bin^{cw}_{j})?1:0)$$

$$\delta_{j,k}(t) = \delta^{cw_1}_{j}(t) * \delta^{cw_2}_{k}(t)$$

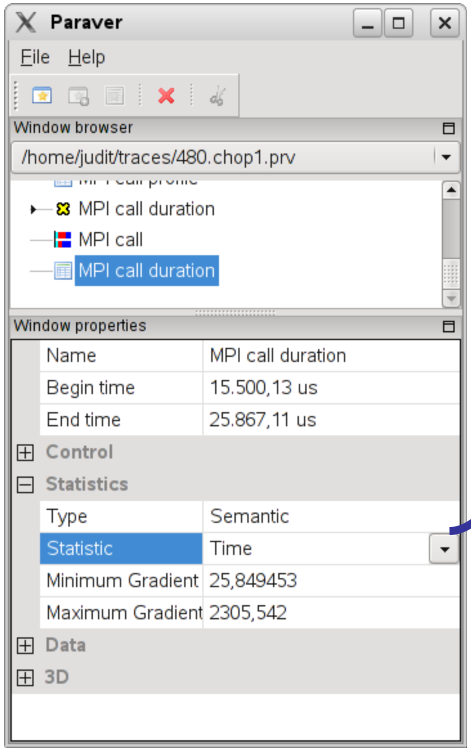
Identify regions with cw’s within the (j,k) bin

- The 3D analysis module computes a cube (or plane in the case of 2D) of statistics

$$M(thread, j, k) = statistic(S^{dw}_{th}(t) * \delta_{th,j,k}(t))$$

- Where the statistic can represent the average value, the number of intervals,….

2D analysis module



Window browser

/home/judit/traces/480.chop1.prv

- MPI call profile
- MPI call duration
- MPI call
- MPI call duration

Window properties

Name	MPI call duration
Begin time	15.500,13 us
End time	25.867,11 us
⊞ Control	
⊟ Statistics	
Type	Semantic
Statistic	Time
Minimum Gradient	25,849453
Maximum Gradient	2305,542
⊞ Data	
⊞ 3D	

- Time
- % Time
- % Time Not Zero
- % Window Time
- # Bursts
- % # Bursts
- Integral
- Average value
- Maximum
- Average Burst Time
- Stdev Burst Time
- Average per Burst
- Average value != 0
- # Bursts != 0
- Sum bursts

$$Time(th, j, k) = \int_{t_{start}}^{t_{end}} \delta_{th,j,k}(t)dt$$

$$\%Time(th, j, k) = \frac{\int_{t_{start}}^{t_{end}} \delta_{th,j,k}(t)dt}{t_{end} - t_{start}}$$

$$\%TimeNotZero(th, j, k) = \frac{\int_{t_{start}}^{t_{end}} \delta_{th,j,k}(t)dt}{\sum_j \int_{t_{sctart}}^{t_{end}} \delta_{tn,j,k}(t)dt}$$

$$NumBurst(th, j, k) = i_{end} - i_{start} + 1$$

$$i_{start} = \min(i) \mid t_i > t_{start}, i_{end} = \max(i) \mid t_i < t_{end}$$

$$Integral(th, j, k) = \int_{t_{start}}^{t_{end}} S_{th}^{dw}(t)\delta_{th,j,k}(t)dt$$

$$Average(th, j, k) = \frac{\int_{t_{start}}^{t_{end}} S_{th}^{dw}(t)\delta_{th,j,k}(t)dt}{\int_{t_{start}}^{t_{end}} \delta_{th,j,k}(t)dt}$$

$$Maximum(th, j, k) = \max(S_{th}^{dw}(t)\delta_{th,j,k}(t)), t = [t_{start}, t_{end})$$

$$SumBurst(th, j, k) = \sum_{i_{start}}^{i_{end}} S_{th}^{dw}(i)\delta_{th,j,k}(t_i), i_{start} = \min(i) \mid t_i > t_{start}, i_{end} = \max(i) \mid t_i < t_{end}$$

Distributed Configurations

Distribution of cfg directories

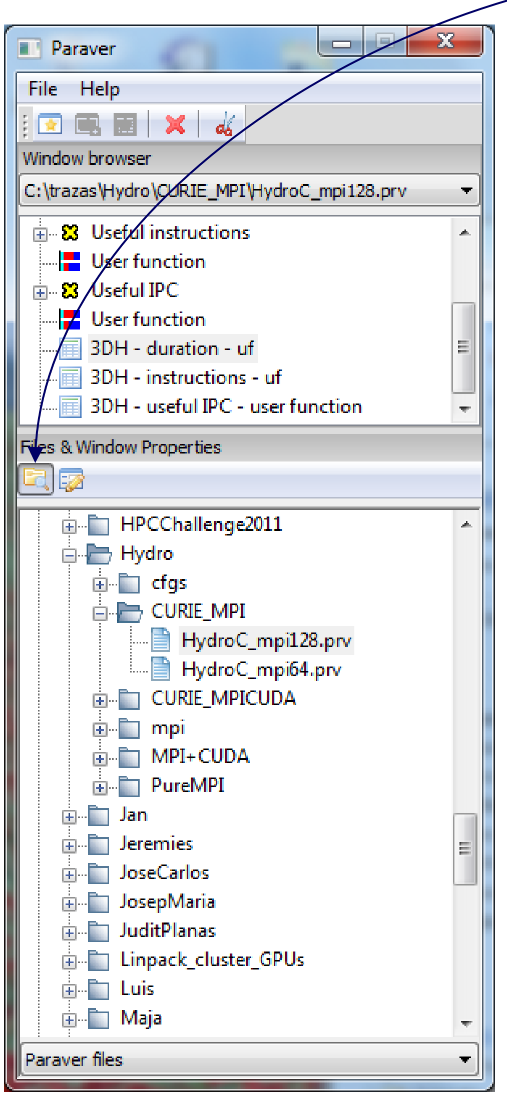
$PARAVER_HOME/cfgs

- CFG
  - General
    - including basic views (timelines) and analyses (2/3D profiles), including views of the user functions and call-stack
  - Counters_PAPI
    - Hardware counter derived metrics. Grouped in directories for
      - Program: related to algorithmic/compilation (i.e. instructions,FP ops,…)
      - Architecture: related to execution on specific architectures (i.e. cache misses,…)
      - Performance: metrics reporting rates per time (i.e. MFLops, MIPS, IPC,…)
  - MPI
    - Grouped in directories displaying views and analysis. Further separated into point to point and collectives.
  - OpenMP
    - Grouped in directories displaying views and analysis

How to …

Main Paraver window

Select to browse in lower panel for traces or cfgs

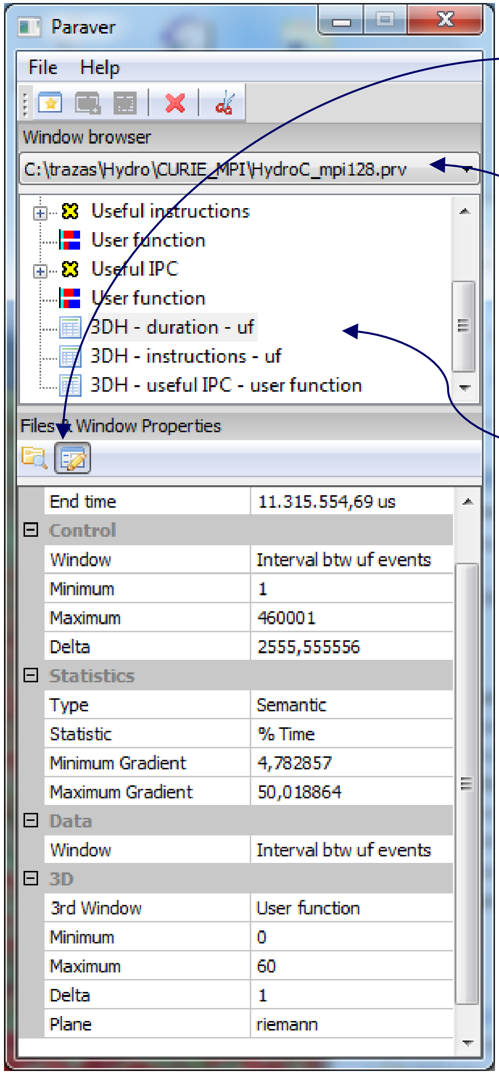


C:\trazas\Hydro\CURIE_MPI\HydroC_mpi128.prv

- Useful instructions
- User function
- Useful IPC
- User function
- 3DH - duration - uf
- 3DH - instructions - uf
- 3DH - useful IPC - user function

- HPCChallenge2011
- Hydro
  - cfgs
  - CURIE_MPI
    - HydroC_mpi128.prv
    - HydroC_mpi64.prv
  - CURIE_MPICUDA
  - mpi
  - MPI+CUDA
  - PureMPI
- Jan
- Jeremies
- JoseCarlos
- JosepMaria
- JuditPlanas
- Linpack_cluster_GPUs
- Luis
- Maja

Paraver files



C:\trazas\Hydro\CURIE_MPI\HydroC_mpi128.prv

- Useful instructions
- User function
- Useful IPC
- User function
- 3DH - duration - uf
- 3DH - instructions - uf
- 3DH - useful IPC - user function

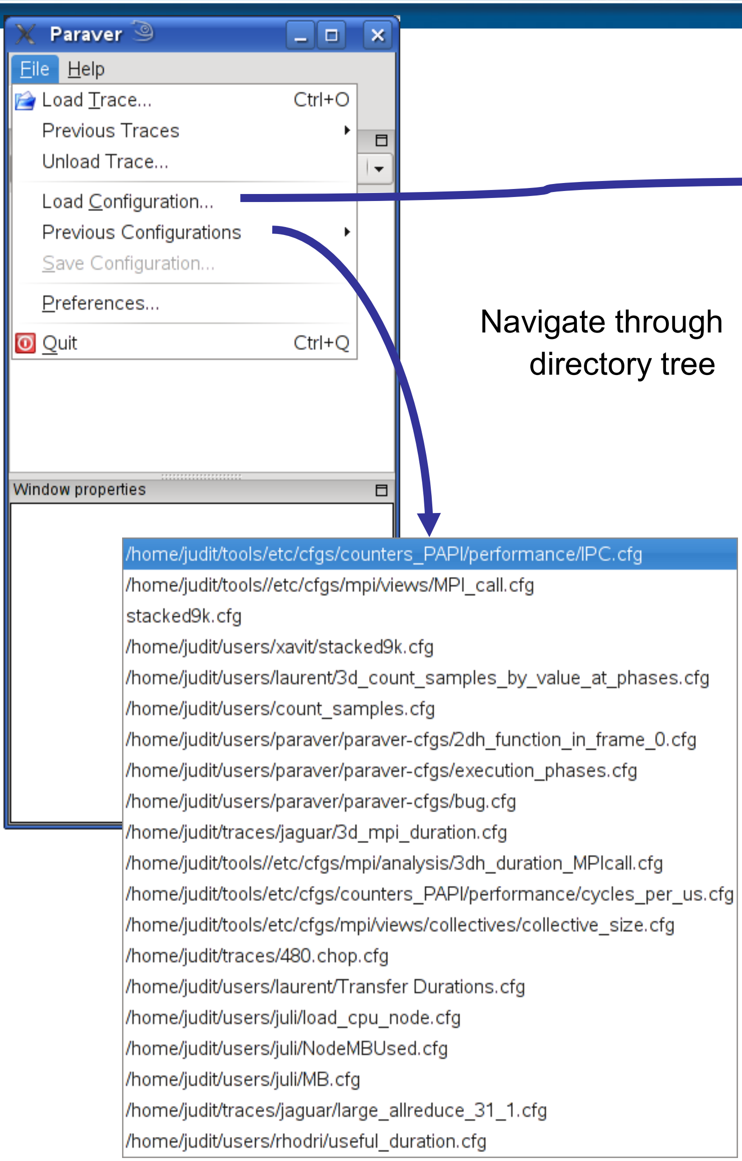
End time	11.315.554,69 us
⊟ Control	
Window	Interval btw uf events
Minimum	1
Maximum	460001
Delta	2555,555556
⊟ Statistics	
Type	Semantic
Statistic	% Time
Minimum Gradient	4,782857
Maximum Gradient	50,018864
⊟ Data	
Window	Interval btw uf events
⊟ 3D	
3rd Window	User function
Minimum	0
Maximum	60
Delta	1
Plane	riemann

Select to browse characteristics of active view or table

Active trace

Available views and tables
Active view or table highlighted

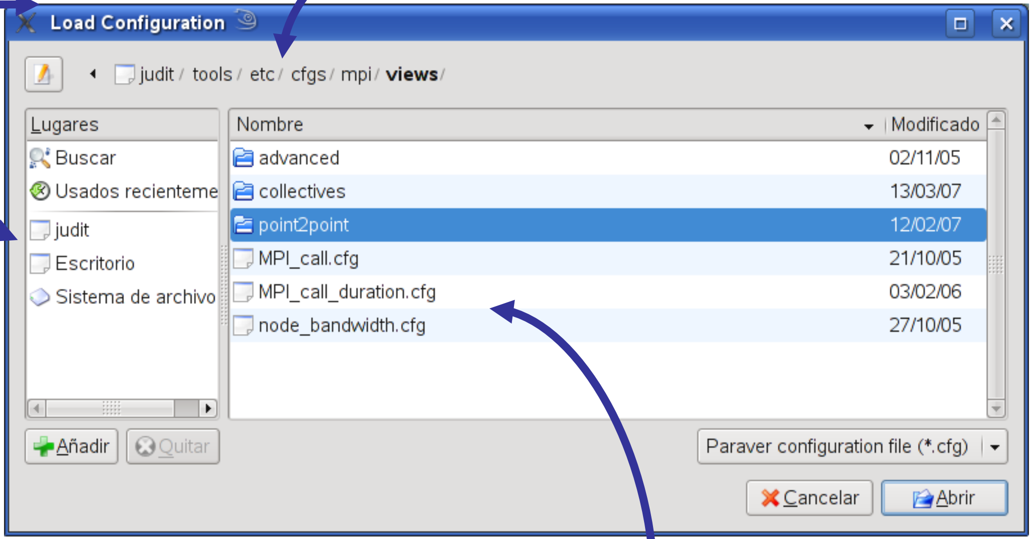
Load configuration files



/home/judit/tools/etc/cfgs/counters_PAPI/performance/IPC.cfg
/home/judit/tools//etc/cfgs/mpi/views/MPI_call.cfg
stacked9k.cfg
/home/judit/users/xavit/stacked9k.cfg
/home/judit/users/laurent/3d_count_samples_by_value_at_phases.cfg
/home/judit/users/count_samples.cfg
/home/judit/users/paraver/paraver-cfgs/2dh_function_in_frame_0.cfg
/home/judit/users/paraver/paraver-cfgs/execution_phases.cfg
/home/judit/users/paraver/paraver-cfgs/bug.cfg
/home/judit/traces/jaguar/3d_mpi_duration.cfg
/home/judit/tools//etc/cfgs/mpi/analysis/3dh_duration_MPIcall.cfg
/home/judit/tools/etc/cfgs/counters_PAPI/performance/cycles_per_us.cfg
/home/judit/tools/etc/cfgs/mpi/views/collectives/collective_size.cfg
/home/judit/traces/480.chop.cfg
/home/judit/users/laurent/Transfer Durations.cfg
/home/judit/users/juli/load_cpu_node.cfg
/home/judit/users/juli/NodeMBUsed.cfg
/home/judit/users/juli/MB.cfg
/home/judit/traces/jaguar/large_allreduce_31_1.cfg
/home/judit/users/rhodri/useful_duration.cfg

Select directory

Navigate through
directory tree

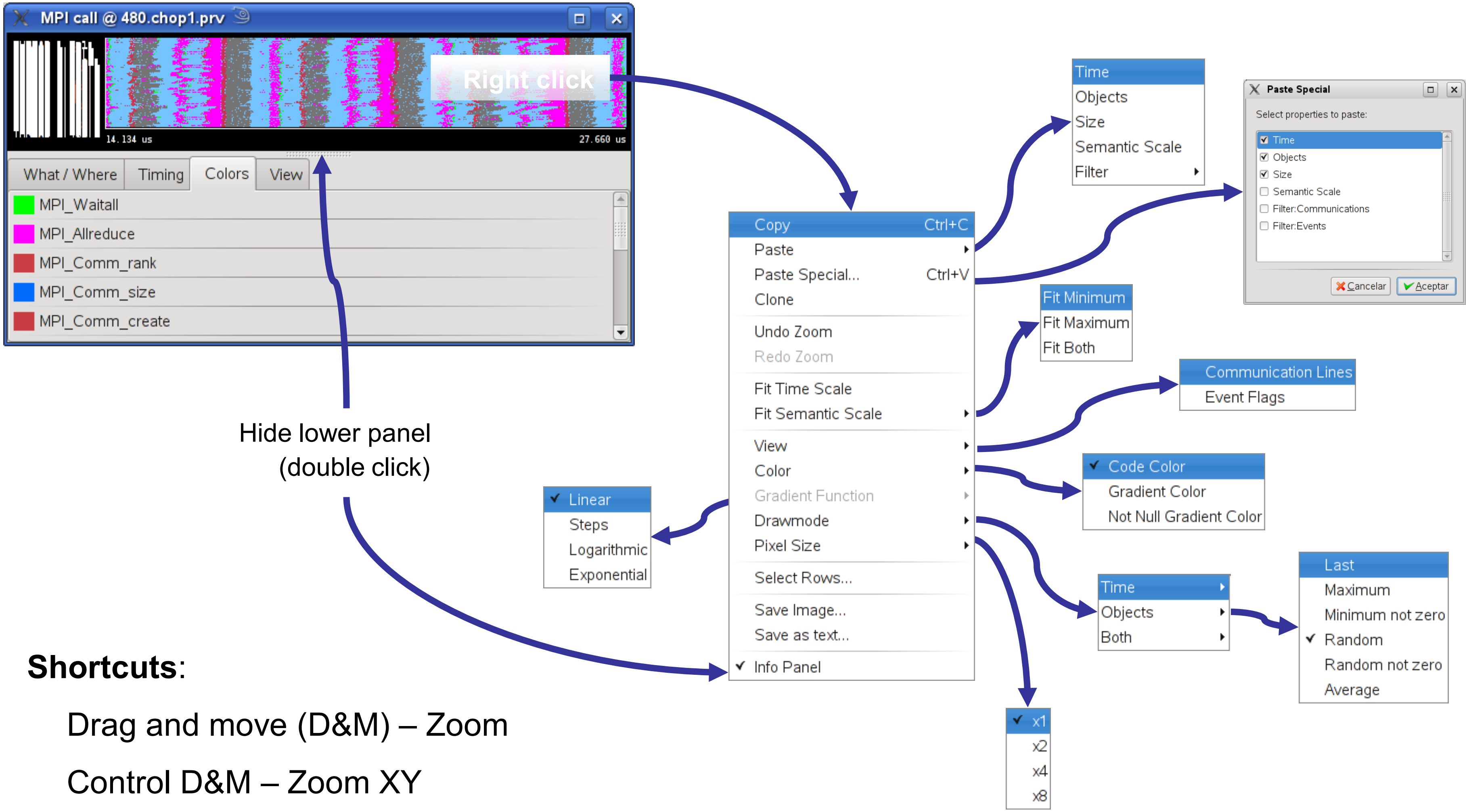


Nombre	Modificado
advanced	02/11/05
collectives	13/03/07
point2point	12/02/07
MPI_call.cfg	21/10/05
MPI_call_duration.cfg	03/02/06
node_bandwidth.cfg	27/10/05

List of directories and configuration
files in current directory

APPLIED TO THE CURRENT TRACEFILE

Navigation



Hide lower panel
(double click)

Shortcuts:

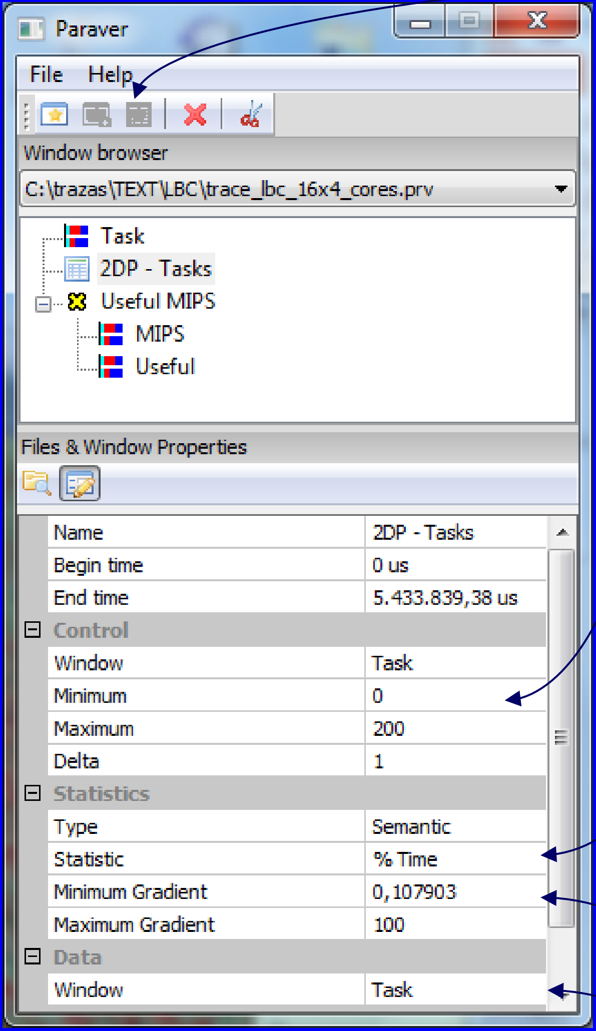
Drag and move (D&M) – Zoom

Control D&M – Zoom XY

Shift D&M – Timing

How to generate table and change statistic

To generate table: click button and select region of the window whose values will determine the columns of the table

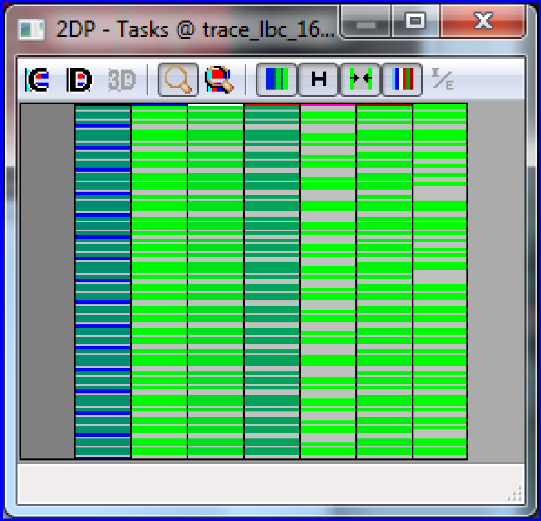


C:\trazas\TEXT\LBC\trace_lbc_16x4_cores.prv

- Task
- 2DP - Tasks
- Useful MIPS
  - MIPS
  - Useful

Name	2DP - Tasks
Begin time	0 us
End time	5.433.839,38 us
Control	
Window	Task
Minimum	0
Maximum	200
Delta	1
Statistics	
Type	Semantic
Statistic	% Time
Minimum Gradient	0,107903
Maximum Gradient	100
Data	
Window	Task

Range and bin width (delta) represented by each column. By default is automatically selected, but can be manually changed



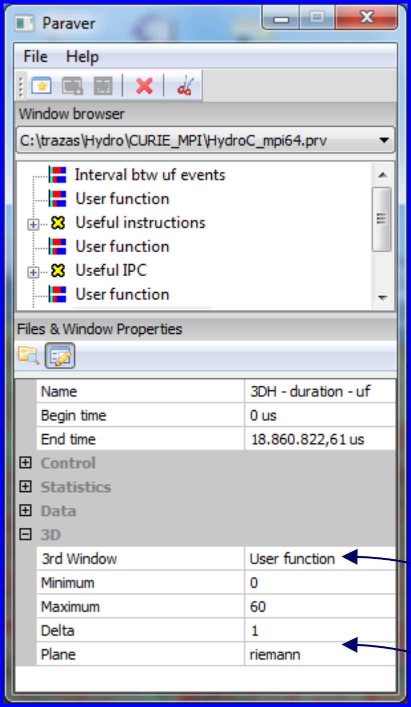
Selection of statistic to appear in each cell

Cell coloring gradient control

Window used to compute statistic (only used by some statistics)

3D tables

- One additional dimension
  - One plane per value of a 3D control window
- Useful to categorize histograms
  - i.e. histogram of duration of specific user function



Paraver

File Help

Window browser

C:\trazas\Hydro\CURIE_MPI\HydroC_mpi64.prv

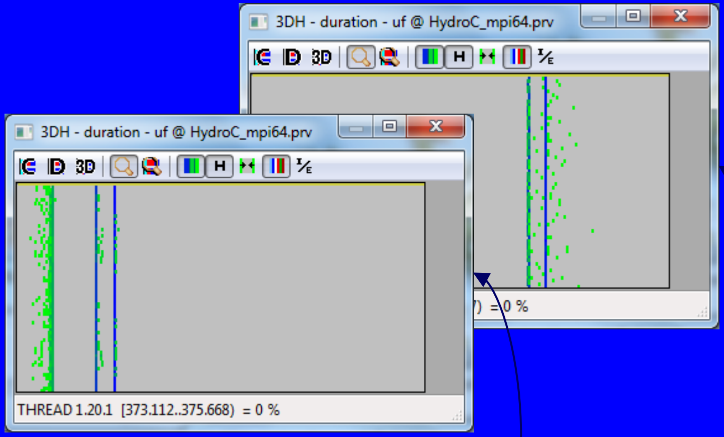
- Interval btw uf events
- User function
- Useful instructions
- User function
- Useful IPC
- User function

Files & Window Properties

Name	3DH - duration - uf
Begin time	0 us
End time	18.860.822,61 us
⊞ Control	
⊞ Statistics	
⊞ Data	
⊟ 3D	
3rd Window	User function
Minimum	0
Maximum	60
Delta	1
Plane	riemann

3D control window: determines planes

Actual Plane on display



3DH - duration - uf @ HydroC_mpi64.prv

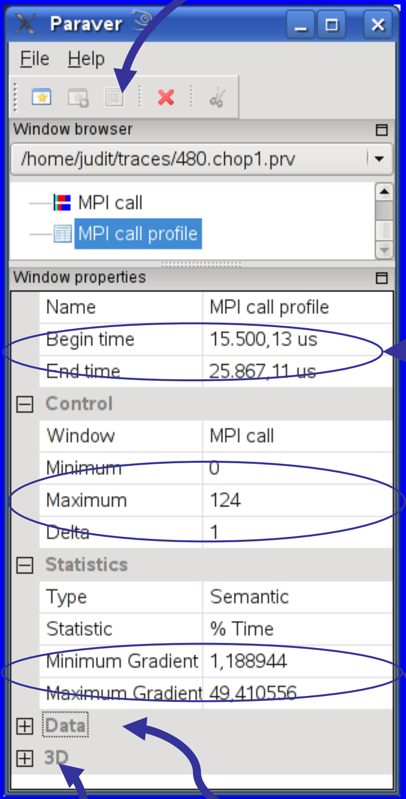
) = 0 %

3DH - duration - uf @ HydroC_mpi64.prv

THREAD 1.20.1 [373.112..375.668) = 0 %

Table information and control

Create a new table



Name	MPI call profile
Begin time	15.500,13 us
End time	25.867,11 us
Control	
Window	MPI call
Minimum	0
Maximum	124
Delta	1
Statistics	
Type	Semantic
Statistic	% Time
Minimum Gradient	1,188944
Maximum Gradient	49,410556
Data	
3D	

Region analyzed

Bin definition

Change Data window

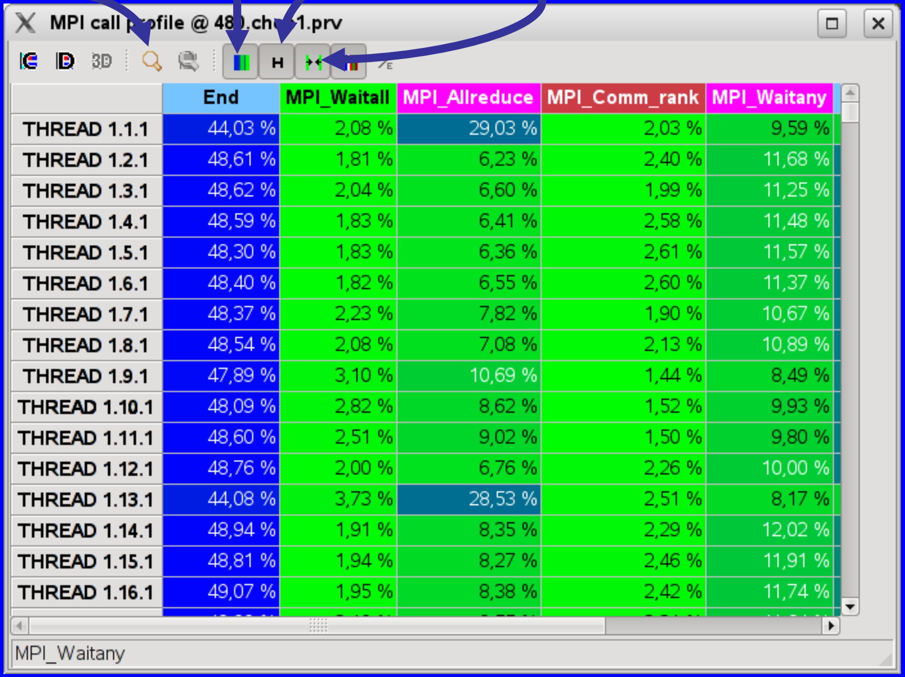
Activate 3D analysis

Display whole table / cell text

Color/not cells

Transpose

Hide null columns



	End	MPI_Waitall	MPI_Allreduce	MPI_Comm_rank	MPI_Waitany
THREAD 1.1.1	44,03 %	2,08 %	29,03 %	2,03 %	9,59 %
THREAD 1.2.1	48,61 %	1,81 %	6,23 %	2,40 %	11,68 %
THREAD 1.3.1	48,62 %	2,04 %	6,60 %	1,99 %	11,25 %
THREAD 1.4.1	48,59 %	1,83 %	6,41 %	2,58 %	11,48 %
THREAD 1.5.1	48,30 %	1,83 %	6,36 %	2,61 %	11,57 %
THREAD 1.6.1	48,40 %	1,82 %	6,55 %	2,60 %	11,37 %
THREAD 1.7.1	48,37 %	2,23 %	7,82 %	1,90 %	10,67 %
THREAD 1.8.1	48,54 %	2,08 %	7,08 %	2,13 %	10,89 %
THREAD 1.9.1	47,89 %	3,10 %	10,69 %	1,44 %	8,49 %
THREAD 1.10.1	48,09 %	2,82 %	8,62 %	1,52 %	9,93 %
THREAD 1.11.1	48,60 %	2,51 %	9,02 %	1,50 %	9,80 %
THREAD 1.12.1	48,76 %	2,00 %	6,76 %	2,26 %	10,00 %
THREAD 1.13.1	44,08 %	3,73 %	28,53 %	2,51 %	8,17 %
THREAD 1.14.1	48,94 %	1,91 %	8,35 %	2,29 %	12,02 %
THREAD 1.15.1	48,81 %	1,94 %	8,27 %	2,46 %	11,91 %
THREAD 1.16.1	49,07 %	1,95 %	8,38 %	2,42 %	11,74 %

Color encoding

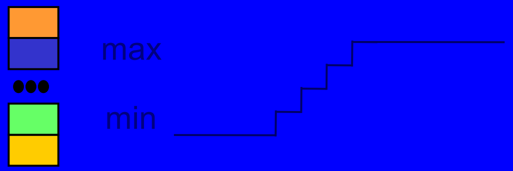


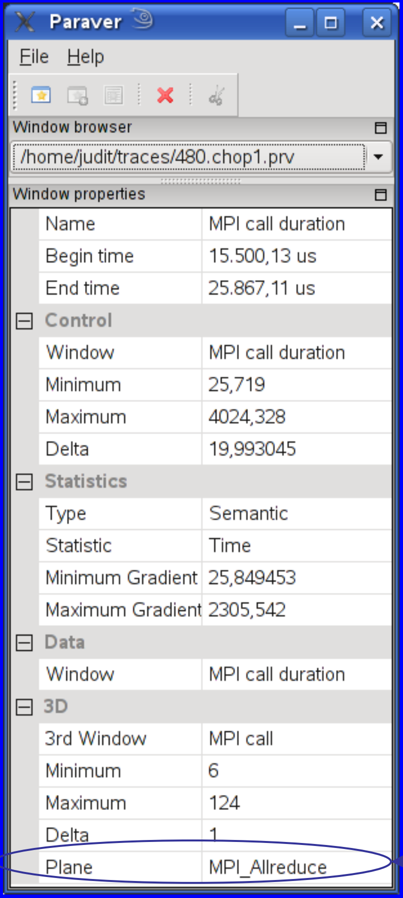
Table information and control

Open Data window

Open Control window

Open 3D window

Generate a timeline, derived form control window with the range of values selected clicking in the table (zoom mode only)



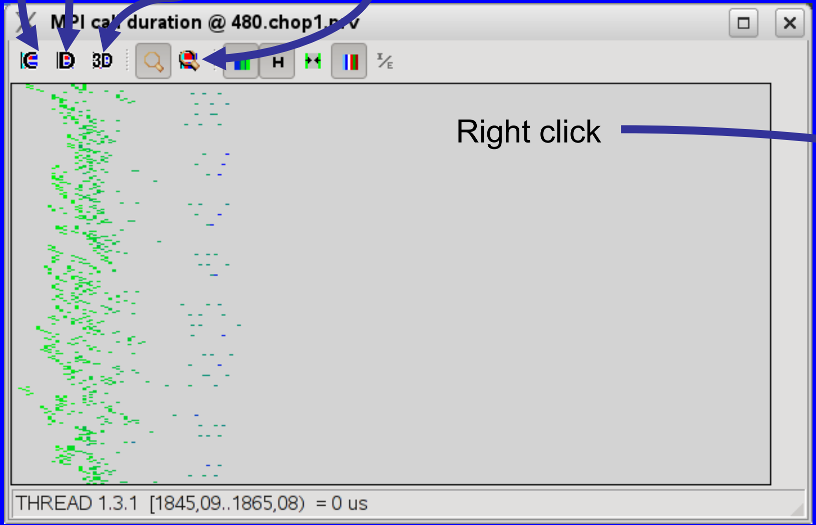
Window browser

/home/judit/traces/480.chop1.prv

Window properties

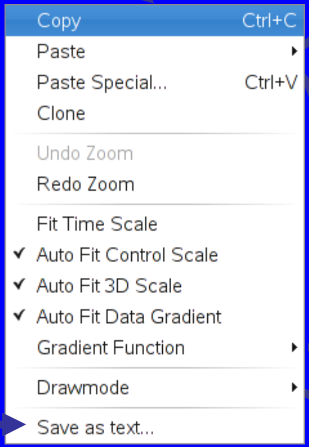
Name	MPI call duration
Begin time	15.500,13 us
End time	25.867,11 us
Control	
Window	MPI call duration
Minimum	25,719
Maximum	4024,328
Delta	19,993045
Statistics	
Type	Semantic
Statistic	Time
Minimum Gradient	25,849453
Maximum Gradient	2305,542
Data	
Window	MPI call duration
3D	
3rd Window	MPI call
Minimum	6
Maximum	124
Delta	1
Plane	MPI_Allreduce

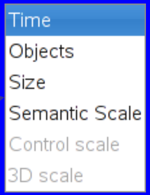
Selected plane

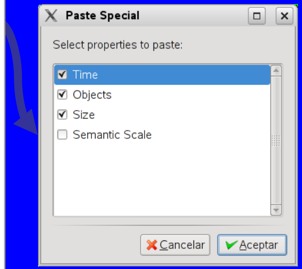


THREAD 1.3.1 [1845,09..1865,08) = 0 us

Right click

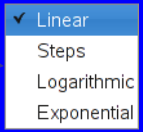




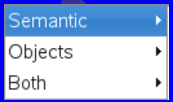


Select properties to paste:

☑ Time
☑ Objects
☑ Size
☐ Semantic Scale



Generate ASCII file with table data



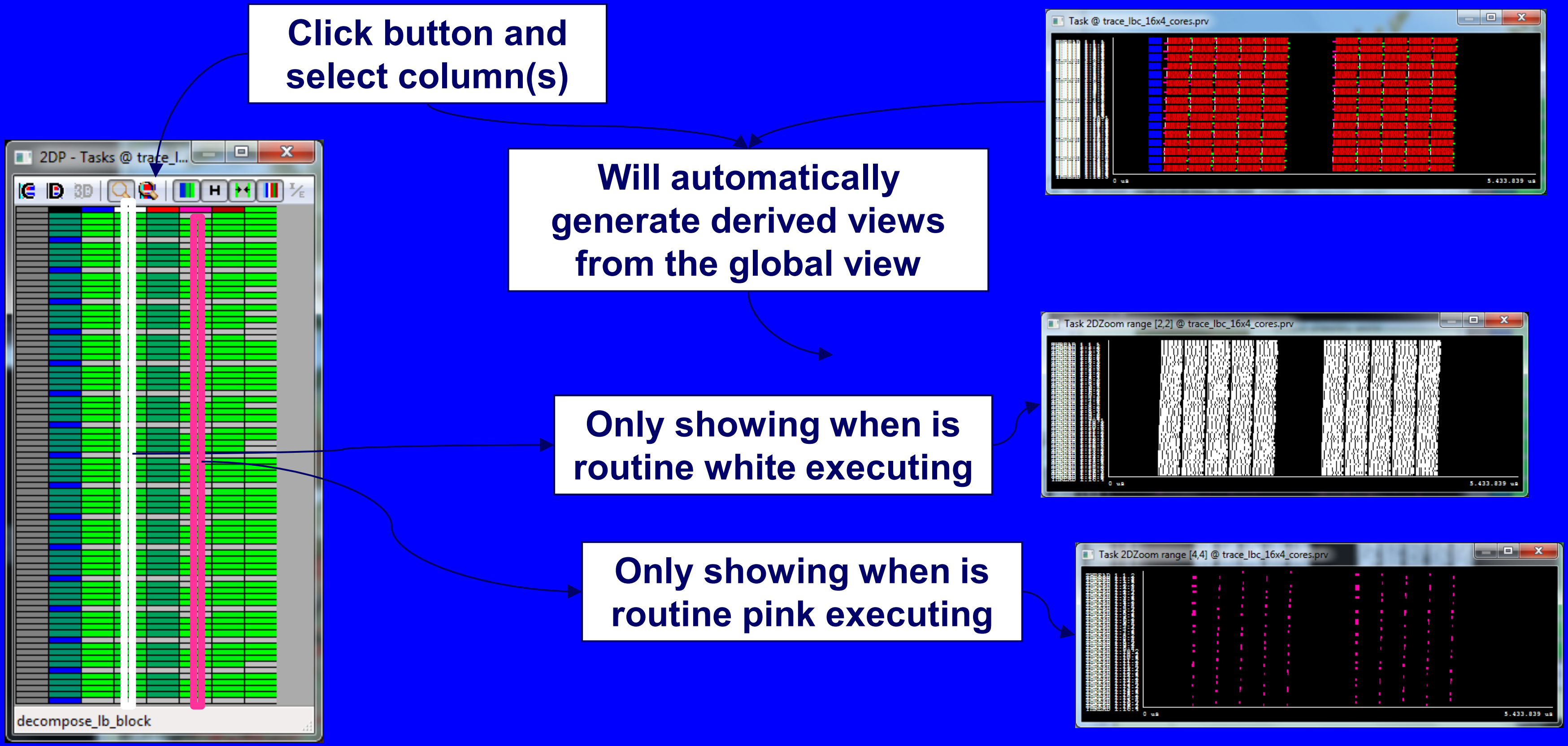
Shortcuts (zoom mode only):

Drag and move (D&M) – Zoom

Control D&M – Zoom XY

From tables to timelines

- Where in the timeline do the values in certain table columns appear?
  - ie. want to see the time distribution of a given routine?



2DP - Tasks @ trace_l...
decompose_lb_block
Task @ trace_lbc_16x4_cores.prv
Task 2DZoom range [2,2] @ trace_lbc_16x4_cores.prv
Task 2DZoom range [4,4] @ trace_lbc_16x4_cores.prv
0 us
5.433.839 us

Click button and select column(s)

Will automatically generate derived views from the global view

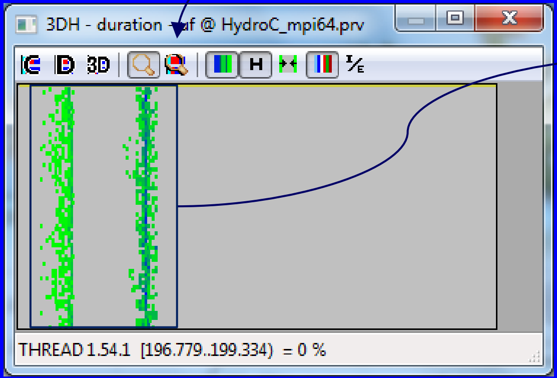
Only showing when is routine white executing

Only showing when is routine pink executing

From tables to timelines

- Where in the timeline do the values in certain table columns appear?
  - ie. want to see where the timeline happen computation bursts of a given length?

Click button and select column(s)

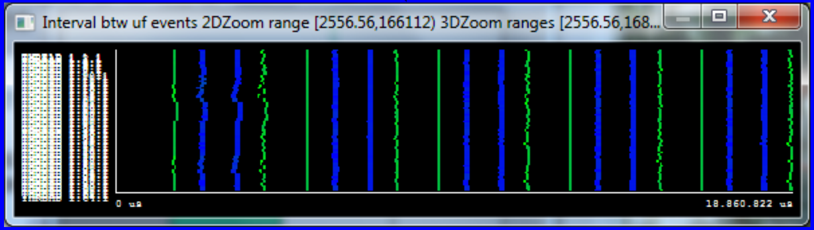


3DH - duration - uf @ HydroC_mpi64.prv

THREAD 1.54.1 [196.779..199.334) = 0 %

3D histogram of duration of routine foo

Will automatically generate



Interval btw uf events 2DZoom range [2556.56,166112) 3DZoom ranges [2556.56,168...

Only showing duration of routine foo

Trace manipulation

Handling very large traces

- Paraver data handling utilities
  - If trying to load a very large trace, Paraver will ask if you want to filter it

- Three steps:
  - Filter original trace discarding most of the records only keeping most relevant information (typically computation bursts longer than a given lower bound)
  - Analyze coarse grain structure of trace. Typically useful_duration.cfg
  - Cut original trace to obtain a fully detailed trace for the time interval considered representative or of interest

Guided hands-on available in
http://www.bsc.es/computer-sciences/performance-tools/documentation → Trace Preparation

Filtering very large traces

Trace to which it will be applied

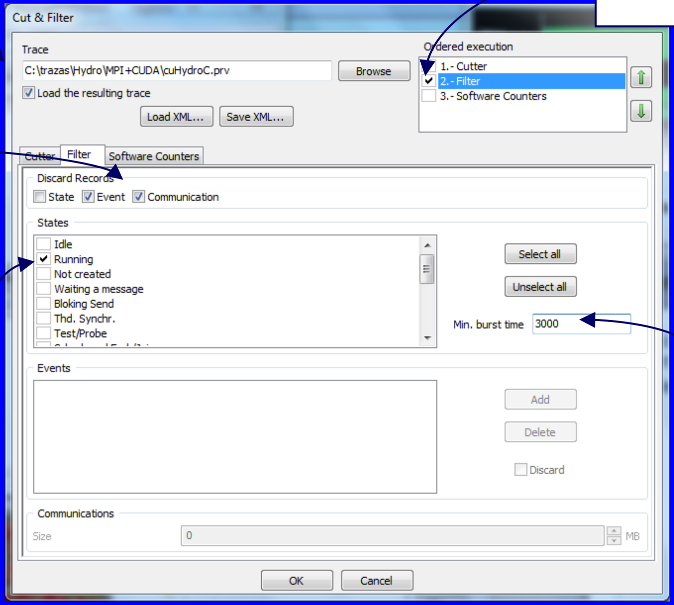
A trace with basename.filter1.prv will be generated

Select filtering option

Discard events and communications

Keep only Running bursts ….

--- longer than 3000 ns



Cut & Filter

Trace

C:\trazas\Hydro\MPI+CUDA\cuHydroC.prv

☑ Load the resulting trace

Ordered execution

- □ 1.- Cutter
- ☑ 2.- Filter
- □ 3.- Software Counters

Discard Records

□ State ☑ Event ☑ Communication

States

- □ Idle
- ☑ Running
- □ Not created
- □ Waiting a message
- □ Bloking Send
- □ Thd. Synchr.
- □ Test/Probe

Min. burst time 3000

Events

□ Discard

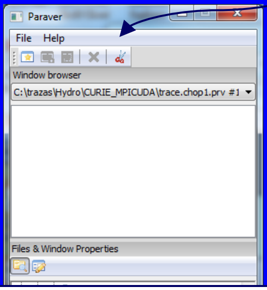
Communications

Size 0 MB

Cutting very large traces

- Load a filtered trace and use the scissors tool

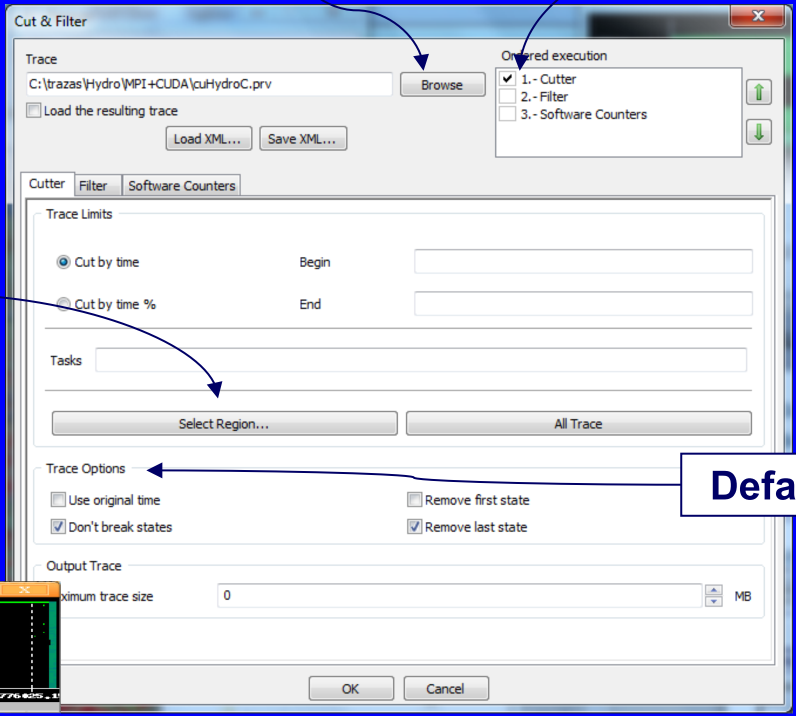
Scissors tool



C:\trazas\Hydro\CURIE_MPICUDA\trace.chop1.prv #1

Browse to select file from which the cut will be obtained

Select cutter



Trace

C:\trazas\Hydro\MPI+CUDA\cuHydroC.prv

□ Load the resulting trace

Ordered execution

☑ 1.- Cutter
□ 2.- Filter
□ 3.- Software Counters

Trace Limits

● Cut by time Begin

○ Cut by time % End

Tasks

Trace Options

□ Use original time □ Remove first state

☑ Don't break states ☑ Remove last state

Output Trace

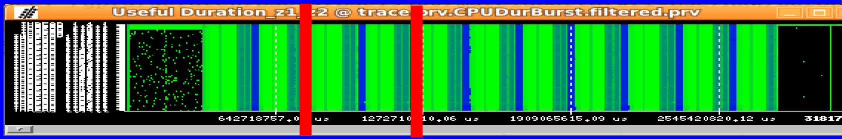
ximum trace size 0 MB

Click to select region

Select time interval by clicking left and right limits in a window of the filtered trace previously loaded

Recommended cuts within long computation bursts

Default setups



Extrae

Adapt job submission script

appl.job

```
#!/bin/bash

export NP=8
export INPUT=$1

cleo-submit -np $NP            ./HydroC -i $INPUT
```

Adapt job submission script

appl.job

```
#!/bin/bash

export NP=8
export INPUT=$1

cleo-submit -np $NP ./trace.sh ./HydroC -i $INPUT
```

trace.sh

```
#!/bin/bash

export EXTRAE_HOME=/export/hopsa/BSCtools/tools/extrae-2.3
export EXTRAE_CONFIG_FILE=extrae/extrae.xml

export LD_PRELOAD=$EXTRAE_HOME/lib/libmpitrace.so

export EXE=$1
export TRACENAME=${EXE}_$3.prv

$@
```

Trace control .xml

extrae.xml

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

  <trace enabled="yes“
    home="/home/judit/tools/extrae-2.3"
    initial-mode="detail"
    type="paraver"
    xml-parser-id="Id: xml-parse.c 799 2011-10-20 16:02:03Z harald $"
  >

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

  <openmp enabled="no">
    <locks enabled="no" />
    <counters enabled="yes" />
  </openmp>

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

…
```

Activate MPI tracing and emit hardware counters at MPI calls

Do not activate OpenMP tracing

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

Details in $EXTRAE_HOME/share/example/MPI/extrae_explained.xml

Trace control .xml (cont)

extrae.xml (cont)

```
<user-functions enabled="no" list="/home/bsc41/bsc41273/user-functions.dat">
  <max-depth enabled="no">3</max-depth>
  <counters enabled="yes" />
</user-functions>
```

…

Add instrumentation at specified user functions

Requires Dyninst based mpitrace

Trace control .xml (cont)

extrae.xml (cont)

```
…

  <counters enabled="yes">

    <cpu enabled="yes" starting-set-distribution="1">
      <set enabled="yes" domain="all" changeat-globalops="5">
        PAPI_TOT_INS,PAPI_TOT_CYC,PAPI_L2_DCM
        <sampling enabled="no" frequency="100000000">PAPI_TOT_CYC
      </set>
      <set enabled="yes" domain="user" changeat-globalops="5">
        PAPI_TOT_INS,PAPI_FP_INS,PAPI_TOT_CYC
      </set>
    </cpu>

    <network enabled=“no" />

    <resource-usage enabled=“no" />

    <memory-usage enabled="no" />

  </counters>

…
```

Emit counters or not

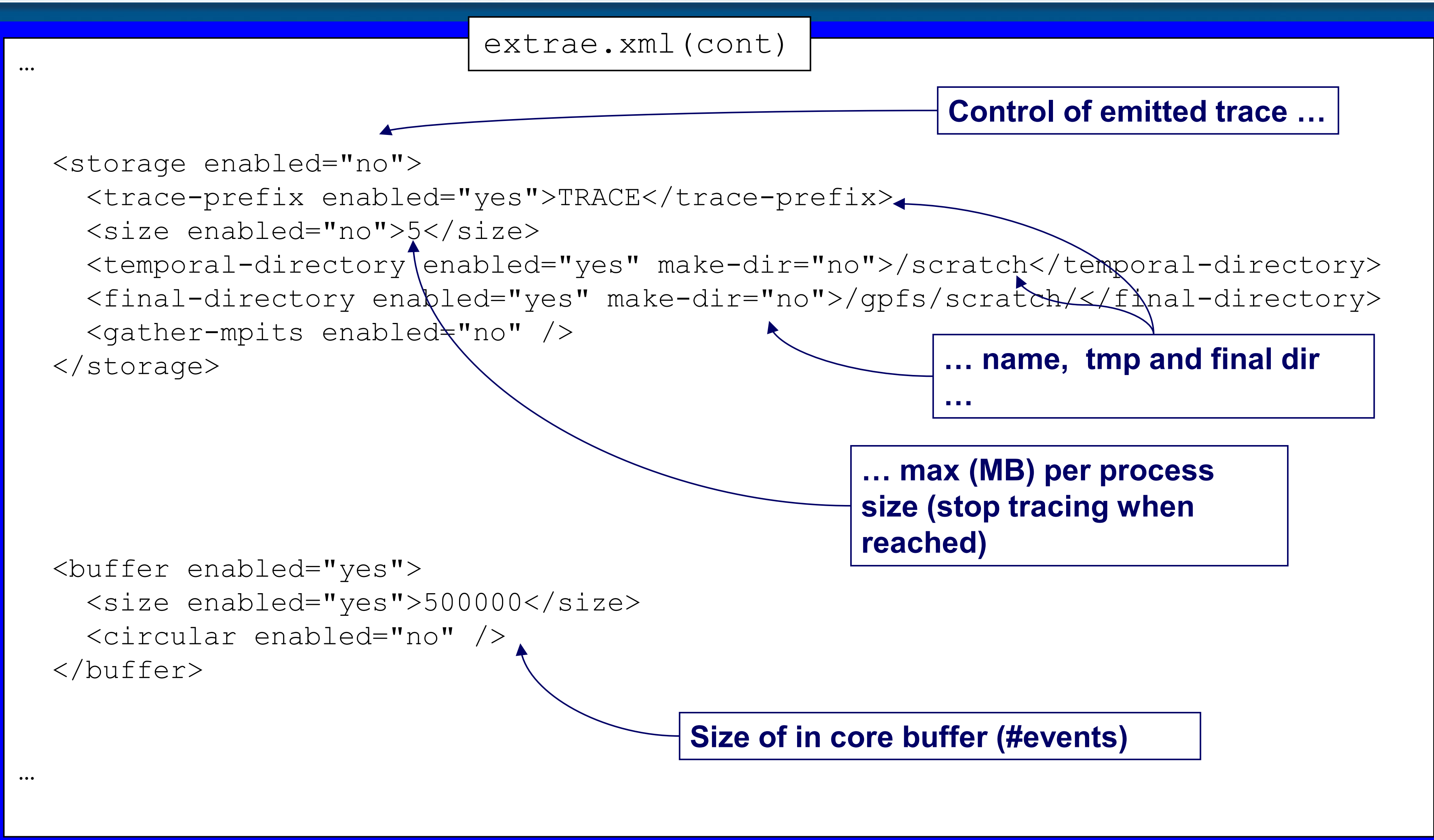
When to rotate between groups

Groups

Interconnection network counters
Just at end of trace because of large acquisition overhead

OS info (context switches,….)

Trace control .xml (cont)



extrae.xml (cont)

```
…

  <storage enabled="no">
    <trace-prefix enabled="yes">TRACE</trace-prefix>
    <size enabled="no">5</size>
    <temporal-directory enabled="yes" make-dir="no">/scratch</temporal-directory>
    <final-directory enabled="yes" make-dir="no">/gpfs/scratch/</final-directory>
    <gather-mpits enabled="no" />
  </storage>

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

…
```

Control of emitted trace …

… name, tmp and final dir …

… max (MB) per process size (stop tracing when reached)

Size of in core buffer (#events)

Trace control .xml (cont)

extrae.xml (cont)

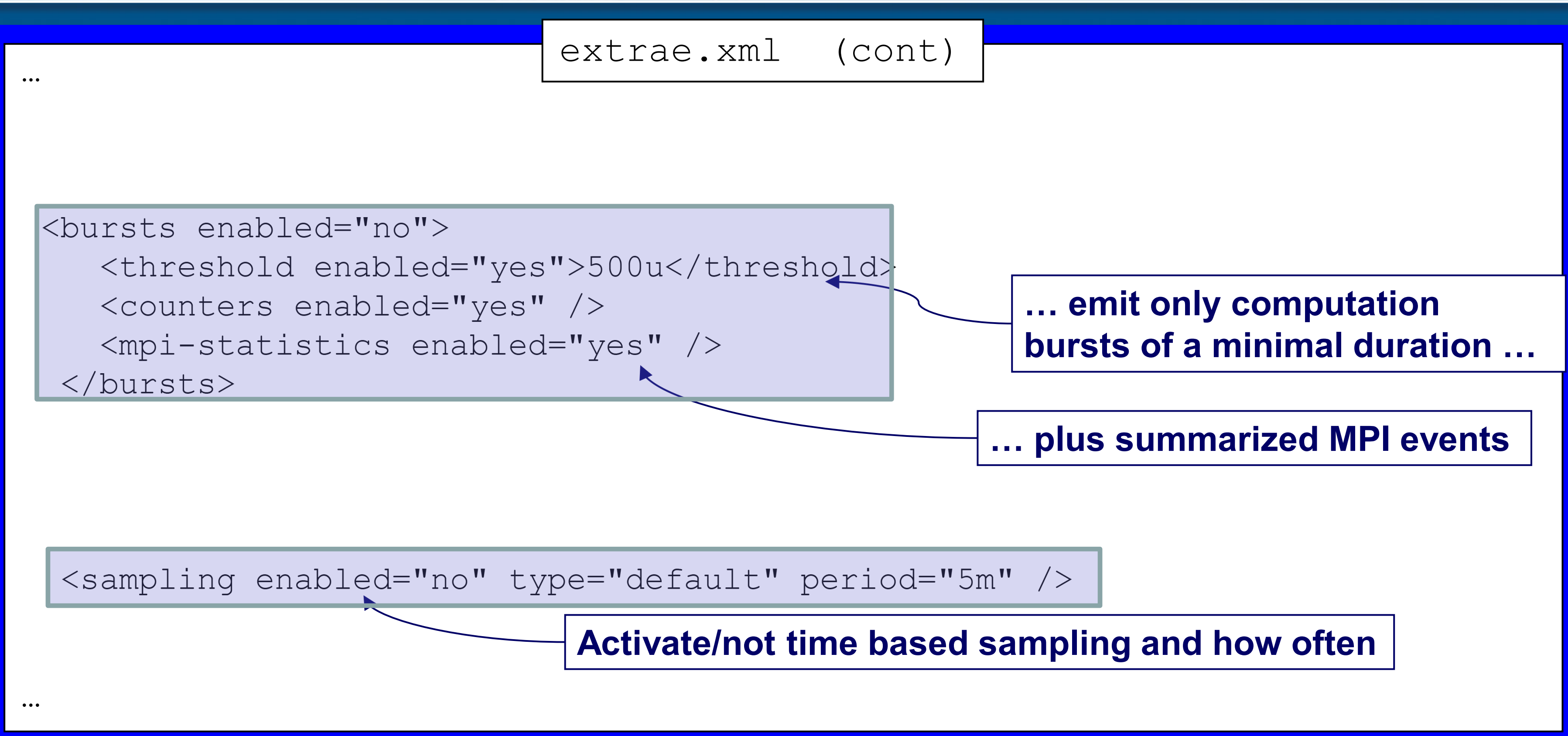
```
…

  <trace-control enabled="yes">
    <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=“no">
    <minimum-time enabled="no">10M</minimum-time>
    <terminate-on-signal enabled="no">USR2</terminate-on-signal>
  </others>

…
```

External activation of tracing (creation of file will start tracing)

Stop tracing after elapsed time …

… or when signal received

Trace control .xml (cont)



extrae.xml (cont)

…

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

… emit only computation bursts of a minimal duration …

… plus summarized MPI events

```
<sampling enabled="no" type="default" period="5m" />
```

Activate/not time based sampling and how often

…

Trace control .xml (cont)

extrae.xml (cont)

```
…

  <merge enabled="yes"
    synchronization="default"
    binary="$EXE$"
    tree-fan-out="16"
    max-memory="512"
    joint-states="yes"
    keep-mpits="yes"
    sort-addresses="yes"
  >
    $TRACENAME$
  </merge>


</trace>
```

Merge individual traces into global application trace at end of run …

… into this trace name

LD_PRELOAD library selection

- Library depends on programming model

Programming model	Library
Serial	libseqtrace
Pure MPI	libmpitrace[f][1]
Pure OpenMP	libomptrace
Pure Pthreads	libpttrace
CUDA	libcudatrace
MPI + OpenMP	libompitrace[f] [1]
MPI + Pthreads	libptmpitrace[f] [1]
Mpi + CUDA	libcudampitrace[f] [1]

[1] for Fortran codes

Using Dimemas

Dimemas trace generation

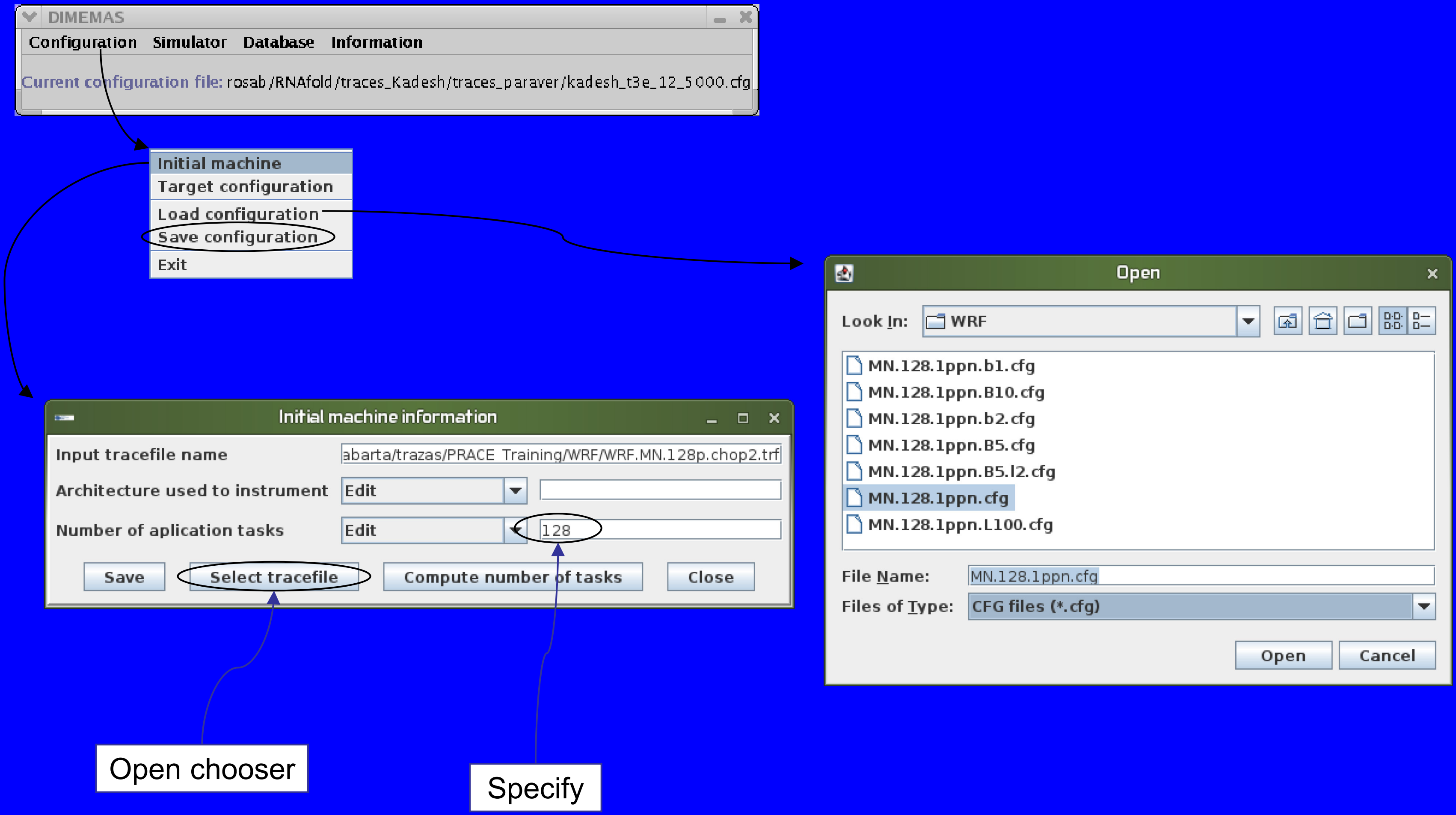
- Paraver → Dimemas trace Generation
  - prv2dim original.prv dimemas.dim
  - Default: duration of each computation region taken from .prv computation duration

```
Usage:
  prv2dim -i <iprobe_miss_threshold> -b <hw_counter_type>,<factor>
  <paraver_trace> <dimemas_trace>

  -h                              This help
  -n                              No generate initial idle states
  -i <iprobe_miss_threshold>      Maximun MPI_Iprobe misses to discard Iprobe
                                  area CPU burst
  -b <hw_counter_type>,<factor>   Hardware counter type and factor used to
                                  generate burst durations
…
```

Force synchronized start of all threads

Computation region duration derived from hardware counters assuming/modeling a given performance (<factor>)
Ie. estimate impact if we get rid of IPC imbalances

Dimemas GUI – Specify trace to simulate



Current configuration file: rosab/RNAfold/traces_Kadesh/traces_paraver/kadesh_t3e_12_5000.cfg

Input tracefile name: abarta/trazas/PRACE_Training/WRF/WRF.MN.128p.chop2.trf

Architecture used to instrument: Edit

Number of aplication tasks: Edit 128

Look In: WRF

- MN.128.1ppn.b1.cfg
- MN.128.1ppn.B10.cfg
- MN.128.1ppn.b2.cfg
- MN.128.1ppn.B5.cfg
- MN.128.1ppn.B5.l2.cfg
- MN.128.1ppn.cfg
- MN.128.1ppn.L100.cfg

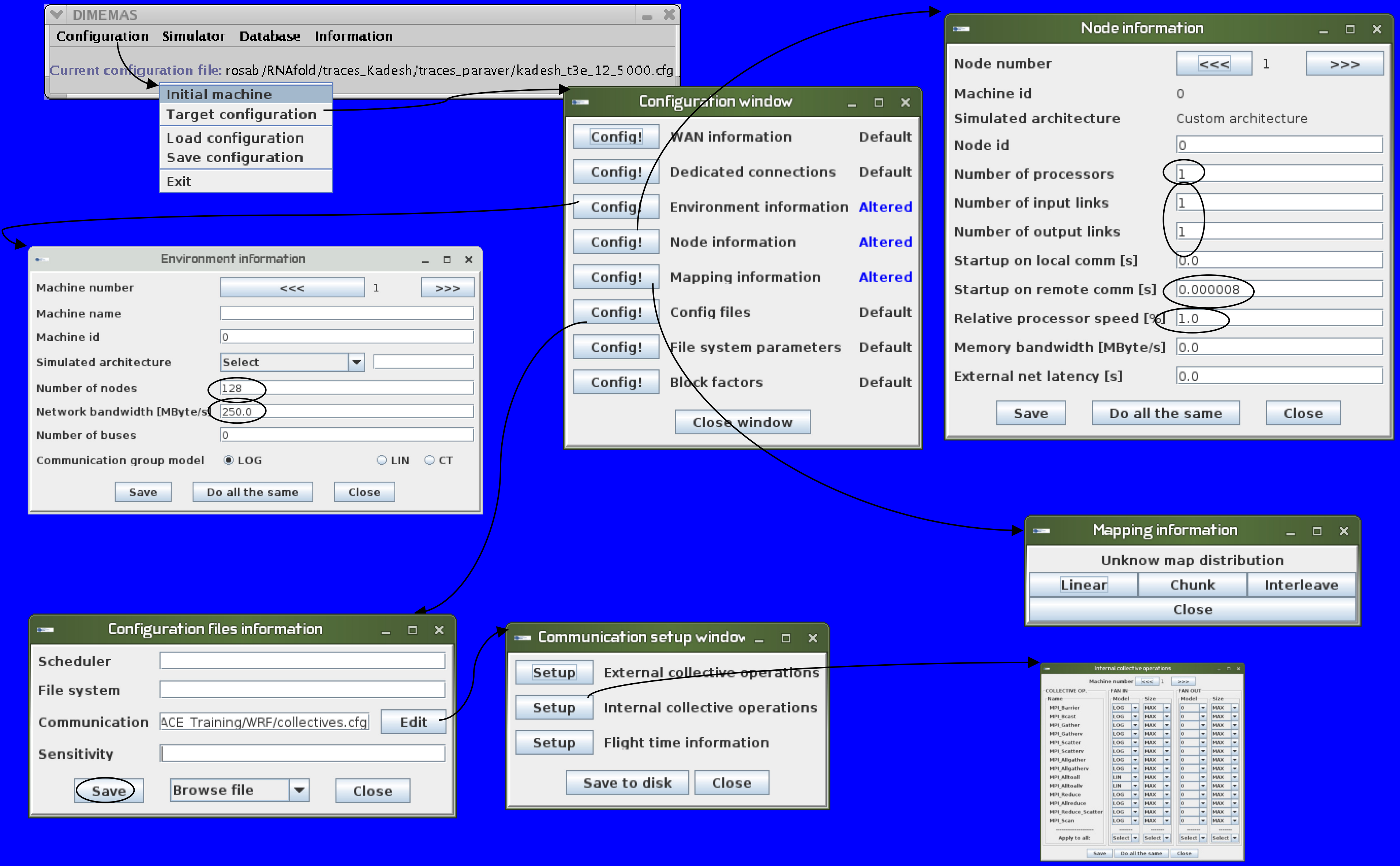
File Name: MN.128.1ppn.cfg

Files of Type: CFG files (*.cfg)

Open chooser

Specify

Dimemas GUI – Specify target machine



Current configuration file: rosab/RNAfold/traces_Kadesh/traces_paraver/kadesh_t3e_12_5000.cfg

Configuration window

Config!	WAN information	Default
Config!	Dedicated connections	Default
Config!	Environment information	Altered
Config!	Node information	Altered
Config!	Mapping information	Altered
Config!	Config files	Default
Config!	File system parameters	Default
Config!	Block factors	Default

Node information

Node number	1
Machine id	0
Simulated architecture	Custom architecture
Node id	0
Number of processors	1
Number of input links	1
Number of output links	1
Startup on local comm [s]	0.0
Startup on remote comm [s]	0.000008
Relative processor speed [%]	1.0
Memory bandwidth [MByte/s]	0.0
External net latency [s]	0.0

Environment information

Machine number	1
Machine name	
Machine id	0
Simulated architecture	Select
Number of nodes	128
Network bandwidth [MByte/s]	250.0
Number of buses	0
Communication group model	● LOG ○ LIN ○ CT

Mapping information

Unknow map distribution

Configuration files information

Scheduler	
File system	
Communication	ACE_Training/WRF/collectives.cfg
Sensitivity	

Communication setup window

- Setup External collective operations
- Setup Internal collective operations
- Setup Flight time information

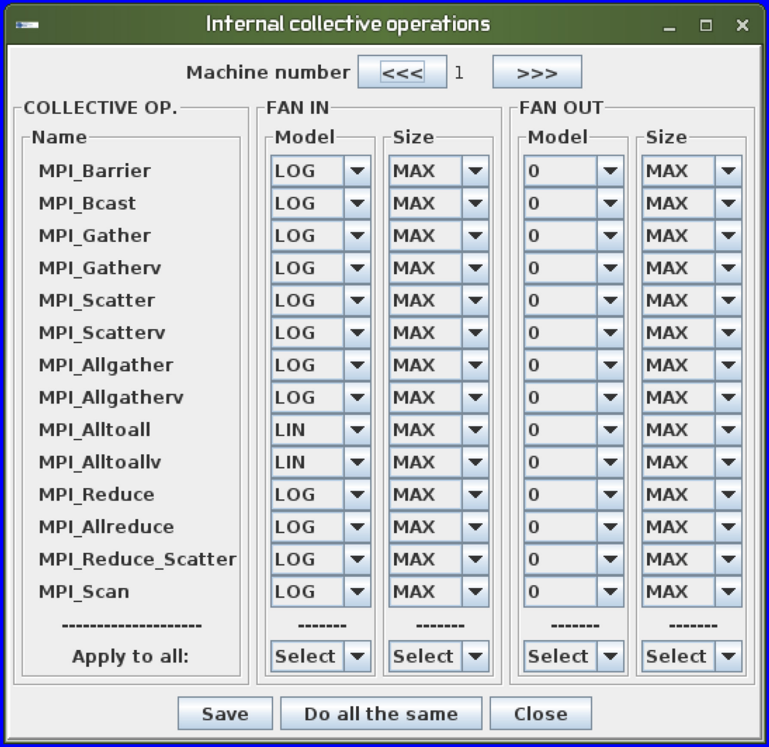
Internal collective operations

Machine number 1

COLLECTIVE OP. Name	FAN IN Model	FAN IN Size	FAN OUT Model	FAN OUT Size
MPI_Barrier	LOG	MAX	0	MAX
MPI_Bcast	LOG	MAX	0	MAX
MPI_Gather	LOG	MAX	0	MAX
MPI_Gatherv	LOG	MAX	0	MAX
MPI_Scatter	LOG	MAX	0	MAX
MPI_Scatterv	LOG	MAX	0	MAX
MPI_Allgather	LOG	MAX	0	MAX
MPI_Allgatherv	LOG	MAX	0	MAX
MPI_Alltoall	LIN	MAX	0	MAX
MPI_Alltoallv	LIN	MAX	0	MAX
MPI_Reduce	LOG	MAX	0	MAX
MPI_Allreduce	LOG	MAX	0	MAX
MPI_Reduce_Scatter	LOG	MAX	0	MAX
MPI_Scan	LOG	MAX	0	MAX
--------------------	--------	--------	--------	--------
Apply to all:	Select	Select	Select	Select

Collective Communication Model

- Per call model
  - Model factor
    - Lin
    - Log
    - Const
  - Size of message
    - Min over all processes
    - Mean over all processes
    - Max over all processes
  - Specified in input file



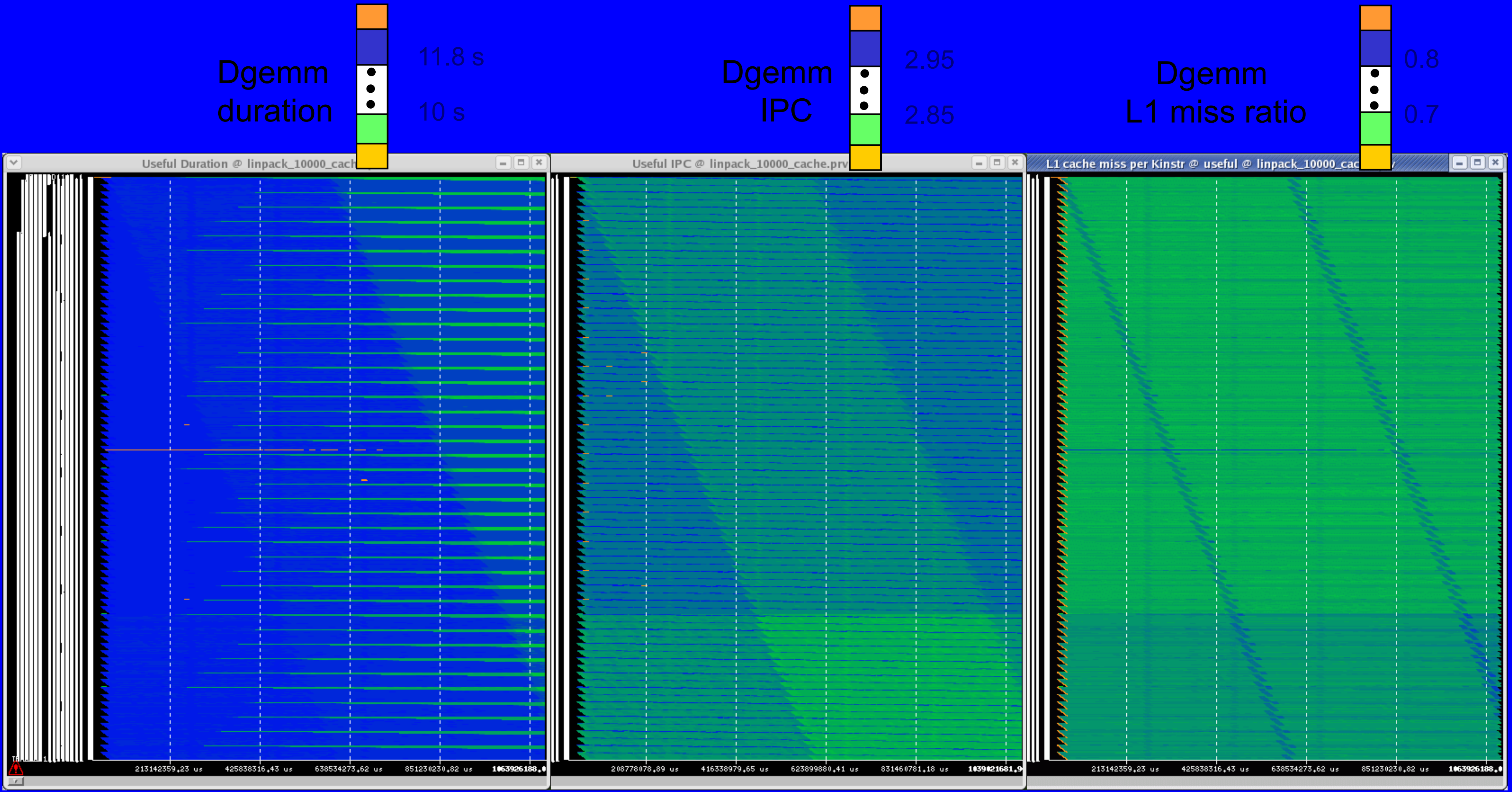
Machine number 1

COLLECTIVE OP.	FAN IN		FAN OUT	
Name	Model	Size	Model	Size
MPI_Barrier	LOG	MAX	0	MAX
MPI_Bcast	LOG	MAX	0	MAX
MPI_Gather	LOG	MAX	0	MAX
MPI_Gatherv	LOG	MAX	0	MAX
MPI_Scatter	LOG	MAX	0	MAX
MPI_Scatterv	LOG	MAX	0	MAX
MPI_Allgather	LOG	MAX	0	MAX
MPI_Allgatherv	LOG	MAX	0	MAX
MPI_Alltoall	LIN	MAX	0	MAX
MPI_Alltoallv	LIN	MAX	0	MAX
MPI_Reduce	LOG	MAX	0	MAX
MPI_Allreduce	LOG	MAX	0	MAX
MPI_Reduce_Scatter	LOG	MAX	0	MAX
MPI_Scan	LOG	MAX	0	MAX
--------------------	-------	-------	-------	-------
Apply to all:	Select	Select	Select	Select

Scalability

Scalability of Presentation

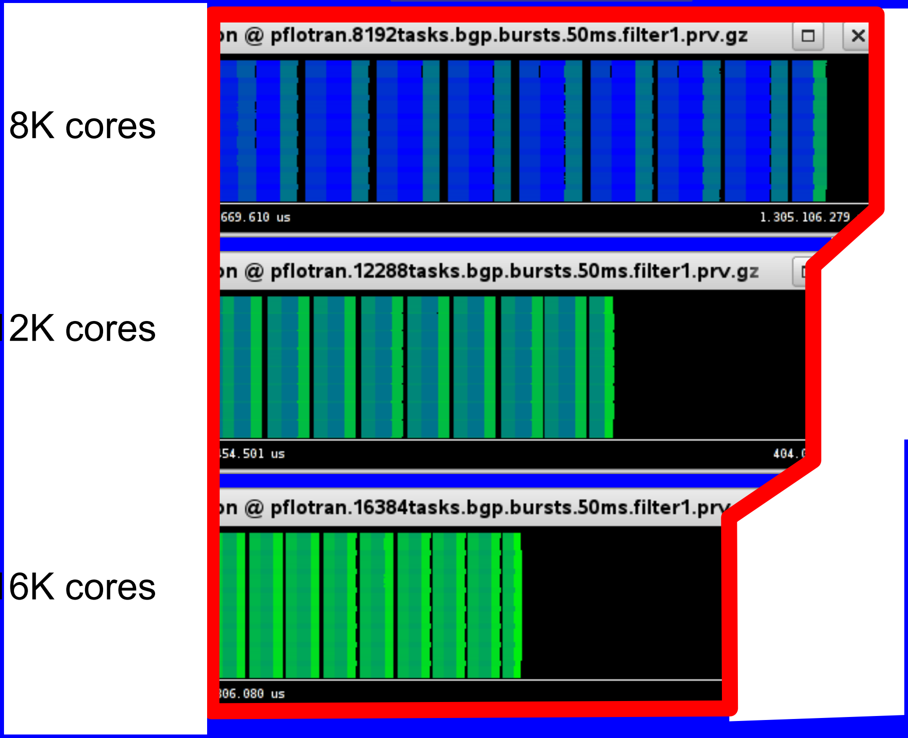
- Linpack @ Marenostrum: 10k cores x 1700 s



Scalability of analysis

Jugene

~ 105 seconds



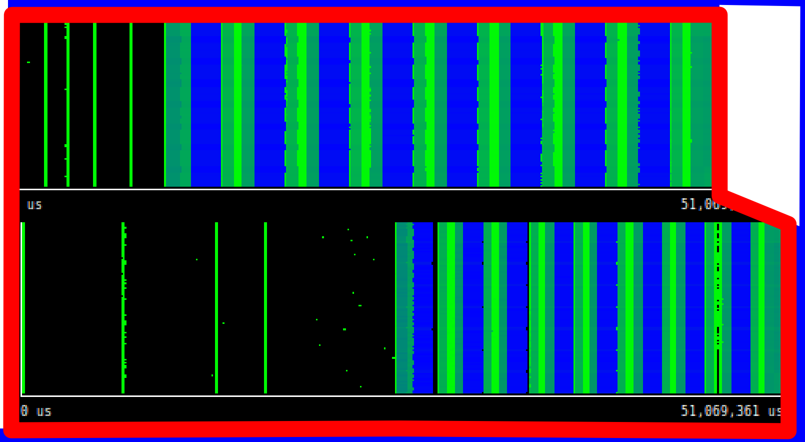
8K cores
on @ pflotran.8192tasks.bgp.bursts.50ms.filter1.prv.gz
669.610 us
1.305.106.279
12K cores
on @ pflotran.12288tasks.bgp.bursts.50ms.filter1.prv.gz
54.501 us
404.0
16K cores
on @ pflotran.16384tasks.bgp.bursts.50ms.filter1.prv
06.080 us

Tran

Flow

Jaguar

~ 47 seconds



Flow

Tran

PFLOTRAN

Data reduction techniques

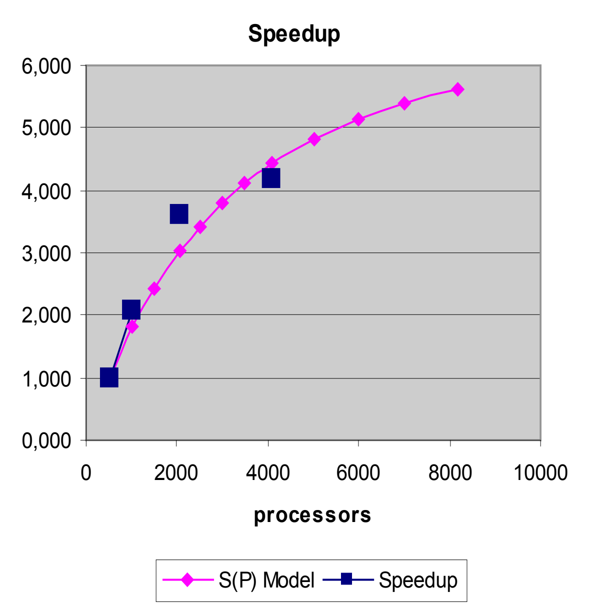
- Software counters
  - Summarize information of some event types (ie. MPI calls) by emitting aggregate counts
  - Emit counts at structurally relevant points (i.e. begin and end of long computation phases)

- Representative cuts
  - Emit full detail only on selected intervals, representative of full program execution

- On and off line combinations
  - By instrumentation
  - By paraver filtering

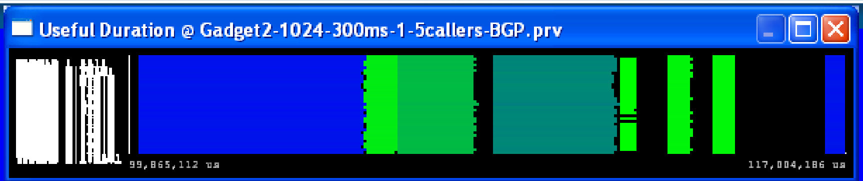
J. Labarta, et al.: "Scalability of tracing and visualization tools", PARCO 2005

Software counters

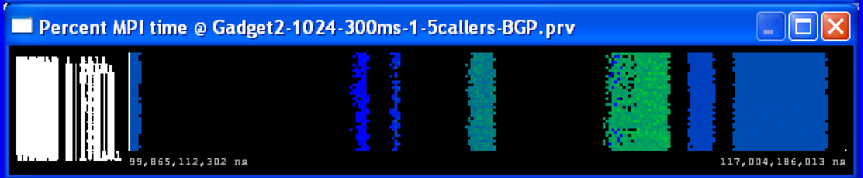


GADGET, PRACE Case A, 1024 procs

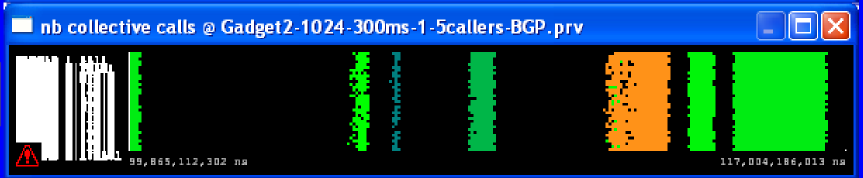
Useful duration



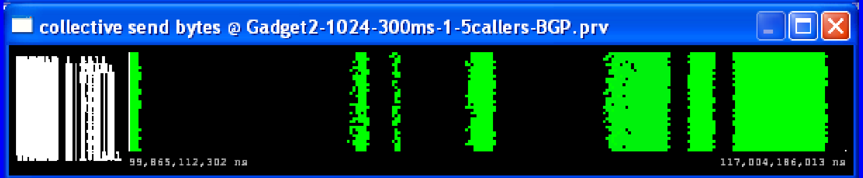
% MPI time



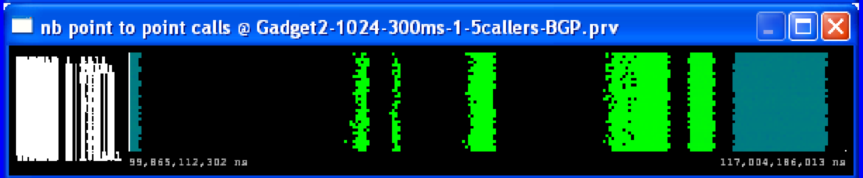
# collectives



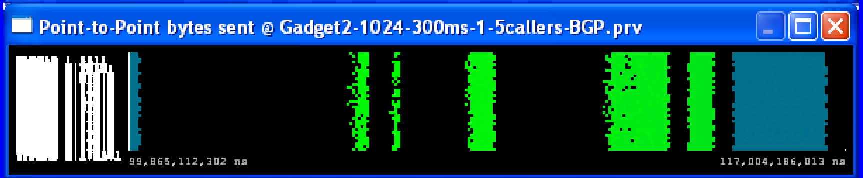
Collective bytes


collective send bytes @ Gadget2-1024-300ms-1-5callers-BGP.prv
99,865,112,302 ns
117,004,186,013 ns

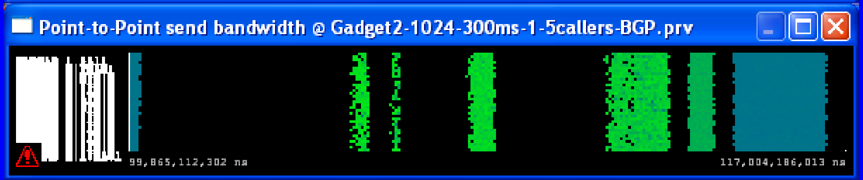
# p2p


nb point to point calls @ Gadget2-1024-300ms-1-5callers-BGP.prv
99,865,112,302 ns
117,004,186,013 ns

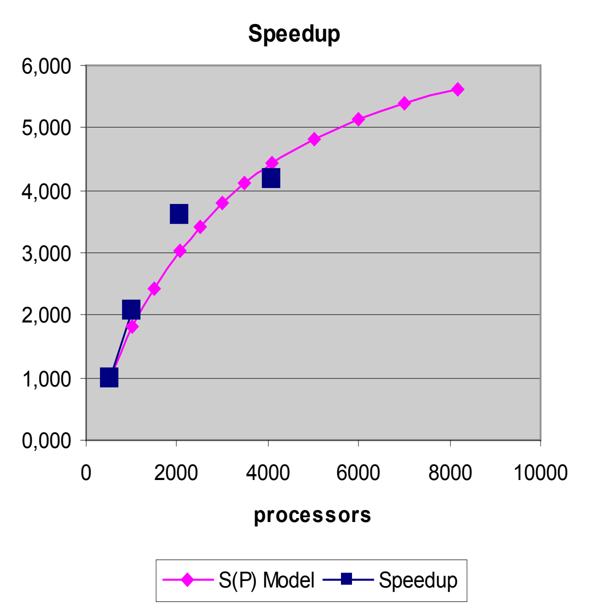
p2p bytes


Point-to-Point bytes sent @ Gadget2-1024-300ms-1-5callers-BGP.prv
99,865,112,302 ns
117,004,186,013 ns

p2p BW

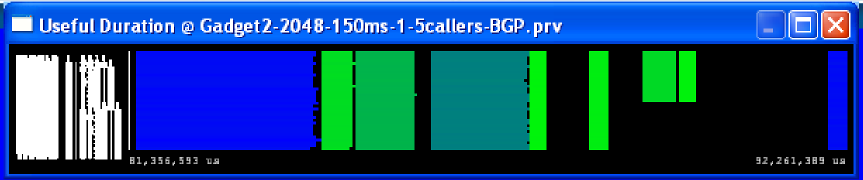

Point-to-Point send bandwidth @ Gadget2-1024-300ms-1-5callers-BGP.prv
99,865,112,302 ns
117,004,186,013 ns

Software counters

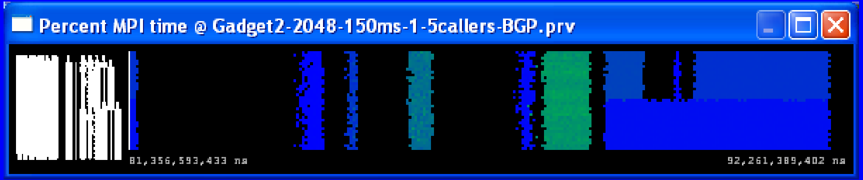


GADGET, PRACE Case A, 2048 procs

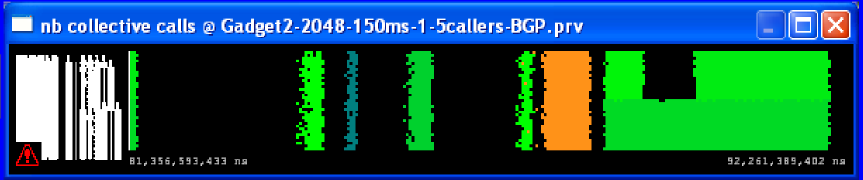
Useful duration


Useful Duration @ Gadget2-2048-150ms-1-5callers-BGP.prv

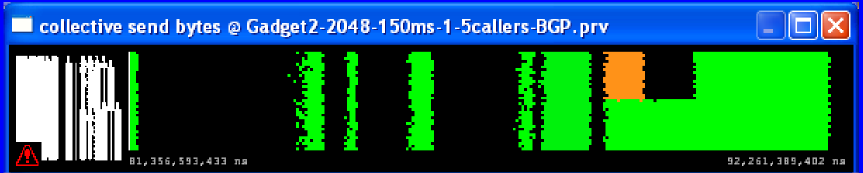
% MPI time


Percent MPI time @ Gadget2-2048-150ms-1-5callers-BGP.prv

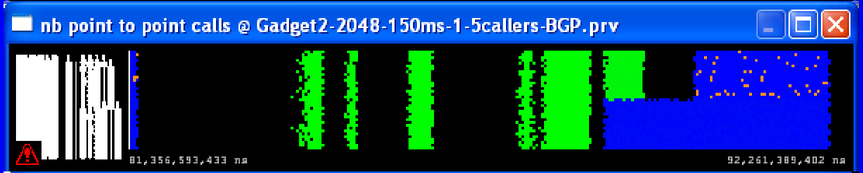
# collectives


nb collective calls @ Gadget2-2048-150ms-1-5callers-BGP.prv

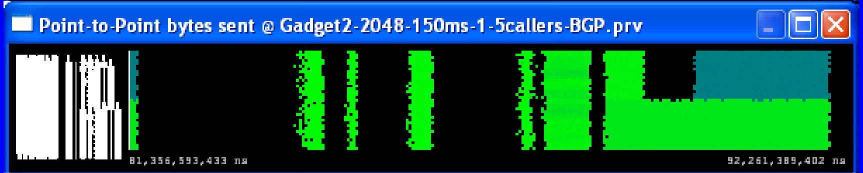
Collective bytes


collective send bytes @ Gadget2-2048-150ms-1-5callers-BGP.prv

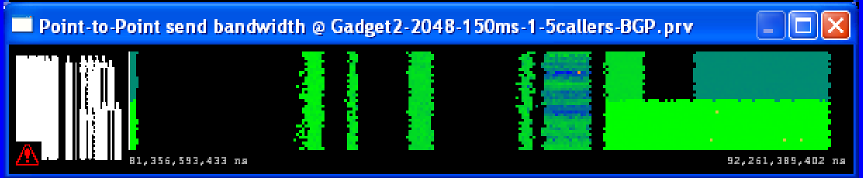
# p2p


nb point to point calls @ Gadget2-2048-150ms-1-5callers-BGP.prv

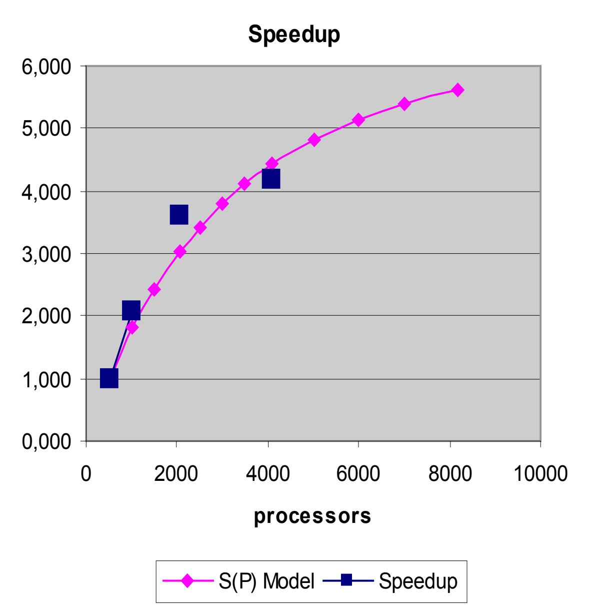
p2p bytes


Point-to-Point bytes sent @ Gadget2-2048-150ms-1-5callers-BGP.prv

p2p BW

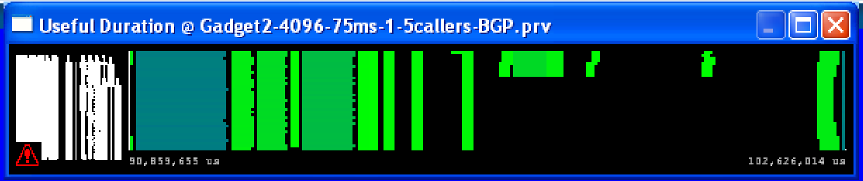

Point-to-Point send bandwidth @ Gadget2-2048-150ms-1-5callers-BGP.prv

Software counters

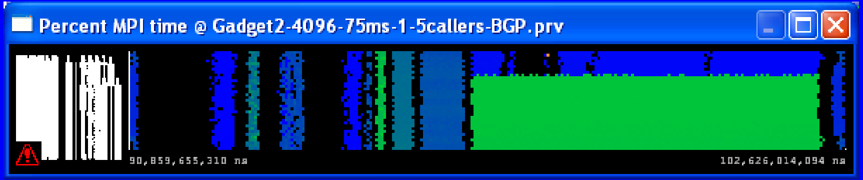


GADGET, PRACE Case A, 4096 procs

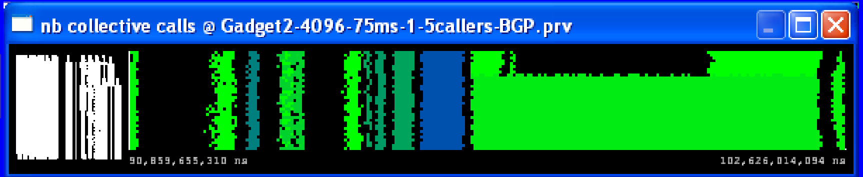
Useful duration



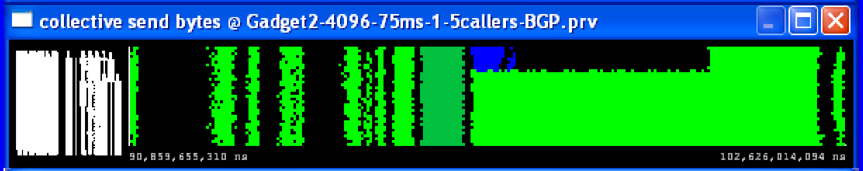
% MPI time



# collectives

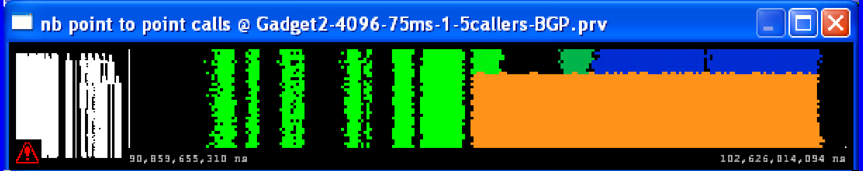


Collective bytes



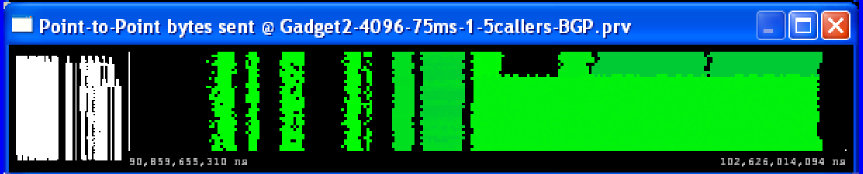
collective send bytes @ Gadget2-4096-75ms-1-5callers-BGP.prv
90,859,655,310 ns
102,626,014,094 ns

# p2p



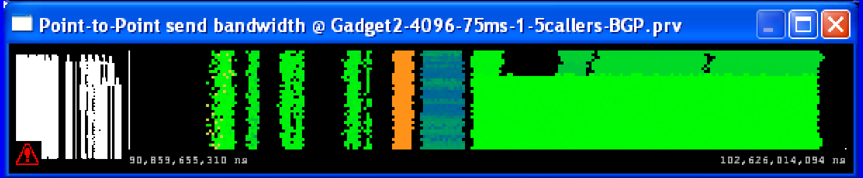
nb point to point calls @ Gadget2-4096-75ms-1-5callers-BGP.prv
90,859,655,310 ns
102,626,014,094 ns

p2p bytes



Point-to-Point bytes sent @ Gadget2-4096-75ms-1-5callers-BGP.prv
90,859,655,310 ns
102,626,014,094 ns

p2p BW



Point-to-Point send bandwidth @ Gadget2-4096-75ms-1-5callers-BGP.prv
90,859,655,310 ns
102,626,014,094 ns