



**Barcelona  
Supercomputing  
Center**  
Centro Nacional de Supercomputación



# Understanding applications with Paraver

Judit Gimenez

[judit@bsc.es](mailto:judit@bsc.es) / [tools@bsc.es](mailto:tools@bsc.es)

5/9/2024

POP3 Training

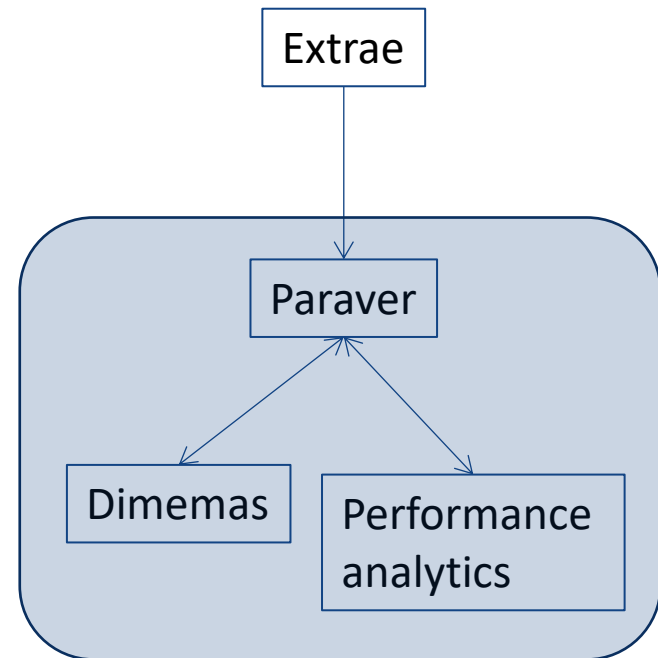
# Humans are visual creatures

- Films or books? PROCESS
  - Two hours vs. days (months)
- Memorizing a deck of playing cards STORE
  - Each card translated to an image (person, action, location)
- Our brain loves pattern recognition IDENTIFY
  - What do you see on the pictures?



# BSC Performance Tools

- Since 1991
- Based on traces
- Open Source (<http://tools.bsc.es>)
- Focus
  - Detail, variability, flexibility
  - Visual analysis
  - Intelligence: Performance Analytics
  - Behavioral structure vs. syntactic structure
  - Key factors



# Paraver



**Barcelona  
Supercomputing  
Center**

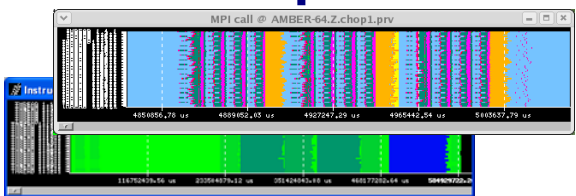
Centro Nacional de Supercomputación

# Paraver – Performance data browser



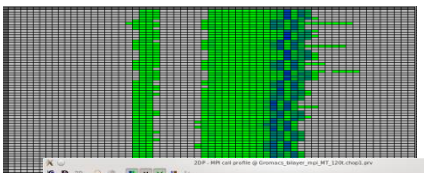
Trace visualization/analysis

+ trace manipulation

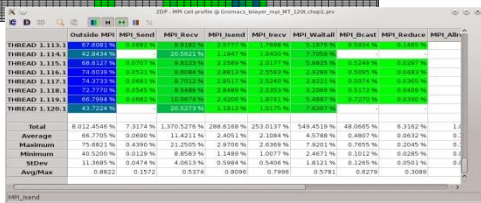


Timelines

Goal = Flexibility  
No semantics  
Programmable



2/3D tables  
(Statistics)



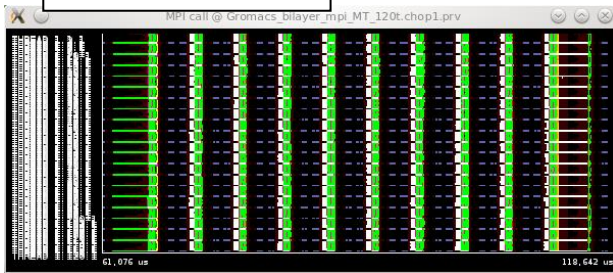
	Outside MPI	MPI_Send	MPI_Recv	MPI_send	MPI_recv	MPI_Waitall	MPI_Bcast	MPI_Reduce	MPI_All
THREAD 1.113.1	57.870%								
THREAD 1.114.1	52.855%								
THREAD 1.115.1	68.812%								
THREAD 1.116.1	72.022%								
THREAD 1.117.1	74.373%								
THREAD 1.118.1	72.777%								
THREAD 1.119.1	60.769%								
THREAD 1.120.1	43.724%								
Total	8.012 45.6%	7.3174 %	1.370 52.76 %	288.8368 %	253.0137 %	549.4510 %	48.8685 %	8.3182 %	1.0
Average	46.770%	0.8066%	11.4211%	3.4051%	2.1084%	43.768%	0.8637%	0.8332%	0.1
Maximum	75.6821%	0.4396%	21.2505%	2.976%	2.6365%	7.9201%	0.7855%	0.2045%	0.1
Minimum	40.520%	0.0119%	0.8063%	1.1486%	1.0077%	2.4871%	0.1021%	0.0205%	0.0
StdDev	11.366%	0.0274%	4.0611%	0.5684%	0.3406%	1.8121%	0.1295%	0.0501%	0.1
Avg*Max	0.8822	0.1572	0.5374	0.8096	0.7998	0.5781	0.6278	0.3089	

Comparative analyses  
Multiple traces  
Synchronize scales

# From timelines to tables

- From timelines to tables

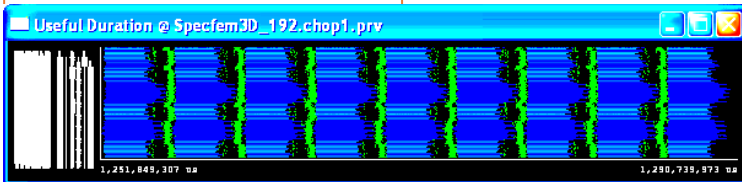
MPI calls



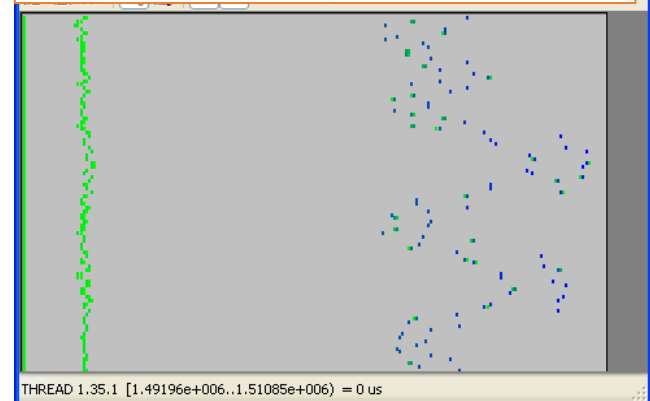
MPI calls profile

	Outside MPI	MPI Send	MPI Recv	MPI Isend	MPI Irecv	MPI Waitall	MPI Bcast	MPI Reduce	MPI All
THREAD 1.113.1	67.6081 %	0.0592 %	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 %	-	-	-	
THREAD 1.117.1	74.3733 %	0.0591 %	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 %	-	-	-	
<b>Total</b>	8,012.4546 %	7.3174 %	1,370.5276 %	288.6168 %	253.0137 %	54			
<b>Average</b>	66.7705 %	0.0690 %	11.4211 %	2.4051 %	2.1084 %				
<b>Maximum</b>	75.6821 %	0.4390 %	21.2505 %	2.9706 %	2.6369 %				
<b>Minimum</b>	40.5200 %	0.0129 %	8.8583 %	1.1489 %	1.0077 %				
<b>StDev</b>	11.3685 %	0.0474 %	4.0613 %	0.5984 %	0.5406 %				
<b>Avg/Max</b>	0.8822	0.1572	0.5374	0.8096	0.7996				

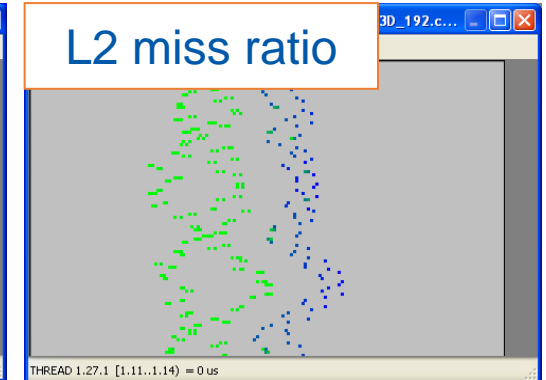
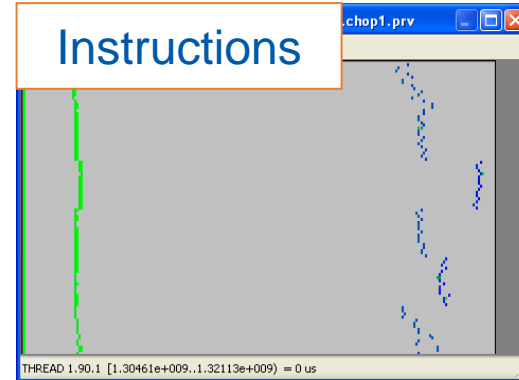
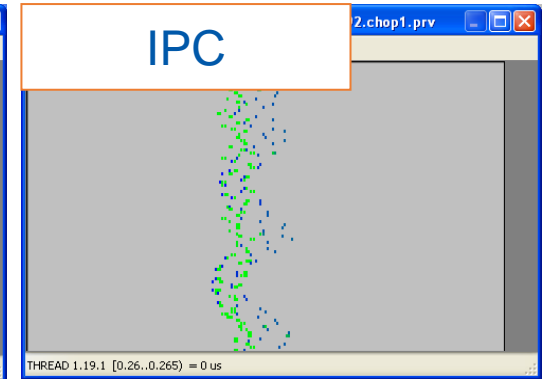
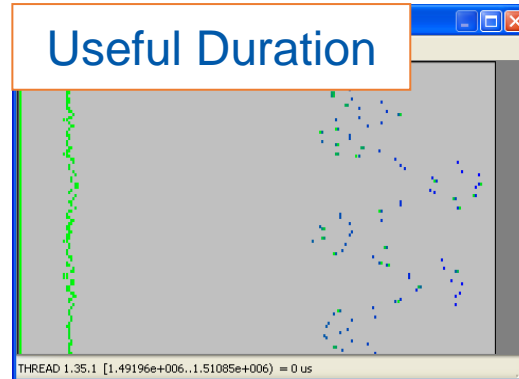
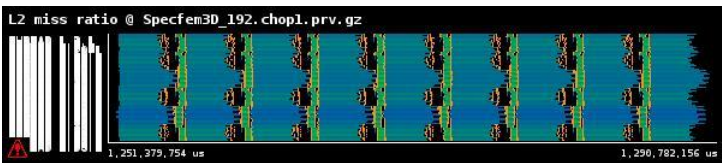
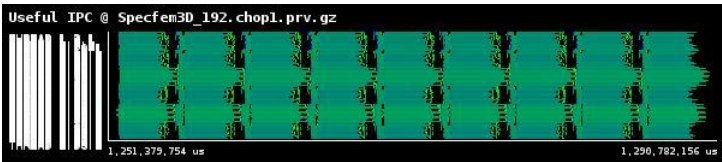
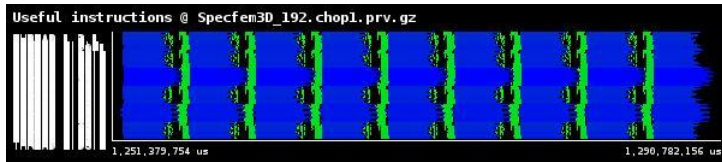
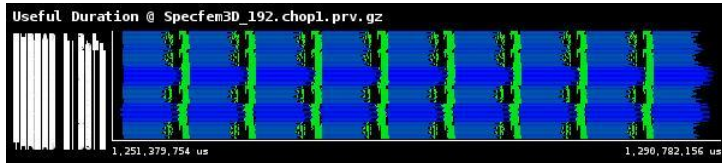
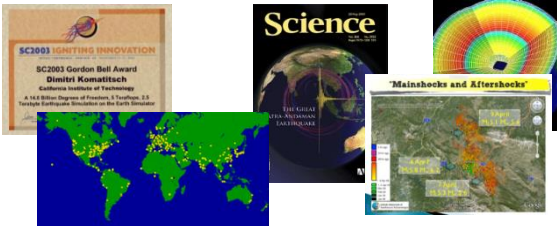
Useful Duration



Histogram Useful Duration

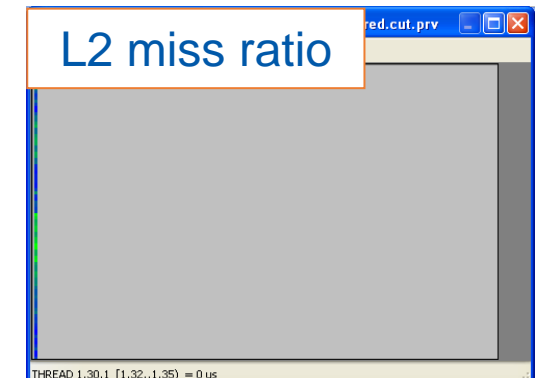
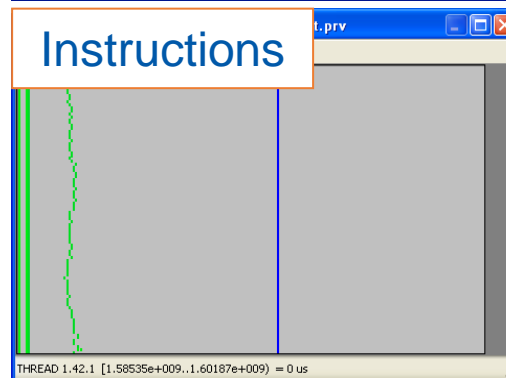
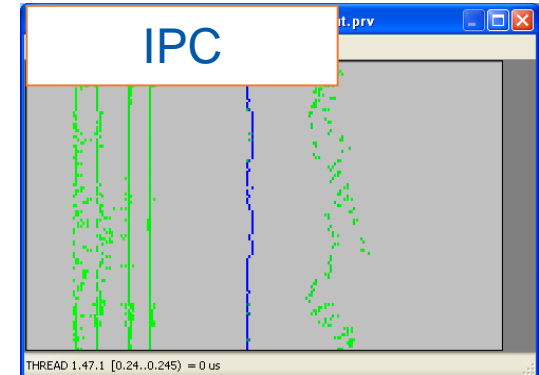
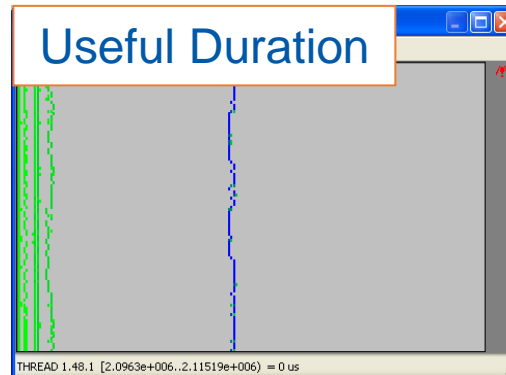


# Analyzing variability



# Analyzing variability

- By the way: six months later ....

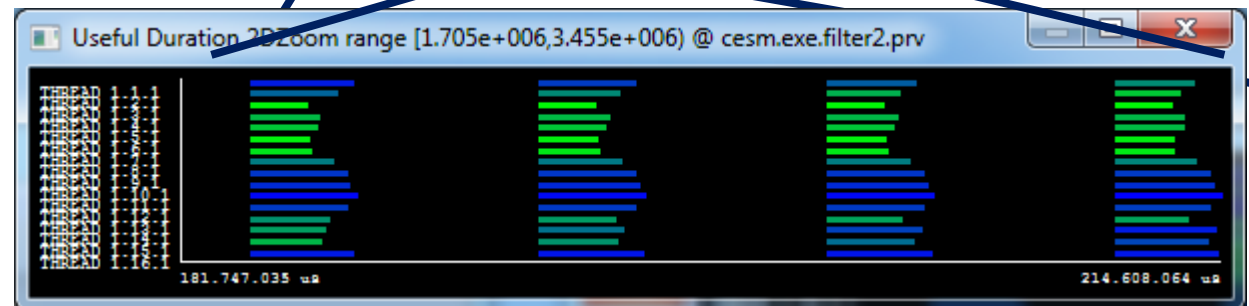
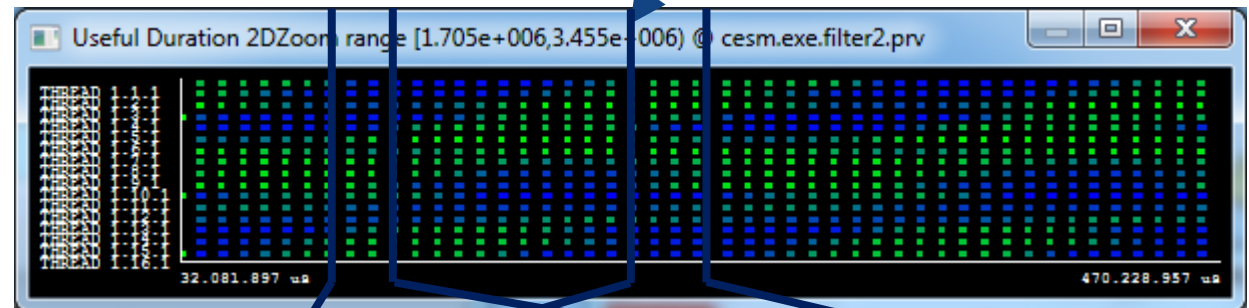
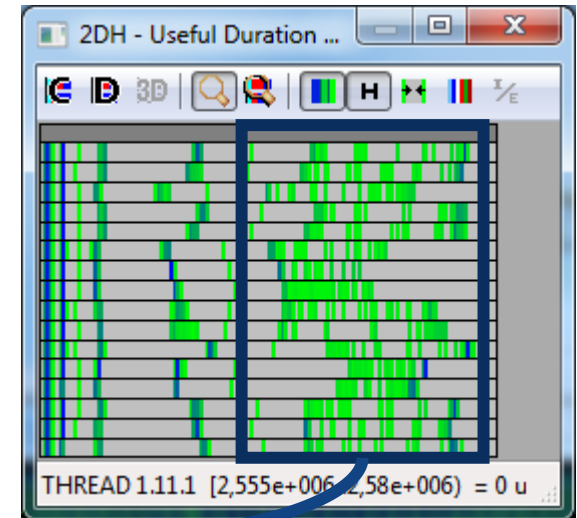




# From tables to timelines

CESM: 16 processes, 2 simulated days

- Histogram useful computation duration shows high variability
- How is it distributed?
- Dynamic imbalance
  - In space and time
  - Day and night.
  - Season ? ☺





# Dimemas

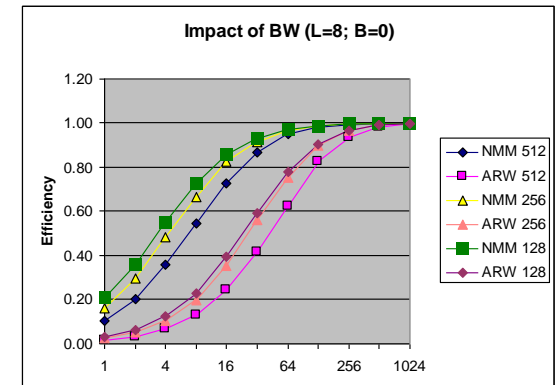
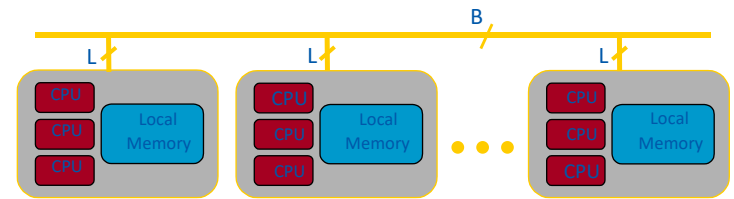


**Barcelona  
Supercomputing  
Center**

Centro Nacional de Supercomputación

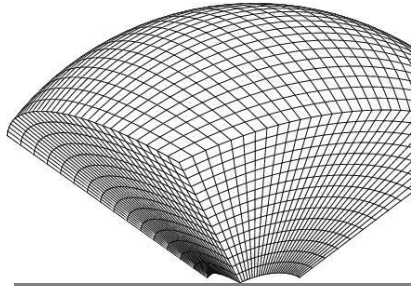
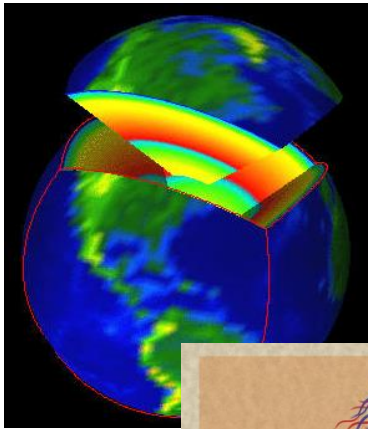
# Dimemas: Coarse grain, Trace driven simulation

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



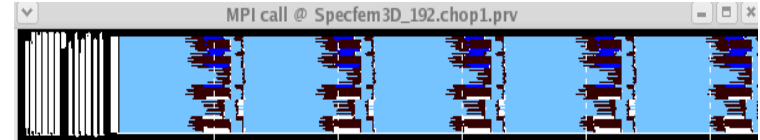
# What if we had asynchronous comms

- SPECFEM3D



Courtesy Dimitri Komatitsch

Real



Ideal



Prediction  
MN



# Ideal machine

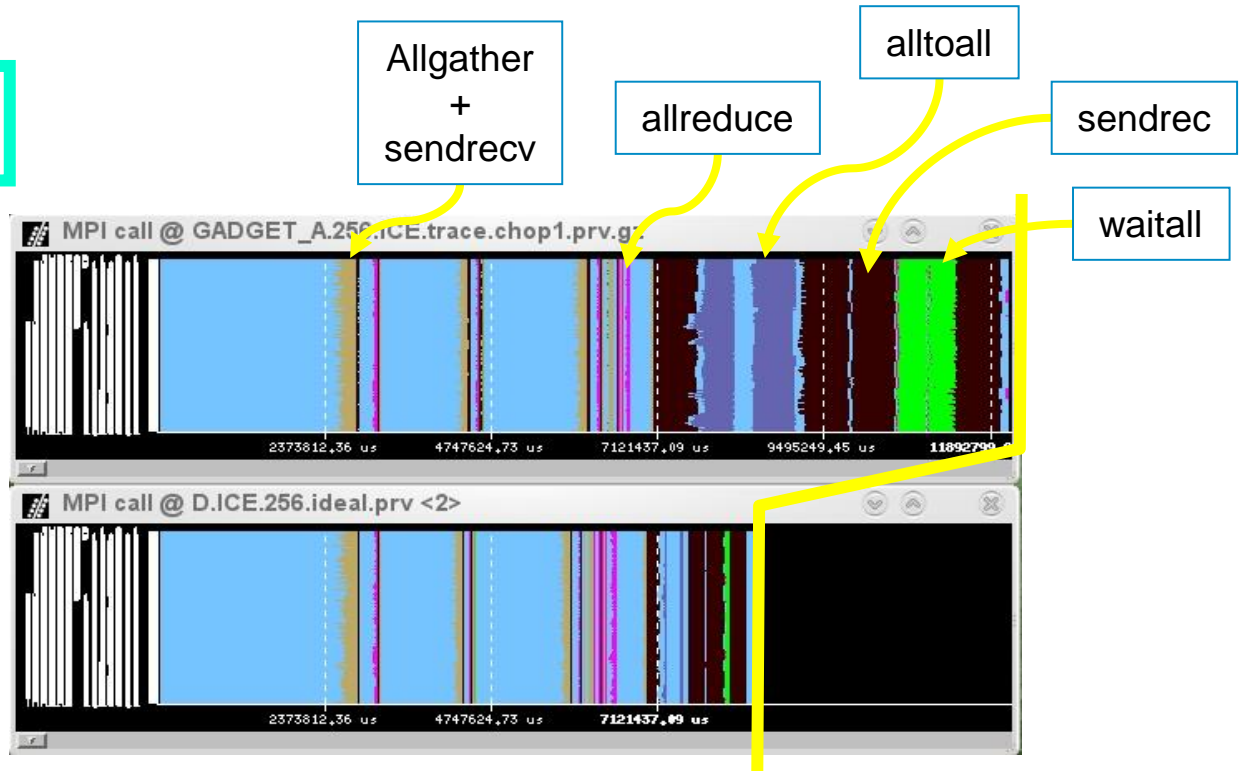
The impossible machine:  $BW = \infty$ ,  $L = 0$

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

GADGET @ Nehalem cluster  
256 processes

Real  
run

Ideal  
network

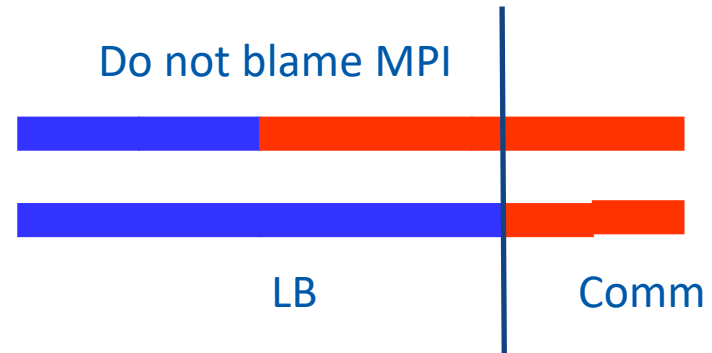
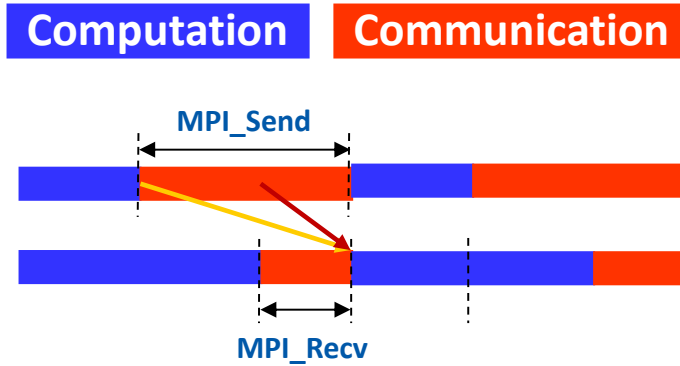


# Efficiency Model



**Barcelona  
Supercomputing  
Center**  
Centro Nacional de Supercomputación

# Parallel efficiency model

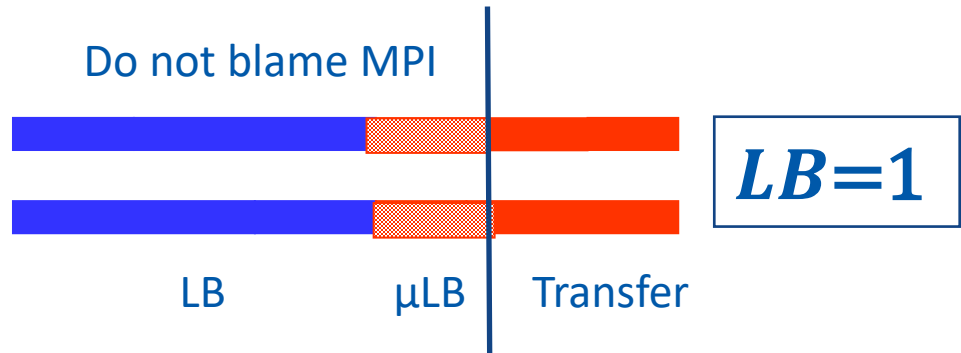
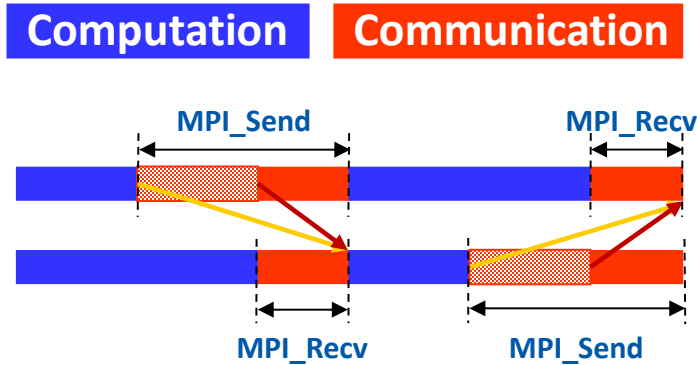


- Parallel efficiency = LB eff \* Comm eff

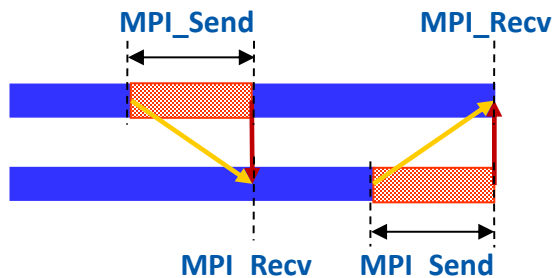


# Parallel efficiency refinement:

$$LB * \mu LB * Tr$$



- Serializations / dependences ( $\mu LB$ )
- Dimemas ideal network  $\rightarrow$  Transfer (efficiency) = 1

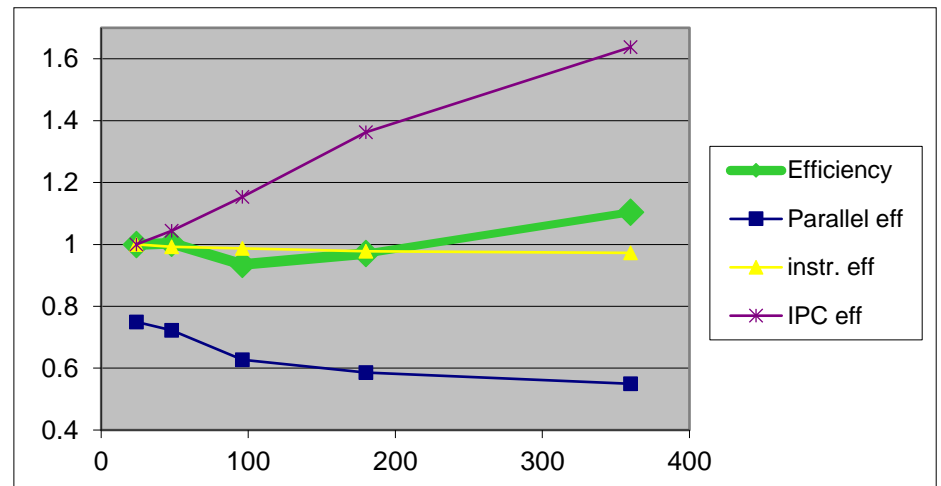
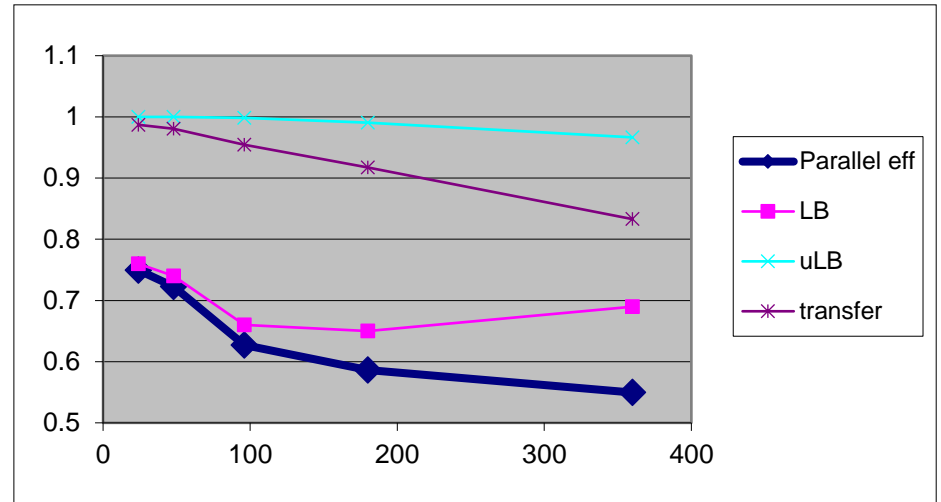
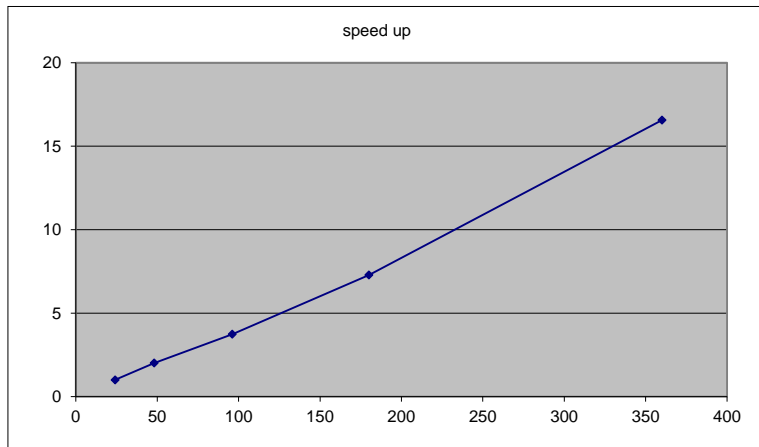


# Why scaling?

$$\eta_{\parallel} = LB * Ser * Trf$$

CG-POP mpi2s1D - 180x120

Good scalability !!  
Should we be happy?

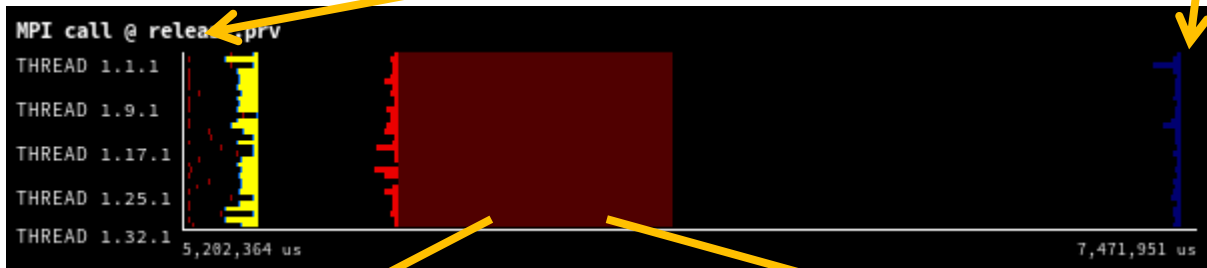


# Why efficient?

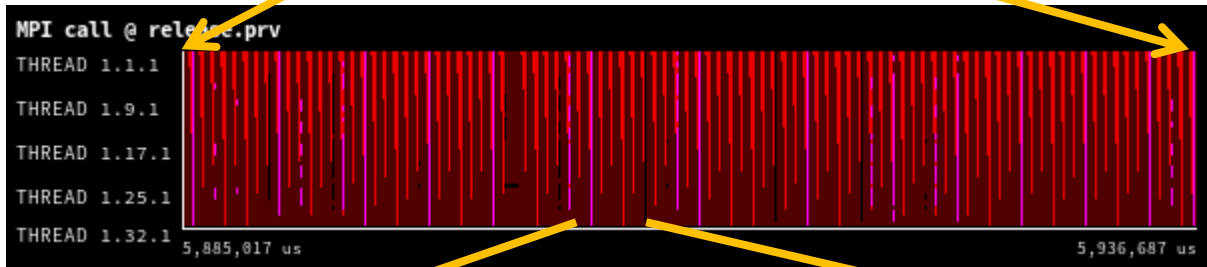
Parallel efficiency = 93.28  
Communication = 93.84



Parallel efficiency 77.93  
Communication eff . 79.79



Parallel efficiency 28.84  
Communication eff . 30.42



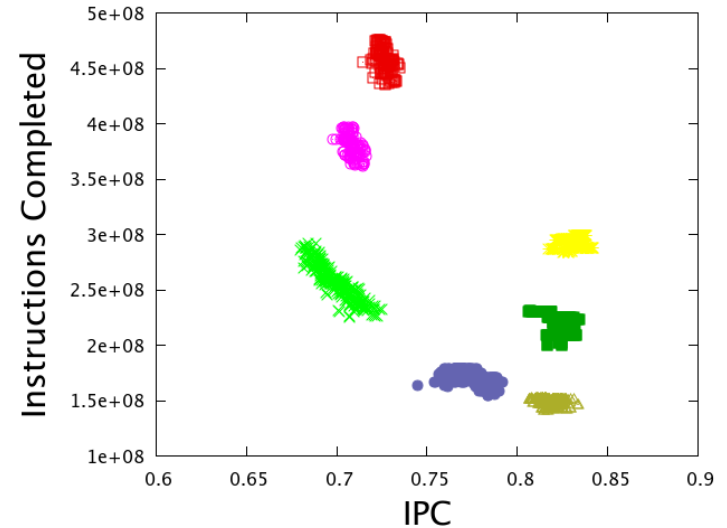
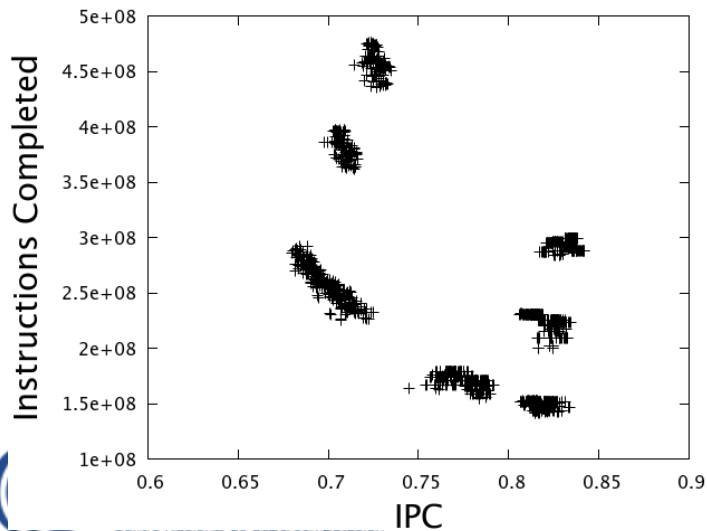
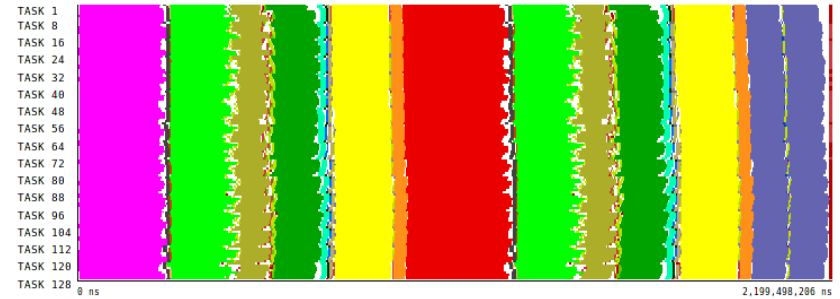
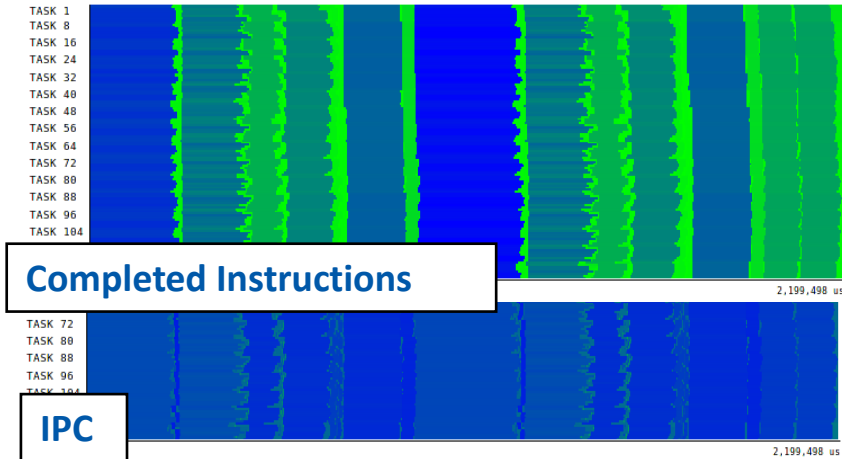
# Analytics



**Barcelona  
Supercomputing  
Center**

Centro Nacional de Supercomputación

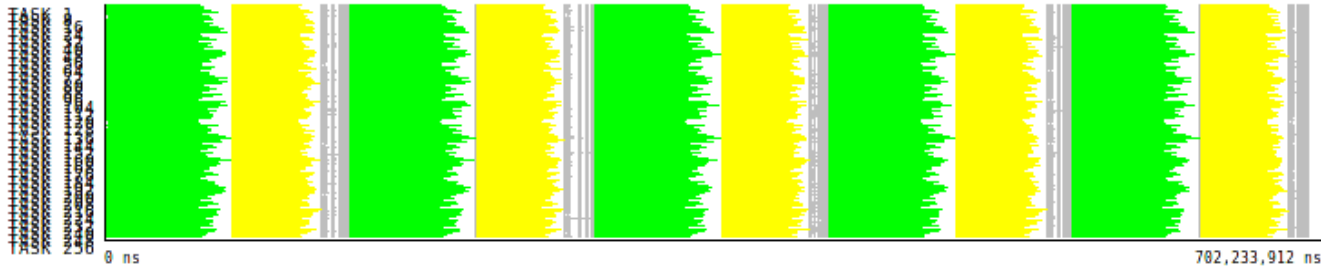
# Using Clustering to identify structure



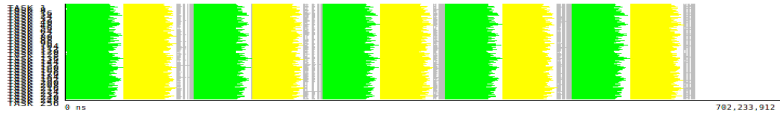
# What should I improve?

What if ....

PEPC

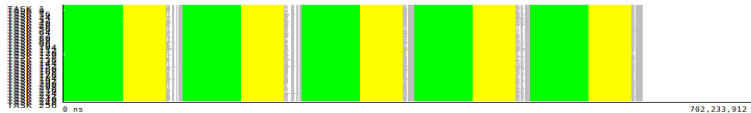
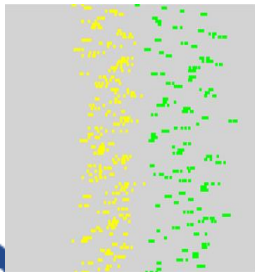


... we increase the IPC of Cluster1?



13% gain

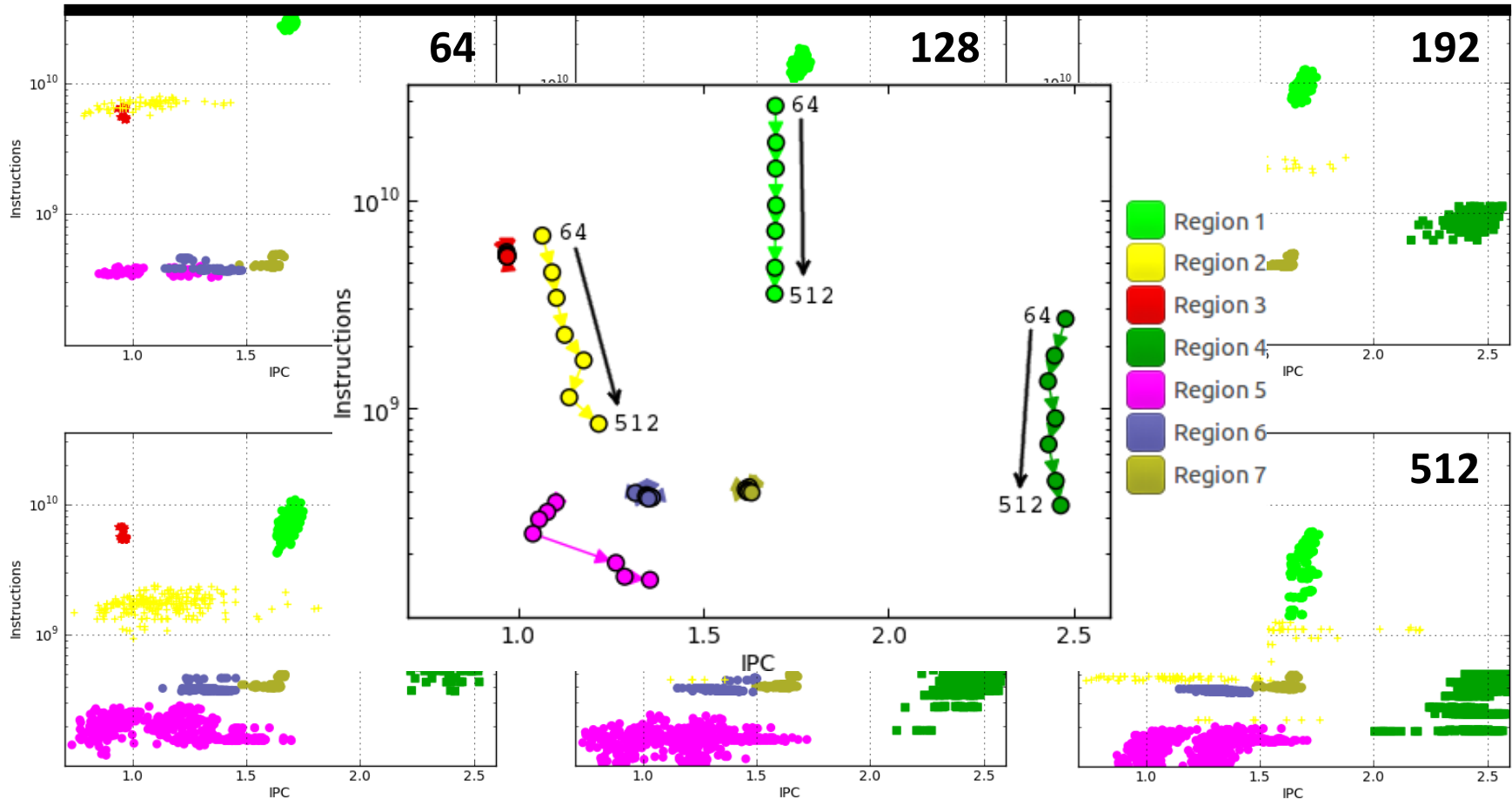
... we balance Clusters 1 & 2?



19% gain

# Tracking scalability through clustering

- OpenMX (strong scale from 64 to 512 tasks)



# Methodology



**Barcelona  
Supercomputing  
Center**  
Centro Nacional de Supercomputación



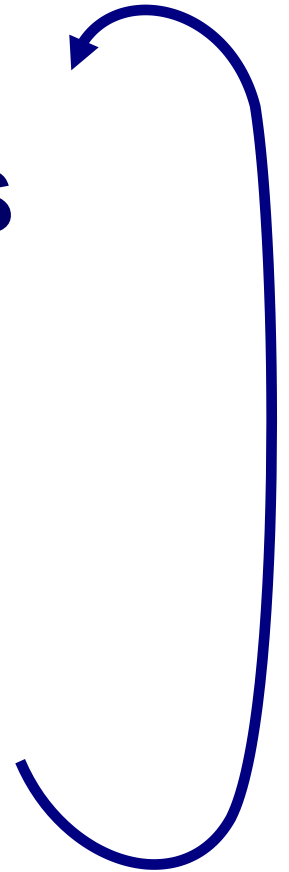
# Performance analysis tools objective

**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?

Paraver Tutorial:  
Introduction to Paraver and Dimemas methodology

# BSC Tools web site

- tools.bsc.es
  - 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

# Some recommendations

- The power of understanding → Keep asking questions
- Our brain is more efficient with images → Use visual tools
- The key uses to be in the details → Do not miss them
- Do not expect what may be expected → Be flexible

And as Bruce Lee said, “Be water my friend!”

# Demo



**Barcelona  
Supercomputing  
Center**

Centro Nacional de Supercomputación

# Same code, different behaviour

- Lulesh 2.0
  - Easy to install
  - Requires a cube number of MPI ranks
- What about 27? Check how the system reacts to a “weird” request

Code	Parallel efficiency	Communication eff.	Load Balance eff.
lulesh@mn3	90.55	<b>99.22</b>	91.26
lulesh@leftraru	<b>69.15</b>	99.12	<b>69.76</b>
lulesh@uv2 (mpt)	70.55	96.56	73.06
lulesh@uv2 (impi)	85.65	95.09	90.07
lulesh@mt	83.68	95.48	87.64
lulesh@cori	90.92	98.59	92.20
lulesh@thunderX	73.96	97.56	75.81
lulesh@jetson	75.48	<b>88.84</b>	84.06
lulesh@claix	77.28	92.33	83.70
lulesh@jureca	88.20	98.45	89.57
lulesh@inti	88.16	98.65	89.36
lulesh@archer	88.01	97.95	89.86
lulesh@romeo	89.56	99.01	90.45
lulesh@mn4	<b>91.02</b>	98.38	<b>92.52</b>
lulesh@ stampede2 (skl)	85.76	97.63	87.84
lulesh@ stampede2 (knl)	89.21	98.42	90.64
lulesh@isambard	90.32	97.16	92.96
lulesh@hawk (mpt)	80.16	98.98	80.98
lulesh@hawk (openmpi)	87.82	98.28	89.35

Warning::: Higher parallel efficiency does not mean faster!