Understanding applications using the BSC performance tools

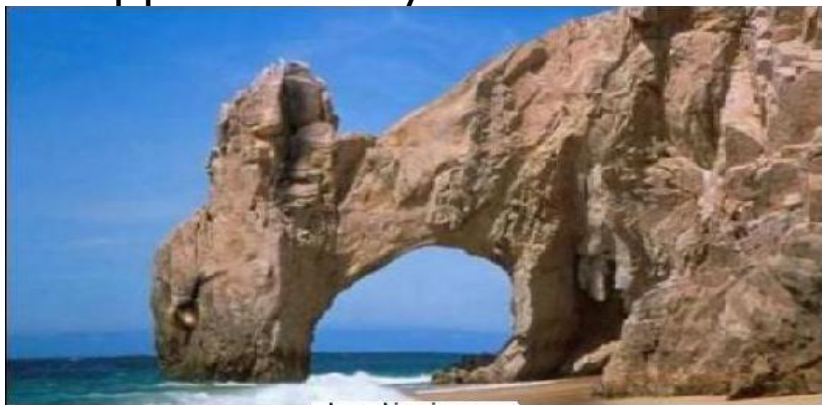
Lau Mercadal, Germán Llort

✉ tools@bsc.es

Barcelona Supercomputing Center

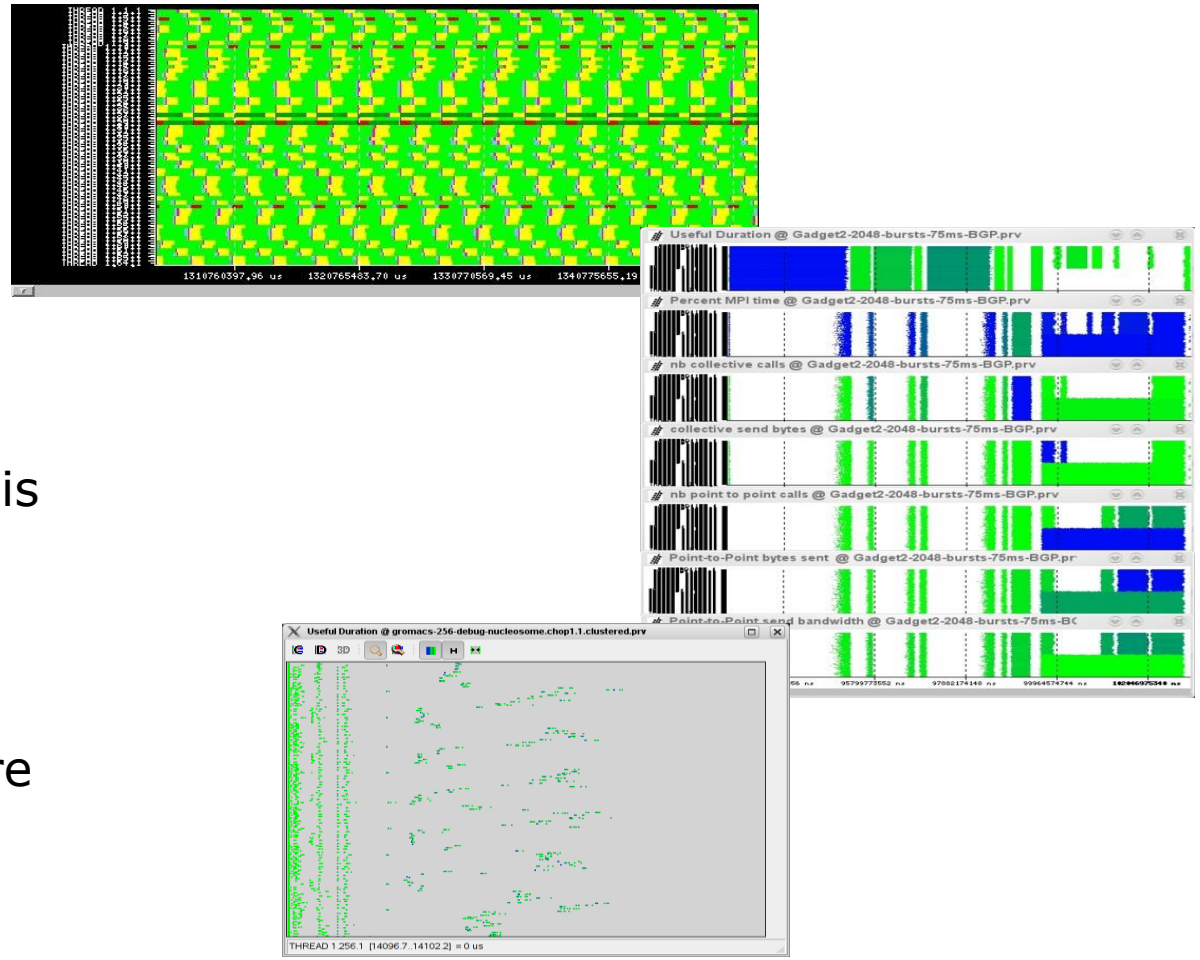
Humans are visual creatures

- Painting / photo or description? PROCESS
  - Our brain processes visual impressions 60,000 times faster than text
  - It takes only 13 ms for the human brain to process an image
- Memorizing a deck of playing cards STORE
  - Each card translated to an image (person, action, location)
- Our brain loves pattern recognition IDENTIFY
  - 90% of all communication happens visually
  - The human eye can differentiate approximately 10 million different colours

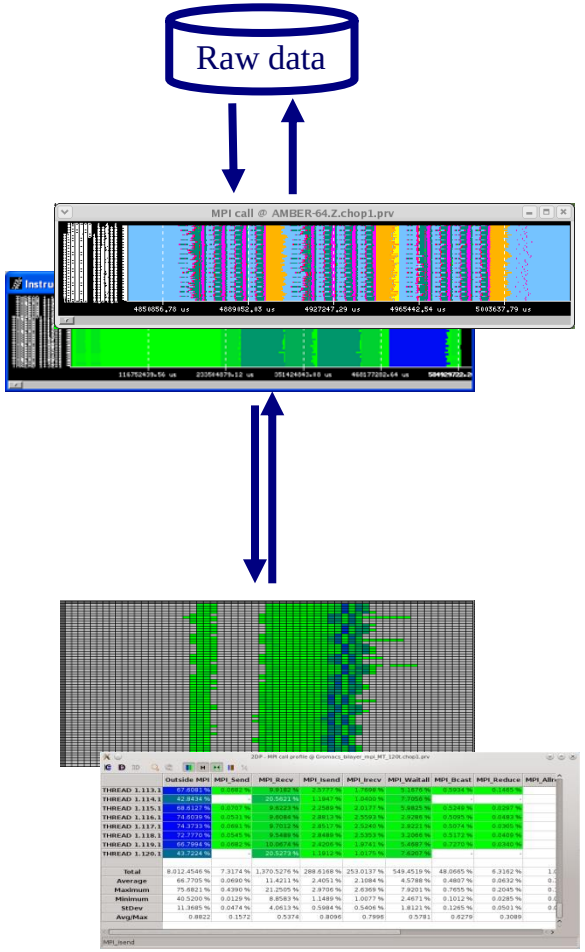

Our Tools

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



Paraver

Paraver: Performance data browser



Timelines

2/3D tables (Statistics)

Trace visualization/analysis

+ trace manipulation

Goal = Flexibility

No semantics

Programmable

Comparative analyses

Multiple traces

Synchronize scales

From timelines to tables

MPI calls

MPI calls profile

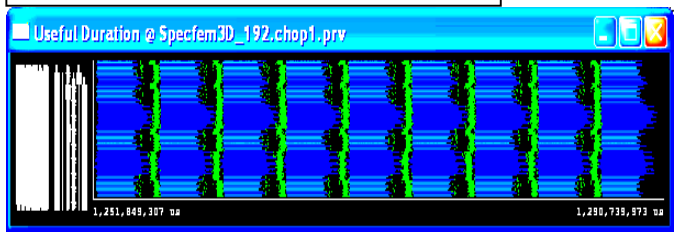
	Outside MPI	MPI_Send	MPI_Recv	MPI_Isend	MPI_Irecv	MPI_Waitall	MPI_Bcast	MPI_Reduce	MPI_Allr
THREAD 1.113.1	67.6081 %	0.0682 %	9.9182 %	2.5777 %	1.7698 %	5.1676 %	0.5934 %	0.1465 %	
THREAD 1.114.1	42.8434 %	-	20.5621 %	1.1947 %	1.0400 %	7.7056 %	-	-	
THREAD 1.115.1	68.6127 %	0.0707 %	9.6223 %	2.2589 %	2.0177 %	5.9825 %	0.5249 %	0.0297 %	
THREAD 1.116.1	74.6039 %	0.0531 %	9.6084 %	2.8813 %	2.5593 %	2.9286 %	0.5095 %	0.0483 %	
THREAD 1.117.1	74.3733 %	0.0691 %	9.7012 %	2.8517 %	2.524				
THREAD 1.118.1	72.7770 %	0.0545 %	9.5489 %	2.8489 %	2.535				
THREAD 1.119.1	66.7994 %	0.0682 %	10.0674 %	2.4206 %	1.974				
THREAD 1.120.1	43.7224 %	-	20.5273 %	1.1912 %	1.01				
Total	8,012.4546 %	7.3174 %	1,370.5276 %	288.6168 %	253.013				
Average	66.7705 %	0.0690 %	11.4211 %	2.4051 %	2.108				
Maximum	75.6821 %	0.4390 %	21.2505 %	2.9706 %	2.636				
Minimum	40.5200 %	0.0129 %	8.8583 %	1.1489 %	1.007				
StDev	11.3685 %	0.0474 %	4.0613 %	0.5984 %	0.540				
Avg/Max	0.8822	0.1572	0.5374	0.8096	0.7				

MPI_Isend

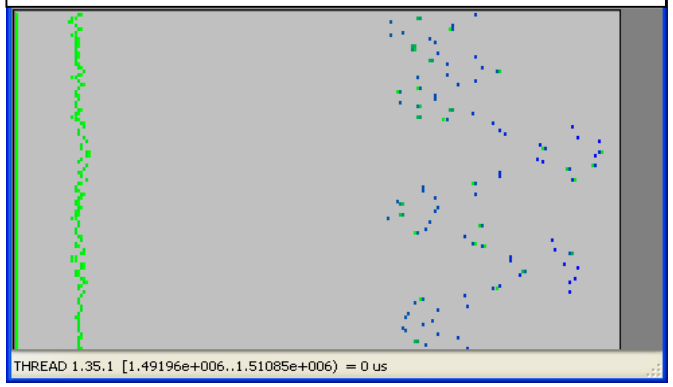
2DP - MPI call profile @ Gromacs_bilayer_mpi_MT_120t.c...

THREAD 1.107.1 MPI_Sendrecv = 11.3412 %

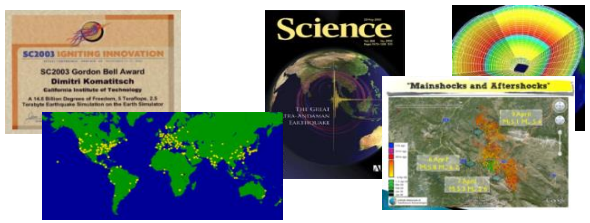
Useful Duration



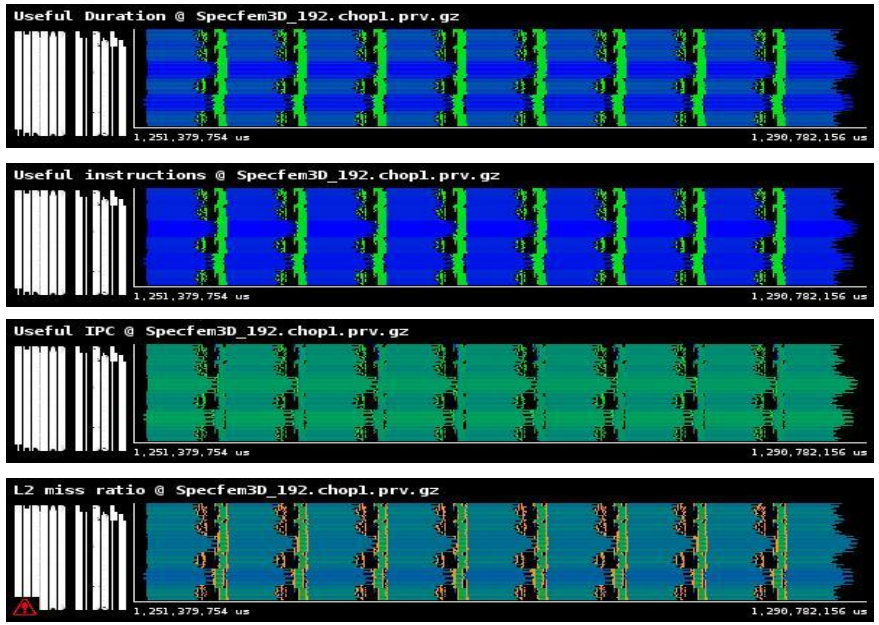
Histogram Useful Duration



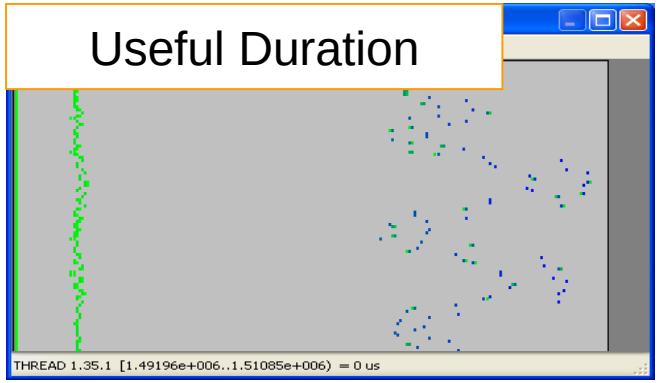
Analyzing variability through histograms and timelines



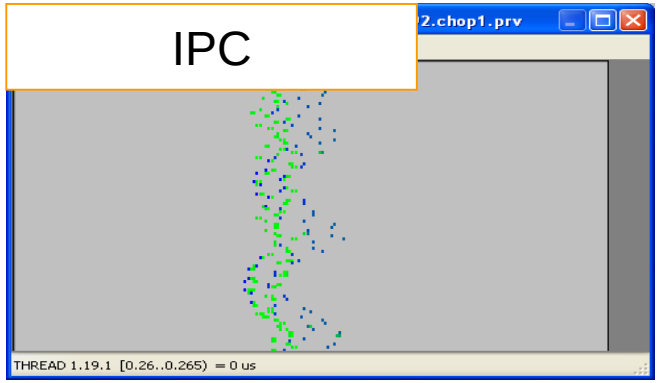
SPECFEM3D



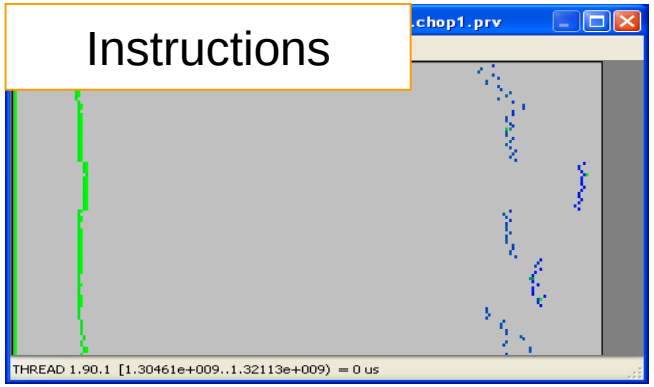
Useful Duration



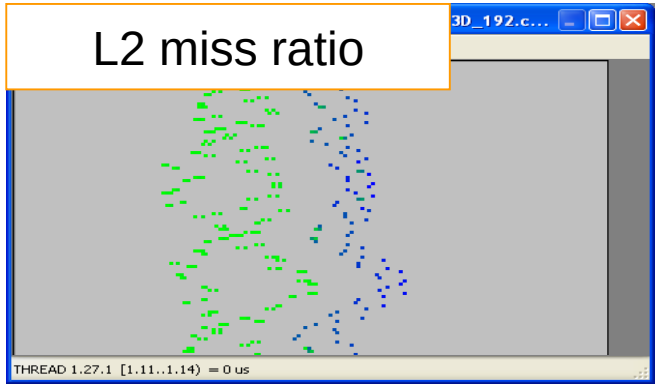
IPC



Instructions

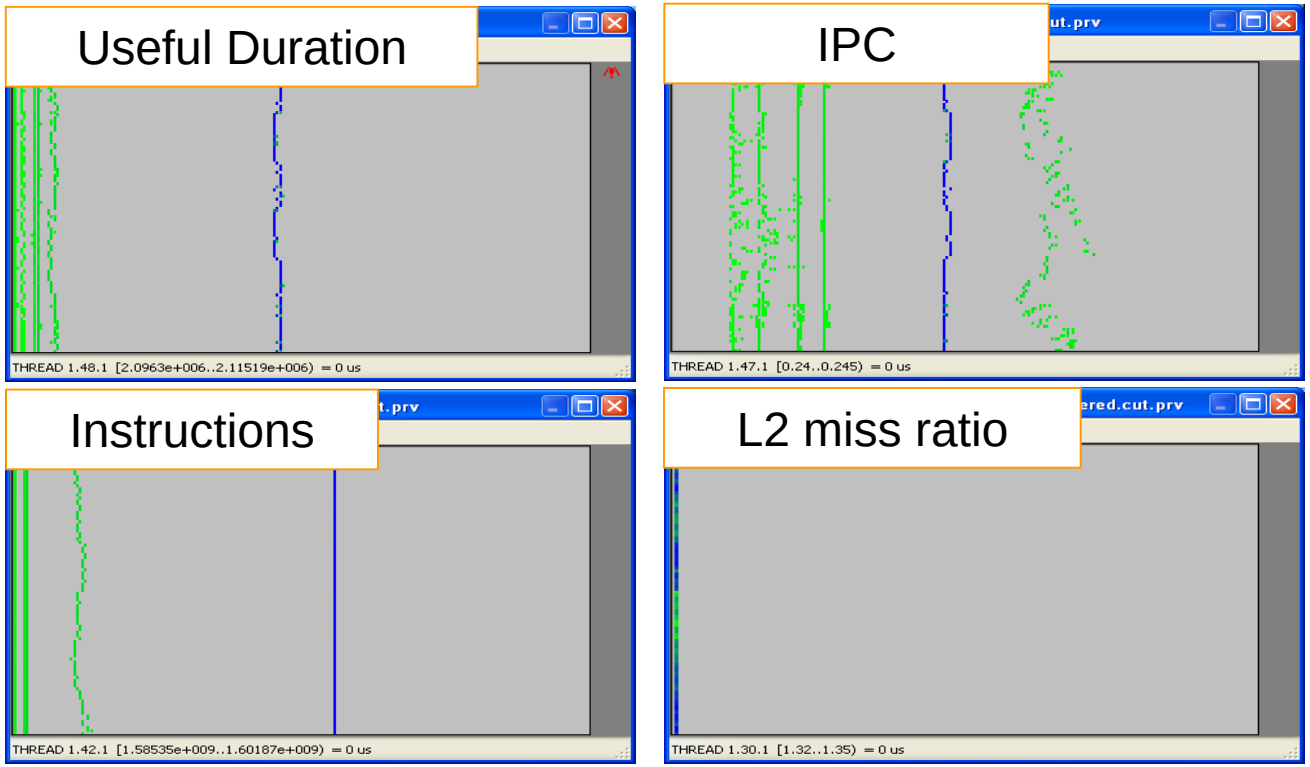


L2 miss ratio



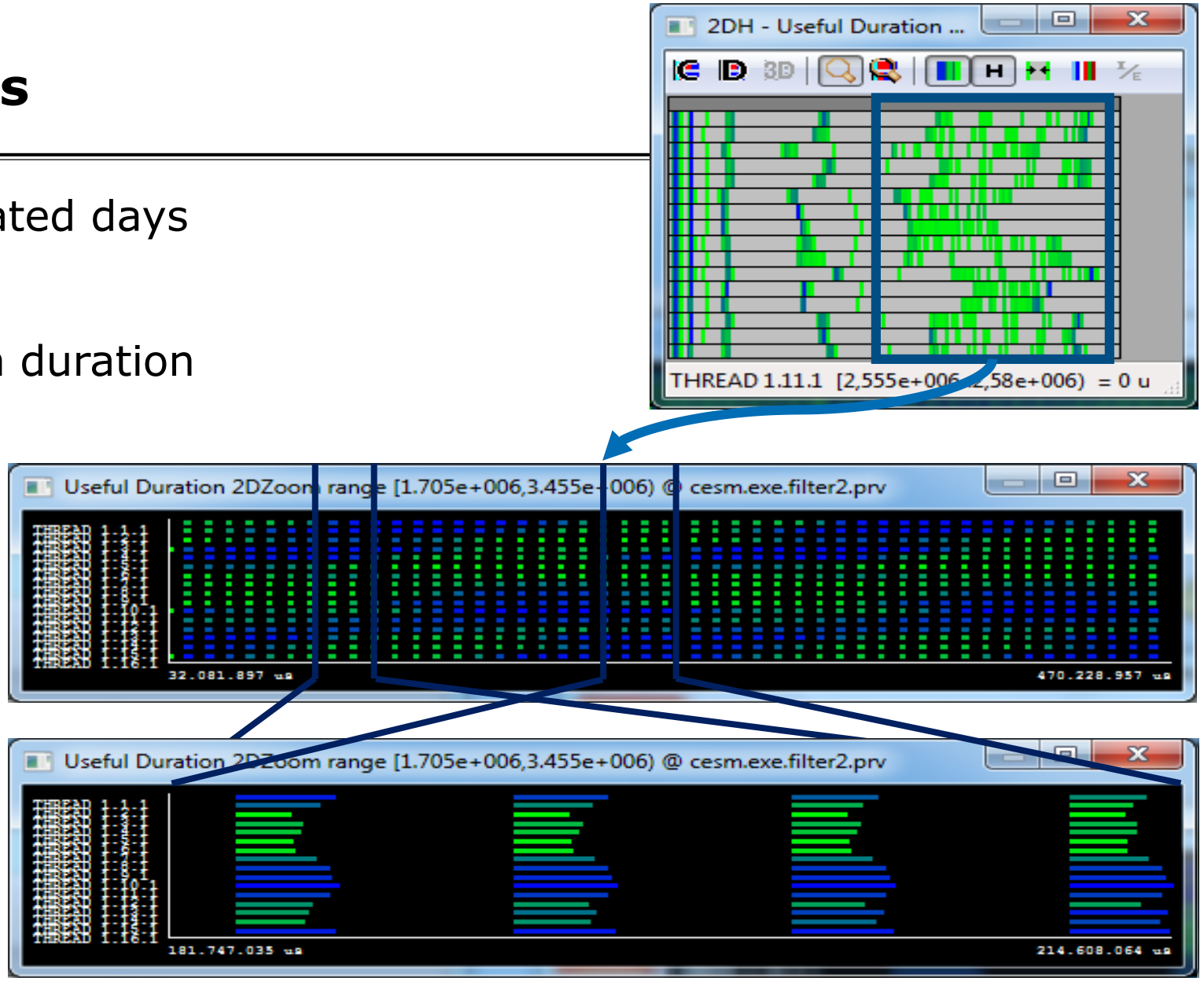
Analyzing variability through histograms and timelines

- By the way: six months later …



From tables to timelines

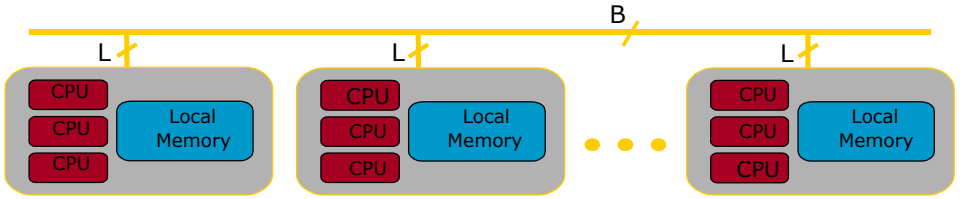
- CESM: 16 processes, 2 simulated days
- Histogram useful computation duration shows high variability
  - How is it distributed?
- Dynamic imbalance
  - In space and time
  - Day and night.
  - Season ? ☺

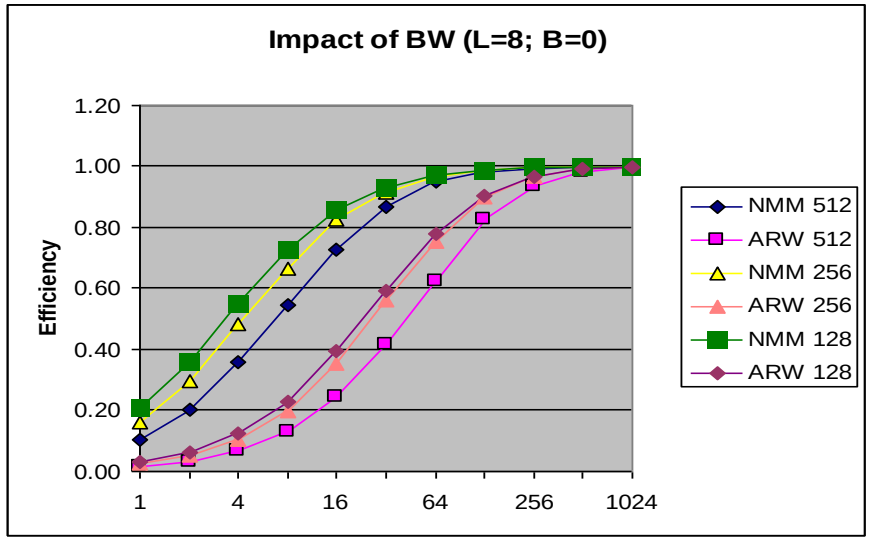


Dimemas

Dimemas: Coarse grain, Trace driven simulation

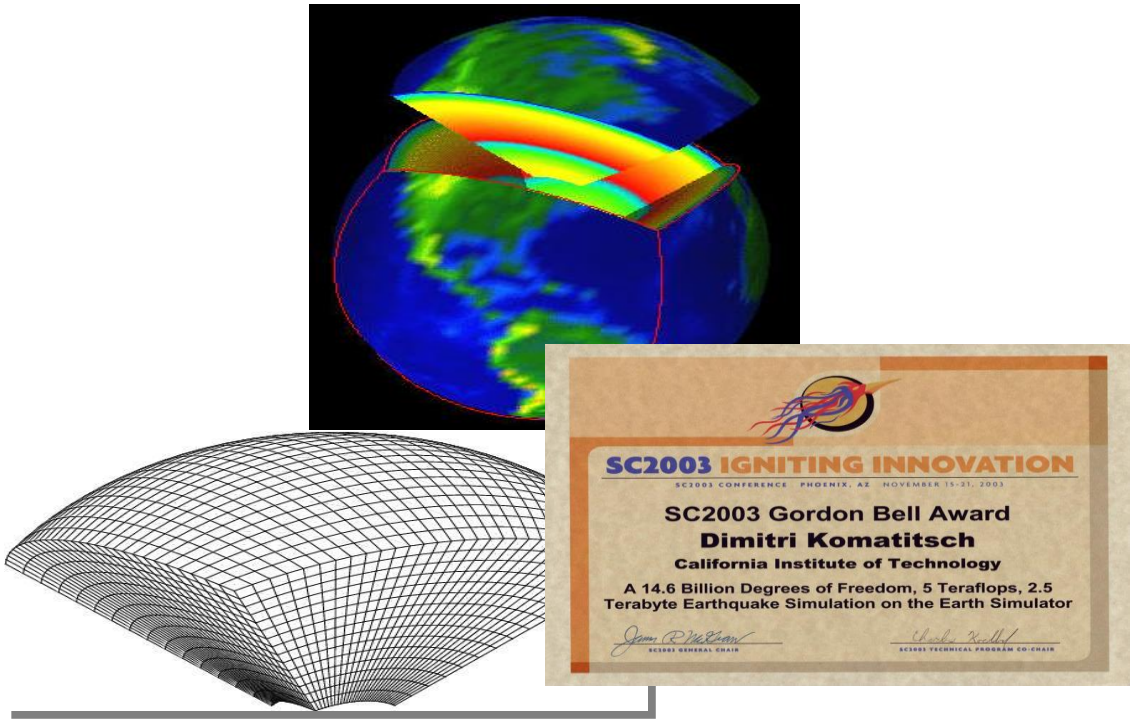
- Simulation: Highly non linear model
  - MPI protocols, resources contention…
- Parametric sweeps
  - On abstract architectures
  - On application computational regions
- What if analysis
  - Ideal machine (instantaneous network)
  - Estimating impact of ports to MPI+OpenMP/CUDA/…
  - Should I use asynchronous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



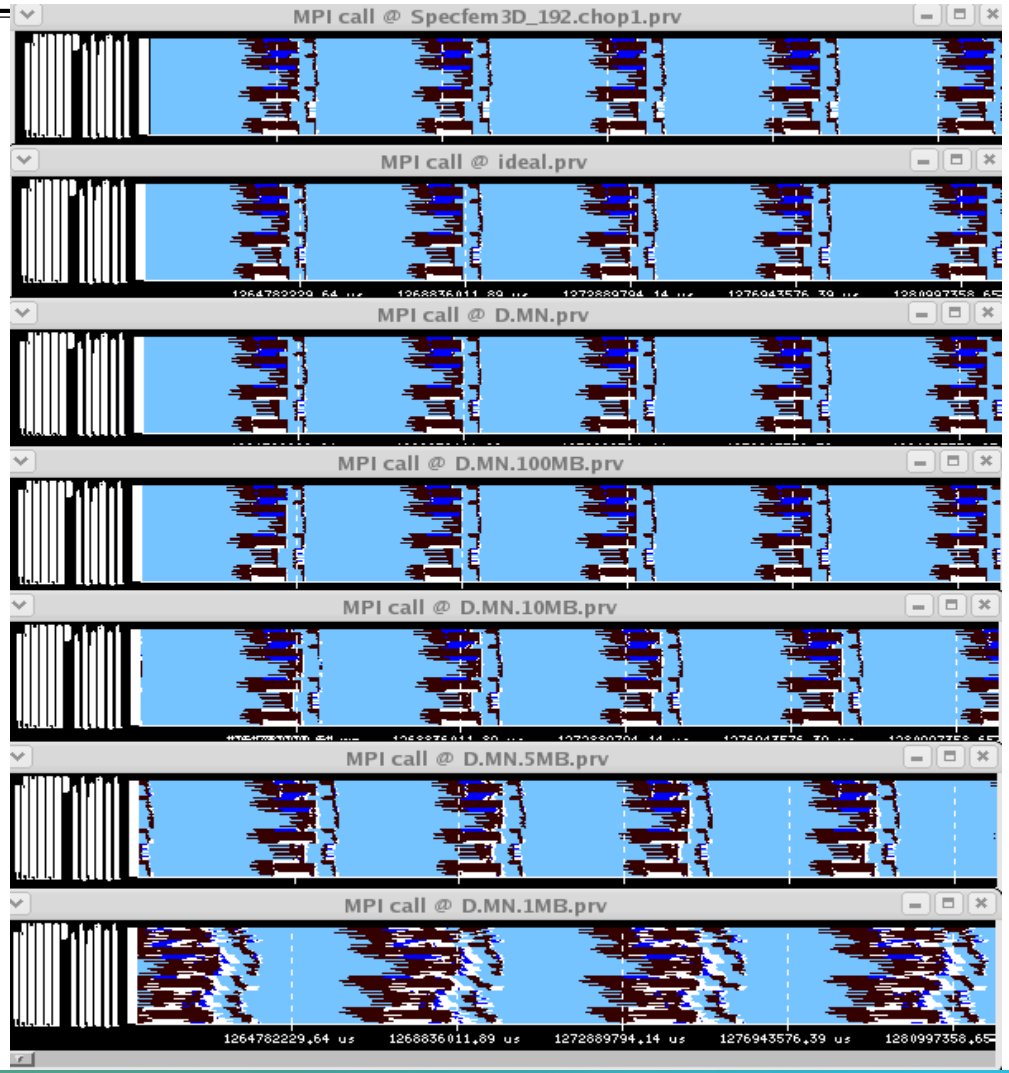


What if we had asynchronous communications?

- SPECFEM3D



Courtesy Dimitri Komatitsch



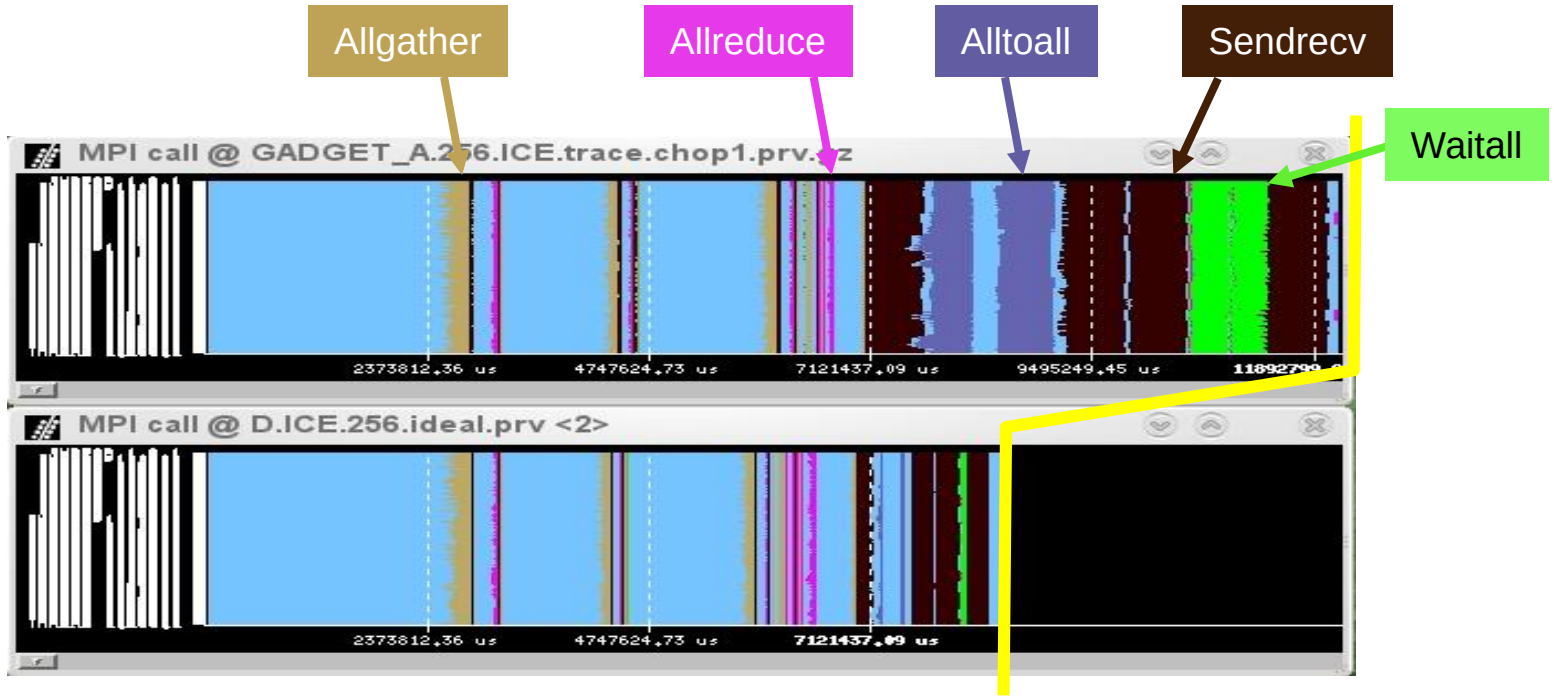
Ideal machine

- The impossible machine: BW = ∞, L = 0
  - Actually describes/characterizes Intrinsic application behavior
    - Load balance problems?
    - Dependencies problems?

GADGET @ Nehalem
256 processes

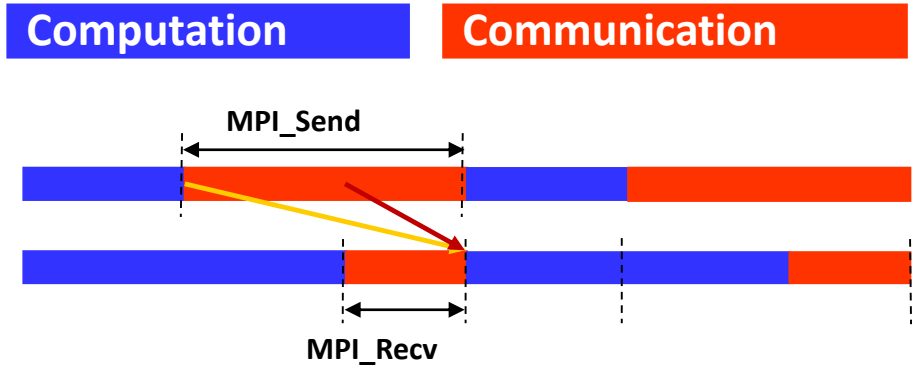
Real run

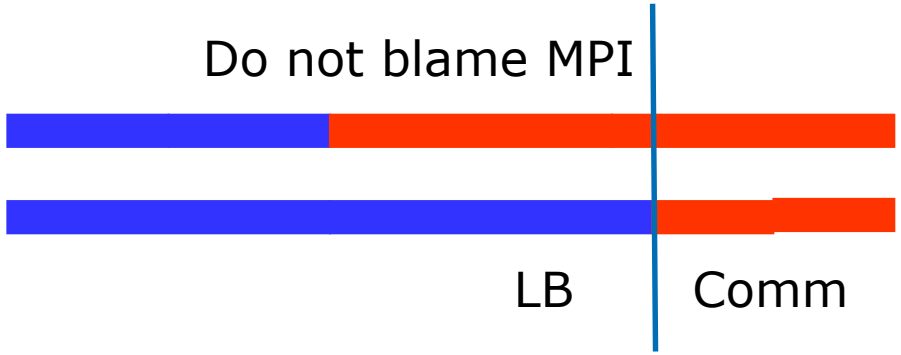
Ideal network



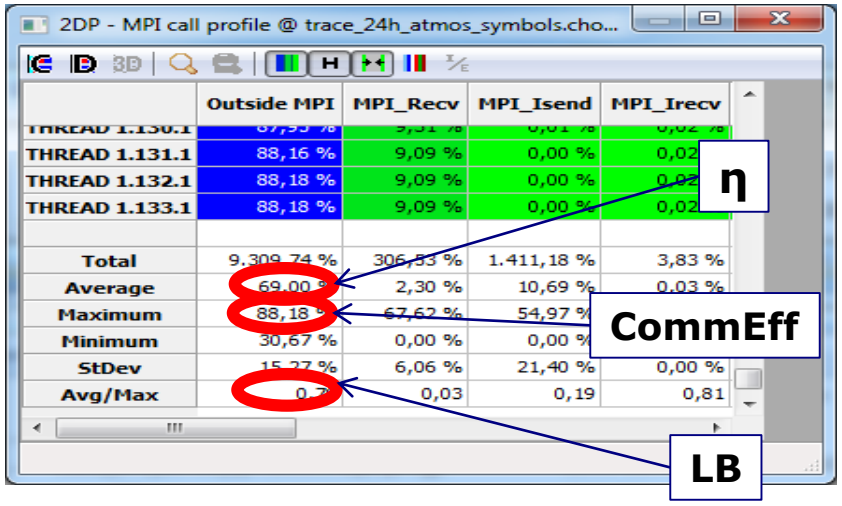
Models

Parallel efficiency model

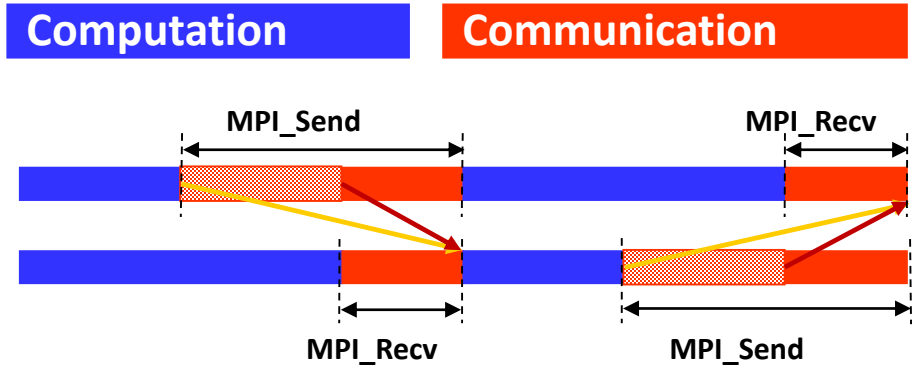


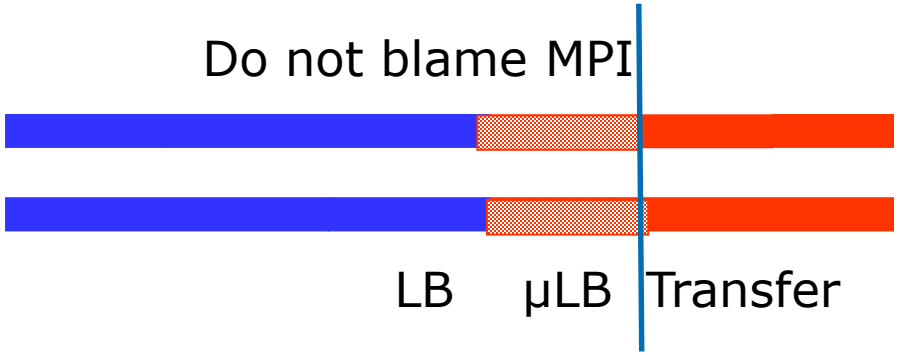


- Parallel efficiency = LB eff * Comm eff



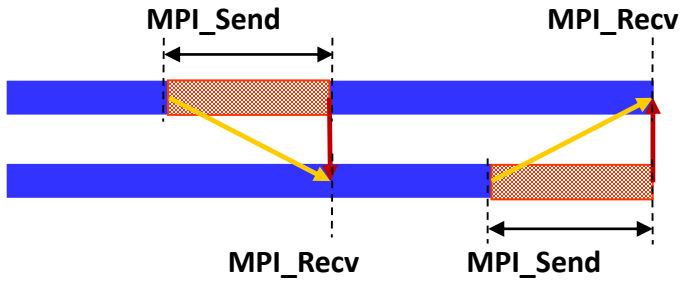
Parallel efficiency refinement: LB * µLB * Transfer





$$LB=1$$

- Serializations / dependences (µLB)
- Dimemas ideal network → Transfer (efficiency) = 1

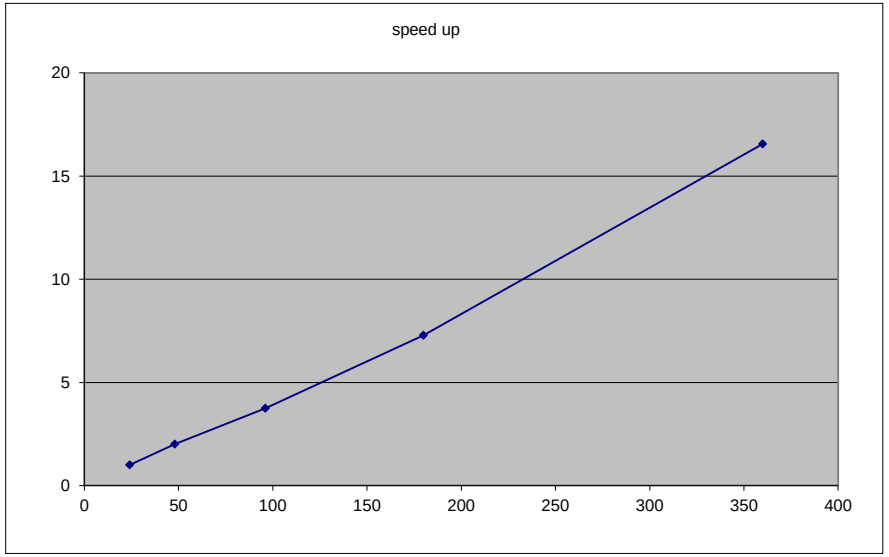


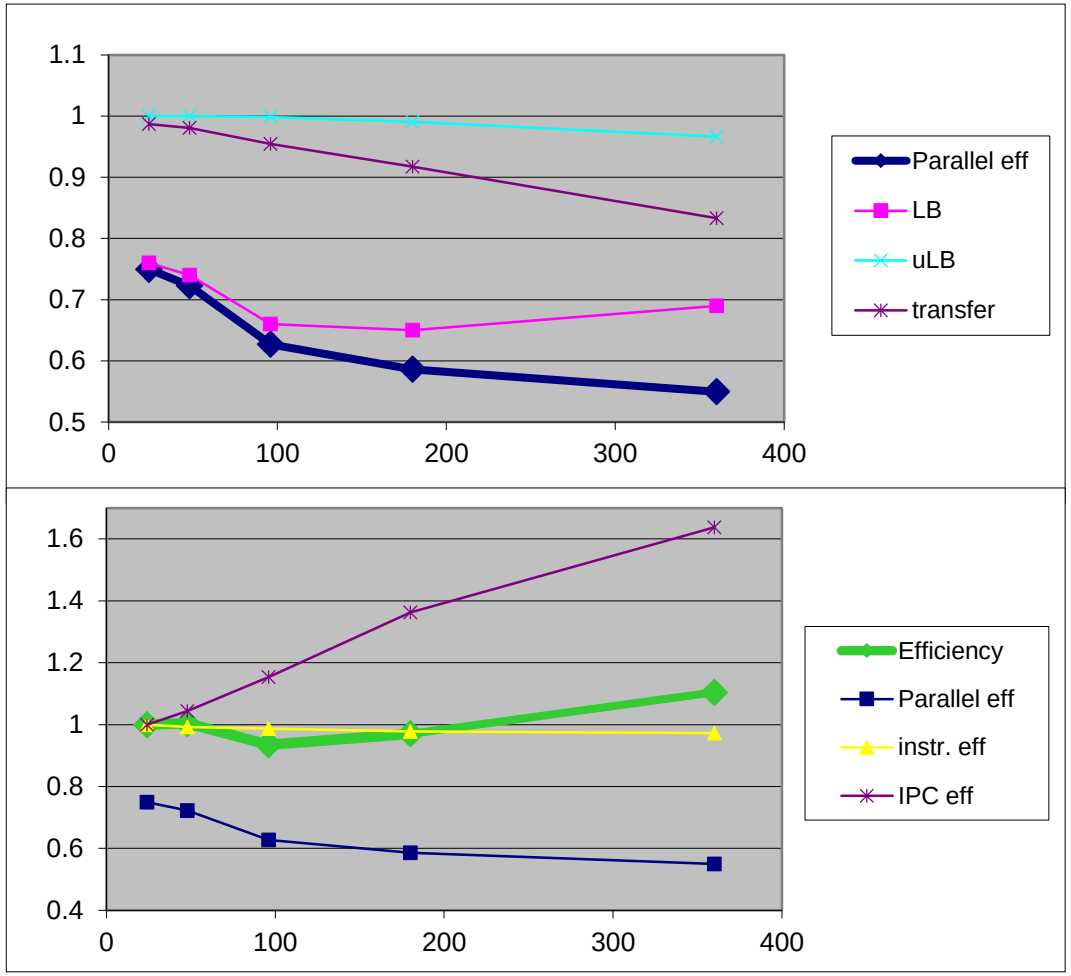
Why scaling?

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

CG-POP mpi2s1D - 180x120

Good scalability !!
Should we be happy?



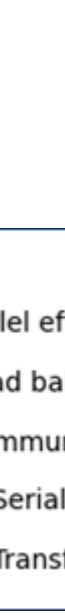


Analytics

Basic Analysis

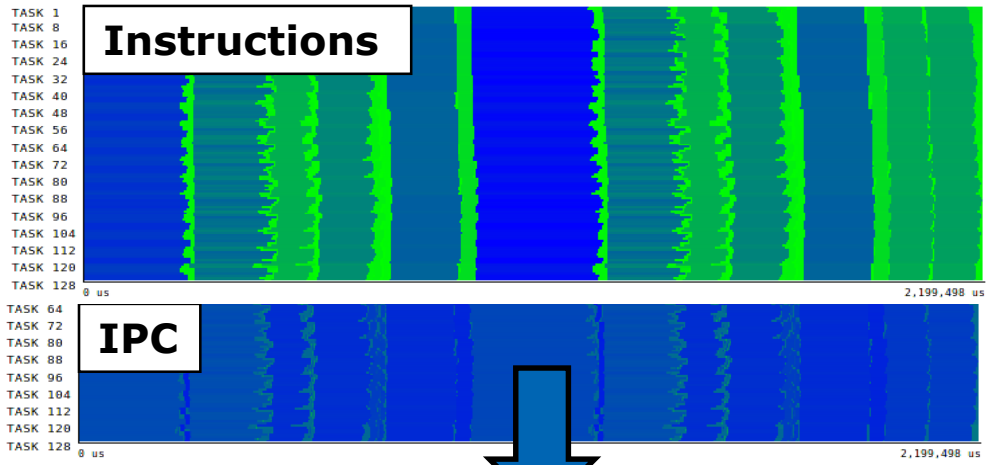
- Scripts that automatically compute the efficiency model from a Paraver trace (or many traces for scalability studies)
- Dig down from global to detailed efficiencies

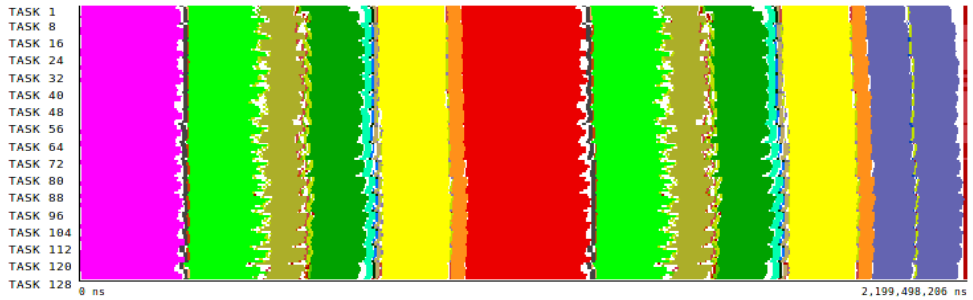
	256	512	1024
Global efficiency	84.32	83.48	84.28
-- Parallel efficiency	84.32	82.79	83.15
-- Load balance	93.51	93.33	93.30
-- Communication efficiency	90.17	88.71	89.11
-- Serialization efficiency	90.79	89.26	89.76
-- Transfer efficiency	99.32	99.38	99.28
-- Computation scalability	100.00	100.83	101.36
-- IPC scalability	100.00	101.02	101.43
-- Instruction scalability	100.00	99.91	99.86
-- Frequency scalability	100.00	99.90	100.07

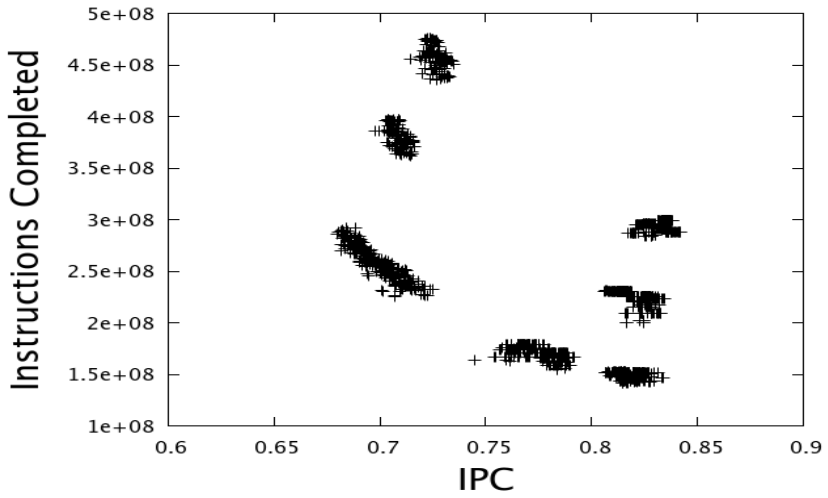


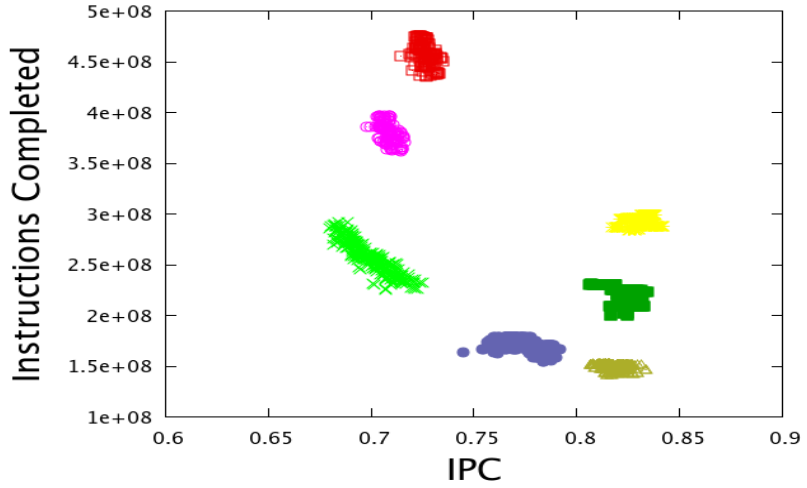
	512[1]	512[2]	512[3]	512[4]
Parallel efficiency	85.27	44.77	81.89	72.25
-- Load balance	86.08	88.13	82.82	74.97
-- Communication efficiency	99.05	50.80	98.88	96.38
-- Serialization efficiency	99.19	51.07	98.89	97.68
-- Transfer efficiency	99.86	99.47	99.99	98.66

Using Clustering to identify structure



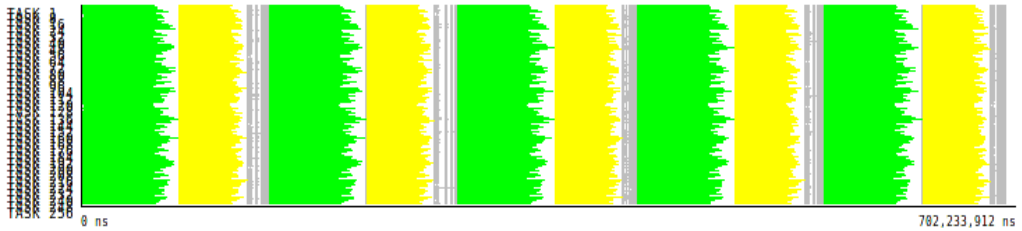






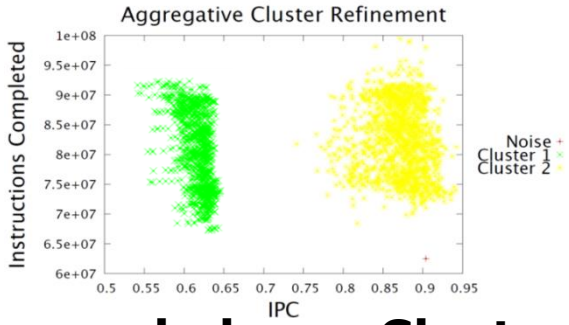
Integrating models and analytics

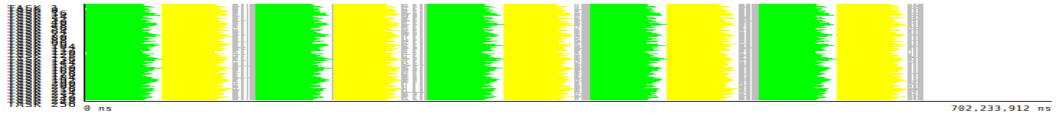
What if ….



PEPC

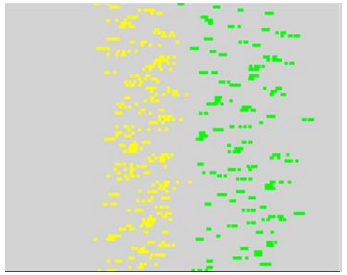
… we increase the IPC of Cluster1?





13% gain

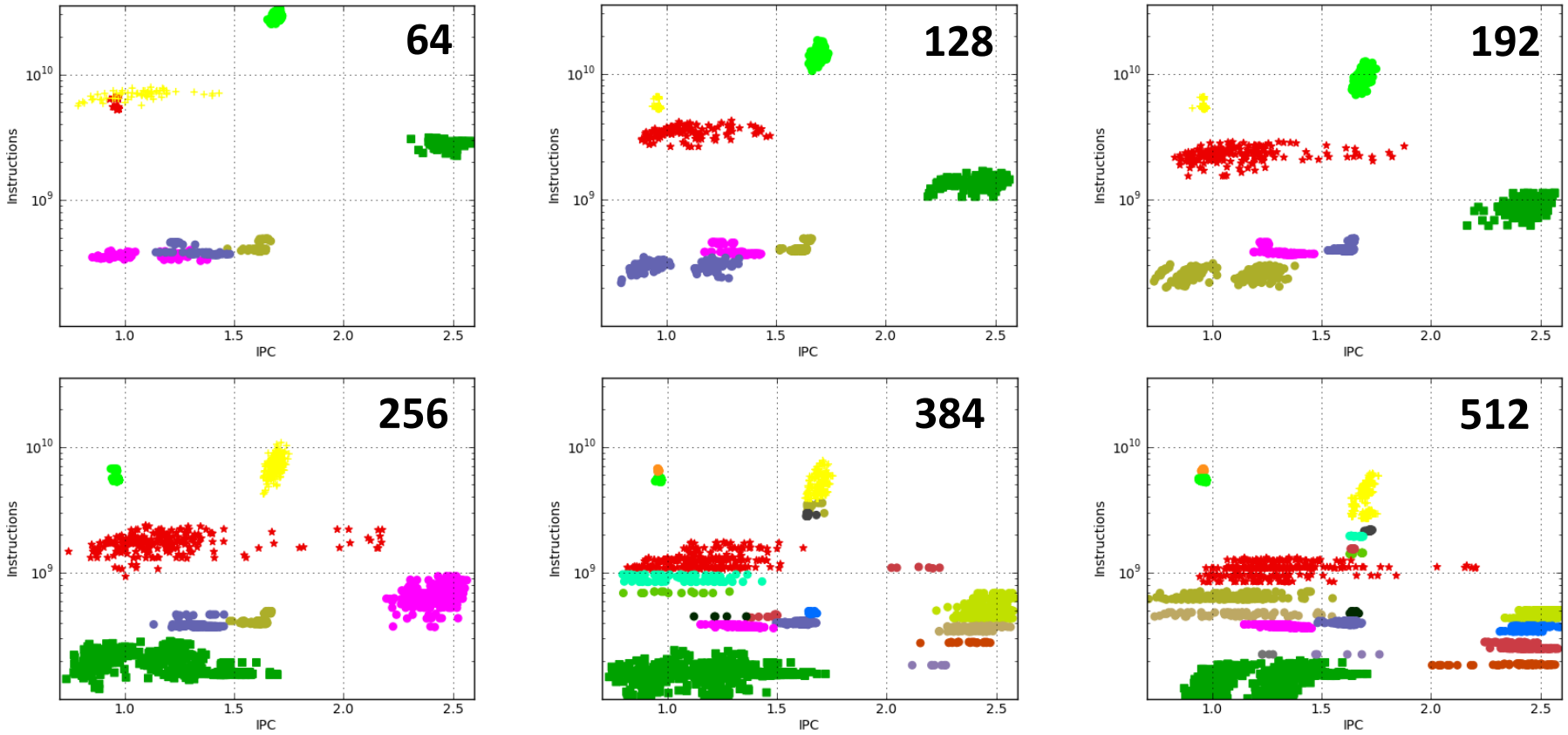
… we balance Clusters 1 & 2?



19% gain

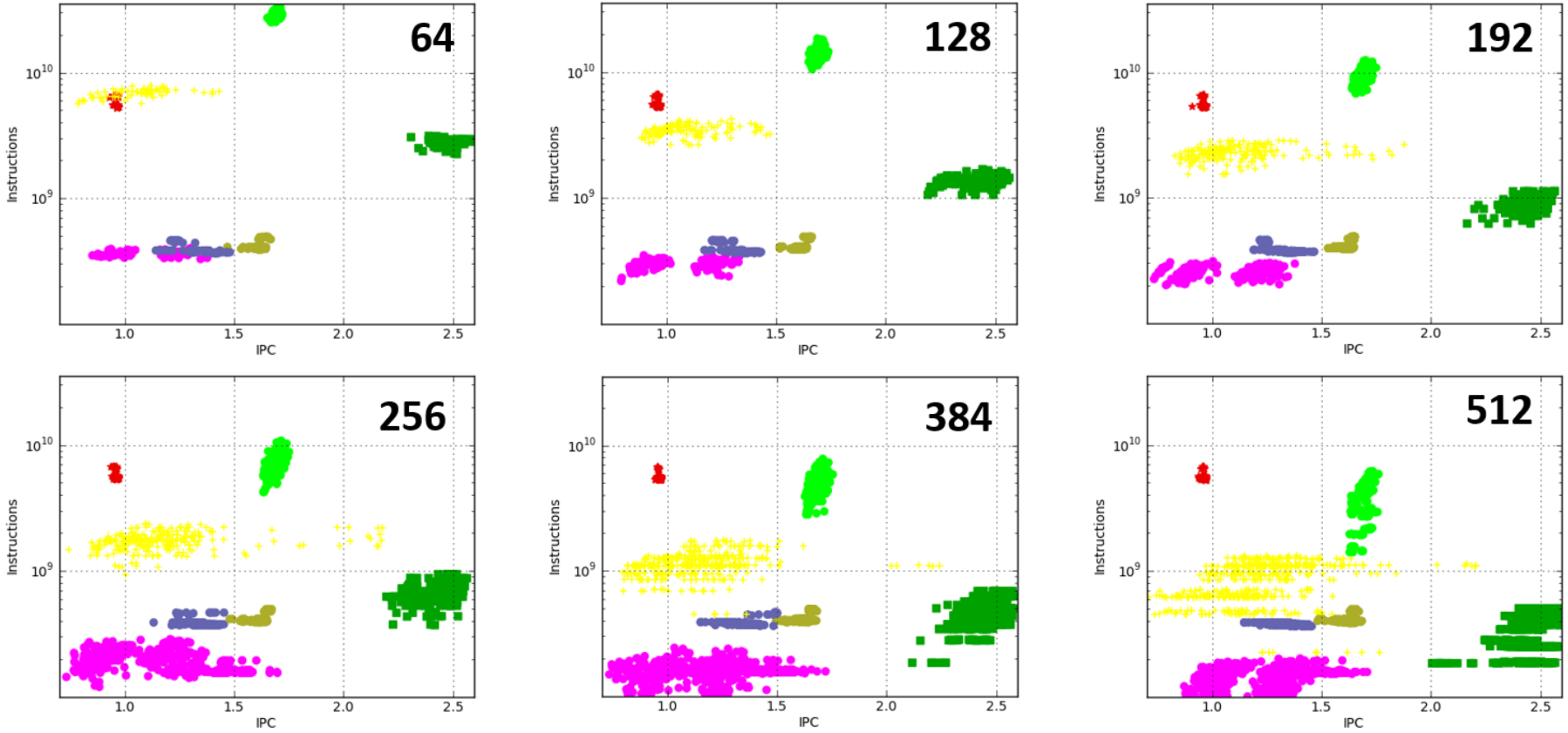
Tracking scalability through clustering

- Study the scalability of the computing regions
  - Increasing the scale from 64 to 512 tasks



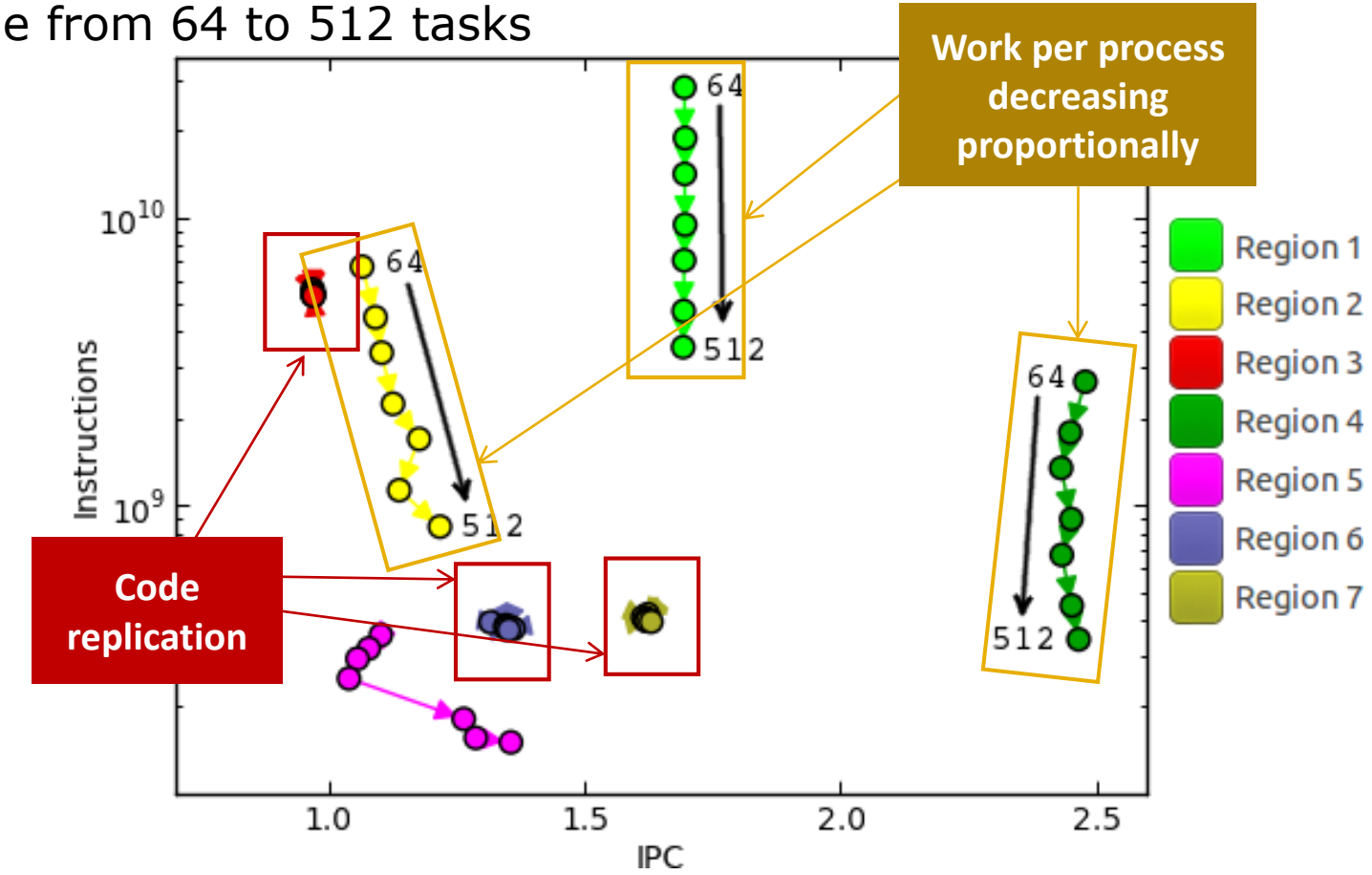
Tracking scalability through clustering

- Study the scalability of the computing regions
  - Increasing the scale from 64 to 512 tasks



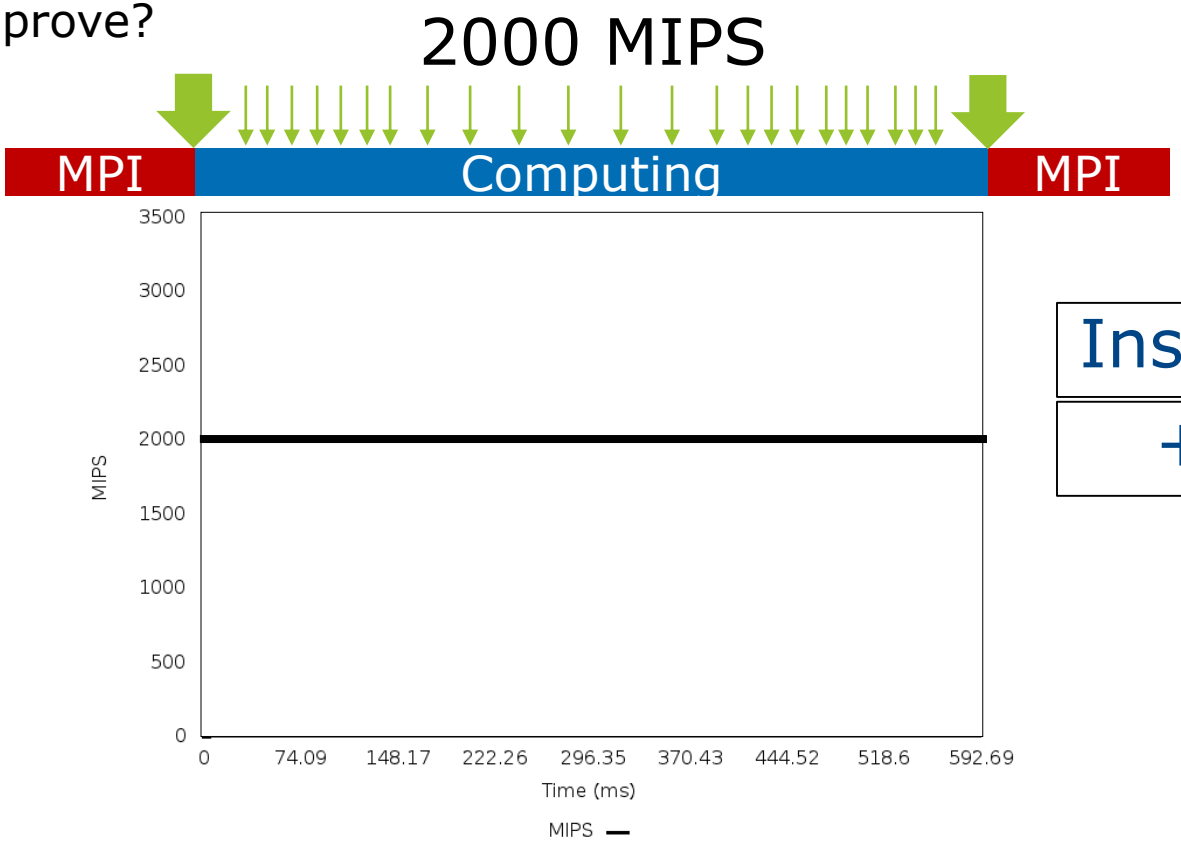
Tracking scalability through clustering

- Study the scalability of the computing regions
  - Increasing the scale from 64 to 512 tasks



Folding to increase details

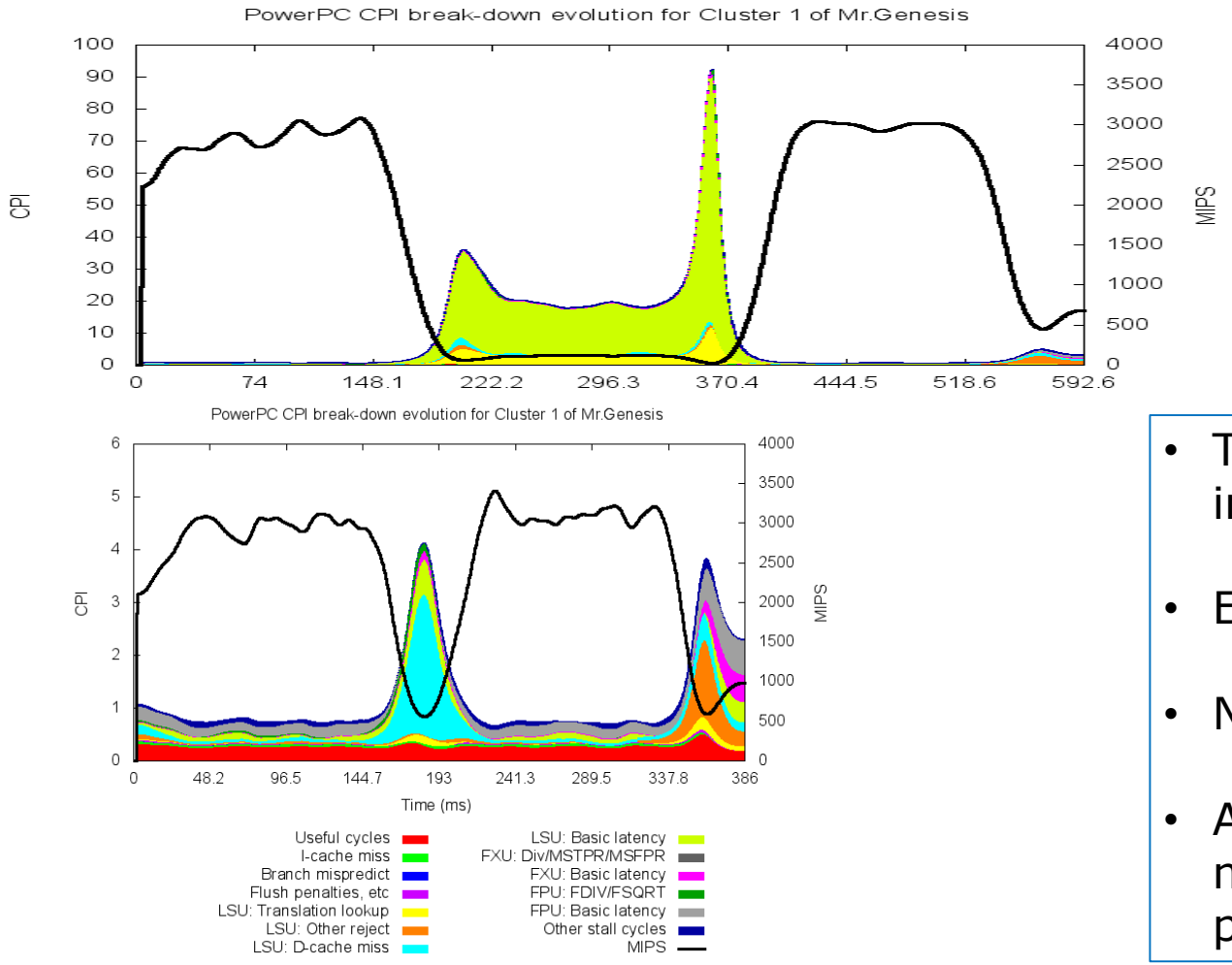
- What is the performance of a serial region?
  - Is it good enough?
  - Is it easy to improve?



Instrumentation

+ Sampling

Folding: CPI and HWC stack models



MR-GENESIS

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

Methodology

Performance analysis tools objective

Help generate hypotheses

Help validate hypotheses

Qualitatively

Quantitatively

First steps

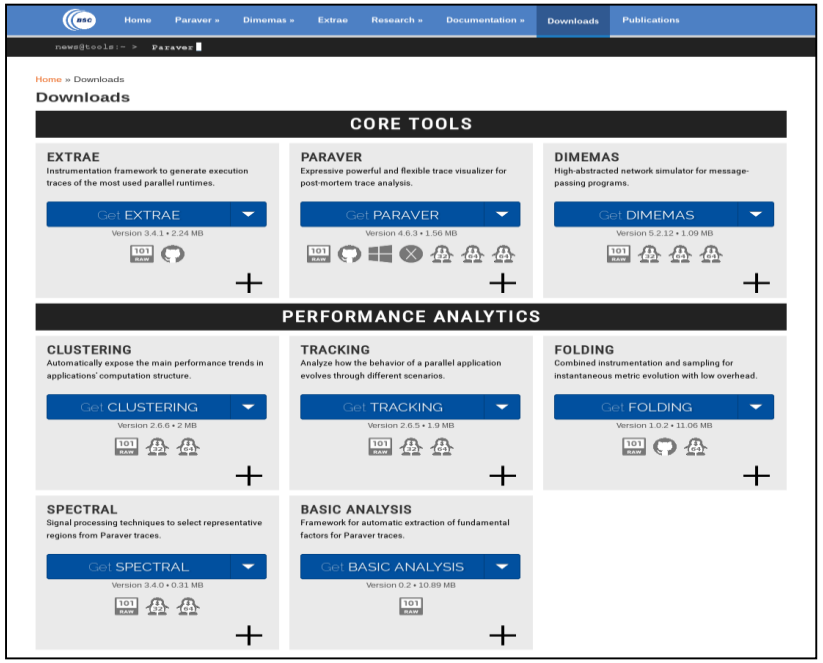
- Parallel efficiency – percentage of time invested on computation
  - Identify sources for "inefficiency":
    - Load Balance
    - Communication /synchronization
- Serial efficiency – how far from peak performance?
  - IPC, correlate with other counters
- Scalability – code replication?
  - Total instructions
- Behavioural structure? Variability?

Tutorial:
Introduction to Paraver & Dimemas methodology

BSC Tools web site

- https://tools.bsc.es
  - Downloads
    - Sources / Binaries

  - Documentation
    - Training guides
    - Tutorial slides



- Getting started
  - Start wxparaver
  - Help → Tutorials
  - Follow training guides
    - Paraver introduction (MPI): Navigation & Basic Understanding of Paraver operation

Takeaway:

- The importance of understanding
  → **Keep asking questions**
- Use your brain
  → **Use visual tools**
- The devil is in the details
  → **Do not miss them**
- Don’t over-theorize about your code
  → **Look at it open-minded**