Understanding applications using the BSC performance tools

Judit Gimenez (judit@bsc.es), German Llort

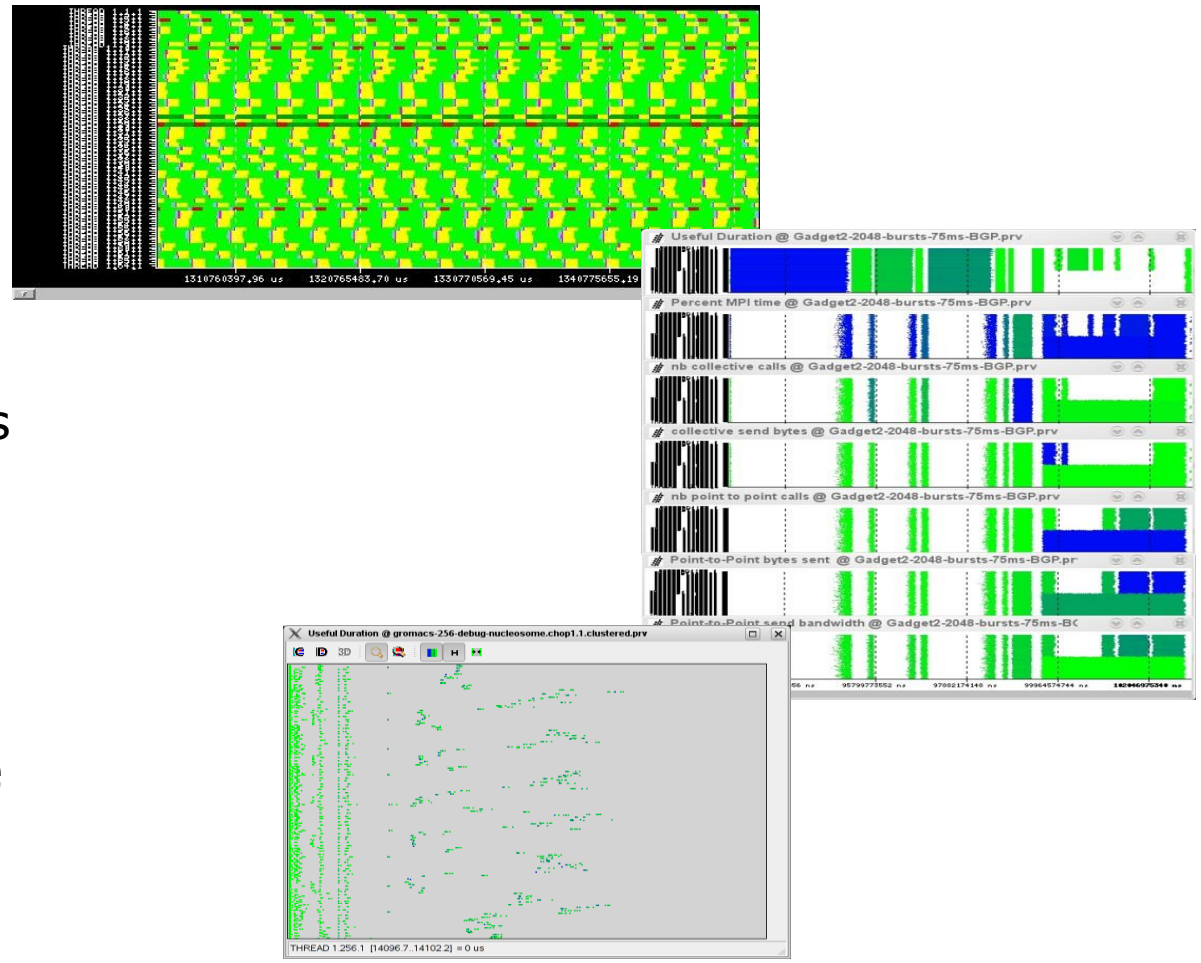
Barcelona Supercomputing Center

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?

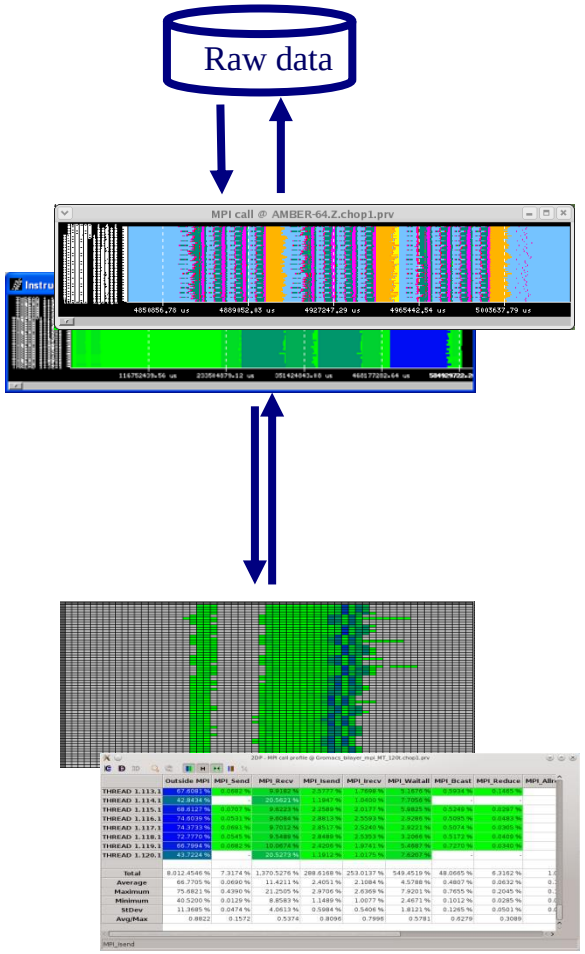
Our Tools

- 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, variability, flexibility
  - Behavioral 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

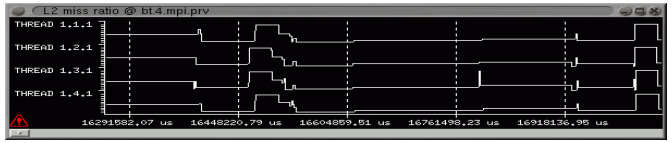
Comparative analyses

Multiple traces

Synchronize scales

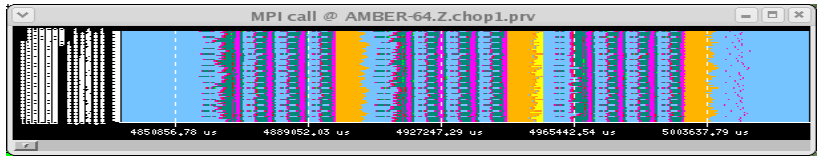
Timelines

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



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

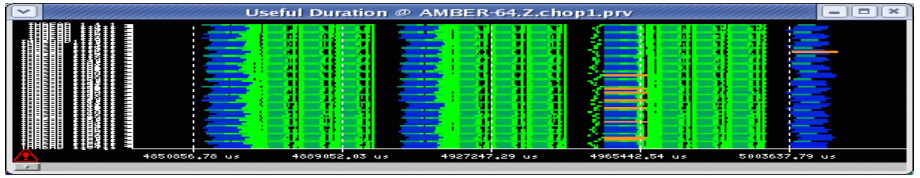
- Types of functions
  - Categorical
    - State, user function, outlined routine



$$S_i \in [0, n] \subset N, \quad n <$$

$$S_i \in \{0, 1\}$$

  - Logical
    - In specific user function, In MPI call, In long MPI call
  - Numerical
    - IPC, L2 miss ratio, Duration of MPI call, duration of computation burst



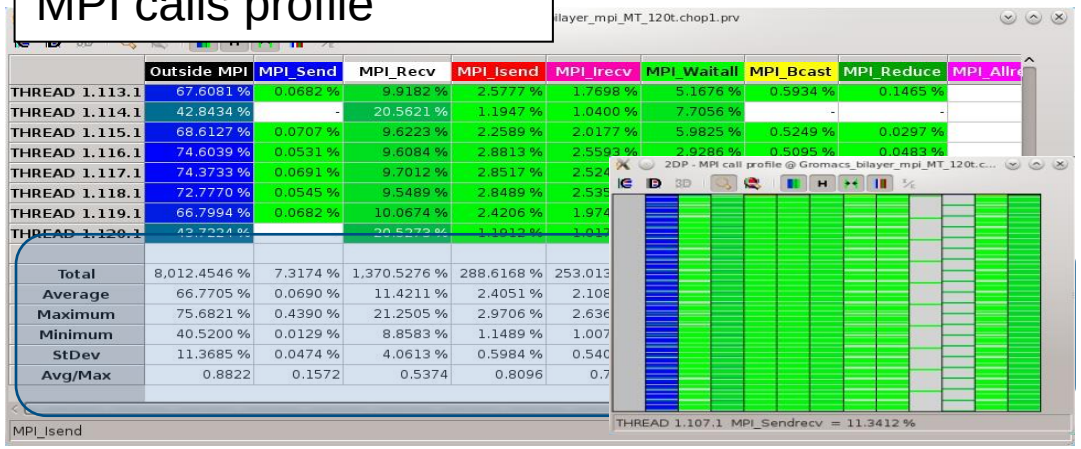
$$S_i \in R$$

Tables: Profiles, histograms, correlations

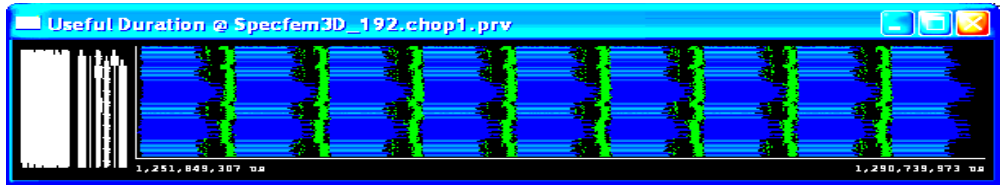
- From timelines to tables

MPI calls

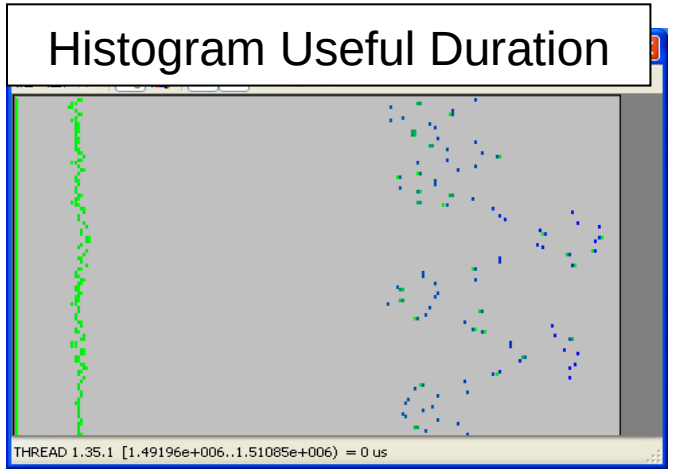
MPI calls profile



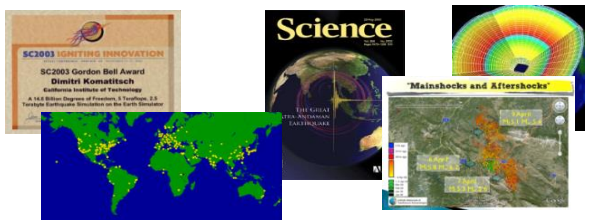
Useful Duration



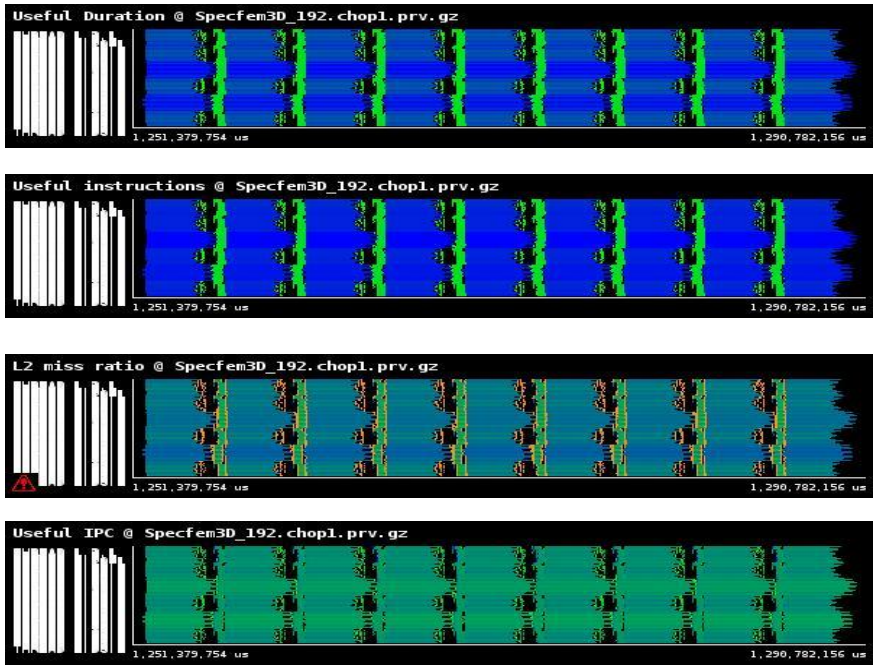
Histogram Useful Duration

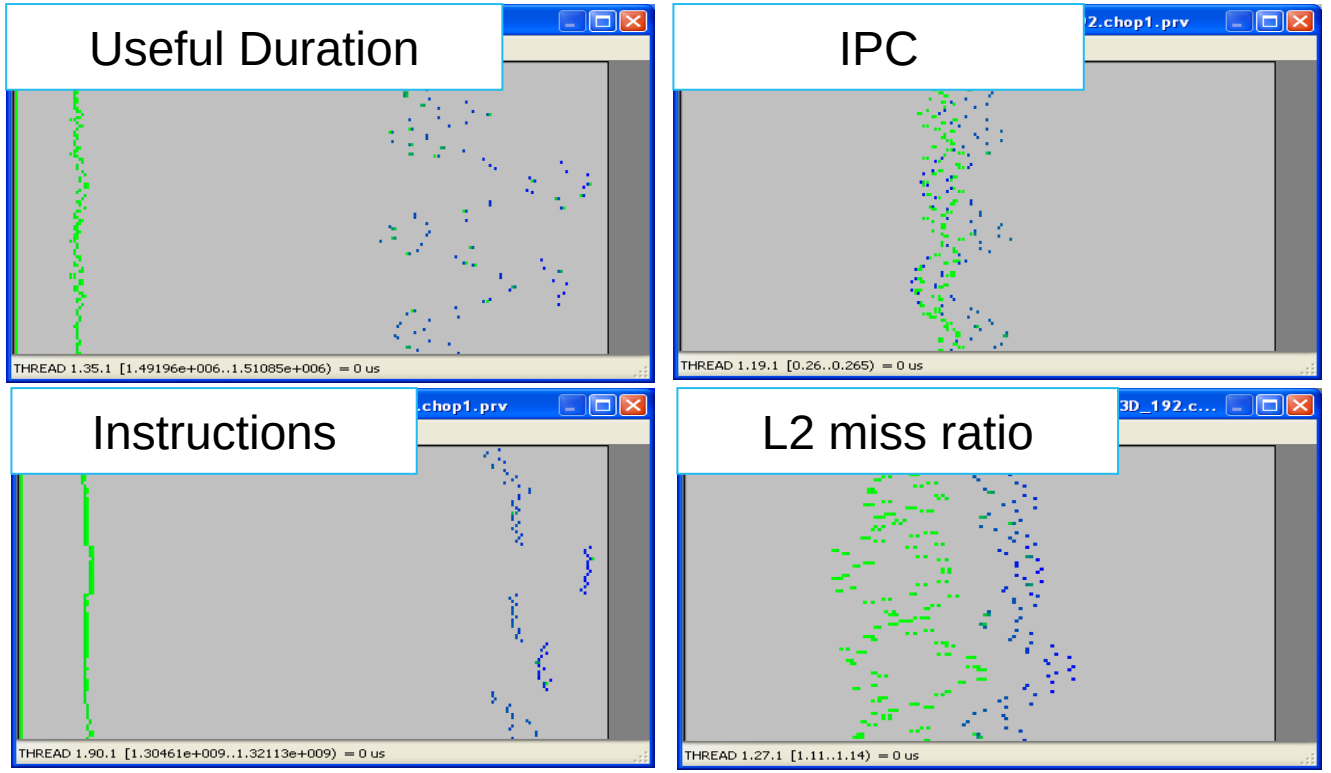


Analyzing variability through histograms and timelines



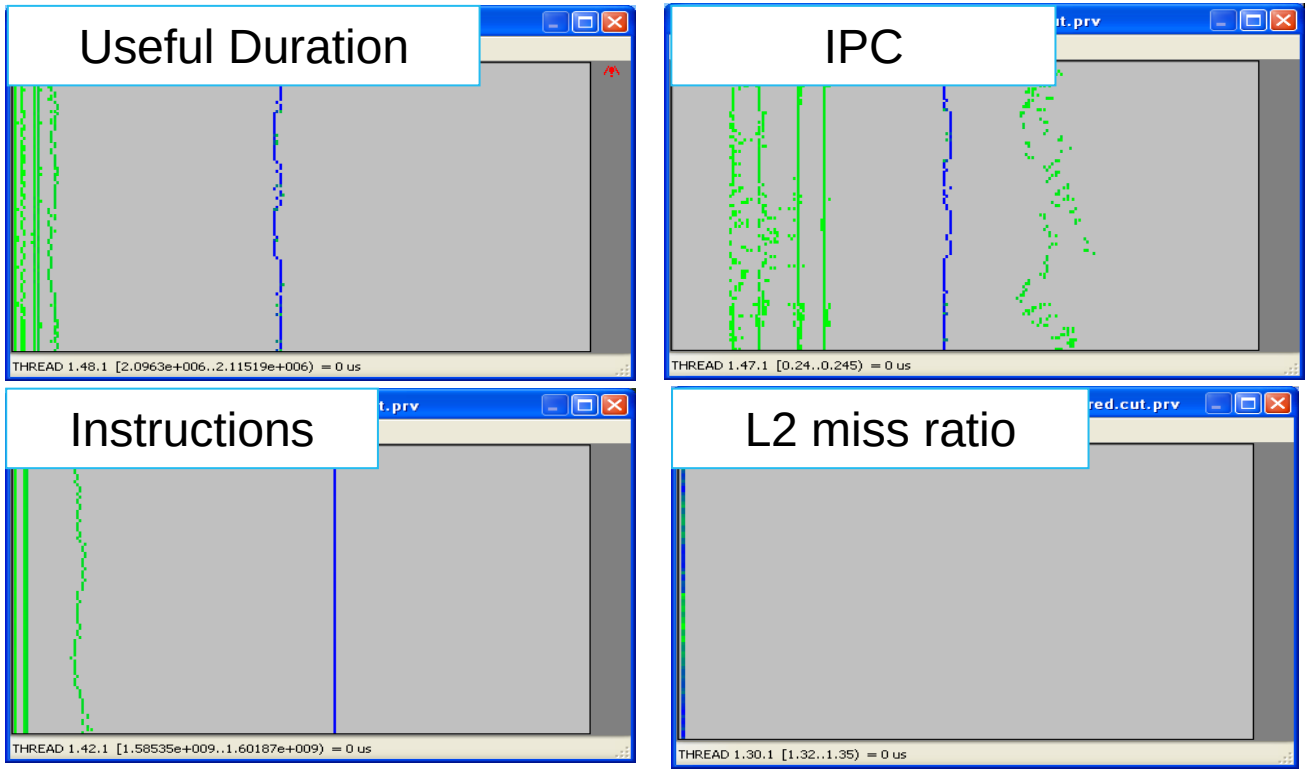
SPECFEM3D





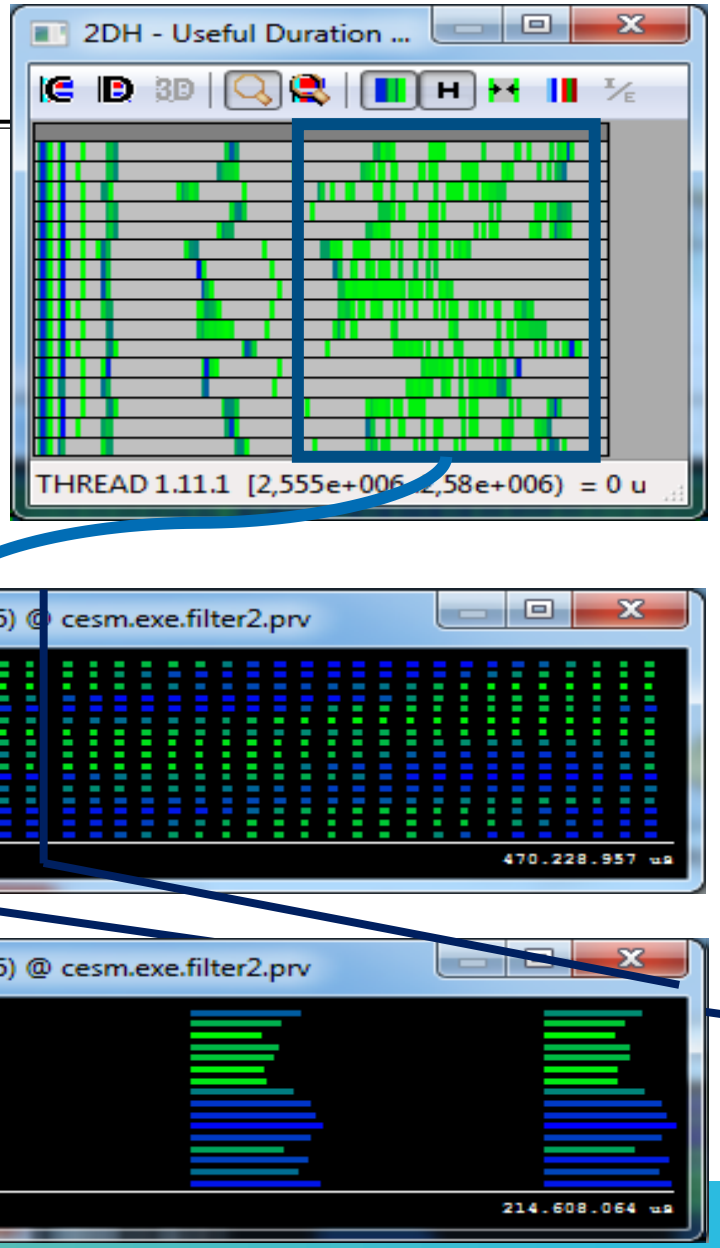
Analyzing variability through histograms and timelines

- By the way: six months later ….



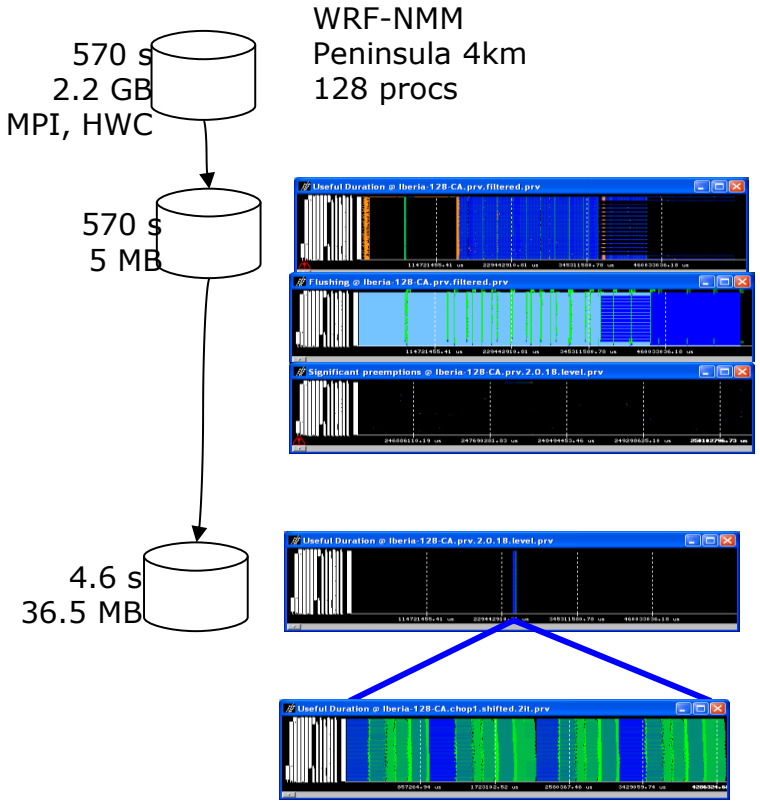
Variability … is everywhere

- 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 ? ☺



Trace manipulation

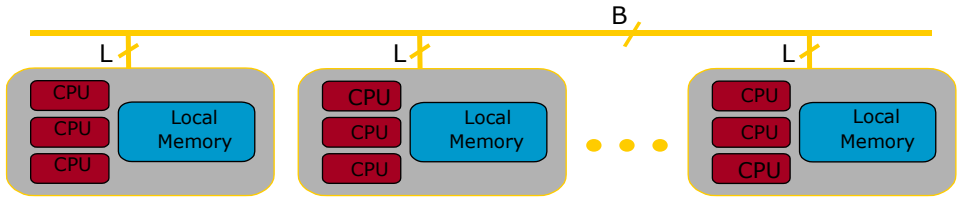
- Data handling/summarization capability
- Filtering
  - Subset of records in original trace
  - By duration, type, value,…
  - Filtered trace IS a paraver trace and can be analysed with the same cfgs (as long as needed data kept)
- Cutting
  - All records in a given time interval
  - Only some processes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

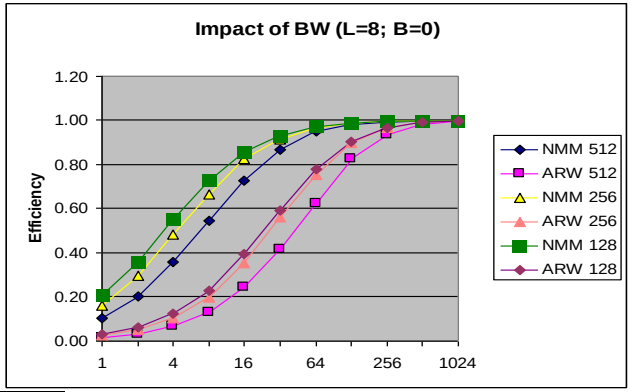


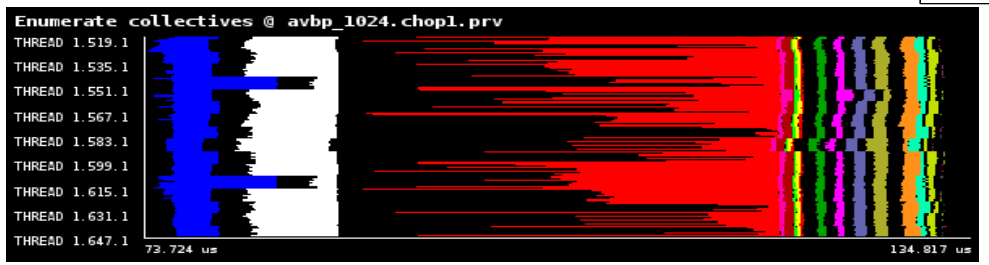
Dimemas

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
  - Modeling nominal
- Paraver – Dimemas tandem
  - Analysis and prediction
  - What-if from selected time window







4ms

Detailed feedback on simulation (trace)

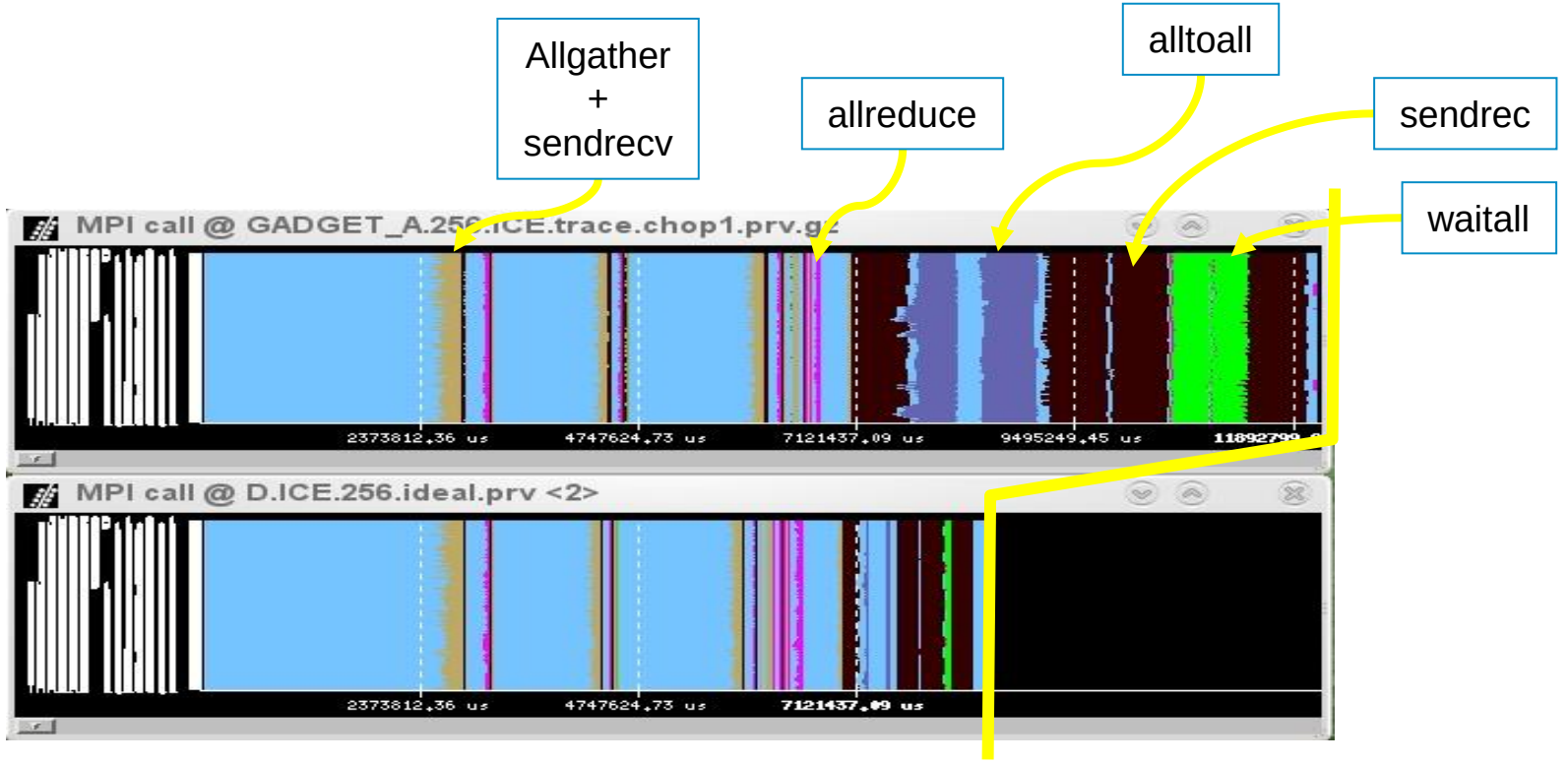
Ideal machine

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

GADGET @ Nehalem cluster
256 processes

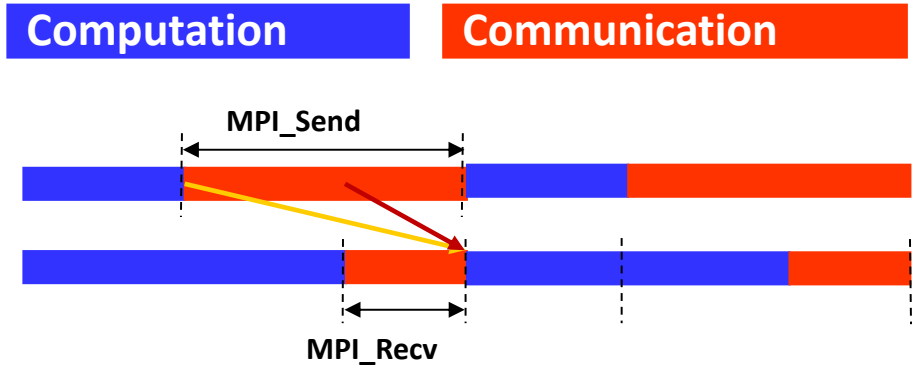
Real run

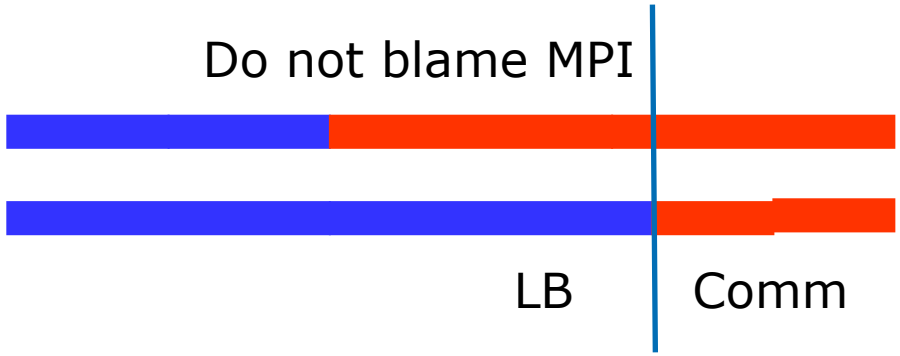
Ideal network



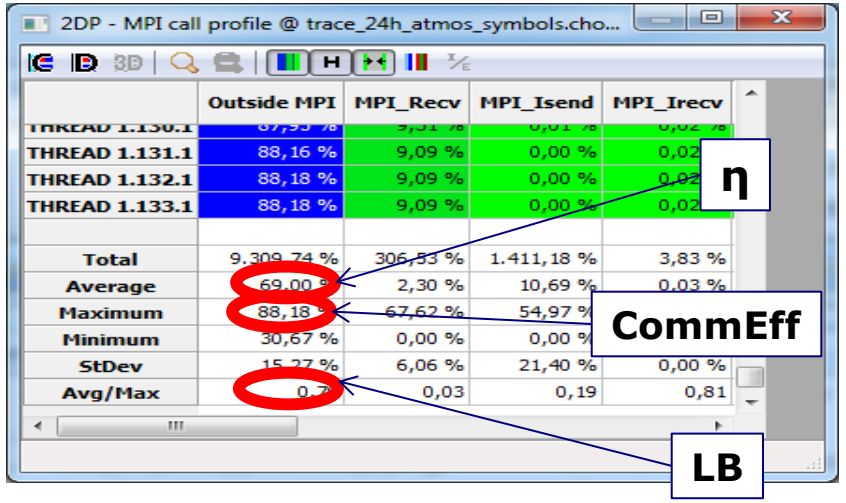
Models

Parallel efficiency model

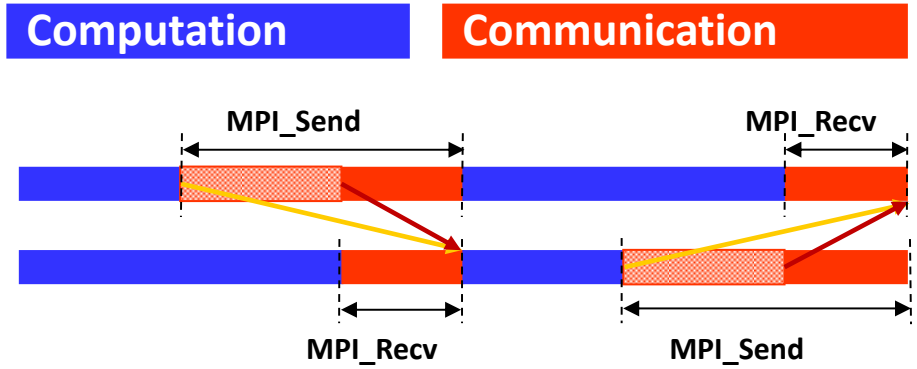


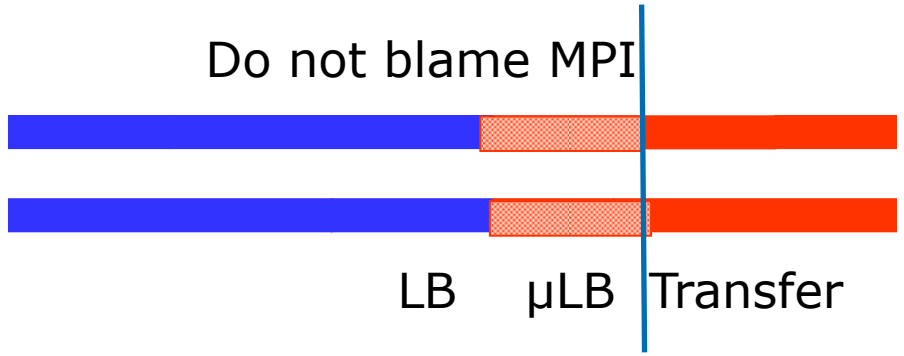


- Parallel efficiency = LB eff * Comm eff



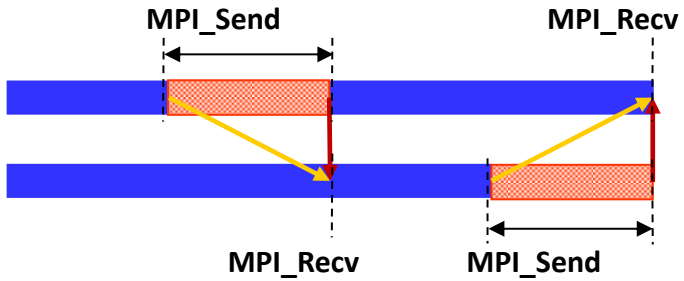
Parallel efficiency refinement: LB * µLB * Transfer





$$LB=1$$

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

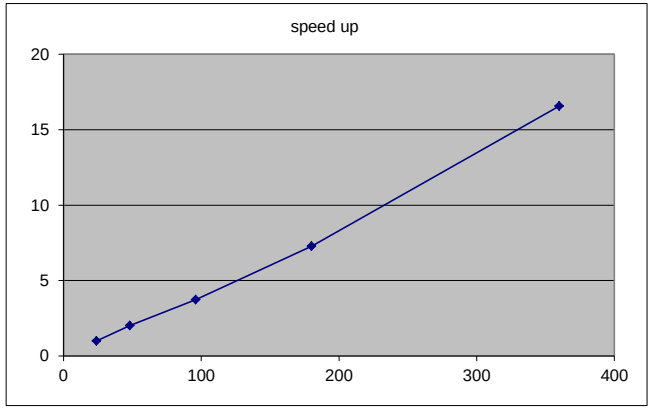


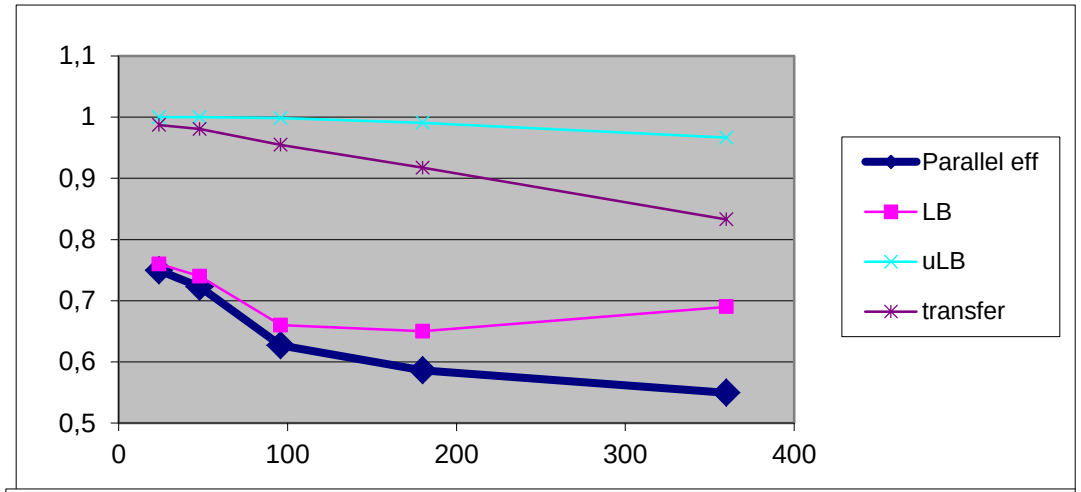
Why scaling?

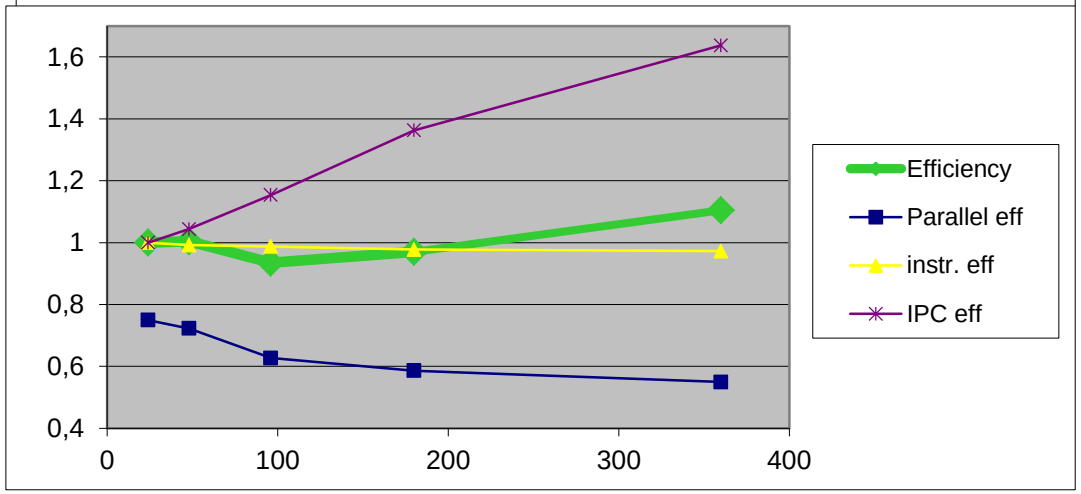
$$\eta_{\parallel} = LB * Ser * Trf$$

CG-POP mpi2s1D - 180x120

Good scalability !!
Should we be happy?

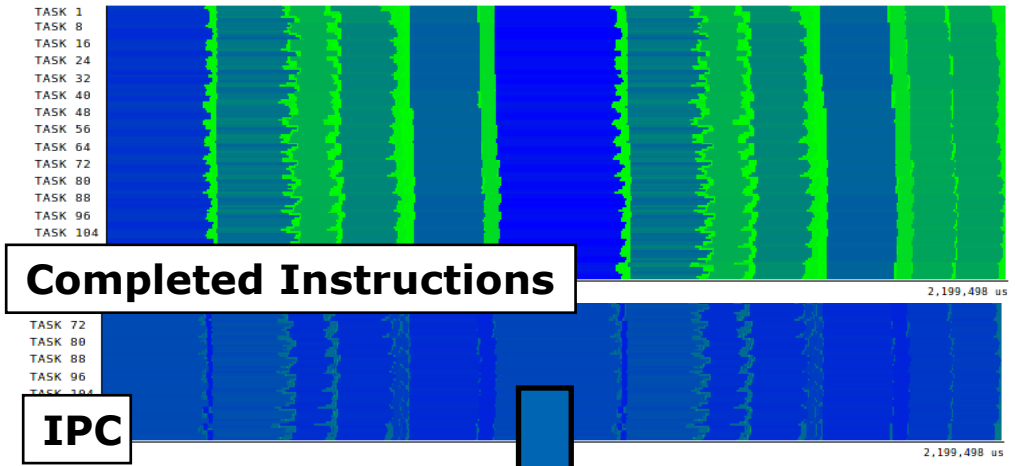


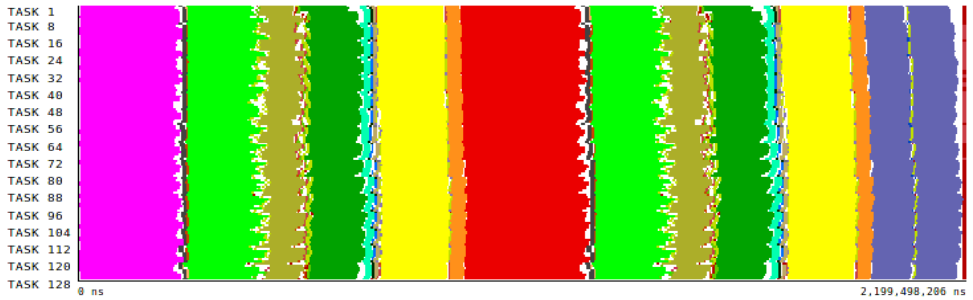


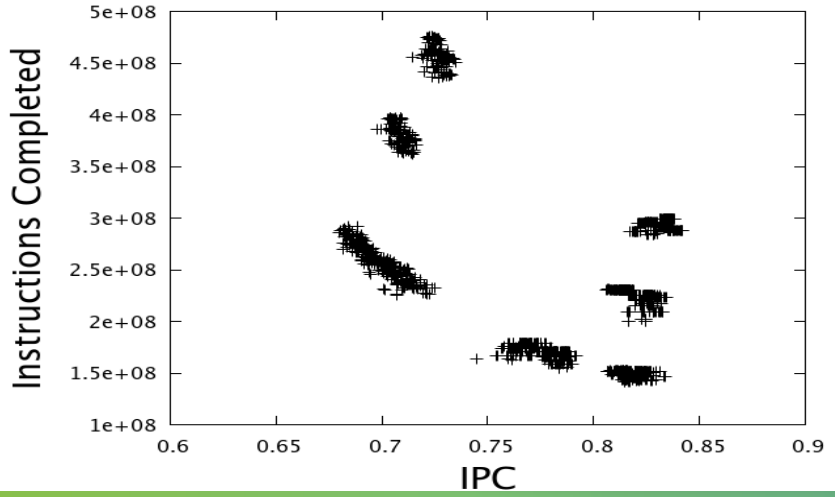


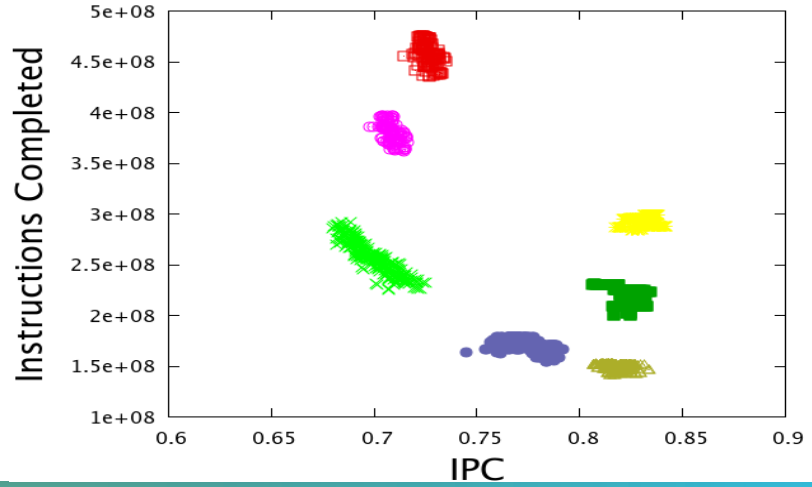
Peformance Analytics

Using Clustering to identify structure



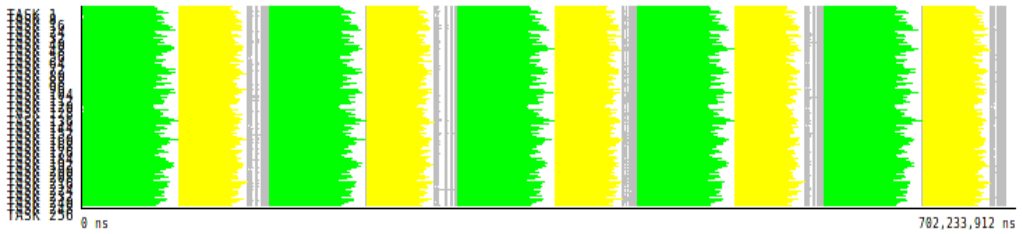






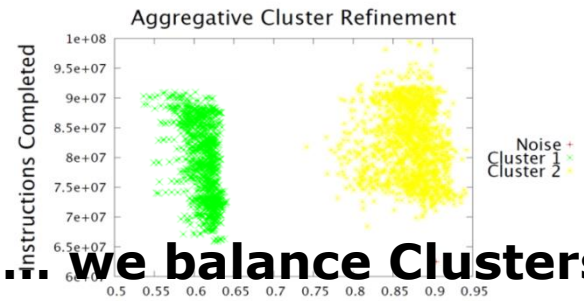
Integrating models and analytics

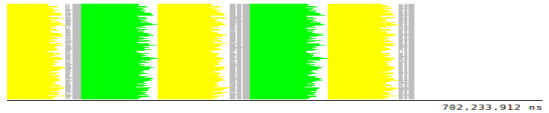
What if ….



PEPC

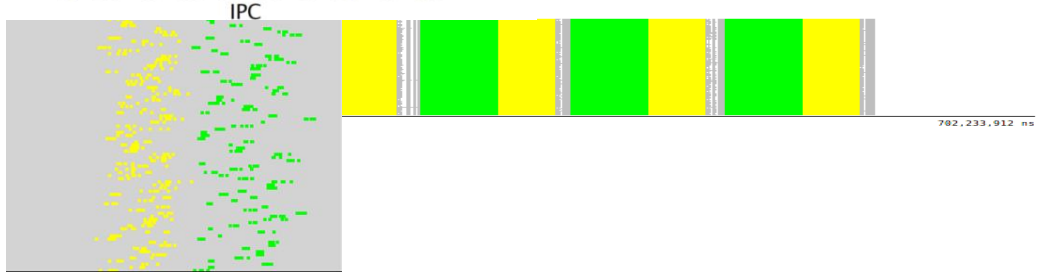
… we increase the IPC of Cluster1?





13% gain

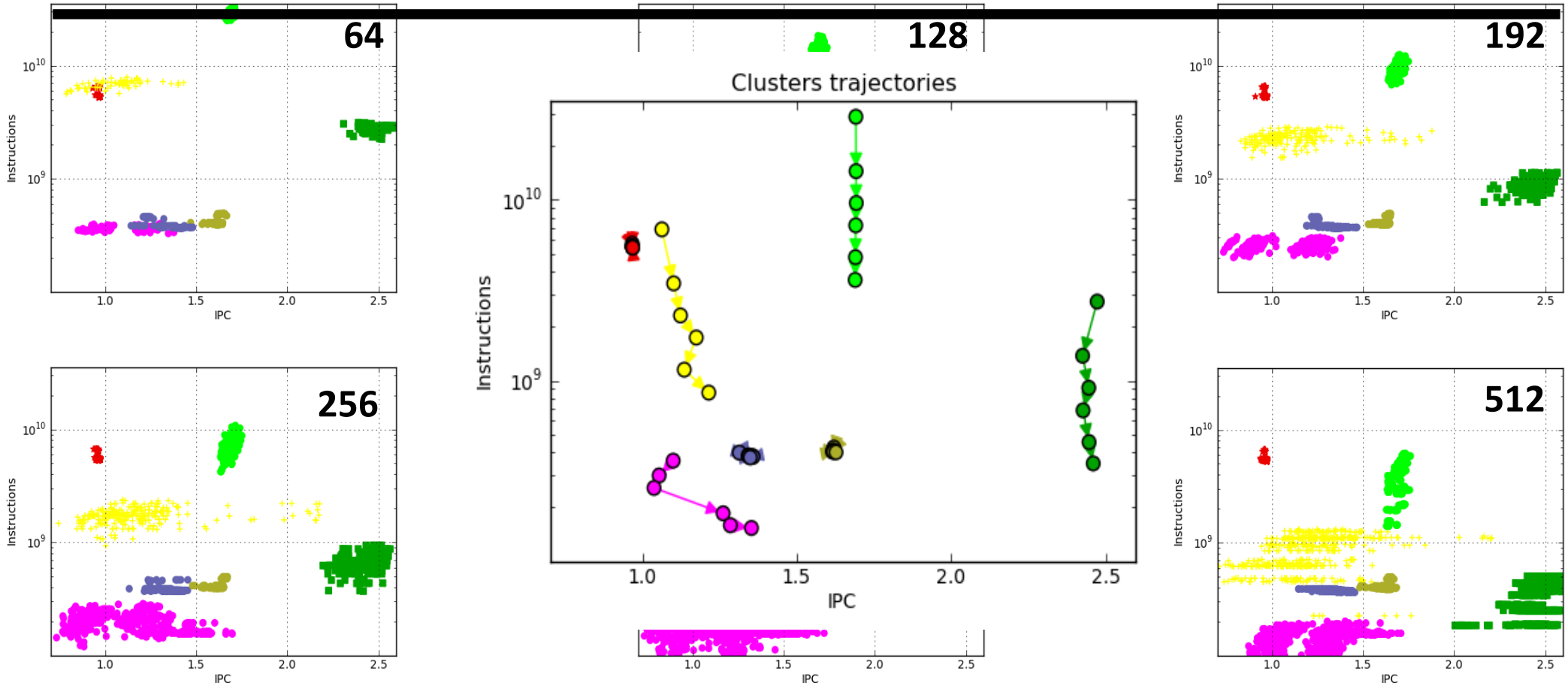
… we balance Clusters 1 & 2?



19% gain

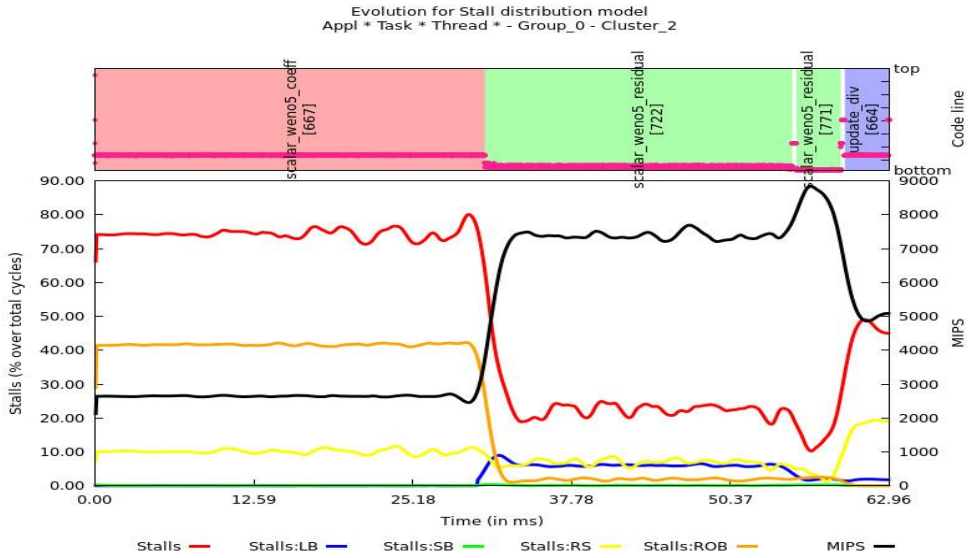
Tracking: scability through clustering

OpenMX (strong scale from 64 to 512 tasks)

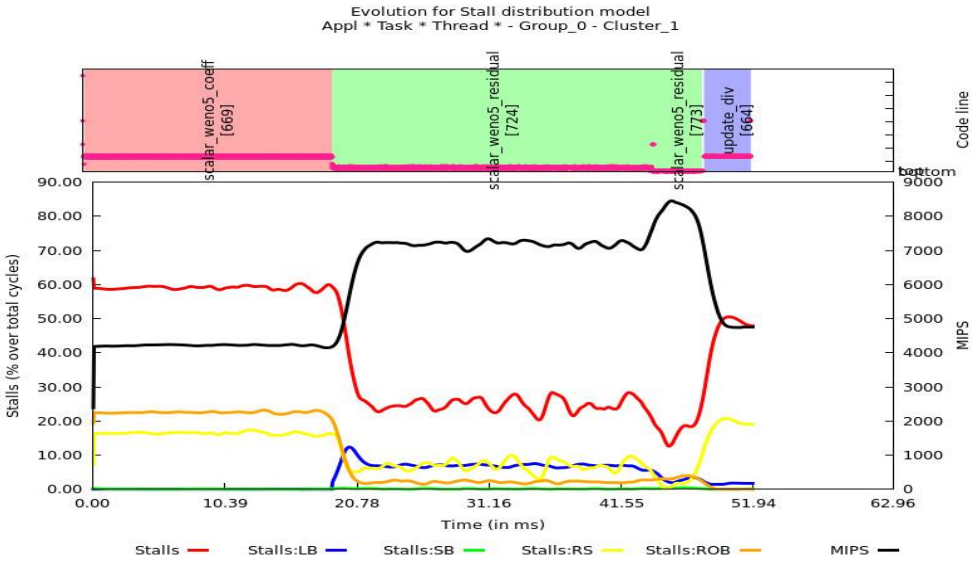


"Blind" optimization

- From folded samples of a few levels to timeline structure of "relevant" routines

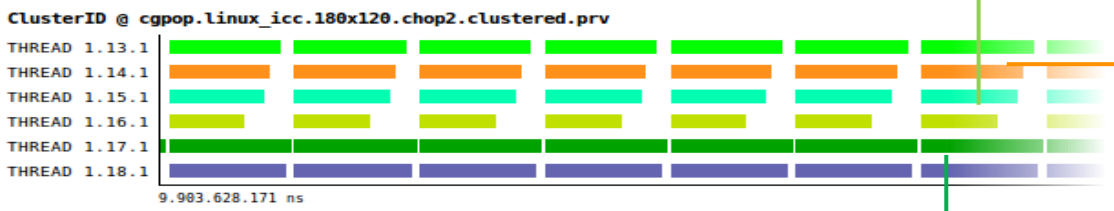


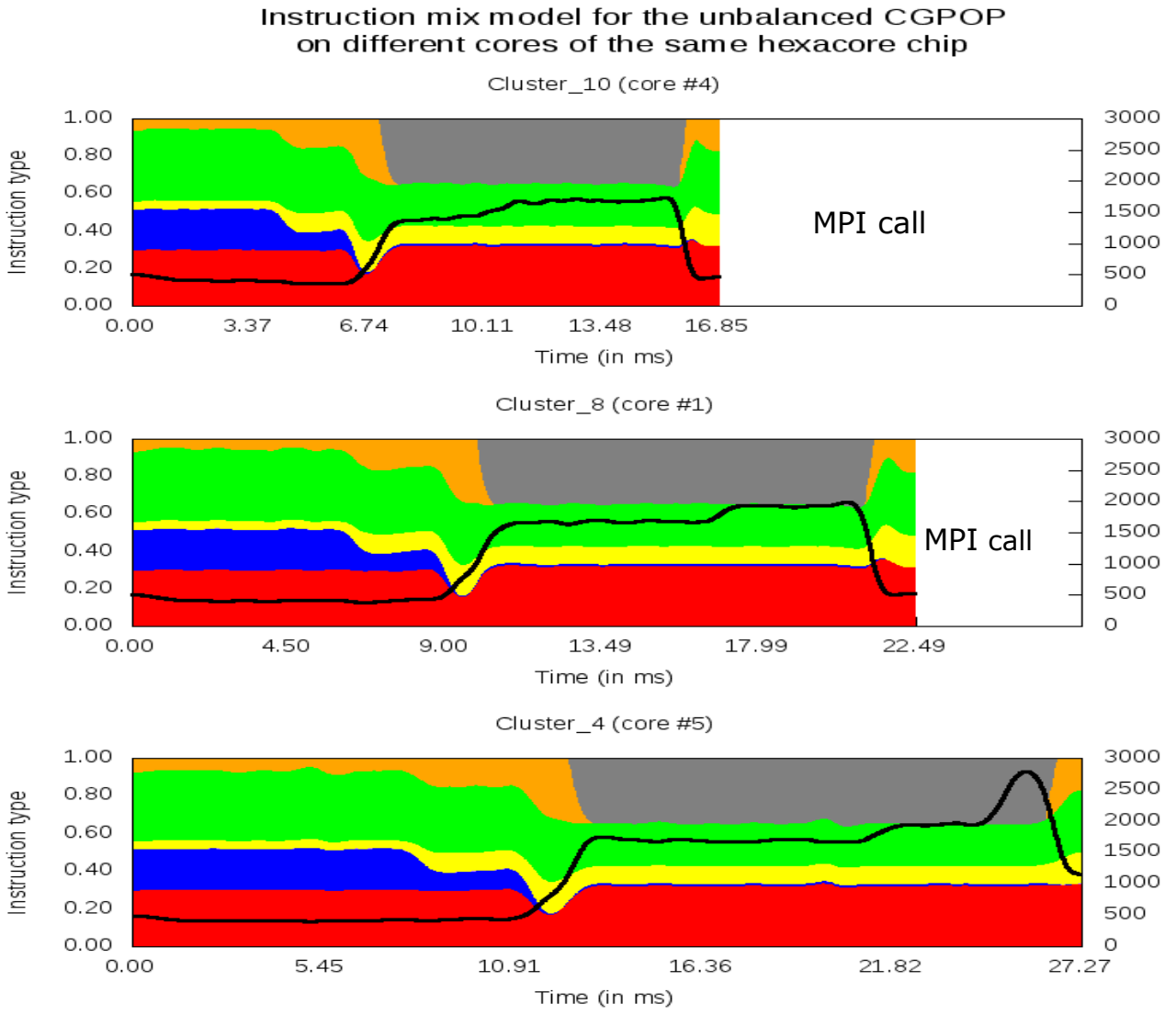
Recommendation without access to source code



Multicore effect

- CG-POP: Unbalanced MPI application
  - Same code
  - Different duration
  - Different performance





Methodology

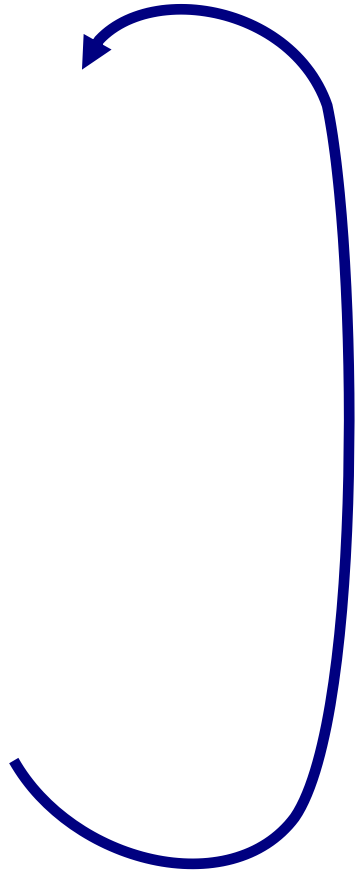
Performance analysis tools objective

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?

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

Paraver Demo

Some examples of efficiencies

Code	Parallel efficiency	Communication efficiency	Load Balance efficiency
Gromacs	66.77	75.68	88.22
BigDFT	59.64	78.97	75.52
CG-POP	80.98	98.92	81.86
ntchem_mini(taxol, 32)@pi	92.56	94.94	97.49
nicam@pi	87.10	75.97	89.22
lulesh@mn3	90.55	99.22	91.26
lulesh@leftraru	69.15	99.12	69.76
lulesh@uv2 (mpt)	70.55	96.56	73.06
lulesh@uv2 (impi)	85.65	95.09	90.07