Understanding applications using the BSC performance tools

Judit Gimenez (judit@bsc.es), Germán Llort, Lau Mercadal

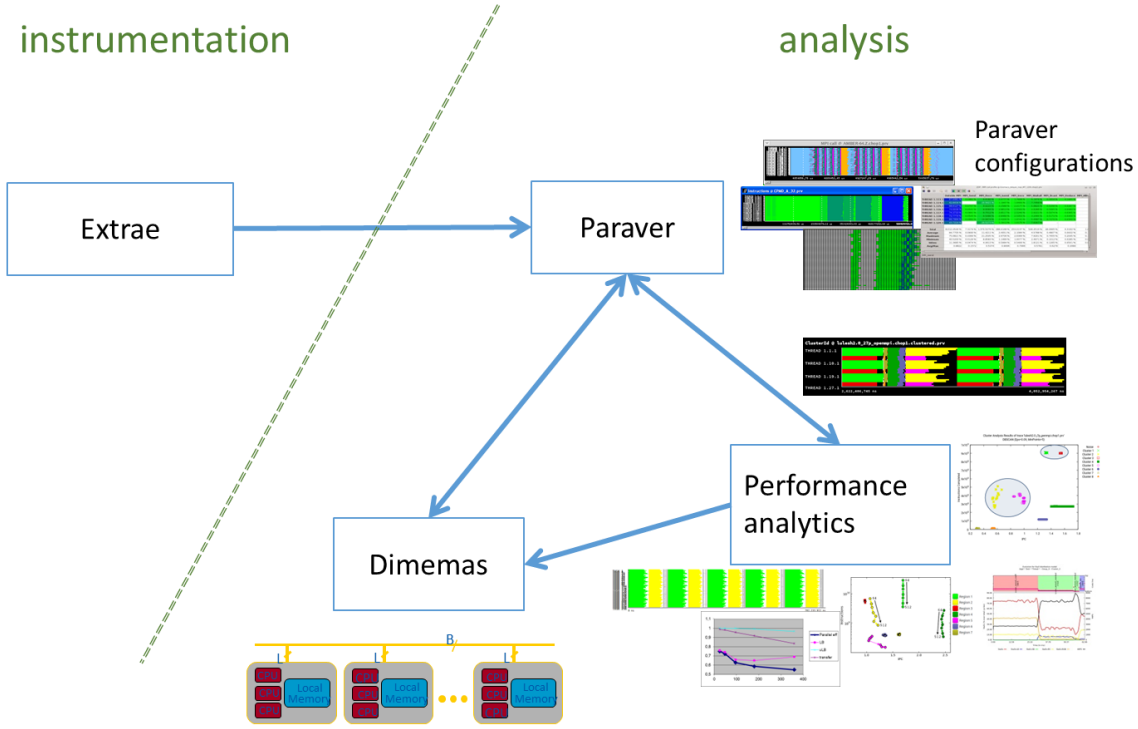
Barcelona Supercomputing Center

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?

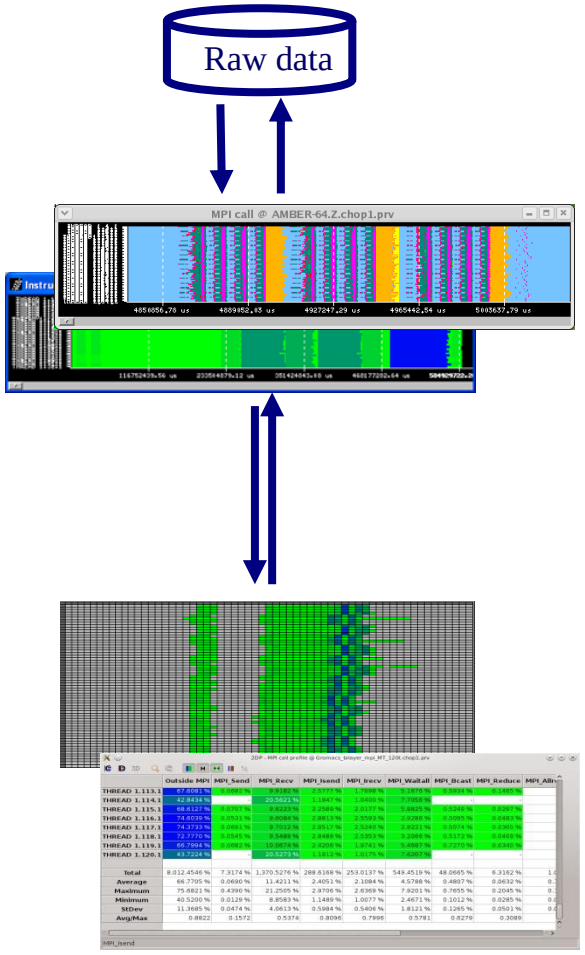
Our tools

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



Paraver

Paraver: Performance data browser



Timelines

2/3D tables
(Statistics)

Trace visualization/analysis

+ trace manipulation

Goal = Flexibility

No semantics

Programmable

Comparative analyses

Multiple traces

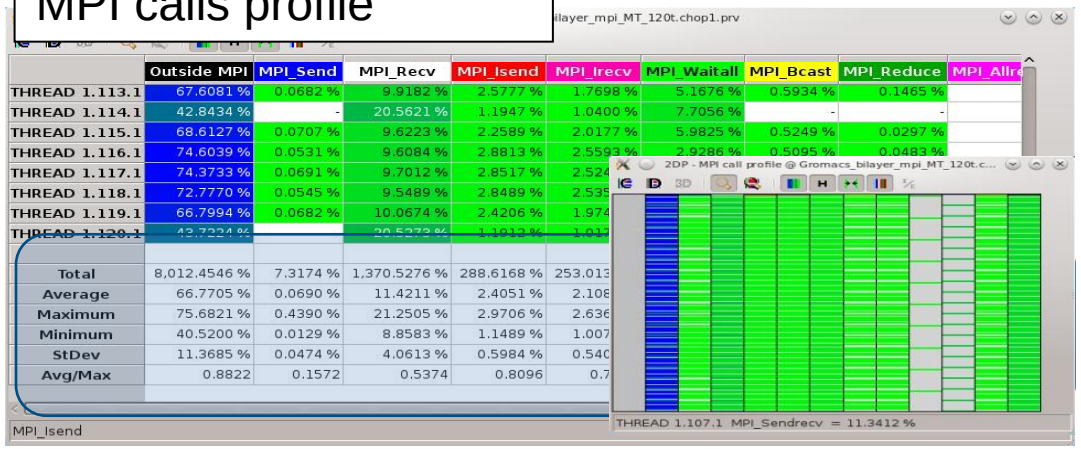
Synchronize scales

Tables: Profiles, histograms, correlations

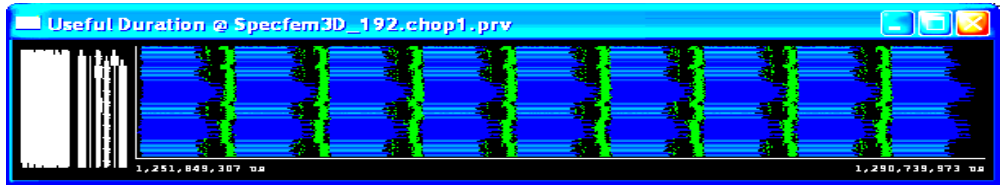
- From timelines to tables

MPI calls

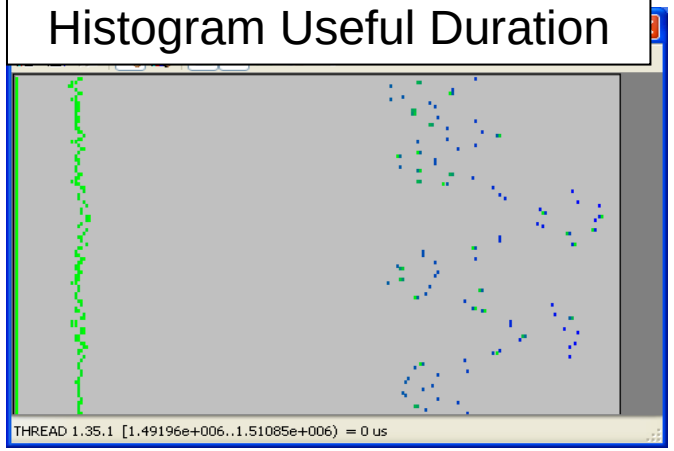
MPI calls profile



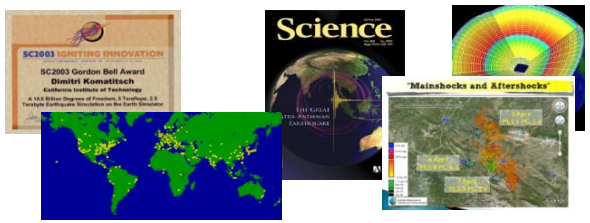
Useful Duration



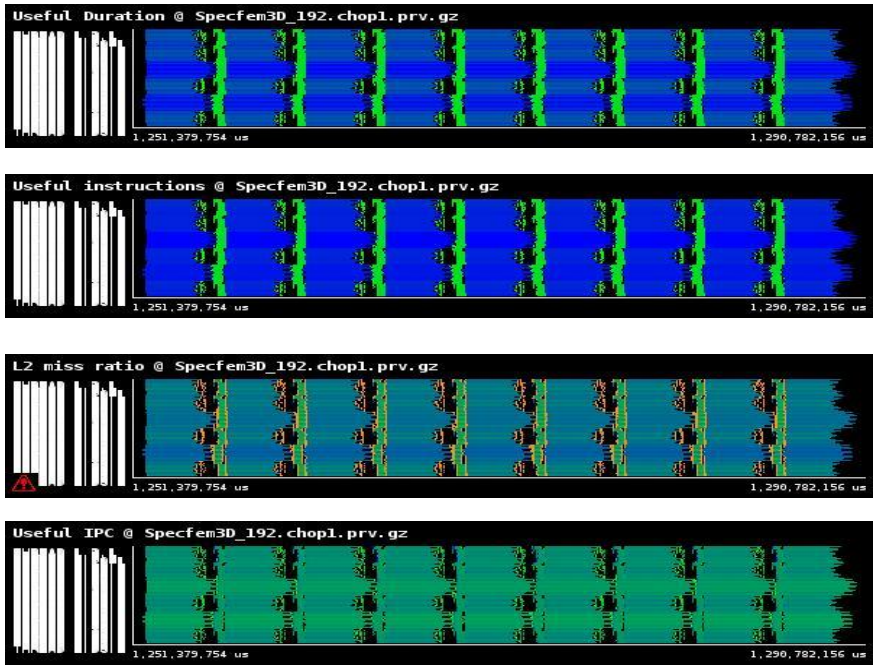
Histogram Useful Duration

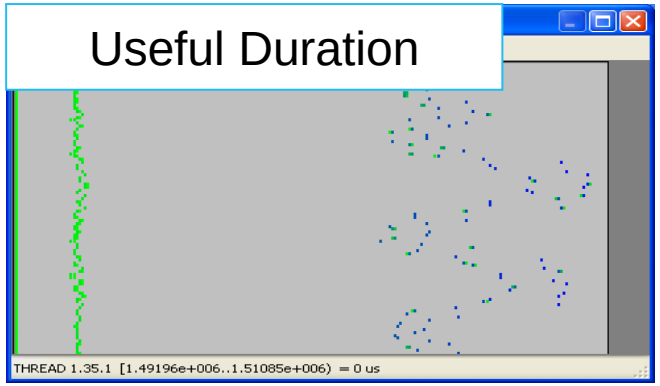


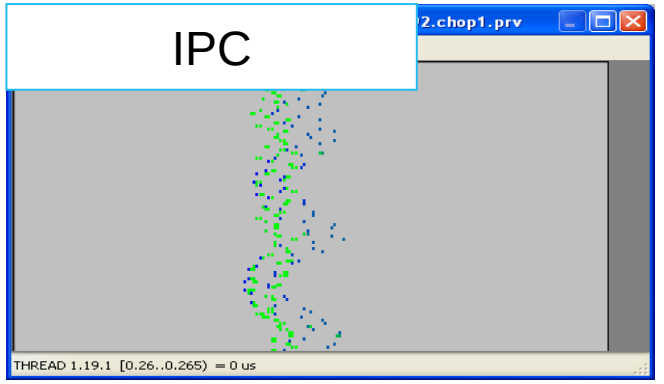
Analyzing variability through histograms and timelines

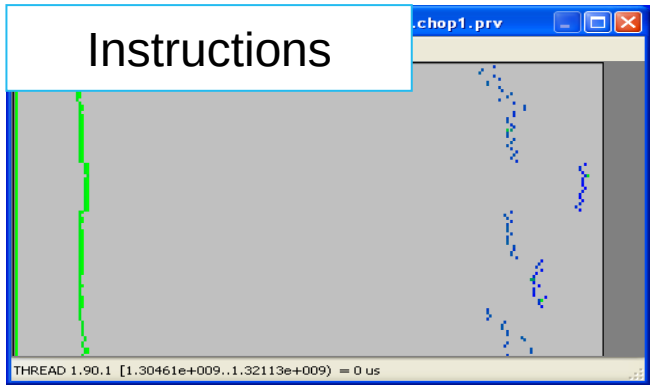


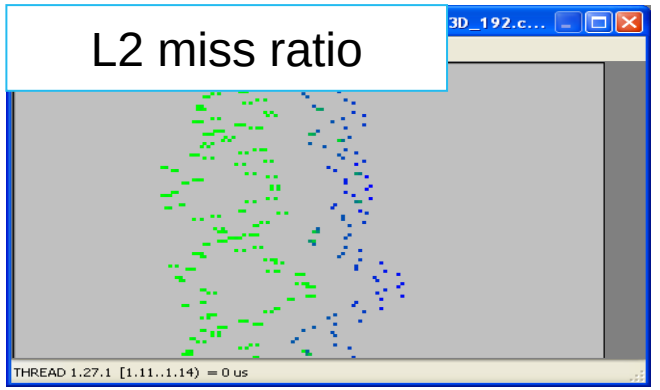
SPECFEM3D





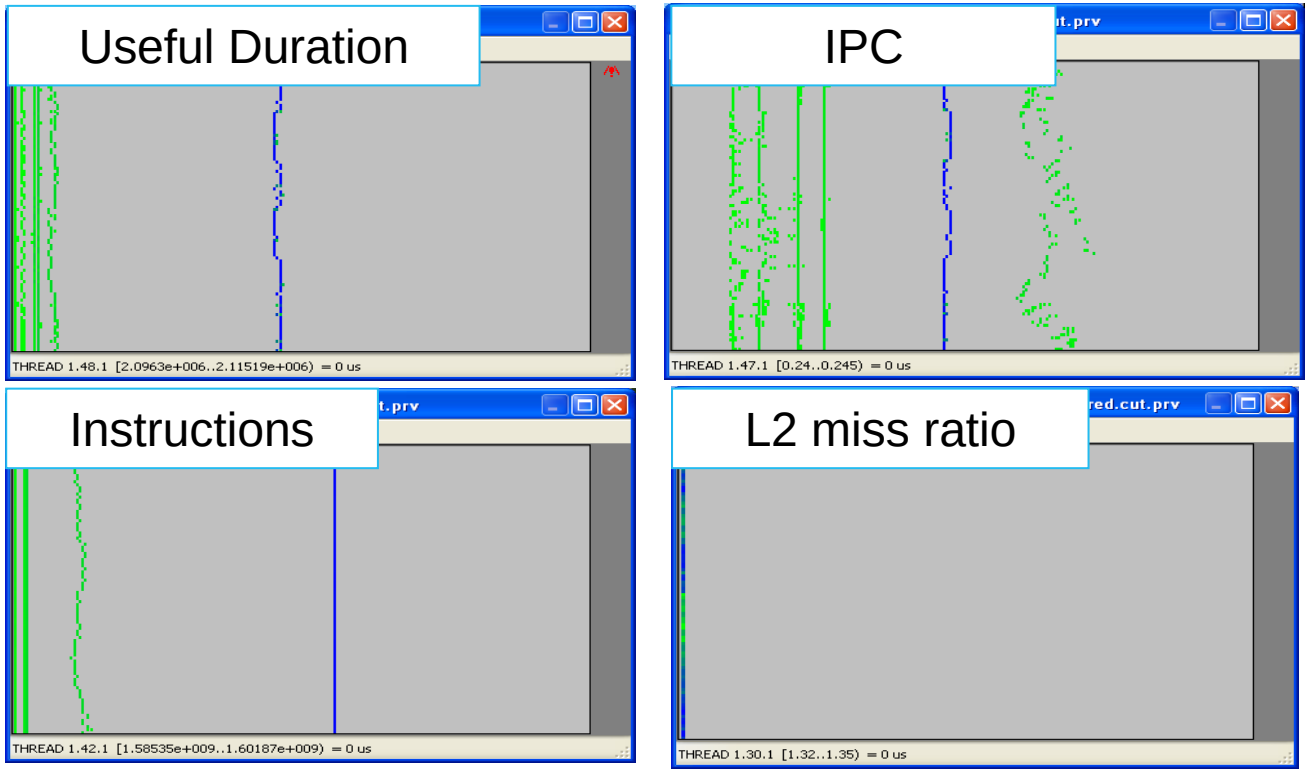






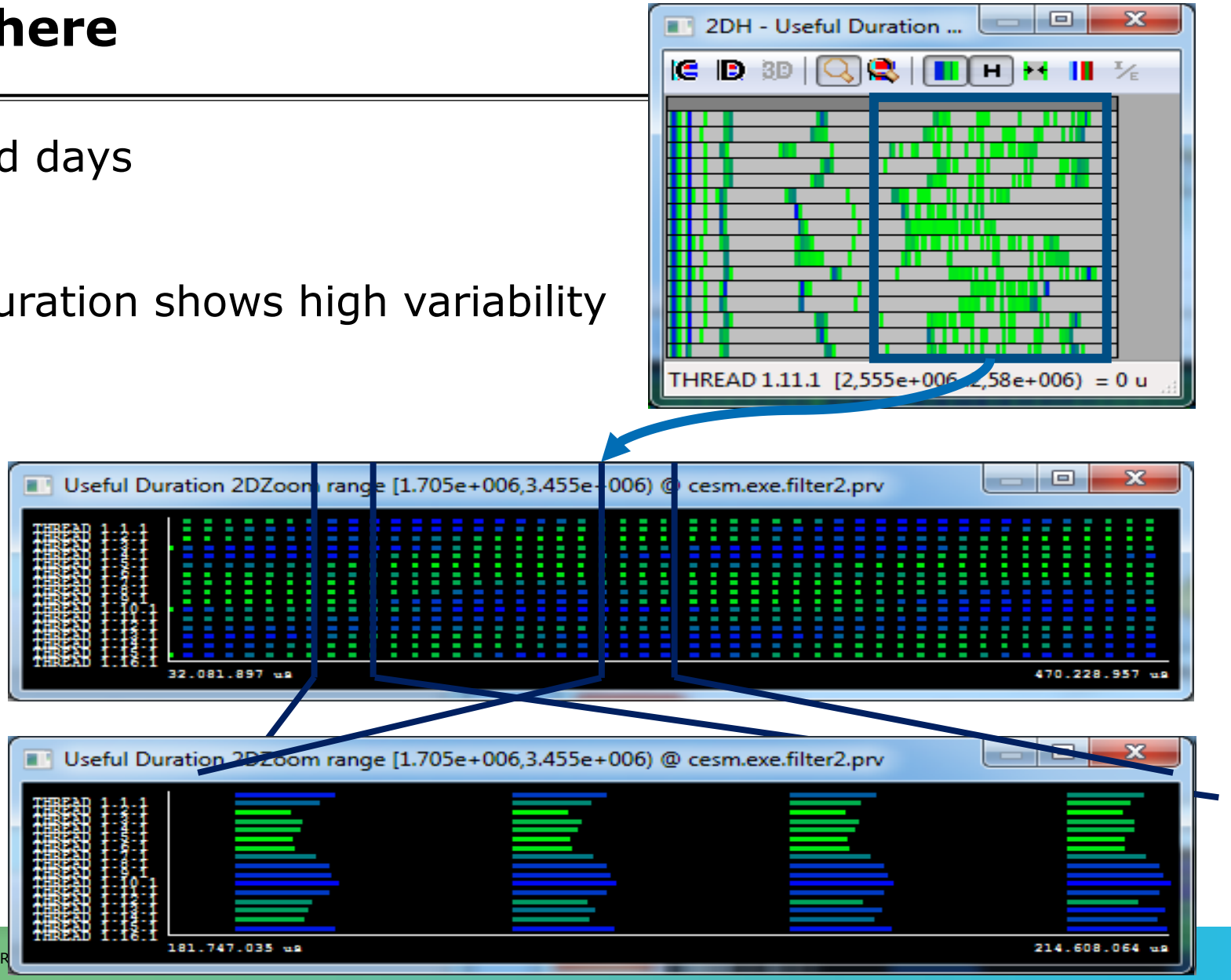
Analyzing variability through histograms and timelines

- By the way: six months later ….



Variability … is everywhere

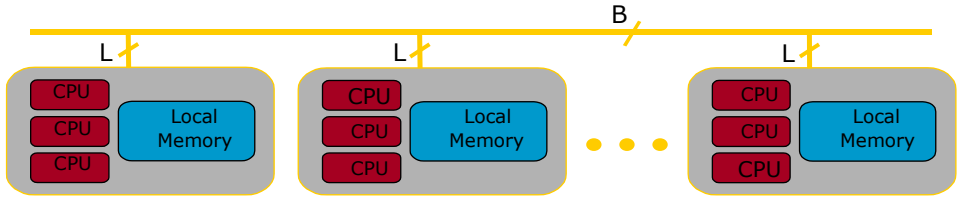
- CESM: 16 processes, 2 simulated days
- Histogram useful computation duration shows high variability
- How is it distributed?
- Dynamic imbalance
  - In space and time
  - Day and night

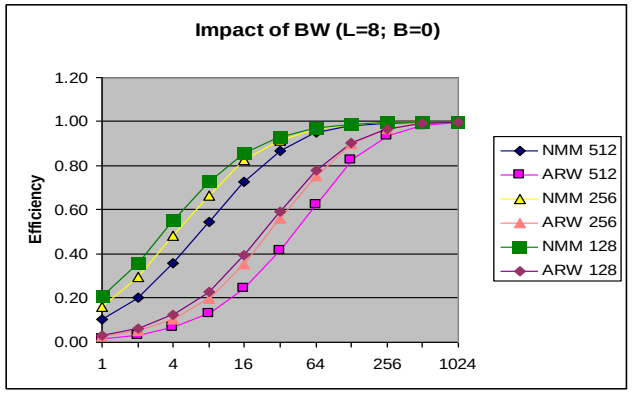


Dimemas

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

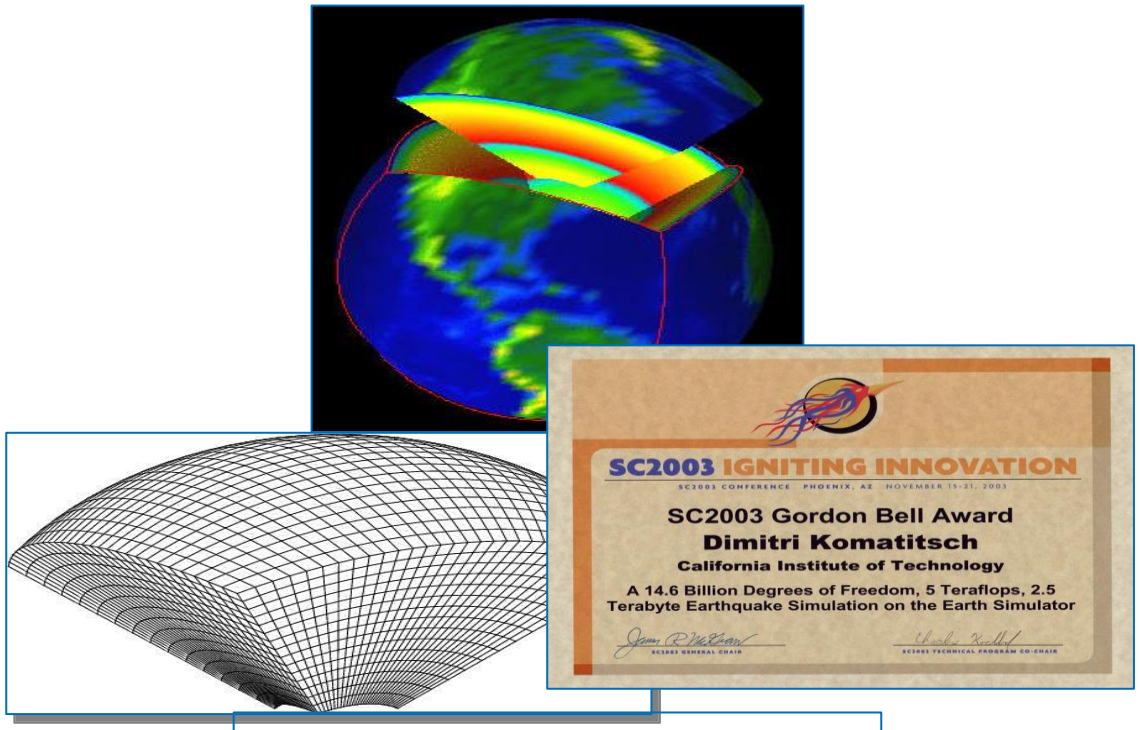




Detailed feedback on simulation (trace)

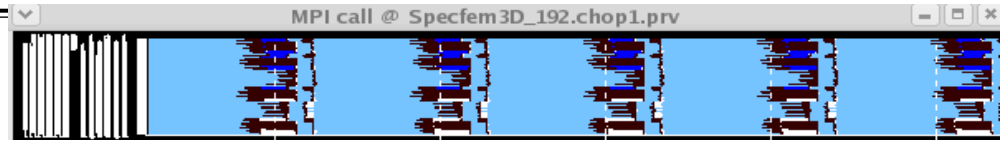
Would I will benefit from asynchronous communications?

- SPECFEM3D



Courtesy Dimitri Komatitsch

Real



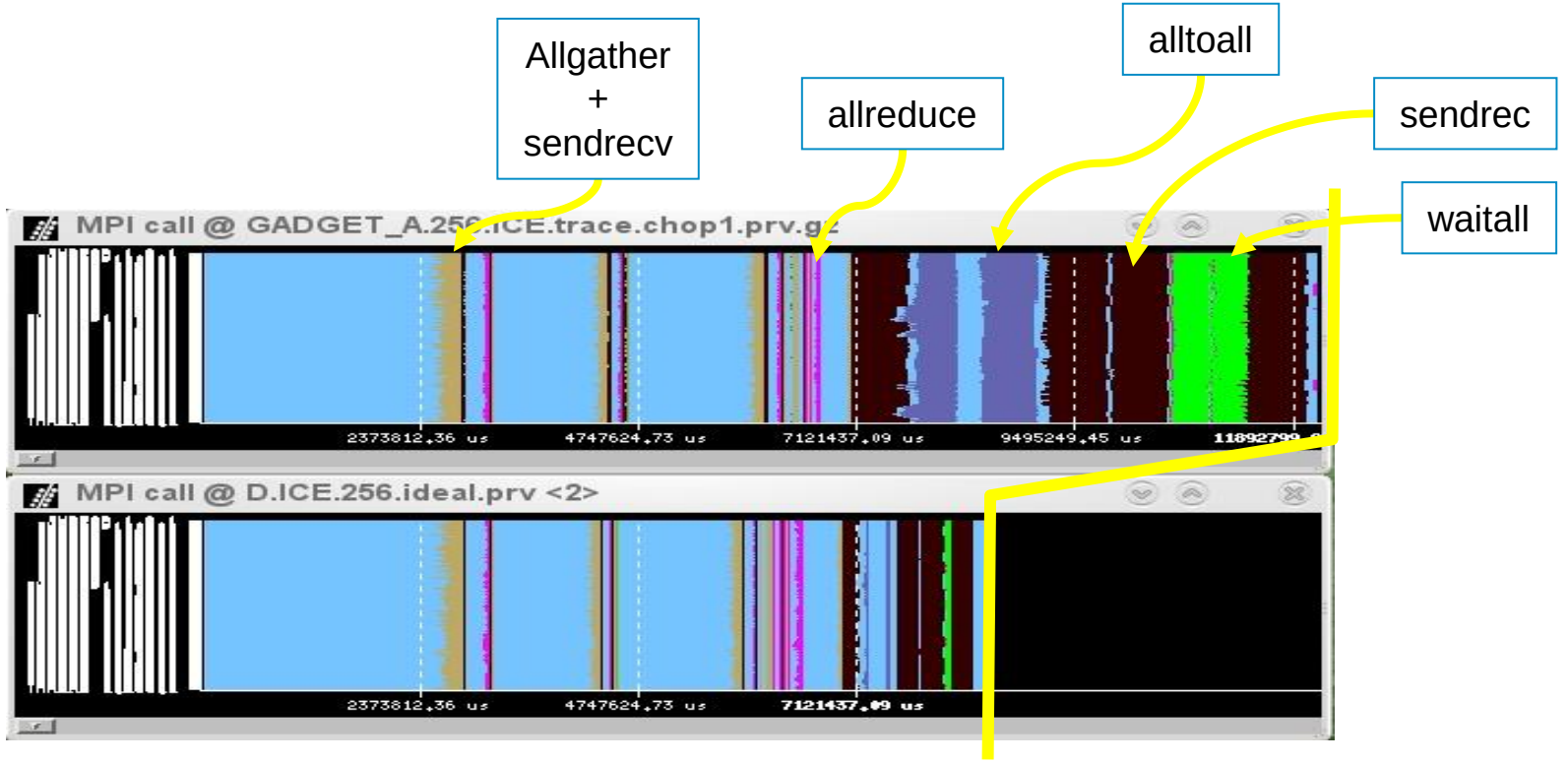
Ideal machine

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

GADGET @ Nehalem cluster
256 processes

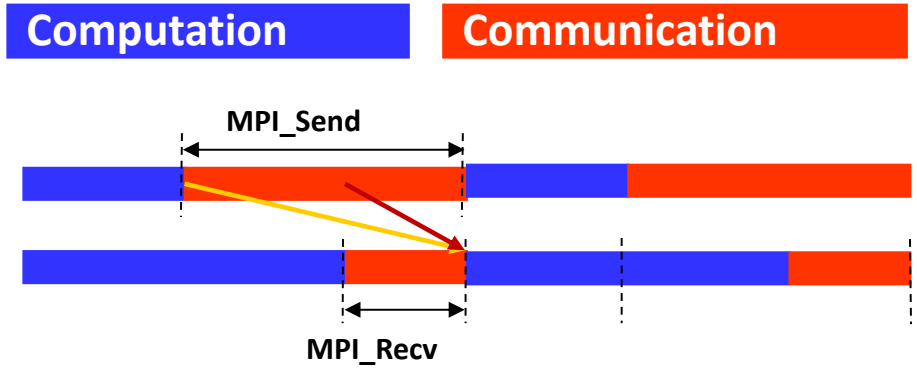
Real run

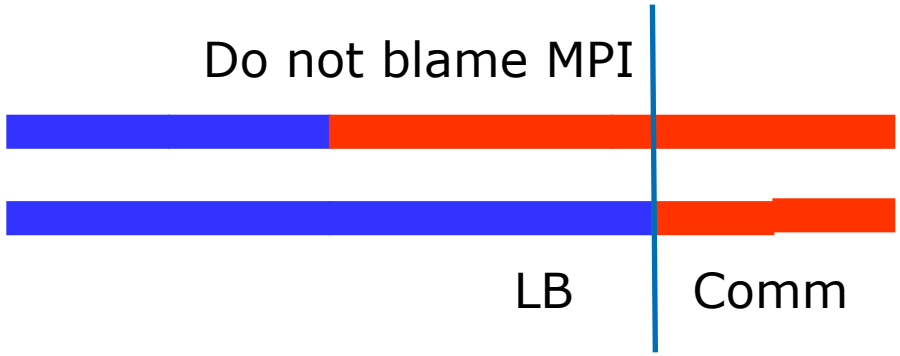
Ideal network



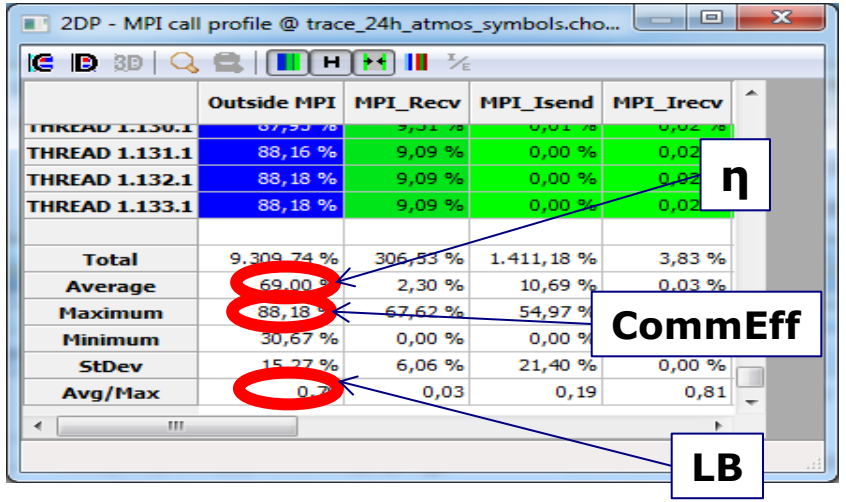
Efficiency Model

Parallel efficiency model

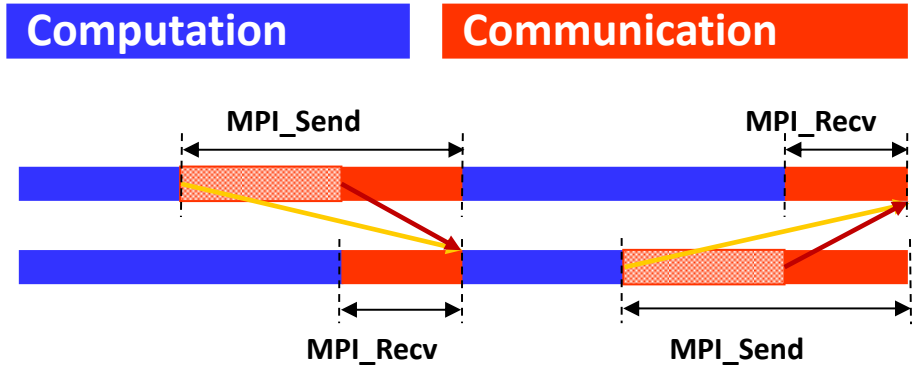


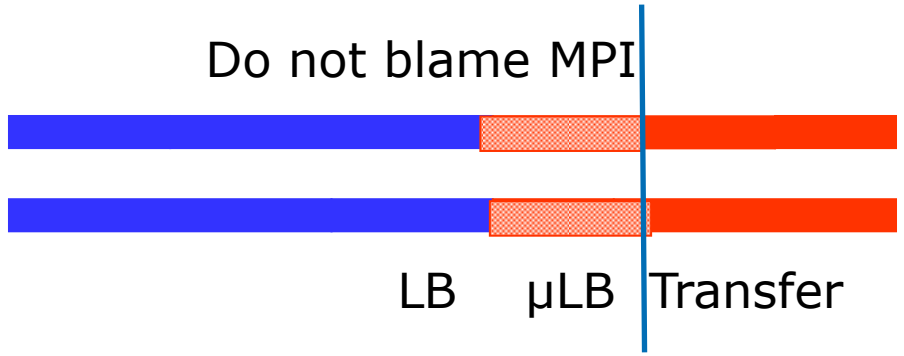


- Parallel efficiency = LB eff * Comm eff



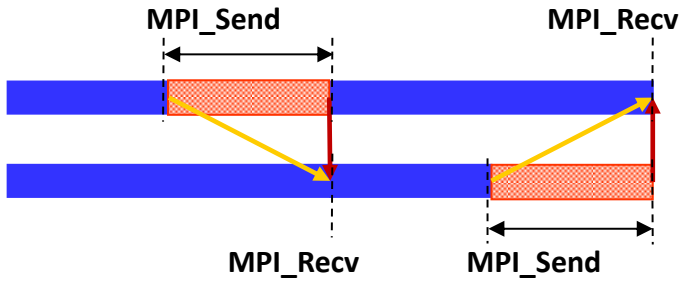
Parallel efficiency refinement: LB * µLB * Transfer





$$LB = 1$$

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

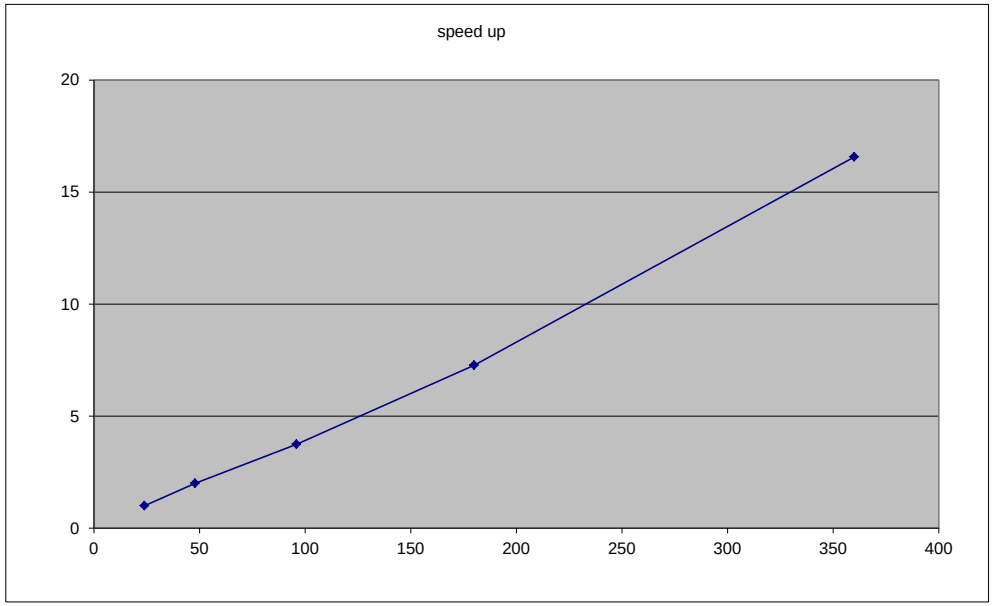


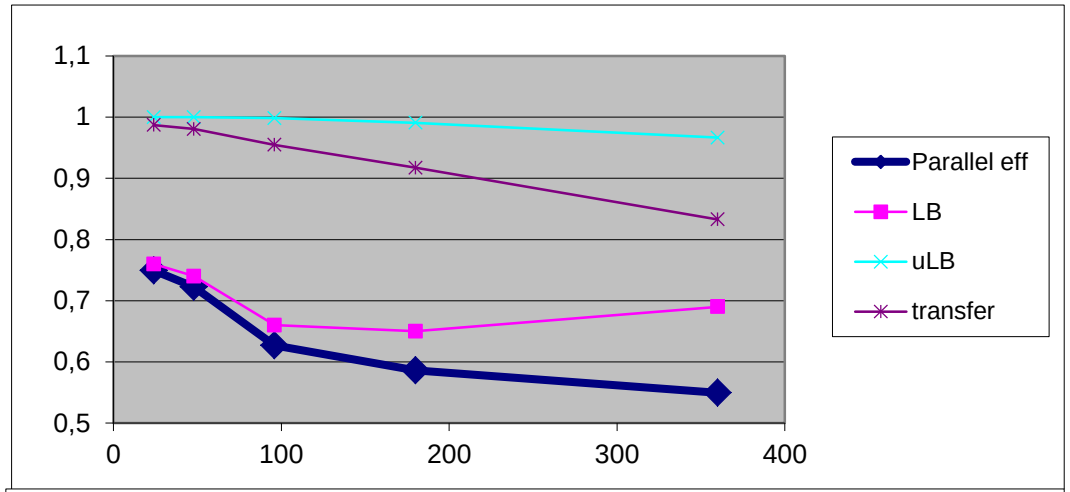
Why scaling?

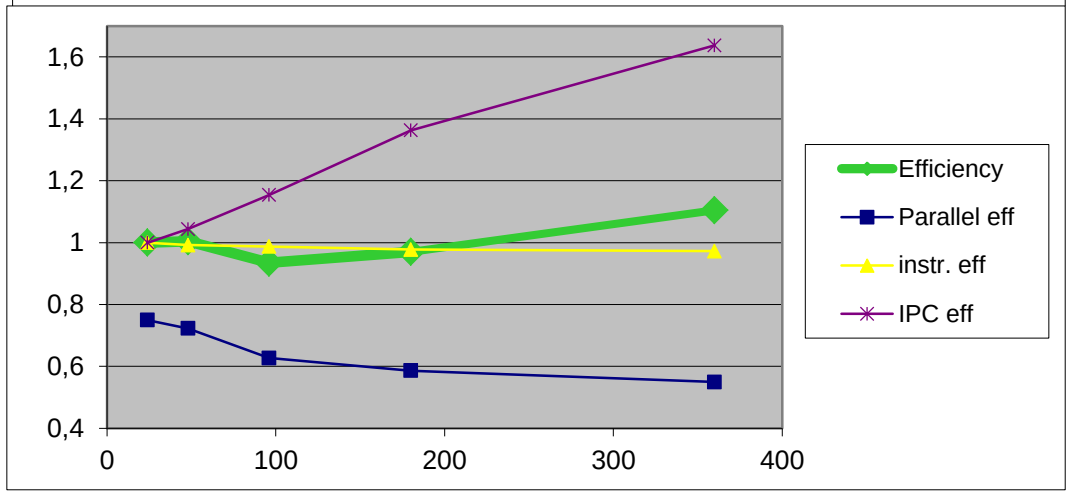
CG-POP mpi2s1D - 180x120

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

Good scalability !!
Should we be happy?





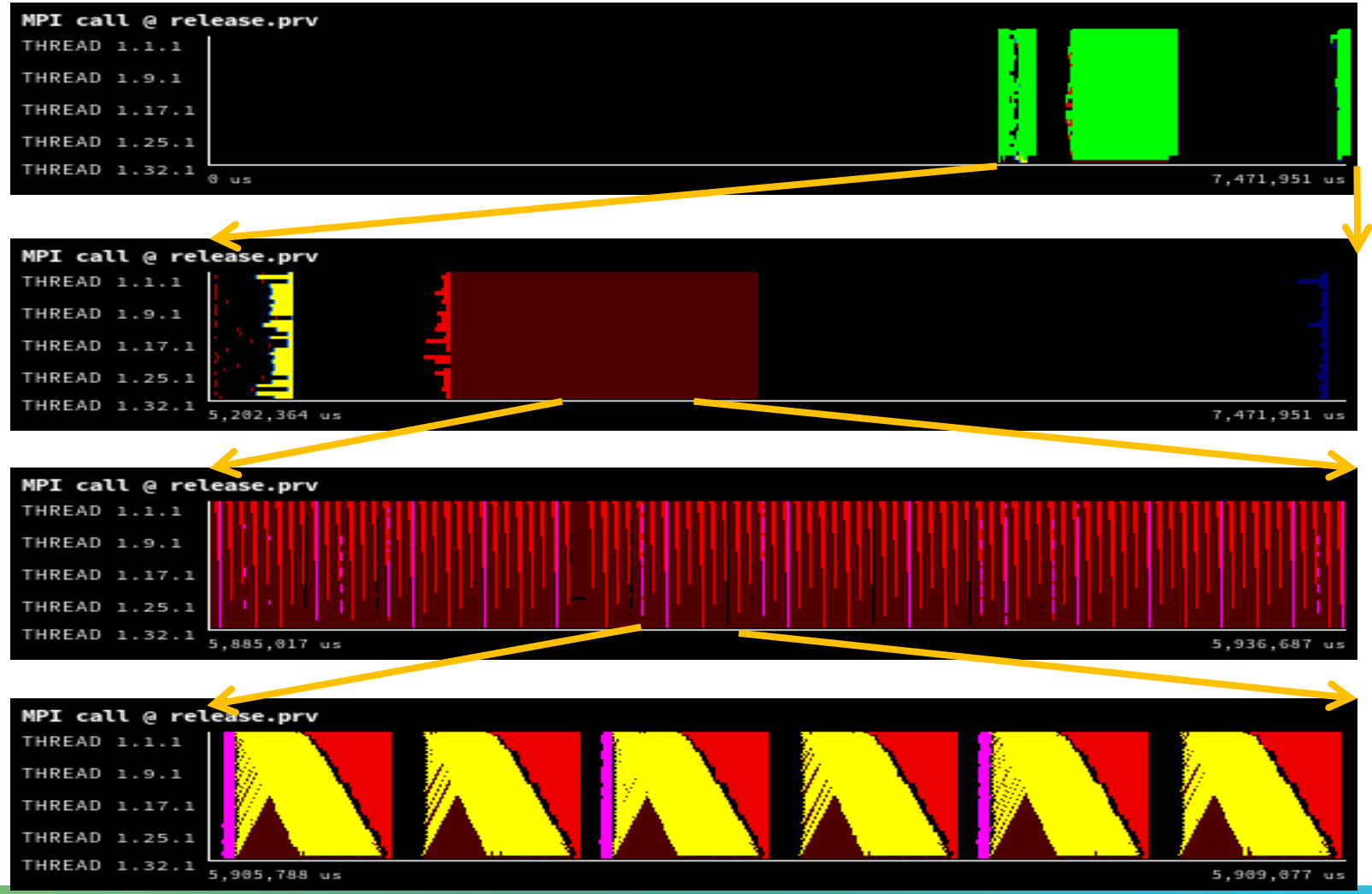


Why efficient?

Parallel efficiency =93.28
Communication = 93.84

Parallel efficiency = 77.93
Communication = 79.79

Parallel efficiency = 28.84
Communication eff = 30.42



Methodology

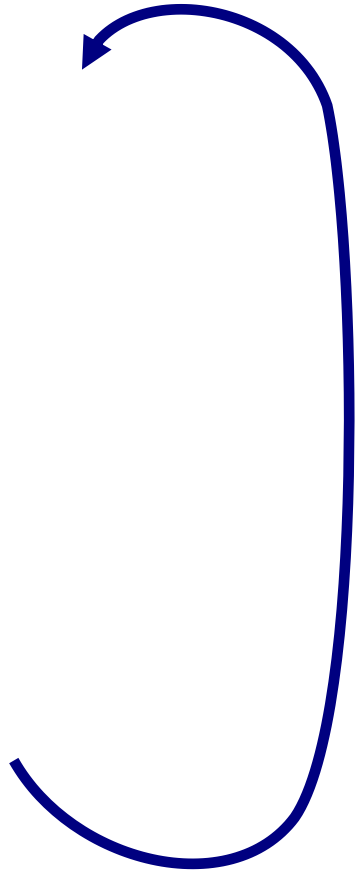
Performance analysis tools objective

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?

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