

Understanding applications using the BSC performance tools

Judit Giménez (judit@bsc.es), Jesús Labarta

Barcelona Supercomputing Center

Humans are visual creatures

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

PROCESS

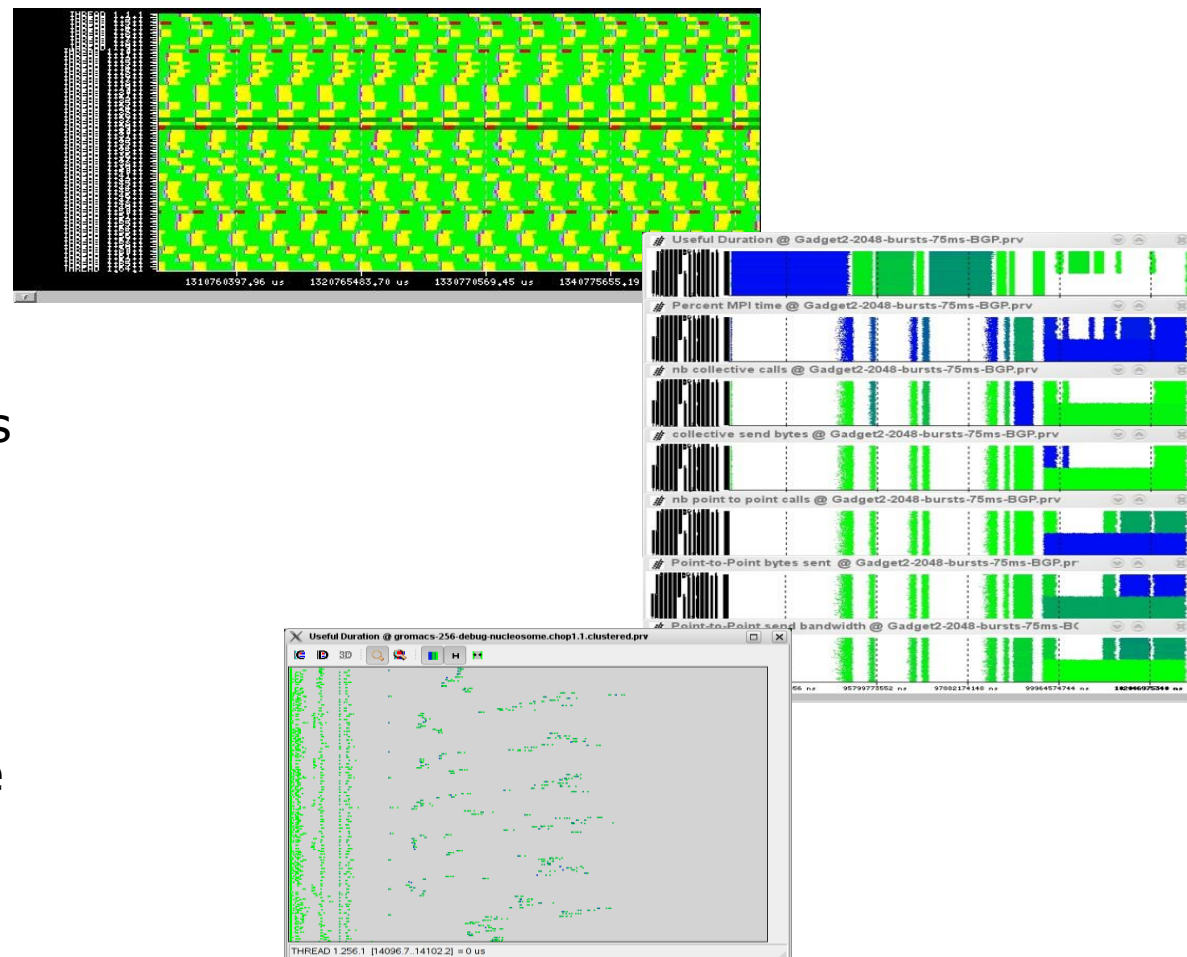
STORE

IDENTIFY



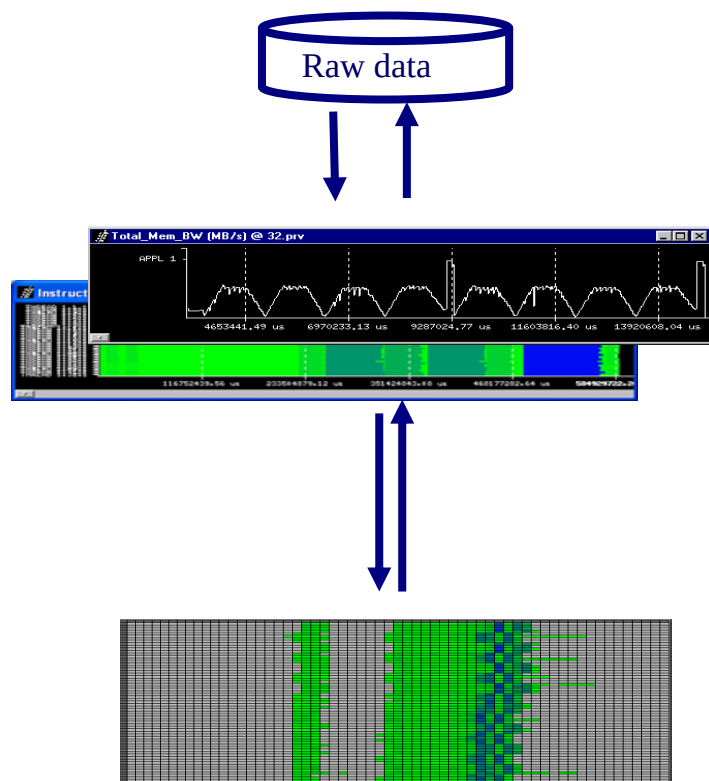
Our Tools

- 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

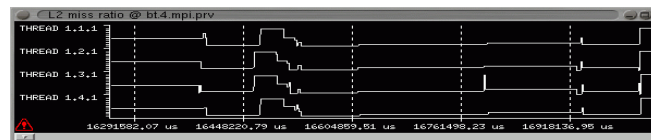
Comparative analyses

Multiple traces

Synchronize scales

Timelines

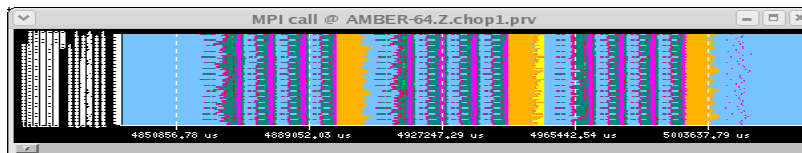
- Each window displays one view
 - Piecewise constant** function of time



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

- Types of functions

- Categorical
 - State, user function, outlined routine

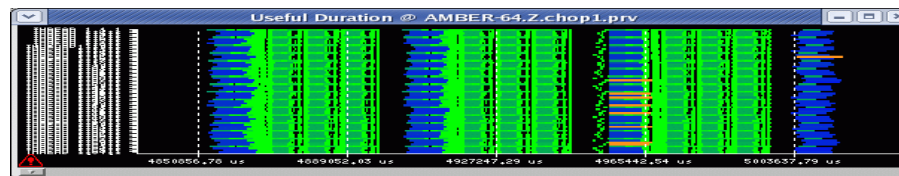


$$S_i \in [0, n] \subset N, \quad n <$$

- Logical
 - In specific user function, In MPI call, In long MPI call

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

- Numerical
 - IPC, L2 miss ratio, Duration of MPI call, duration of computation burst

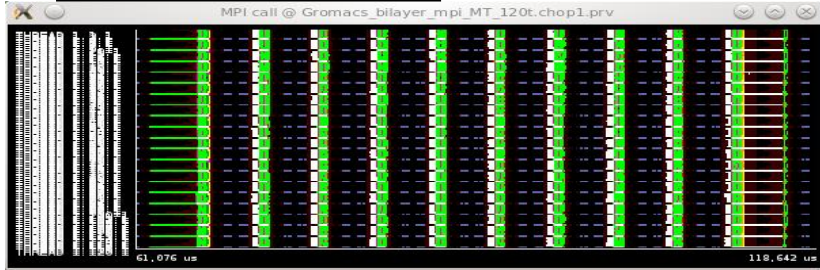


$$S_i \in R$$

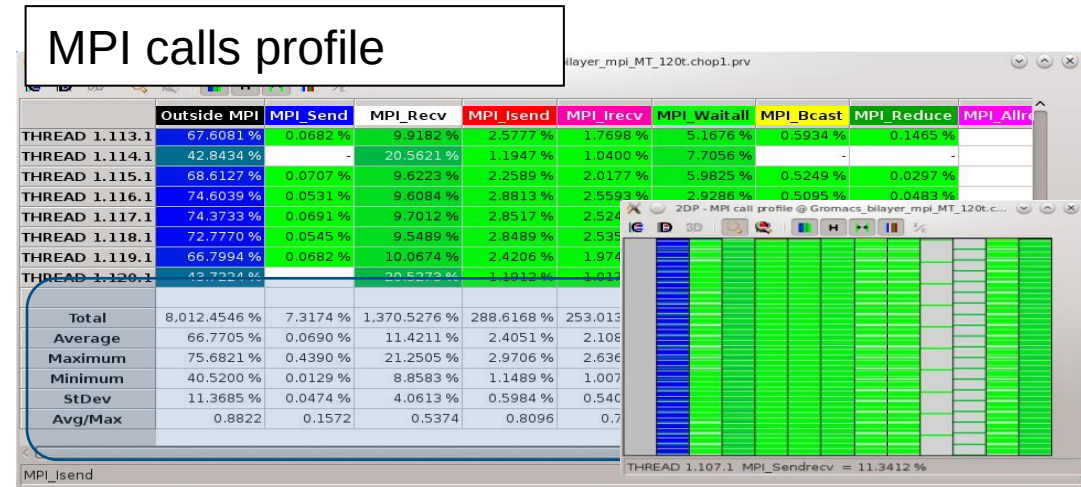
Tables: Profiles, histograms, correlations

- From timelines to tables

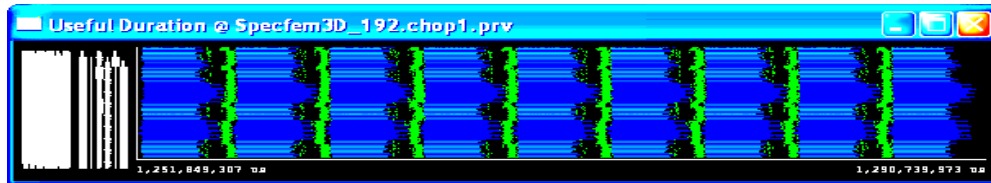
MPI calls



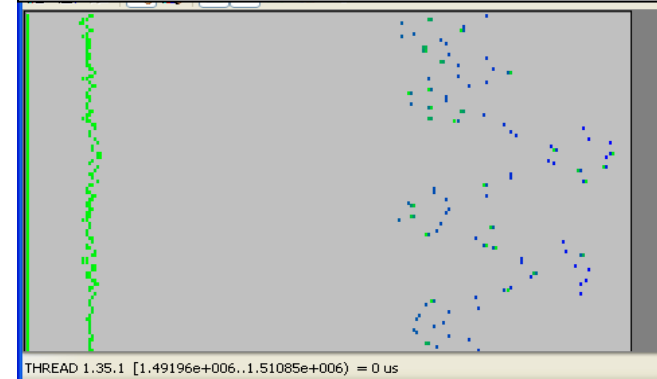
MPI calls profile



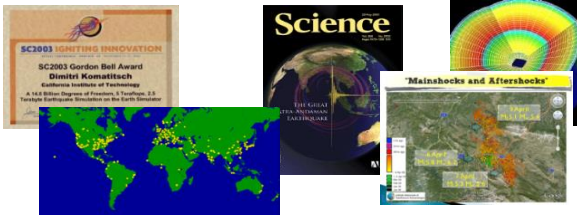
Useful Duration



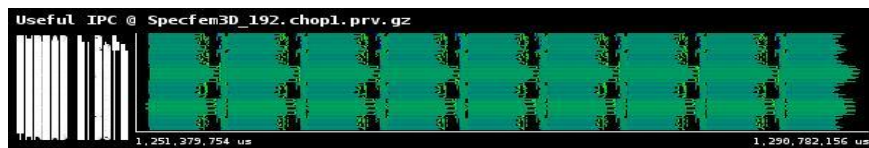
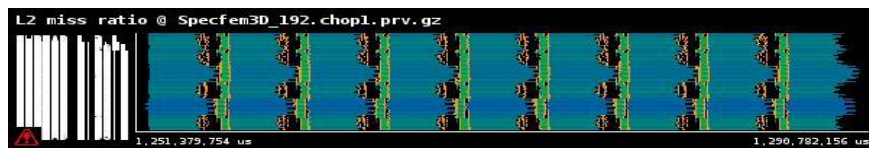
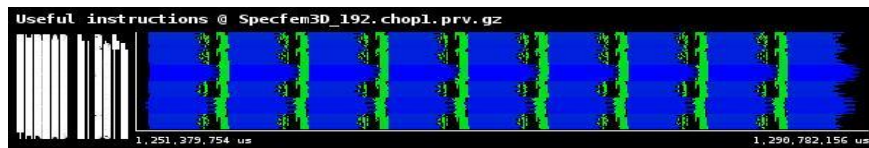
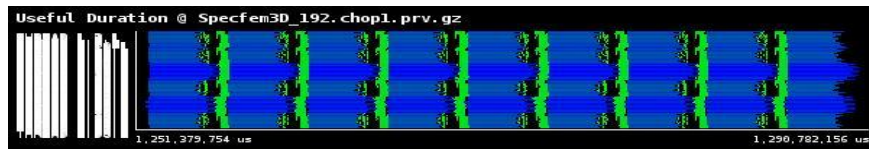
Histogram Useful Duration



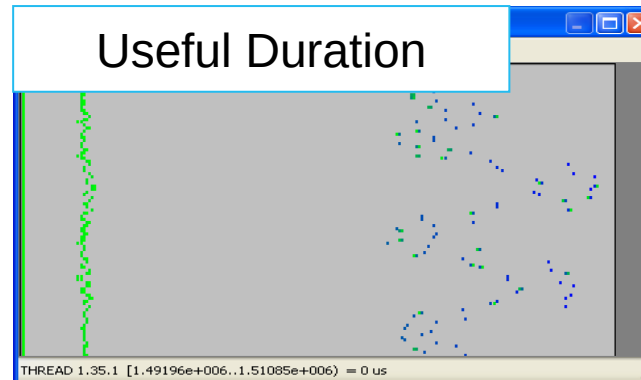
Analyzing variability through histograms and timelines



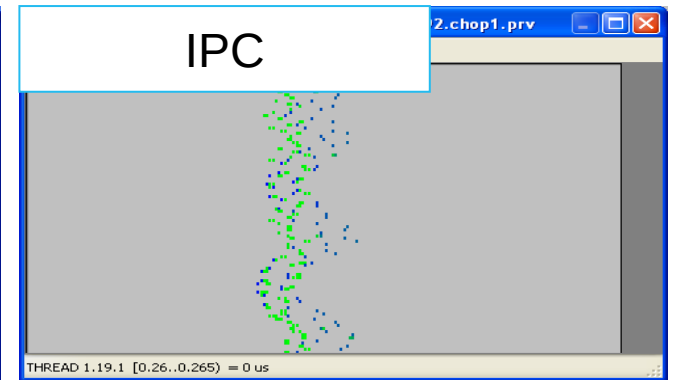
SPECFEM3D



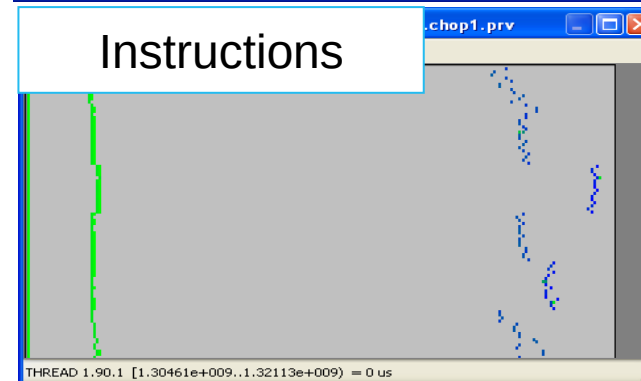
Useful Duration



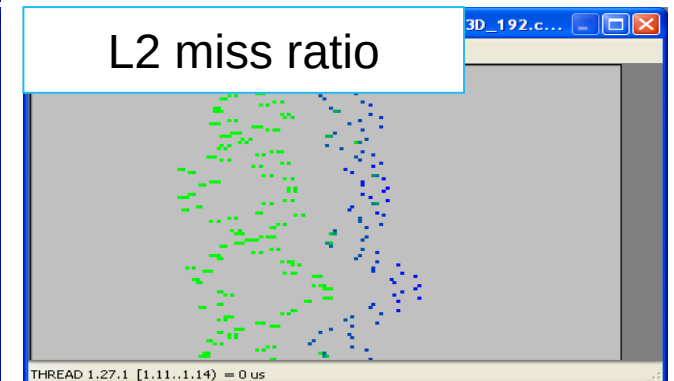
IPC



Instructions

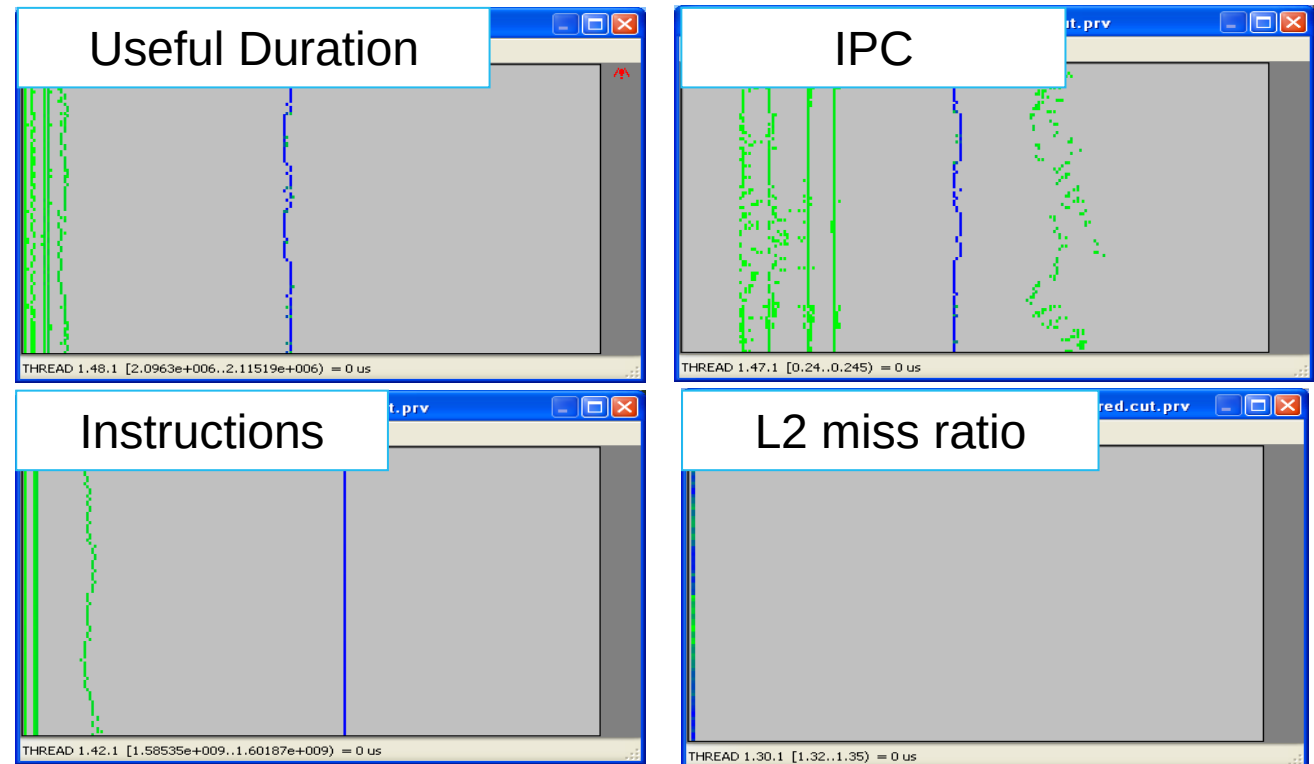


L2 miss ratio



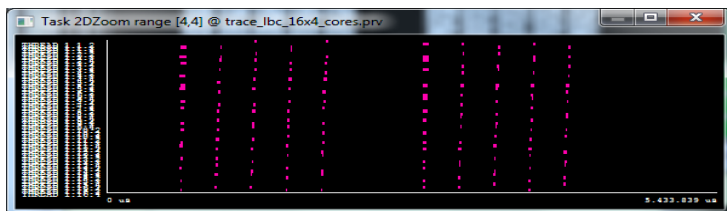
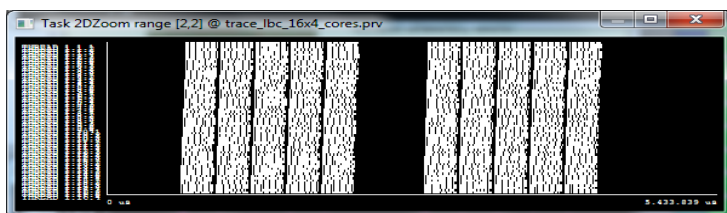
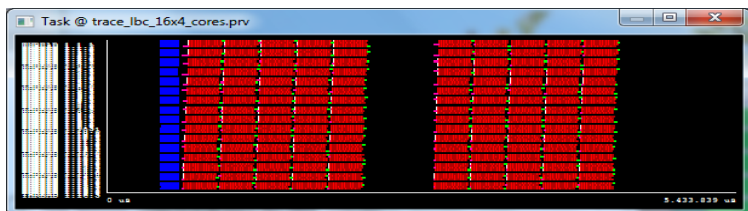
Analyzing variability through histograms and timelines

- By the way: six months later

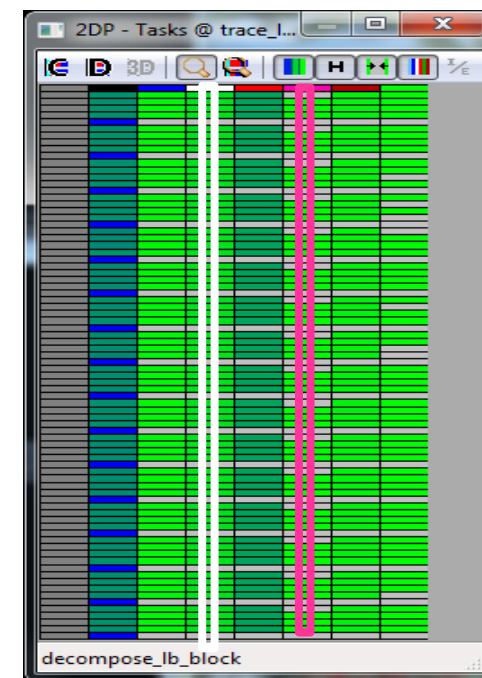


From tables to timelines

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

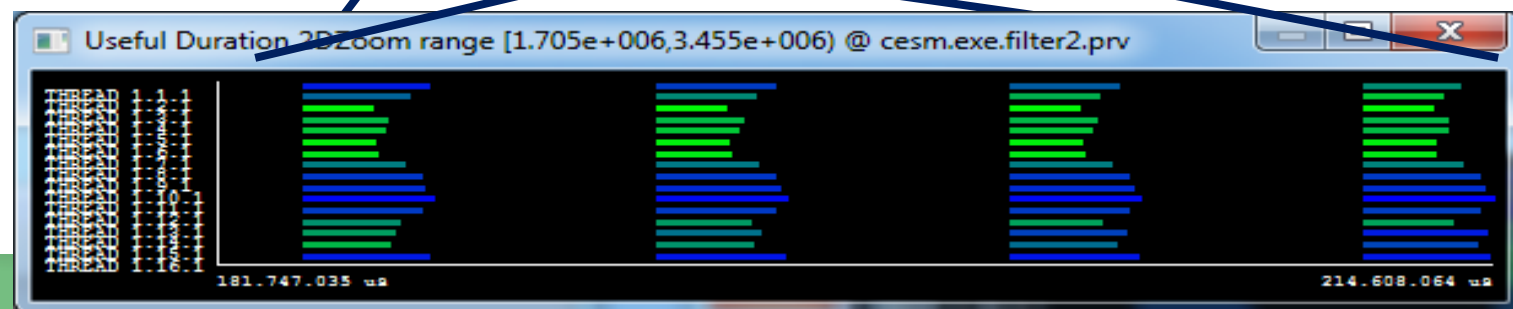
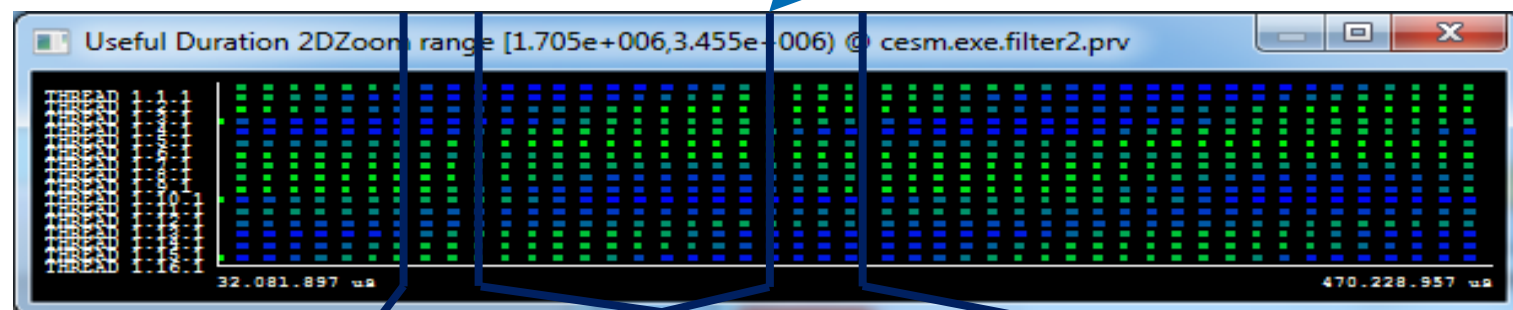
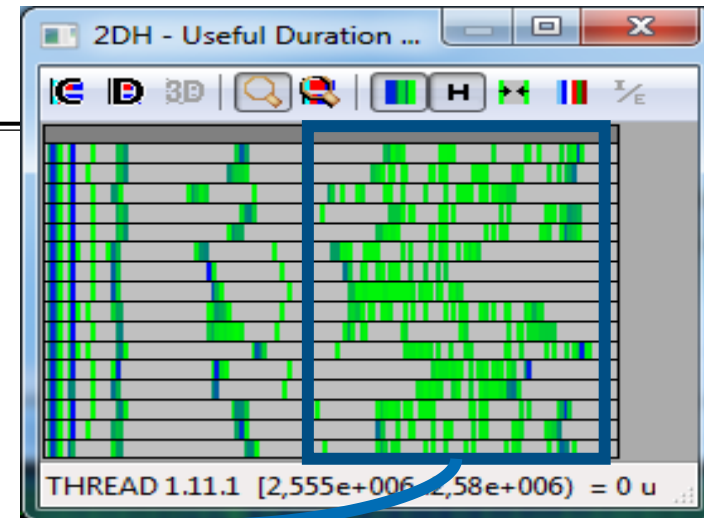


Only showing when a given
value happens



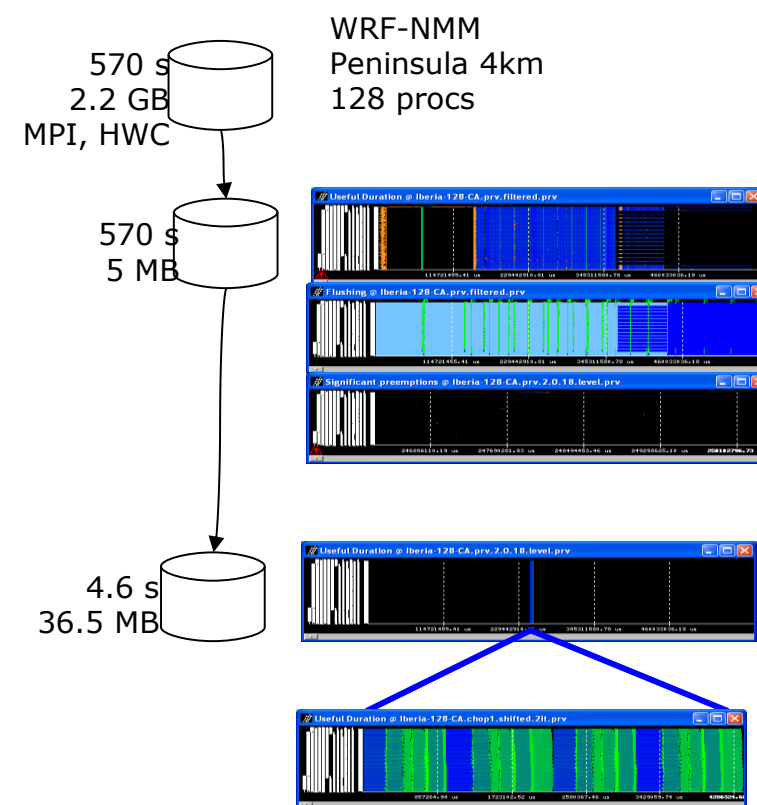
Variability ... is everywhere

- 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 ? ☺



Trace manipulation

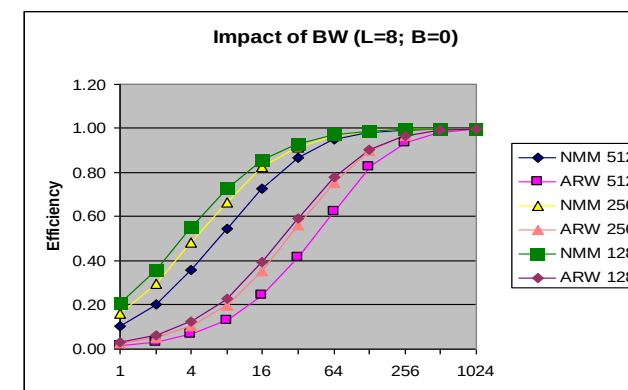
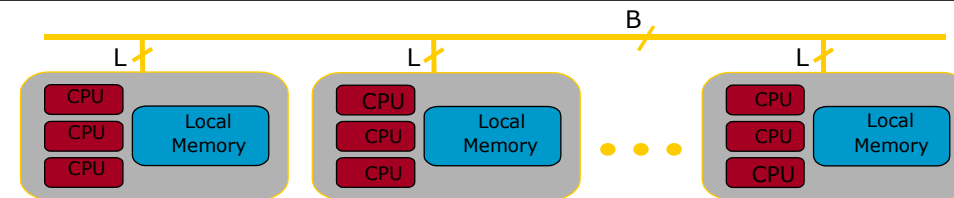
- Data handling/summarization capability
- Filtering
 - Subset of records in original trace
 - By duration, type, value,...
 - Filtered trace IS a paraver trace and can be analysed with the same cfgs (as long as needed data kept)
- Cutting
 - All records in a given time interval
 - Only some processes
- 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

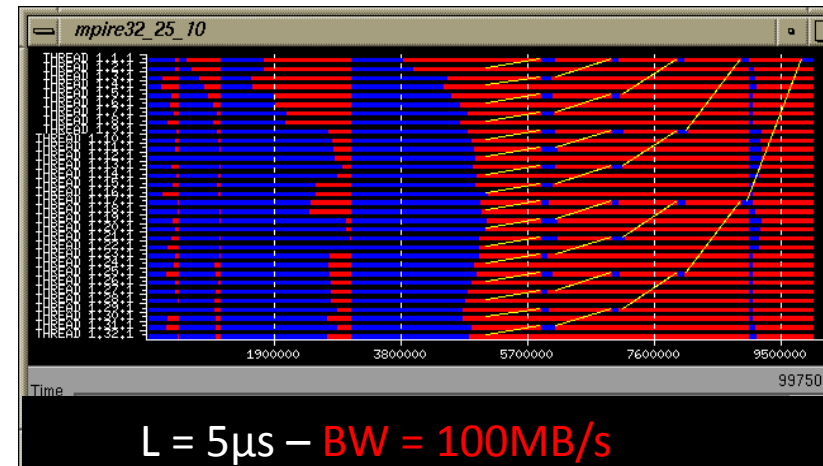
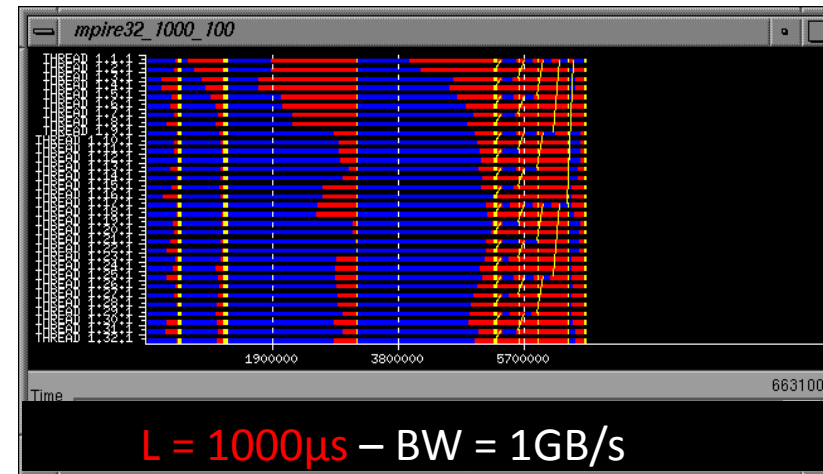
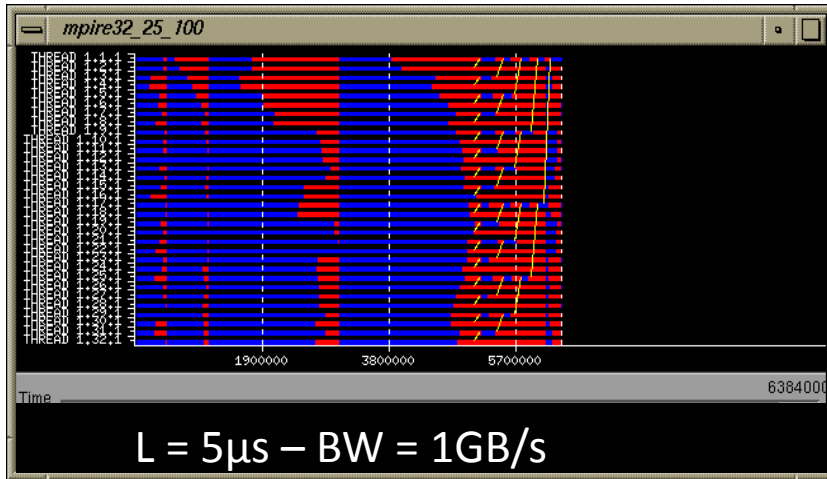
- Simulation: Highly non linear model
 - MPI protocols, resources contention...
- Parametric sweeps
 - On abstract architectures
 - On application computational regions
- What if analysis
 - Ideal machine (instantaneous network)
 - Estimating impact of ports to MPI+OpenMP/CUDA/...
 - Should I use asynchronous 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



Detailed feedback on simulation (trace)

Network sensitivity

- MPIRE 32 tasks, no network contention



All windows
same scale

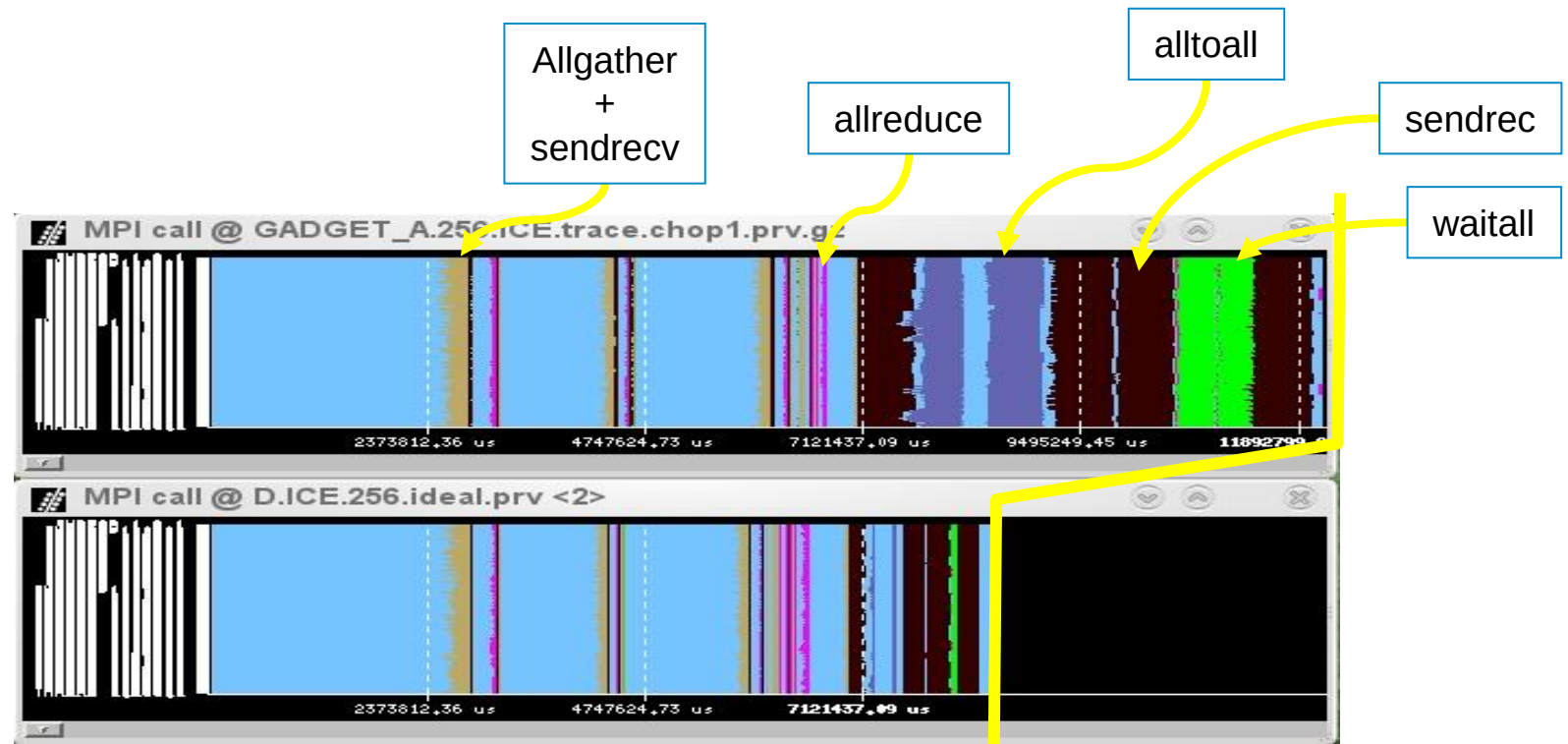
Ideal machine

- 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

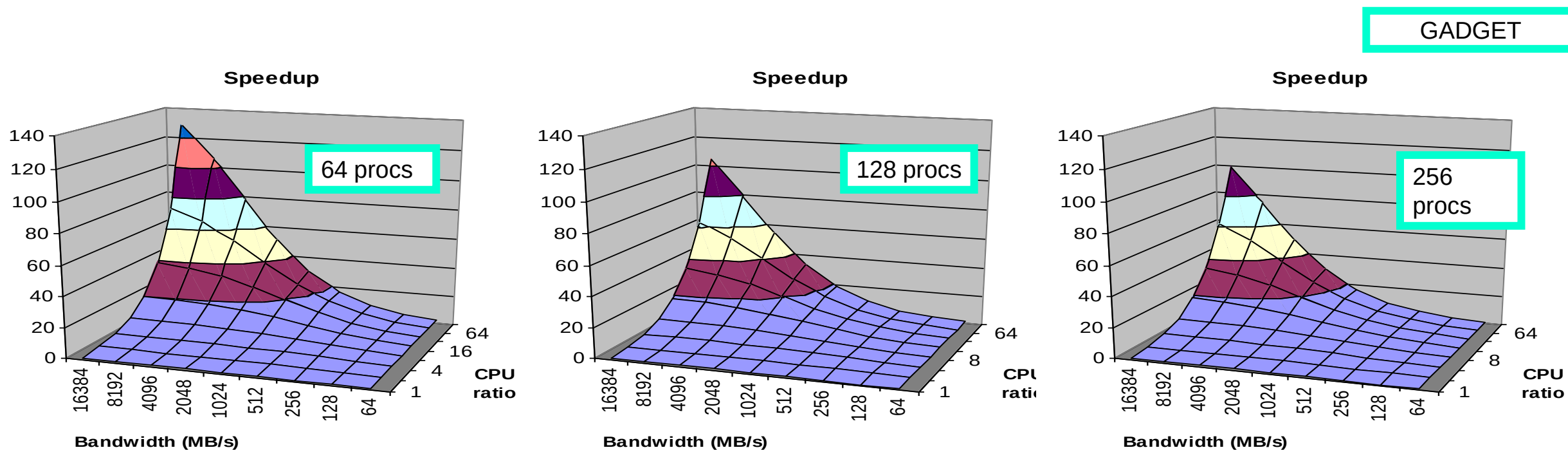
Ideal
network



Impact on practical machines?

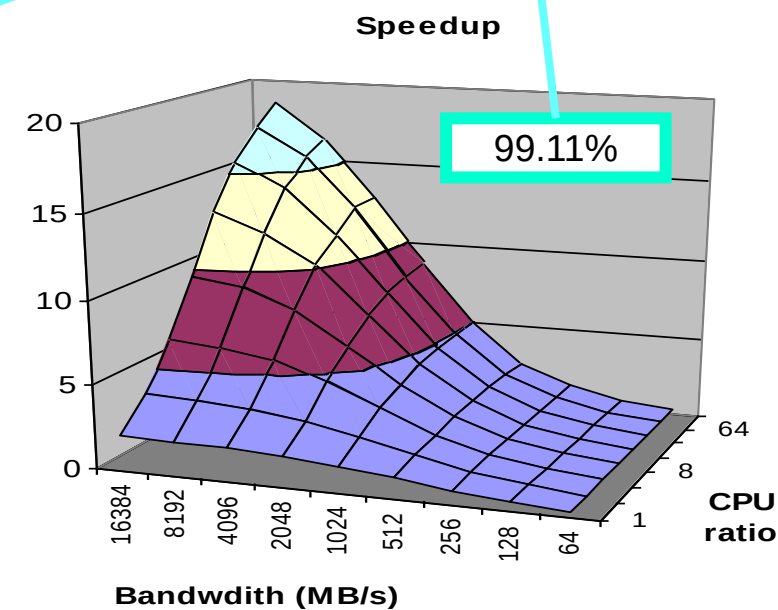
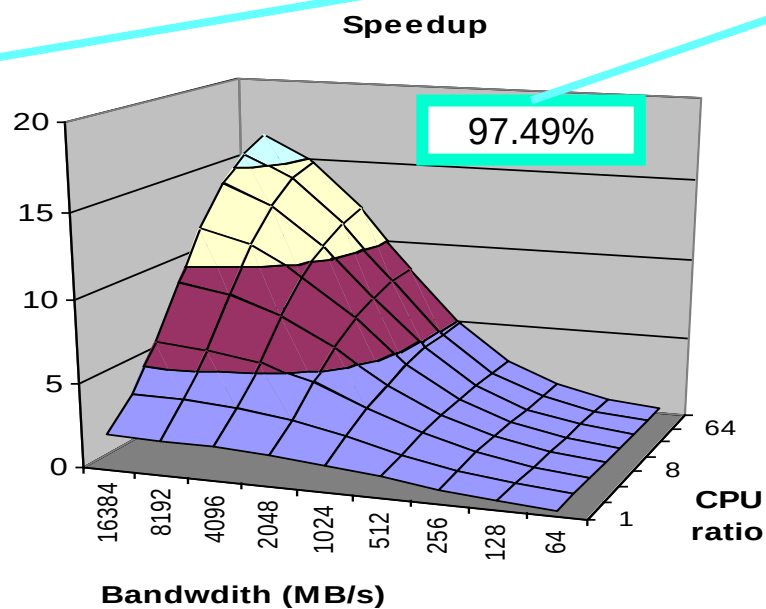
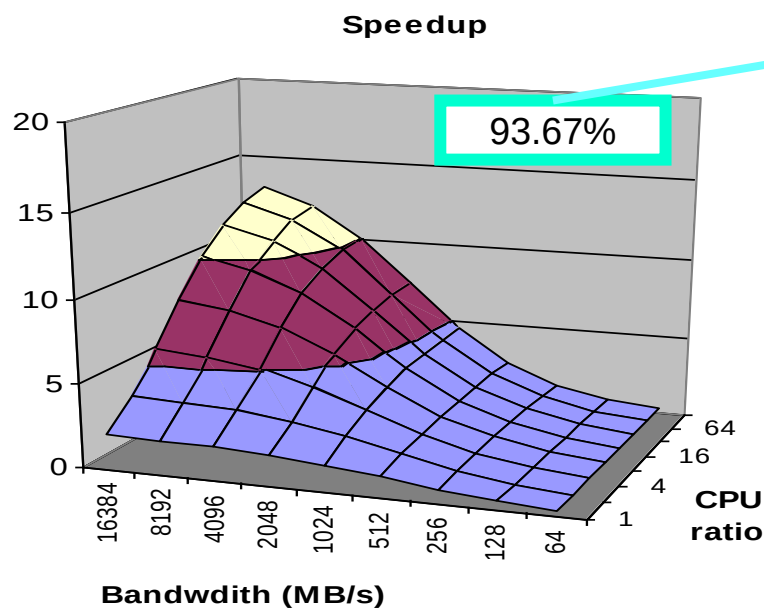
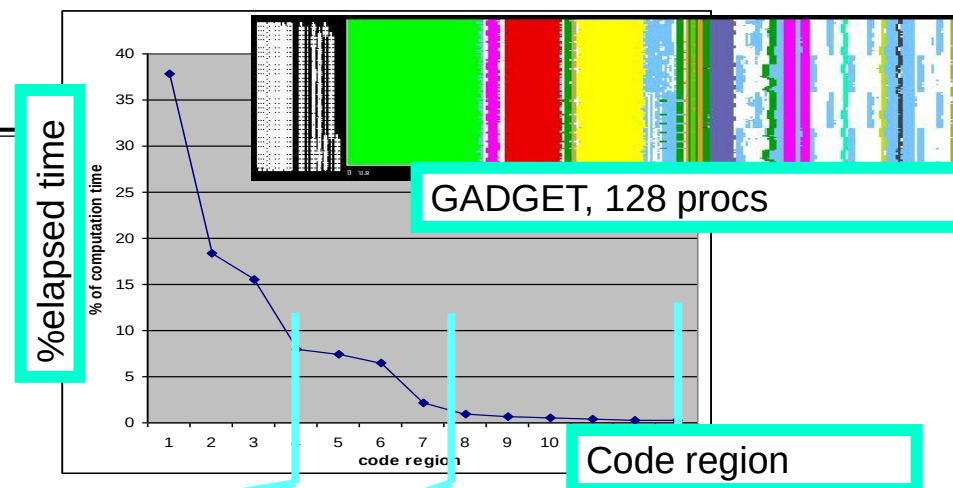
Impact of architectural parameters

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



Expectations of Hybrid parallelization

- Beware of the **hybrid Amdahl's law**
→ **asynchrony + dynamic load balancing mechanisms !!!**

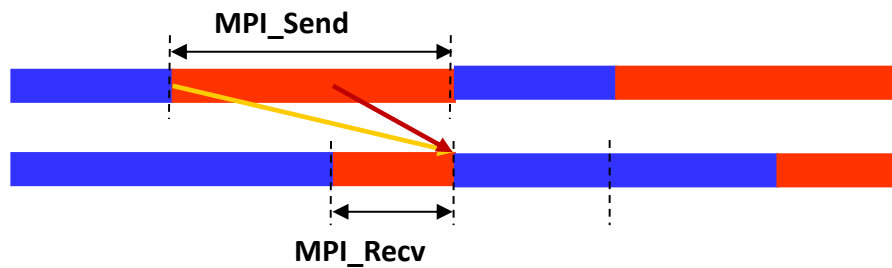


Models

Parallel efficiency model

Computation

Communication

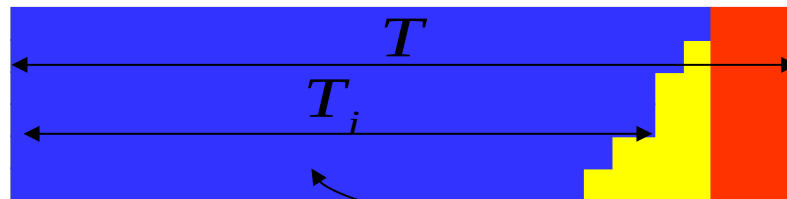


Do not blame MPI

LB

Comm

- Parallel efficiency = LB eff * Comm eff



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

$$CommEff = \max(eff_i)$$

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

IPC
instr

2DP - MPI call profile @ trace_24h_atmos_symbols.cho...

	Outside MPI	MPI_Recv	MPI_Isend	MPI_Irecv
THREAD 1.130.1	87,95 %	9,09 %	0,00 %	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 %	
Minimum	30,67 %	0,00 %	0,00 %	
StDev	15,27 %	6,06 %	21,40 %	0,00 %
Avg/Max	0,79	0,03	0,19	0,81

η

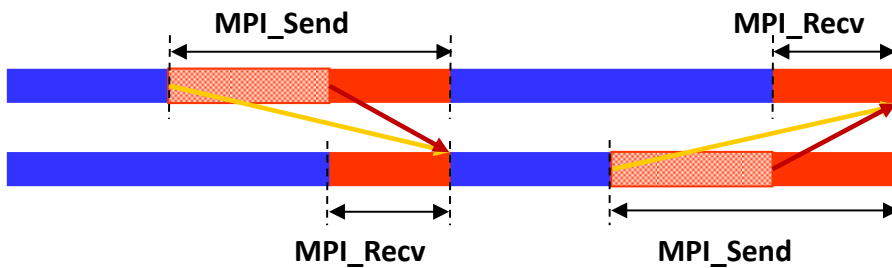
CommEff

LB

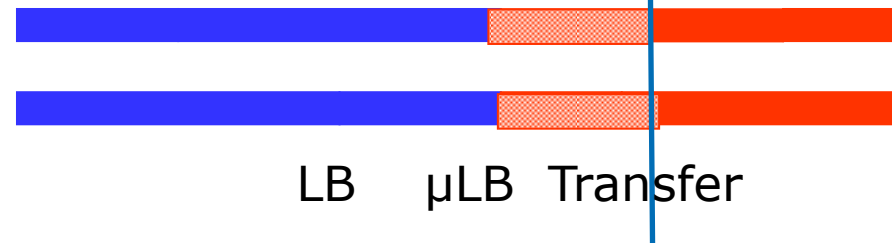
Parallel efficiency refinement: $LB * \mu LB * Transfer$

Computation

Communication



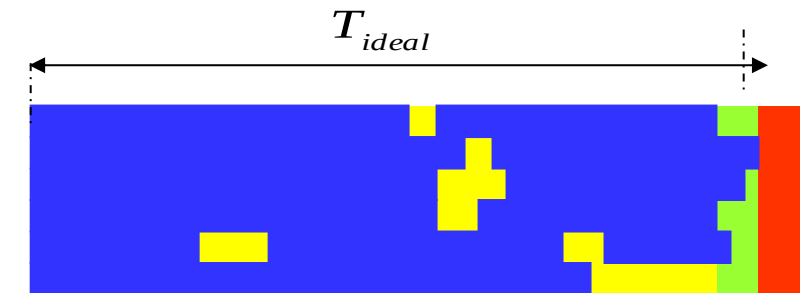
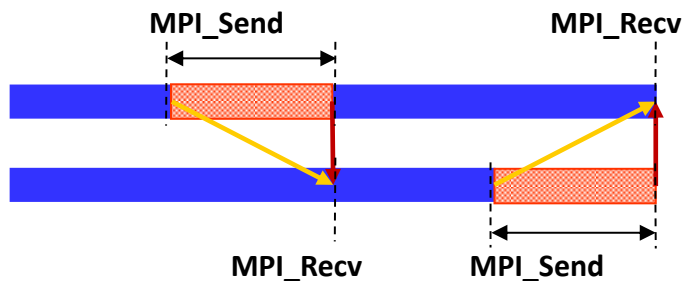
Do not blame MPI



$LB=1$

LB μLB Transfer

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



Migrating/local load imbalance
Serialization

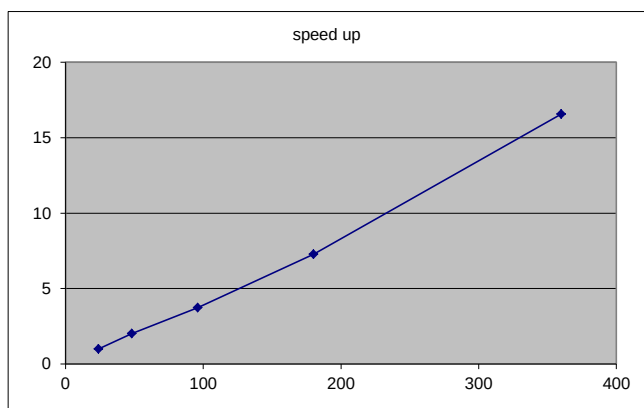
$$\mu LB = \frac{\max(T_i)}{T_{ideal}}$$

$$Transfer = \frac{T_{ideal}}{T}$$

Why scaling?

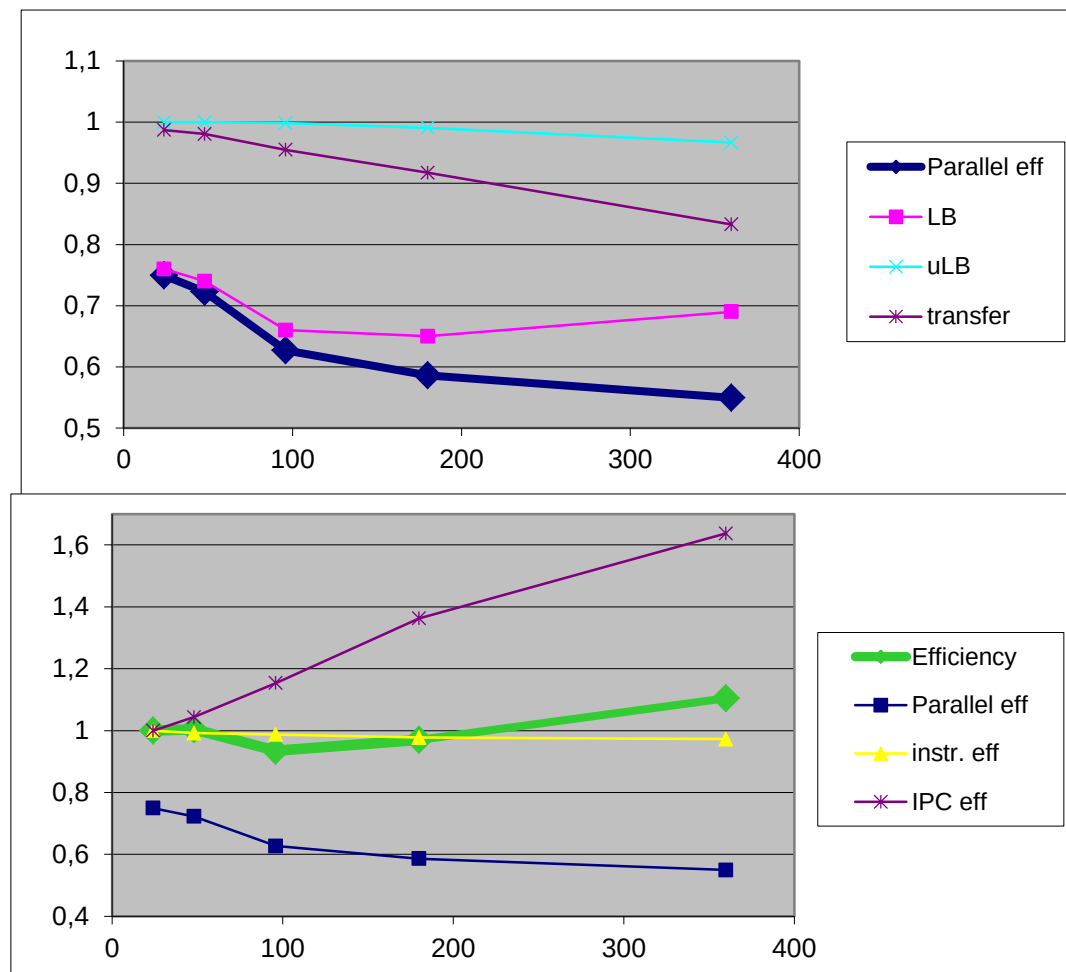
CG-POP mpi2s1D - 180x120

Good scalability !!
Should we be happy?



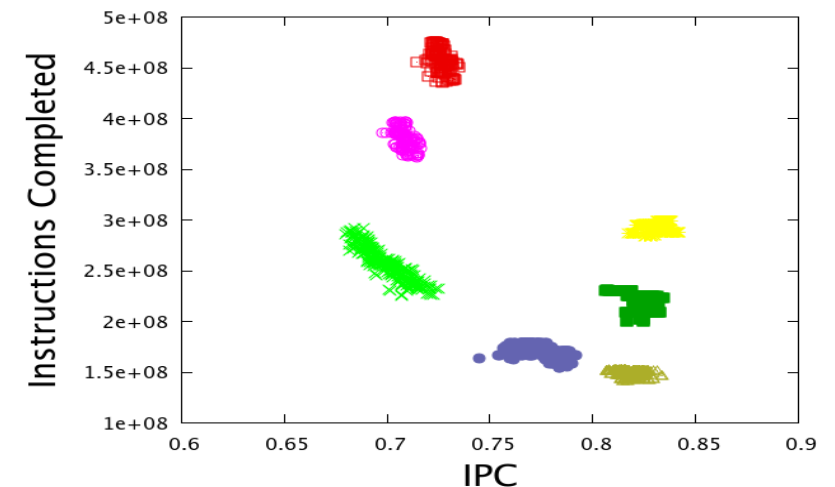
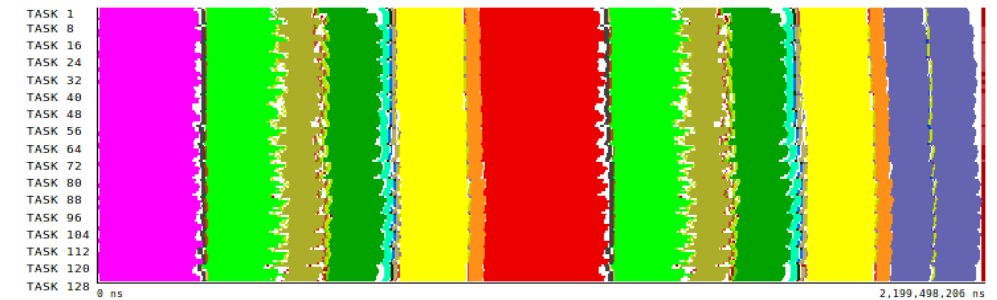
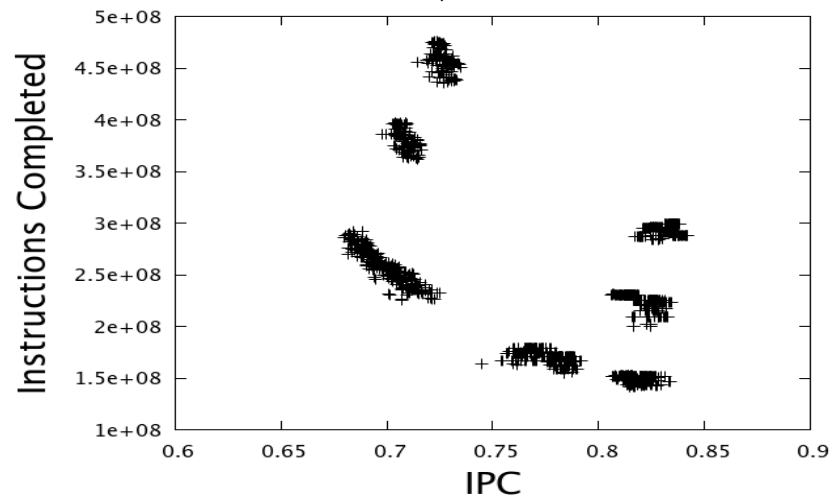
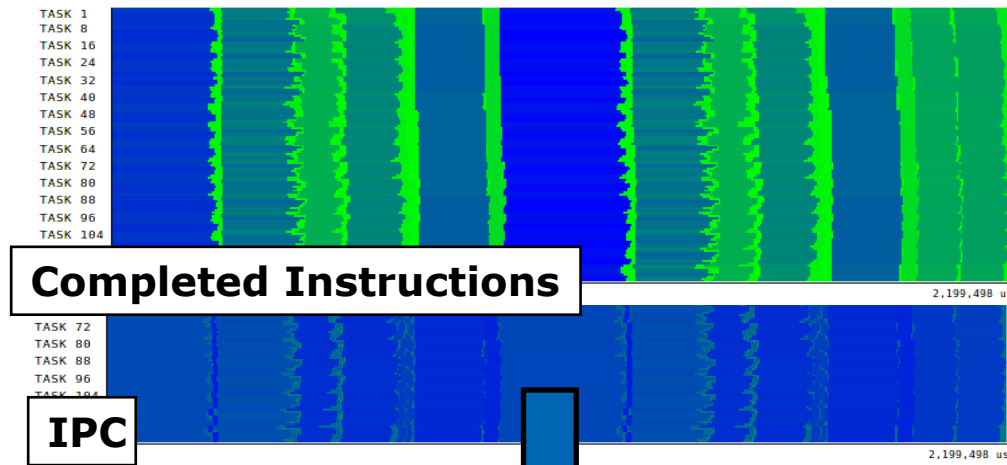
$$\eta_{\parallel} = LB * Ser * Trf$$

$$\eta = \eta_{\parallel} * \eta_{instr} * \eta_{IPC}$$

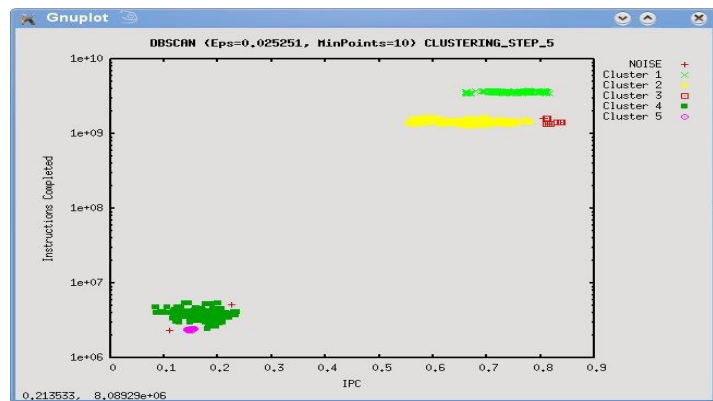


Clustering

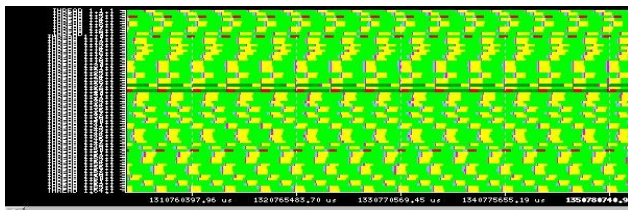
Using Clustering to identify structure



Performance @ serial computation bursts

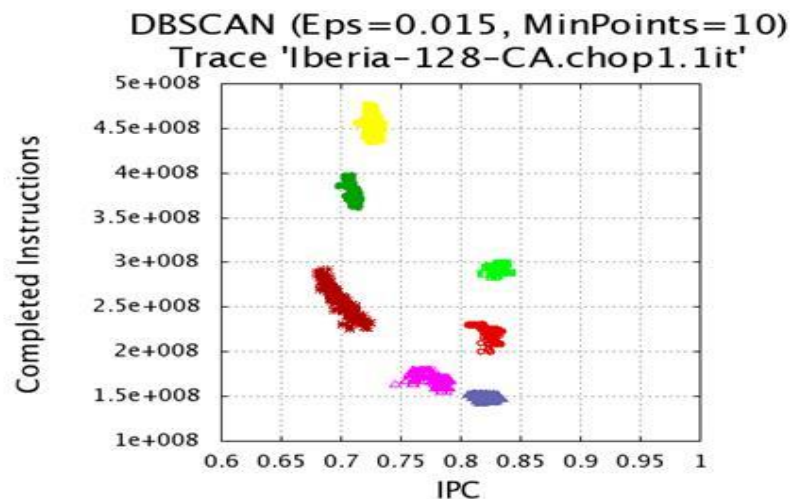


SPECfem3D

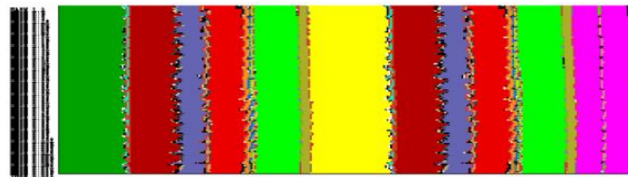


Asynchronous SPMD

Balanced #instr variability
in IPC



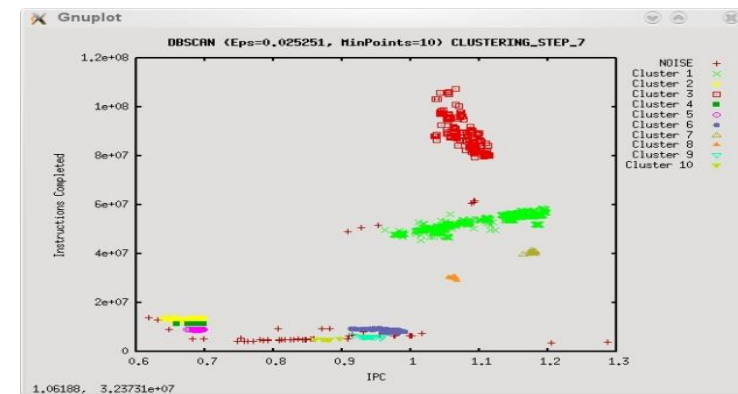
WRF 128 cores



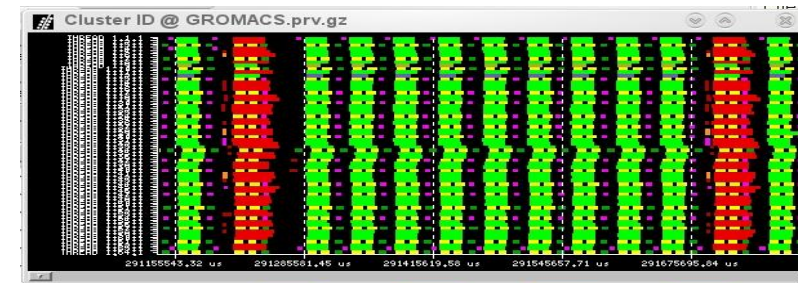
SPMD

Repeated substructure

Coupled imbalance



GROMACS

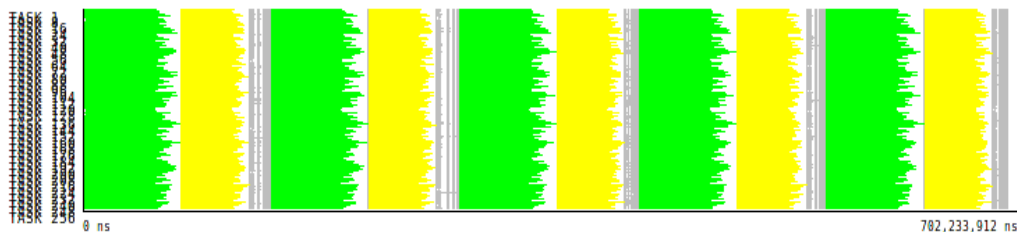


MPMD structure

Different coupled
imbalance trends

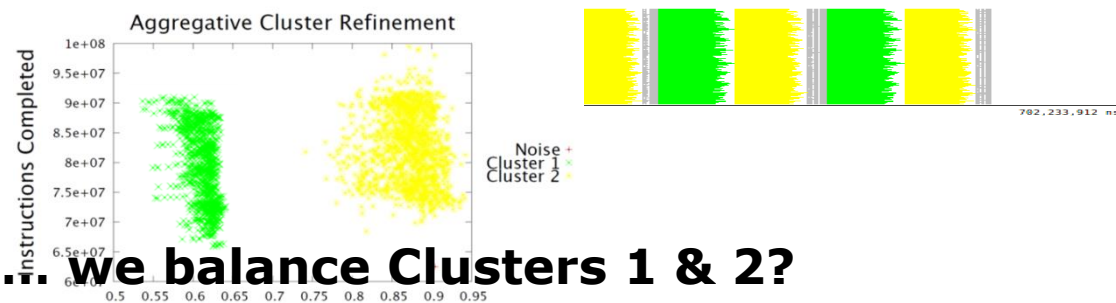
Integrating models and analytics

What if



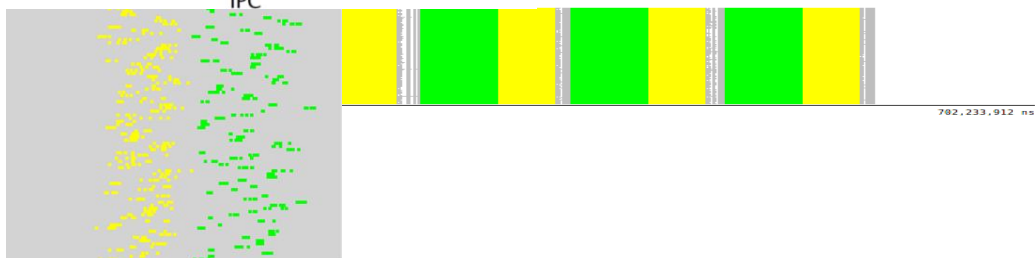
PEPC

... we increase the IPC of Cluster1?



13% gain

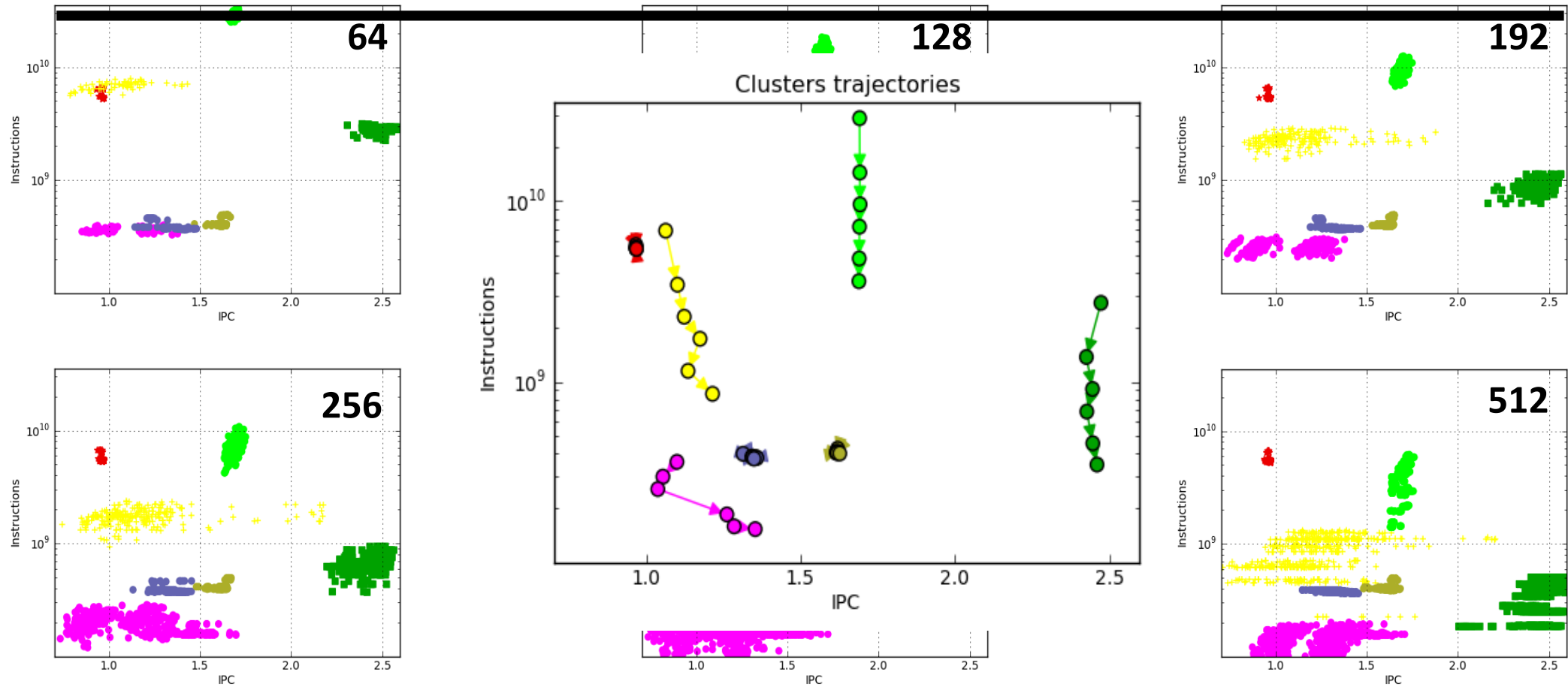
... we balance Clusters 1 & 2?



19% gain

Tracking: scalability through clustering

OpenMX (strong scale from 64 to 512 tasks)



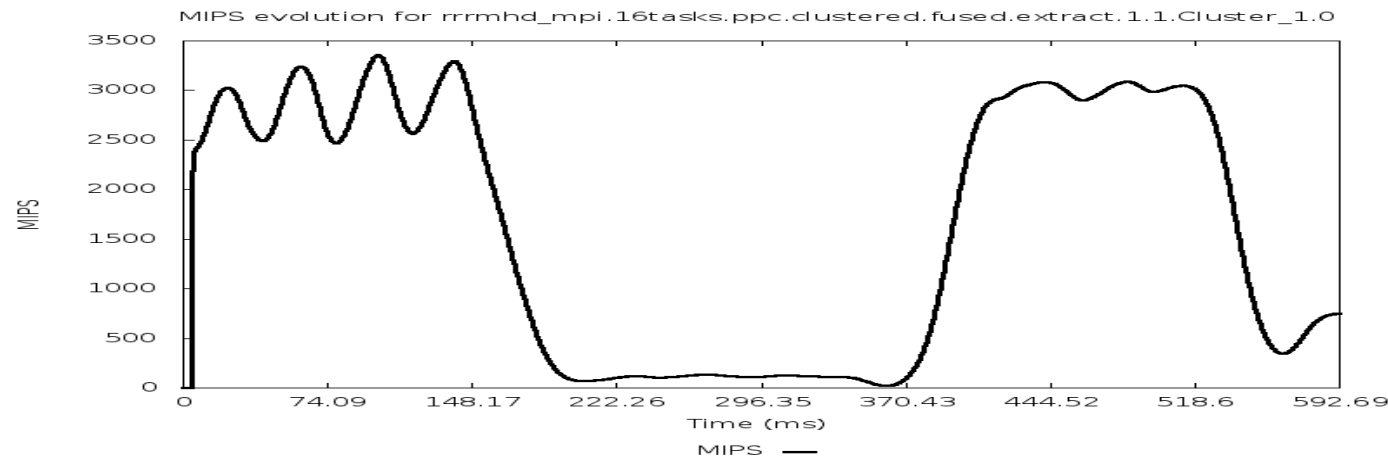
Folding

Folding: Detailed metrics evolution

Performance of a sequential region = 2000 MIPS

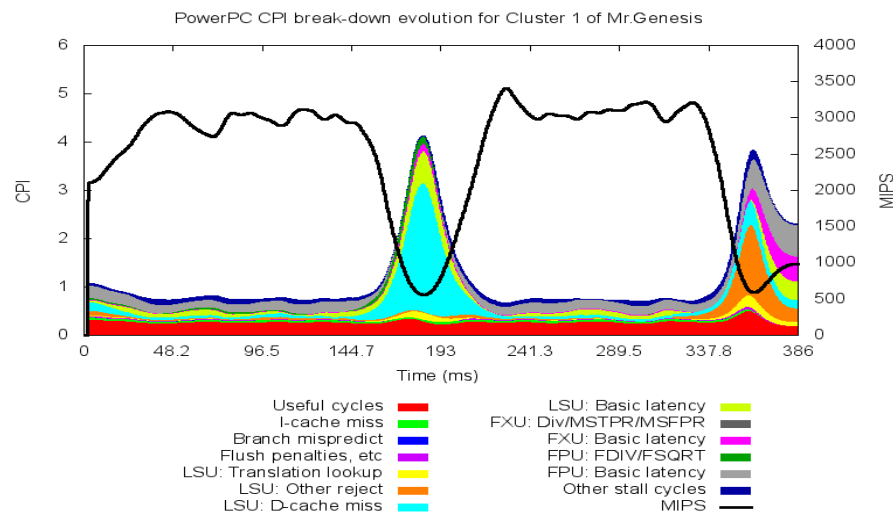
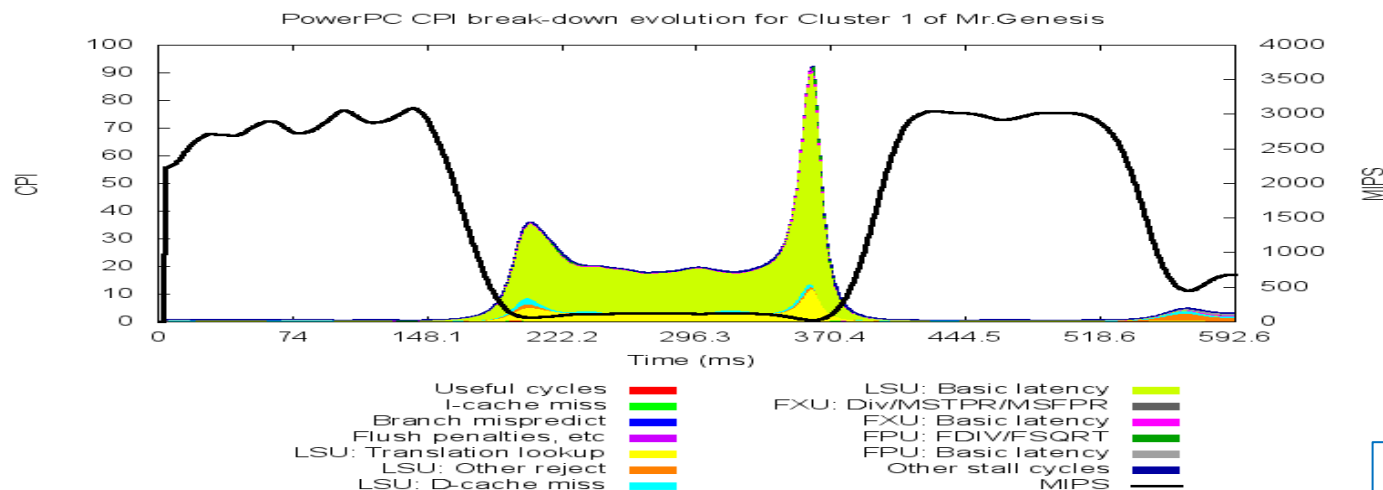
Is it good enough?

Is it easy to improve?



Folding: Instantaneous CPI stack

MRGENESIS

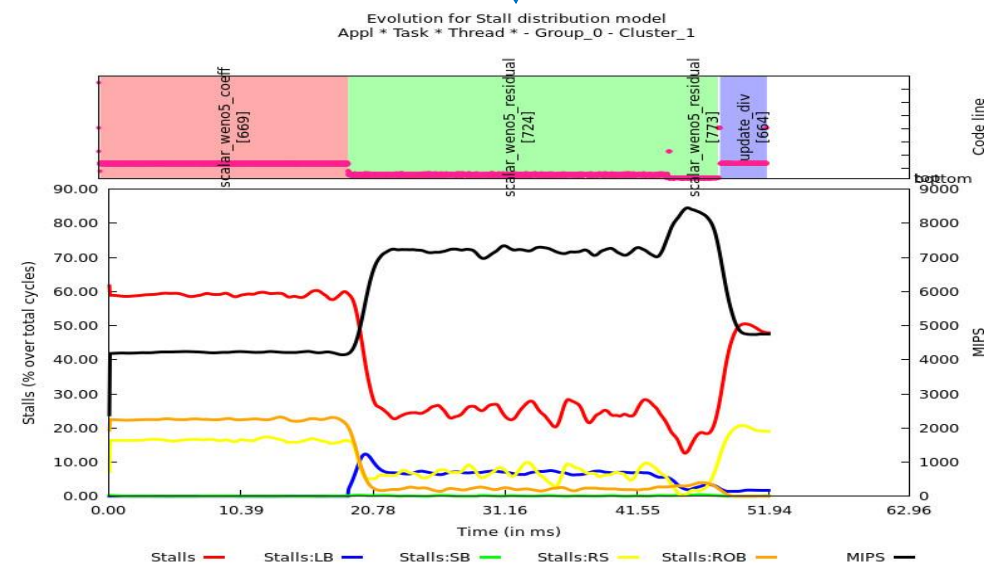
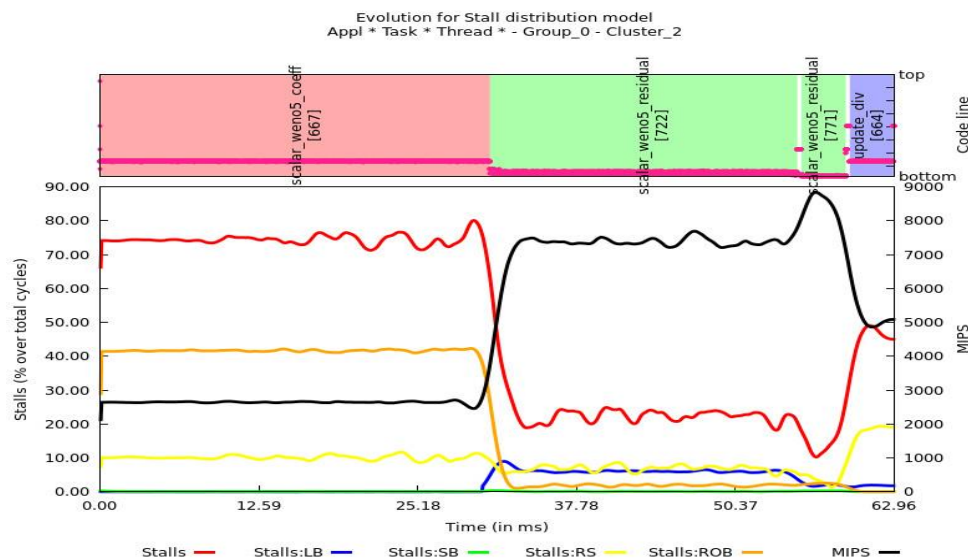


- Trivial fix.(loop interchange)
- Easy to locate?
- Next step?
- Availability of CPI stack models for production processors?
 - Provided by manufacturers?

“Blind” optimization

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

Recommendation without access to source code



Multicore effect

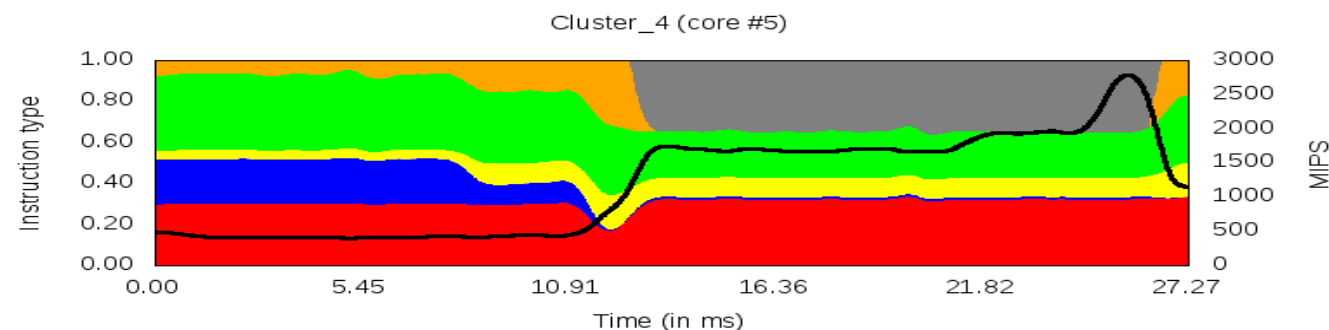
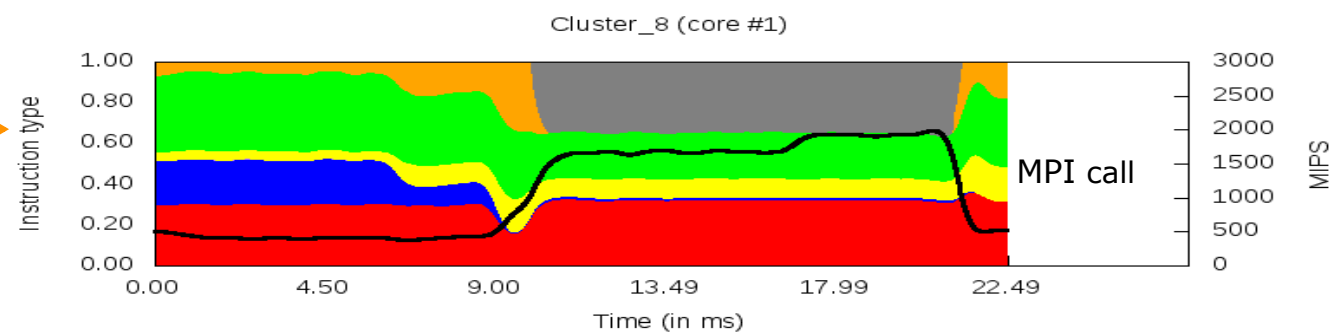
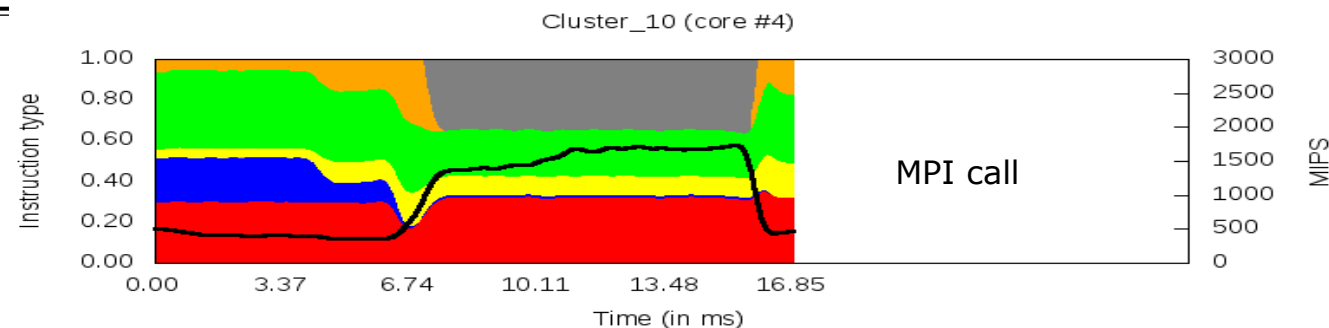
■ CG-POP: Unbalanced MPI application

- Same code
- Different duration
- Different performance

ClusterID @ cgpop.linux_icc.180x120.chop2.clustered.prv



Instruction mix model for the unbalanced CGPOP on different cores of the same hexacore chip



LD ST uncond BR cond BR FP VEC sp+dp Others MIPS

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

- 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
lulesh@leftraru	69.15	99.12	69.76
ntchem_mini(h2o)@pi	91.83	98.26	93.45
ntchem_mini(taxol, 32)@pi	92.56	94.94	97.49
nicam@pi	87.10	75.97	89.22

Demo: ntchem-mini (taxol) @ pi

