

Understanding applications using the BSC performance tools

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

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

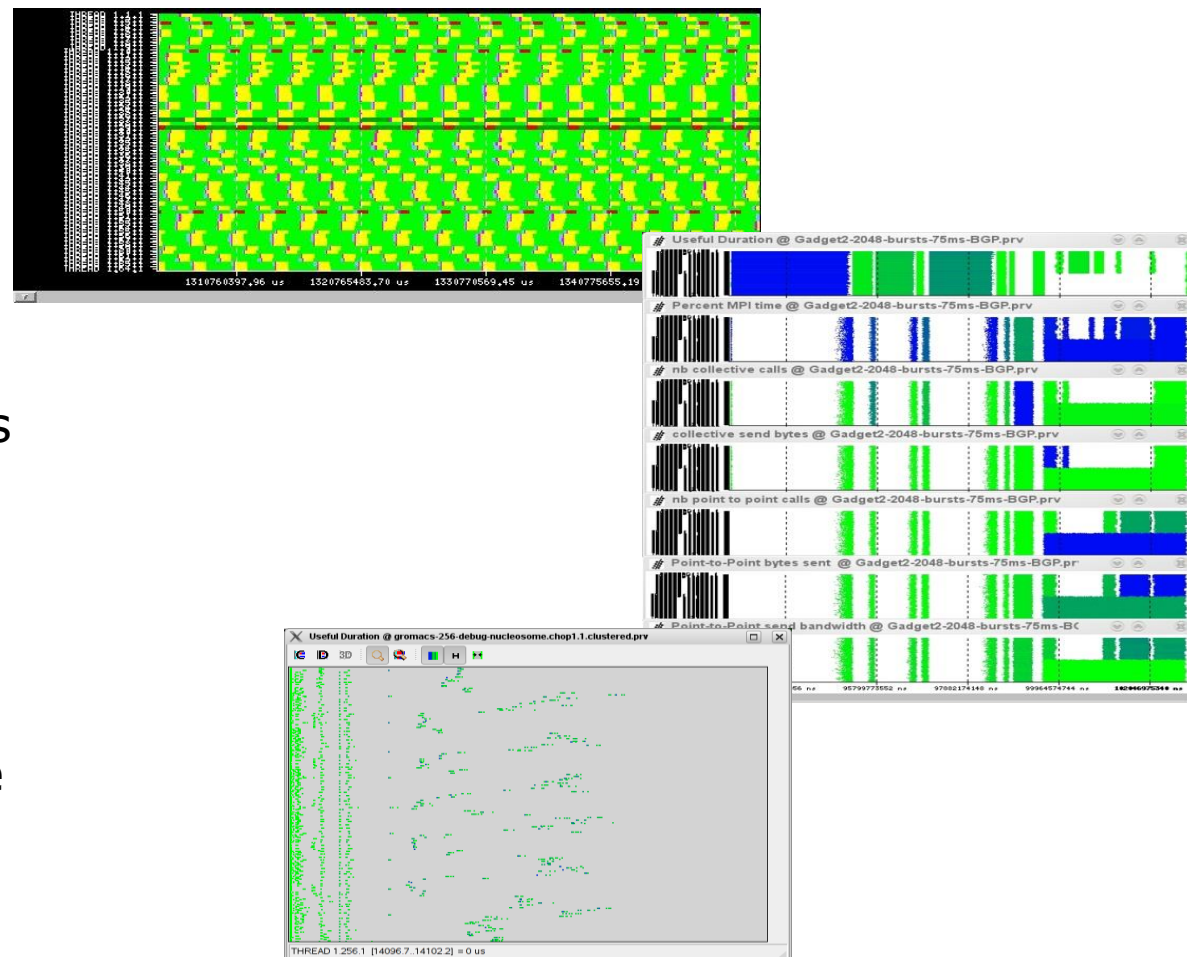
STORE

IDENTIFY



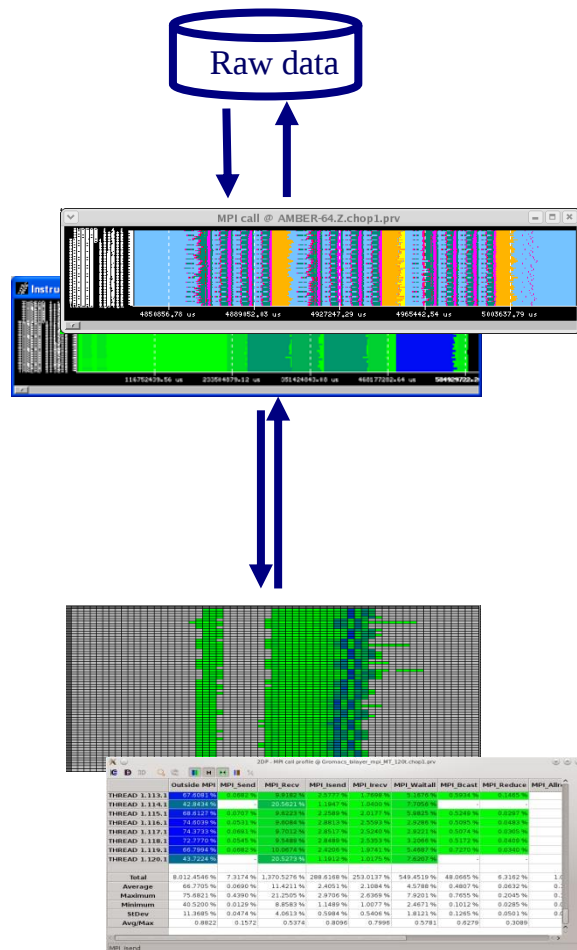
Our Tools

- 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



Trace visualization/analysis

+ trace manipulation

Goal = Flexibility

No semantics

Programmable

2/3D tables
(Statistics)

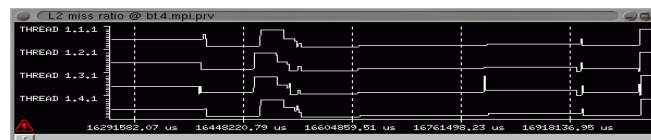
Comparative analyses

Multiple traces

Synchronize scales

Timelines

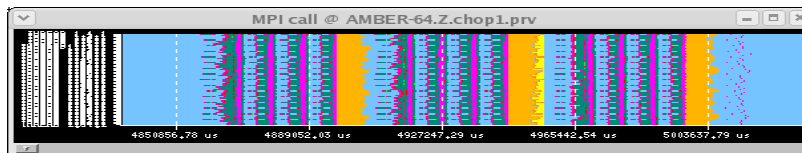
- Each window displays one view
 - Piecewise constant** function of time



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

- Types of functions

- Categorical
 - State, user function, outlined routine

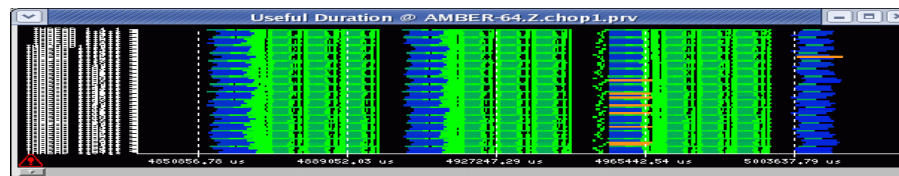


$$S_i \in [0, n] \subset N, \quad n <$$

- 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

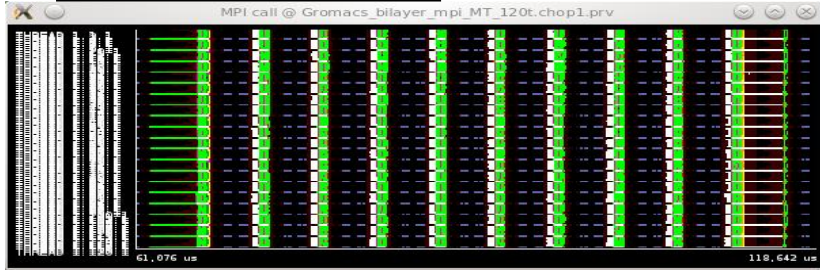


$$S_i \in R$$

Tables: Profiles, histograms, correlations

- From timelines to tables

MPI calls



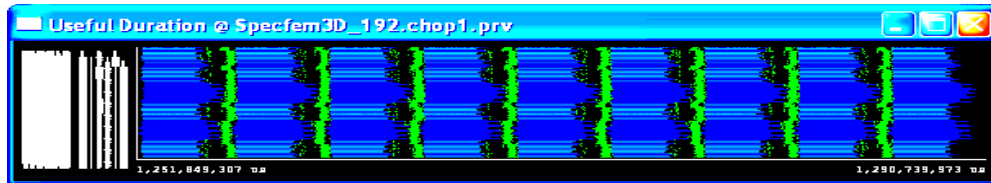
MPI calls profile

	Outside MPI	MPI_Send	MPI_Recv	MPI_Isend	MPI_Irecv	MPI_Waitall	MPI_Bcast	MPI_Reduce	MPI_Allreduce
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 %	0.5065 %	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 %	-	24.8273 %	1.1012 %	1.634 %	-	-	-	
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				

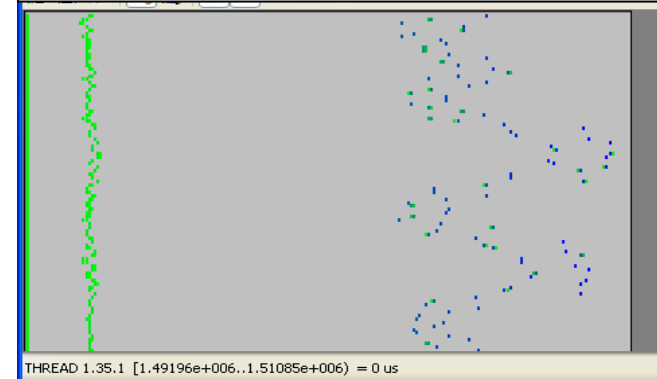
MPI_Isend

THREAD 1.107.1 MPI_Sendrecv = 11.3412 %

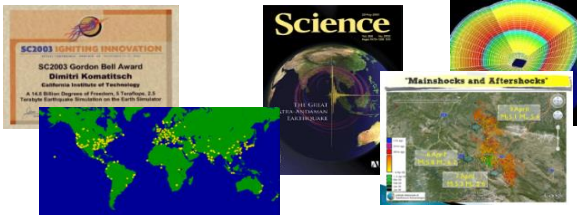
Useful Duration



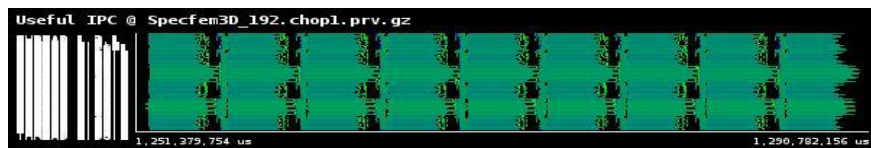
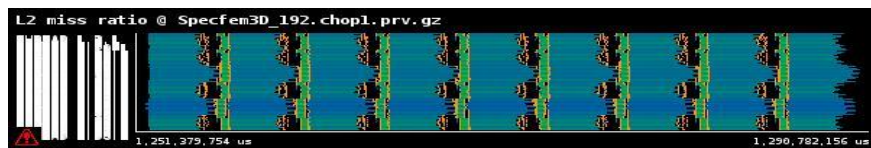
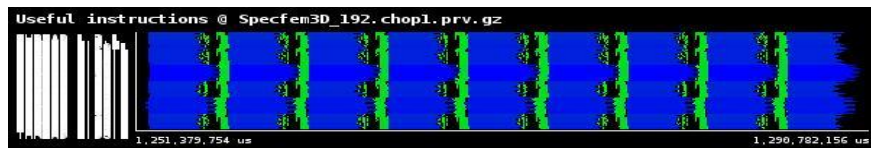
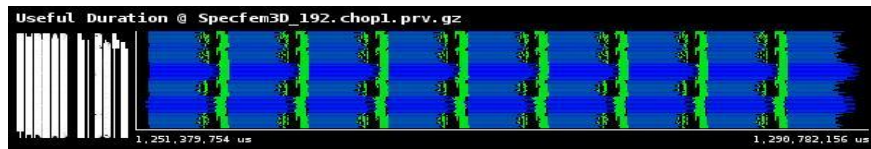
Histogram Useful Duration



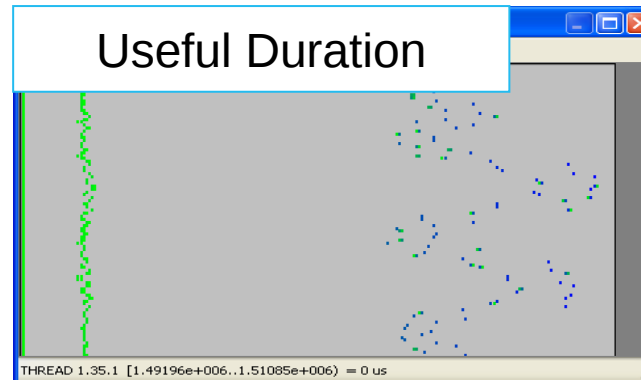
Analyzing variability through histograms and timelines



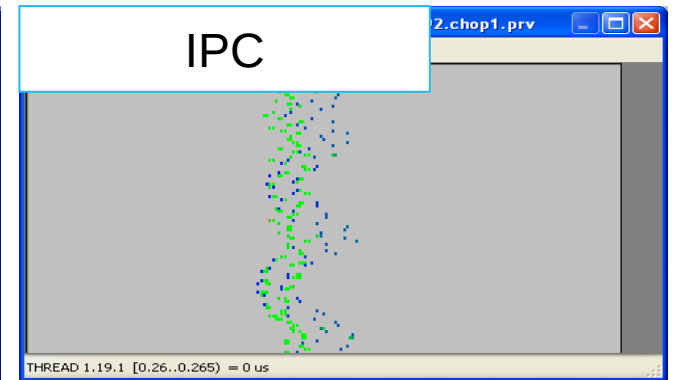
SPECFEM3D



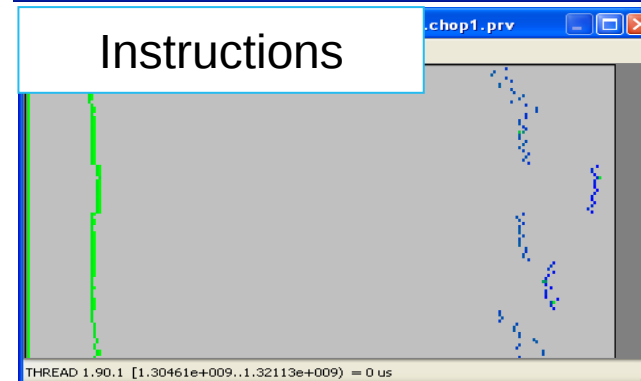
Useful Duration



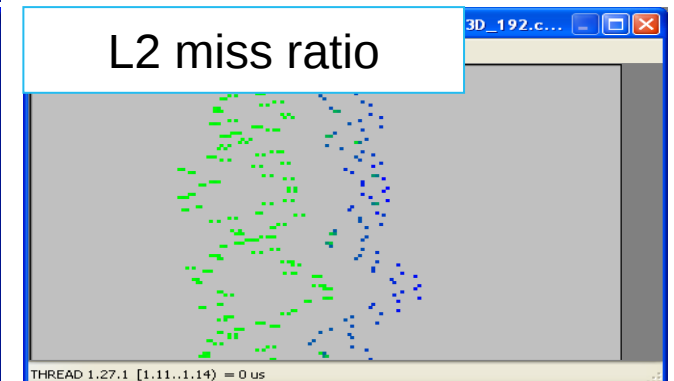
IPC



Instructions

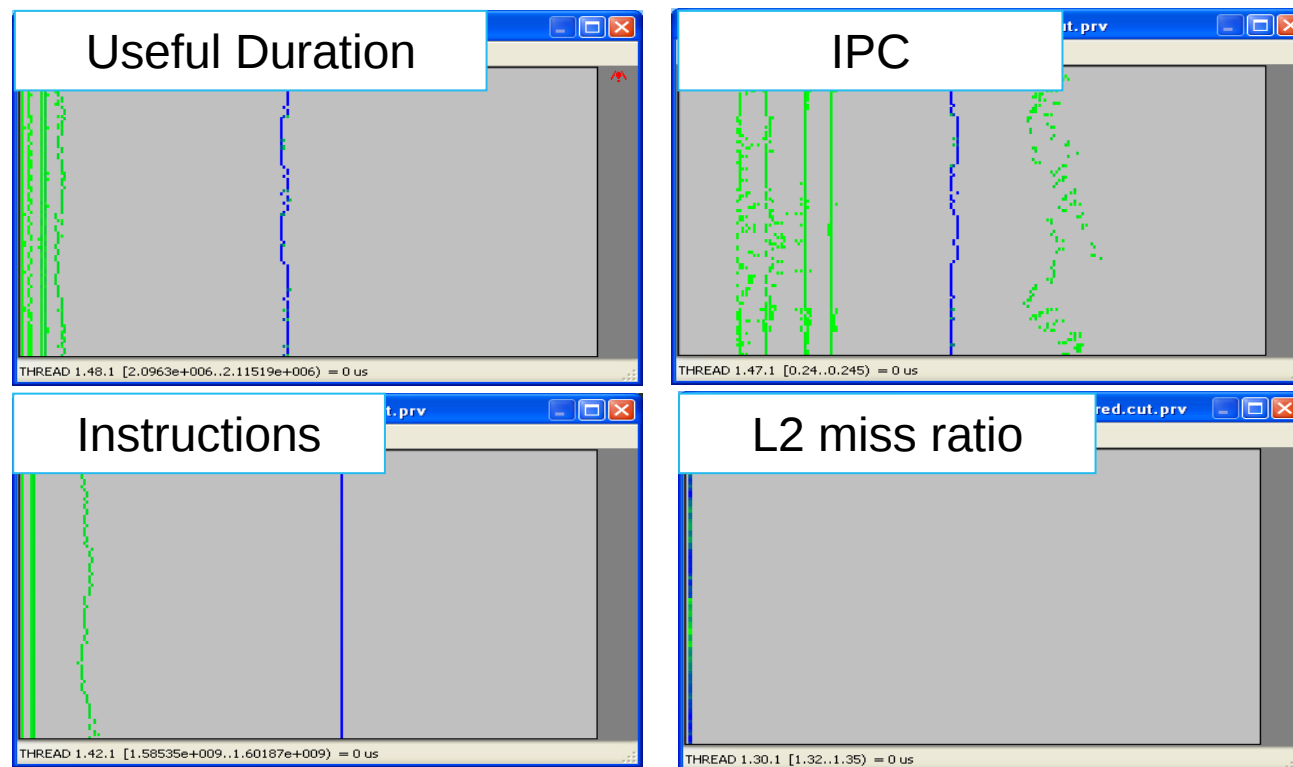


L2 miss ratio



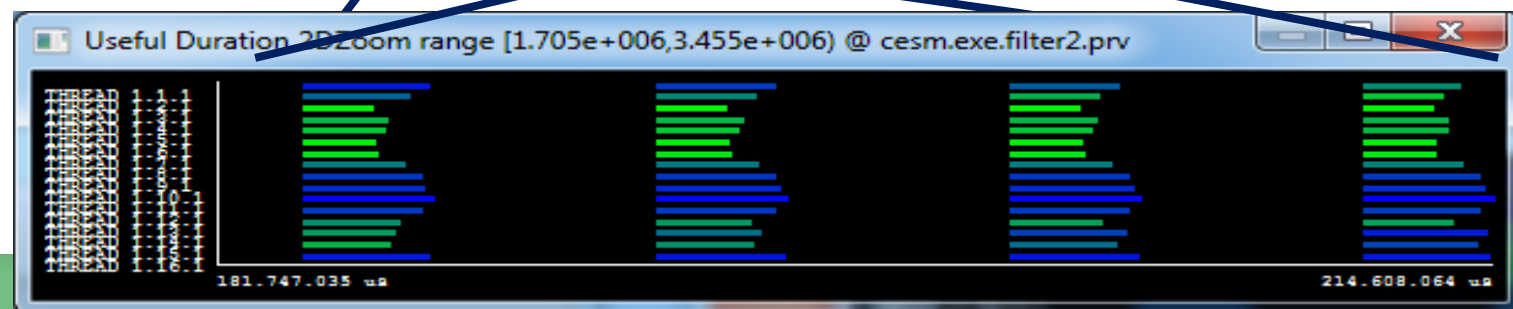
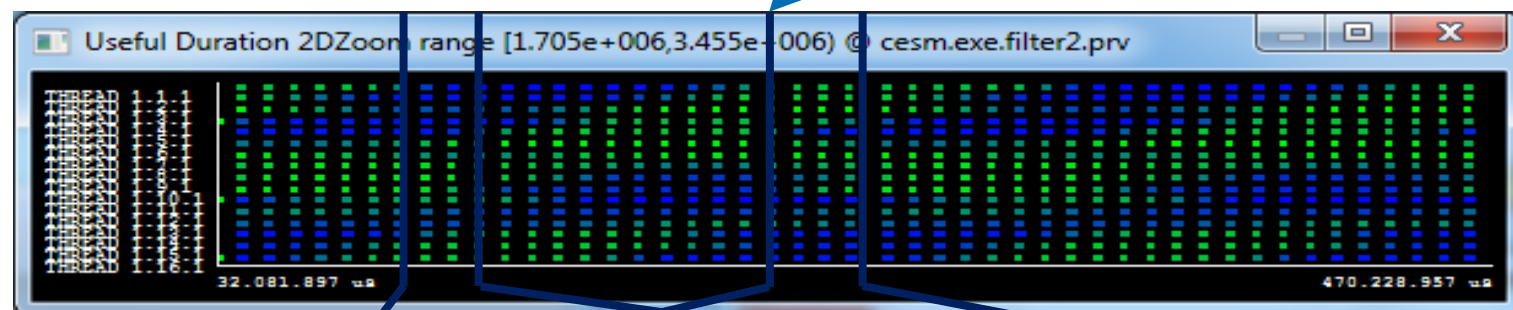
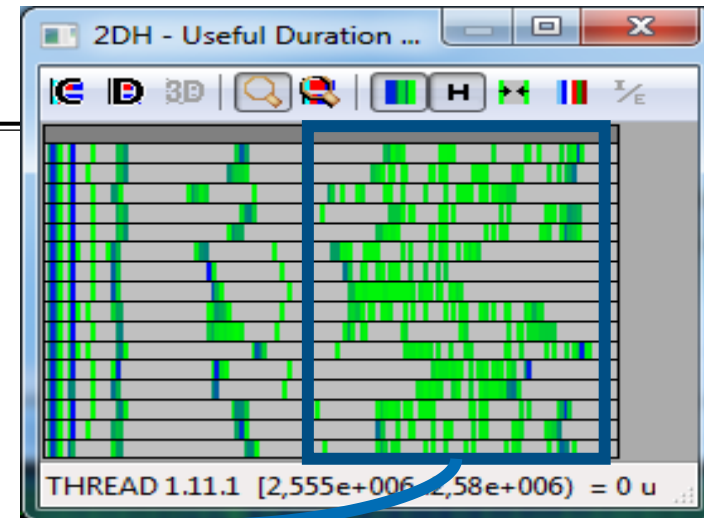
Analyzing variability through histograms and timelines

- By the way: six months later



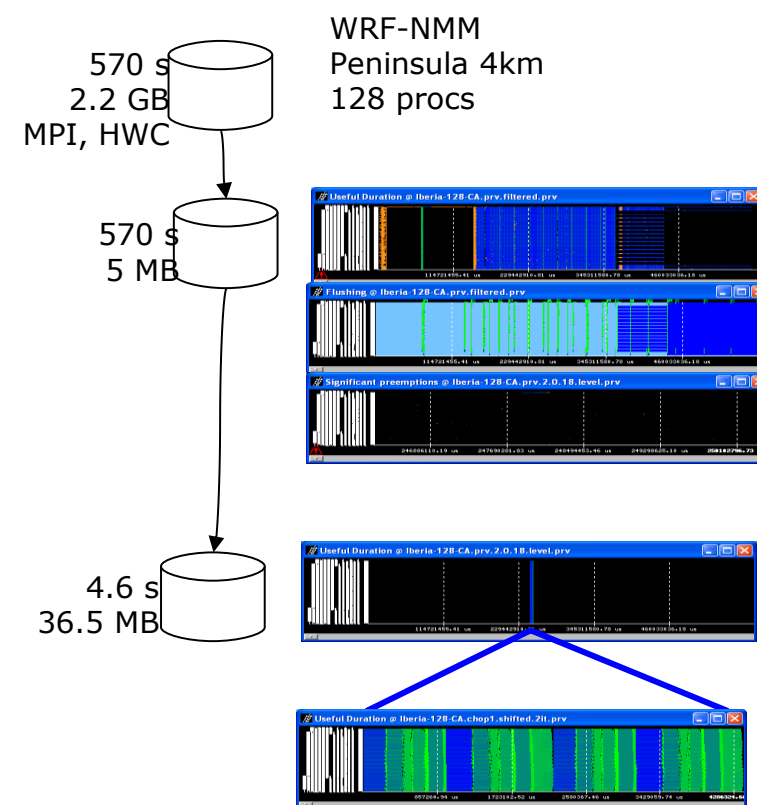
Variability ... is everywhere

- 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

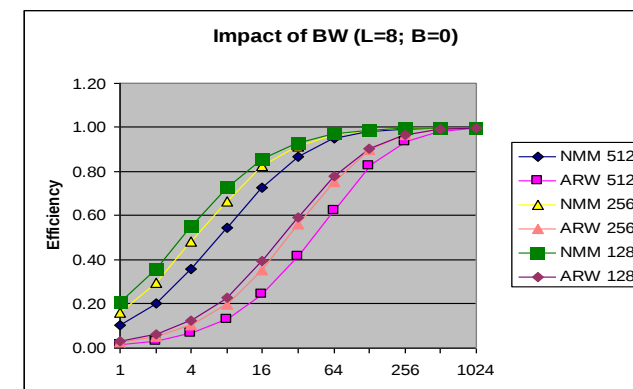
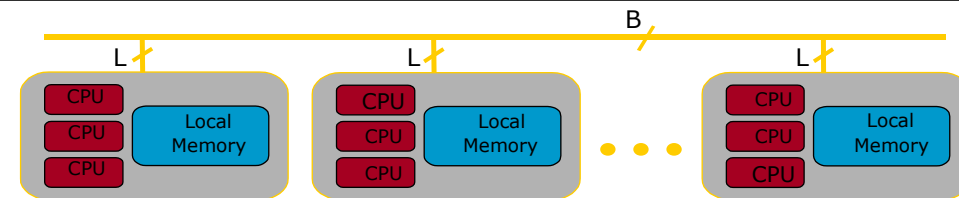
- Data handling/summarization capability
- Filtering
 - Subset of records in original trace
 - By duration, type, value,...
 - Filtered trace IS a paraver trace and can be analysed with the same cfgs (as long as needed data kept)
- Cutting
 - All records in a given time interval
 - Only some processes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

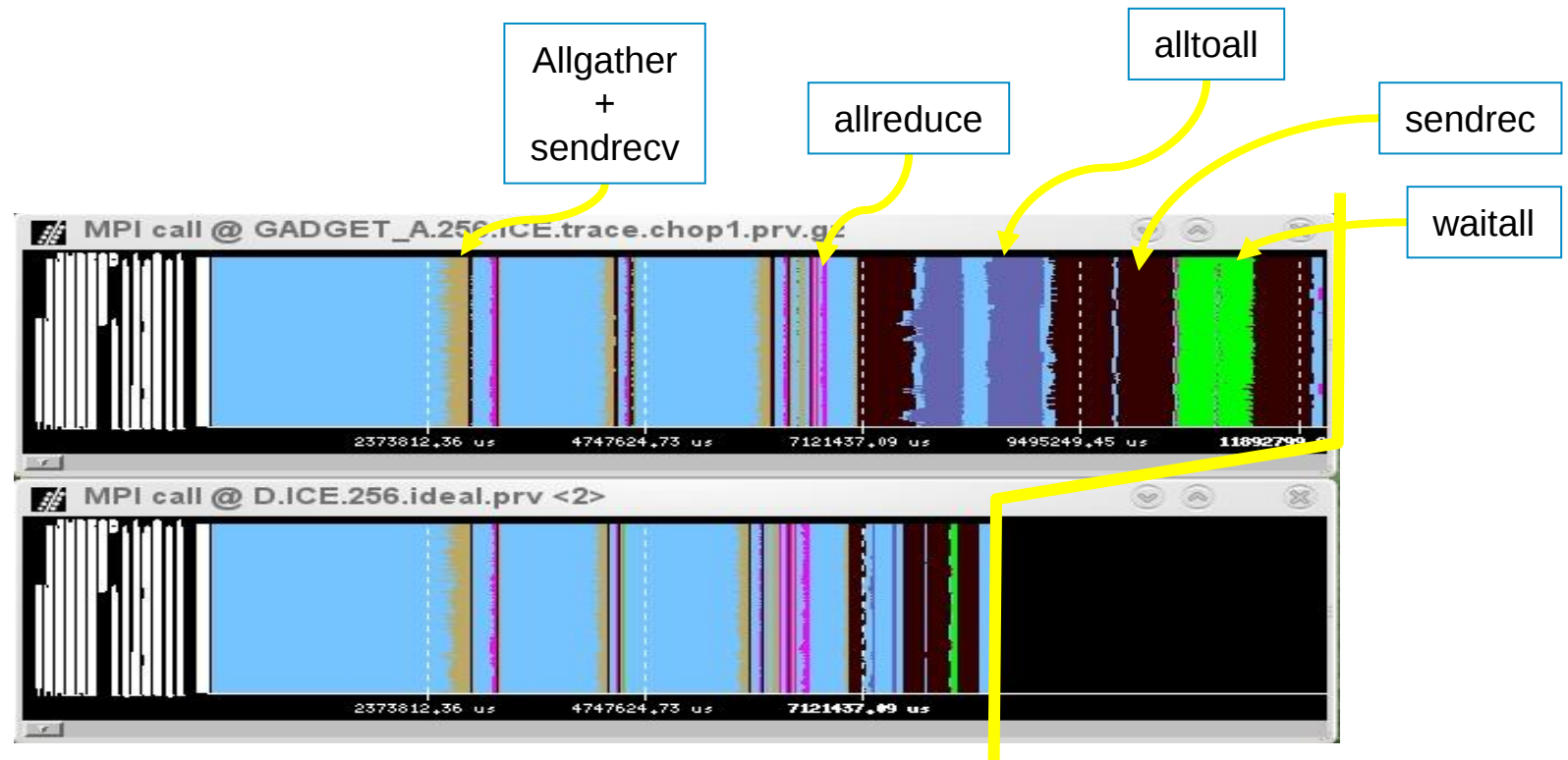
Ideal machine

- 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

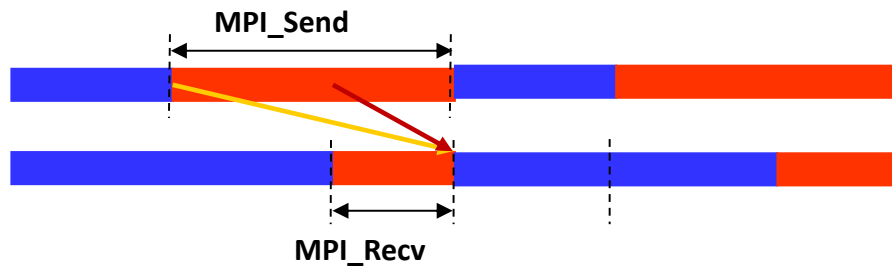


Models

Parallel efficiency model

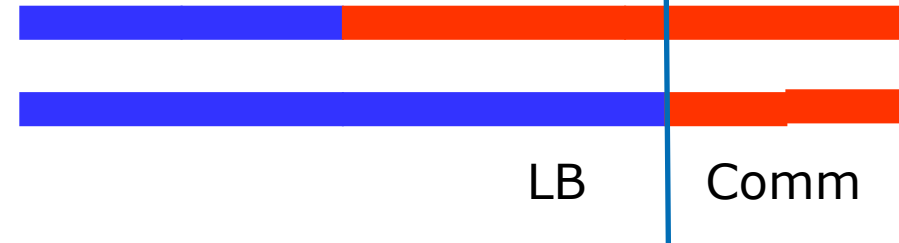
Computation

Communication



- Parallel efficiency = LB eff * Comm eff

Do not blame MPI



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

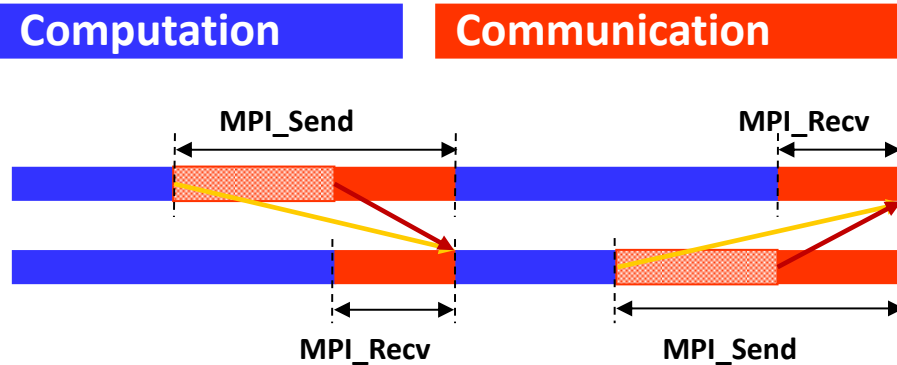
	Outside MPI	MPI_Recv	MPI_Isend	MPI_Irecv
THREAD 1.130.1	87,95 %	9,09 %	0,00 %	0,02 %
THREAD 1.131.1	88,16 %	9,09 %	0,00 %	0,02 %
THREAD 1.132.1	88,18 %	9,09 %	0,00 %	0,02 %
THREAD 1.133.1	88,18 %	9,09 %	0,00 %	0,02 %
Total	9,309,74 %	306,53 %	1,411,18 %	3,83 %
Average	69,00 %	2,30 %	10,69 %	0,03 %
Maximum	88,18 %	67,62 %	54,97 %	
Minimum	30,67 %	0,00 %	0,00 %	
StDev	15,27 %	6,06 %	21,40 %	0,00 %
Avg/Max	0,77	0,03	0,19	0,81

η

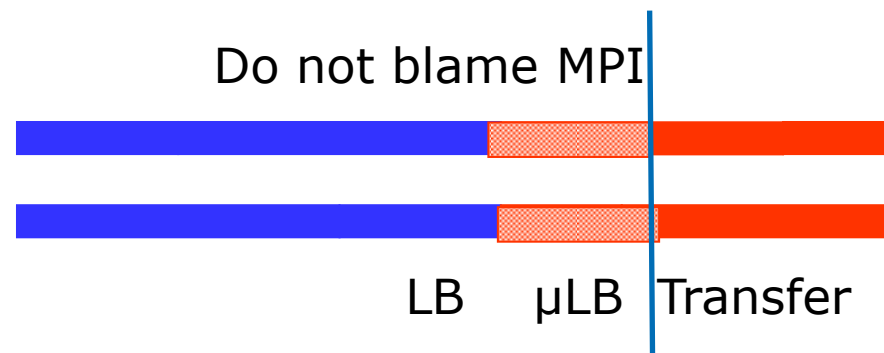
CommEff

LB

Parallel efficiency refinement: $LB * \mu LB * \text{Transfer}$

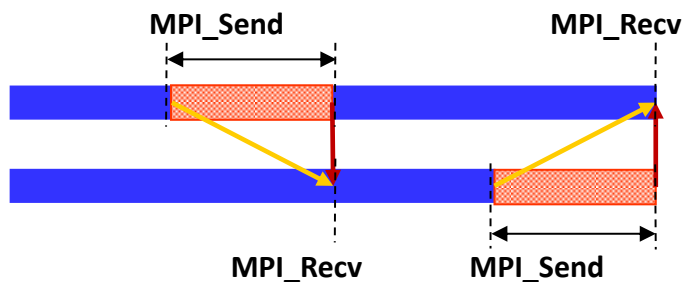


Do not blame MPI



$$LB=1$$

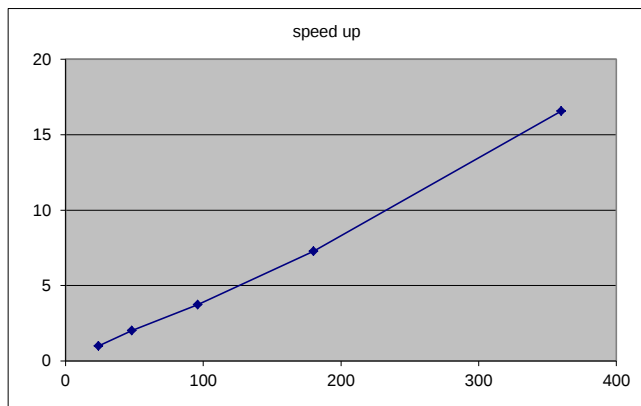
- Serializations / dependences (μLB)
- Dimemas ideal network $\rightarrow$ Transfer (efficiency) = 1



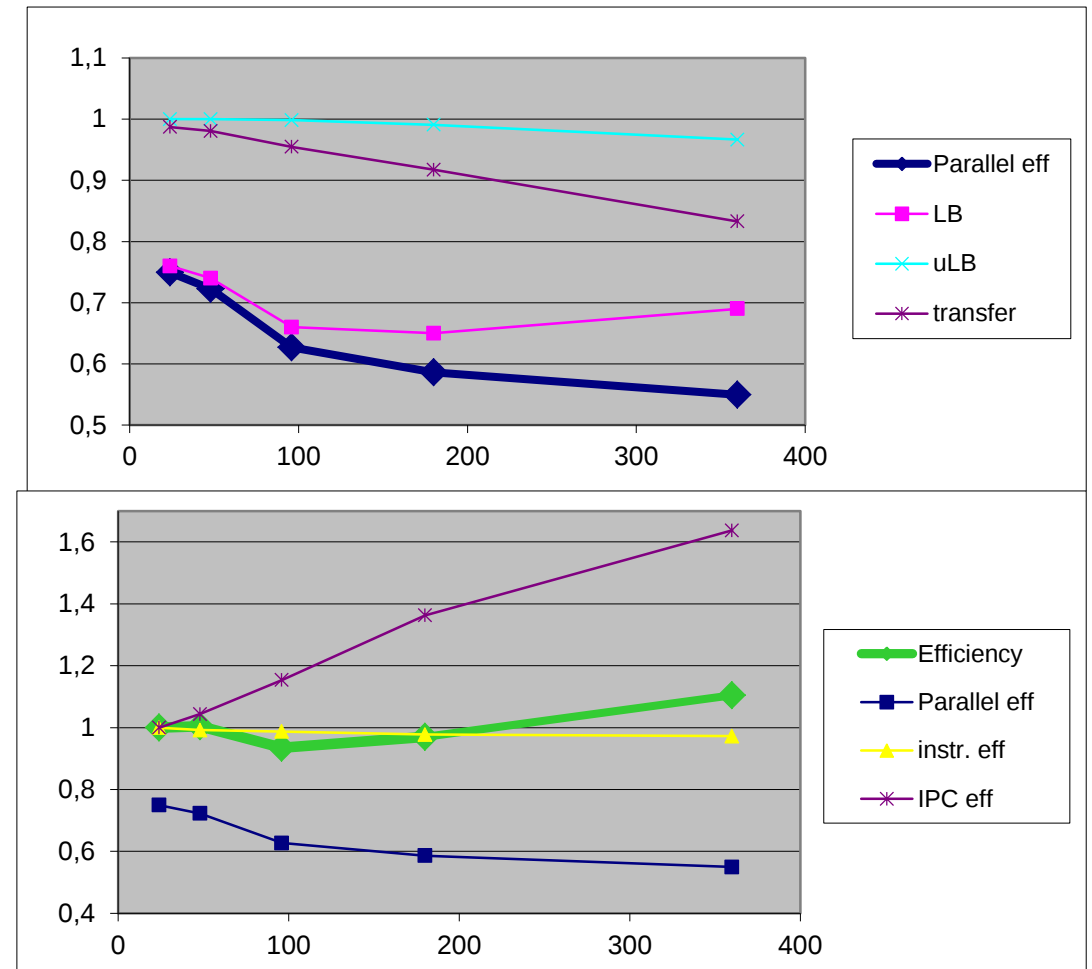
Why scaling?

CG-POP mpi2s1D - 180x120

Good scalability !!
Should we be happy?

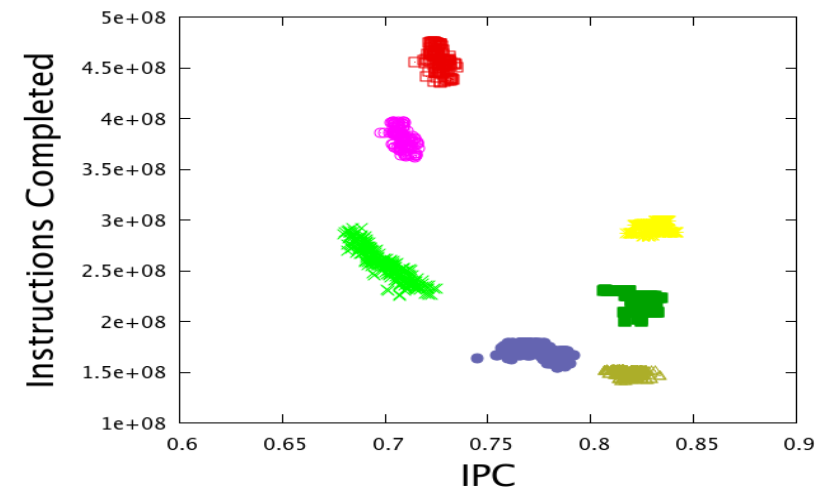
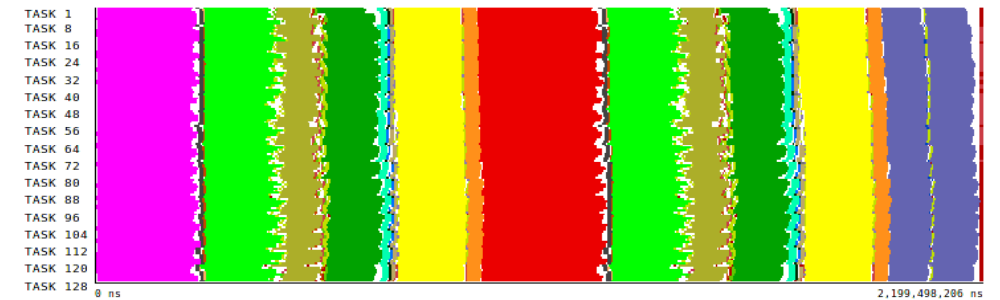
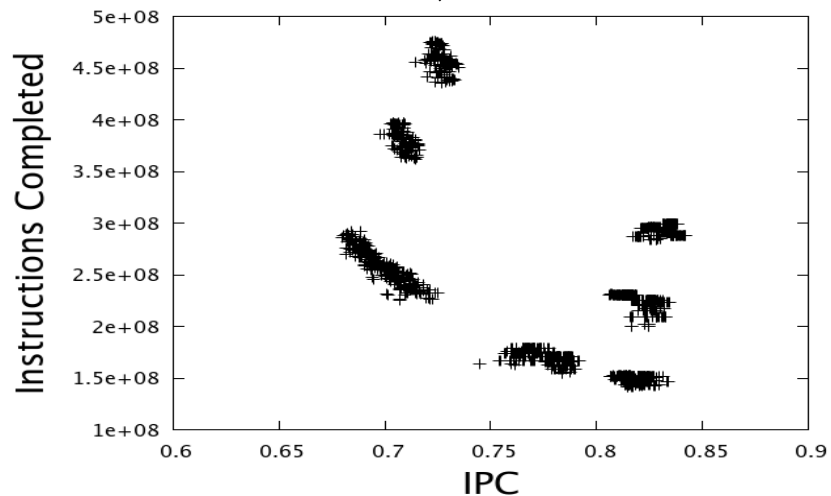
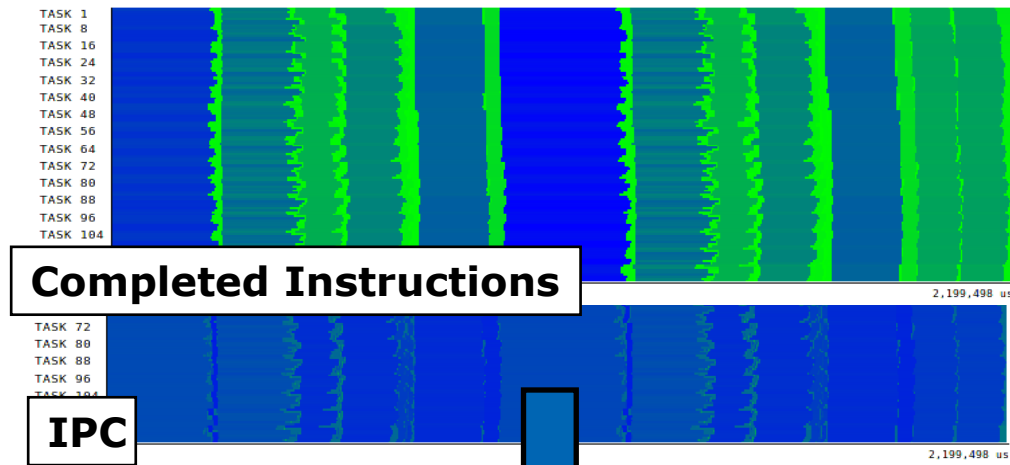


$$\eta_{\parallel} = LB * Ser * Trf$$



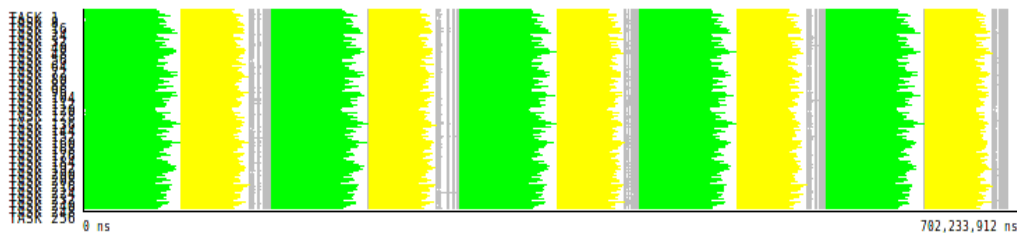
Performance Analytics

Using Clustering to identify structure



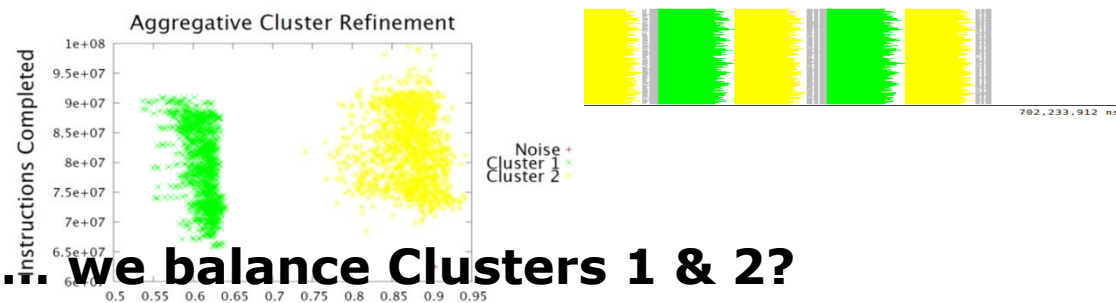
Integrating models and analytics

What if



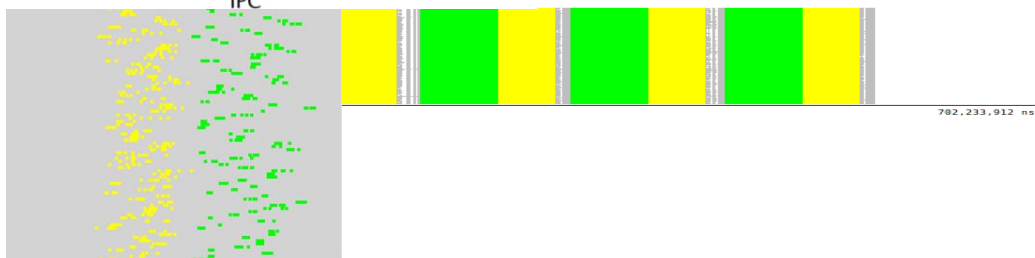
PEPC

... we increase the IPC of Cluster1?



13% gain

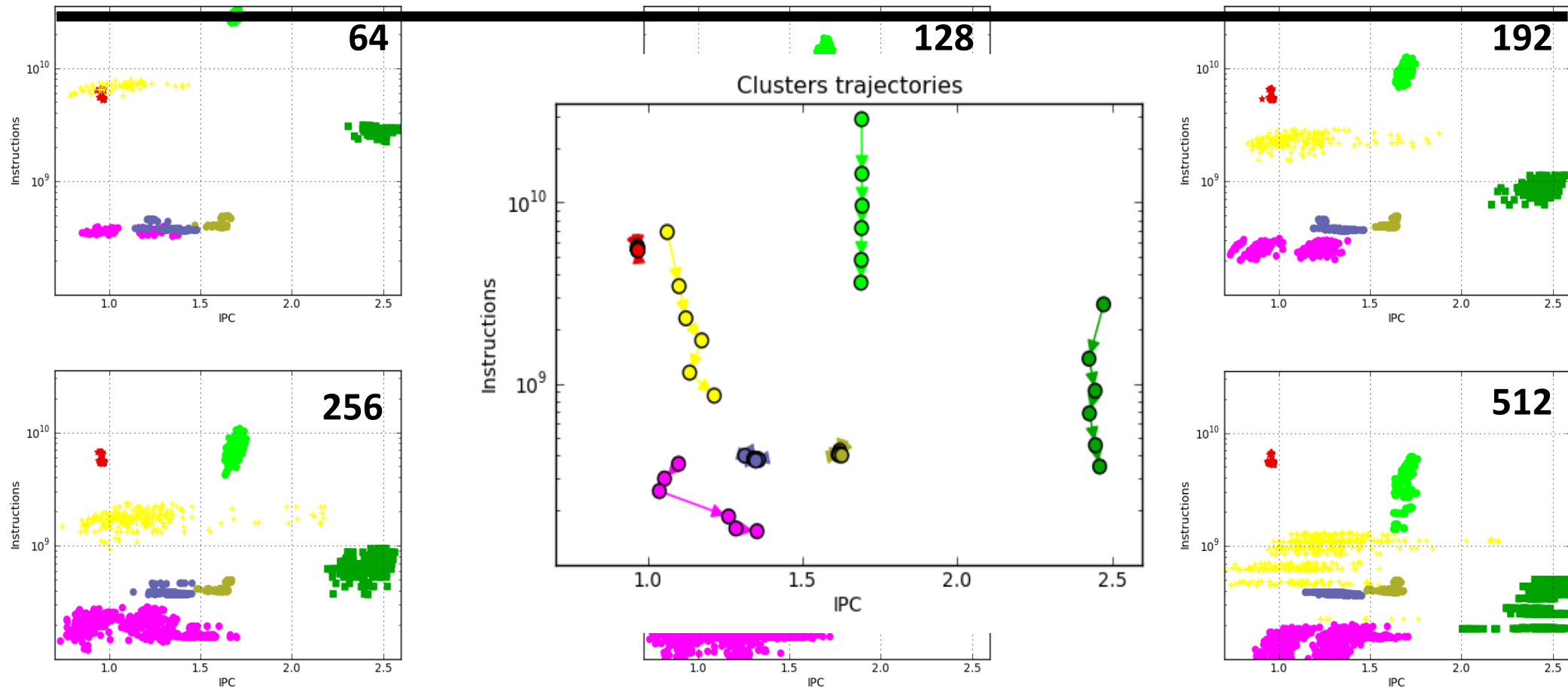
... we balance Clusters 1 & 2?



19% gain

Tracking: scalability through clustering

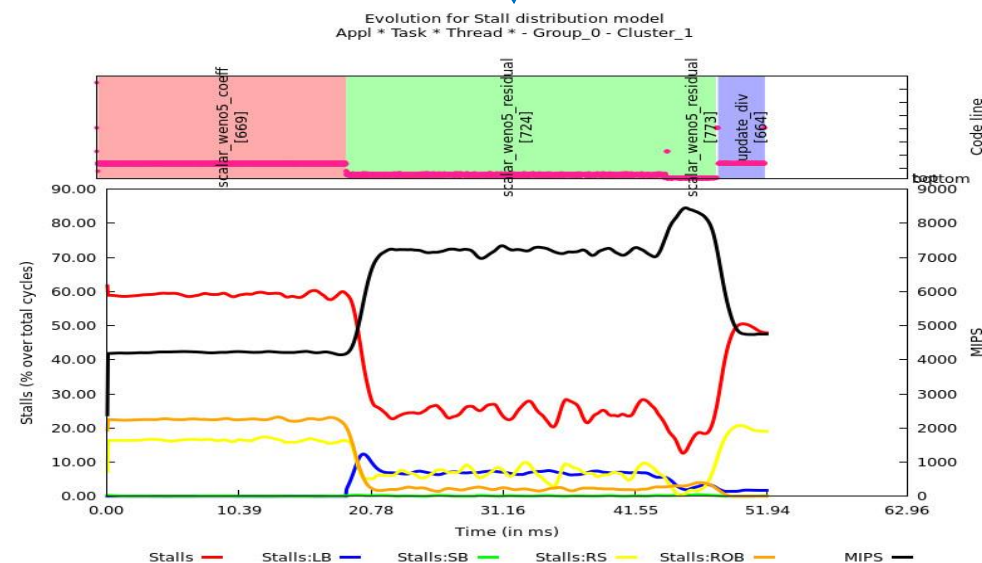
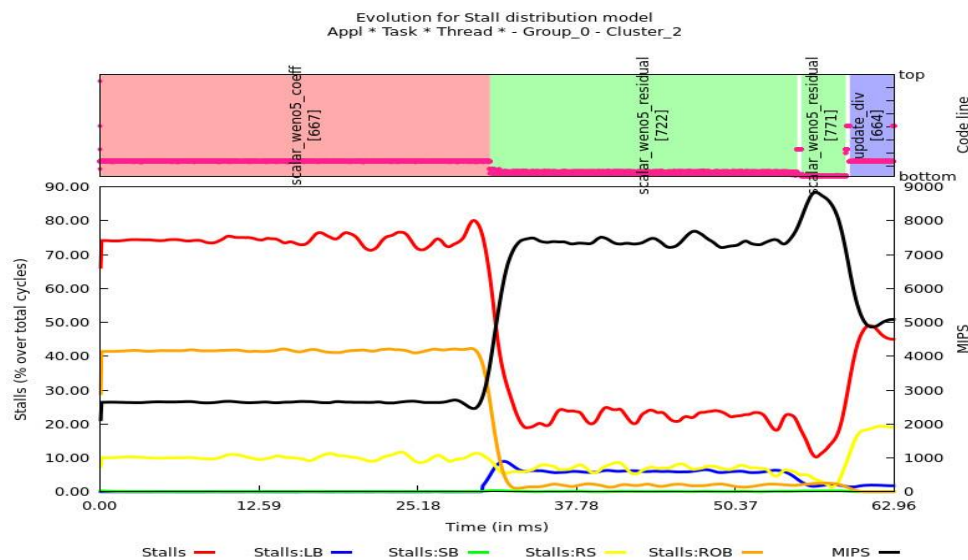
OpenMX (strong scale from 64 to 512 tasks)



“Blind” optimization

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

Recommendation without access to source code



Multicore effect

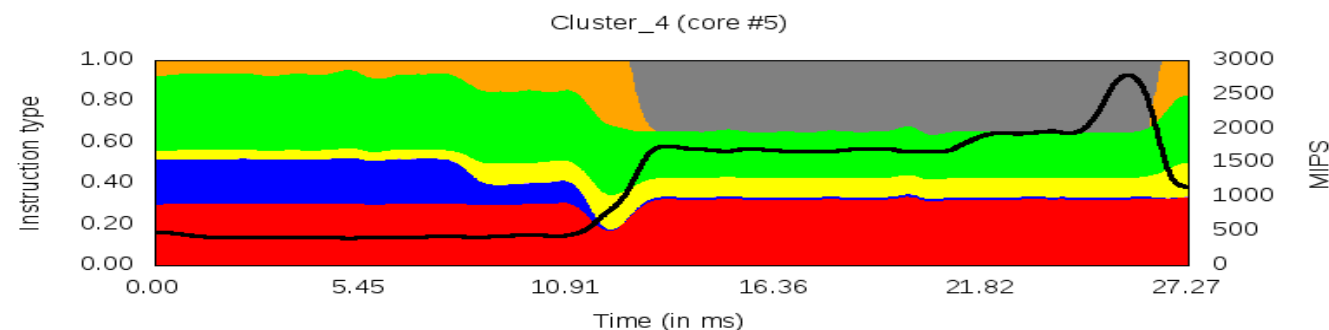
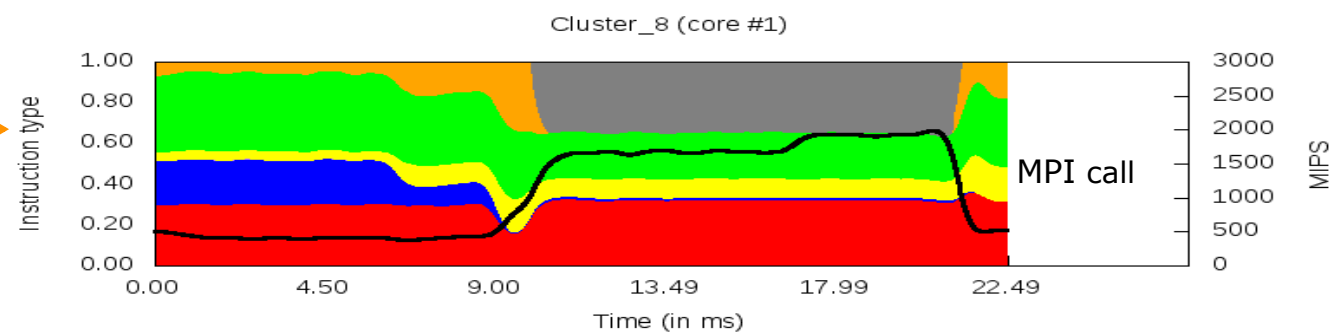
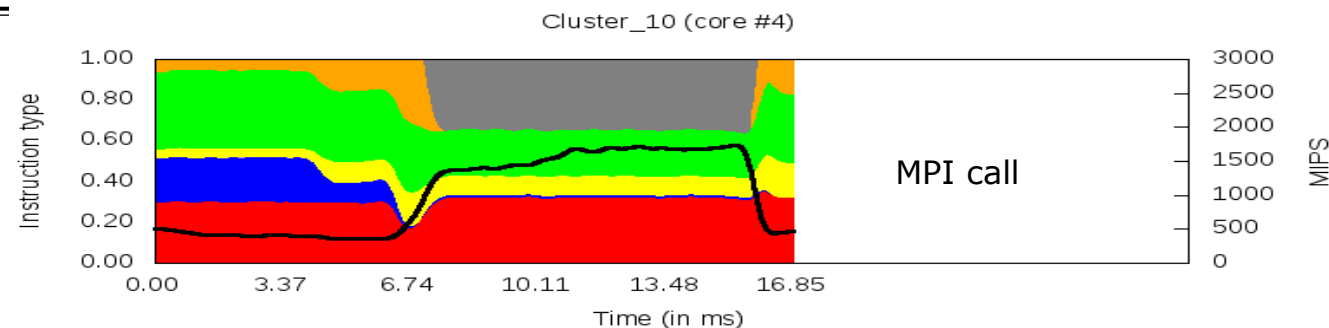
■ CG-POP: Unbalanced MPI application

- Same code
- Different duration
- Different performance

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
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