

Understanding applications using the BSC performance tools

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

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

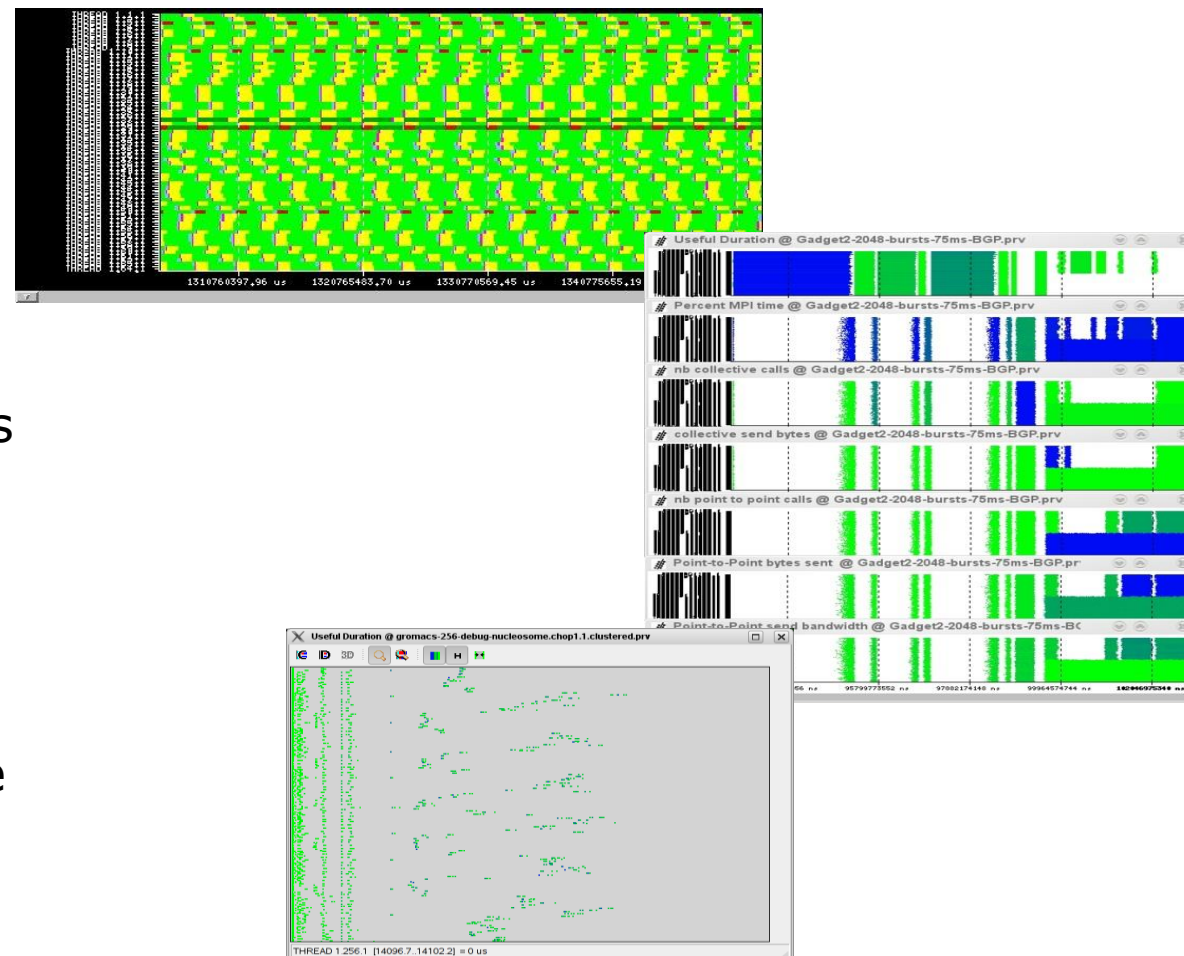
STORE

IDENTIFY



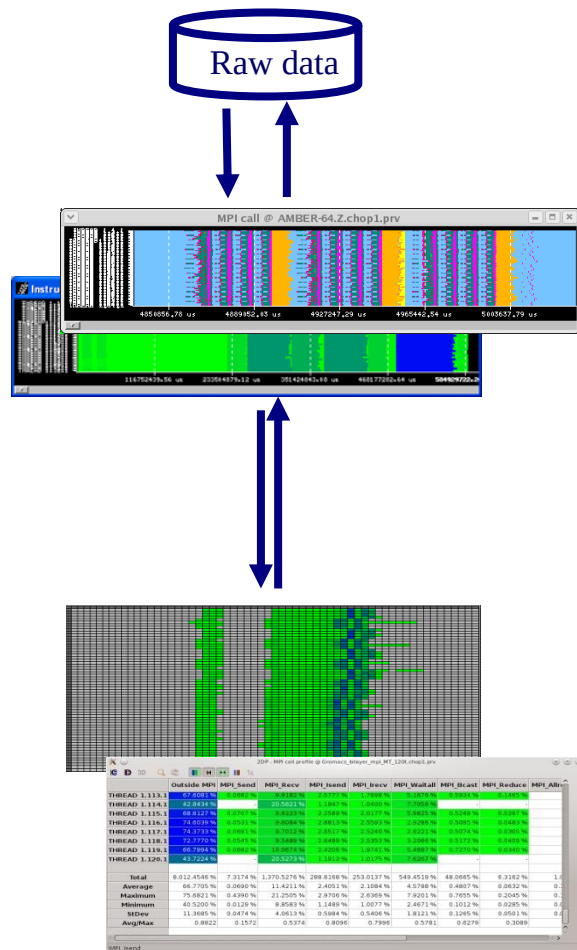
Our Tools

- Since 1991
- Based on traces
- Open Source
 - <http://tools.bsc.es>
- 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

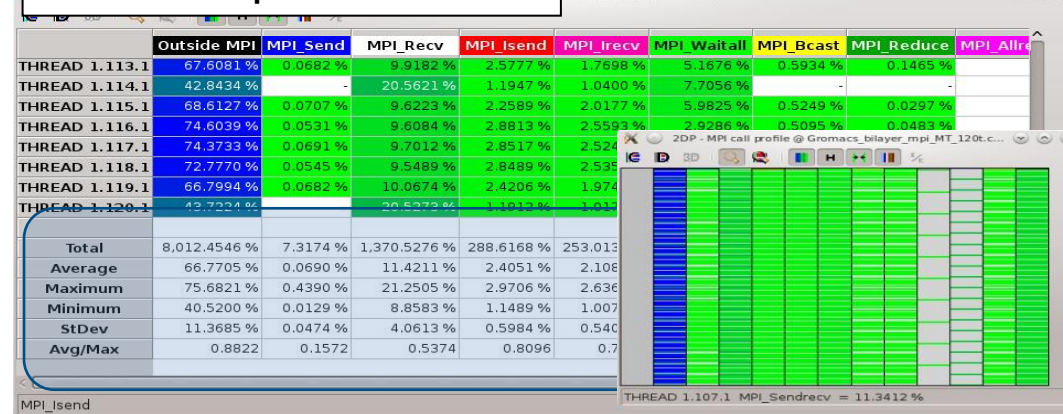
Tables: Profiles, histograms, correlations

- From timelines to tables

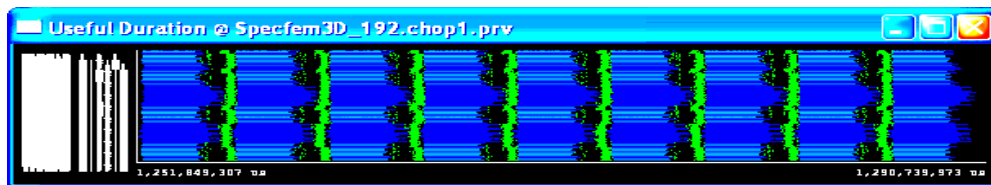
MPI calls



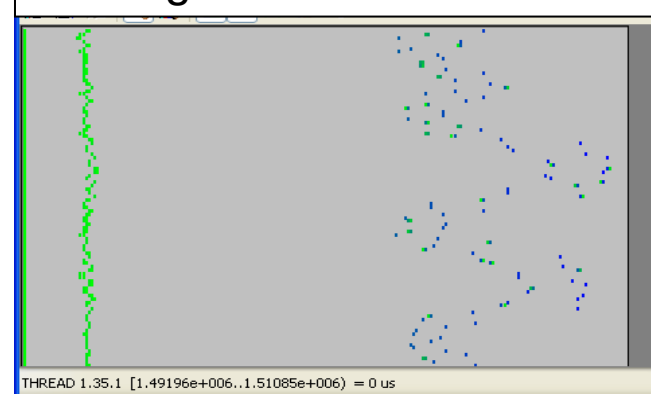
MPI calls profile



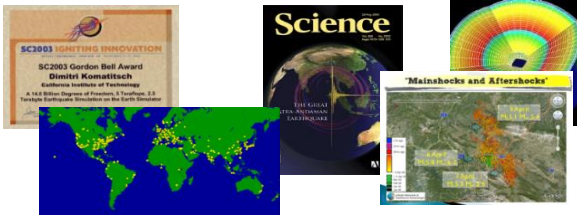
Useful Duration



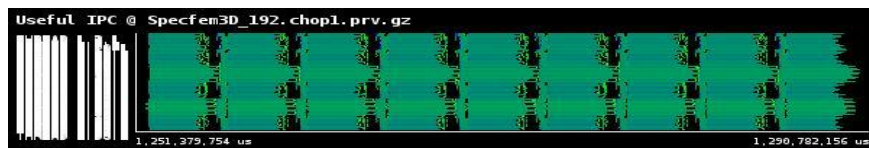
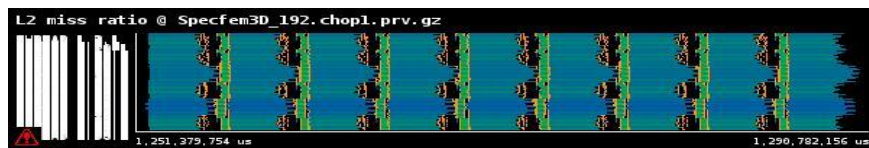
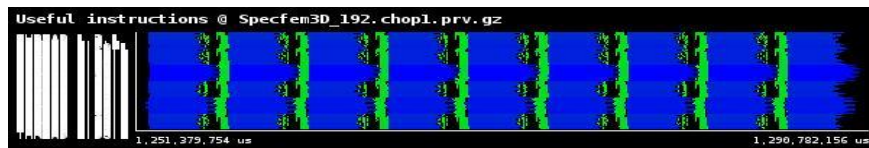
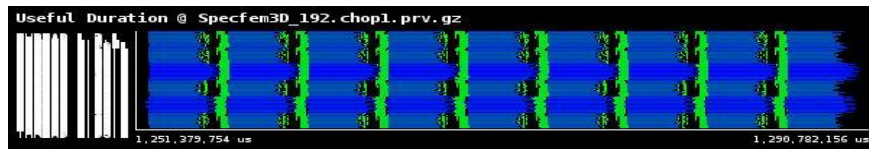
Histogram Useful Duration



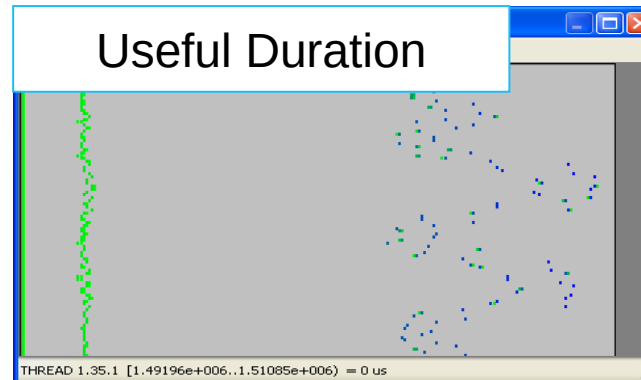
Analyzing variability through histograms and timelines



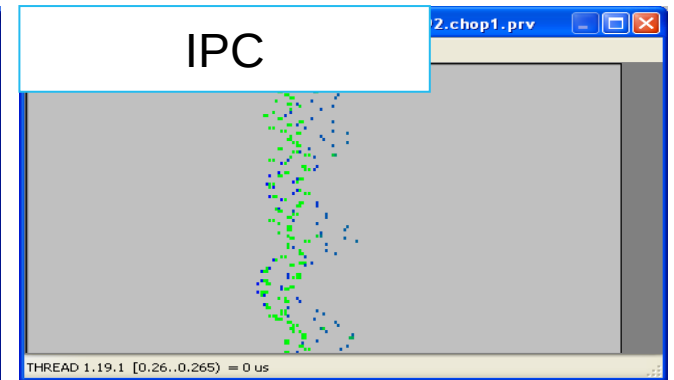
SPECFEM3D



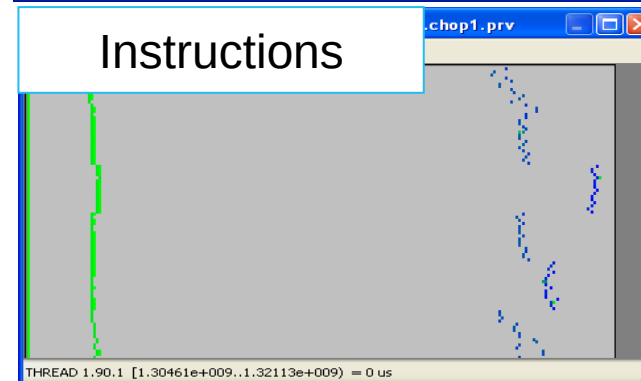
Useful Duration



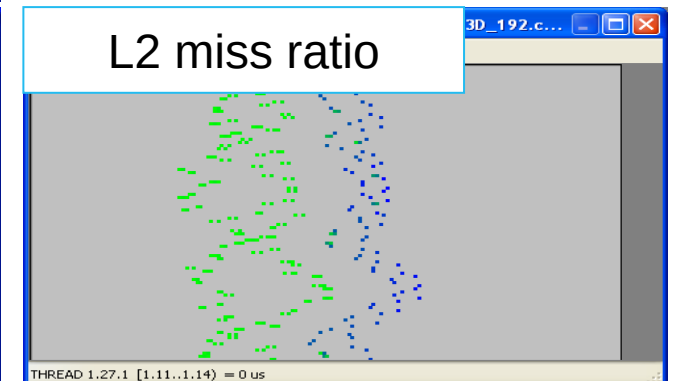
IPC



Instructions

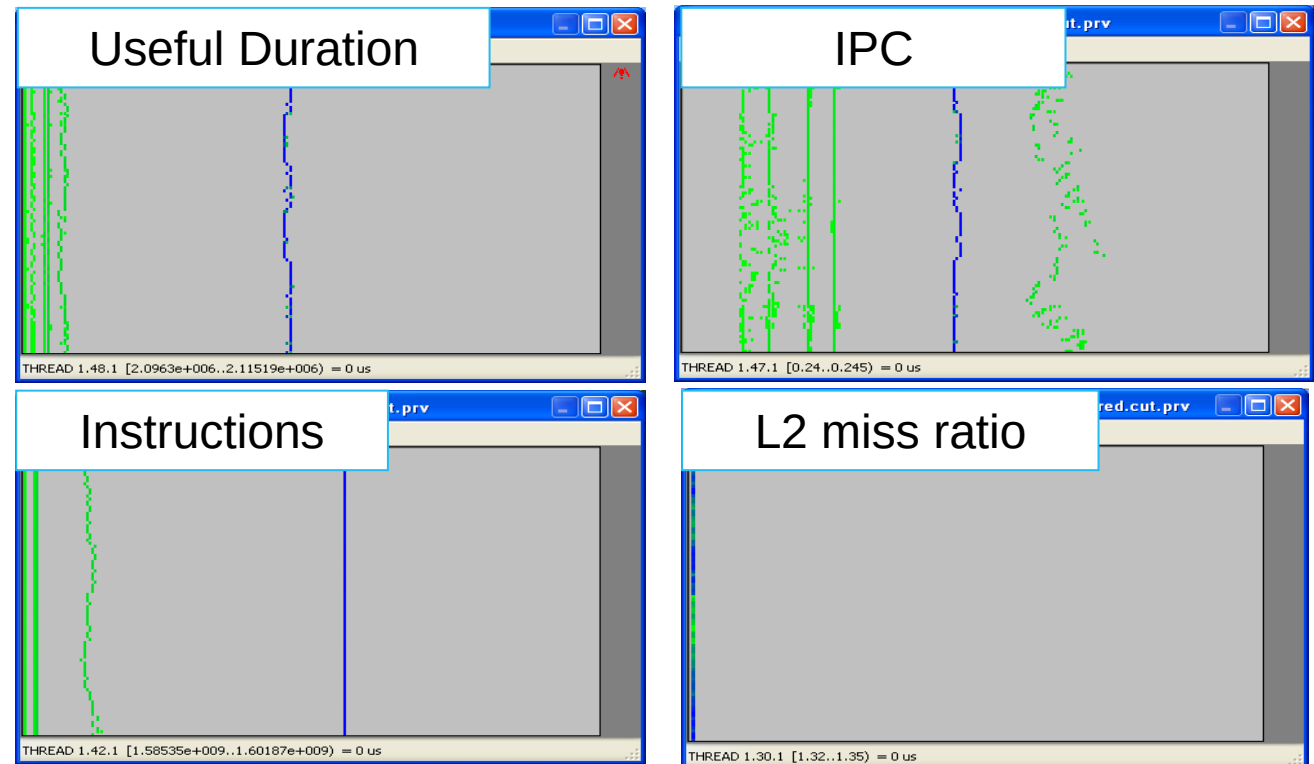


L2 miss ratio



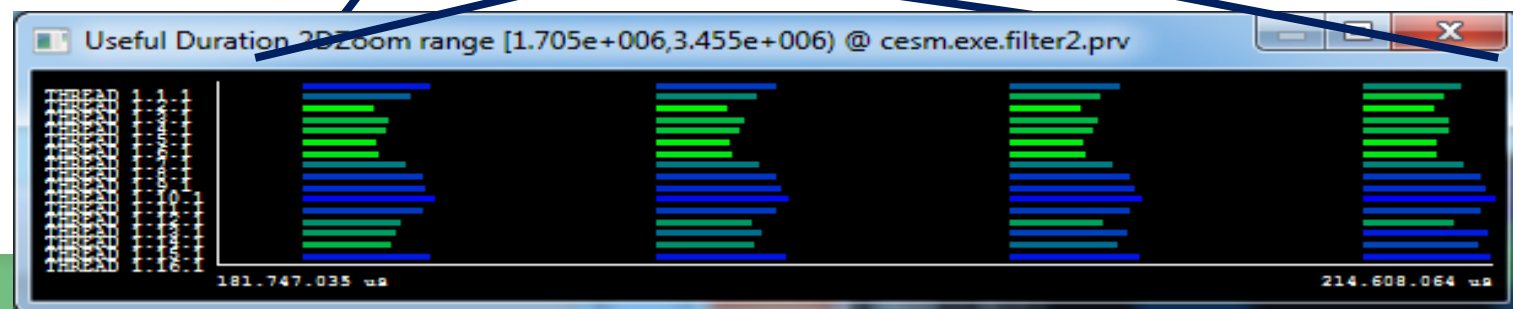
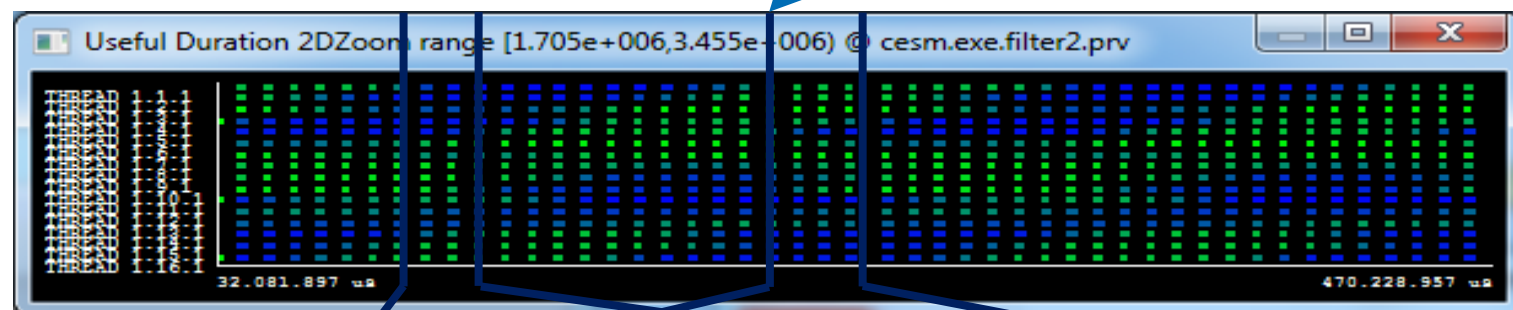
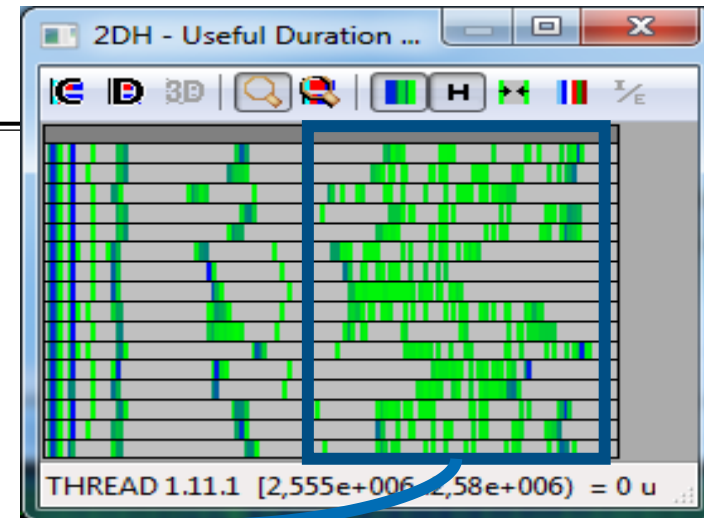
Analyzing variability through histograms and timelines

- By the way: six months later



Variability ... is everywhere

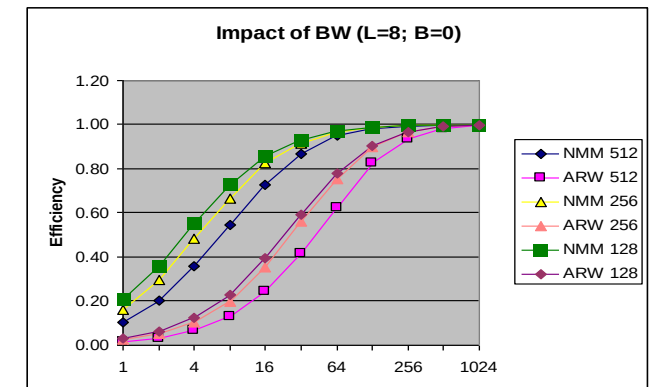
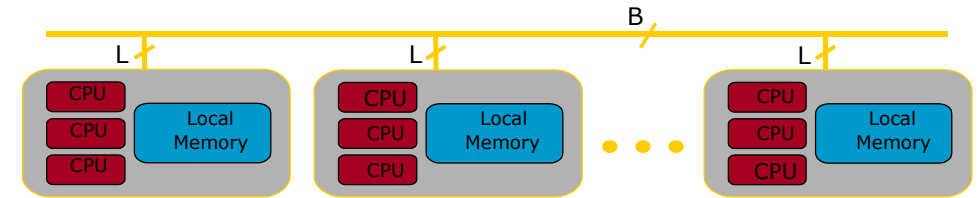
- CESM: 16 processes, 2 simulated days
- Histogram useful computation duration shows high variability
- How is it distributed?
- Dynamic imbalance
 - In space and time
 - Day and night.
 - Season ? ☺



Dimemas

Dimemas: Coarse grain, Trace driven simulation

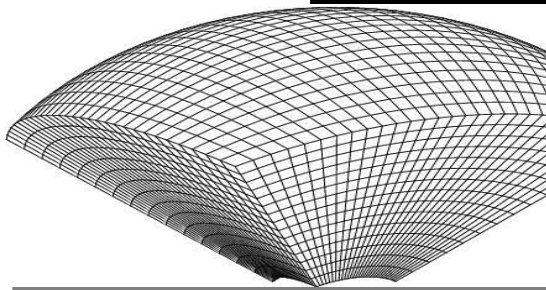
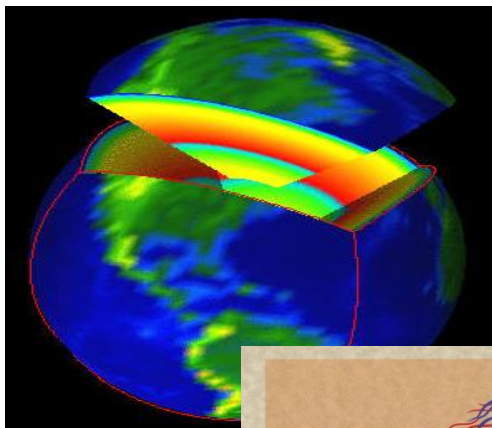
- Simulation: Highly non linear model
 - MPI protocols, resources contention...
- Parametric sweeps
 - On abstract architectures
 - On application computational regions
- What if analysis
 - Ideal machine (instantaneous network)
 - Estimating impact of ports to MPI+OpenMP/CUDA/...
 - Should I use asynchronous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

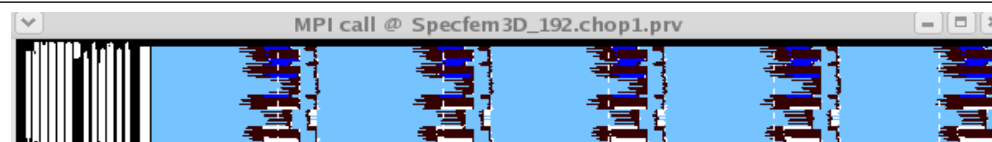
Would I will benefit from asynchronous communications?

- SPECFEM3D



Courtesy Dimitri Komatitsch

Real



Ideal



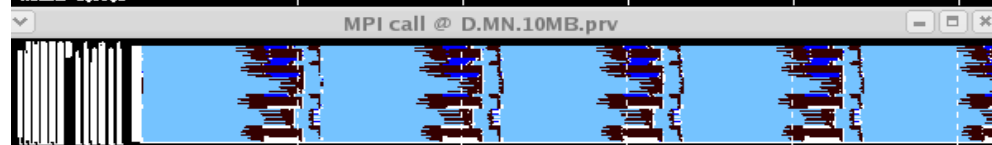
Prediction
MN



Prediction
100MB/s



Prediction
10MB/s



Prediction
5MB/s



Prediction
1MB/s



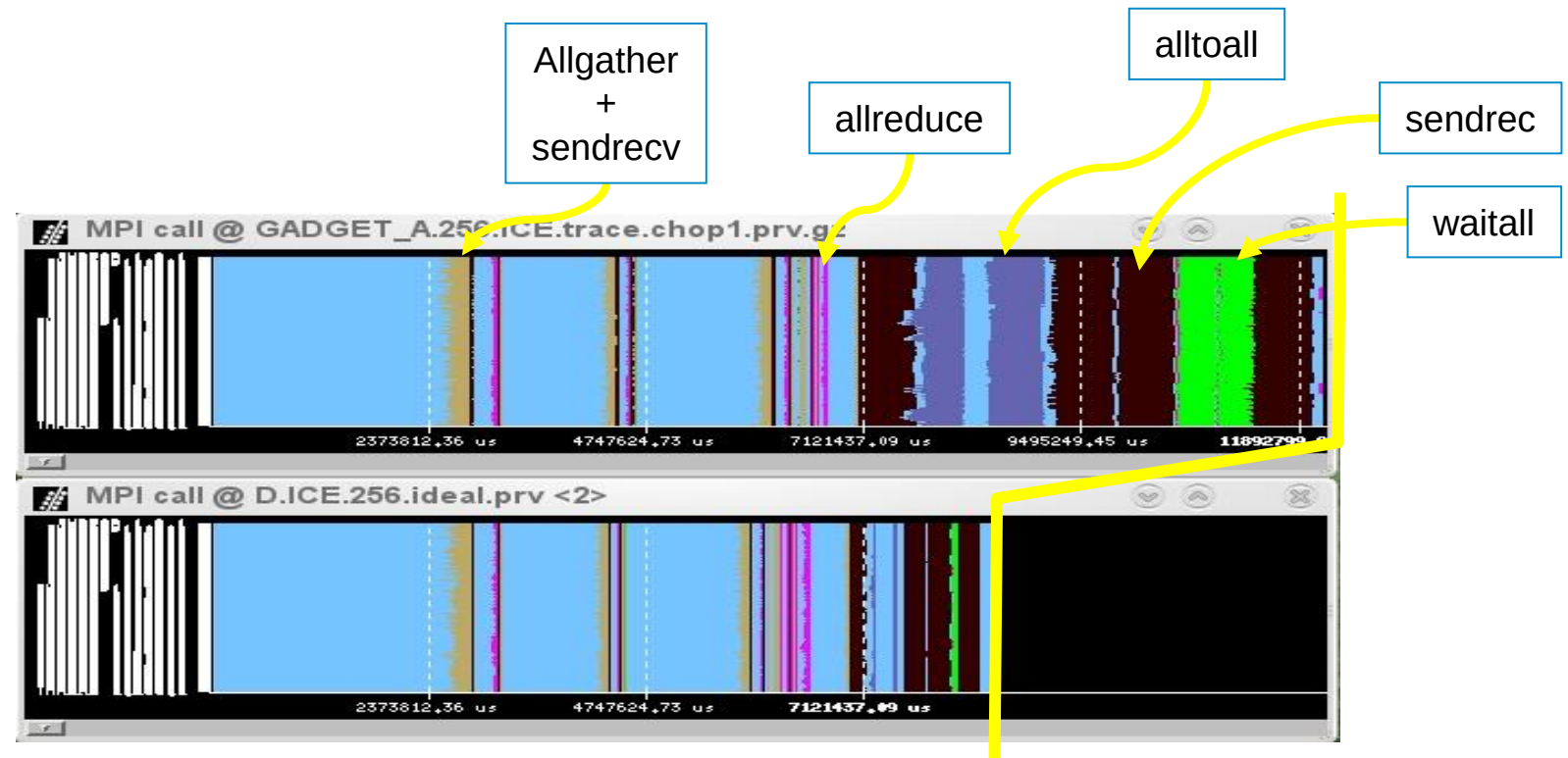
Ideal machine

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

GADGET @ Nehalem cluster
256 processes

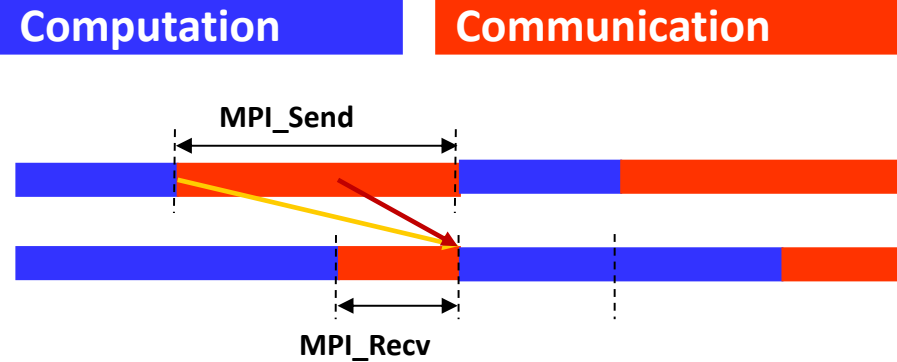
Real
run

Ideal
network

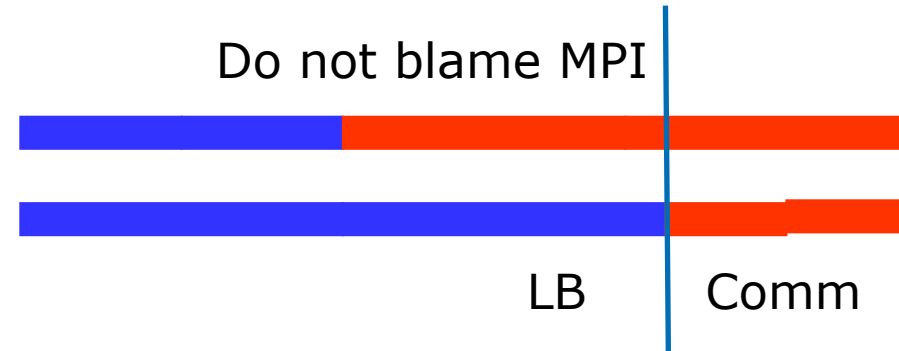


Models

Parallel efficiency model



- Parallel efficiency = LB eff * Comm eff



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

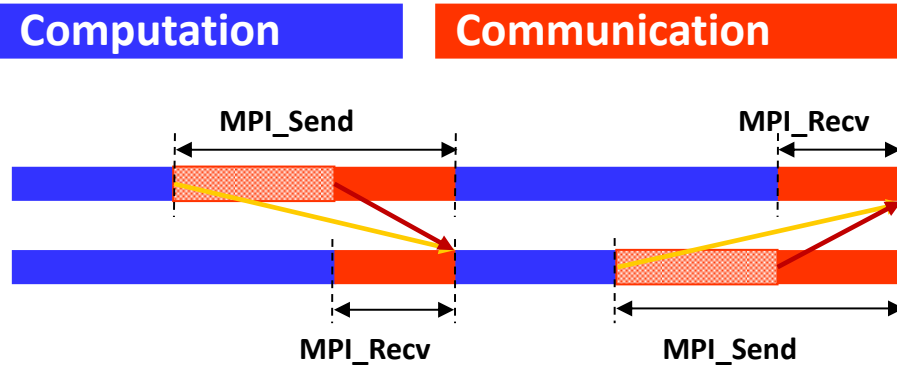
	Outside MPI	MPI_Recv	MPI_Isend	MPI_Irecv
THREAD 1.130.1	87,95 %	9,09 %	0,00 %	0,02 %
THREAD 1.131.1	88,16 %	9,09 %	0,00 %	0,02 %
THREAD 1.132.1	88,18 %	9,09 %	0,00 %	0,02 %
THREAD 1.133.1	88,18 %	9,09 %	0,00 %	0,02 %
Total	9,309,74 %	306,53 %	1,411,18 %	3,83 %
Average	69,00 %	2,30 %	10,69 %	0,03 %
Maximum	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

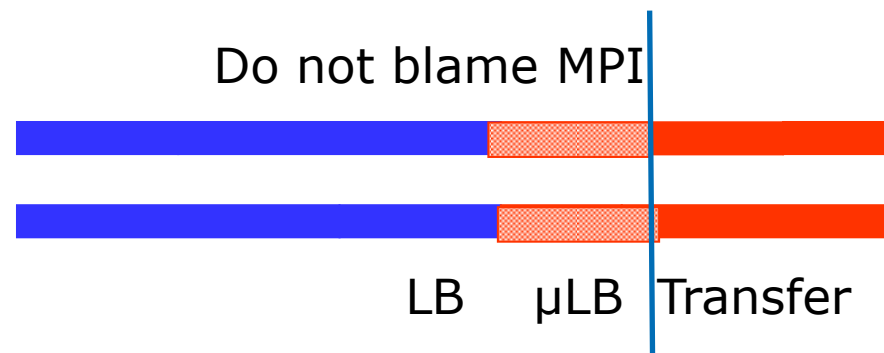
CommEff

LB

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

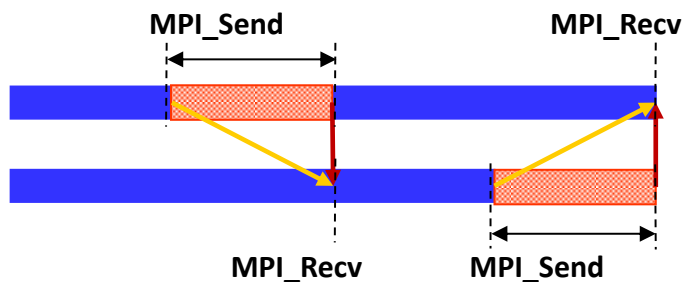


Do not blame MPI



$$LB=1$$

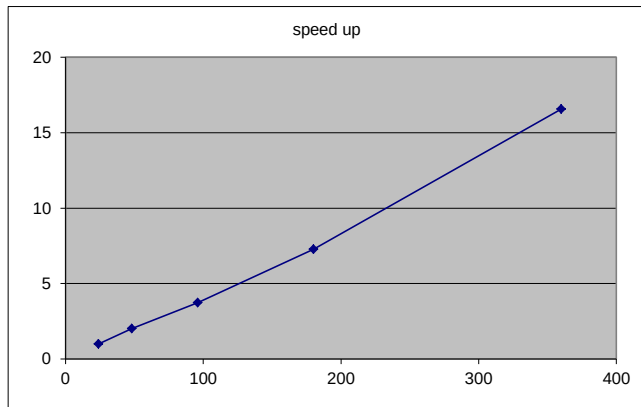
- Serializations / dependences (μLB)
- Dimemas ideal network $\rightarrow$ Transfer (efficiency) = 1



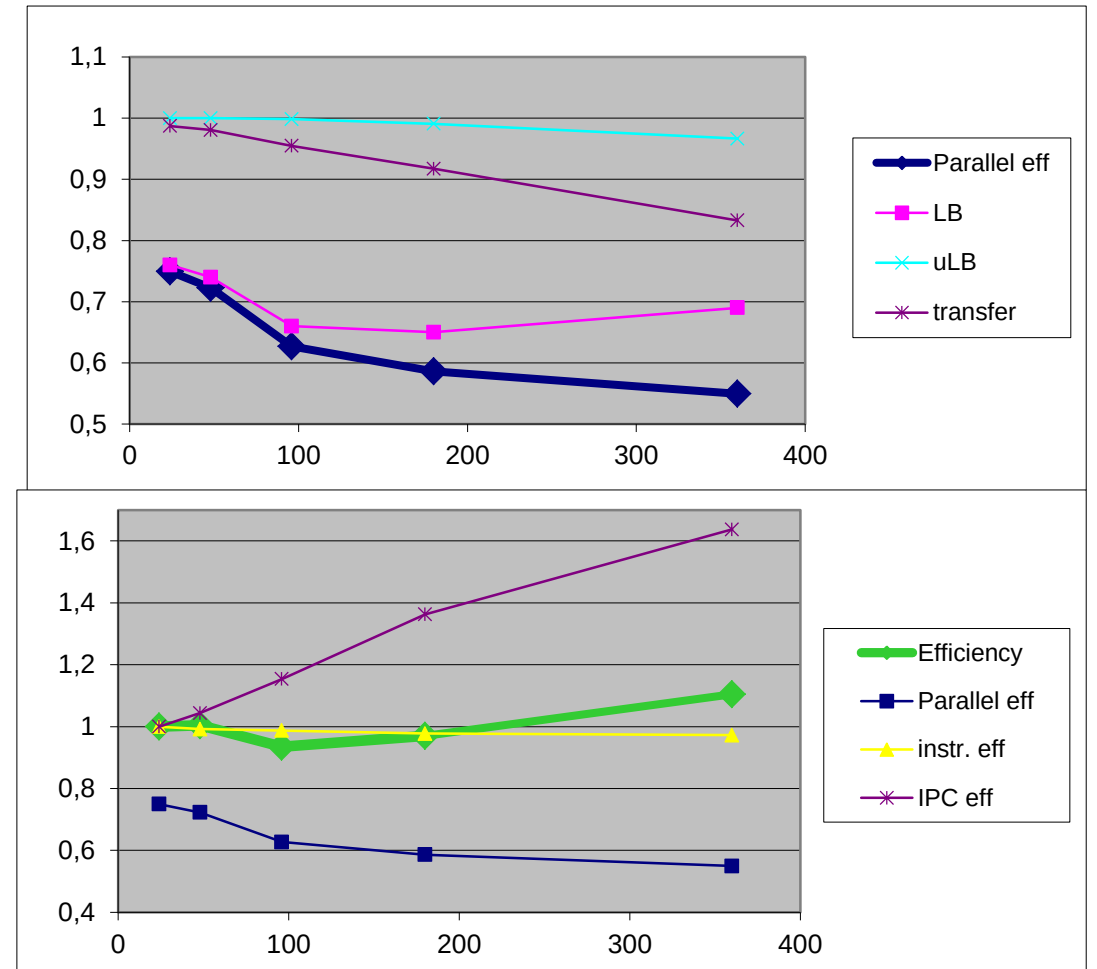
Why scaling?

CG-POP mpi2s1D - 180x120

Good scalability !!
Should we be happy?

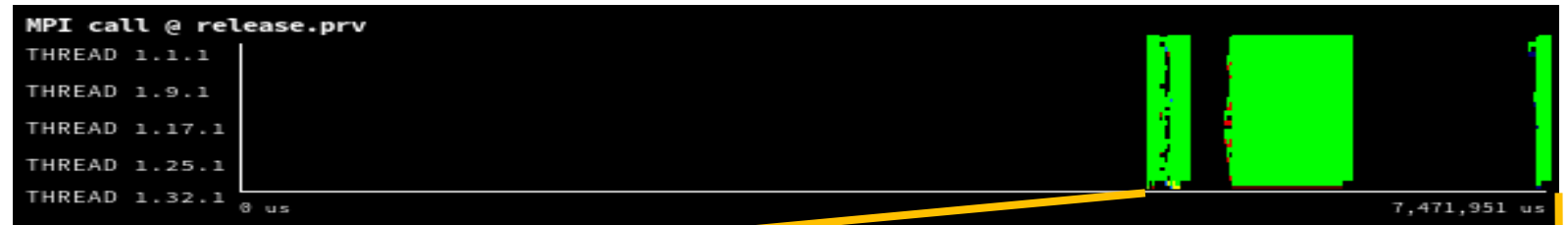


$$\eta_{\parallel} = LB * Ser * Trf$$



Why efficient?

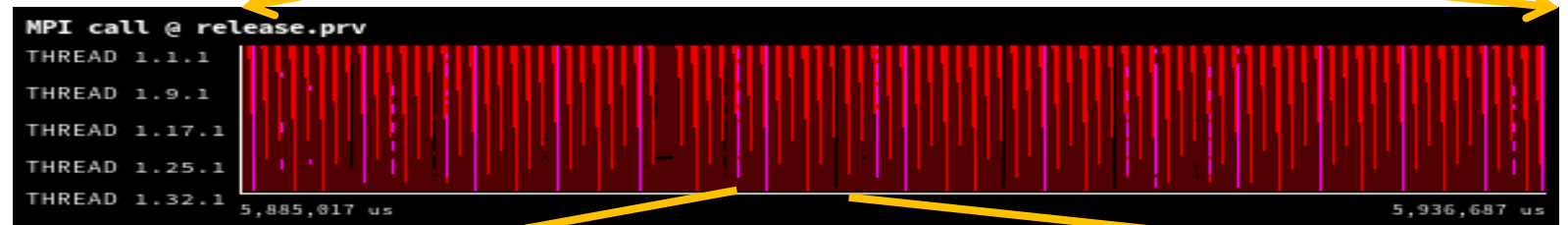
Parallel efficiency = 93.28
Communication = 93.84



Parallel efficiency = 77.93
Communication = 79.79

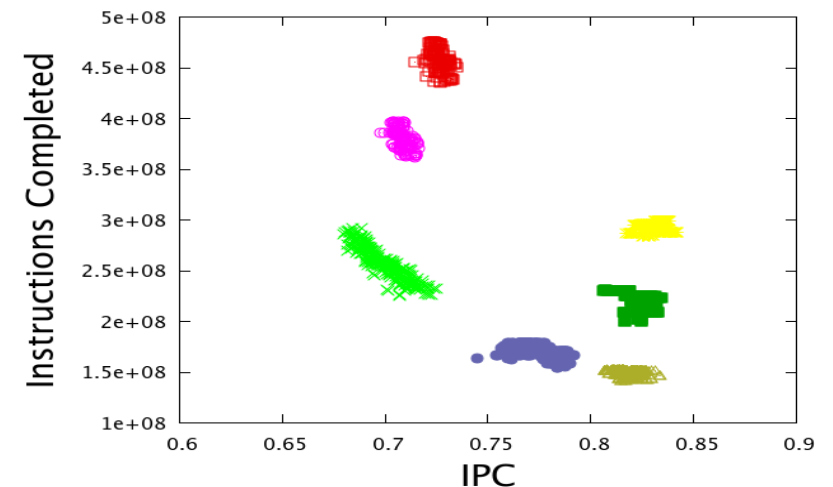
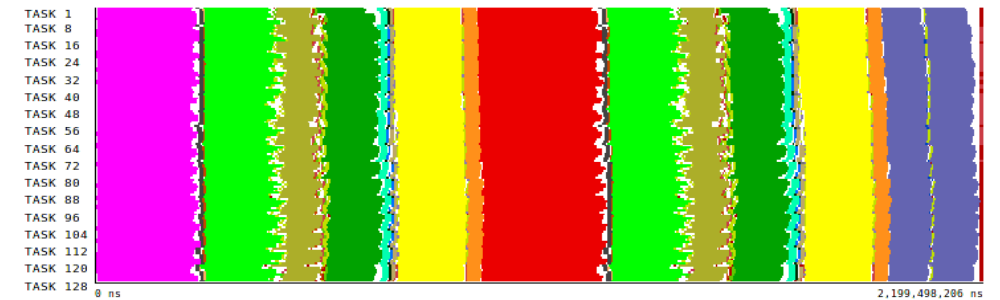
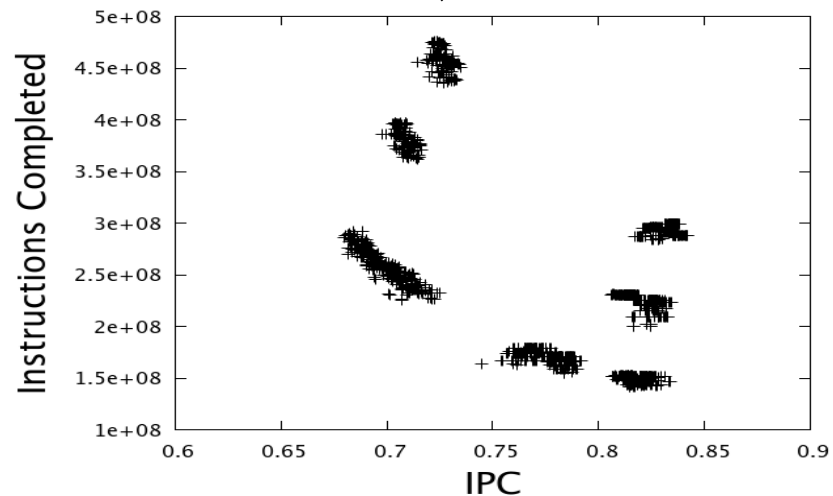
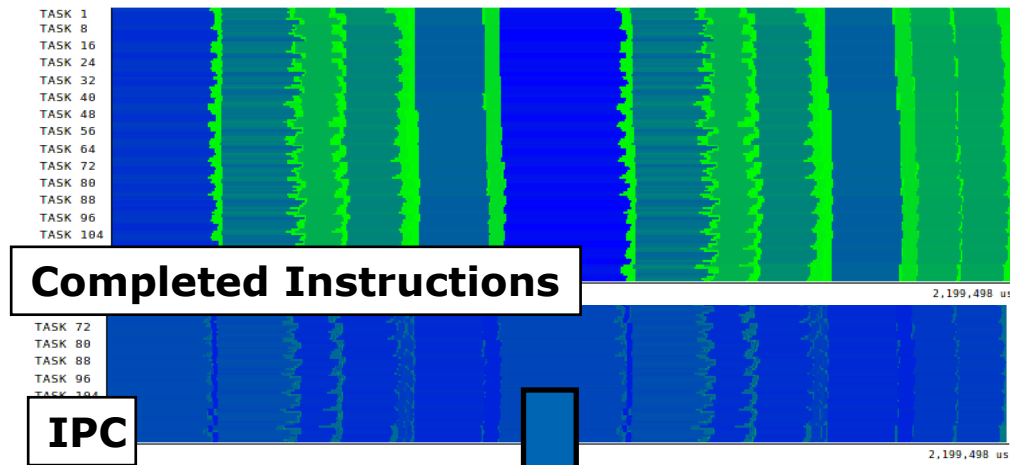


Parallel efficiency = 28.84
Communication eff = 30.42



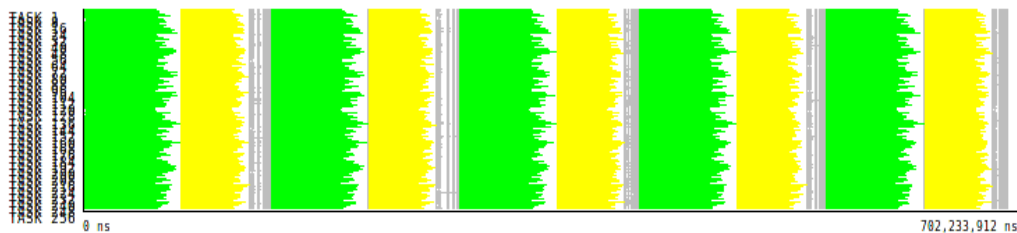
Clustering

Using Clustering to identify structure



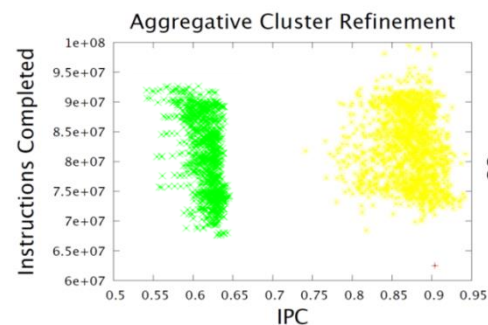
Integrating models and analytics

What if



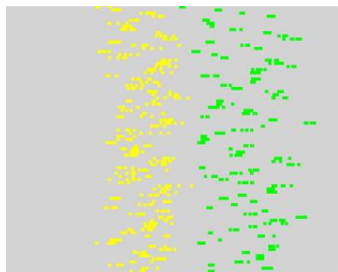
PEPC

... we increase the IPC of Cluster1?



13% gain

... we balance Clusters 1 & 2?



19% gain

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

Paraver Demo

Same code, different behaviour

Code	Parallel efficiency	Communication eff.	Load Balance eff.
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
lulesh@mt	83.68	95.48	87.64
lulesh@cori	90.92	98.59	92.20
lulesh@thunderX	73.96	97.56	75.81
lulesh@jetson	75.48	88.84	84.06
lulesh@claix	77.28	92.33	83.70
lulesh@jureca	88.20	98.45	89.57
lulesh@mn4	86.59	98.77	87.67
lulesh@inti	88.16	98.65	89.36
lulesh@archer	88.01	97.95	89.86
lulesh@romeo	89.56	99.01	90.45
lulesh@mn4	91.02	98.38	92.52
lulesh@ stampede2 (skl)	85.76	97.63	87.84
lulesh@ stampede2 (knl)	89.21	98.42	90.64
lulesh@isambard	90.32	97.16	92.96

Warning::: Higher parallel efficiency does not mean faster!