Understanding applications using the BSC performance tools

Germán Llort, Lau Mercadal

✉ tools@bsc.es

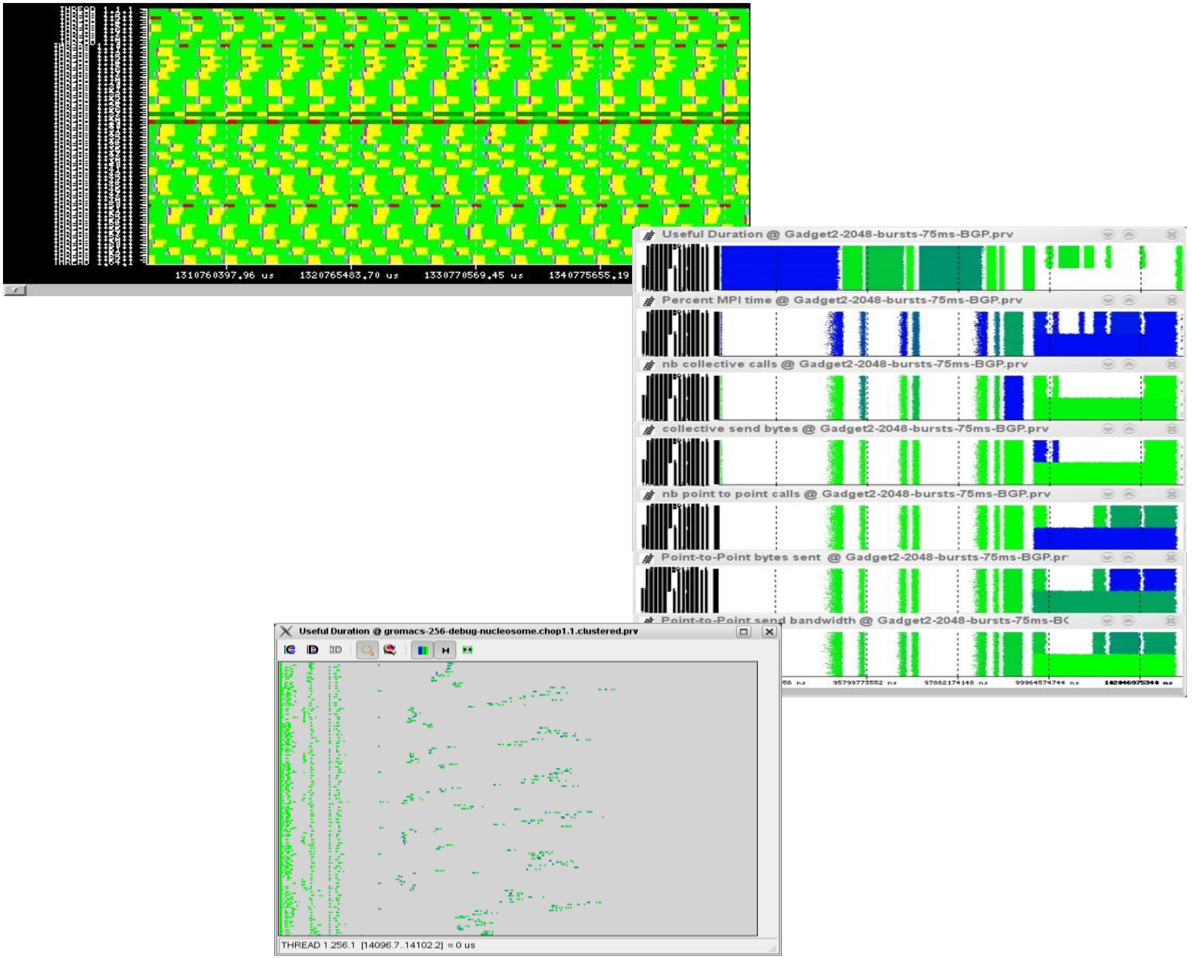
Barcelona Supercomputing Center

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

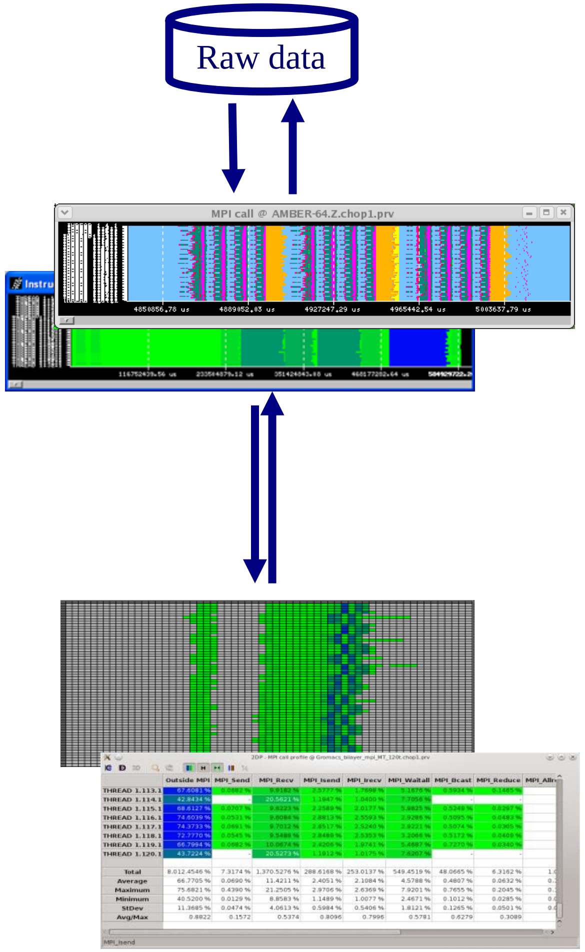
- Since 1991
- Based on traces
- Open Source
    - https://tools.bsc.es
- Core tools:
    - Paraver (paramedir) – offline trace analysis
    - Dimemas – message passing simulator
    - Extrae – instrumentation
- Focus
    - Detail, variability, flexibility
    - Behavioral structure vs. syntactic structure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


1310760397,96 us
1320765483,70 us
1330770569,45 us
1340775655,19
Useful Duration @ Gadget2-2048-bursts-75ms-BGP.prv
Percent MPI time @ Gadget2-2048-bursts-75ms-BGP.prv
nb collective calls @ Gadget2-2048-bursts-75ms-BGP.prv
collective send bytes @ Gadget2-2048-bursts-75ms-BGP.prv
nb point to point calls @ Gadget2-2048-bursts-75ms-BGP.prv
Point-to-Point bytes sent @ Gadget2-2048-bursts-75ms-BGP.pr
Point-to-Point send bandwidth @ Gadget2-2048-bursts-75ms-B(
56 ns
95799773552 ns
97882174148 ns
99964574744 ns
Useful Duration @ gromacs-256-debug-nucleosome.chop1.1.clustered.prv
3D
H

THREAD 1.256.1 [14096.7..14102.2] = 0 us

Paraver

Paraver: Performance data browser



	Outside MPI	MPI_Send	MPI_Recv	MPI_Isend	MPI_Irecv	MPI_Waitall	MPI_Bcast	MPI_Reduce	MPI_Allr
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 %	9.8223 %	2.2589 %	2.0177 %	5.9825 %	0.5249 %	0.0297 %	
THREAD 1.116.1	74.6039 %	0.0531 %	9.0084 %	2.8813 %	2.5593 %	2.9286 %	0.5095 %	0.0483 %	
THREAD 1.117.1	74.3733 %	0.0691 %	9.7012 %	2.8517 %	2.5240 %	2.9221 %	0.5074 %	0.0305 %	
THREAD 1.118.1	72.7770 %	0.0545 %	9.5489 %	2.8489 %	2.5353 %	3.2066 %	0.5172 %	0.0409 %	
THREAD 1.119.1	66.7994 %	0.0682 %	10.0674 %	2.4206 %	1.9741 %	5.4687 %	0.7270 %	0.0340 %	
THREAD 1.120.1	43.7224 %	-	20.5273 %	1.1912 %	1.0175 %	7.6207 %	-	-	
Total	8,012.4546 %	7.3174 %	1,370.5276 %	288.6168 %	253.0137 %	549.4519 %	48.0665 %	6.3162 %	1.(
Average	66.7705 %	0.0690 %	11.4211 %	2.4051 %	2.1084 %	4.5788 %	0.4807 %	0.0632 %	0.:
Maximum	75.6821 %	0.4390 %	21.2505 %	2.9706 %	2.6369 %	7.9201 %	0.7655 %	0.2045 %	0.:
Minimum	40.5200 %	0.0129 %	8.8583 %	1.1489 %	1.0077 %	2.4671 %	0.1012 %	0.0285 %	0.(
StDev	11.3685 %	0.0474 %	4.0613 %	0.5984 %	0.5406 %	1.8121 %	0.1265 %	0.0501 %	0.(
Avg/Max	0.8822	0.1572	0.5374	0.8096	0.7996	0.5781	0.6279	0.3089	

Timelines

2/3D tables (Statistics)

Trace visualization/analysis

+ trace manipulation

Goal = Flexibility

No semantics

Programmable

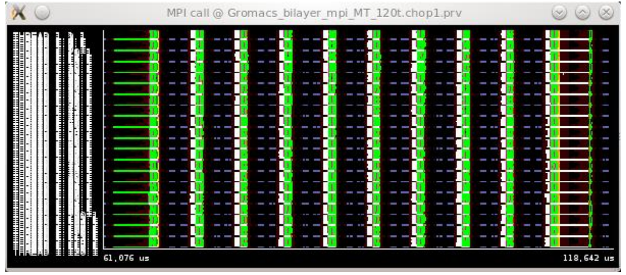
Comparative analyses

Multiple traces

Synchronize scales

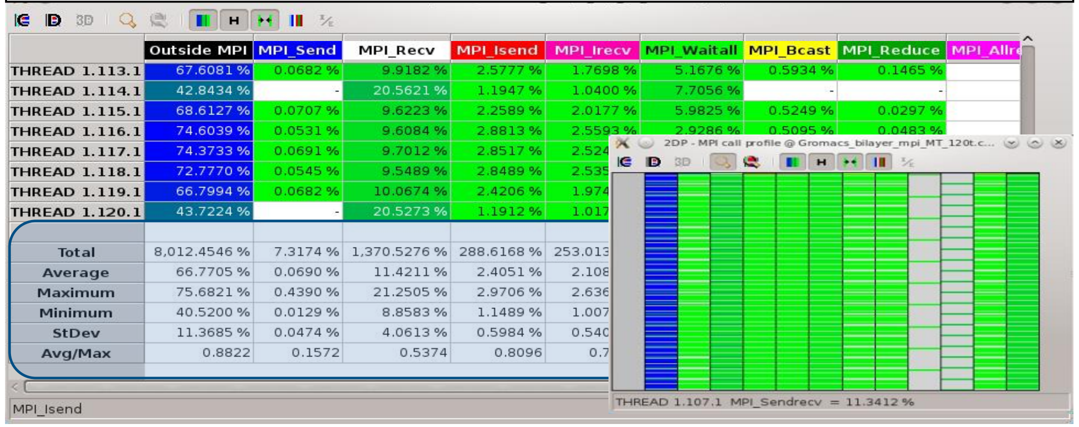
From timelines to tables

MPI calls



MPI call @ Gromacs_bilayer_mpi_MT_120t.chop1.prv

MPI calls profile



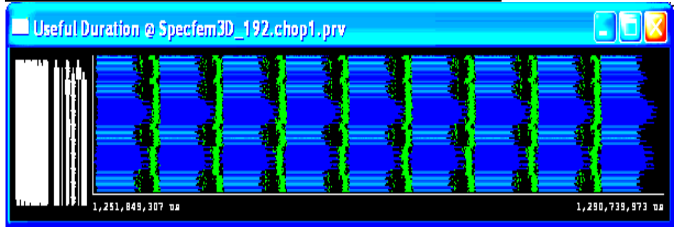
	Outside MPI	MPI_Send	MPI_Recv	MPI_Isend	MPI_Irecv	MPI_Waitall	MPI_Bcast	MPI_Reduce	MPI_Allre
THREAD 1.113.1	67.6081 %	0.0682 %	9.9182 %	2.5777 %	1.7698 %	5.1676 %	0.5934 %	0.1465 %	
THREAD 1.114.1	42.8434 %	-	20.5621 %	1.1947 %	1.0400 %	7.7056 %	-	-	
THREAD 1.115.1	68.6127 %	0.0707 %	9.6223 %	2.2589 %	2.0177 %	5.9825 %	0.5249 %	0.0297 %	
THREAD 1.116.1	74.6039 %	0.0531 %	9.6084 %	2.8813 %	2.5593 %	2.9286 %	0.5095 %	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 %	-	20.5273 %	1.1912 %	1.017				
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

2DP - MPI call profile @ Gromacs_bilayer_mpi_MT_120t.c...

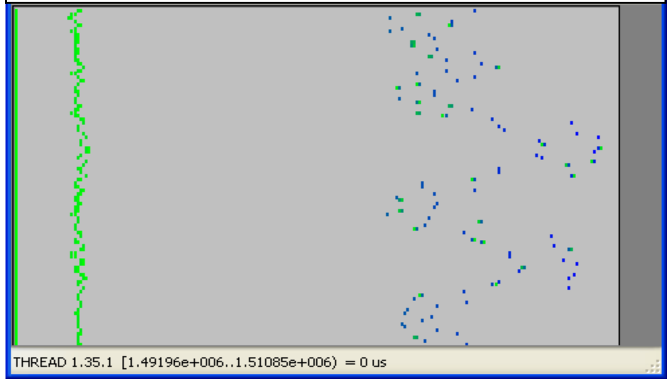
THREAD 1.107.1 MPI_Sendrecv = 11.3412 %

Useful Duration



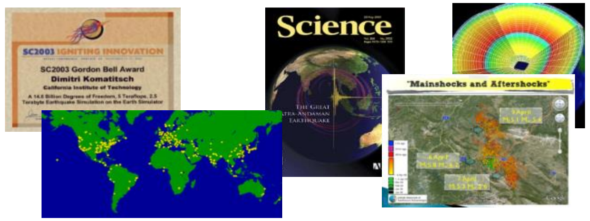
Useful Duration @ Specfem3D_192.chop1.prv

Histogram Useful Duration

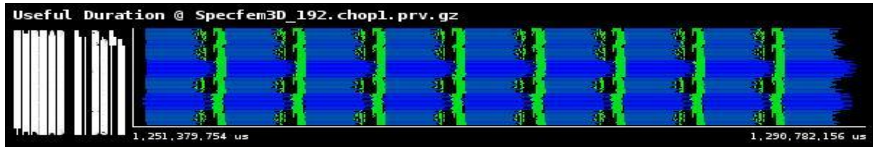


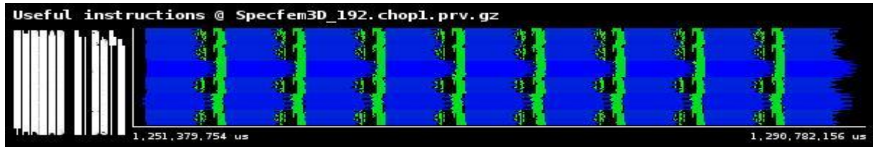
THREAD 1.35.1 [1.49196e+006..1.51085e+006) = 0 us

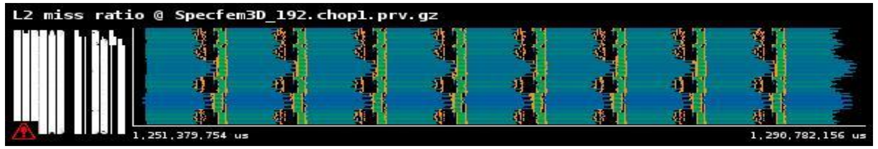
Analyzing variability through histograms and timelines

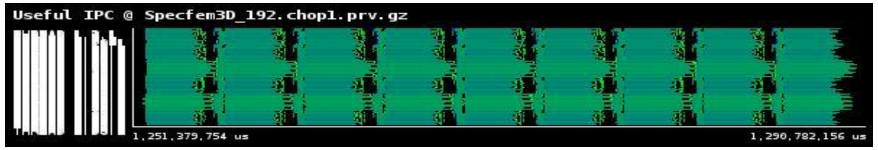

SC2003 IGNITING INNOVATION
SC2003 Gordon Bell Award
Dimitri Komatitsch
California Institute of Technology
Science
THE GREAT SUMATRA-ANDAMAN EARTHQUAKE
"Mainshocks and Aftershocks"

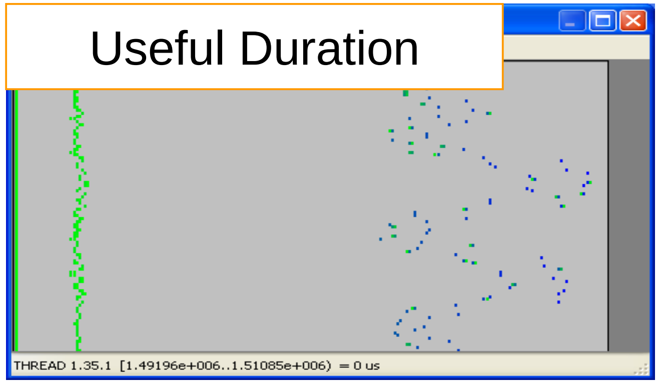
SPECFEM3D



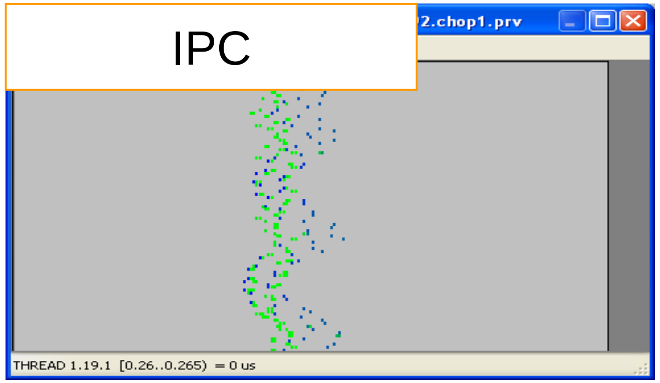




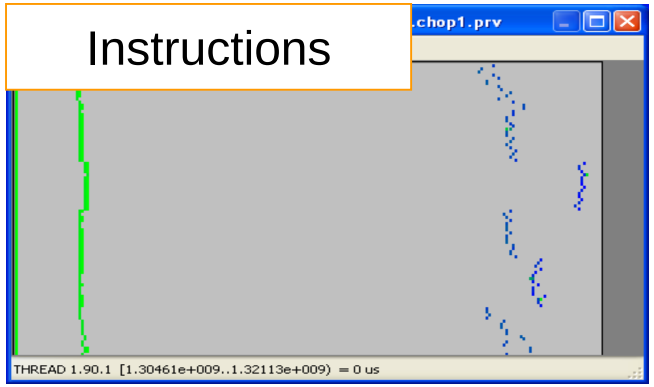



Useful Duration

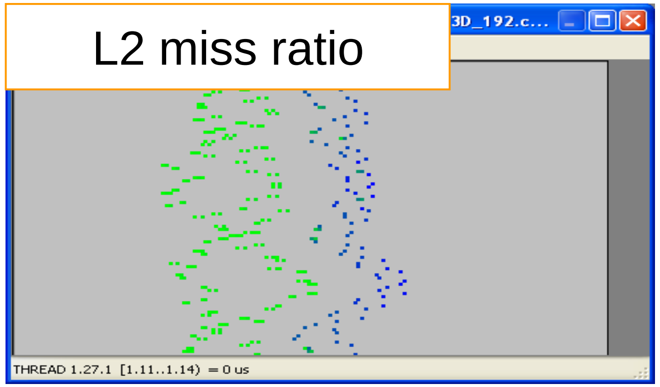
THREAD 1.35.1 [1.49196e+006..1.51085e+006) = 0 us


IPC

THREAD 1.19.1 [0.26..0.265) = 0 us


Instructions

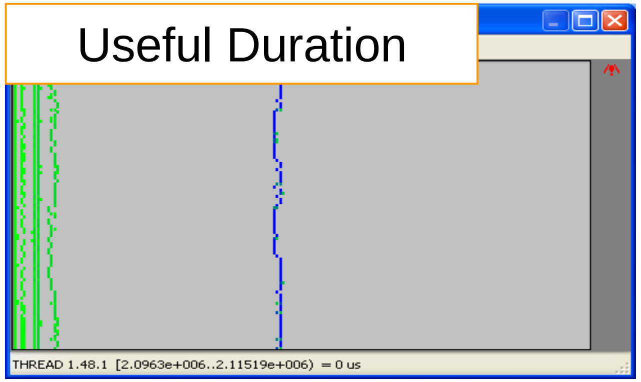
THREAD 1.90.1 [1.30461e+009..1.32113e+009) = 0 us


L2 miss ratio

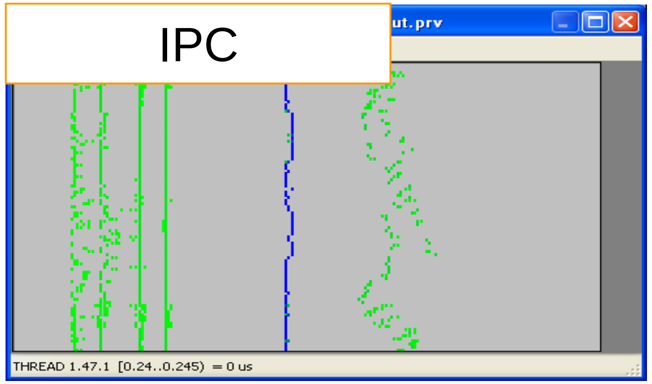
THREAD 1.27.1 [1.11..1.14) = 0 us

Analyzing variability through histograms and timelines

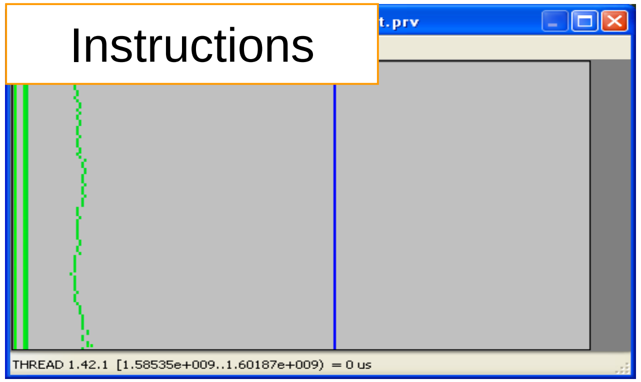
- By the way: six months later ….


Useful Duration

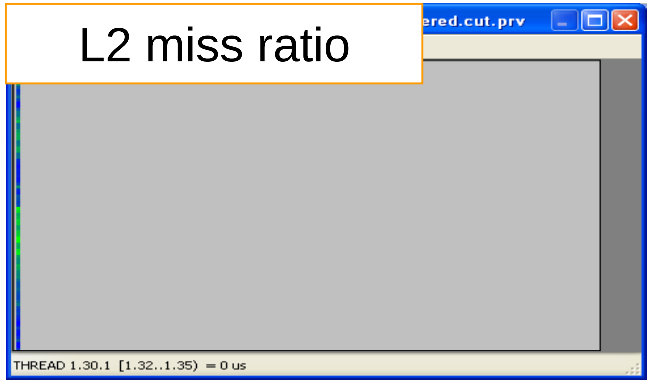
THREAD 1.48.1 [2.0963e+006..2.11519e+006) = 0 us


IPC

THREAD 1.47.1 [0.24..0.245) = 0 us


Instructions

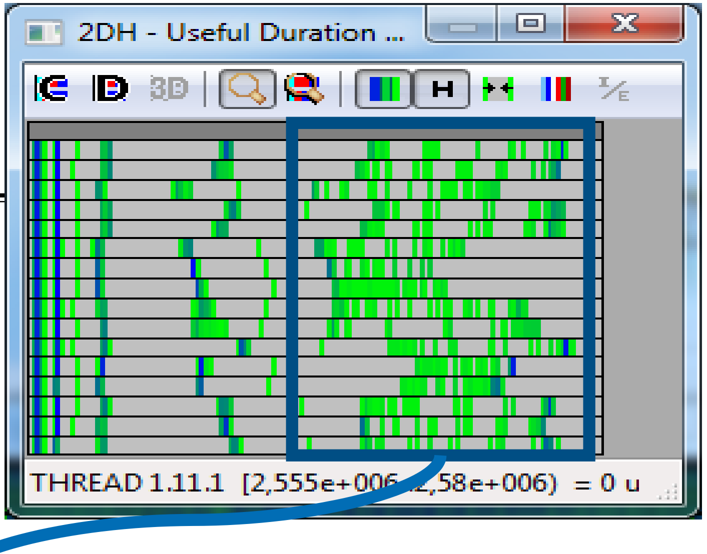
THREAD 1.42.1 [1.58535e+009..1.60187e+009) = 0 us


L2 miss ratio

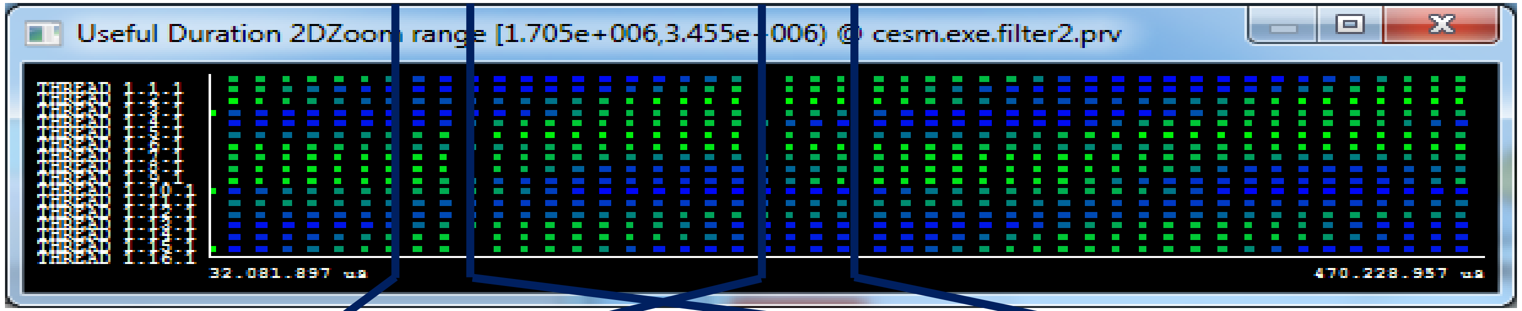
THREAD 1.30.1 [1.32..1.35) = 0 us

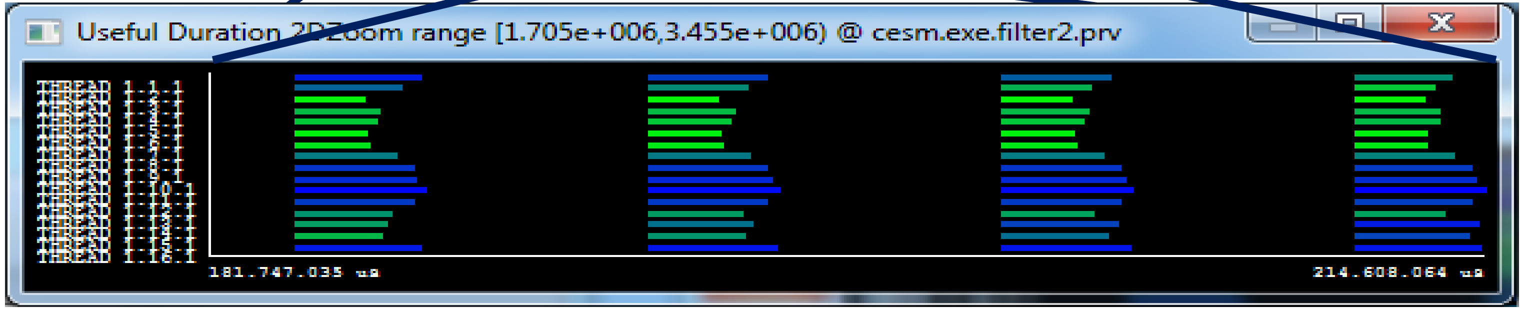
From tables to timelines

- 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 ? ☺


2DH - Useful Duration ...

THREAD 1.11.1 [2,555e+006 ..2,58e+006) = 0 u

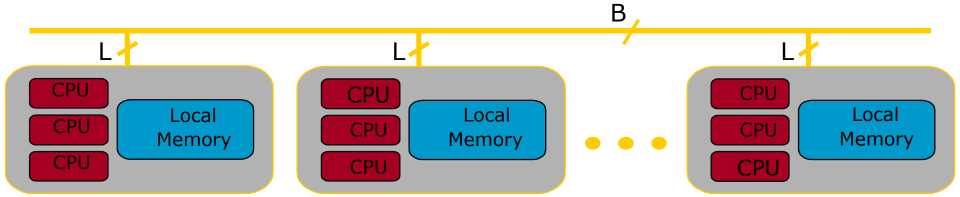

Useful Duration 2DZoom range [1.705e+006,3.455e+006) @ cesm.exe.filter2.prv

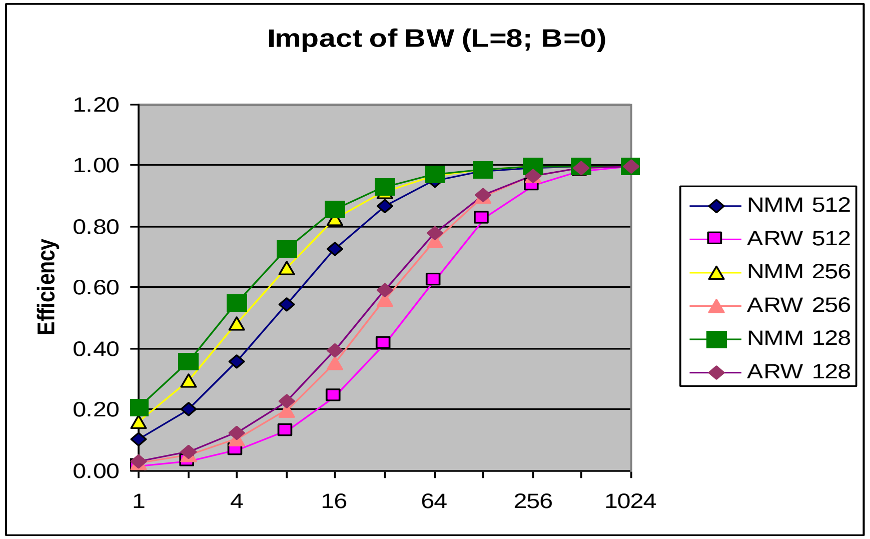

Useful Duration 2DZoom range [1.705e+006,3.455e+006) @ cesm.exe.filter2.prv

Dimemas

Dimemas: Coarse grain, Trace driven simulation

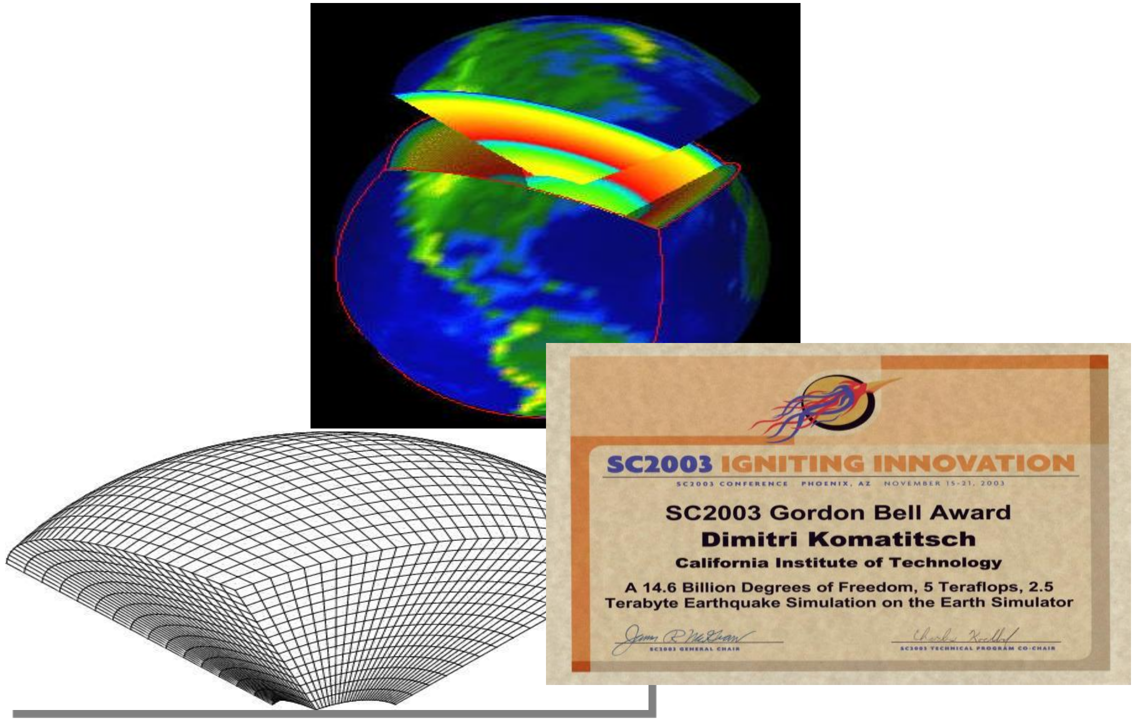
- Simulation: Highly non linear model
  - MPI protocols, resources contention…
- Parametric sweeps
  - On abstract architectures
  - On application computational regions
- What if analysis
  - Ideal machine (instantaneous network)
  - Estimating impact of ports to MPI+OpenMP/CUDA/…
  - Should I use asynchronous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





What if we had asynchronous communications?

- SPECFEM3D



SC2003 IGNITING INNOVATION

SC2003 CONFERENCE PHOENIX, AZ NOVEMBER 15-21, 2003

SC2003 Gordon Bell Award

Dimitri Komatitsch

California Institute of Technology

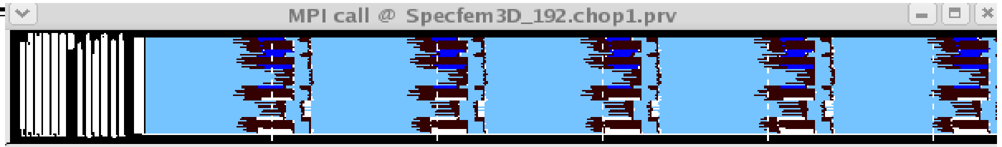
A 14.6 Billion Degrees of Freedom, 5 Teraflops, 2.5 Terabyte Earthquake Simulation on the Earth Simulator

SC2003 GENERAL CHAIR

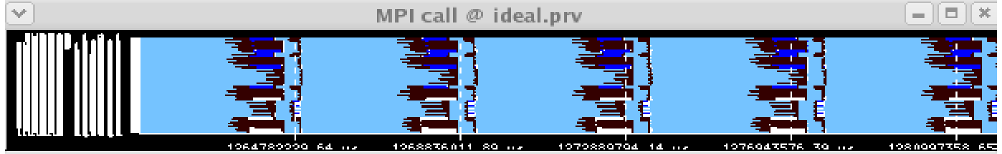
SC2003 TECHNICAL PROGRAM CO-CHAIR

Courtesy Dimitri Komatitsch

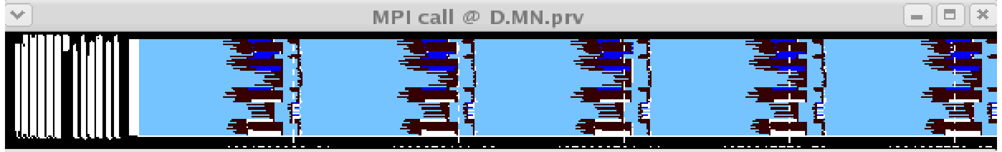
Real



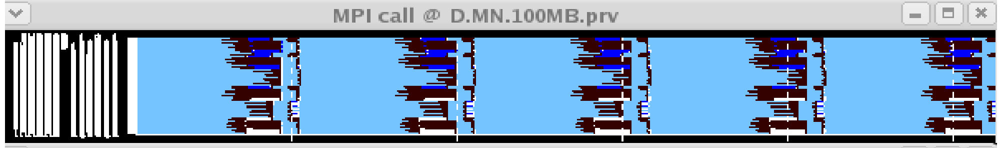
Ideal



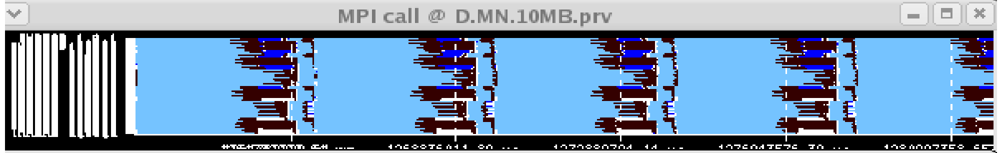
Prediction MN



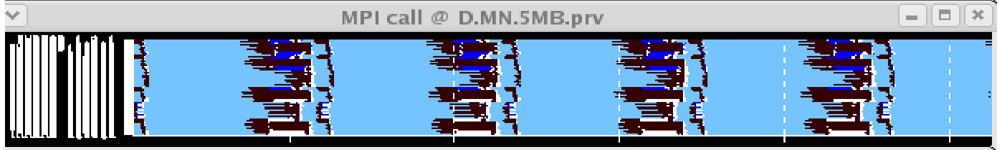
Prediction 100MB/s



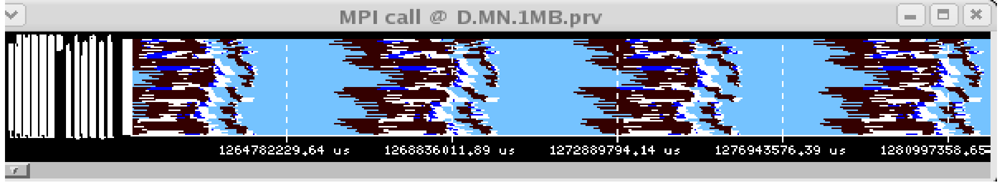
Prediction 10MB/s



Prediction 5MB/s



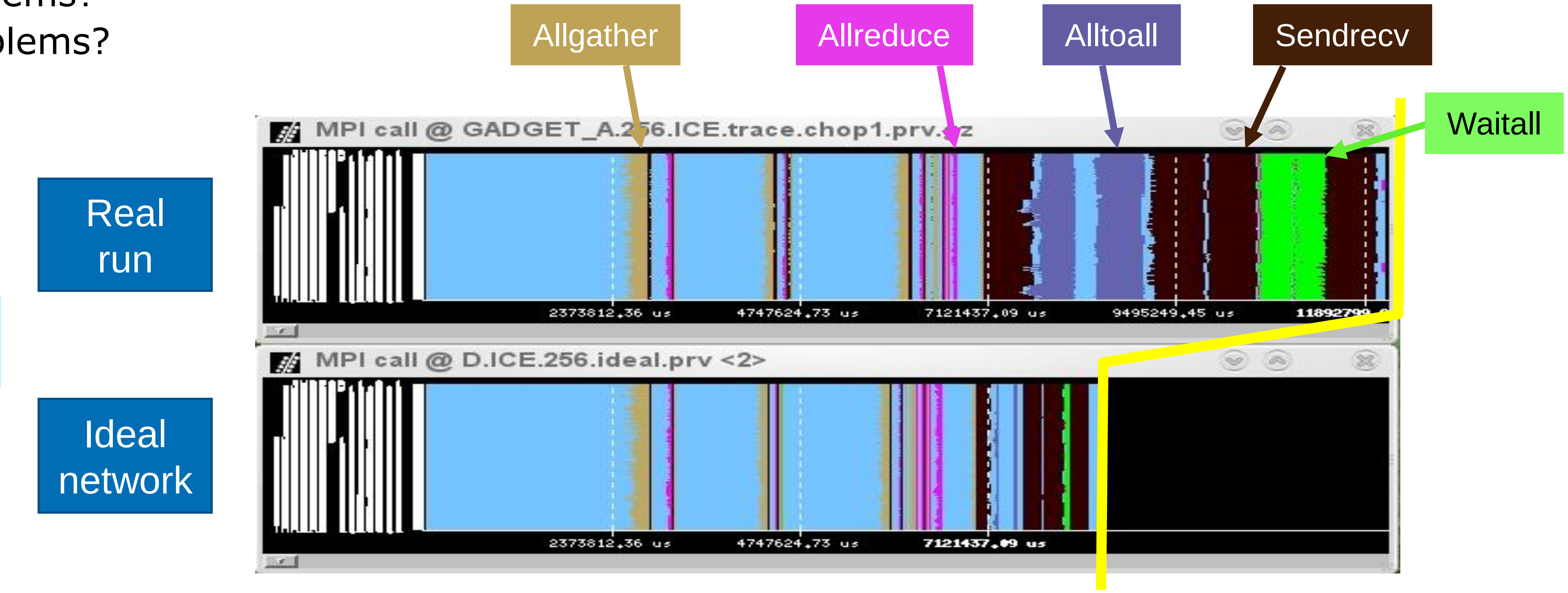
Prediction 1MB/s



Ideal machine

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

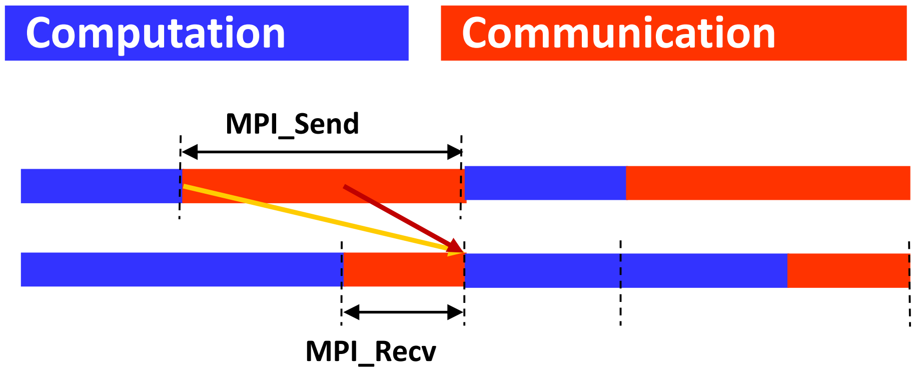
GADGET @ Nehalem
256 processes

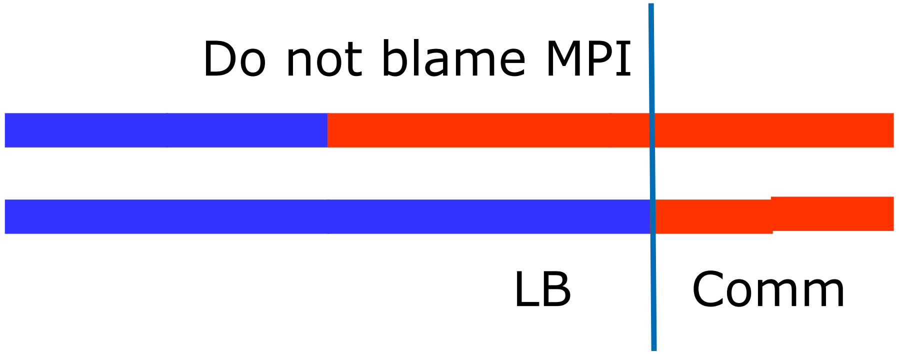


Impact on practical machines?

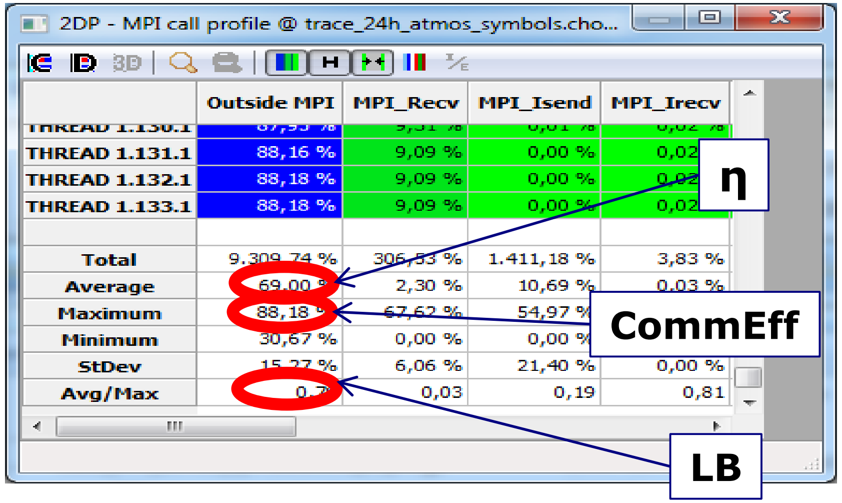
Models

Parallel efficiency model



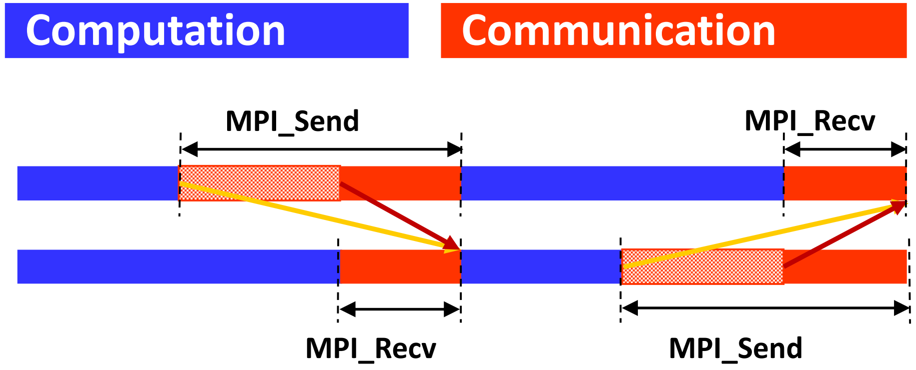


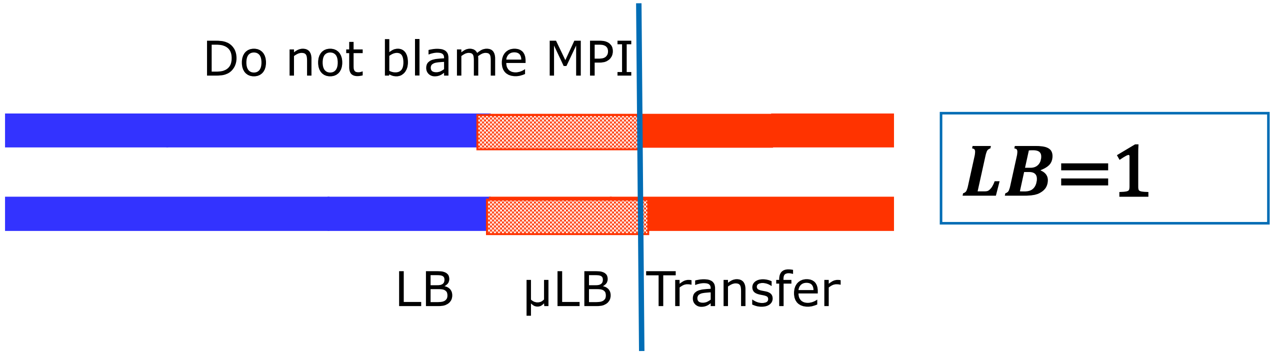
- Parallel efficiency = LB eff * Comm eff



	Outside MPI	MPI_Recv	MPI_Isend	MPI_Irecv
THREAD 1.130.1	87,93 %	9,31 %	0,01 %	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	0,7	0,03	0,19	0,81

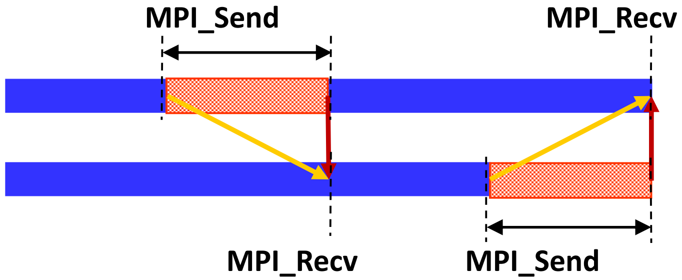
Parallel efficiency refinement: LB * µLB * Transfer





$$LB = 1$$

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

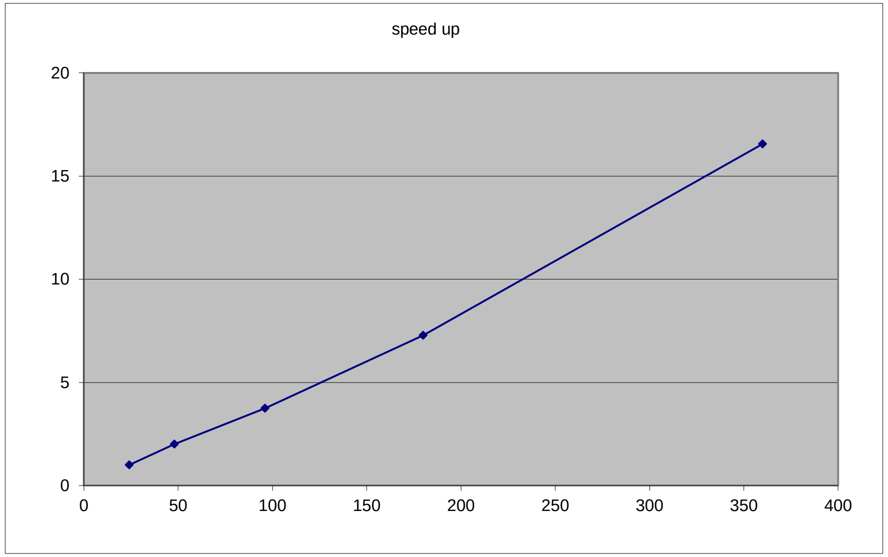


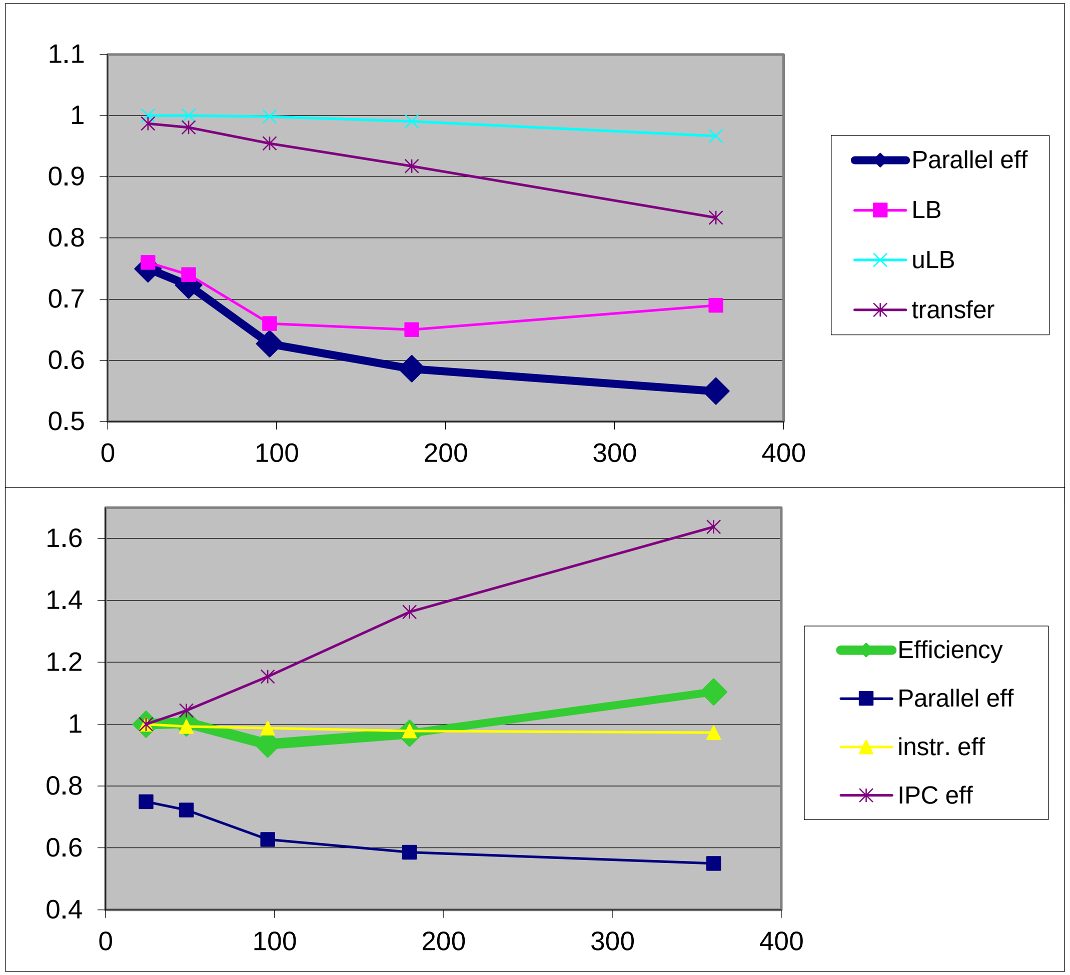
Why scaling?

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

CG-POP mpi2s1D - 180x120

Good scalability !!
Should we be happy?



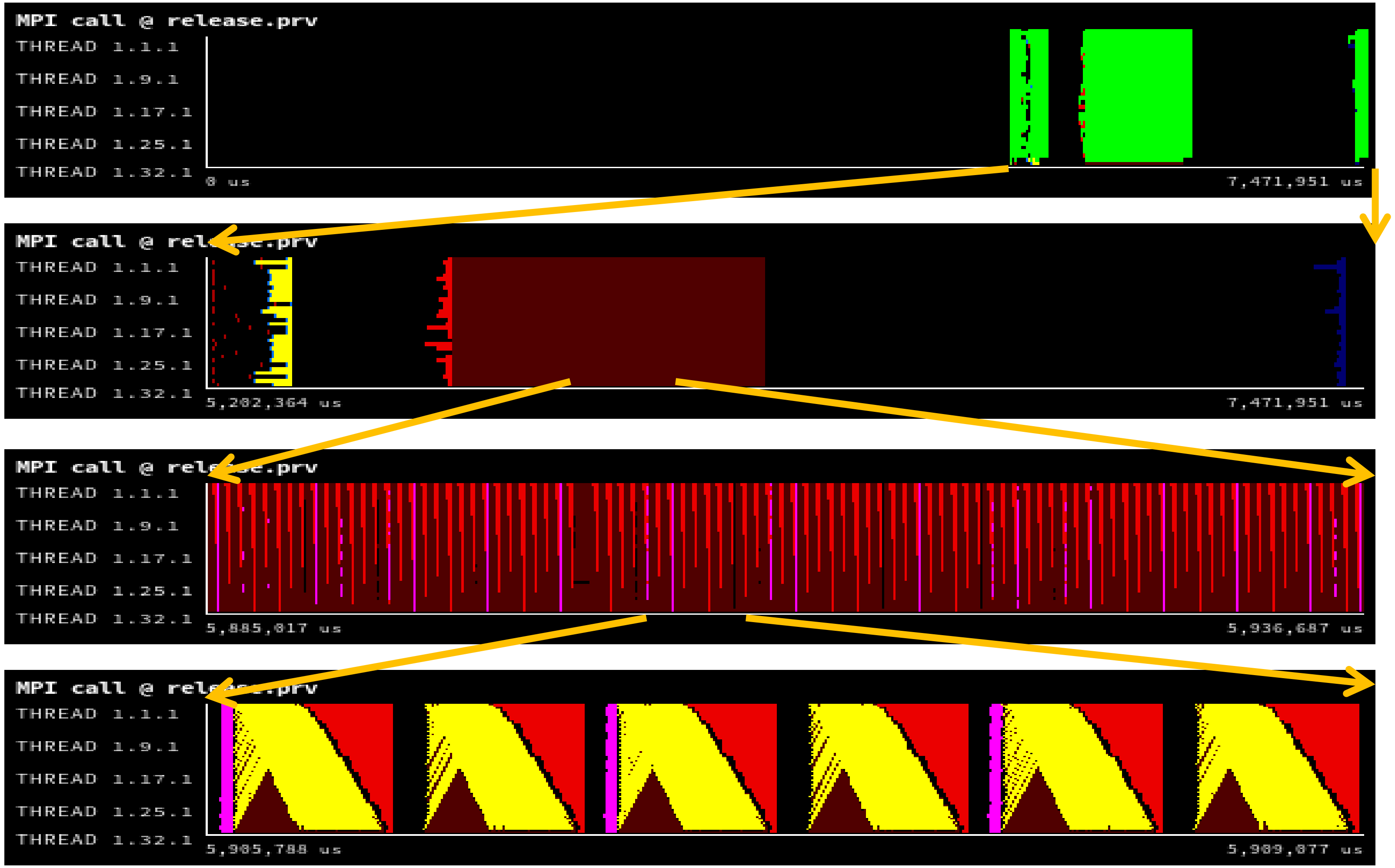


Why efficient?

Parallel efficiency =93.28
Communication = 93.84

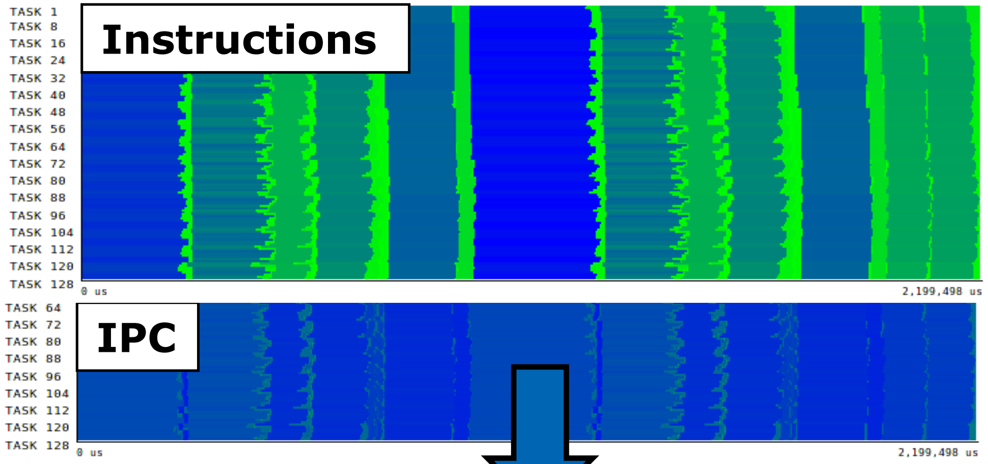
Parallel efficiency = 77.93
Communication = 79.79

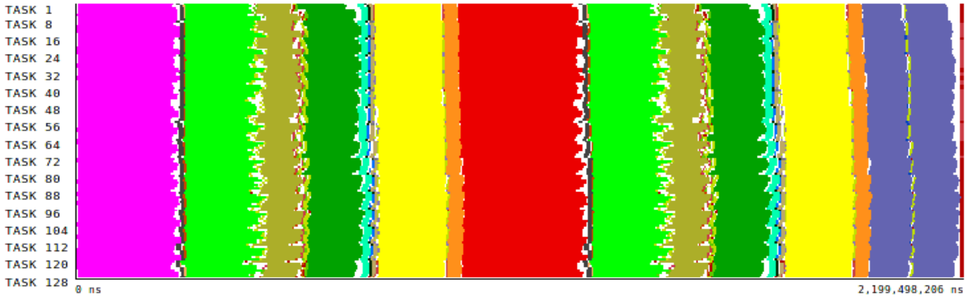
Parallel efficiency = 28.84
Communication = 30.42

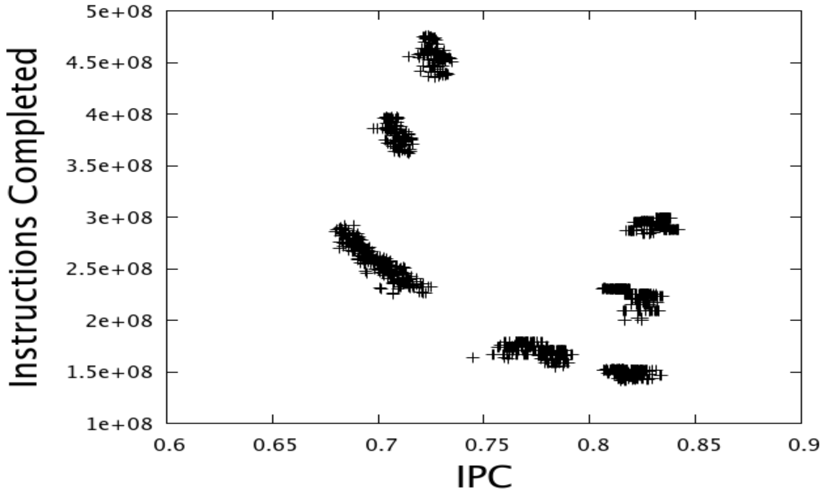


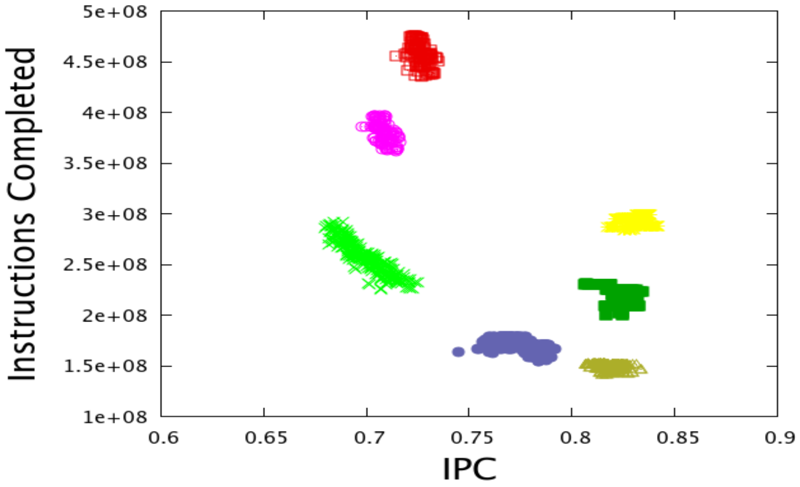
Clustering

Using Clustering to identify structure



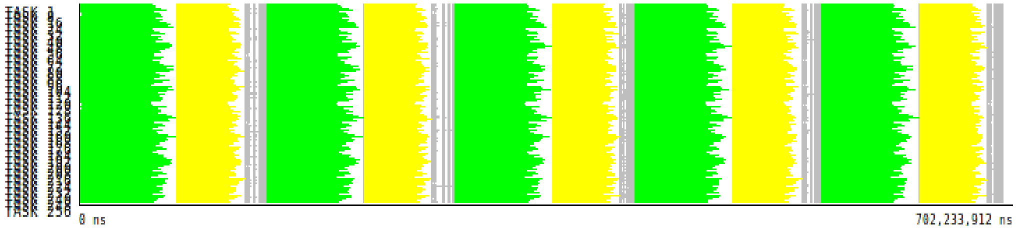






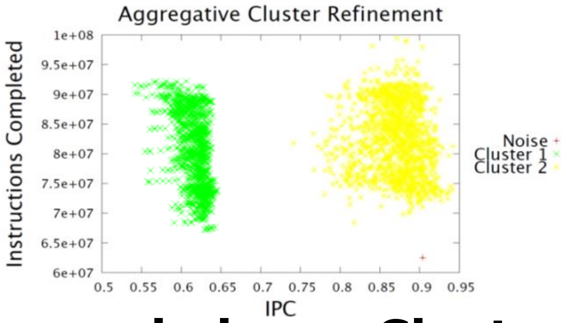
Integrating models and analytics

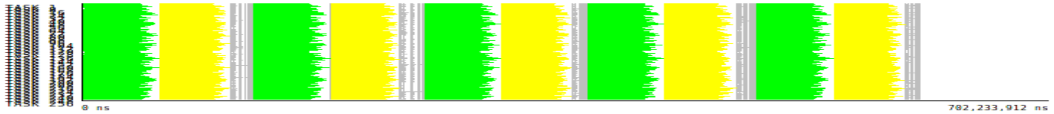
What if ….



PEPC

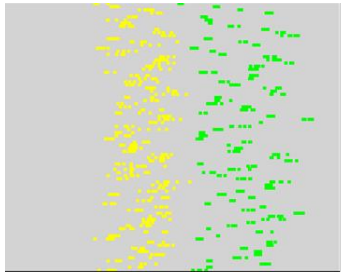
… we increase the IPC of Cluster1?

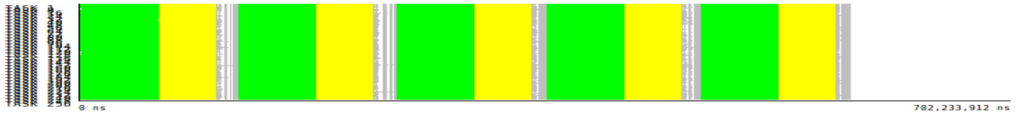




13% gain

… we balance Clusters 1 & 2?

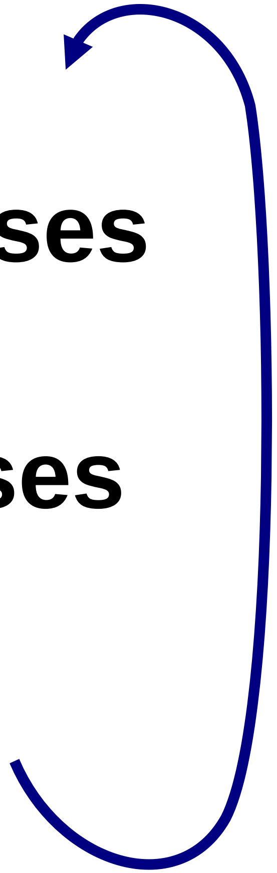




19% gain

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

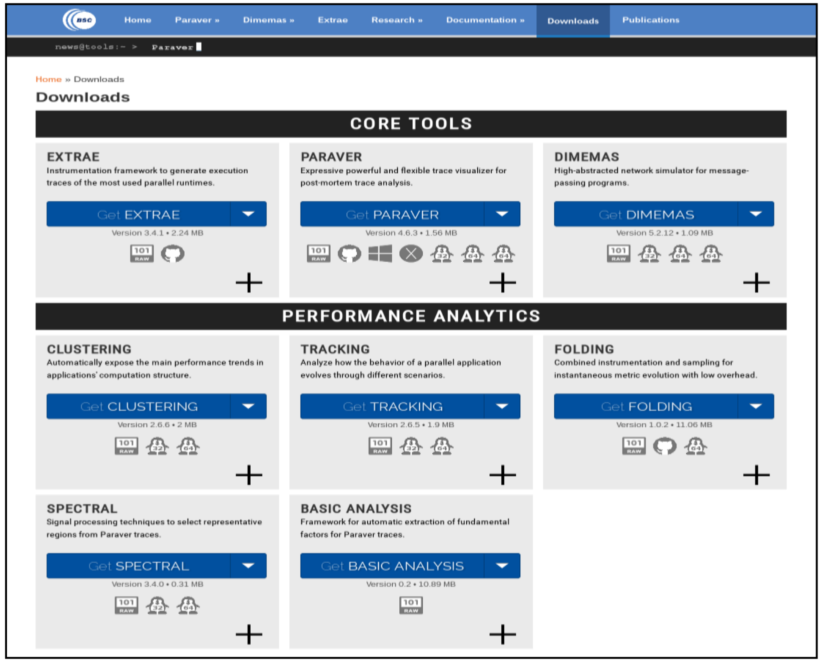
- Scalability – code replication?
  - Total instructions

- Behavioral structure? Variability?

Tutorial:
Introduction to Paraver & Dimemas methodology

BSC Tools web site

- https://tools.bsc.es
  - Downloads
    - – Sources / Binaries
  - Documentation
    - – Training guides
    - – Tutorial slides
- Getting started
  - Start wxparaver
  - Help → Tutorials
  - Follow training guides
    - Paraver introduction (MPI): Navigation & Basic Understanding of Paraver operation



Home » Downloads

Downloads

CORE TOOLS

EXTRAE
Instrumentation framework to generate execution traces of the most used parallel runtimes.
Version 3.4.1 • 2.24 MB

PARAVER
Expressive powerful and flexible trace visualizer for post-mortem trace analysis.
Version 4.6.3 • 1.56 MB

DIMEMAS
High-abstracted network simulator for message-passing programs.
Version 5.2.12 • 1.09 MB

PERFORMANCE ANALYTICS

CLUSTERING
Automatically expose the main performance trends in applications' computation structure.
Version 2.6.6 • 2 MB

TRACKING
Analyze how the behavior of a parallel application evolves through different scenarios.
Version 2.6.5 • 1.9 MB

FOLDING
Combined instrumentation and sampling for instantaneous metric evolution with low overhead.
Version 1.0.2 • 11.06 MB

SPECTRAL
Signal processing techniques to select representative regions from Paraver traces.
Version 3.4.0 • 0.31 MB

BASIC ANALYSIS
Framework for automatic extraction of fundamental factors for Paraver traces.
Version 0.2 • 10.89 MB

Paraver Demo

Same code, different behaviour

Code	Parallel Efficiency	Communication Efficiency	Load Balance Efficiency
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

⚠ **Higher Parallel Efficiency ≠ Faster!**