Barcelona Supercomputing Center
Centro Nacional de Supercomputación

EXCELENCIA SEVERO OCHOA

Modeling Tools

Jesús Labarta

BSC

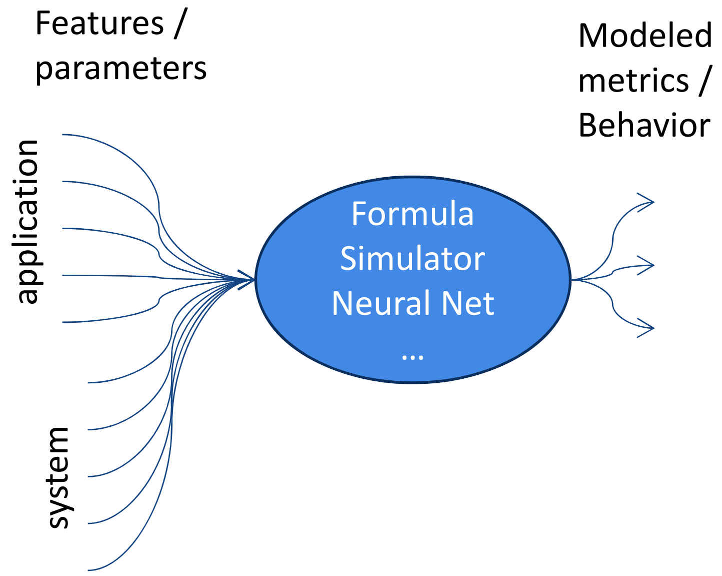
ModSim 2017

Seattle, August 9th 2017

Models

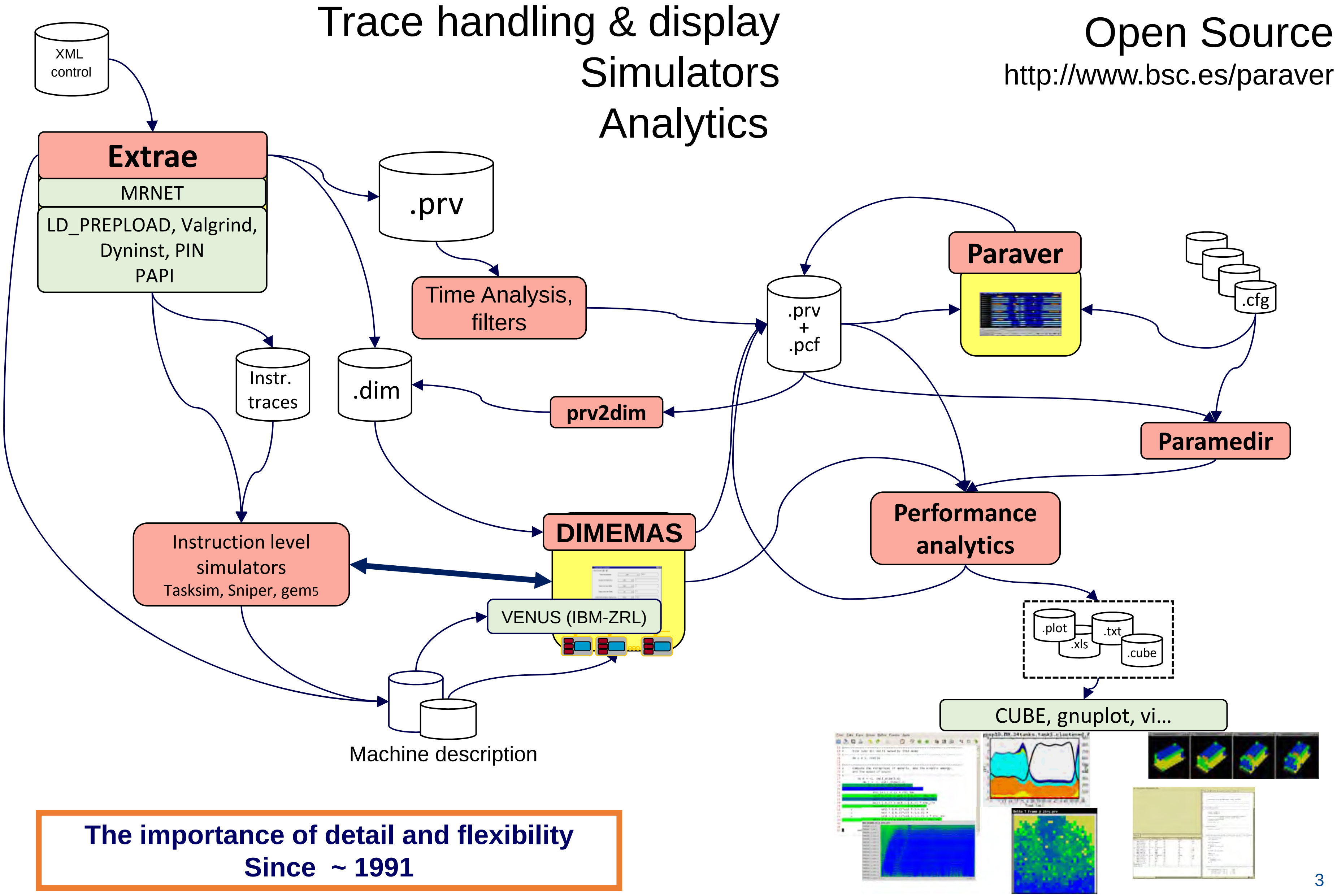
- Model
  - physical, conceptual, or mathematical **representation of a real** phenomenon that is difficult to observe directly
  - a **simplified and idealized** understanding of physical systems

- Used to:
  - **Explain, understand, insight**
    - Current behavior
    - Steer refactoring
    - Co-design runtime
  - Control
  - Predict



A model is as good as the data you use to build it
A model is as good as the use you make of it

BSC Tools framework

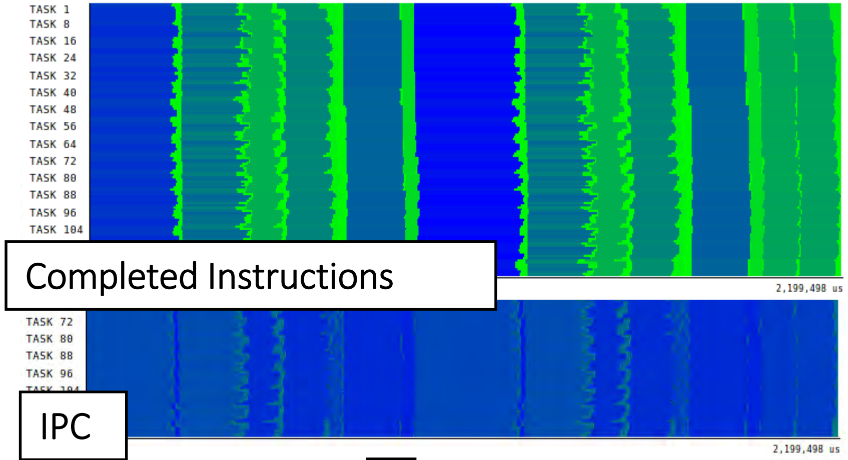


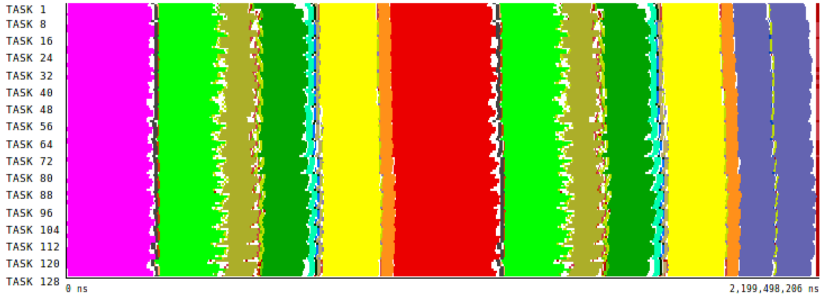
The importance of detail and flexibility
Since ~ 1991

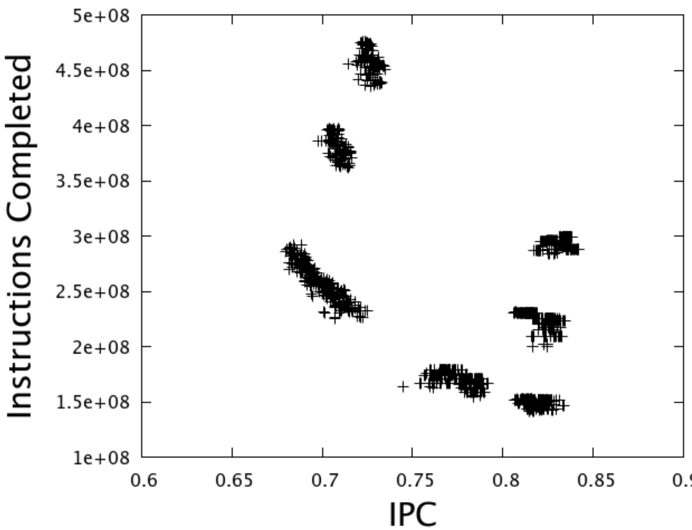
Modeling in our framework …

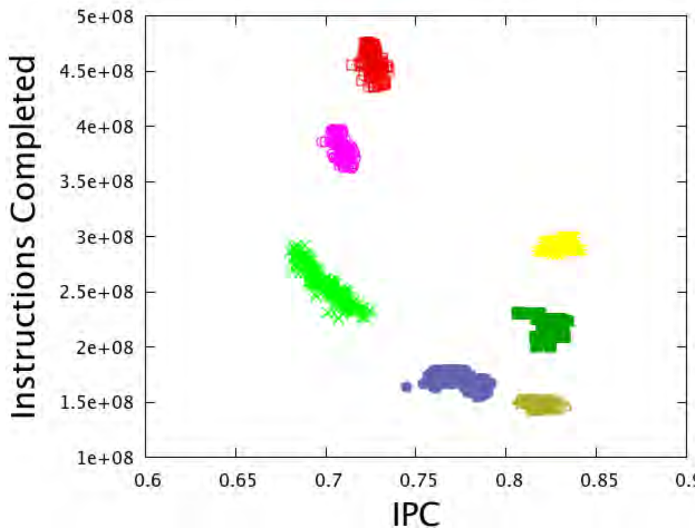
- Trace manipulation
  - Filter, cut, shift, …
- Performance Analytics to detect structure & grain
  - Clustering
  - Tracking
  - Folding
  - spectral
- Modeling tools
  - Dimemas
  - Performance/CPIstack, Power
  - Hardware counters projection
  - Efficiency model
  - Performance extrapolation
- The really important thing:
  - “holistic” integration of data manipulation, analytics and models
  - Towards insight

Clustering to identify structure



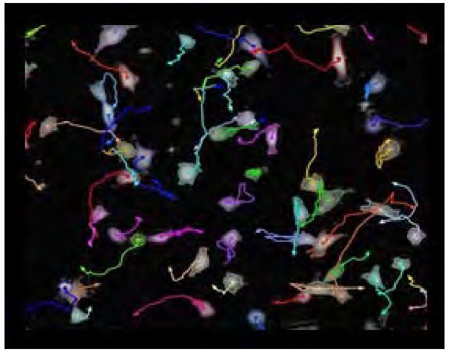




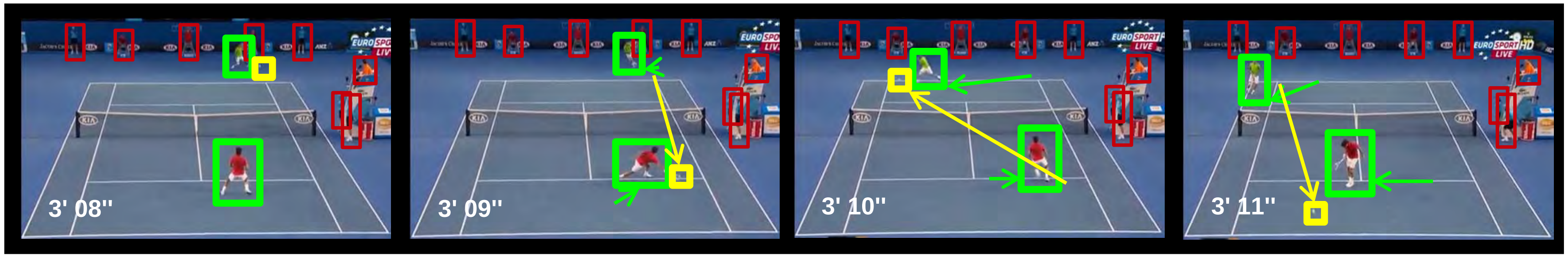


J. Gonzalez et al, "Automatic Detection of Parallel Applications Computation" Phases. (IPDPS 2009)

Object Tracking

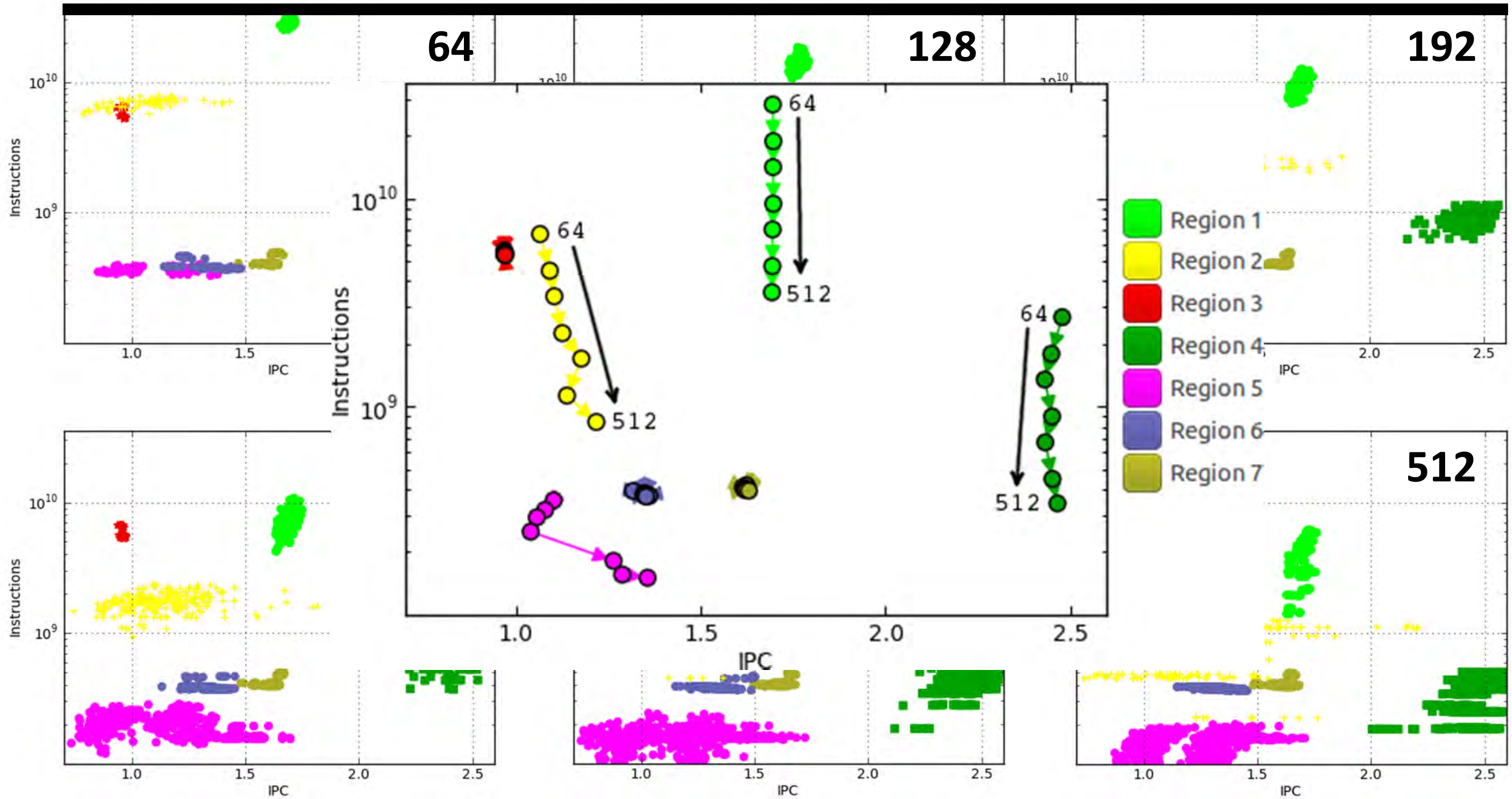
1. Capture a video (sequence of frames).
2. Identify objects of interest within each frame (e.g. contour detection).
3. Correlate the objects between pairs of frames.
   - Rules: characteristics specific to the objects and the physical system (e.g. color, speed)



Tracking structural evolution

- Frame sequence: clustered scatterplot as core counts increases

OpenMX
Strong scaling

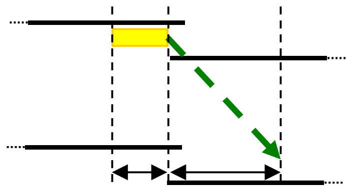


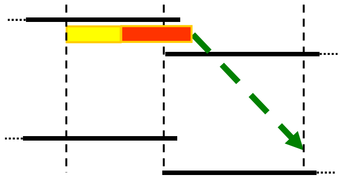
G.Llort et all, "On the Usefulness of Object Tracking Techniques in Performance Analysis", SC 2013

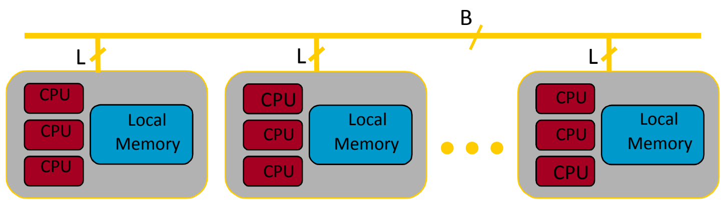
Dimemas: Coarse grain, Trace driven simulation

- Simulation: Highly non linear model
  - Linear components
    - Point to point communication
    - Sequential processor performance
      - Global CPU speed
      - Per block/subroutine
  - Non linear components
    - Synchronization semantics
      - Blocking receives
      - Rendezvous
      - Eager limit
    - Resource contention
      - CPU
      - Communication subsystem
        - links (half/full duplex), busses

$$T = \frac{MessageSize}{BW} + L$$



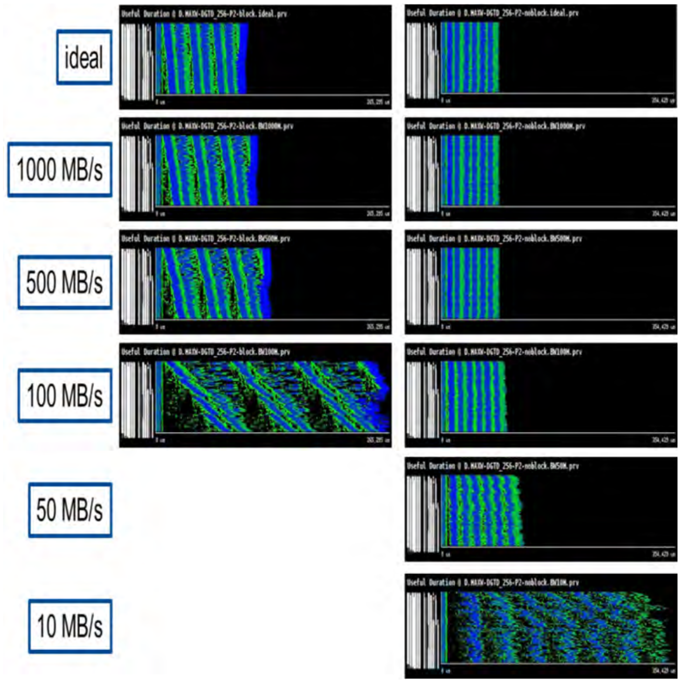




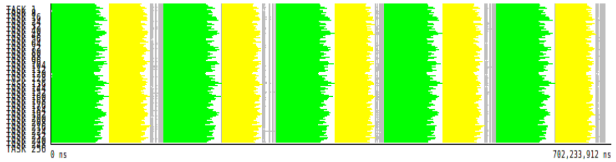
Fast & Fundamentals

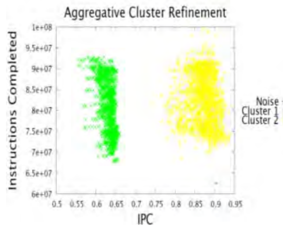
What ifs …

Whatif

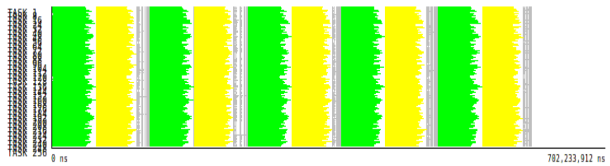


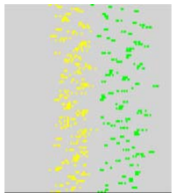
What if …



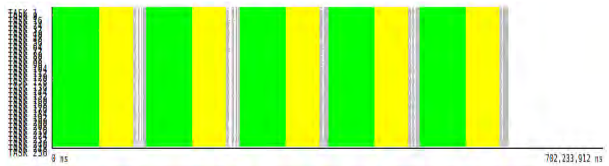


… we increase the IPC of Cluster1?



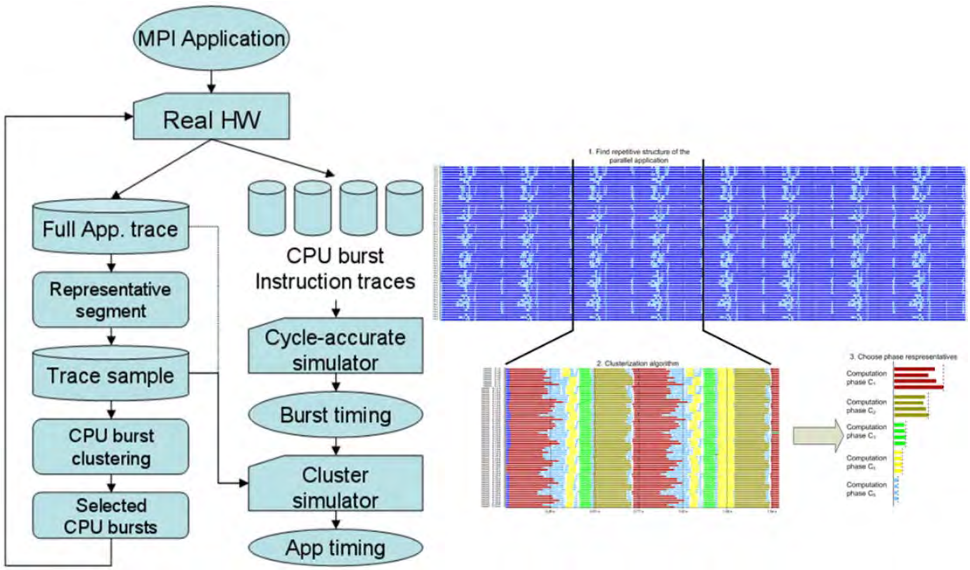


… we balance Clusters 1 & 2?

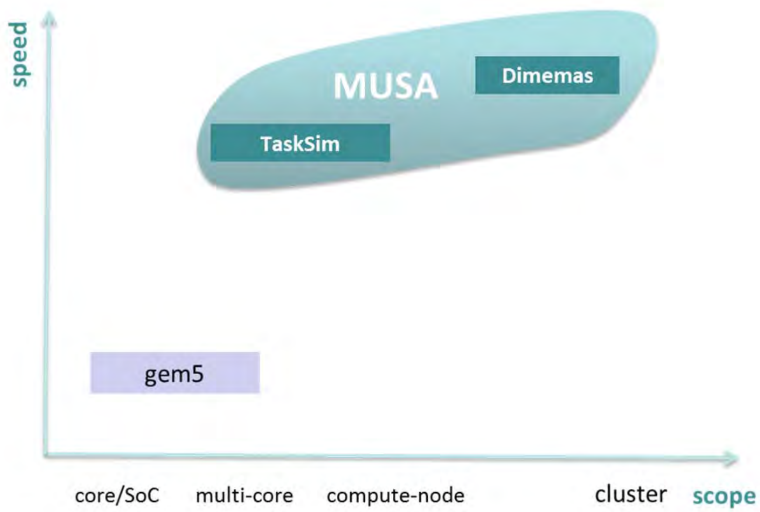


Multiscale Simulation

- Combine use of analytics and simulator
  - Steer what to simulate to reduce total cost
- Combination of trace driven core model, execution driven runtime



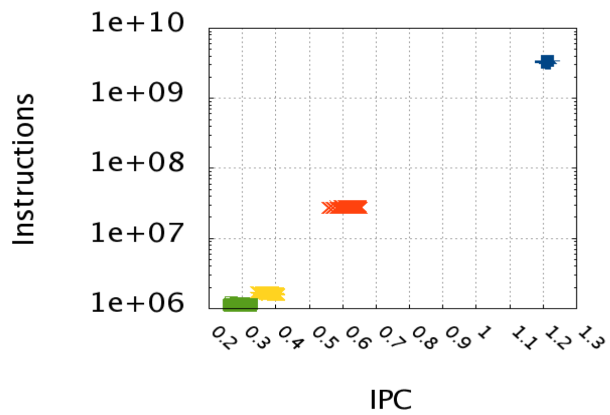
“Simulating Whole Supercomputer Applications”. González, J, et all. IEEE MICRO May 2011

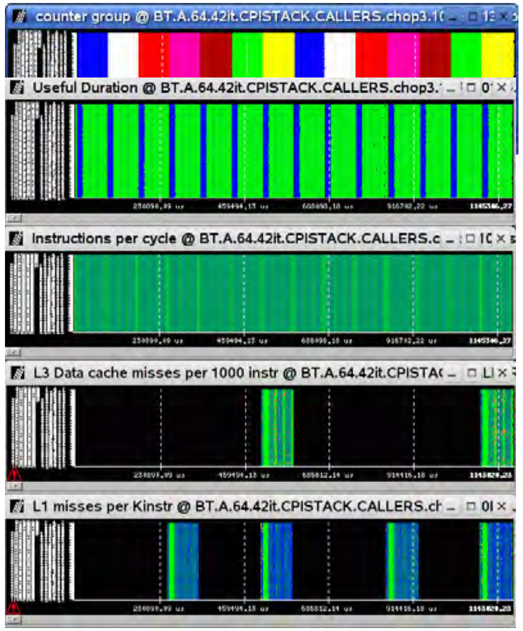


“MUSA: A Multi-level Simulation Approach for Next-Generation HPC Machines”. Grass, T., et all. SC 2016

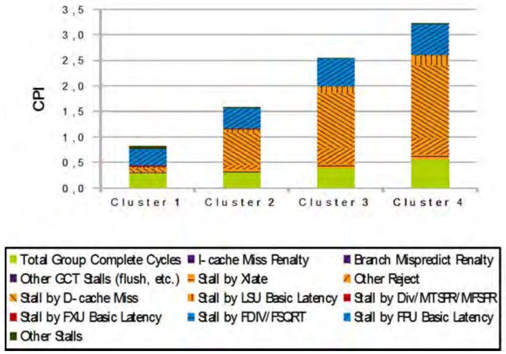
CPI breakdown models

- Analytics: enabler !!



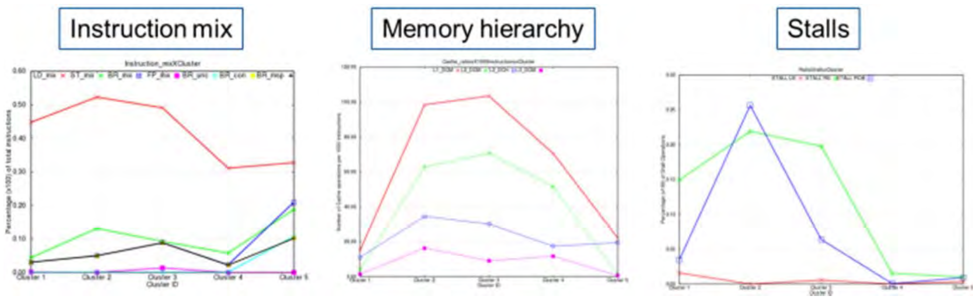


- Vendor models



"CPI analysis on POWER5. Parts 1 & 2"
IBM Systems & Technology Group Systems Performance.
https://www.ibm.com/developerworks/library/pa-cpipower[1-2]

hardwired

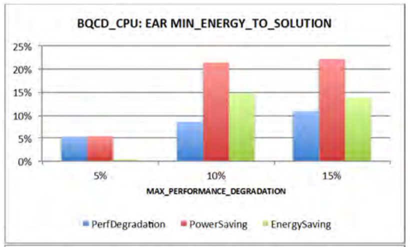


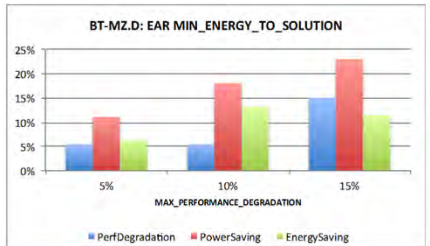
(Re)using - (re)factoring old techniques

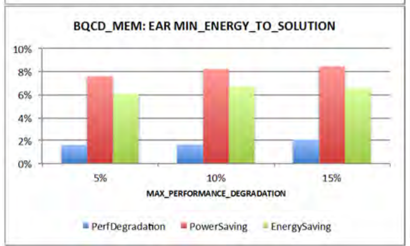
- Energy aware runtime
  - Dynamic periodicity detection (DPD)
  - Performance and Power model

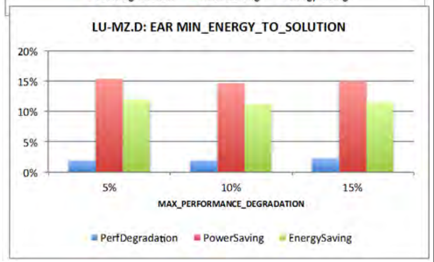
- Dynamic frequency setting policies
  - Minimize one metric with a threshold on other
    - energy to solution
    - time to solution

- Based on periodic structure detection









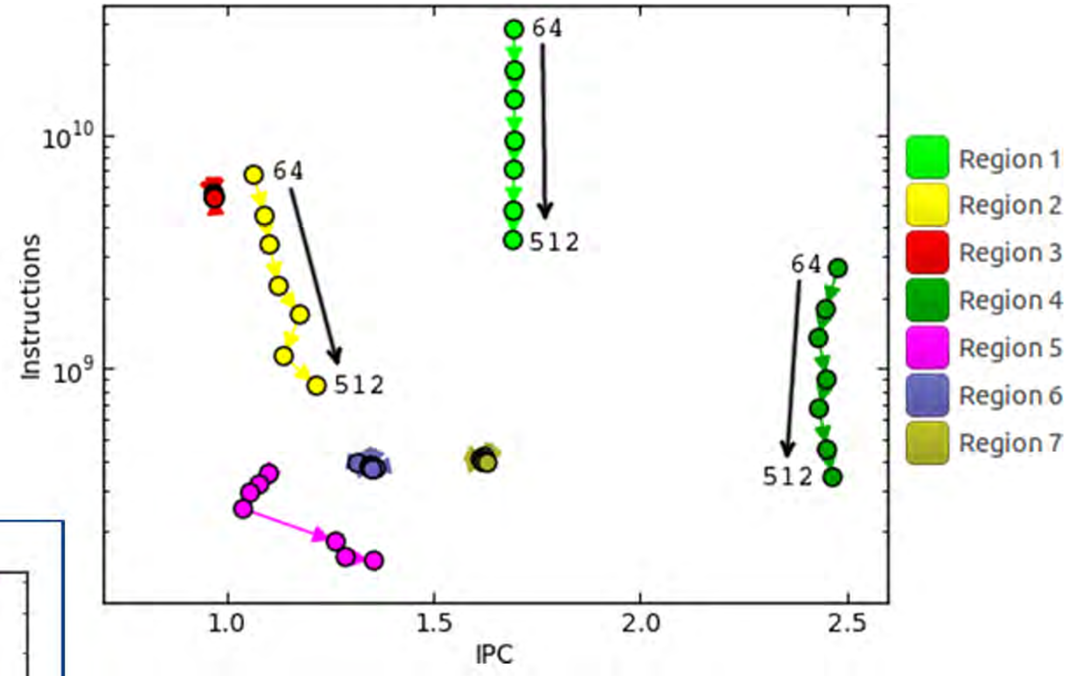
J. Corbalan et al. "Performance-Driven Processor Allocation" OSDI 2000

F. Freitag et al. "A Dynamic Periodicity Detector: Application to Speedup Computation" IPDPS 2001

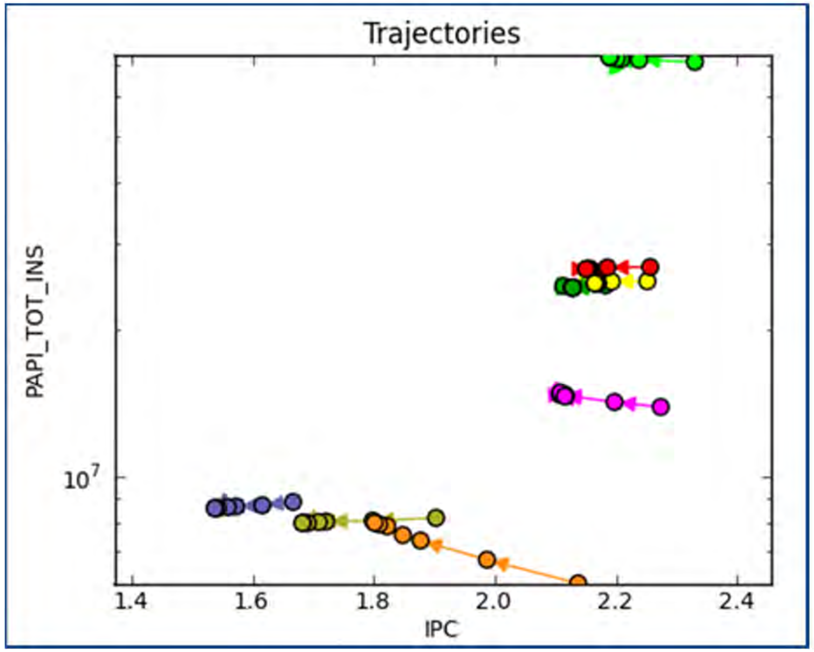
Lenovo - BSC

Simple model fitting

- Constant
- Linear



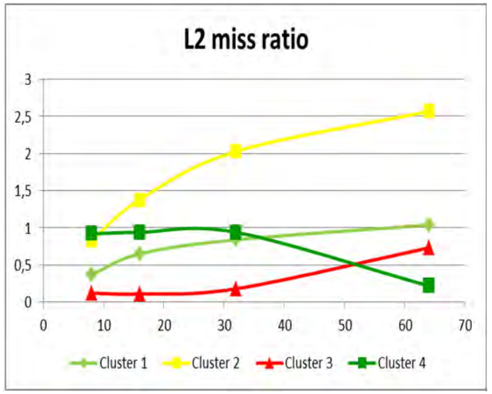
OpenMX

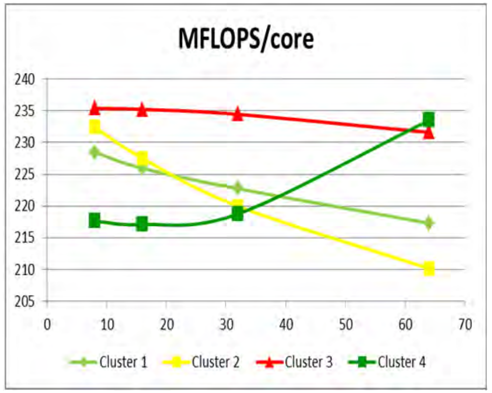


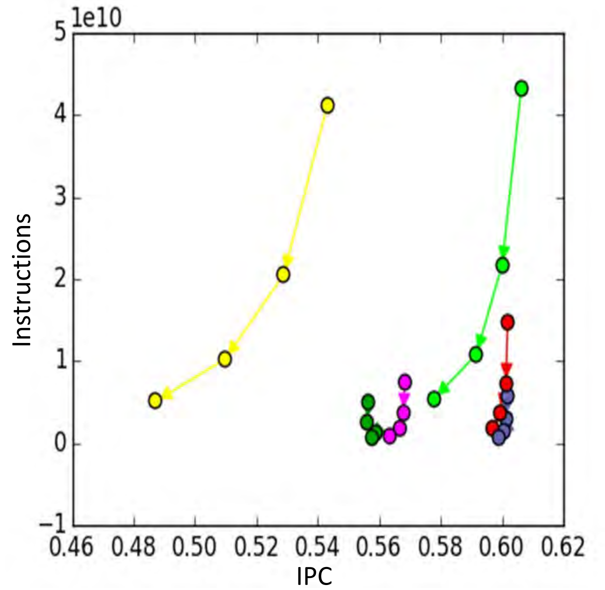
AMG2013

OpenIFS @ Thunder-X1

- Strong scaling tracking of IPC
  - Sharing effect: **L2**, BW, …

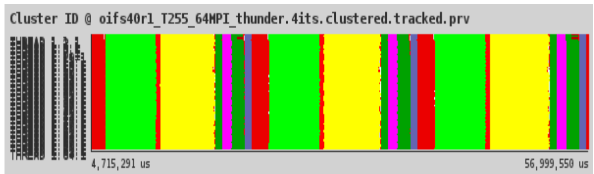






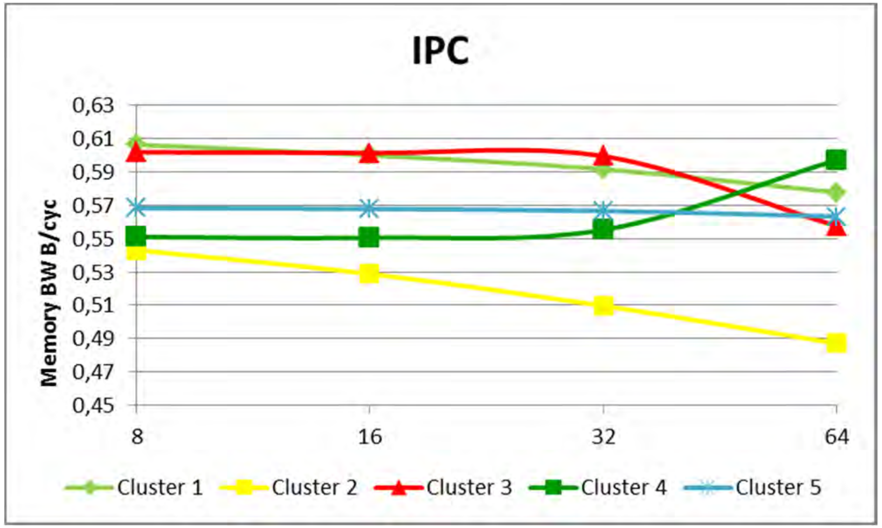
- Core Cost Model
  - Ideal CPI (1.6) , L2 hit cost (62.2) , L2 miss cost (129,4)
  - “reverse” engineering form real production code runs

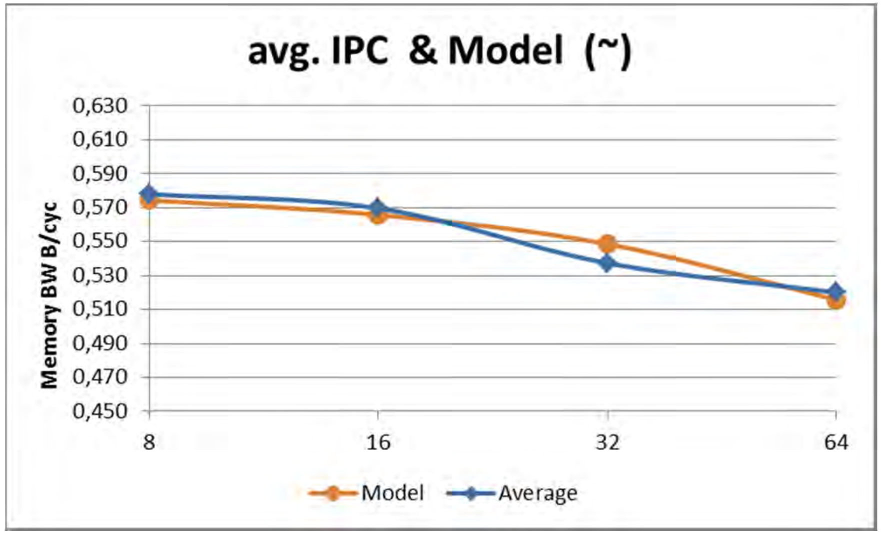
- Socket IPC: ~ 26 !!

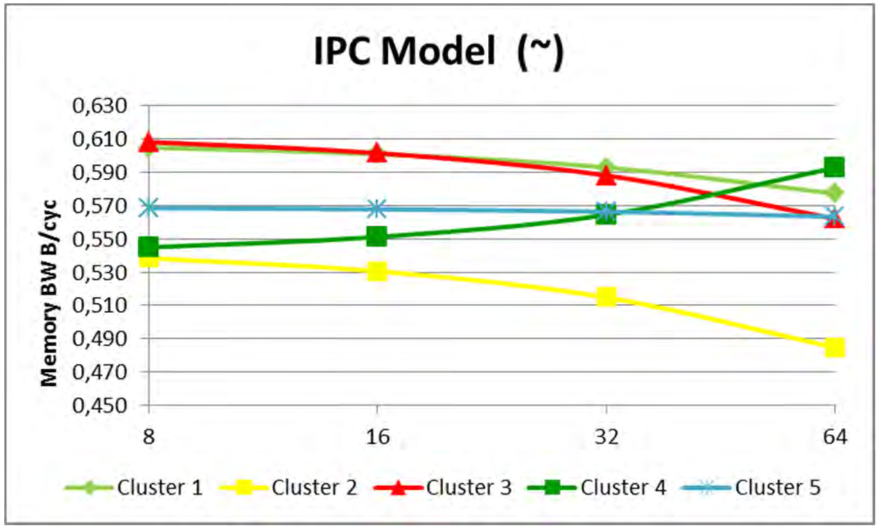


OpenIFS @ Thunder-X1

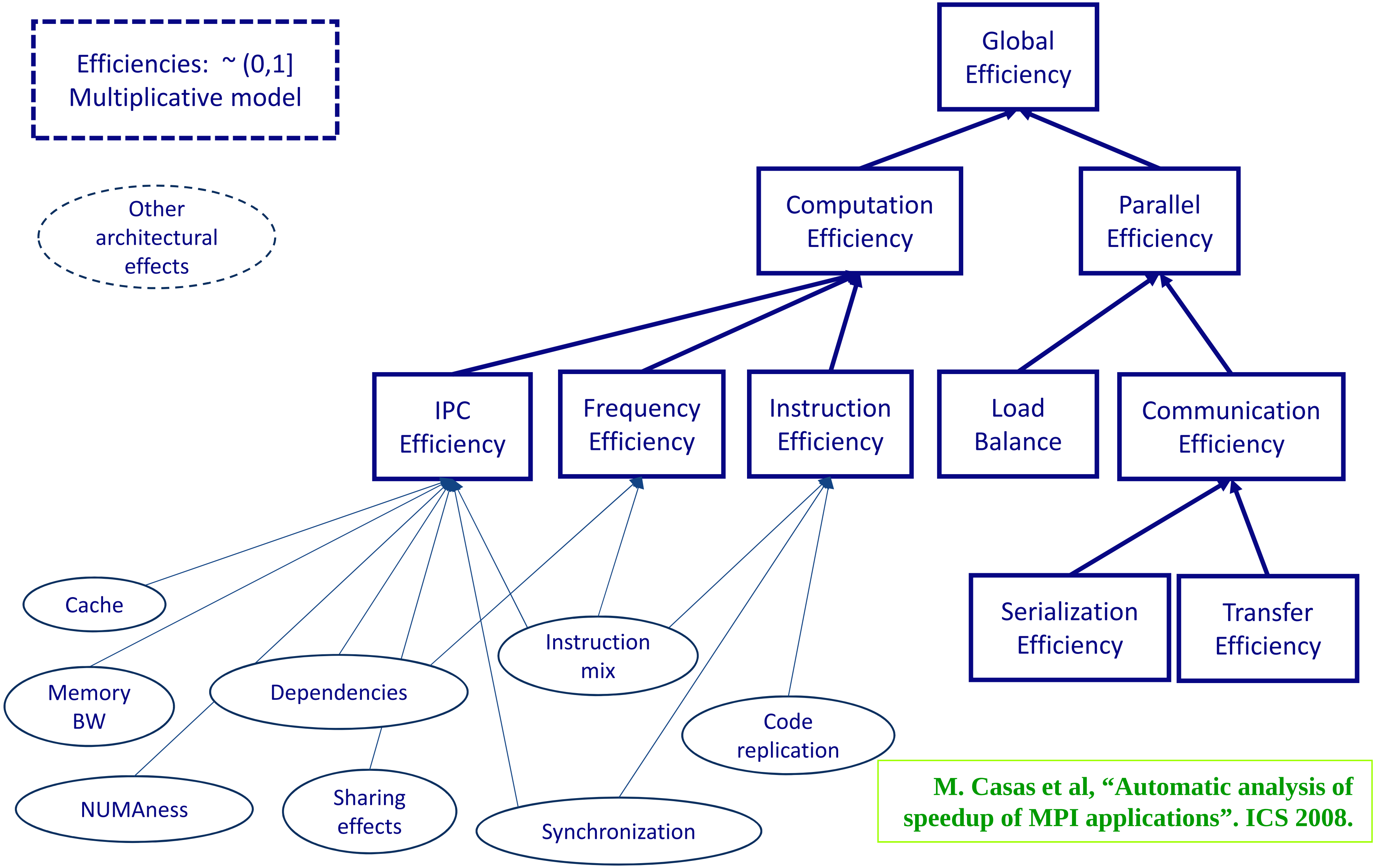
- Fit IPC
  - Numerical ?
  - Behavioral ?
  - Interpretation ?





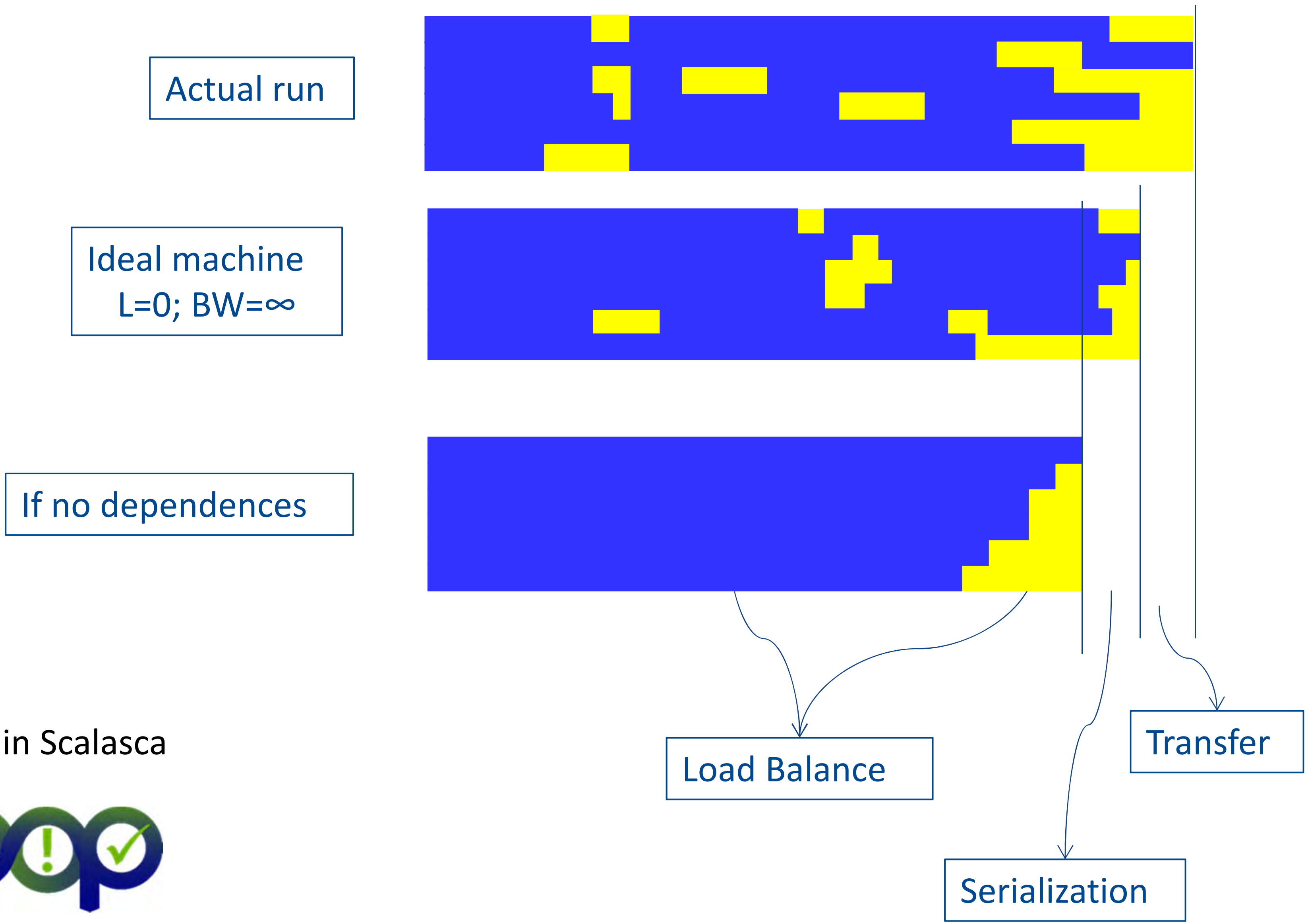


Application characterization



M. Casas et al, “Automatic analysis of speedup of MPI applications”. ICS 2008.

Scaling model



Also in Scalasca

Efficiencies

	2	4	8	16
Parallel Efficiency	0.9834	0.9436	0.8980	0.8478
Load Balance	0.9871	0.9687	0.9099	0.9177
Serialization efficiency	0.9975	0.9770	0.9938	0.9395
Transfer Efficiency	0.9988	0.9970	0.9931	0.9833
Computation Efficiency	1.000	0.9590	0.8680	0.6953
Global efficiency	0.9834	0.9049	0.7795	0.5894

	2	**4**	**8**	**16**
IPC Scaling Efficiency	1.000	0.9932	0.9591	0.8421
Instruction Scaling Efficiency	1.000	0.9721	0.9393	0.9075
Core frequency efficiency	1.000	0.9932	0.9635	0.9098

On performance portabilility/predictability

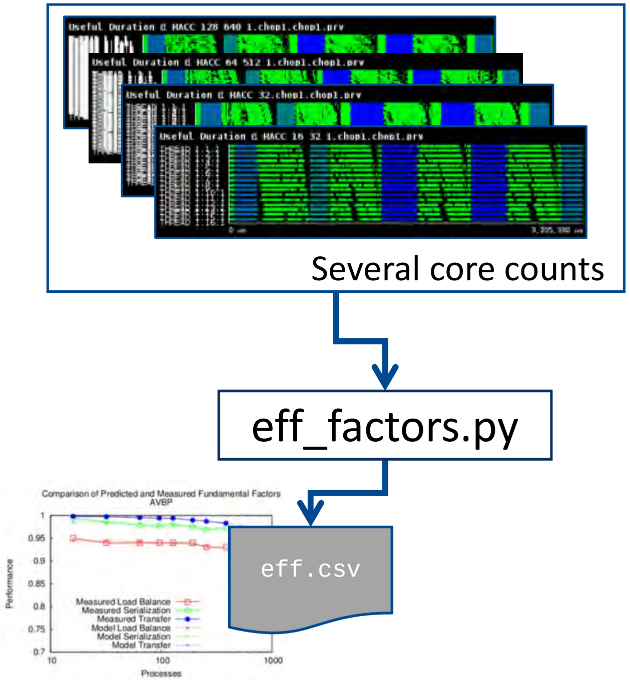
- Analyses from different training courses
  - Running lulesh on default system setup

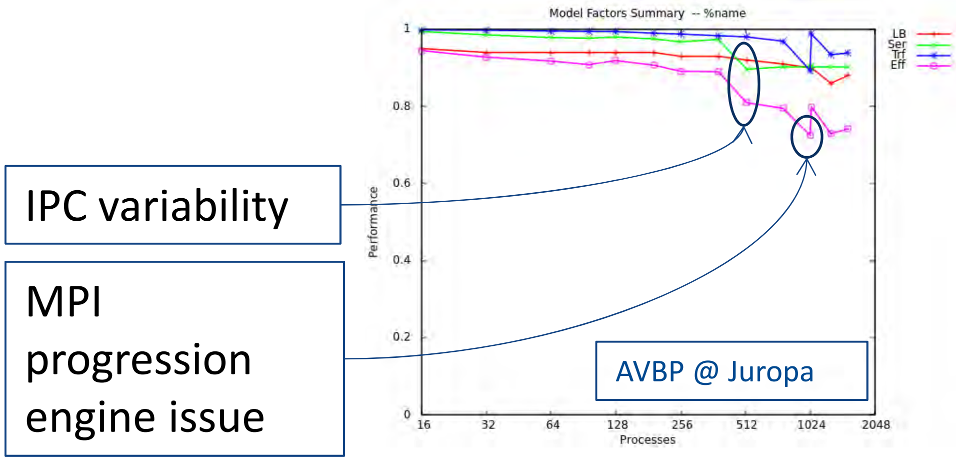
Code	Parallel efficiency	Communication efficiency	Load Balance efficiency
lulesh@machine1	90.55	**99.22**	91.26
lulesh@machine2	**69.15**	99.12	**69.76**
lulesh@machine3	70.55	96.56	73.06
lulesh@machine4	83.68	95.48	87.64
lulesh@machine5	**90.92**	98.59	**92.20**
lulesh@machine6	73.96	97.56	75.81
lulesh@machine7	75.48	**88.84**	84.06
lulesh@machine8	77.28	92.33	83.70
lulesh@machine9	88.20	98.45	89.57
lulesh@machine10	81.26	91.58	88.73

Huge variability. How will the next machine behave?

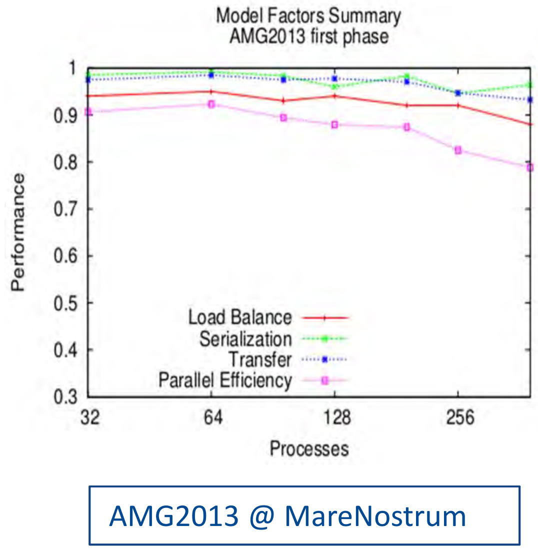
J. Gimenez, "it is me, or it is the machine?". Scalable Tools Workshop 2017.

Powerful analysis …

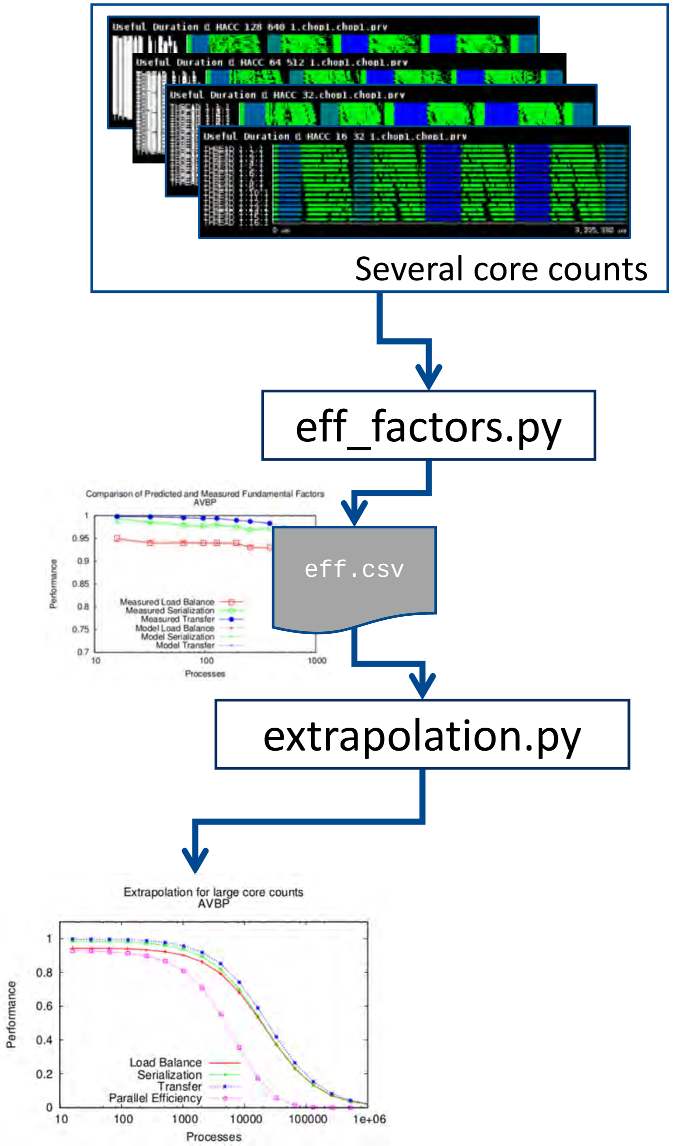




- Identify relevance of different factors …
- … trends and outliers
- In conjunction with detailed analysis of individual points



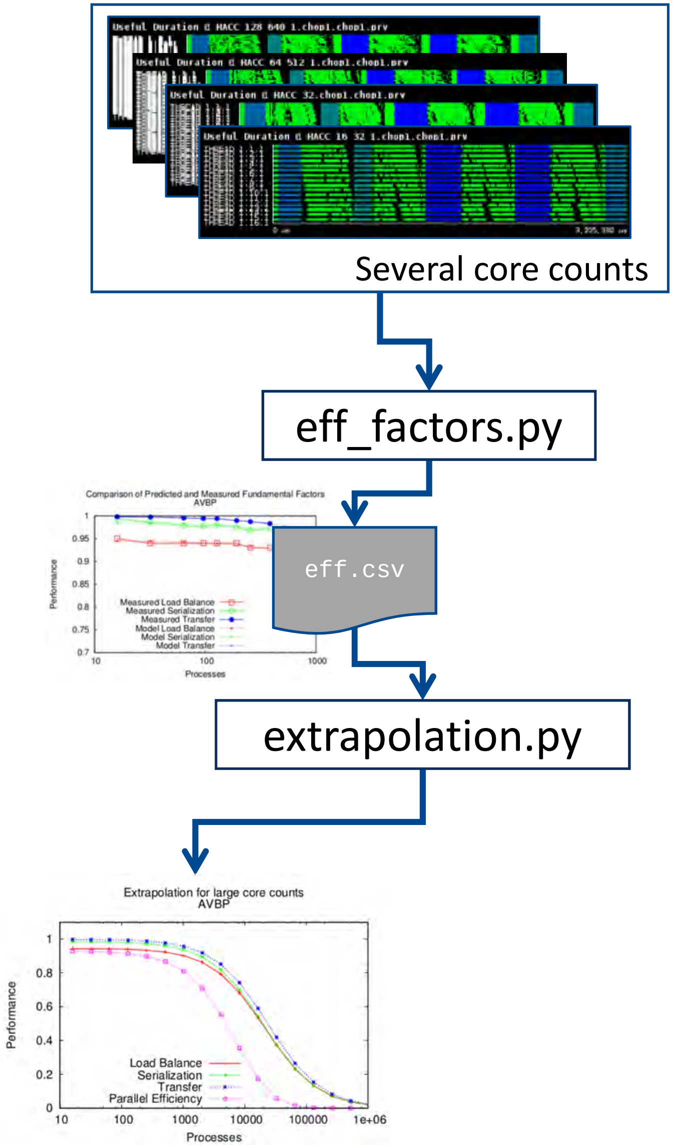
Extrapolations …



"Scalability prediction for fundamental performance factors "
J. Labarta et al.
SuperFRI 2014

Intel – BSC
Exascale Lab

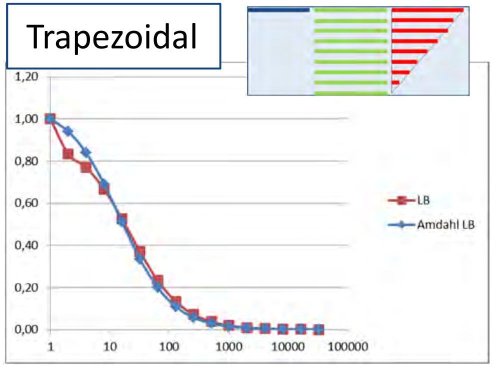
Extrapolations …

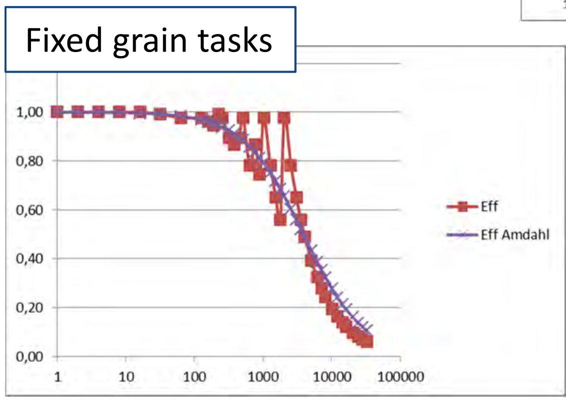


- Amdahl fit …

$$Amdahl_{fit} = \frac{metric_0}{f_{metric} + (1 - f_{metric}) * P}$$

- …reasonable in the absence of more detailed behavioral information

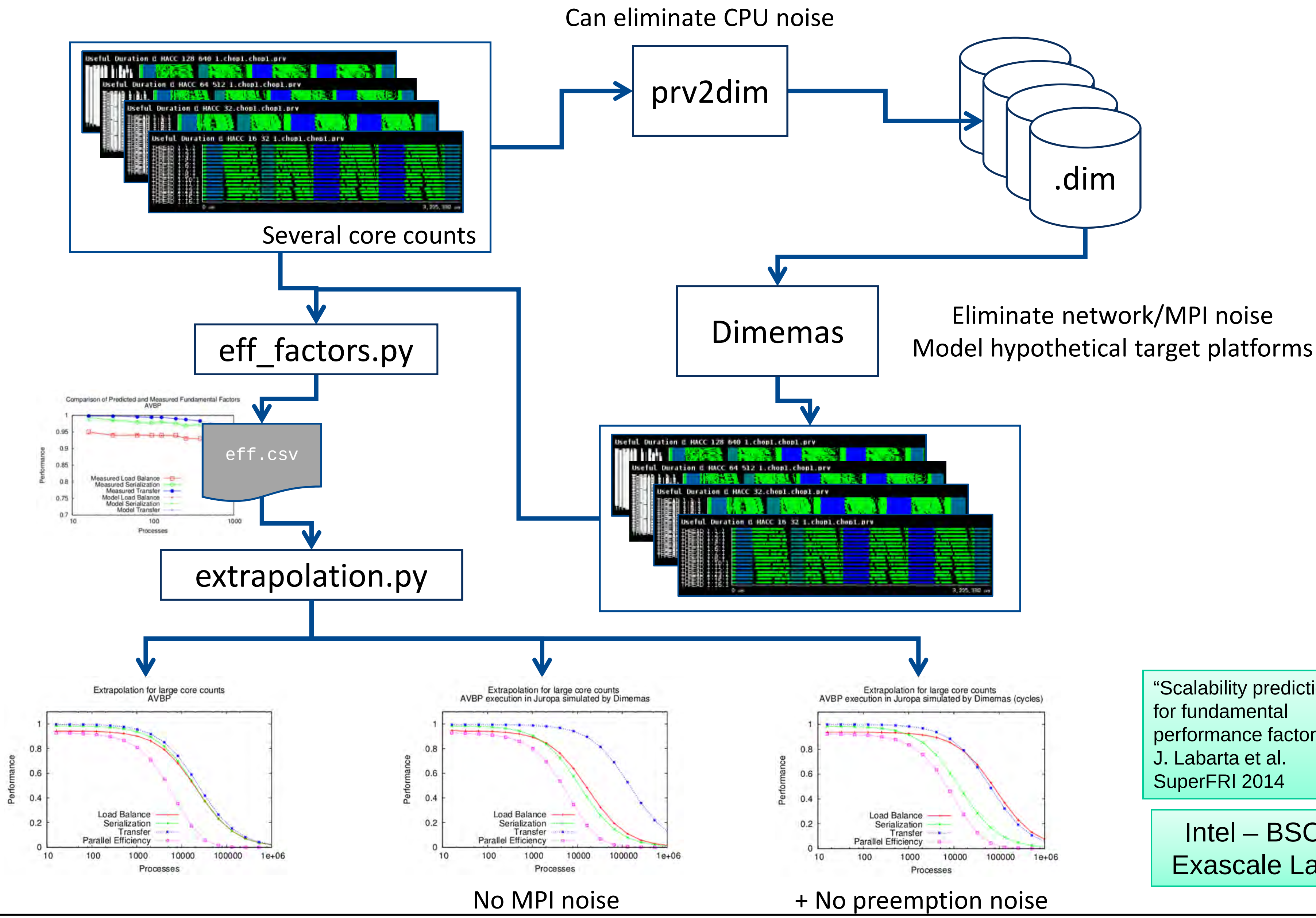




“Scalability prediction for fundamental performance factors ”
J. Labarta et al.
SuperFRI 2014

Intel – BSC
Exascale Lab

Powerful what ifs …



"Scalability prediction for fundamental performance factors"
J. Labarta et al.
SuperFRI 2014

Intel – BSC Exascale Lab

About Analytics AND infrastructure

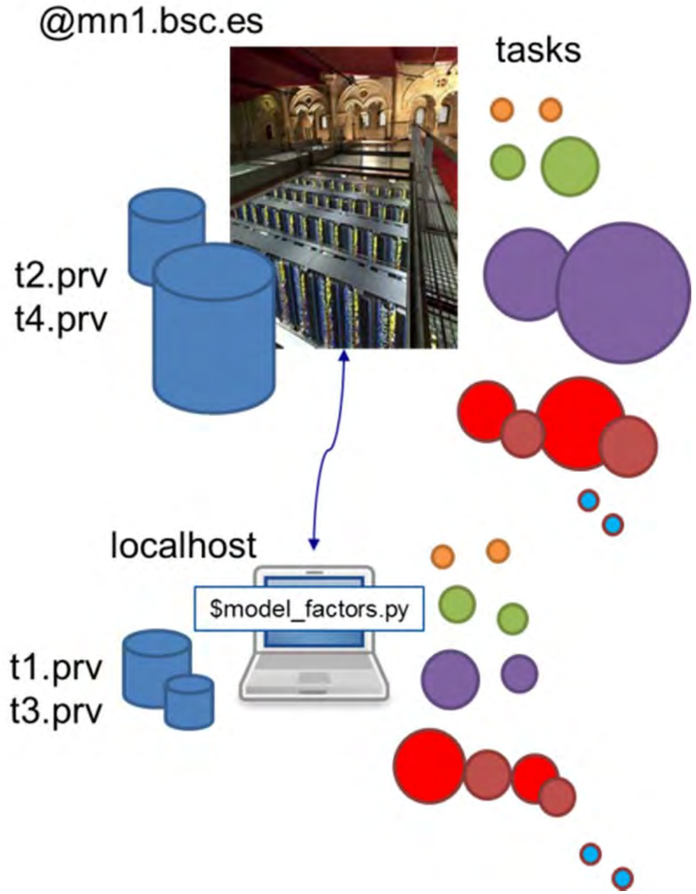
PyCOMPSs

Main

```
for name in list_traces:
    prv = name
    row = prv[:-4] + '.row'
    pcf = prv[:-4] + '.pcf'

    dim = os.path.basename(prv)[:-4] + '.dim'
    prv_ideal = 'D.' + os.path.basename(prv)
    pcf_ideal = 'D.' + os.path.basename(pcf)
    row_ideal = 'D.' + os.path.basename(row)

    np = get_num_prcs_trace(prv)

    prv2dim(prv, pcf, row, dim)

    dimemas(pcf, row, dim, prv_ideal, pcf_ideal, row_ideal,
        head_file, tail_file, coll4dim, dim_ideal_sim)

    ulb = paramedir(prv_ideal, pcf_ideal, row_ideal,
        '2dp_mpi_stats.cfg')
    lb, trf = paramedir(prv, pcf, row, '2dp_mpi_stats.cfg')

    collect_per_trace.append([np, ulb, lb, trf])

from pycompss.api.api import compss_wait_on
    collect_per_trace = compss_wait_on(collect_per_trace)
```

Tasks

```
@task(trace_prv=FILE_IN, trace_pcf=FILE_IN,
trace_row=FILE_IN, cfg=FILE_IN, returns=list)
def paramedir (trace_prv, trace_pcf, trace_row, cfg,
                trace_id_name):
    //Body of function
```



Performance analytics and modeling towards insight

- The importance of Integrated data handling, analytics and models

- Balance between first principles, pure statistical, black box ?

    - “Proper” choice of feature vector

- We use models much less than desirable !!!
    - → we actually use unquantified mental models
    - Training / Best practices issue

- A couple of “recommendations”
    - Performance tools: leverage methods from ALL areas & big data infrastructures
    - Runtimes: malleability and adaptability

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

EXCELENCIA SEVERO OCHOA

Thank you

Jesus.labarta@bsc.es

07/07/2017