The POP Project

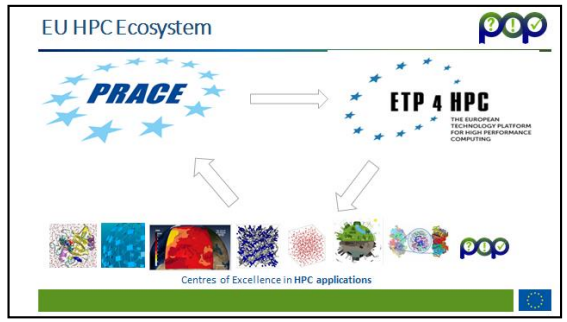
Jesús Labarta (BSC)

ISC 2016 BOF

Frankfurt, June 22rd, 2016

POP CoE

- A **Center of Excellence**
  - On **Performance Optimization and Productivity**
  - Promoting **best practices in parallel programming**
- Providing **Services**
  - Precise understanding of application and system behavior
  - Suggestion/support on how to refactor code in the most productive way
- **Horizontal**
  - Transversal across application areas, platforms, scales
- **For academic AND industrial codes and users !**



Partners

- **Who?**
  - BSC (coordinator), ES
  - HLRS, DE
  - JSC, DE
  - NAG, UK
  - RWTH Aachen, IT Center, DE
  - TERATEC, FR

A team with

- Excellence in performance tools and tuning
- Excellence in programming models and practices
- Research and development background AND
  proven commitment in application to real academic and industrial use cases

Motivation

Why?

- Complexity of machines and codes
  → Frequent lack of quantified understanding of actual behavior
  → Not clear most productive direction of code refactoring
- Important to maximize efficiency (performance, power) of compute intensive applications and the productivity of the development efforts

Target

- Parallel programs , mainly MPI /OpenMP … although can also look at CUDA, OpenCL, Python, …

3 levels of services

Apply @
http://www.pop-coe.eu

? Application Performance Audit

- Primary service
- Identify performance issues of customer code (at customer site)
- Small Effort (< 1 month)

! Application Performance Plan

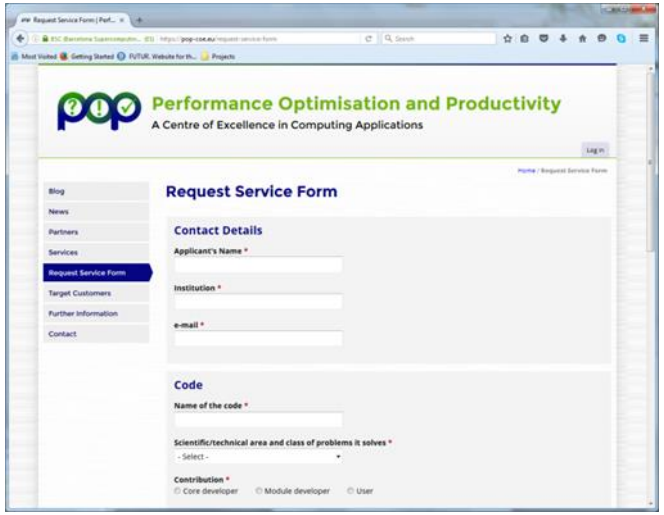
- Follow-up on the service
- Identifies the root causes of the issues found and qualifies and quantifies approaches to address the issues
- Longer effort (1-3 months)

✓ Proof-of-Concept

- Experiments and mock-up tests for customer codes
- Kernel extraction, parallelization, mini-apps experiments to show effect of proposed optimizations
- 6 months effort

Reports

Software demonstrator



+ Training & support

- Better follow analyses
- Do it yourself advanced users

Target customers

- **Code developers**
  - Assessment of detailed actual behavior
  - Suggestion of most productive directions to refactor code
- **Users**
  - Assessment of achieved performance in specific production conditions
  - Possible improvements modifying environment setup
  - Evidence to interact with code provider
- **Infrastructure operators**
  - Assessment of achieved performance in production conditions
  - Possible improvements from modifying environment setup
  - Information for computer time allocation processes
  - Training of support staff
- **Vendors**
  - Benchmarking
  - Customer support
  - System dimensioning/design

Best practices in Performance analysis

- **Powerful tools …**
  - Extrae + Paraver
  - Score-P + Scalasca/TAU/Vampir + Cube
  - Dimemas, Extra-P
  - Other commercial tools
- **… and techniques**
  - Clustering, modeling, projection, extrapolation, memory access patterns,
  - … with extreme detail …
  - … and up to extreme scale
- **Unify methodologies**
  - Structure
    - Spatio temporal / syntactic
  - Metrics
    - Parallel fundamental factors: Efficiency, Load balance, Serialization
    - Programming model related metrics
    - User level code sequential performance
  - Hierarchical search
    - From high level fundamental behavior to its causes
- **To deliver insight**
- **To estimate potentials**

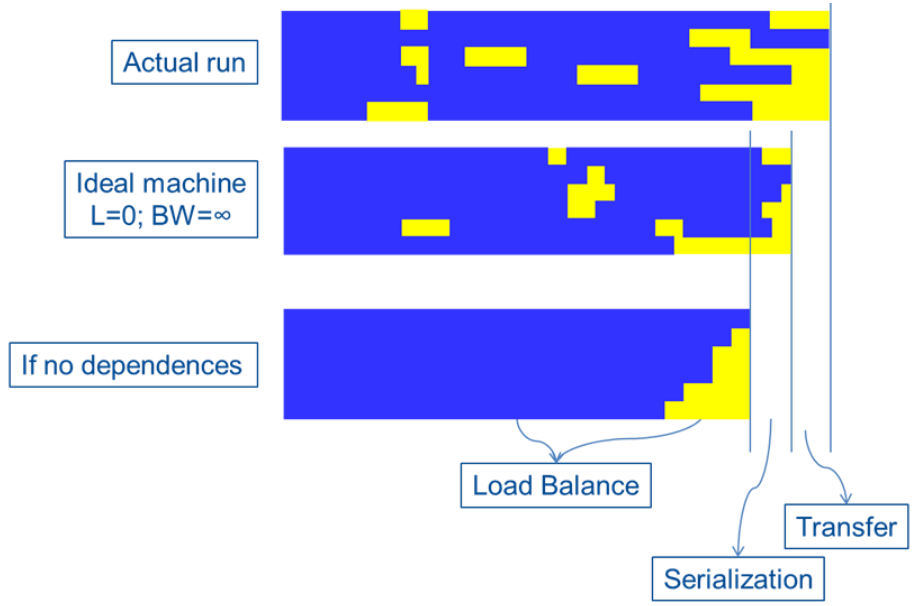
Fundamental performance factors

- Factors modeling parallel efficiency
  - **Load balance** (LB)
  - **Communication**
    - **Serialization** (or Micro load balance)
    - **Transfer**

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

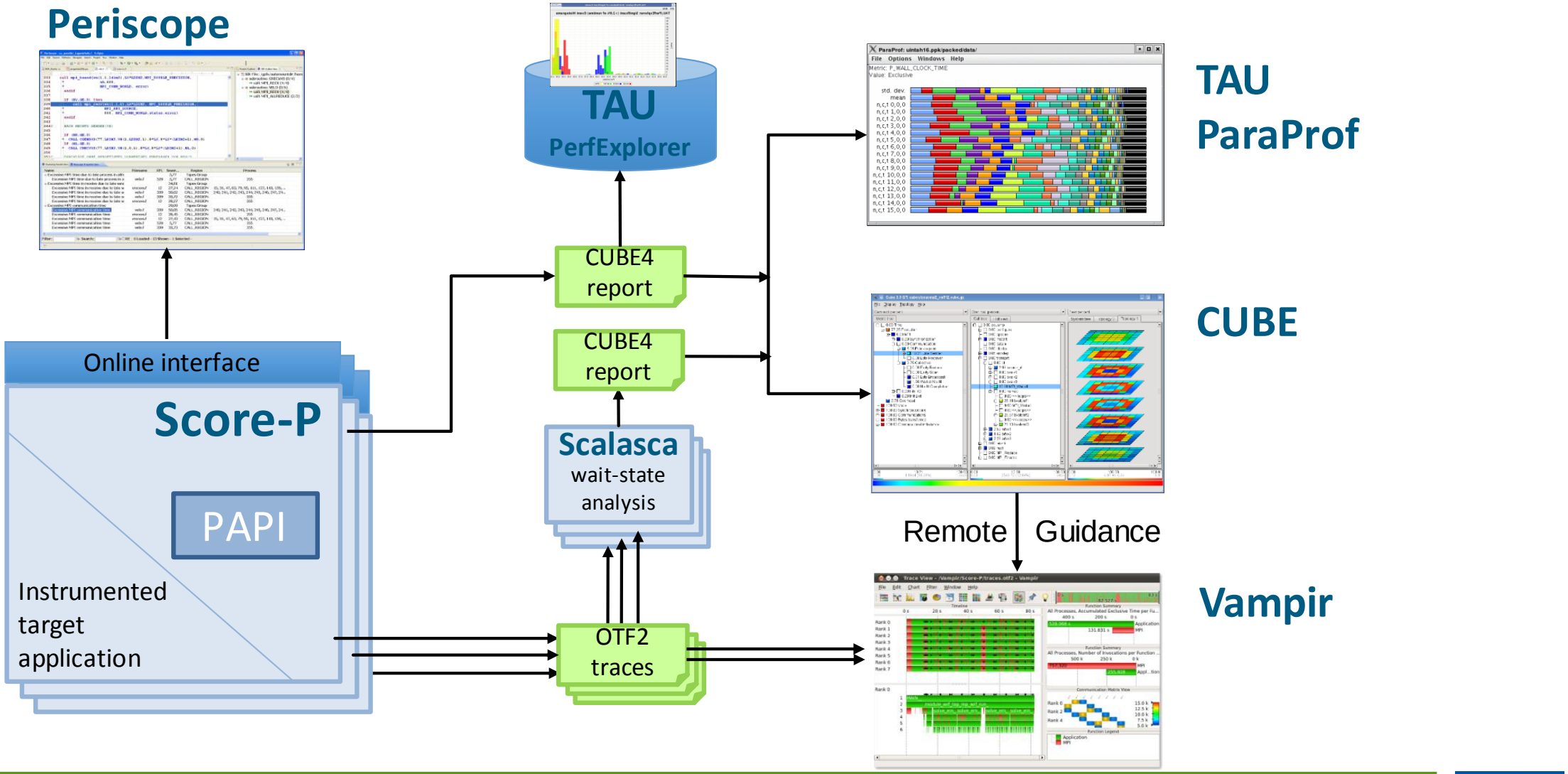
CommEff

- Factors describing serial behavior
  - Computational complexity: **#instr**
  - Performance: **IPC**
  - **Core frequency**
  - Actual values, scaling behavior, impact on parallel efficiency factors



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

Score-P Tool Ecosystem -- Overview



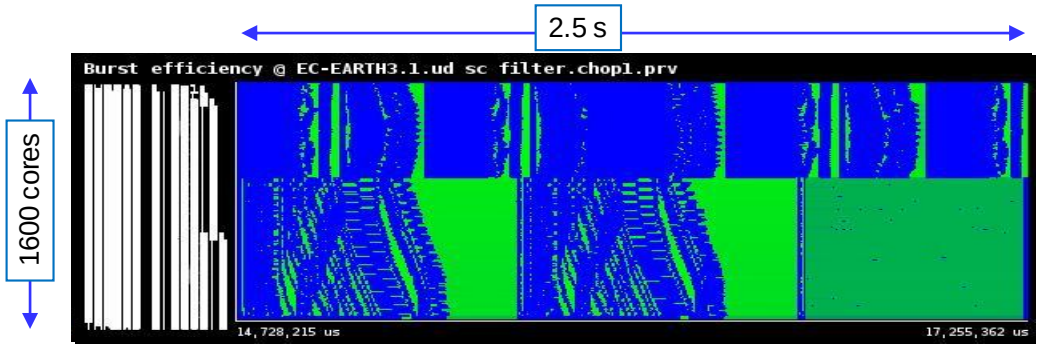
Score-P Tool Ecosystem -- Status

- **Score-P** (www.score-p.org)
  - Parallel Program Instrumentation and Profile/Trace Measurement
    - MPI, OpenMP, SHMEM, CUDA, OpenCL, OmpSs support
  - Latest version: 3.0-rc1
    - New: User function sampling + MPI measurement, OpenACC support
- **Scalasca** (www.scalasca.org)
  - Scalable Profile and Trace analysis
  - Latest version: 2.3.1
    - New: More platforms (Xeon Phi, K computer, ARM64, …), Score-P 2.X and 3.x support
- **Cube** (www.scalasca.org)
  - Profile browser
  - Latest version: 4.3.4
    - Soon: Client/server architecture, more analysis plugins, performance improvements

BSC Performance Tools (www.bsc/es/paraver)

Flexible trace visualization and analysis

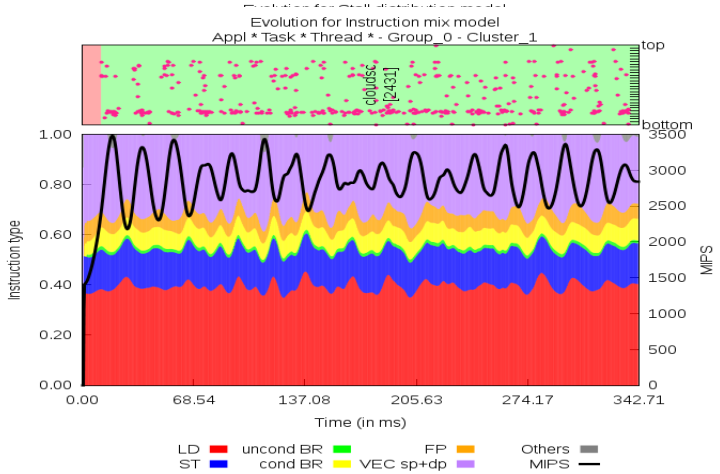
Adaptive burst mode tracing



BSC-ES – EC-EARTH

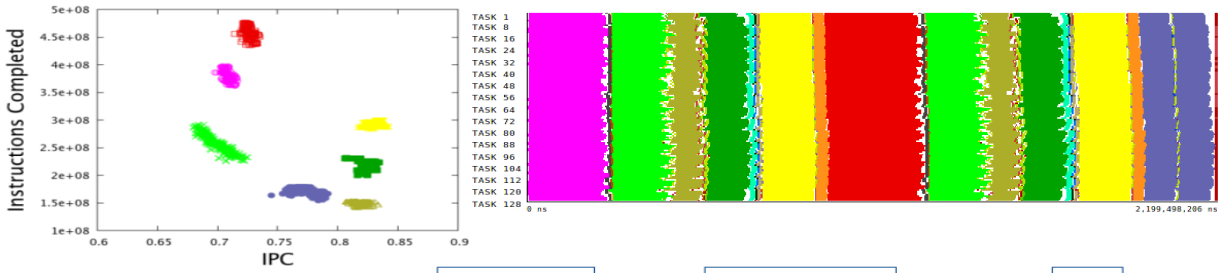
26.7MB trace
Eff: 0.43; LB: 0.52; Comm:0.81

Instantaneous metrics for ALL hardware counters at "no" cost

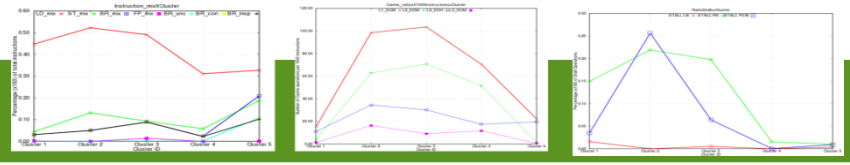


BSC-ES – EC-EARTH

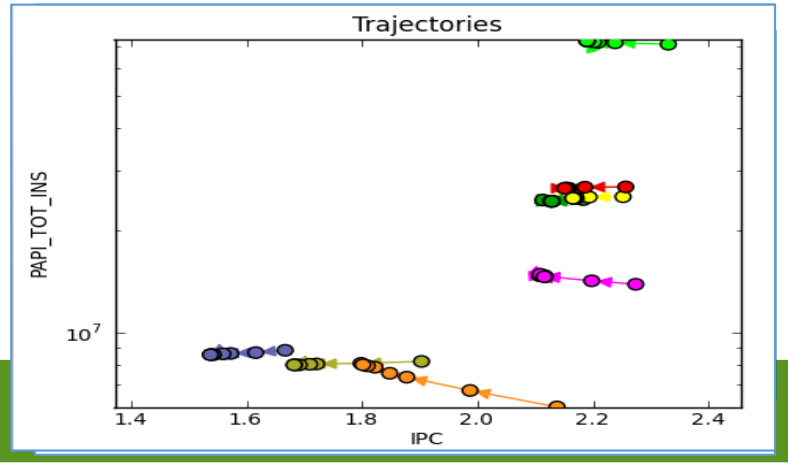
Advanced clustering algorithms



Instruction mix | Memory hierarchy | Stalls



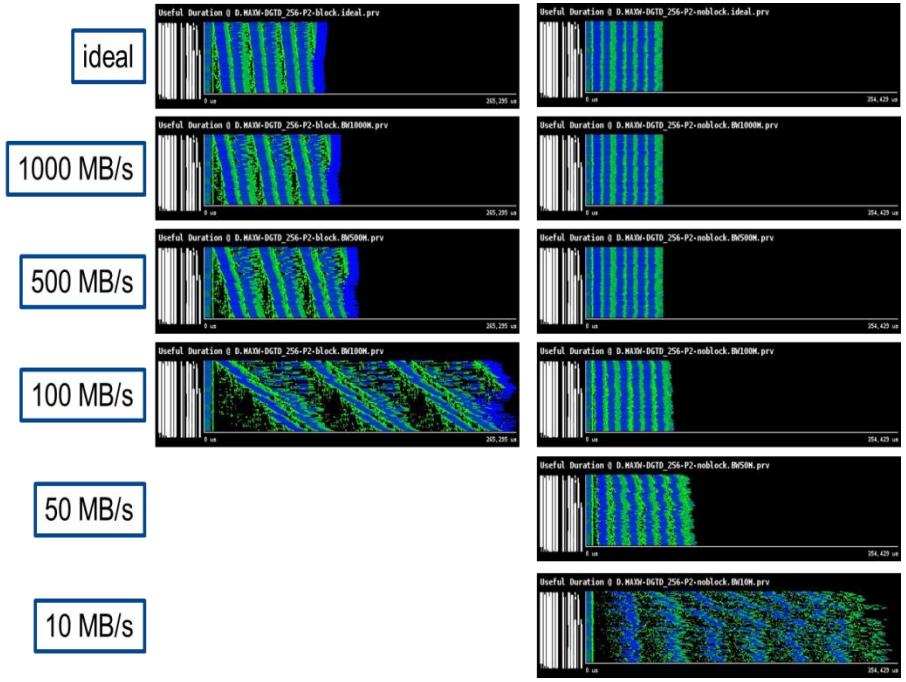
Tracking performance evolution



AMG2013

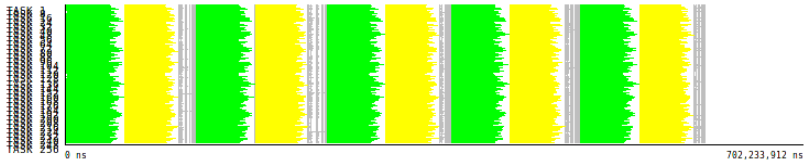
BSC Performance Tools (www.bsc/es/paraver)

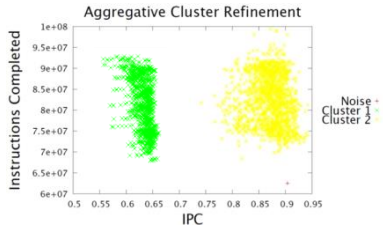
What if



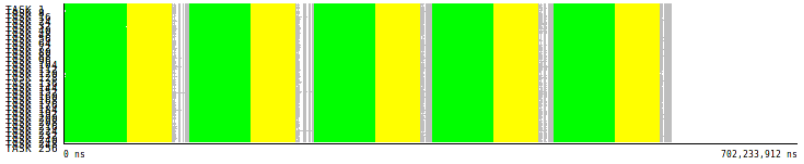
What if …

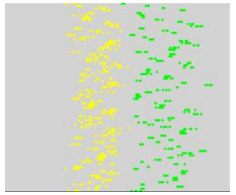
… we increase the IPC of Cluster1?





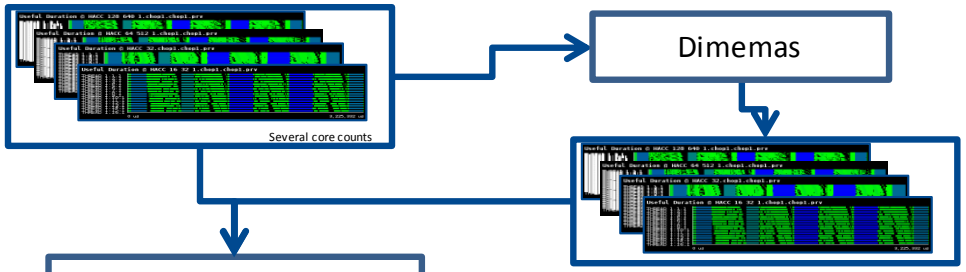
… we balance Clusters 1 & 2?



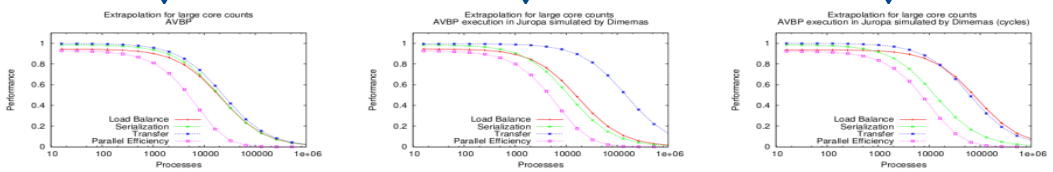


BSC Performance Tools (www.bsc/es/paraver)

Models and Projection

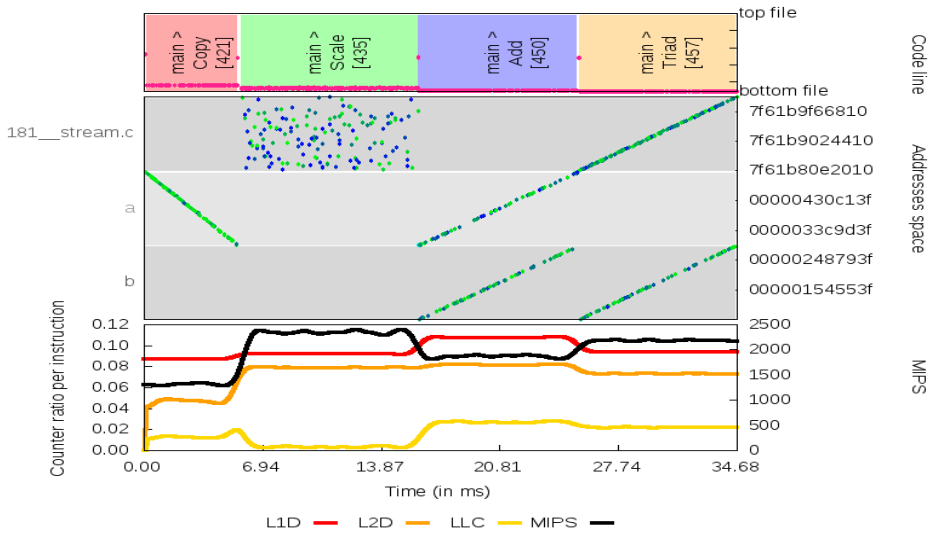


$$\eta_{\parallel} = LB * Ser * Trf$$

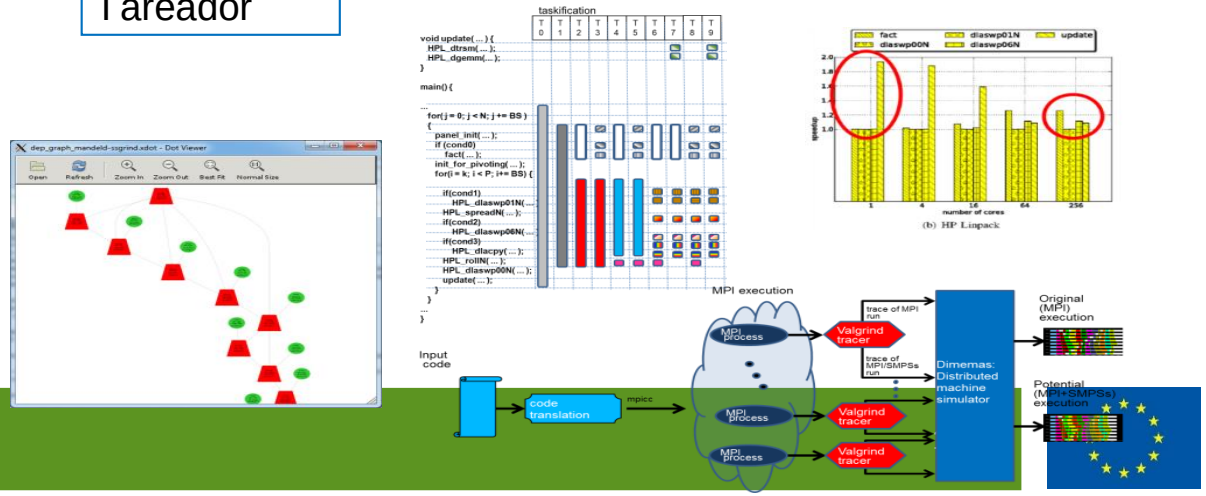


Intel–BSC Exascale Lab

Data access patterns



Tareador



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

Summary …

Apply : www.pop-coe.eu