



Performance Optimization and Productivity

EU H2020 Center of Excellence (CoE)



1 October 2015 – 31 March 2018 (30 months)



- **A Center of Excellence**
 - On **Performance Optimization and Productivity**
 - Promoting **best practices in performance analysis and 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



? 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

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

Request Service Form

Performance Optimisation and Productivity
A Centre of Excellence in Computing Applications

Request Service Form

Contact Details

Applicant's Name *

Institution *

e-mail *

Code

Name of the code *

Scientific/technical area and class of problems it solves *

Contribution *

Core developer Module developer User



Target customers



- **Code developers**

- Assessment of detailed actual behavior
- Suggestion of more productive directions to refactor code

- **Users**

- Assessment of achieved performance on specific production conditions
- Possible improvements modifying environment setup
- Evidences to interact with code provider

- **Infrastructure operators**

- Assessment of achieved performance in production conditions
- Possible improvements modifying environment setup
- Information for allocation processes
- Training of support staff

- **Vendors**

- Benchmarking
- Customer support
- System dimensioning/design



Activities (May 2017)



• Services

- Completed/reporting: 59
- Reporting: 7
- Codes being analyzed: 16
- Waiting user / New: 18

• Type

- Mostly Audits
 - 5 -15 pages
 - Level of detail

OpenMP performance assessment report

Document Information:
 Reference Number: POP_AIR_15
 Author: Neil Gibson (BIC)
 Contributor(s): Steve Latham (BIC)
 Date: March 13th, 2014

Serial Performance

- Evolution of IPC when scaling from 16 to 256 cores
- Tending to lower IPC for higher scales
- In addition, higher dispersion

Application Structure and Focus of Analysis

- Initial Audit: Parallell efficiency drops for more than 200 cores
- Analysis for 16 to 256 cores
- Truncated to the first 50 iterations, i.e. 2.55s out of 20,000s

4. Scalability

Figure 3 highlights the scalability of the code to 256 processors. On the left, the Speedup is plotted against the number of processors. The line indicates that the code is perfectly linear along scaling around 16 processors. On the right, the line indicates that the code is not perfectly linear along scaling around 16 processors. The line indicates that the code is not perfectly linear along scaling around 16 processors. The line indicates that the code is not perfectly linear along scaling around 16 processors.

5. Efficiency

Table 1 and Table 2 show results for fundamental factors and efficiencies from the FGA of the application using 16 to 256 MPI processes. Values are in percentage with higher values being better.

The observed great efficiency of the application decreases steadily from 11.16% at 16 processes to 0.44% at 256 processes, with an inflection point near 128 processes. The decreasing great efficiency is mainly caused by a decreasing load balance and decreasing communication efficiency. As an increasing amount of time (accounted over all processes) is spent in computation for higher process counts, the communication efficiency, however, is very constant and fixed in a high range. Load balance is discussed in more detail in Section 6. The decreasing communication efficiency is also observed by a decreasing number of iterations performed per step (IPC), which decreases to 0.44% at 256 processes.

Table 1: Other efficiencies

	16	64	128	256
Parallel Efficiency	0.9741	0.8103	0.6913	0.6013
Load Balance	0.9871	0.7148	0.5393	0.4413
Communication Efficiency	0.9988	0.9988	0.9988	0.9988
Computation Efficiency	1.13	0.614	0.4513	0.4413
Global efficiency	0.9947	0.7403	0.5393	0.4413

Table 2: Other efficiencies

	16	64	128	256
IPC Scaling Efficiency	1.0000	0.9988	0.9988	0.9988
Iteration Scaling Efficiency	1.0000	0.9713	1.1308	1.1308

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

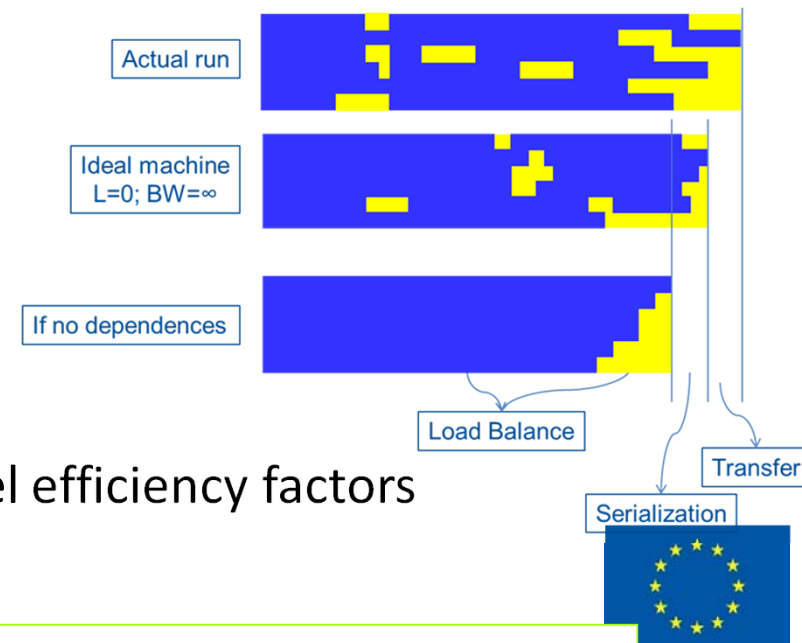
- **Load balance** (LB)
- **Communication**
 - **Serialization** (or Micro load balance)
 - **Transfer**

- Factors describing serial behavior

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

CommEff

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



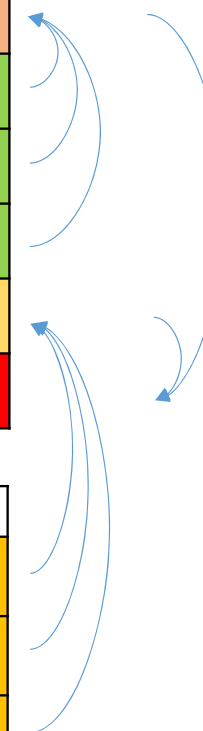
M. Casas et al, "Automatic analysis of speedup of MPI applications". ICS 2008.

Efficiencies

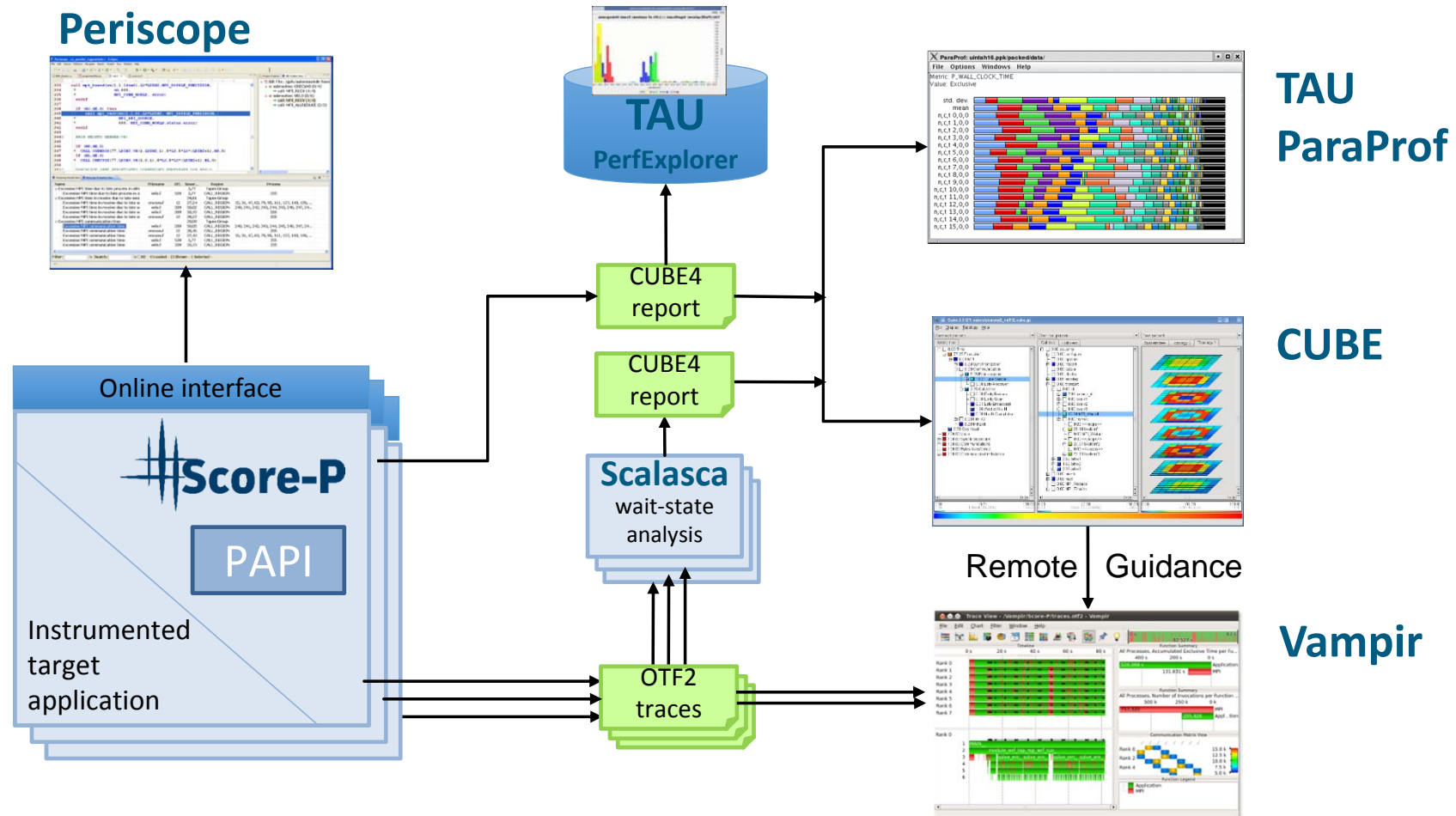


	2	4	8	16
Parallel Efficiency	0.9834	0.9436	0.8980	0.8478
Load Balance	0.9871	0.9687	0.9099	0.9177
Serialization efficiency	0.9975	0.9770	0.9938	0.9395
Transfer Efficiency	0.9988	0.9970	0.9931	0.9833
Computation Efficiency	1.000	0.9590	0.8680	0.6953
Global efficiency	0.9834	0.9049	0.7795	0.5894

	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



Scalasca (www.scalasca.org)

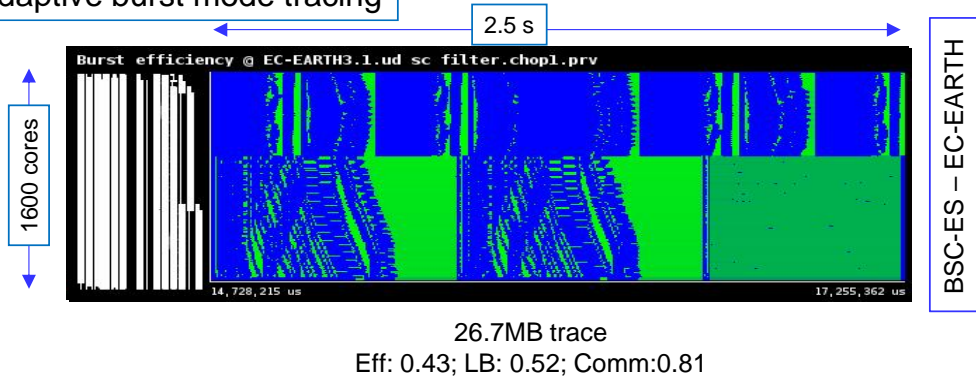


BSC Performance Tools (www.bsc.es/paraver)

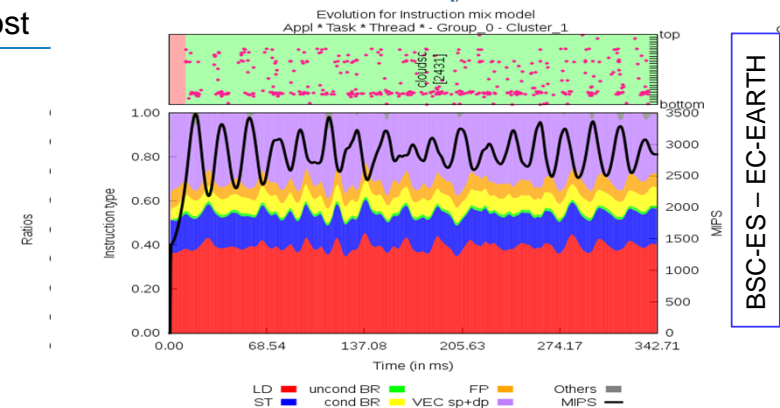


Flexible trace visualization and analysis

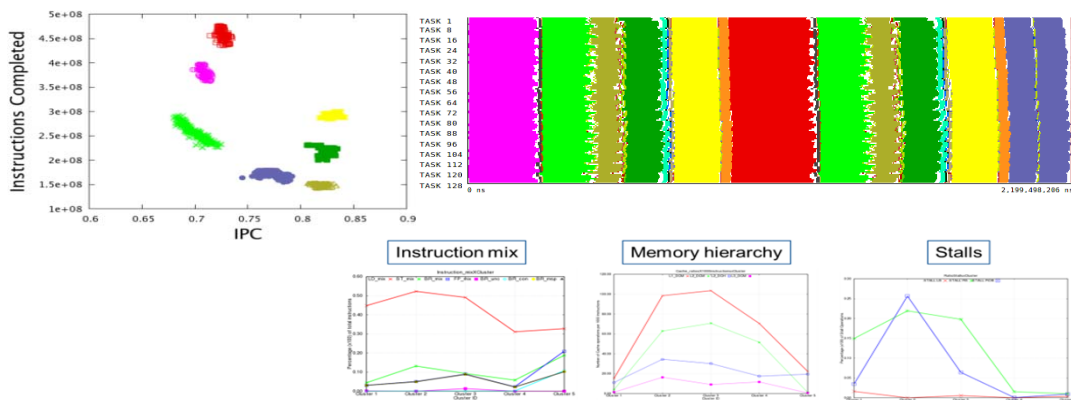
Adaptive burst mode tracing



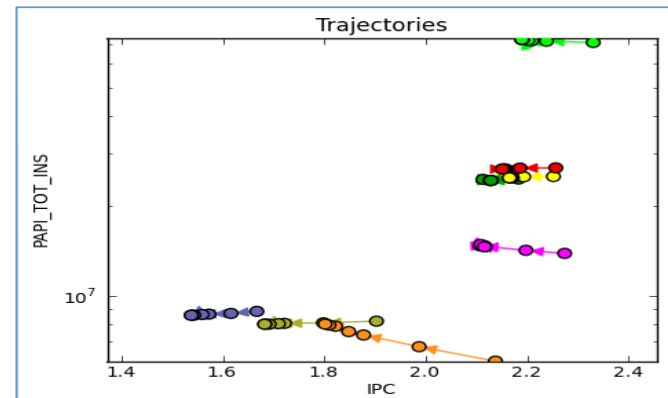
Instantaneous metrics for ALL hardware counters at "no" cost



Advanced clustering algorithms



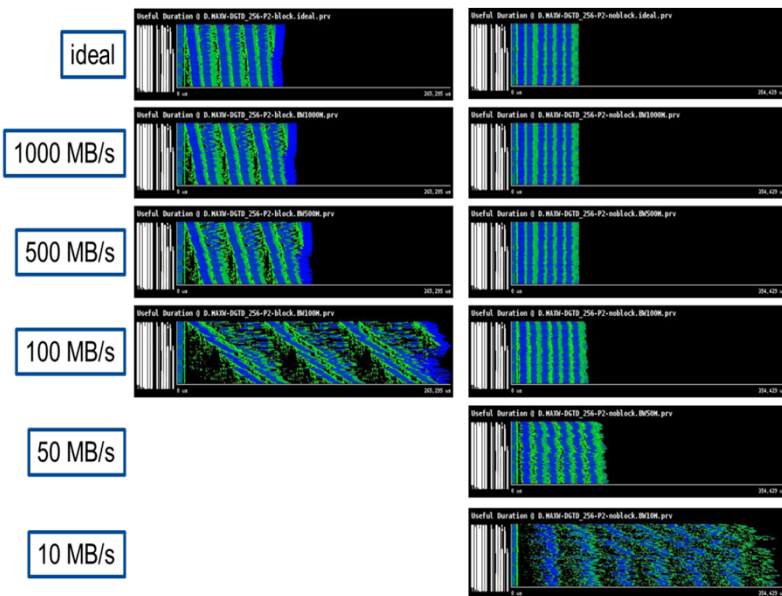
Tracking performance evolution



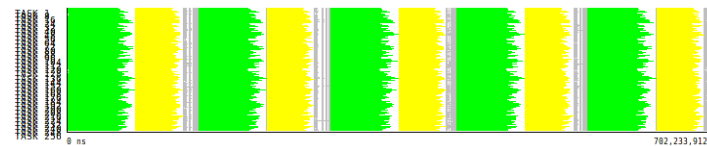
BSC Performance Tools (www.bsc.es/paraver)



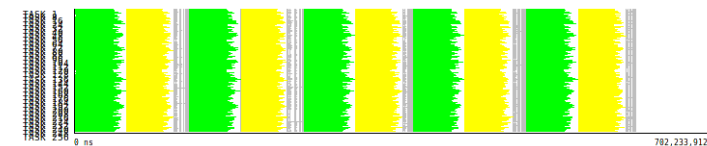
What if



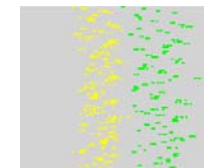
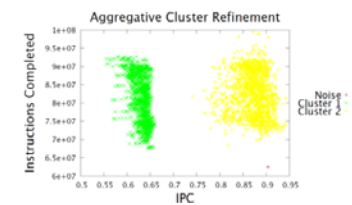
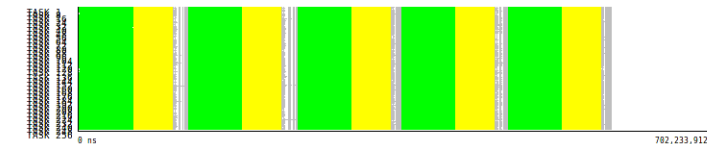
What if ...



... we increase the IPC of Cluster1?



... we balance Clusters 1 & 2?



Other activities



- **Promotion and dissemination**

- Market and community development
- Dissemination material and events

- **Customer advocacy**

- Gather customers feedback, ensure satisfaction, steer activities

- **Sustainability**

- Explore business models

- **Training**

- Best practices on the use of the tools and programming models
 - Cooperation with other CoEs (EoCoE)
 - Lot of interest ... customers want to learn how to do it themselves





Performance Optimisation and Productivity

A Centre of Excellence in Computing Applications

Contact:

<https://www.pop-coe.eu>

<mailto:pop@bsc.es>



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 676553.