



CONTRACT NUMBER RI-222919

DEISA 2
**DISTRIBUTED EUROPEAN INFRASTRUCTURE FOR
SUPERCOMPUTING APPLICATIONS**

European Community Seventh Framework Programme
RESEARCH INFRASTRUCTURES
Integrated Infrastructure Initiative

Initial Report on Enhancing Scalability

Deliverable ID: DEISA2-D9.1
Due date: October 31st, 2008
Author: Mariano Vázquez, BSC

Project start date: May 1st, 2008
Duration: 3 years

Project co-funded by the European Commission within the Seventh Framework Programme (2007-2011)		
Dissemination Level		
PU	Public	X
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Table of Contents

	Table of Contents	1
	List of Figures	1
	List of Tables.....	1
1	Introduction	2
1.1	Executive Summary	2
1.2	Relation with other Work Packages	2
1.3	References and Applicable Documents.....	2
1.4	Document Amendment Procedure	2
1.5	List of Acronyms and Abbreviations	3
2	Enhancing scalability effort	4
2.1	Application selection process.....	6
3	Probing Applications.....	9
4	Plan for the Next Six Months.....	12

List of Figures

Figure 1 – Scalability figures for Implicit and Explicit CFD solvers.	9
Figure 2 – Performance Analysis	10
Figure 3 – Communication Analysis.....	10
Figure 4 – Instructions-per-cycle Comparison for Different Codes	11
Figure 5 – Load Balance Monitoring for OpenMP	11

List of Tables

Table 1 – Comparisons between JS21 and QS2x (Cell B/E) Blades (from [7])	6
Table 2 – Development Effort Comparison between JS21 and QS2x (Cell B/E) Blades (from [7]).....	6

1 Introduction

1.1 Executive Summary

The role of Work Package 9 in DEISA2 (Distributed European Infrastructure for Supercomputing Applications) is to enhance the scalability of a set of selected applications, which must be adapted to efficiently use this infrastructure. As written in the DoW [1], WP9's objective is "to support WP5 by enhancing the scalability of the scientific applications". While WP5 mainly adapts a group of scientific applications to run in supercomputers, WP9 provides additional effort to reach the highest standards to a carefully selected subset of applications.

As described in the DoW, the keywords of the process of enhancing scalability are analysis, design, scaling and verification. The present deliverable, as it corresponds to the first 6-month period, deals mainly with the first one: analysis. Section 2 describes how the analysis stage should be carried on and what the tools to do it are. Section 3 gives an example of the analysis performed on a Computational Mechanics code. The last section sets the future tasks to do.

1.2 Relation with other Work Packages

WP9 is linked to two other WPs of DEISA2: WP7 "Extreme Computing Projects and Benchmark Suite" and WP5 "Applications Enabling".

WP7 manages DECI, the DEISA Extreme Computing Initiative, which is the main source of applications that are the target this work package. The relation is through WP7 Task 1, which is in charge of allocation of resources, account creation for the users, contact with the PI and users, etc.

WP5 is in charge of enabling scientific applications to the architectures the DEISA partners provide. WP5 compiles, ports, tunes and makes the applications run, complementing all this with a first level of optimization and basic enhancement of applications. WP9 must select a core sub-group of applications that will run for a second, higher level of performance enhancement according to the analysis elements WP5 provides.

1.3 References and Applicable Documents

- [1] DEISA2 Description of Work (Annex I of the Grant Agreement)
- [2] DEISA Site: <http://www.deisa.eu>
- [3] Deliverable DEISA2-D1.1: Initial Report on Management
- [4] Deliverable DEISA2-D5.1: Initial Report on Applications Enabling
- [5] PRACE Site: <http://www.prace-project.eu>
- [6] DECI Site: <http://www.deisa.eu/deisa1/grid/initiative.php>
- [7] Alya Site: http://www.bsc.es/plantillaA.php?cat_id=552
- [8] High-Performance Seismic Acoustic Imaging by Reverse-Time Migration on the Cell/B.E. Architecture. M. Araya-Polo, F. Rubio, R. de la Cruz, M. Hanzich and J.M. Cela. Presented at the ISCA 2008, 35th International Symposium on Computer Architecture June 21 -- 25, 2008. Beijing, China
- [9] PARAVÉR Site: http://www.bsc.es/plantillaA.php?cat_id=485

1.4 Document Amendment Procedure

This document is prepared according to the guidelines defined by the management of DEISA2. These rules can be found in section 2.7 of the deliverable DEISA2-D1.1 [3].

1.5 List of Acronyms and Abbreviations

DECI	DEISA Extreme Computing Initiative
DEISA	Distributed European Infrastructure for Supercomputing Applications
DoW	Description of Work (Annex I of the Grant Agreement)
ER	Enabling Report
OpenMP	Method for shared memory access on Multi-Processor environments
PRACE	Partnership for Advanced Computing in Europe
VC	Virtual Communities
WP	Work Package

2 Enhancing scalability effort

As said above, this deliverable mainly deals with the analysis stage. In the analysis stage, the selected codes should pass a first level of tests to measure their current performance and to give a hint of the amount of work necessary to improve it. In order to be selected to pass to the next stage, complementary non-technical information (such as the size of its community of users or the real need for running Grand-Challenge cases) must be provided. Once the codes are selected to pass to the enhancing scalability stage, the WP partners will decide the best strategy to improve its parallel behaviour. This should be done with direct collaboration of the applications own developers and through heavier analyses. Then, the enhancing tasks follow, allowing the selected applications to achieve the desired degrees of scaling. Finally, a verification stage will follow, running different data sets proposed by the users and developers of the targeted applications.

So far, the WP9 effort has been targeted to discuss the analysis stage. In order to perform the analysis in the most efficient way, some sets of metrics that will be used to measure the performance of codes in a standard way should be defined. The choice of the metrics is very important because they must give an objective measure of the potential of an application to pass to the next stages, where the real enhancement work lies. All the expertise of the partners of DEISA-2 in applications enabling and porting and in the use of performance analysis tools guarantees a good choice of the metrics set.

The metrics set must allow comparisons among the different codes, as they run in those architectures each DEISA partner provides. The enhancing scalability tasks of WP9 will be done guided by the performance analysis carried out in the selected codes, chosen from VC codes and 2009 DECI [6] applications. In order to be prepared to act on the codes that are to be ported, enabled and run in production (tasks done in WP5 and WP7), different possibilities of metrics and how to measure them are being evaluated. It is worth to repeat that this is not a completely sequential process, because these rules cannot be totally fixed a priori for they could suffer some minor changes once the measuring techniques are implemented and the target applications analyzed.

In order to have a preview of the kind of enhancing work that is expected, every application (coming from DECI, VC...) is categorized taking into account how the available resources are utilized. These categories are set and described in the WP5 DEISA2-D5.1 [4] (categories A to I). The proposed categories consider issues such as the parallelization paradigm programmed in the application, libraries links, demand for advanced architectures, data handling, DEISA infrastructure use, IO intensive use, etc.

WP5/7 enabling and porting are typically performed with the aid of data sets provided by the applications users. The set of preliminary runs needed to port the application, together with the very enabling effort and the interaction with the application user and/or developer, should lead to make a selection of the applications that will pass to WP9. The proposed performance metrics will quantize the codes' behaviour.

These metrics must accomplish some requirements. Firstly, they must provide sufficient information about the performance for the different platforms, but without increasing the burden of enabling a code. Secondly, they should be obtained using performance analysis tools that are known to each of the DEISA partners, with no imposition on which tool must be used. Assuming that each centre could have their own preferred performance analysis tools, the metrics must be, although informative, as general as possible and easily obtained from a wide range of performance tools. Examples of these metrics are: flops, total number of

instructions or IPC, I/O features, network usage, communications, cache misses, CPU and memory usage, hard and soft scalability, etc. Examples of performance tools that DEISA sites usually use are: HPM Count, MPIP, Paraver or Vampir. The next chapter gives an example on how this metrics could be obtained and reported.

The applications are not typically ported to all the architectures available, but only to those that are requested by the applicant. Porting the applications to the targeted architectures will provide the information needed to decide to go further. In WP9 are represented the most common supercomputing architectures. Examples of these architectures are IBM Blue Gene, JS21 “Marenostrum”, Cray MPP XT4 “Rainier” or Cray vector system “Black Widow”. Additionally, an advanced architecture based on Cell B/E will be available in the future, under a strong collaboration with PRACE [5] project. Each of them has its own particular features, rendering the enhancement of ported applications a totally different task.

For a particular application and considering only enhancement effort, Marenostrum JS21, Rainier Cray or Jülich BG do not differ that much, except for the fact that BG has lower memory available per core than the others, a fact that puts a constraint when data sets are divided in large chunks of memory.

On the other hand, the Black Widow system (now called X2) is a standard vector system, very like the NEC SX series. As with all vector systems, it is characterised by very high memory bandwidth and a very high ratio of achieved/peak performance for codes that vectorise well. Achieving high vectorisation requires the compiler to recognise vectorisable code. For some codes this is all done automatically (e.g. for very long loops with no function calls). For some codes, the nature of the algorithm makes vectorisation very difficult (e.g. most molecular dynamics codes which have complicated branches). For other codes, vectorisation may be possible in principle but might require a rewrite to expose this to the compiler. Although vector compilers are very sophisticated they always benefit from expert (human) help by either code restructuring or additional compiler directives.

Special effort should be made for applications that fall in Category F “Porting to a new machine: advanced architectures”, where strong cooperation with EC Project PRACE (“Partnership for Advanced Computing in Europe”) is extremely important. A paradigmatic example is Cell B/E processors-based architectures, worth to be shortly described here in more detail to show the kind of tasks we face (see Fig. 1). To port a typical Computational Mechanics application to Cell B/E should not represent a large effort. Indeed, programming models are being developed for them, like Cell Super Scalar, MicroMPI or OpenMP for Cell. Also Source-to-Source compilers are available. However, porting does not mean to plainly get all the available power from a computational resource: it is part of the story. On the one hand, the aforementioned programming models and compilers are still under development and of limited use. On the other hand, Cell PPU are special processing units, having two remarkable features: they are vectorial and they have a few Kbytes of local memory. This means that, for a given application, its computationally intensive parts must be firstly identified. Then, they should be analyzed to see if vectorization is possible and if the stringent memory requirements (which include data & code) of PPUs could be fulfilled. These two constraints could be simultaneously verified in several ways. To consider all the different possibilities, the help of the application PI is decisive, because this analysis requires a deep knowledge of the algorithms implemented. The range of options covers ideas like redistribution of loops and rearrangement of data or code partition in small chunks to pass from the SPU to the PPUs, whether to be solved in parallel or streaming from on group of PPUs to another. Next, the resulting parts must be compiled separately for the SPU and the PPUs and linked afterwards. Finally, data sets provided by the PI will run, analyzed and fine-tuned. Considering that to enhance scalability means to squeeze as much power as possible out of a given

supercomputer, once a Cell-based facility shows up, the supplementary effort described above must be dedicated to use it efficiently. It is worth to mention that although the effort of Category F could be very important, each of the other categories has features of their own, like those of intensive I/O use or new algorithms and libraries that make them special on their own.

Platform	Average power [W]	Execution time [s]	Arithmetic Throughput [GFlops]	Energy Efficiency [GFlops/W]
JS21	267	41.0	7.3	0.03
QS20	315	5.5	54.1	0.17
QS21	370	5.0	58.3	0.16

Table 1 – Comparisons between JS21 and QS2x (Cell B/E) Blades (from [8])

Step	Target	Project size [lines of code]	Effort [man –month]	Speedup
1	JS21	1000	1.25	1.3
2	JS21	800	1.00	1.3
3	QS2x	1500	2.00	13.9
4	QS2x	500	1.25	18.9

Table 2 – Development Effort Comparison between JS21 and QS2x (Cell B/E) Blades (from [8])

In order to make a fair decision on which are the applications that will pass through the Enhancing Scalability Process of WP, not just the metrics issue must be taken into account. There is a wide range of points to be considered. For instance:

- Some simulation fields have more social impact than others, making attractive the supplementary effort of enhancement.
- For a given application of a given field, the following questions arise: do we need to solve very large problems, of larger scale than those that the pure enabling process would allow? Is the current status of the application enough or it should be enhanced?
- The availability and commitment of the application developers to be involved in the enhancement process must be also evaluated: without a strong commitment there is no point in dedicating our limited resources to this application enhancement.
- When for one field there is more than one application to be considered, factors like the size of the users community should help to decide.
- PRACE and DEISA actions in enhancement must be coordinated to avoid repeating work. Any previous experience is of great value and must be sought.

2.1 Application selection process

In summary, to select an application to pass to WP9 Enhancing Scalability, a selection process should be completed. It is worth to remark two important issues. On the one hand, every application that reaches this point has been already ported and it is running in the DEISA architectures, because this task belongs to WP7 and WP5. On the other hand, WP9 has a limited amount of man-power, around one tenth of the man-power of WP5 as it is described in the DoW.

The application selection process should follow these steps:

1. The WP9 coordinator must collect the following:

- Enabling Reports (coming from WP5): They will provide a brief description of the project and the effort being done in the application.
- Basic technical information:
 - Application name, enabling centre, execution centre.
Why needed: to double check with the Enabling Reports
 - Developer information. Is the user a developer too?
Needed because if the user does not develop, then the developer should be contacted.
 - Compiler, compiler options, libraries linking information.
To see the most basic optimization performed.
 - Parallelization paradigm: MPI, by-threads, embarrassingly parallelization, hybrid, etc.
It gives a first panorama and where the problems could come from. It also gives the general idea of the kind of improvement effort.
 - Programming language and programming models (if needed)
To pinpoint potential experts among DEISA-2 partners
 - Size of the data set(s) run corresponding to the performance tests: expressed in the usual “units” for each kind of application (elements, cells, molecules, voxels, etc.)
To assess the size of the problem relative to other similar problems solved by the respective community.
- First level performance tests: This information is complementary to the Enabling Reports. It is related to performance issues and based on the set of metrics to be defined. This tests reports should include this basic information:
 - Solution total CPU time
 - Cycle, iteration or time-step CPU time
 - Hard scalability figures: the speed-up with an increasing number of cores, normalized by the total time of the simulation using the lowest amount of resources. The granularity must be also provided, i.e., the mean amount of the aforementioned “units” per core.
 - Soft scalability figures (if possible): the total times for an increasing number of cores keeping constant the number of “units” per core. This means to modify data sets to change their size, which is not always a straight process.
 - Memory usage: total and per-core. If a master-slaves strategy is used, the per-core-usage should be discriminated between both master and slaves.
This is important to set if the application and data set could be ported to another machine with less memory.
 - Input data I/O usage: size of input data, time to load it and its ratio to the total simulation time.
Is there an improvement window in I/O?
 - Output data I/O usage: size of the output data, time to dump it and its ratio to the total simulation time. Usage of supplementary libraries (NetCDF, HDF5, etc.)
Same comment as the previous point. The supplementary libraries information can give hints on known problems and be useful to pinpoint experts.
 - Communications per-cycle: ratio to the total simulation time
 - IPC, flops, network usage, cache misses: this should be included only when available.

To be defined. Depends on how difficult this information could be obtained for the targeted applications.

- Developers' availability: if the application developers agree to contribute to the enhancement process and under what terms.
- Potential for Enhancement: according to the DEISA-2 intervening partner and the applications developers, what are the weak points of the application that could be improved
- Application parallelization history: relation with previous projects, enabling and execution centres, architectures, performance tests already passed, etc.
- Supplementary application data: as described above, i.e., social impact, need for large runs, etc.

2. In the periodic coordination video-conferences, which WP5/7/9 held usually together, the collected Reports will be discussed in order to select which applications would require WP9 resources to be enhanced.

3. The selected applications will be distributed among the WP9 partners. Once the application is assigned to a partner or group of partners, it will pass a deeper performance analysis process to assess performance and thoroughly study different enhancing strategies. Interaction with the developer is now decisive.

3 Probing Applications

This section describes examples on how to probe a generic Computational Mechanics application in the terms described above. The examples come from the work done on BSC's Alya System [7] (a comprehensive list of publications and reports related to the code can be found in its referenced website). It is a code for solving coupled large-scale problems designed with the highest standards on parallel efficiency. It is parallelized following two different although complementary schemes: using MPI with automatic mesh partition by METIS and "by-threads" using OpenMP for elementary loops. It is capable of running in a hybrid way, getting the best from multicore clusters. It has performed benchmark tests up to 5000 thousand processors.

The first and simplest way to measure the performance of an application is through its scalability (Figure 1).

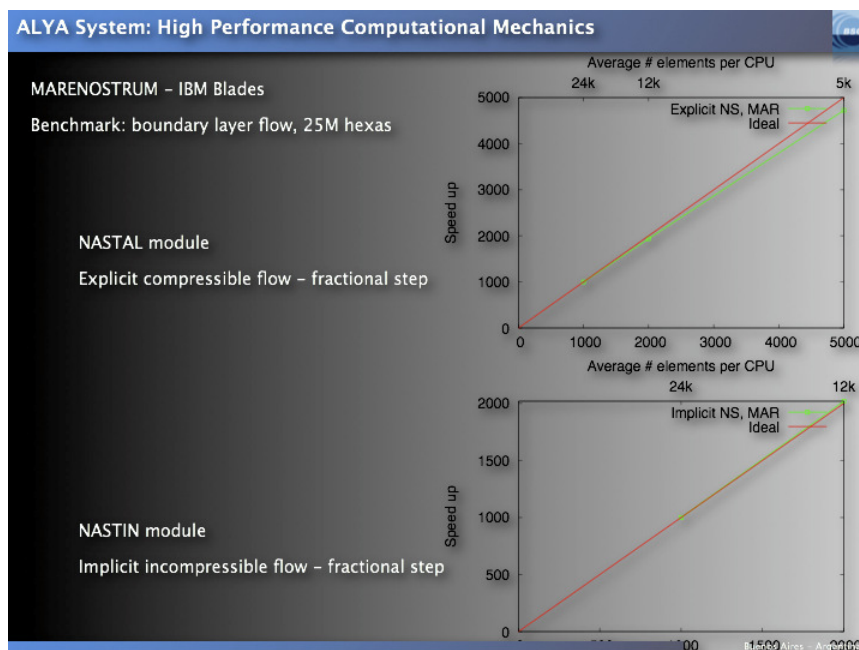


Figure 1 – Scalability figures for Implicit and Explicit CFD solvers

Scalability can be measured on data sets provided by the users. Fig.2 shows plots of *strong* scalability, measured when the same problem, of the same size, is solved in an increasing number of processors.

An example of the use of a performance tool is shown in the following Figures. Paraver [9] is a performance analysis tool developed in BSC's Computer Science Department. This is not intended as commercial brochure of Paraver, but an example on how such kind of tools can be used to obtain the desired metrics.

There are different degrees of probing a code using a tool like Paraver. The deepest degree is achieved when the code developers take active part by placing "counters" inserted in the source code. A code trace is obtained, filtered and analyzed using a graphical tool. In Figure 2, the vertical axis is the number of processors (128 for this small test) and the horizontal axis, the time. The "bluest" the graph, the better, because this means that work sharing is well balanced.

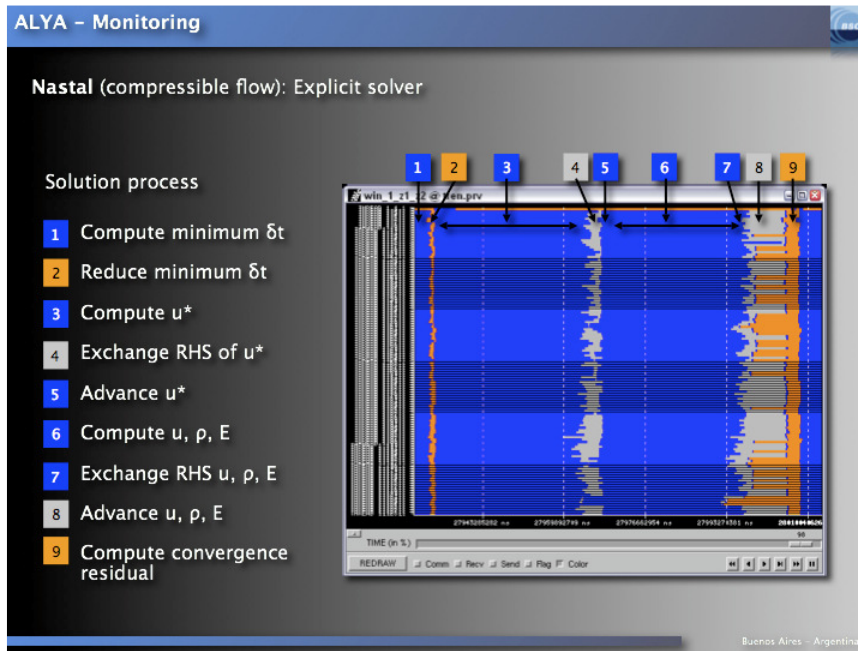


Figure 2 – Performance Analysis

Communication issues can also be studied, as shown in Figure 3. Here, the amount of communication per iteration / process is graphically displayed, including the communication pattern. This is extremely valuable to improve the communication efficiency of the code.

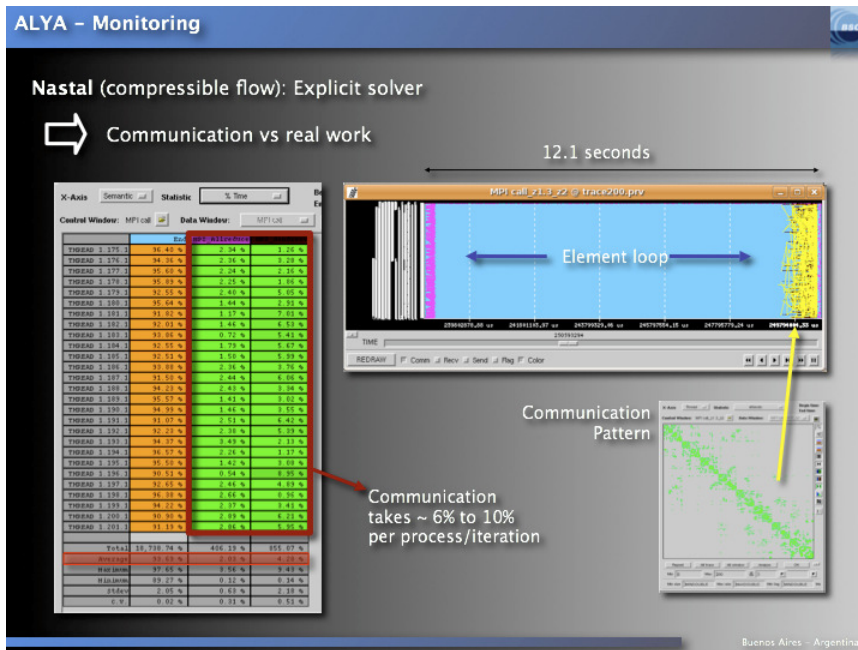


Figure 3 – Communication Analysis

Performance tools can also measure Instructions-per-cycle, as shown in Figure 4. This figure shows an example on how valuable this information is to compare completely different codes. However, it must be remarked that no comparison is fair if the code developers do not provide additional information on how it works, what algorithms are programmed, which libraries, etc.

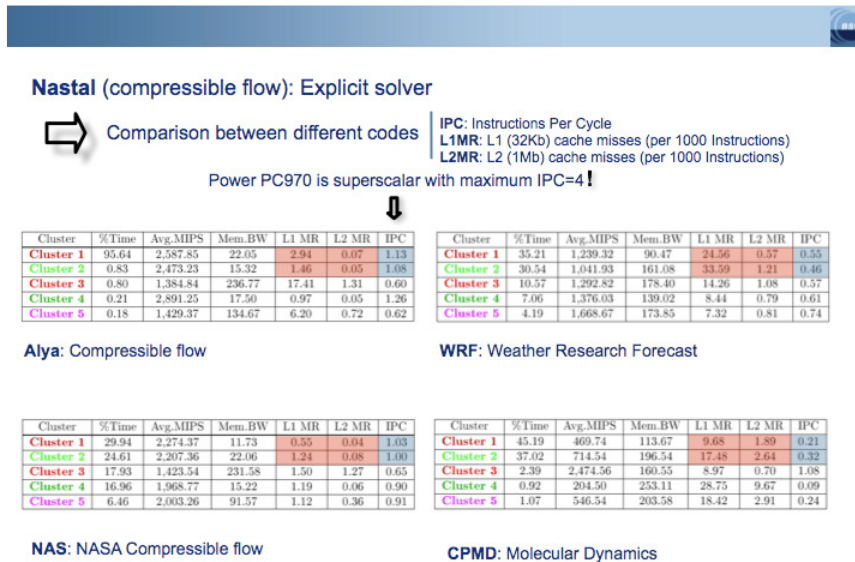


Figure 4 – Instructions-per-cycle Comparison for Different Codes

A last remark is the ability of performance analysis tools to work in shared memory machines as well as distributed memory ones. In Figure 5, three different work distribution strategies are compared for OpenMP threads on elementary loops. By changing the heading pragmas at the loops, the programmer can change the way work is distributed.

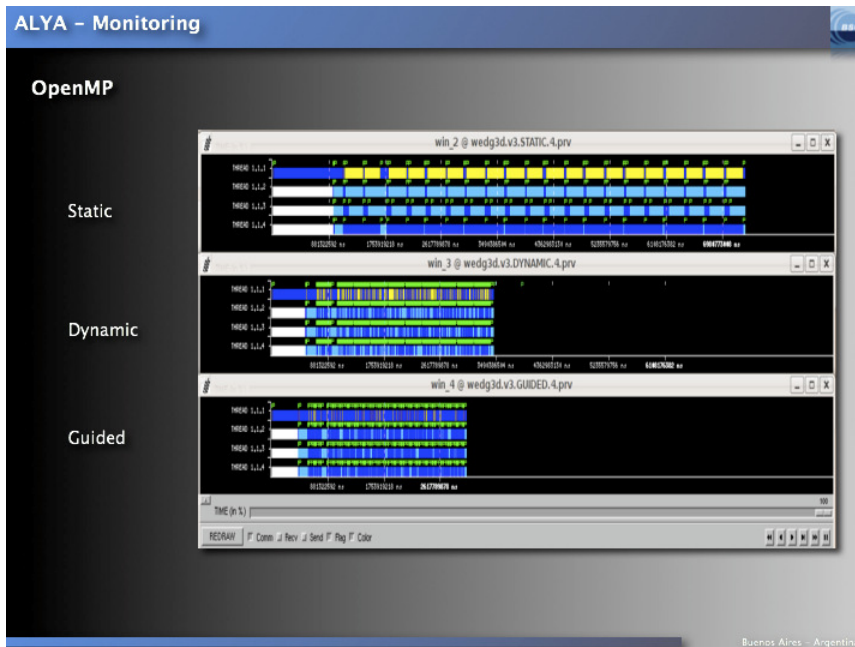


Figure 5 – Load Balance Monitoring for OpenMP

4 Plan for the Next Six Months

At the time this document is delivered to the EC, the new 2009 DECI proposals should be already evaluated and accepted/rejected. In coordination with WP7, that manages DECI and with WP5, that will be focused in the effort enabling of the new applications WP9 will be related to code's performance enhancement.

In the next 6 months, the following tasks are foreseen:

- The basic metrics and the measuring tools will be set after an evaluation process.
- The results coming from the tests should be presented in a simple template that will be designed, analogous to the Enabling Reports of WP5. This template will be also useful to present the results through deliverables.
- By the end of this period, some applications will be proposed as candidates for performance enhancement, according to what is described in section 2. Each of them will be diagnosed to have a preliminary idea on how much work is needed to improve the performance. Experts on their algorithms, parallelization paradigms or physical process simulated will be picked from the DEISA partners to provide supplementary information that helps to assess the work to be done.
- A second, deeper, set of performance analysis metrics and tools will be discussed because the chosen applications should pass a second level round of tests. The second level tests will be devised to probe the applications at a deeper degree, which means that they will take more effort to perform them than the first level.