



CrossGrid User Manual Guide

Performance Prediction Component

Task 2.3.2 - PPC

Document Filename:	CG-2.3.2-PPC-UserManual-v1.0-USC.pdf
Workpackage:	Task 2.3.2 - PPC
Partner(s):	USC
Lead Partner:	USC
Config ID:	CG-2.3.2-PPC-UserManual
Document classification:	PUBLIC

Abstract: This is the user manual for the CrossGrid Performance Prediction Component, version 1.3.0, task WP 2.3.2.



Delivery Slip

	Name	Partner	Date	Signature
From	T.F. Pena & J.C. Cabaleiro	USC	Nov 2004	
Verified By				
Approved By				

Document Log

Version	Date	Summary of changes	Author
1-0-DRAFT	11/12/2004	First draft version	Pena/Cabaleiro

Contents

CopyrightNotice	4
1 Introduction	5
1.1 Product Usage	5
1.2 Running the Product	5
1.3 Basic Operation	7
2 Interface Reference Guide	8
2.1 Grid	8
2.2 Kernel	11
2.3 Display	13
2.4 Help	17
3 Troubleshooting Q and A	18
4 Contact Information and Credits	19
5 EDG License Agreement	20

Copyright Notice

Copyright (c) 2005 by *T. F. Pena, J. C. Cabaleiro, M. Boullón and F. F. Rivera*. All rights reserved.

Use of this product is subject to the terms and licenses stated in the EDG license agreement. Please refer to Chapter 5 for details.

JFreeChart and *JCommon* (C)opyright 2000-2004, *Object Refinery Limited and Contributors*.

JavaHelp is a registered trademark of *Sun Microsystems, Inc.* All rights reserved.

This research is partly funded by the European Commission IST-2001-32243 Project "CrossGrid".

1 Introduction

The PPC (Performance Prediction Component) is a tool to study the behavior of some selected MPI kernels in the Grid. It is based on a set of models that characterize the behavior of the kernels when they are executed in the Grid. These models were obtained from exhaustive executions on different Grid scenarios. From these measurements a methodology to obtain correlations between performance and monitoring information is applied to obtain easy to use functions to characterize the execution of the kernel. Even though the tool is limited to a set of specific kernels, it can be easily extended to others. Note that some of the kernels are general purpose. i.e. communications, that could be useful for any user.

The tool shows performance information based on predictions taken from the models. The way to show this information is interactive and easy to use. Also the tool can get monitoring information provided by JIMS to show the predictions under the present and real scenario, and then the user can interactively change these values to analyze their effects on the kernel.

Some of the main features of the tool are:

- PPC is an application dependent tool. It is specifically applied to selected kernels that were analyzed in terms of performance to obtain analytical models. Anyway, it can be also broadly used to study the performance of communication routines themselves.
- PPC is an interactive tool in the sense that the user can easily change the parameters that characterize the system or the problem, and then analyze their influence.
- PPC uses analytical models, and therefore the predictions are obtained fast. This feature is very important for the interactivity.
- PPC can be used to understand the behavior of the kernels under different Grid features, selected by the user. This is important in any system in which the features change dynamically, like Grids.
- PPC can be easily extended to any heterogeneous system and kernel.

One of the main contributions of PPC to the state of the art in Grid computing is the methodology to obtain the performance models, and to allow the user to understand the behavior of the kernels under hypothetical Grid features.

1.1 Product Usage

The use of the PPC tool is simple. After launching it, the user can select the kernel to be studied, and the Grid features. One of the possibilities is to read these features from a monitoring engine, in our case JIMS. Then the user can see the effect of these features on the performance of the kernel using different views for different aspects of the performance. PPC allows the user to change the features interactively. A help button can be used to obtain detailed information about the facilities of the tool.

1.2 Running the Product

The PPC has been developed to be used from the Migrating Desktop, but it can be also used as a standalone application. To run it as a standalone application, simply execute:

`$PPC_HOME/bin/PPC-standalone.sh`

where `$PPC_HOME` is the directory where PPC has been installed (default `/opt/cg`)

To run it from Migrating Desktop, select it from “Tools” menu or in the main menu bar.

1.2.1 Operating Requirements

If the tool is run from Migrating Desktop, the user needs access to the Roaming Access Server. More information in <http://ras.man.poznan.pl/crossgrid/>.

Local hardware requirements

No special local requirements are needed.

Local software requirements

Standalone application

1. General libraries

- (a) J2RE \geq 1.4.2
Java(TM) 2 Runtime Environment, Standard Edition v1.4.2 01,
<http://java.sun.com/j2se/1.4.2/download.html>

2. CrossGrid Libraries

- (a) ToolPlugin.jar
- (b) JIMSCient.jar

3. Other

- (a) cog-jglobus-1.1.jar
- (b) jfreechart.jar \geq 0.9.18
- (c) jcommon.jar \geq 0.9.3
- (d) axis.jar
- (e) jaxrpc.jar
- (f) saaj.jar
- (g) commons-discovery.jar
- (h) commons-logging.jar
- (i) log4j.jar
- (j) jh.jar

Detailed information can be found in the installation guide:

<https://savannah.fzk.de/distribution/crossgrid/crossgrid/wp2/wp2.4-perf/wp2.4.2-perfpred-PPC/docs/CG-2.3.2-PPC-InstallGuide-v1.0-USC.pdf>

From Migrating Desktop

1. Java Web Start or Web browser with enabled Java Plugin 1.4 or newer and the Java Plugin cache (JAR cache) disabled.

More information in <http://ras.man.poznan.pl/crossgrid/>.

Grid infrastructure requirements

In both ways of running the application, the only required Grid infrastructure that must be available is JIMS (CrossGrid Work Package 3.3), in order to obtain monitoring data from the sites. If JIMS is not available in a specific site, PPC cannot obtain monitoring information from it, but it can be used in most of its functionalities.

1.2.2 Step-by-step Setup

No setup steps are required.

1.3 Basic Operation

After launching the application, in the main window, it appears four menus:

- Grid: get/set the Grid parameters
- Kernel: select the kernel to be predicted
- Display: show prediction results
- Help

For example, for getting and showing grid related information, the following steps are needed:

- Get the information provided by JIMS (Grid menu → Monitoring → Get data)
- Selection of the JIMS server
- Show the grid information (Grid menu → Monitoring → Show data).
- Show the information of a specific node (Select a node number)
- The values showed in the graphs can be modified selecting them with the mouse (two clicks, one for selecting the graph and other for selecting the element, bar or point, to be modified) and using the slide bar or writing the value.

And to select the kernel, for example, Vertlq and to display results:

- Select the kernel to be predicted (second in this exercise, Vertlq).
 - Select the input meteorological data
 - Select the directory with the input data in the Grid Explorer Dialog of MD or in the local file system
- Display the results (Display menu).
 - Display the number of FLOPs (Floating Point Operations) for a specific processor (Display menu → Computations).
 - Display the predicted load balance (Display menu → Load balance).
 - Display the number of FLOPs per mesh point (Display menu → Specific displays → FLOPs per mesh point).

1.3.1 Known problems

Some functionality are not included yet, because of the development of the prediction models is not fully finished.

2 Interface Reference Guide

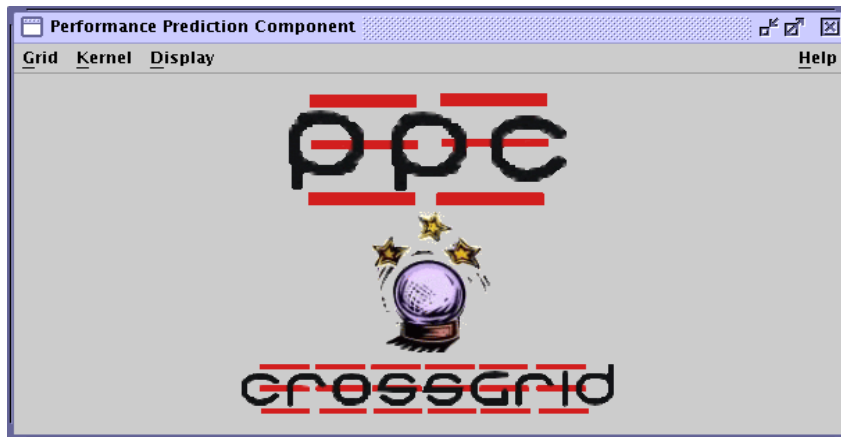


Figure 2.1: PPC main menu

Figure 2.1 shows the main interface of the PPC. In this window there are four main buttons called “Grid”, “Kernel”, “Display” and “Help”. The first one is used to select the source of information about the Grid, the second to select the kernel to be analyzed, and the third to show the results of the analysis. Finally, the “Help” button permits to read the PPC on-line documentation.

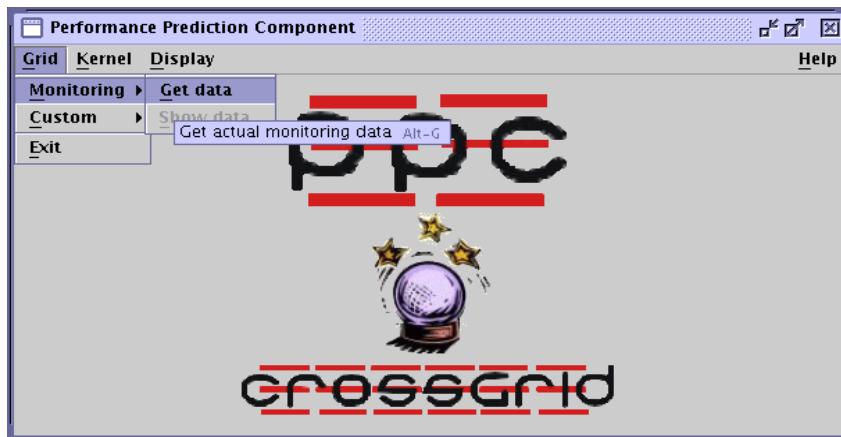


Figure 2.2: Grid menu

2.1 Grid

The “Grid” option of the main menu is used to select the source of information about the Grid, and it presents two possibilities, called “Monitoring” and “Custom” (see figure 2.2). The first one is used to get information about the current status of sites in the grid, using JIMS. The “Custom” option is used

to establish specific network configurations by the user himself. This is useful to do particular studies of some kernel.

2.1.1 Monitoring

The “Monitoring” option of the “Grid” menu have two suboptions, called “Get data” and “Show data”, (see figure 2.2), the first one is to get data from monitoring, and the other is to visualize them 2.3. This last one is disabled until monitoring data from JIMS are obtained.

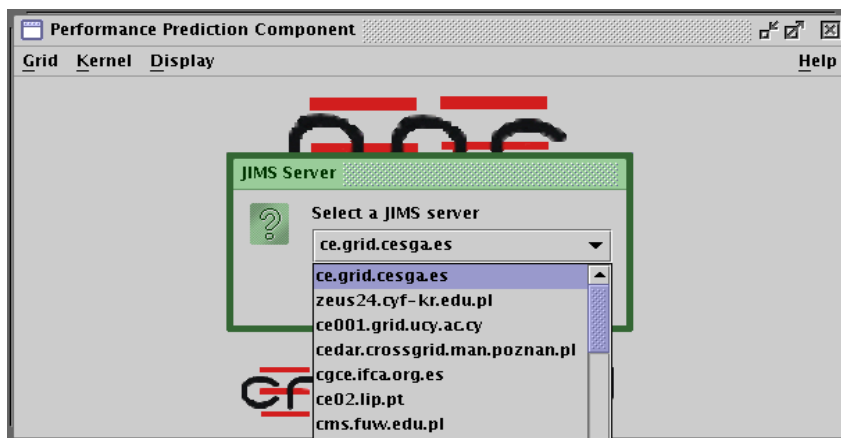


Figure 2.3: List of JIMS servers

Get data

If the “Get data” option is selected a dialog for selecting the site to be monitored appears (Figure 2.3). When “OK” is pressed after selecting one of the JIMS servers from the list, PPC contacts with the server in order to obtain information about the actual status of the grid site. If the server could be monitored without problem a confirmation message appears and the “Show data” menu is enabled.

Show data

Figure 2.4 shows the output for the “Show data” suboption. It shows static info about the grid nodes (model, IP address), the CPU load, and the latency and bandwidth between each pair of nodes in the cluster. If the user clicks on one of the bars of the circles of these charts, numerical information is displayed. The user will have the possibility of change these values, writing directly the new value on the text field or using the slider. The buttons “Restore” and “Restore all” recover the original value for one of the entries or for the entire graph, respectively.

2.1.2 Custom

With this option user can define a “virtual” cluster, specifying the number of nodes, CPU load of each node and the latency and bandwidth between each pair of nodes in the cluster.

When this option is selected, a menu to select the number of nodes in the cluster appears (see figure 2.5). After specifying a number of nodes, user should go to “Grid” → “Custom” → “Show data”, and modify the default values for CPU load, latency and bandwidth.

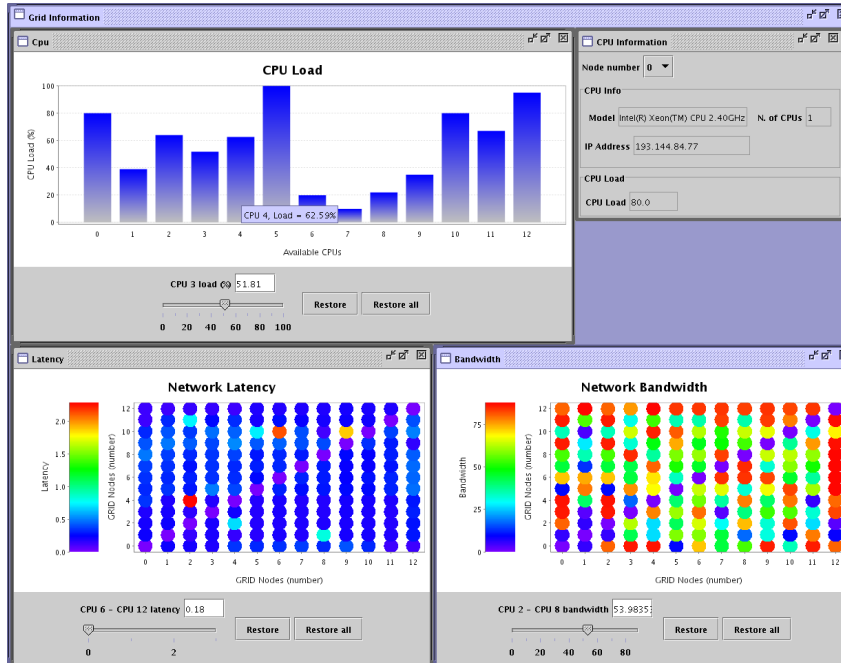


Figure 2.4: Performance information

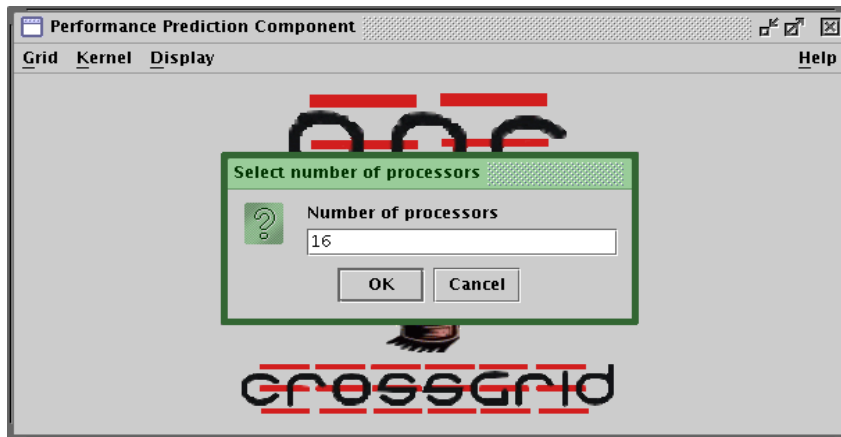


Figure 2.5: Number of nodes in a custom cluster

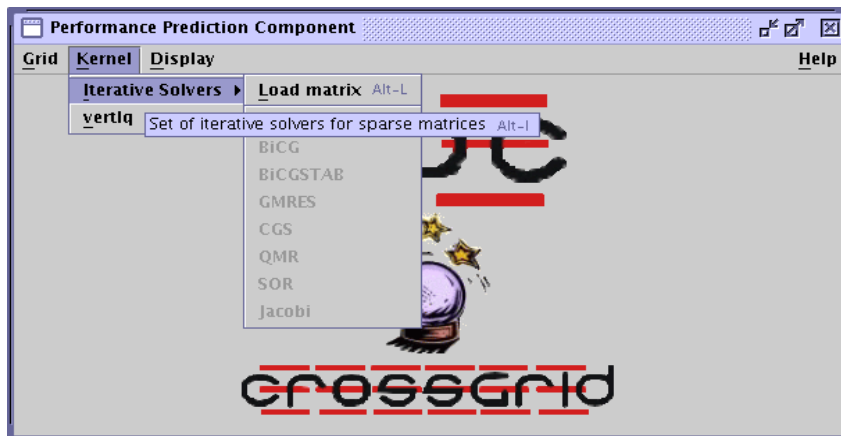


Figure 2.6: Kernel selection

2.2 Kernel

Figure 2.6 shows details about the second button of the main menu. This button is used to select the kernel to be considered by the user. Currently, the options are iterative solvers, and the VERTLQ routine that are interested for specific applications (air pollution, WP1.4). Nowadays we are including more kernels in this list.

2.2.1 Iterative Solvers

For iterative solvers a square sparse matrix must be first loaded from a file. The file must be accessible through the MD Grid Commander if PPC is launched from MD, or reside in local file system if standalone. The matrix must be stored in Column Compressed Storage (CCS) format [Saad96, BB⁺94]. The file must be in ASCII format with the next structure:

1. Number of columns (N) and number of non-zero entries (NNZ) (integers).
2. A N+1 elements vector with the position of the elements which start each column (integers).
3. A NNZ elements vector with the row indexes of the non-zero entries (integers).
4. Values of the non-zero entries, in a column wise fashion (optional).

An example file can be obtained:

- In the SE at CESGA (se.grid.cesga.es/flatfiles/SE00/cg/PPC).
- In <http://www.ac.usc.es/~crossgrid/PPCTutorialFiles/>
- File name: bcpwr10.ccsf

After selecting the file with the matrix, one of eight different iterative solvers can be chosen for simulation, as shown in figure 2.7. This includes most of the standard stationary and non stationary iterative methods [Saad96, BB⁺94]. Solvers come from the Paraiso (Parallel Iterative Solvers) library [BCG⁺02] (<http://www.ac.usc.es/Paraiso>)

For each iterative method different scaling and preconditioner can be selected (figure 2.8). Some preconditioners need additional parameters, which can be introduced with the “Parameters” button.

Some kernels (iterative solvers, matrix-vector multiplication...) have been designed for a processors mesh. The number of rows and columns of the processors mesh have to be introduced (figure 2.9).

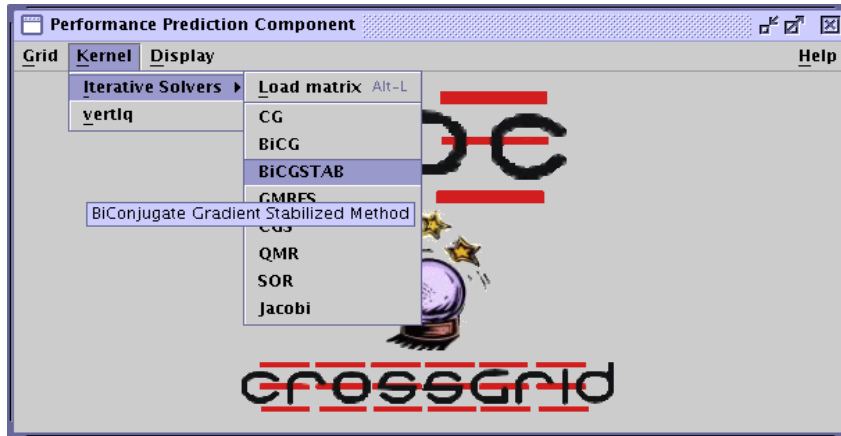


Figure 2.7: Iterative solver selection

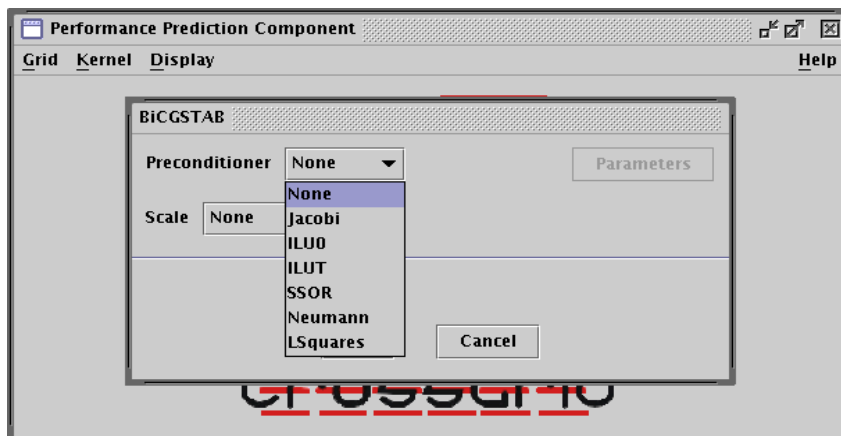


Figure 2.8: Preconditioner selection

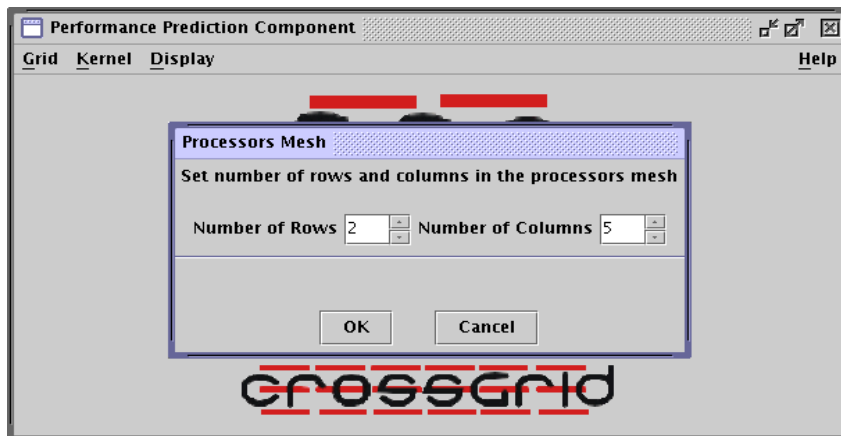


Figure 2.9: Processors mesh

2.2.2 vertlq

VertLQ is the most important routine from the STEM II program [MMG⁺03]. STEM II is the air pollution modeling program used in WP 1.4.2. Routine VertLQ implies typically more than 90% of STEM II total computation time.

In order to simulate the behavior of the VertLQ routine, PPC needs the STEM II input files. These files are provided by Meteorological Simulation (WP 1.4.1), for each day of simulation. These files must be accessible through the MD Grid Commander if PPC is launched from MD, or reside in local file system if standalone.

A set of these files, corresponding to February 15, 2003, can be obtained:

- In the SE at CESGA (se.grid.cesga.es/flatfiles/SE00/cg/PPC/meteo/20030215/).
- In <http://www.ac.usc.es/~crossgrid/PPCTutorialFiles/meteo20030215.tgz> (must be uncompressed & untared)

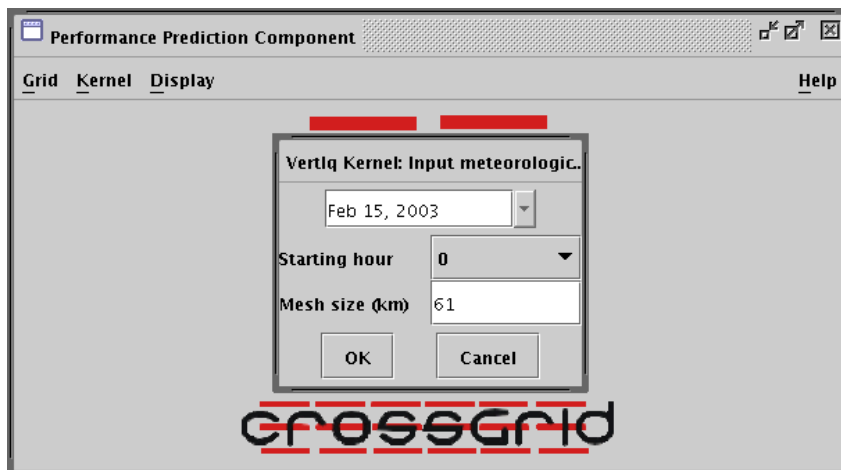


Figure 2.10: Date selection for VertLQ routine

The day and hour of the simulation, and the simulated area have to be selected, using the menu which appears in figure 2.10. The selected day must correspond to the input files day, and the starting hour is the hour of the day to be simulated.

After selecting the simulation parameters, the directory with the input data in the Grid Explorer Dialog of MD or in the local file system if standalone have to be selected. Notice that a directory must be selected (not a file).

2.3 Display

The “Display” button of the main menu is devoted to show the performance results. Different results about the behavior of the kernel can be displayed:

- Number of computations for process (floating point operations).
- Number of communications for process (number of sends and receives).
- Predicted execution time.

- Load balance.
- Other kernel-specific displays.

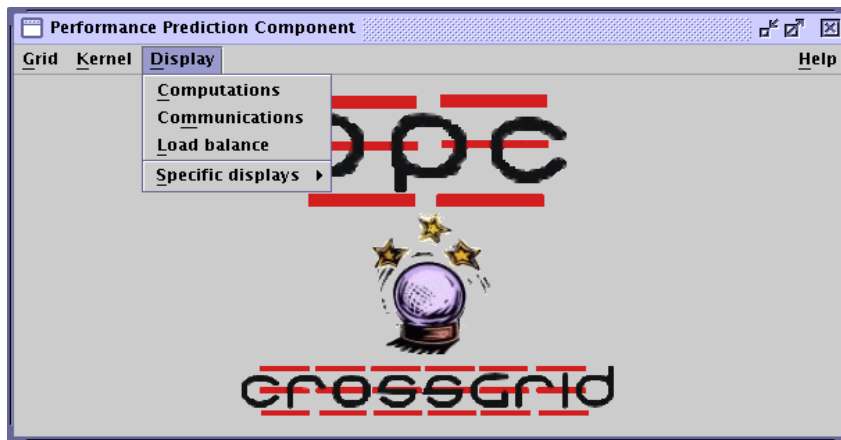


Figure 2.11: Display menu

Currently four suboptions are available (see figure 2.11): “Computations”, “Communications”, “Load balance” and “Specific displays”.

2.3.1 Computations

It shows the number of floating point operations (FLOPs) per node, in the selected grid site. The specific node can be selected.

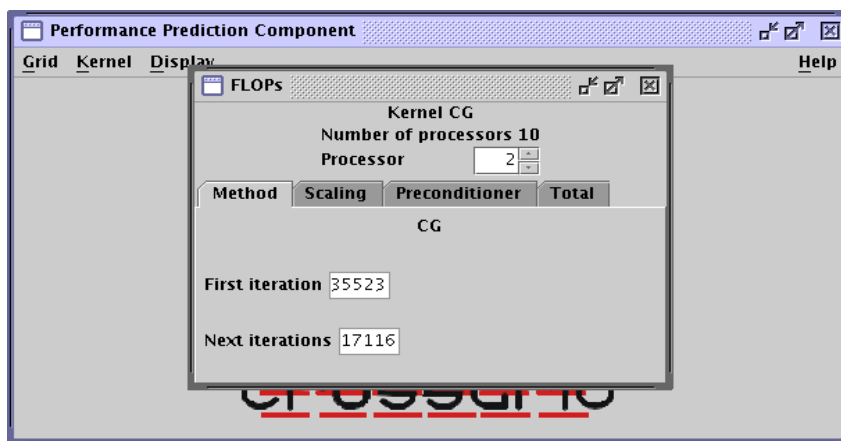


Figure 2.12: Number of FLOPS for an iterative method

For iterative solvers, the FLOPs for different parts of the kernel are showed (figure 2.12):

- Iterative method itself.
- Scaling of the matrix.
- Computation of the preconditioner.

- Application of the preconditioner.
- Total FLOPs for the whole kernel.

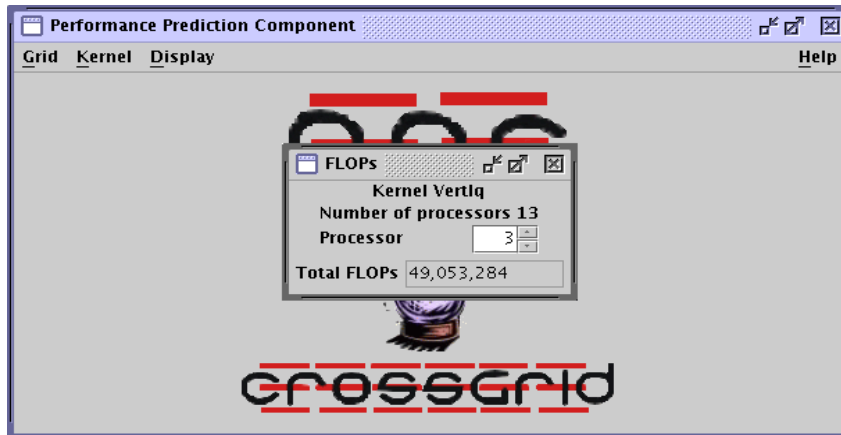


Figure 2.13: Number of FLOPS for VertLQ routine

For VertLQ, the total number of FLOPs for each node is showed (figure 2.13).

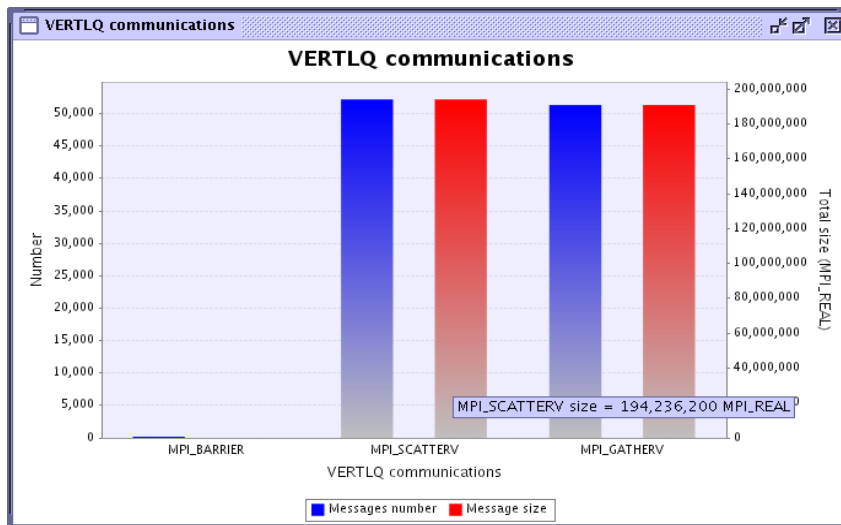


Figure 2.14: Type, number and size of the communications for VertLQ routine

2.3.2 Communications

This facility shows a bar diagram with the number and size of the communications implied in a MPI kernel. At this moment this option is only available for VertLQ kernel. Figure 2.14 shows the type, number and size of the communications implied in the execution of vertlq in the selected grid site.

2.3.3 Load balance

Load balance is shown using Kiviat diagrams [Wu99]. Load balance is expressed in terms of:

- FLOPs.
- Necessary memory.
- Computing time.

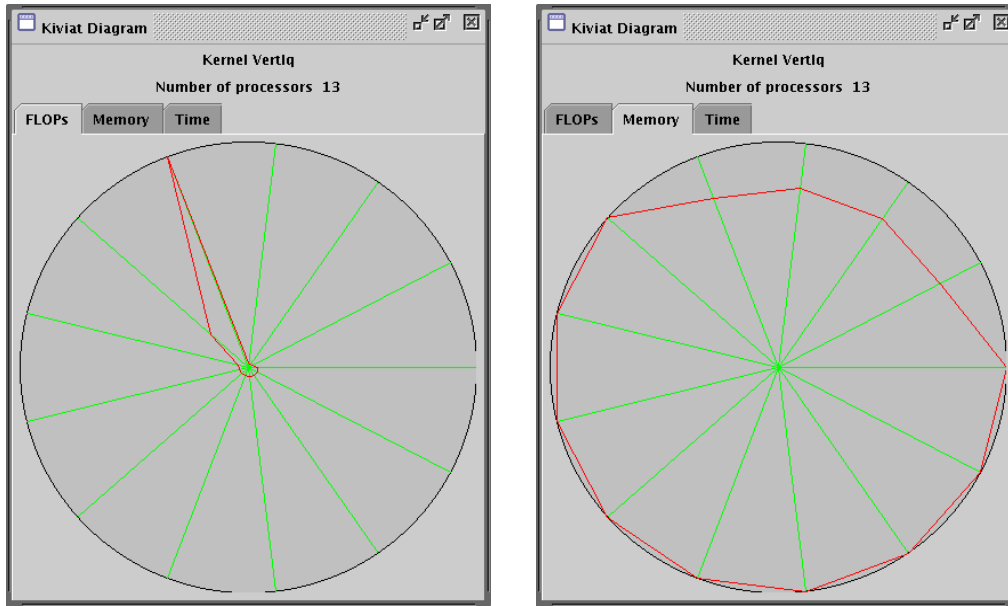


Figure 2.15: Kiviats for FLOPs and memory of VertLQ

Currently, only two different Kiviats diagrams are displayed, for FLOPs and memory usage. For example, figure 2.15 shows the FLOPs and memory Kiviats, respectively, for VertLQ.

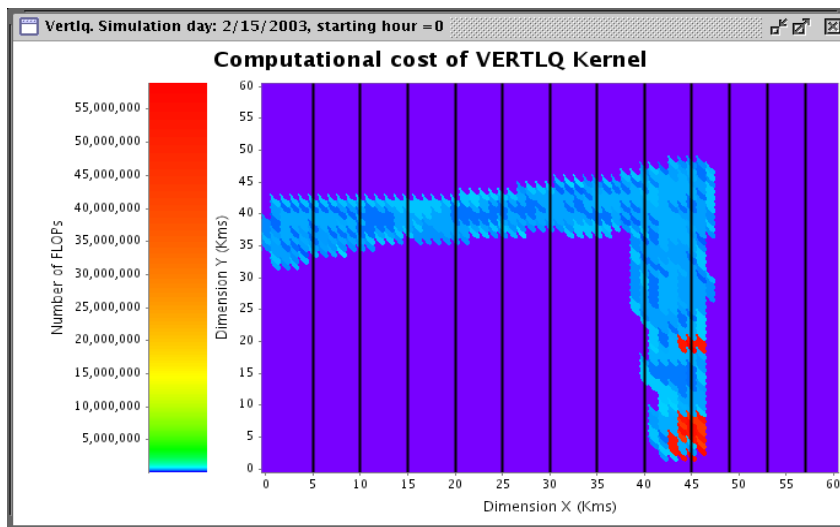


Figure 2.16: FLOPs per mesh point for VertLQ

2.3.4 Specific displays

Shows information that is application dependent, as the distributions of floating point operations in function of the simulation mesh points for VertLQ kernel, figure 2.16. The vertical lines show how the FLOPs are distributed among the cluster nodes.

2.4 Help

Detailed information about PPC usage can be obtained using the available online help.

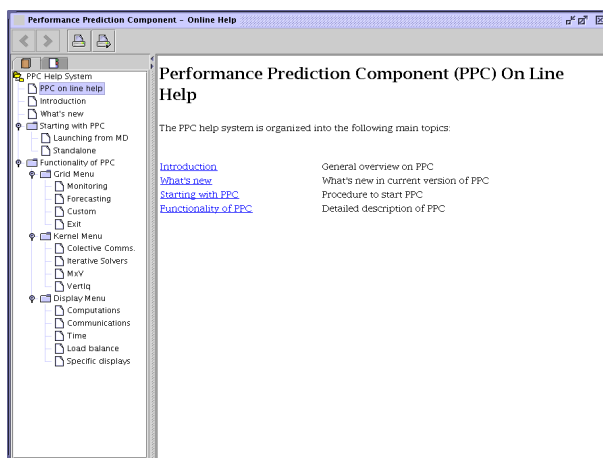


Figure 2.17: On-line help

2.4.1 Help

This facility shows the online help (figure 2.17).

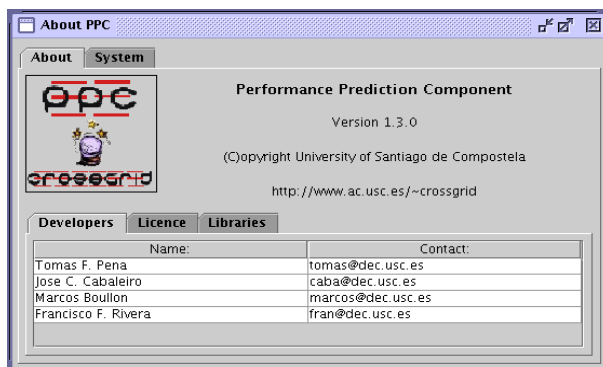


Figure 2.18: About PPC

2.4.2 About

This facility shows about PPC (figure 2.18).

3 Troubleshooting Q and A

Q: Contacting with a JIMS server the message “Cannot get Grid information” appears.

A: The server you are contacting is not running JIMS or it is running a JIMS incompatible version (< 1.4.2). Try to contact another server.

Q: When PPC is reading bandwidth information a message like this:

```
Error: Bandwidth for 149.156.9.34-149.156.9.18 return Infinity
```

appears in console, and the graph for BW appears strange.

A: This is problem with the method that JIMS uses to determine the bandwidth. If the bandwidth between two nodes is too high, JIMS returns an infinity value, that distorts the appearance of the bandwidth graph. Try to contact later with the server, or use another one.

4 Contact Information and Credits

People involved in PPC project are:

- Marcos Boullón Magán <marcos@dec.usc.es>
- J. Carlos Cabaleiro <caba@dec.usc.es>
- Tomás F. Pena <tomas@dec.usc.es>
- Francisco F. Rivera <fran@dec.usc.es>

For bug reporting and commenting on the PPC, please contact us in <ppc@dec.usc.es>

5 EDG License Agreement

This section should contain the EDG agreement, under which CrossGrid software is being licensed. If your software follows a different licensing pattern, replace this text with another license, appropriate for your software.

Copyright (c) 2005 CrossGrid. All rights reserved.

This software includes voluntary contributions made to the CrossGrid Project. For more information on CrossGrid, please see <http://www.eu-crossgrid.org>.

Installation, use, reproduction, display, modification and redistribution of this software, with or without modification, in source and binary forms, are permitted. Any exercise of rights under this license by you or your sub-licensees is subject to the following conditions:

1. Redistributions of this software, with or without modification, must reproduce the above copyright notice and the above license statement as well as this list of conditions, in the software, the user documentation and any other materials provided with the software.
2. The user documentation, if any, included with a redistribution, must include the following notice:

This product includes software developed by the CrossGrid Project (<http://www.eu-crossgrid.org>).

Alternatively, if that is where third-party acknowledgments normally appear, this acknowledgment must be reproduced in the software itself.

3. The names CrossGrid and CG may not be used to endorse or promote software, or products derived therefrom, except with prior written permission by cgooffice@cyfronet.krakow.pl.
4. You are under no obligation to provide anyone with any bug fixes, patches, upgrades or other modifications, enhancements or derivatives of the features, functionality or performance of this software that you may develop. However, if you publish or distribute your modifications, enhancements or derivative works without contemporaneously requiring users to enter into a separate written license agreement, then you are deemed to have granted participants in the CrossGrid Project a worldwide, non-exclusive, royalty-free, perpetual license to install, use, reproduce, display, modify, redistribute and sub-license your modifications, enhancements or derivative works, whether in binary or source code form, under the license conditions stated in this list of conditions.

5. DISCLAIMER

THIS SOFTWARE IS PROVIDED BY THE CROSSGRID PROJECT AND CONTRIBUTORS AS IS AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, OF SATISFACTORY QUALITY, AND FITNESS FOR A PARTICULAR PURPOSE OR USE ARE DISCLAIMED. THE CROSSGRID PROJECT AND CONTRIBUTORS MAKE NO REPRESENTATION THAT THE SOFTWARE, MODIFICATIONS, ENHANCEMENTS OR DERIVATIVE WORKS THEREOF, WILL NOT INFRINGE ANY PATENT, COPYRIGHT, TRADE SECRET OR OTHER PROPRIETARY RIGHT.

6. LIMITATION OF LIABILITY

THE CROSSGRID PROJECT AND CONTRIBUTORS SHALL HAVE NO LIABILITY TO LICENSEE OR OTHER PERSONS FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, CONSEQUENTIAL, EXEMPLARY, OR PUNITIVE DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES, LOSS OF USE, DATA OR PROFITS, OR BUSINESS INTERRUPTION, HOWEVER CAUSED AND ON ANY THEORY OF CONTRACT, WARRANTY, TORT (INCLUDING NEGLIGENCE), PRODUCT LIABILITY OR OTHERWISE, ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Bibliography

- [BB⁺94] R. Barret, M. Berry, et al., *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*, SIAM, 1994.
- [Saad96] Y. Saad, *Iterative Methods for Sparse Linear Systems*, PWS Publishing Co., 1996.
- [BCG⁺02] V. Blanco, J.C. Cabaleiro, P. González, D.B. Heras, T.F. Pena, J.J. Pombo, and F.F. Rivera. Performance of parallel iterative solvers: a library, a prediction model, and a visualization tool. *Journal of Information Science and Engineering. Special Issue: Parallel and Distributed Computing*, 18(5):763–785, September 2002.
- [MMG⁺03] J. C. Mouriño, M. J. Martín, P. González, M. Boullón, J. C. Cabaleiro, T. F. Pena, F. F. Rivera, and R. Doallo. A grid-enabled air quality simulation. In *First European Across Grids Conference*, Santiago de Compostela, Spain, February 2003.
- [Wu99] Xingfu Wu, *Performance Evaluation, Prediction and Visualization of Parallel System*, Kluwer Academic Pub., 1999.