Mirador II - A Monitoring and Management Tool for the SPP3

Dr. Onofre Trindade Jr.[1], Maxweel Silva Carmo[2], Rômulo Eugênio Ribeiro[3]

[1] Computation and Mathematics Science Institute, University of S. Paulo
PO Box 668 - 13560-970 - São Carlos – SP — Brazil
{otjunior@lcad.icmc.sc.usp.br}
[2] Computation and Mathematics Science Institute, University of S. Paulo
PO Box 668 - 13560-970 - São Carlos – SP — Brazil
{maxweel@lcad.icmc.sc.usp.br}
[3] Computation and Mathematics Science Institute, University of S. Paulo
PO Box 668 - 13560-970 - São Carlos – SP — Brazil
{romulo@lcad.icmc.sc.usp.br}

Abstract—

The application of workstation clusters as a parallel machine has been of widespread use. Although highly cost-effetive, the ease of use of such systems can be improved with the utilization of monitoring and management tools. The Mirador II was developed to make easy the user interaction with the SPP3, a parallel architecture that provides high computational power at low cost using commonly avaliable hardware components. User account management, cluster resources monitoring, management and monitoring of running processes, Myrinet network monitoring, parallel machine autonomous management and Internet interface are among its main features. The Mirador II can be also employed, with minor configuration, as a monitoring and management tool for workstation clusters. The daily use of the Mirador II on the SPP3 have shown its usability, making common tasks much easier to accomplish by inexperienced Unix users.

Keywords— **Parallel computing, high speed network monitoring, cluster management.**

I. Introduction

The use of computers for problem solving was based, a long time ago, in strictly sequential solutions imposed by the Von Neumann architecture. In spite of the exponential growth of performance in the last years, physical limitations have imposed restrictions on the maximum processing speed that can be obtained, from Von Neumann computers.

It is known that the need of processing power has been increasing systematically. The solution to some problems demand high computational power that cannot be supplied by a sequential machine. New computational models have been proposed to solve this class of problems. MIMD parallel architectures with distributed memory have considerable importance for offering, among other benefits, high performance at low cost.

With the advent of the parallel machines, there is a need for a new class of tools encompassing procedures and programming paradigms that can support the development of parallel applications.

Frequently, it is necessary to analyze the dynamic behavior of a running parallel application or even interact directly with it. This is not a trivial task without the appropriate tool. Several factors, such as the multiplicity of resources and the concurrent execution of the programs, contribute to make this task not easy. The tools for the management and monitoring of parallel machines make available for the user different information regarding the state of the parallel system and the state of the running applications. These tools allow for, as a management task, the user's direct intervention in the system.

The Mirador II tool was developed for the SPP3 parallel machine and it is an evolution of the Mirador tool [ARA98] for the SPP2 [TRI95a, TRI95b].

The Mirador II stands out as an almost complete tool (some extensions that increase its functionality are related in the *Future Extensions* section) for the management/monitoring of parallel architectures, having several features to accomplish the necessary tasks. Its development was carried out to fill functional gaps present in other tools of its class. Table I presents a comparison among available tools, including the Mirador and Mirador II. This table presents features that are important for the monitoring/management of parallel systems.

The Procps Cluster [PRO99] is an extension of the Top [JOHN99]. Does not have a graphical interface and it does tasks and CPU/memory monitoring only. The VT allows the monitoring of several devices of the parallel machine, such as the hard disk and the communication network. The VT was developed for use on the RS/6000 [SET98] and does not allow management operations.

The AIX PSSP was developed for the IBM SP2 [AGE99]. It provides several management and monitoring operations.

The SCMS was developed for Beowulf machines [STE95, SAL98]. It allows task management and CPU/memory monitoring, among other features.

According to the table I, the Mirador II has inherited all the features of Mirador and includes the main features of other tools. New features were also proposed and implemented.

When compared with the SCMS, the Mirador II presents additional features such as user identification by password, hardware monitoring/management and Internet access.

TABLE I

COMPARISON AMONG MANAGEMENT AND MONITORING TOOLS

Features/ Tool	Procyn Cluster	VT	AIX PSSP	nWatch	Mirador	SCMS	Mirador II
Cluster monitoring	X	X	X	X	X	X	X
Tasks monitoring	X	X	X		X	X	X
Parallel applications monitoring		X		X	X		X
Processing nodes management			X		X	X	X
Tasks management	X		X		X	X	X
Platform independency					X		X
Myrinet Monitoring							X
Parallel Commands support						X	X

II. The SPP3 Parallel Machine

The SPP3 (and the Mirador, among other developments) is the result of a six-year research activity on high performance computing at the LCAD (High- Performance Computing Laboratory), University of São Paulo.

The SPP3 parallel machine is a MIMD architecture with distributed memory. One of its main goals is the use of low cost hardware, following a tendency in the development of distributed memory MIMD machines [STA99]. The SPP3 can be scaled up to 256 processors nodes. Figure 1 shows a block diagram of the SPP3.

Each node of SPP3 is composed, basically, of a standard PC motherboard, a processor, main memory and three network interfaces. The nodes are interconnected together by three different communication networks. A high-speed network (Myrinet) [BOD95] provides inter-processor communication using Myricom adapters and switches [FUJ99]. This network supports the communication among the parallel processes within a parallel application. A fast-ethernet network [IEE95] supports housekeeping functions, such as remote booting. It is also through this network that the user interacts with the processing nodes to accomplish the monitoring and management tasks. This is the only connection between the nodes and the host, allowing them to be separated by a distance of up to 100m. Finally, a I2C network [VOG97]

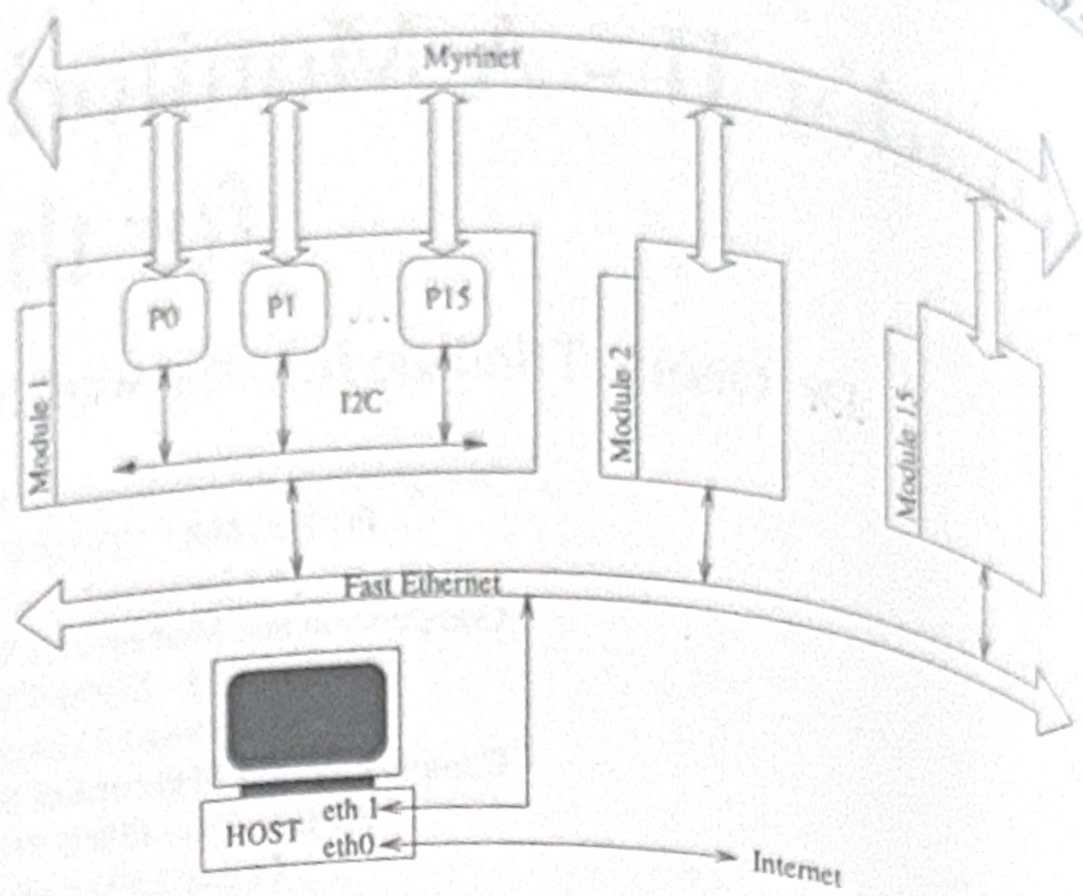

Fig. 1. SPP3 Block Diagram

supports panel and hardware control functions. This network is implemented using a PIC microcontroller and provides a bargraph view of the memory and CPU usage. Some specific tasks, such as the removal from operation of the nodes presenting abnormal behavior, are also available.

III. The Mirador II Tool

The development of the Mirador II was carried out to provide management and monitoring facilities to the SPP3. It is possible, with minor configuration, to use the Mirador II as a management/monitoring tool for a network of workstations acting as parallel machine. The development of the Mirador II was strongly based on the concept of remote use though the Internet. This tool allows the user to gain the control of the machine, making possible:

- To obtain information from the processing nodes such as CPU load, main memory and *swap* memory (size, utilization), current running tasks, processor specification, and time;
- To abort processes running on the nodes;
- To obtain information about the user's tasks to keep track of the execution of parallel applications;
- To manage the user' accounts in a simple and efficient way;
- To monitor the Myrinet communication network;
- To perform management tasks autonomously. The Mirador II can effect some management operations by itself, without user interference;
- To perform monitoring and management tasks remotely through the Internet using a standard Web Browser;
- To monitor the hardware devices of the system namely, motherboard temperature, fan speed (CPU and power supply) and power supply voltage levels;
- To execute selected Unix commands (ls, mv, etc.) in

parallel, making easier the interaction of the user with several processing nodes.

The Mirador II tool is quite flexible, allowing the users to configure several parameters to fulfill their needs.

A. *The Mirador II tool architecture*

The Mirador II tool is based on a client/server model. Client and server modules run on different parts of the parallel machine: processing nodes, user's desktop, the host machine and the PMG (Management and Monitoring Panel) board.

The architecture of the Mirador II tool is presented in the figure 2.

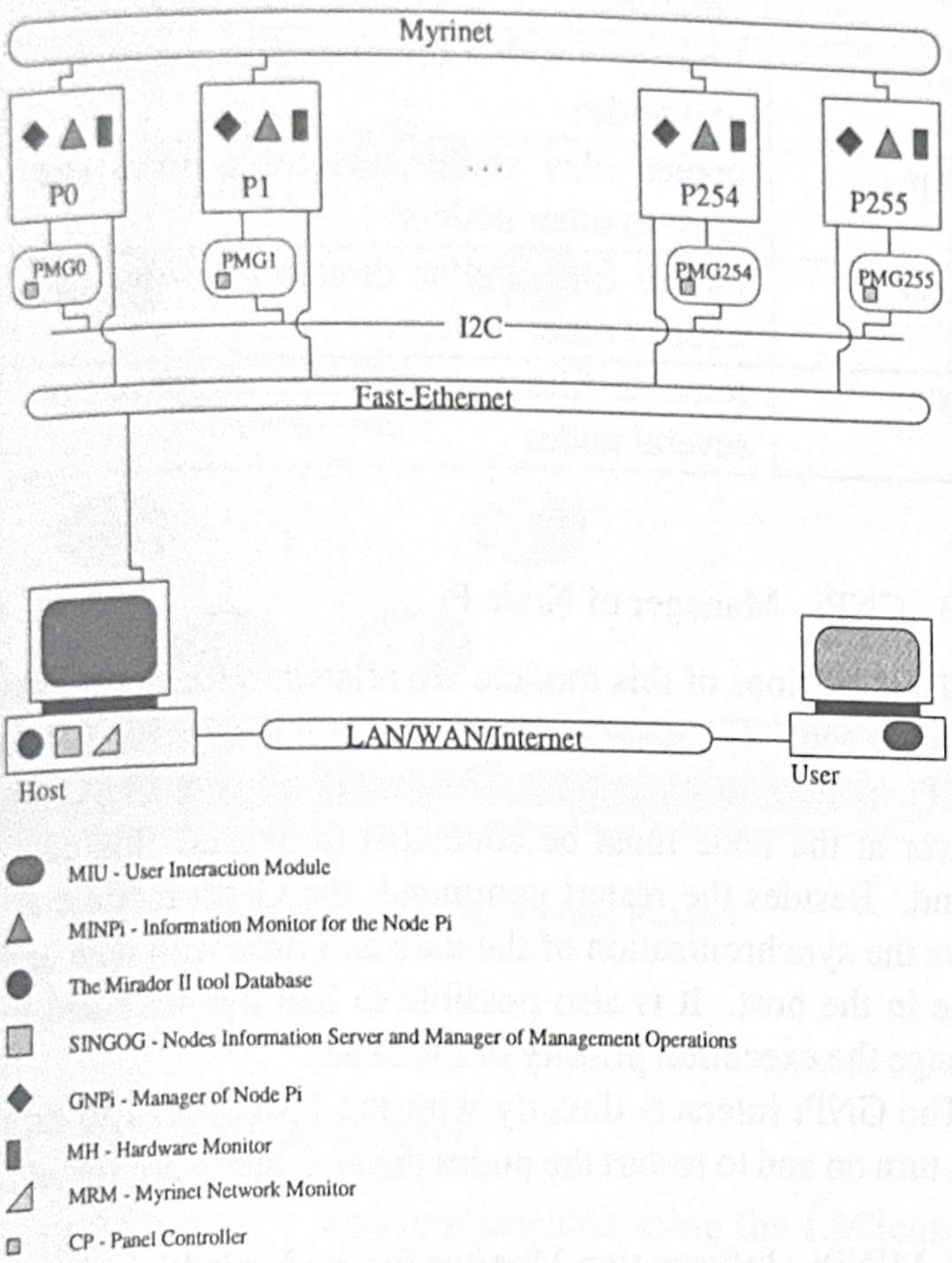

Fig. 2. The Mirador II tool architecture

The Mirador II tool consists of the following modules:

A.1 MIU - User Interaction Module

This module is in charge of the interaction between the user and the SPP3, being executed locally (in the user's desktop). A graphics interface presents the information under monitoring in structured and modular views, as shown in figure 3.

The MIU is also in charge of all management operations and user configuration options (user profile configuration).

The Mirador II main window is quite uncluttered and is presented in the figure 4. Each icon represents a processing

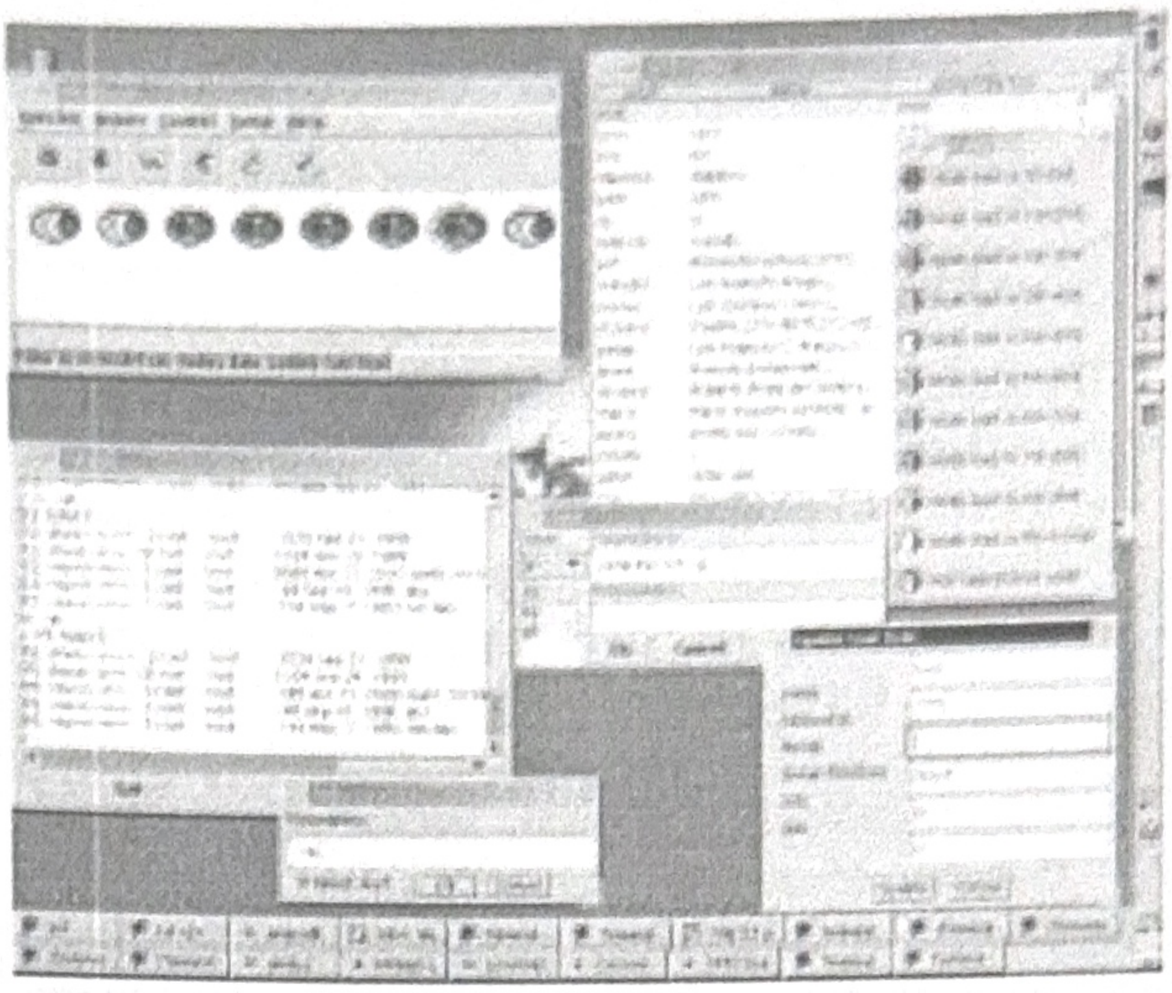

Fig. 3. The Mirador II tool graphics interface

node of the parallel machine.

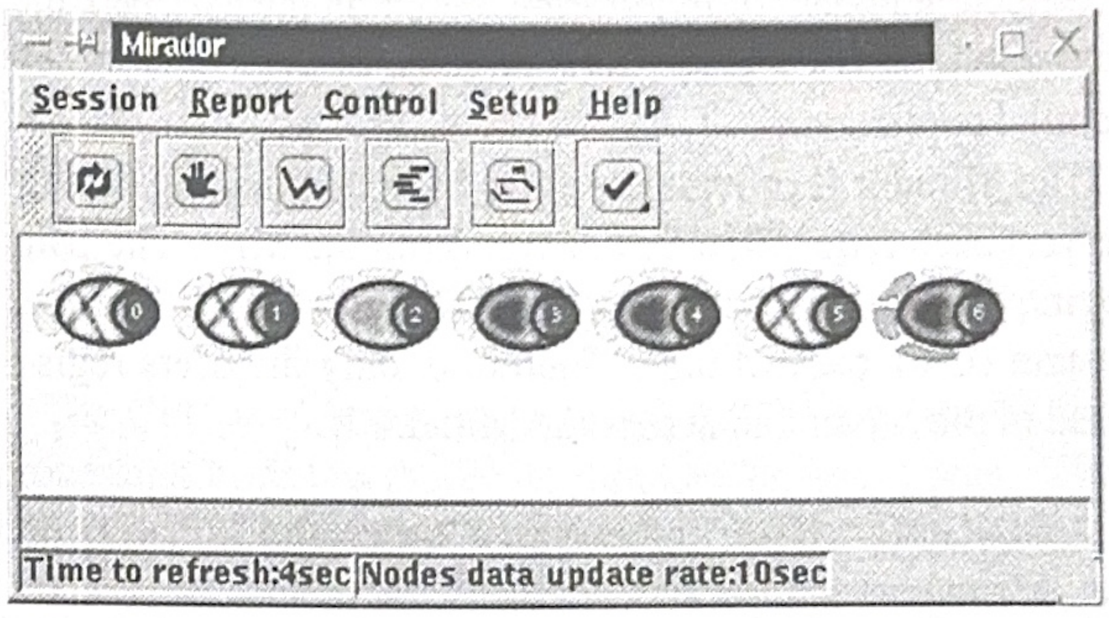

Fig. 4. The Mirador II main window

According to the figure 4, five nodes are under monitoring. The colors of the ellipses change to show CPU loads, allowing for an easy identification of the computational activity on each node. The black color, for example, indicates the absence of computational activity, while the white color indicates a high activity level. The ellipses labeled with an "X" indicate inoperative nodes. The colors of the small circles on the right side of each ellipse indicate whether the node is selected for management or not. The nodes selected for management allow user interaction. For instance, to turn off a group of nodes, each node must be selected for management before the Turn Node Off command be issued. The semicircles around the ellipses indicate memory utilization. Their values have the following meaning:

0%;
from 1 up to 20%;
from 20 up to 40%;
from 40 up to 60%;
from 60 up to 80%;
from 80 up to 100%.

The selection of the node(s) under monitoring can be made using parameters (or filters):

- Selecting a certain number of nodes. A range of nodes can be selected for monitoring. For example, the nodes numbered from 0 to 3 (0-3). Also, specific nodes can be selected. For example, the nodes 0, 2, 5 and 8 (0,2,5,8);
- Selecting nodes by specific tasks. The nodes executing a certain task(s) are selected. For example, the nodes that are executing the Netscape Navigator;
- Selecting nodes by specific user. Only the nodes that are executing at least one task owned by a specific user are selected for monitoring;
- Selecting nodes by specific user and specific task. In this case, only the nodes that are executing a specific task owned by a specific user are selected for monitoring;
- Selecting nodes under computational activity. In this case, nodes out of order or turned off are not monitored.

Selecting nodes for monitoring, based on the user/task information, makes it easier to keep track of the execution of parallel applications.

The Mirador II provides security by authenticating (in the server) each requisition that arrives from the MIU. The pair login/password is checked against the password list in the system (Unix passwd file). Therefore, only the users registered in the server can access the Mirador II.

A.2 SINGOG - Node Information Server and Manager of Management Operations

The SINGOG module, running in the host, manages all the communication between the user and the processing nodes. It gathers and stores all the management information of each node. Periodically, all the modules that run in the nodes contact the SINGOG module and send monitoring information (CPU load, memory utilization, date, tasks in execution, etc.). This information is filtered by the SINGOG module and sent on request to the user. This implementation optimizes the use of the communication network, avoiding excessive traffic.

The SINGOG module also triggers management operations, such as Kill Tasks. It gets the management requisition, contacts the node(s) where the management operation will be executed, waits for a response of the node(s) and returns the response (success or error) to the user.

The SINGOG module also has specific functions:

- User Accounting - the Mirador II performs user account management, in a simple and functional way, using a graphics interface provided by the MIU. The SINGOG module implements the functions that realize this activity. The basic operations are: add user, remove user and update users' data;
- Parallel Command Execution: to facilitate the interaction between the user and the nodes, basic Unix commands can be executed in parallel. Table II shows the implemented commands.

TABLE II
COMMANDS EXECUTED IN PARALLEL

Command	Function
ls	lists files and/or directories for one or several nodes
find	finds files and/or directories in one or several nodes
cp	copies files and/or directories from one node to other node(s)
mv	moves files and/or directories from one node to other node(s)
rm	removes files and/or directories for one or several nodes

A.3 GNP*i* - Manager of Node P*i*

The functions of this module are related to the node tasks management. For each of the nodes in activity there are a GNP*i* module in execution. To restart a node, the GNP*i* server at the node must be contacted to execute this command. Besides the restart command, the GNP*i* module allows the synchronization of the date and time with date and time in the host. It is also possible to halt the node and to change the execution priority of the tasks.

The GNP*i* interacts directly with the PMG board to turn off, turn on and to restart the nodes through hardware signals.

A.4 MINP*i* - Information Monitor for the Node P*i*

This module gathers and process information about main and *swap* memory usage, tasks under execution and CPU load. At the time ticks specified by the user, such information is sent to the host (SINGOG).

The MINP*i*, like the GNP*i*, runs in each processing node.

A.5 MRM - Myrinet Network Monitor

This module monitors the Myrinet high-speed network. The SNMP agent in the Myrinet switch maintains the monitoring data. The data is obtained through an Ethernet connection. At the power up, the host supplies an IP address (DHCP) to the switch.

The Myrinet monitoring data include physical and configuration parameters (number of ports, switch communication

timeout, etc.) and statistics about the flow of packets on each port (number of packets, number of bad CRC packets, port status, etc.).

The MRM also makes available some management functions to enable/disable ports and to restart the switch.

The network traffic is presented in a graphic format, as illustrated in figure 5. The color and format of the connections change according to level of the flow of data. Other monitoring data is available in table form.

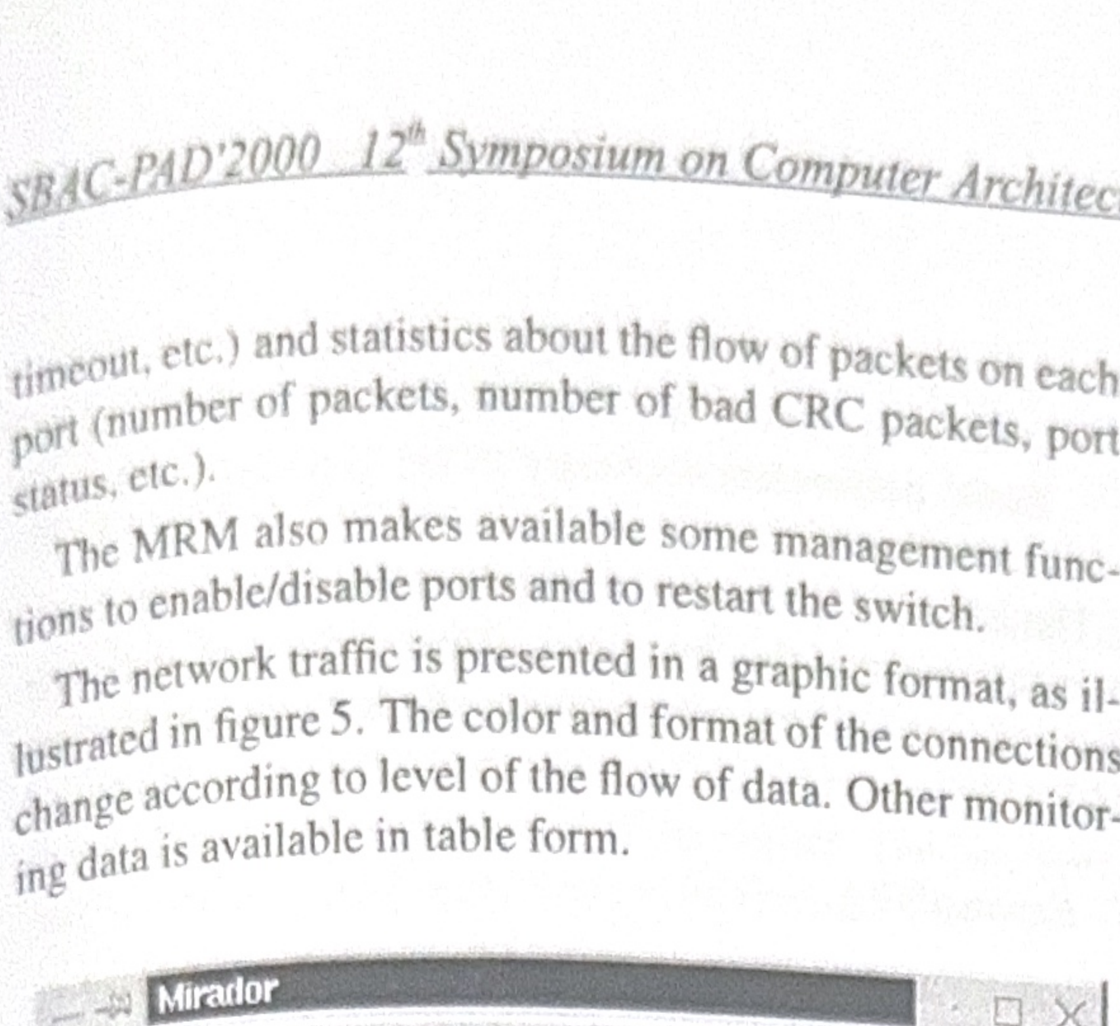

Fig. 5. Monitoring the Myrinet Network

A.6 MH - Hardware Monitor

The MH module was implemented using the LMSensor package [NAT00]. Such application interacts directly with the sensing hardware of the ATX motherboards performing temperature measurement (motherboard), voltage measurement (power supply and CPU core) and fan speed (power supply and CPU).

The MH deals with the management of hardware malfunctions. It checks the measured data (all types) against user-established minimum and maximum values. If these values are out of the established limits, a message is sent to the administrator by e-mail, informing which device/value is out of range. The user can configure the system to turn off the malfunctioning node if no action is done to correct the out-of-cimits data after a specified amount of time. This is a protective procedure to avoid permanent hardware failure to the system.

A.7 CP - Panel Controller

This module runs on the PMG and comprehends the low level functions that coordinate all of its activities, accomplishing monitoring and management tasks.

A.8 PMG - Management and Monitoring Panel

The PMG board was designed to accomplish two basic functions:

- Monitor CPU load and memory utilization;
- Turn on/off and reset nodes.

The PMG board has two bargraph displays, a micro controller, an I2C interface and a RS232 serial port that is connected to the node. The RS232 port is the communication channel among the processing nodes and the PMG boards. Monitoring tasks are periodically being executed, while management tasks are executed under user or system request.

The board is also connected to the reset and to the on/off control inputs in the processing nodes. Figure 6 shows the interaction among PMG boards and nodes.

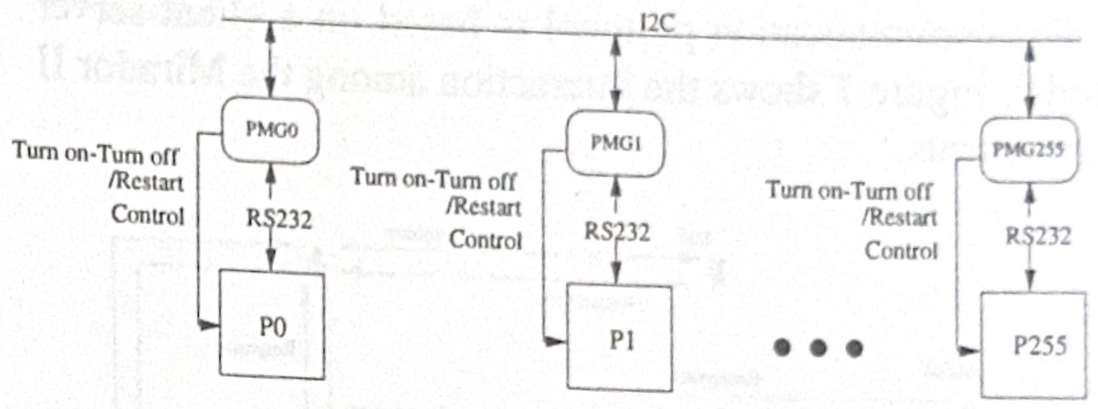

Fig. 6. SPP3 Control Network

The CPU load and memory usage values are sent by the processing nodes to the PMG thought the serial port. The bargraph display present these values.

The management tasks demand, most of the time, some interaction among the PMG boards via the I2C network. Management operations require at least one fully functional processing node to send commands to the control network. A List_of_Nodes parameter specifies which nodes must be affected by the command execution.

The first fully functional processing node (ordered by node number) is chosen to send management commands to the control network. For example, consider a situation where the nodes 0, 1 and 2 are being monitored and only node 1 is fully functional. To turn off the other nodes, a command is issued by node 1 to the PMGs of nodes 0 and 2.

A.9 The Mirador II Database

The Mirador II database is located in the host. It stores five types of information:

Main Configuration:

- Processing nodes specification (processor type and speed, amount of main and swap memory);

- Node icons presentation format;
- Time interval for automatic screen refreshing;
- Connection timeout (user desktop to the host computer);
- Time interval between consecutive data gathering on the nodes (by the host);
- Connection timeout (host computer to the nodes).

User Configuration: The Mirador II allows each user to have his/her own configuration settings, including monitoring/management options, communication timeouts, refreshing rate, etc.

Parallel Machine Configuration: number of processing nodes and node specification.

Node Monitoring Information: CPU and memory utilization, tasks under execution, date and time, CPU and fan speed, CPU core voltage, power supplier voltage and motherboard temperature.

Myrinet Network Configuration: number of Myrinet switches and node/port listing.

B. Communication Protocol

The communication protocol is based on a client-server model. Figure 7 shows the interaction among the Mirador II components.

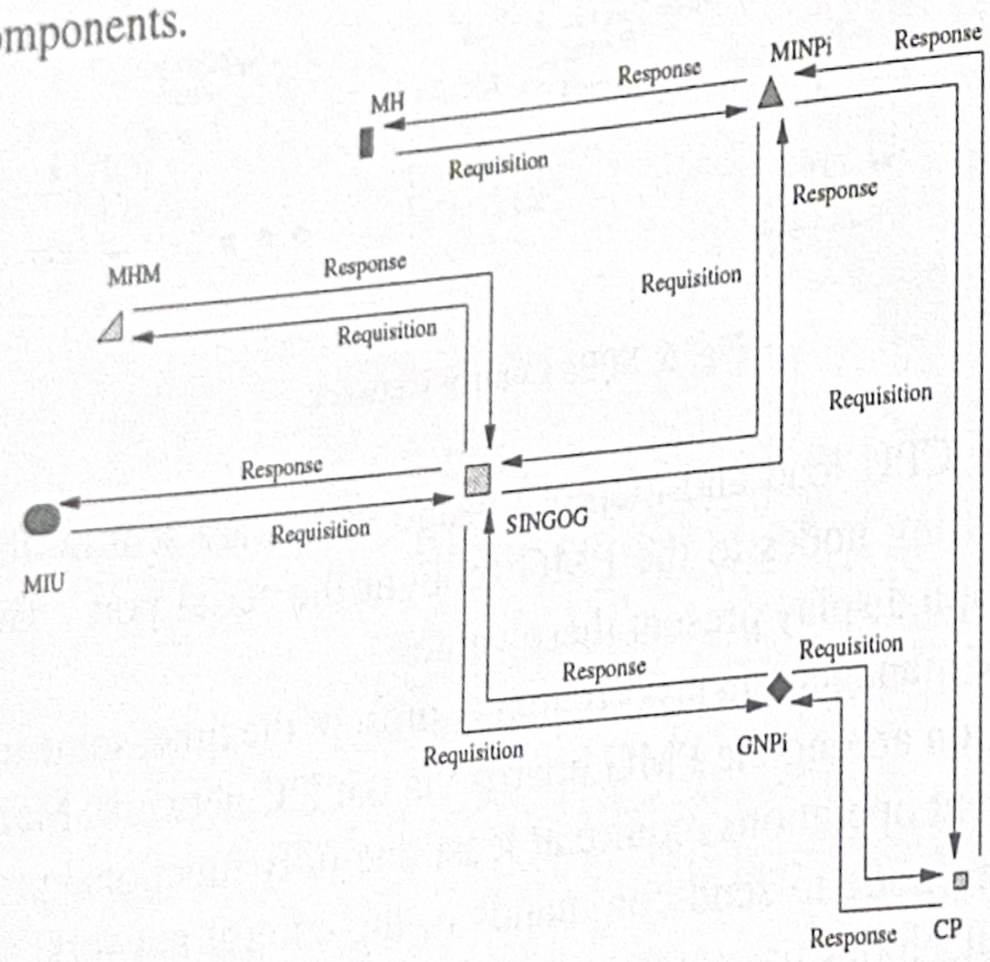

Fig. 7. The Mirador II Communication Protocol

A service requisition includes the following parameters:

- Service identification. A module can implement several functions (services). It is necessary to inform specifically which service is being requested;
- User identification. To access some services provided by SINGOG, the system requests the user's identification (login and password) in order to maintain security of the system (for example, a common user can not abort the execution of tasks not owned by him);
- Operation to be executed by the service. Some services make available more than an operation and it is necessary specify which one is being requested;
- Necessary data to the execution of the service. Additional parameters necessary to the execution of the requested service.

The Mirador II was designed to use low bandwidth communication over the Internet. The amount of data exchanged was reduced, comparing to the first implementation ([ARA98]).

C. Expansibility

One of the main features of the Mirador II is its expansibility, presenting mechanisms for the inclusion of new functionalities in agreement with the user's specific needs. Such feature is supported by the design of the Mirador II communication interface, making available a well defined and consistent API (Application Programming Interface) for easy utilization. A practical example of an extension is the development of new or the use of available visualization interfaces. These interfaces can request the monitored data making use of the available API services.

D. Resource Utilization

The Mirador II was designed to be as non-intrusive as possible to the monitored processing nodes. It has very low needs of computational power and memory.

E. Flexibility

The Mirador II tool was developed for the SPP3 parallel machine. It can be also used, with minor configuration and some loss of functionality, in a network of up to 256 Linux workstations (Beowulf-like machines).

The Myrinet module can be ignored at compile time when the Myrinet network is not available. The user can also excludes other modules at compile time, as a sort of customization mechanism, to make the Mirador II appropriate for his/her needs.

F. Usability

The usability of the Mirador II has been increased by evolutive maintenance motivated by daily use. The main features of Mirador II to accomplish better usability are:

- Several options to choose the nodes under monitoring allows a better trace of the running parallel application;
- The autonomous management provided by an integration of software and dedicated hardware helps machine maintenance routines such as turning off nodes that are presenting abnormal behavior (e.g. high temperature or fan failure);
- In addition, the Mirador II allows the user (administrator), to restart, to halt and to turn off/on processing nodes with great facility;

- The remote usage trough the Internet also improves the usability level of the Mirador II.

G. Future Extensions

Further work was proposed to extend the functionality of the Mirador II. Commands can be added to allow the user to make a pre-allocation of nodes for exclusive use. Load balancing is another issue. Dynamic load balancing would benefit the overall performance of the cluster. The batch processing of user's jobs is another useful functionality.

IV. Conclusion

With the growing use of MIMD architectures (workstation clusters) to solve problems that demand high computational power, there is a need for tools to monitor and manage this class of architectures. The Mirador II was designed mainly to be used in the SPP3, although it can be employed in similar machines such as the SPP2, Beowulf and network based clusters.

The Mirador II is a wide system, providing tools ranging from users' account management to hardware monitoring. Internet access is an important characteristic, allowing for remote monitoring and/or remote management of the parallel machine.

The use of the Mirador II in the SPP3 at the LCAD has demonstrated its usability, effectiveness and flexibility.

References

[AGE99] AGERWALA, T.; MARTIN, J.; SADLER, D.; DIAS, D.; SNIR, M. SP2 system architecture. **IBM Systems Journal**, v38, n.2-3, p.414-448, 1999.

[ARA98] ARAÚJO, L. Mirador - **Uma ferramenta para monitoramento e gerenciamento do SPP2**. Master's thesis, São Carlos: University of S. Paulo, 1998.

[BOD95] BODEN, N. et al. Myrinet: a gigabit-per-second local-area network. **IEEE Micro**, Los Alamitos, v.15, n.1, p.29–36, Feb. 1995.

[FUJ99] FUJISAKI, S. **Avaliação de tecnologia e rede de alto desempenho para utilização no SPP2**. Master's thesis, São Carlos: University of S. Paulo, 1999.

[IEE95] INSTITUTE OF ELECTRICAL AND ELECTRONIC ENGINEERS. **Local and metropolitan area networks-supplement—media access control (MAC) parameters, physical layer, medium attachment units and repeater for 100Mb/s operation, type 100BASE-T (clauses 21–30)**, IEEE 802.3u-1995. New York, NY, 1995.

[NAT00] LM78 Microprocessor System Hardware Monitor. http://www.national.com/pf/LM/LM78.html, 2000.

[PRO99] Procps-Pare. http://www.sc.cs.tu-bs.de/pare/results/procps.html, 1999.

[SAL98] SALMON, J. **Scaling of Beowulf-class Distributed Systems**. In: SC'98: High Performance Networking and Computing: Proceedings of the 1998 ACM/IEEE SC98 Conference: Orange County Convention Center, Orlando Florida, USA, Nov. 1998.

[SET98] SETHU, H.; STUNKEL, C.; STUCKE, R. **IBM RS/6000 SP Interconnection Network Topologies for Large Systems**. In: Proceedings of the 1998 International Conference on Parallel Processing (ICPP '98), pp. 620-628, IEEE USA, Aug. 1998.

[STA99] STALLINGS, W. **Computer Organization and Architecture: Designing for Performance**. Prentice Hall, fifth edition, 1999.

[STE95] STERLING, T.; SAVARESE, D.; BECKER, D.; DORBAND, J.; RANAWAKE, U.; PACKER, V. **BEOWULF: A Parallel Workstation for Scientific Computation**. In: Proceedings of the 24th International Conference on Parallel Processing, p. I:11-14, Aug. 1995.

[TRI95a] TRINDADE, O.; MARQUES, E.; JEUKENS, I. **A parallel Architecture based on personal computers - requirements and definitions**. In: Simpósio Nipo-Brasileiro de Ciência e Tecnologia, p. 203-212, Aug. 1995.

[TRI95b] TRINDADE, O.; MARQUES, E.; JEUKENS, I. **A parallel Architecture based on personal computers - requirements and definitions - an overview**. In: XV International Conference of the Chilean Computer Society, p. 479-490, Nov. 1995.

[JOHN99] JOHNSON, M. Index of docs/procps-2.0.6. http://iglu.org.il/doc/procps-2.0.6, 1999.

[VOG97] VOGL, Simon. Using the I2C Bus with Linux. **Linux Journal**, v35, Mar. 1997.