Using an MPI Cluster in the Control of a Mobile Robots System

Mohamed Salim LMIMOUNI, Saïd BENAISSA, Hicham MEDROMI, Adil SAYOUTI
Equipe Architectures des Systèmes (EAS), Laboratoire d'Informatique, Systèmes et Energies Renouvelables (LISER)
Hassan II University–Aïn Chock, Ecole Nationale Supérieure d'Electricité et de Mécanique (ENSEM)
Casablanca, Morocco

Abstract—**Recently, HPC (High Performance Computing) systems have gone from supercomputers to clusters. The clusters are used in all tasks that require very high computing power such as weather forecasting, climate research, molecular modeling, physical simulations, cryptanalysis, etc. The use of clusters is increasingly important in the scientific community, where the need for high performance computing (HPC) is still growing. In this paper, we propose an improvement of a mobile robots system control by using an MPI (Message Passing Interface) cluster. This cluster will launch, manipulate and process data from multiple robots simultaneously.**

Keywords—clusters; MPI; parallel programming; mobile systems; mobile robots

I. Introduction

Today, almost all industries require fast processing power. With the increasing availability of cheaper and faster computers, many companies are interested in the technological benefits. Supercomputers were developed to meet the needs of fast processing. However, a typical supercomputer can cost usually over a million dollars because of specialized hardware and software. Therefore, clusters have been presented as an alternative to supercomputers.

A cluster is defined as a group of independent computers linked with a computational network and operating as a single computer [1]. In other words, a cluster is a collection of independent and cheap machines, used together as a supercomputer to provide a solution [1].

Using clusters has many advantages:

- The computers that form a cluster are less expensive than supercomputers.
- You can add other nodes (computers) to the cluster as needed.
- On clusters, you can use open source software to reduce software costs.
- Clusters allow multiple computers to work together to solve several problems.

Mobile robots are used as autonomous systems to make autonomous processes more flexible and efficient. In this kind of application, mobile robots systems are guided and controlled by supervisory systems. There are numerous scientific papers about parallel algorithm and scheduling methods in this area but there is no reliable implementation on mobile robots systems in the industry.

By using an MPI cluster in the control of a mobile robots system, we can launch multiple mobile robots at the same time, it also allows us to handle a large number of mobile robots in real time as well as the treatment of recovered data from different sensors of these mobile robots more efficiently.

In this paper, some cluster definitions, a short history, the typical architecture, various types of clusters and MPI are presented in the second section. The third section gives a short view on mobile robots. The fourth section shows the existing architecture and the proposed improvement. In the fifth section, we describe the implementation.

II. Clusters

A. Definitions

A cluster is a single system comprised of interconnected computers that communicate with one another either via a message passing; or by direct, internode memory access using a single address space [2]. We can also define a cluster as a commonly found computing environment consisting of many PCs or workstations connected together by a local-area network [3].

B. History

The official technical basis of clusters as a way to do a parallel work of any kind was probably invented by Gene Amdahl of IBM, who released in 1967 what is considered the seminal paper of parallel processing: the Amdahl's law [4]. Amdahl's Law describes mathematically the performance gain expected after the parallelization of a task on a parallel architecture.

But if we take a glance at the history of clusters, we find that the first commercial cluster was ARCnet [5], developed by Datapoint in 1977. ARCnet was not a commercial success and clusters have really taken off after the development of VAXcluster by Digital Equipment Corporation in 1984 for the VAX / VMS operating system.

The history of clusters would not be complete without mentioning the crucial role played by the development of the Parallel Virtual Machine (PVM) in 1989 [6]. This open source software based on the TCP/IP protocol has allowed the creation of a virtual supercomputer from any system connected through a TCP/IP network.

PVM and the availability of cheap computers have led, in 1993, to a NASA project that consists on the development of supercomputers based on clusters. In 1995, the Beowulf cluster [7] has emerged.

Today, the clusters of Beowulf type are ubiquitous and occupy the top spots in the TOP500 site [8] [9].

C. Typical architecture

The typical architecture of a cluster is shown in Figure 1 [10]. A node of the cluster can be a single or multiprocessor system, such as a PC, workstation, or SMP (Symmetric MultiProcessor). The nodes must be connected via a LAN (Local Area Network) based on Ethernet, Myrinet or InfiniBand. The cluster middleware offers an illusion of a united system of the independent nodes. Parallel programming environments offer portable, efficient and easy-to-use tools for developing parallel applications.

D. Types

There are three varieties of clusters, each one offers different benefits for the user. These varieties are:

- **Load balancing clusters:** used to provide a single interface for a set of resources that can grow arbitrarily. We can imagine a web server that redirects client requests to another node when it has reached its limit of load. This is called "load balancing". Only the node that handles the distribution is visible from the outside.
- **High performance clusters:** They consist of a set of computers linked together to provide maximum power in solving a problem. The heart of these clusters is formed of compute nodes that will receive the code to execute. On smaller clusters we can count ten nodes, while the largest have more than 80 000. The network architecture in place for communication between nodes becomes very heavy, expensive and it limits its performance. You should know also that the ratio of the number of nodes and the performance of this type of
- Clusters is not linear. It is necessary that the program executed is highly parallelizable and that it requires little communication between the computing units.
- **High availability clusters:** The High Availability clusters are built to provide a secure and fault tolerant environment. The redundancy is the most used method. It consists on multiplying the material that could be subject to failure. Server applications are installed the same way on the cluster nodes.

E. MPI

MPI (Message Passing Interface) is a specification for a standard library for message passing that was defined by the MPI Forum, a broadly based group of parallel computer vendors, library writers, and applications specialists [11]. MPI is not an IEEE or ISO standard, but has become the industry standard for writing message passing programs on HPC platforms.

There are many reasons for using MPI:

- **Standardization:** MPI is the only message passing library which can be considered a standard. It is supported on virtually all HPC platforms. Practically, it has replaced all previous message passing libraries.
- **Portability:** There is little or no need to modify your source code when you port your application to a different platform that supports the MPI standard.
- **Performance Opportunities:** Vendor implementations should be able to exploit native hardware features to optimize performance.
- **Functionality:** There are over 440 routines defined in MPI-3, which includes the majority of those in MPI-2 and MPI-1.
- **Availability:** A variety of implementations are available, both vendor and public domain.

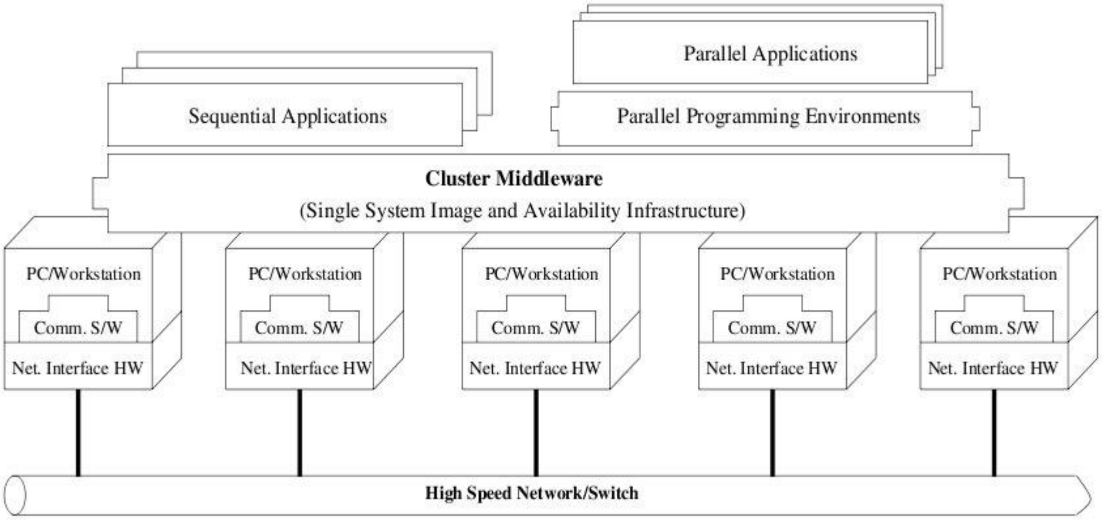

Fig. 1. Typical architecture of a cluster

III. Mobile Robots

A. Definitions

The word robot was introduced in 1921 by Czech writer Karel Capek in his play RUR (Rossum's Universal Robots), as the name for the artificial people created by an unseen inventor (Rossum) to replace humans in any job. Capek described these artificial people as robots, a word derived from the Czech word robota meaning literally serf labor. Robots were constructed to serve and to free humans from any type of labor [12].

This concept has evolved over time and several definitions for the word now exist in the literature.

In mobile robot systems, robot control architecture is necessary. Robot control architecture is defined as a mapping of sensory information into actions in the real world, in order to accomplish a certain task. It is a way of integrating different kinds of hardware and software modules. Furthermore, it plays an important role in maintenance tasks and in the addition of new modules.

B. Khepera III

The K-team [13] robot, Khepera III [14] is a small, circular mobile robot running on two wheels and a sliding support. The diameter is about 130 mm, the height about 70 mm and the weight about 690 g. Different views of the Khepera III are presented in Figure 2, the top row showing a prototype and the bottom row the current commercially available version.

In its basic configuration, the Khepera III is equipped with two motors with associated controllers, a ring of 9 infrared (IR) sensors attached to the bottom layer of the robot's internal structure, another ring of ultrasonic (US) sensors attached to the second layer and an additional pair of IR sensors pointing downward (called ground sensors). Communication with and control of these devices is mediated by a dsPIC 30F5011 microprocessor.

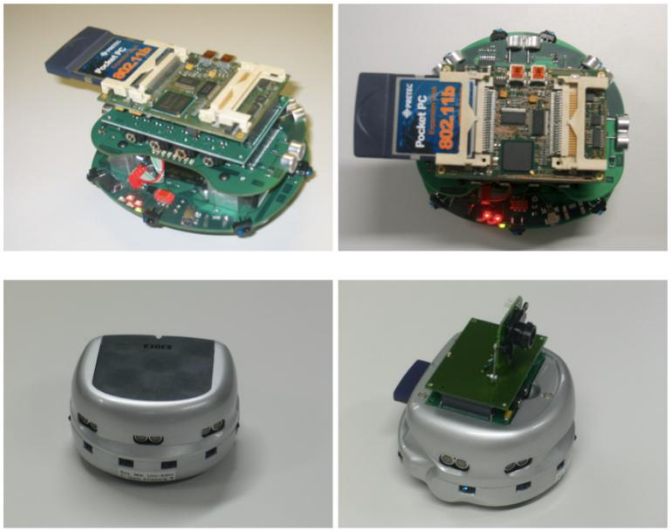

Fig. 2. Khepera III

IV. Proposed Improvement

A. Existing architecture

As part of research conducted within the EAS (Equipe Architectures des Systèmes) team, a mobile system architecture was proposed in 2008 (Figure 3) [15].

The user, through its web browser (Internet Explorer, Google Chrome, etc.), connects to a web server (step 1) using the TCP/IP protocol and download an application on his workstation (step 2). A connection is established to the server responsible for the management of targeted mobile systems (step 3) and after verification, the user can take its control. Parallel to the 3rd step, connections are also established to multimedia servers offering views (video, sound) of the controlled system.

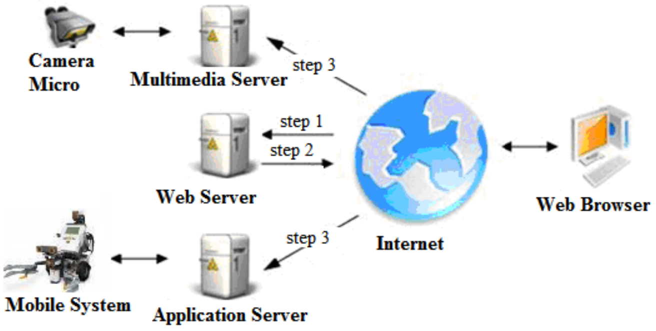

Fig. 3. Mobile system architecture proposed by the EAS team

B. Proposed architecture

In the previous paragraph, we presented the mobile system architecture proposed by the EAS team. This architecture presents some limitations in the Application Server that manages the mobile robots if they are numerous or if they use data from multiple sensors.

To overcome these limitations we propose the architecture in Figure 4 [16] where we replace the Application Server on the old architecture with an MPI Cluster that can launch

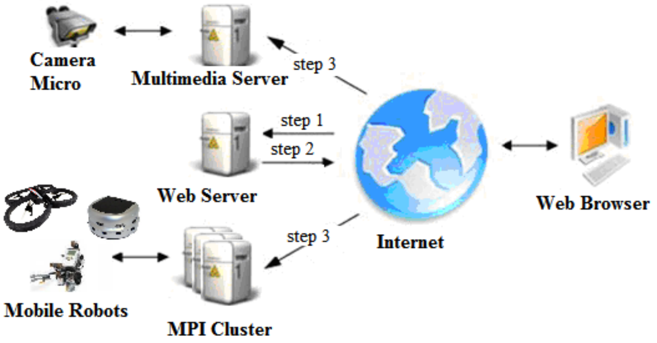

Fig. 4. Improved architecture of the mobile system proposed by the EAS team

multiple mobile robots at the same time, handle data from multiple mobile robots in real time and process recovered data from different sensors of these mobile robots more efficiently.

Figure 5 shows the different layers of the proposed improvement:

- **MMI (Man Machine Interface):** The layer that will allow users to launch, manipulate and process data of mobile robots using graphical tools.
- **Parallel Program:** The code that will launch several mobile robots at the same time, manipulate and process data of these mobile robots in a parallel way.
- **MPI:** The library that will compile and run parallel programs on the cluster.
- **Cluster:** The different nodes (machines) of the cluster. These nodes are connected by a network.
- **Communication:** The layer that will ensure the communication between the cluster's machines and the mobile robots.
- **Mobile Robots:** The different mobile robots that will be handled and whose data will be processed using the cluster.

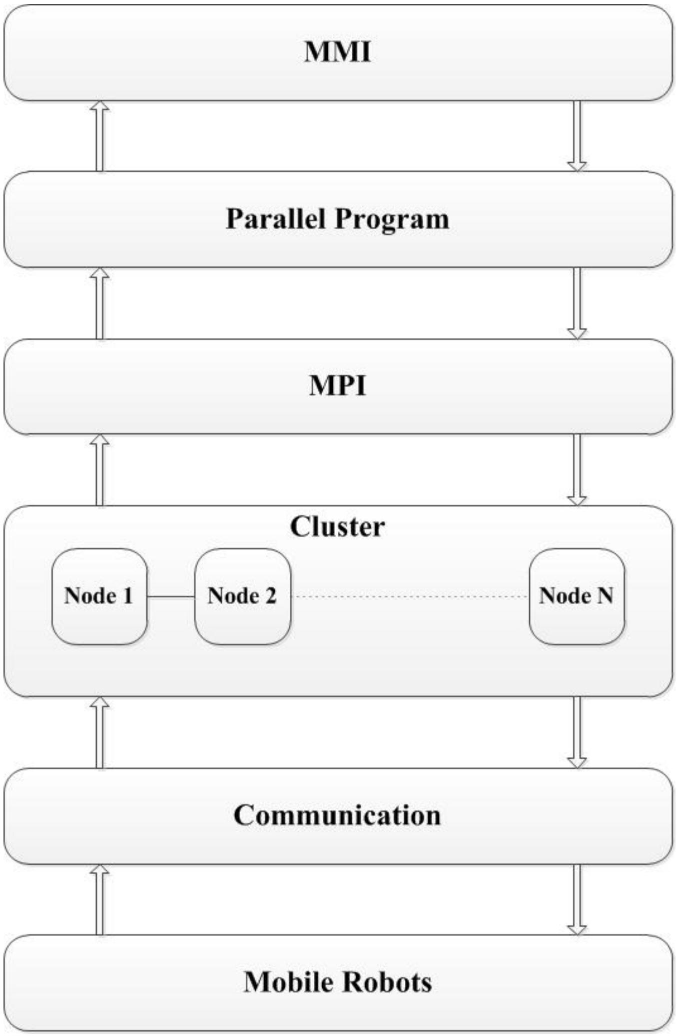

Fig. 5. Layers of the proposed improvement

V. Implementation

A. Virtual MPI cluster

We need at least two machines if we want a real cluster. In our case, we built a virtual MPI cluster based on two Ubuntu [17] virtual machines on a DELL OPTIPLEX 780:

- **Processor:** Intel Core 2 Quad Q8400 @ 2.66GHz
- **RAM:** 4 GB
- **HDD:** 500GB

We installed and configured some packages on the two Ubuntu virtual machines:

- **GNU C compiler and GNU FORTRAN compiler:** to be able to compile parallel programs on our virtual MPI cluster.
- **OpenSSH Server and OpenSSH Client:** to use SSH as the way to communicate between different virtual machines.
- **MPICH [18]:** to compile and run parallel programs on our virtual MPI cluster.

Figure 6 shows the components of our virtual MPI cluster.

B. Cluster configuration

We configured our virtual MPI cluster to be able to communicate with the K-team robot, Khepera III.

The challenge posed by the Khepera III platform concerns the setup of a cross compilation tool chain for the ARM Linux system running on the KoreBot board, i.e. an environment that allows to compile executable programs (e.g. from C or C++ source code) for the ARM Linux system.

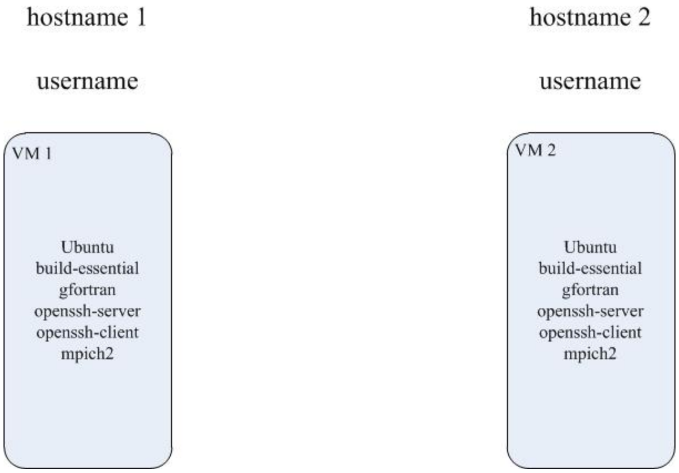

Fig. 6. Virtual MPI cluster

C. Parallel algorithm

We used C language and MPI to program the parallel launch of two Khepera III robots.

The program is executed on our virtual MPI cluster. Each node is responsible of the launch of a robot.

The algorithm pseudo code is demonstrated as follows:

```
BEGIN
        Initialize MPI environment
        IF node1 THEN
                Launch robot1
        ELSE
                Launch robot2
        ENDIF
        Terminate MPI environment
END
```

VI. CONCLUSIONS

The growing need for powerful and cheaper computer processors in the world increases the use of clusters. Today, clusters are used in several areas (commercial, scientific, etc.). The field of mobile robots system, an area in large changes, also needs to exploit the advantages of the use of clusters.

That is why we have proposed in this paper the use of an MPI cluster in the control of a mobile robots system to improve its performances and overcome its limitations.

Creating graphical interfaces for interacting with users and testing the parallel processing of data sent by multiple mobile robots are among the perspectives of our work.

REFERENCES

[1] S. Aydin and O. F. Bay, “Building a high performance computing clusters to use in computing course applications” Procedia - Social and Behavioral Sciences, vol. 1, pp. 2396-2401, Feb. 2009.

[2] G. Bell and J. Gray, “What’s Next in High-Performance Computing”, Communications of the ACM, vol. 45, issue 2, pp. 91–95, Feb. 2002.

[3] J. Dongarra, “Trends in high performance computing: a historical overview and examination of future developments”, Circuits and Devices Magazine, IEEE, vol. 22, Issue 1, pp. 22-27, Feb. 2006.

[4] G. M. Amdahl, “Validity of the single processor approach to achieving large scale computing capabilities”, in *Proc. AFIPS’67*, 1967, p. 483-485.

[5] ARCNET Resource Center. [Online]. Available: http://www.arcnet.com

[6] PVM: Parallel Virtual Machine. [Online]. Available: http://www.csm.ornl.gov/pvm/

[7] Beowulf.org | Mailing List and Archives. [Online]. Available: http://www.beowulf.org/

[8] Home | TOP500 Supercomputer Sites. [Online]. Available: http://www.top500.org/

[9] V. Kindratenko and P. Trancoso, “Trends in High-Performance Computing”, Computing in Sciences & Engineering, vol. 13, Issue 3, pp. 92-95, Jun. 2011.

[10] R. Buyya, “*High Performance Cluster Computing: Programming and Applications”*, Prentice Hall PTR, NJ, USA, 1999.

[11] W. Gropp, E. Lusk, N. Doss and A. Skjellum, “A high-performance, portable implementation of the MPI message passing interface standard” Elsevier Parallel Computing, vol. 22, Issue 6, pp. 789-828, Sept. 1996.

[12] R. C. Arkin, *Behavior-Based Robotics*, The MIT Press, 1998.

[13] K-Team Corporation | Mobile Robotics. [Online]. Available: http://www.k-team.com/

[14] Khepera III. [Online]. Available: http://www.k-team.com/mobile-robotics-products/khepera-iii

[15] A. Sayouti, “Conception et Réalisation d’une Architecture de Contrôle à Distance Via Internet à Base des Systèmes Multi-Agents”, PhD thesis, Ecole Nationale Supérieure d’Electricité et de Mécanique, Hassan II University, Casablanca, Morocco, Jul. 2009.

[16] M. S. Lmimouni, H. Medromi and A. Sayouti, “Utilisation d’un “cluster” HPC dans le contrôle via internet d’un système mobile”, in Proc. JDTIC’12, 2012.

[17] The world’s most popular free OS | Ubuntu. [Online]. Available: http://www.ubuntu.com/

[18] MPICH | High-Performance Portable MPI. [Online]. Available: http://www.mpich.org/