Chapter 1: Introduction

This chapter presents the context in which the present thesis has been developed. First of all, we shall present a short description of numerical applications and the relation between matrix multiplication and other numerical operations and methods for the operations arising from linear algebra in general.

From the processing hardware perspective, computers local networks are described as another alternative of parallel computing, and their relation to other parallel processing platforms currently used is presented in detail. Following the same line, we shall identify with a higher degree of specification the costs associated to parallel computing in the installed computers networks comparing them to other parallel computing architectures.

Finally, a summary of objectives, contributions and the method used in the development of the present thesis is shown, together with an explanation of the contents organization.

1.1 Parallel Applications and Architectures

Parallel processing architectures have been extensively and intensively used in several applications, and the area of numerical problems has been the general starting point from which the possibility of parallel computing has been studied and utilized. Several solution methods have been developed for numerical problems, most of them with a clear orientation to at least two senses [1] [59] [101] [102] [84]:

- Numerical stability when these problems are solved with arithmetic involving some kind of error. The arithmetic usually assumed is generally called of "floating point".
- Direct implementation, or with a minimum cost, into some programming language to be solved in a computer.

From the point of view of someone who has a numerical problem to solve, the use of a computer is nothing but a way to obtain what is needed. Moreover, whether the computer is parallel or not, or the way in which the computer obtains the proper results, is not of much importance either, except for the time needed to use the result. In fact, the term supercomputer or the denomination high-performance computer have been used for a long time depicting this reality: speed is what matters, if it is achieved with parallel processing, then it is utilized [113].

Traditionally, the area of problems arising from linear algebra has taken advantage of the performance offered by the available (parallel) computing architectures. One of the most representative results in these fields might be the researchers' effort to develop a routine library considered of utmost importance. This library was called LAPACK (Linear Algebra PACKage) [7] [8] [LAPACK]. ScaLAPACK (Scalable Linear Algebra PACKage) [21] [27] [29] [ScaLAPACK] was developed as a more specific effort in the context of parallel computing.

Within the scope of linear algebra applications, a set of operations or computing routines has been identified, from which the overall LAPACK can be defined, for instance. Such routines are denominated BLAS (Basic Linear Algebra Subroutines), and both for their classification and their computing and memory requirements identification, these routines have been divided in three levels: level 1, level 2 and level 3 (Level 1 or L1 BLAS, Level 2 or L2 BLAS and Level 3 or L3 BLAS). From the performance perspective, level 3 routines (L3 BLAS) have to be optimized in order to obtain near-optimal performance of each machine, and in fact, several standard microprocessors companies provide BLAS libraries with a strong emphasis on optimization and the rising performance of routines included in level 3 BLAS.

From the very definition of level 3 BLAS routines and, more specifically, from [77], matrix multiplication is considered as the pillar or the routine from which the rest included in this BLAS level can be defined. It may be for this reason, and/or for its simplicity, that most of the research reports in this area of parallel processing starts from the parallel matrix multiplication "problem", and that there exist several proposals and publications in this regard [20] [122] [57] [30] [120] [26]. In other words, by optimizing matrix multiplication, all level 3 BLAS is optimized in some way; and thus, most of linear algebra-based applications depending on the routines optimization carried out by the operations arising

from linear algebra would be optimized. In spite of the fact that this optimization is not necessarily direct, we can assert that the type of processing to be applied to solve matrix multiplication is very similar to the rest of the routines defined as level 3 BLAS, and that it is also very similar to more specific problems solved by linear algebra operations. In this sense, it is very likely that what is done to optimize the matrix multiplication (in parallel or not) will be useful in other operations. In terms of a problem to be solved or a processing to be carried out in parallel, this thesis specifically aims at matrix multiplication with some comments oriented to the generalization due to its importance and representativity in the field of linear algebra.

Figure 1.1 is the shows schematically the relation of matrix multiplication with numerical problems in general. Within numerical problems computationally solved, there exists a well-defined area: linear algebra. As previously explained, a standard library called LAPACK has been defined de facto in this area. A basic set of well-defined computing routines can be used for building all LAPACK: BLAS. This basic routine set has been divided in three levels in order to identify more distinctively which of them is the most appropriate for obtaining the best possible performance in a given computing architecture. Level 3 or L3 BLAS routines are the ones to be analyzed and implemented with greater emphasis in the optimization, and this overall BLAS level can be defined in function of the matrix multiplication.

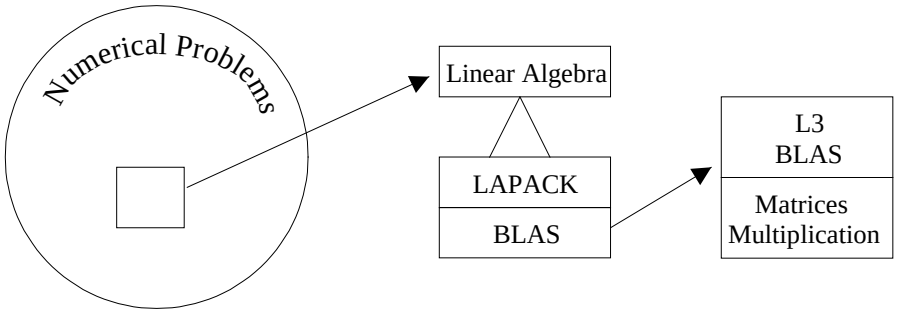

Figure 1.1: matrix multiplication within Numerical Problems.

Sequential computers have several constraints as regards the maximum processing capacity, reason why we resort to the parallel processing in general [2]. In fact, both classical ways of obtaining the best computing performance have been exploited simultaneously and complementarily:

- Increase of a processor sequential computing speed.
- Increase of functional units and / or processors that can be used simultaneously.

Actually, we can assert that the parallel processing raison d'être is the performance.

The field of parallel computing has evolved in several senses, from computing hardware to algorithms and libraries specialized in particular areas. From the processing hardware point of view, there have been great changes as regards design objectives and associated costs. In fact, most of the companies -highly reputed in their times- that create parallel designs and machines have ceased to exist or have been merged to other companies. As an example, we can mention companies such as Cray or Thinking Machines (creator of the Connection Machine). The "great lesson" that can be drawn from this is that the costs associated to a

specific (parallel) computer cannot always be afforded, and there exist alternatives less costly with similar results or, at least, suitable for the problems that have to be solved. The specific (parallel) computer costs are related to [113]:

- Design: The parallel computer is developed in terms of hardware almost from the beginning. Thus, well-qualified and experienced specialists have to be involved.
- Building technology: Both in terms of materials and manufacture mechanisms, the costs are really much higher than those of any massive manufacture and sale computer. The difference is quantified in several extents.
- Installation: Installations both of the physical place and the computer assembling comprise several specific characteristics with a really high cost. This entails, for instance, from the personnel carrying out the installation to temperature conditions.
- Hardware maintenance: It is and has always been a percentage of a machine total cost , and thus is of the same or similar order of magnitude.
- Software: Both the base software and developing or running programs software (which is as or more important) are specific. In this case, the hardware specificity is combined with the complexity of an operating system or a parallel applications developing and debugging environment. In this case, the cost not only involves time and money but also turns errors in the software developed for these computers more likely.
- Operation: Both the software production and the system monitoring and tuning personnel have to be specifically trained.

On the other hand, the processing basic hardware used massively in desk computers has also increased its capacity in orders of magnitude and, at the same time, it has reduced its costs to end users. Both the parallel computers high cost, and the cost reduction and the increase of processing hardware capabilities - which can be called standard and of massive use- have led the current parallel computers to have a strong tendency to incorporate this standard hardware as basics. The benefits have a close relation to the reduction of all the mentioned costs and, besides, it proved to be feasible.

As the number of installed computers in a same office and institutions increased, proposals and the study of problems and solutions related to local networks increased as well. Simultaneously and from various points of views, the existence of a relatively large quantity of network interconnected computers has given place to distributed operating systems and to local networks massive use as useful ways to solve problems in environments regarding users and applications. However, as the number of local networks increased, several possible and low-cost processing alternatives were soon identified:

- Use of the idle periods of local network interconnected machines [90].
- Use of more than one network interconnected computer to solve a problem, making use of parallel computing concepts [9].

From local networks massive interconnection to the Internet, the ideas of "Internet Computing" or "Metacomputing" [24] [13] have also been introduced, and, nowadays, the idea of "Grid Computing" is being introduced as a way of sharing resources in general [54] [75] [53] [55].

The idea of parallel computing in local networks is feasible in some way from the very interconnection of computers. However, it is evident that several problems have to be solved so that this way of parallel computing is used reliably and with an acceptable performance. Even though it is quite difficult to define the term acceptable, we consider

two important underlying ideas to this respect:

- Maximum use of the available resources, especially of the computing resources (processors).
- Processing time necessary to solve a problem.

It is possible that the processing time to solve a problem is acceptable without using the available resources to the utmost; but since computers interconnected in a local network are usually the ones of lowest performance in the market, the likelihood of this is relatively low in general.

A parallel computing alternative with desk standard computers that has proved to be really satisfactory in some areas is the one associated to Beowulf installations [19] [103] [99] [121] [111] [BEOWULF]. Although almost every local network used for the parallel computation can be considered a Beowulf installation [37], there exists certain consensus as regards the fact that:

- Beowulf installation computers are homogeneous PCs, with at least one computer in charge of the complete system administration. In fact, these installations are created for parallel computing and, in principle, it makes no sense installing multiple types of computers (heterogeneous).
- The basic interconnection network is Ethernet [73] of 100 Mb/s, and the wiring should include the use of interconnection switches capable of isolating communications between pair of computers. Anyhow, the best interconnection networks are never discarded though there is a general tendency to the lowest cost.

With the aim of studying and distinguishing different characteristics, multiple processing architecture classifications have been proposed [50] [51] [33]. From the capability and costs points of view, current parallel computing platforms can be organized in descending order in a pyramid such as Figure 1.2. shows.

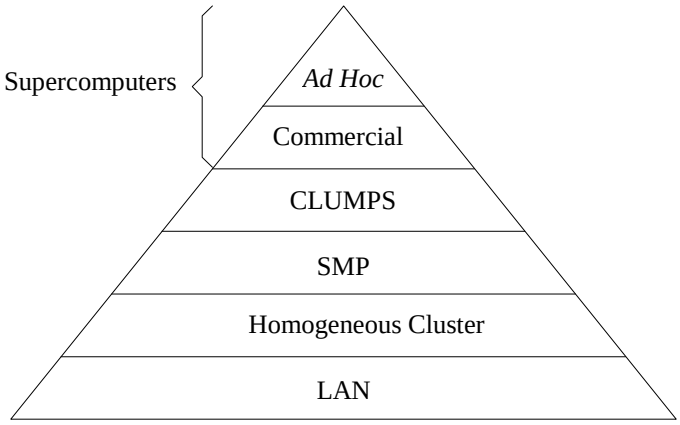

Figure 1.2: A Parallel Computing Platform Classification

Ad Hoc. Parallel computers built under the so-called Accelerated Strategic Computing Initiative (ASCI) program of the USA's Energy Department [61]. These computers are among those with highest absolute computing power, such as reported in [86]. They are the classical parallel computers built ad hoc, and they are called ASCI Red, ASCI Blue-Pacific,

ASCI Blue Mountain, ASCI White. They reach several Teraflop/s and are proportionally costly (actually, they are as costly as the custom-built). In fact, they can be currently considered as the only parallel computers tailored to an application or a set of applications needs.

Commercial parallel computers. They are specifically dedicated to scientific computation, and most of the centers with large computing requirements have one or more of these computers. They are usually constructed with basic hardware promoted as "scalable", since processors, memory, disks, etc. can be added according to the following needs: IBM SP, Compaq AlphaServer, Hitachi SR, SGI Origin, Cray T3E, computers with vector processors (Fujitsu).

CLUMPs: SMP Clusters (Symmetric MultiProcessing) [12] or SMPs network connected do not seem to be explored in general - particularly in / for parallel computing- because it is relatively new or, at least, there are not many publications in this regard. From the parallelization point of view, it is really important the combination of the following models:

- Of shared memory in a SMP that can be considered as MIMD tightly coupled.
- Of message-passing or at least distributed memory MIMD provided by the use of more than one SMP machine interconnected by a net.

SMP: Symmetric Multiprocessing is usually oriented to large data volumes storage and minimum processing in disk and / or "web servers". There are not many reports of its use in scientific applications though there exits a large quantity of publications of its use / utility in the fields of disk-storage data recovery. Many companies promoted them directly as "servers". Its use in parallel applications in general, and particularly in scientific applications, is immediate. Despite having restrictions as regards scalability, and more precisely in relation to the maximum processors quantity that can be handled, they offer the user a relatively large quantity of processing on a unique shared memory, all of which makes multiprocessors-oriented parallel algorithms' use immediate.

Homogeneous Clusters: set of computers dedicated to parallel computing. "Classical" workstations (Sun, SGI), or homogenous PCs. From the technical point of view, they can be considered as Beowulf installations that arose as a set of PCs connected in a local network (generally, Fast Ethernet). From the very start, there was a great investment in the communications network (basically, by means of switches in the Ethernet network wiring). Several tools have been developed both for the administration of the whole parallel system and for the parallel software production. The other homogeneous clusters (with "classical" workstations) do not have many differences with Beowulf installations, but in [99] [111] they are excluded -almost without any doubt- because they do not have Linux as operating system in each computing basic node (in general, it is a PC). The hardware cost is quite higher than that of Beowulf systems, for instance, in relation to the costs between a PC and a Sun or SGI workstation. In terms of performance and reliability, the difference is not so clear-cut.

By the hardware cost relation between a PC and a "classical" workstation, it is very difficult to establish which is the most powerful configuration. In general, with equal cost, the computing capability (or at least, the sum of the processors capability) is greater in

Beowulf systems. However, the difference is even more difficult to quantify and establish a priori when each system tool cost, reliability, etc. are being considered.

LAN: installed computers local area networks that can be used for parallel computing. Each computer has one or one set of users that enable them for parallel computing for certain periods of time. Other alternative in this context is the possibility of transferring a percentage or fraction of computation of each machine. It is very difficult to differentiate the terms used in this context such as "clusters", "NOW" (Networks of Workstations), "COW" (Cluster of Workstations) and "Workstation Clusters". In [12], for instance, these terms are claimed to be synonyms, and the various possibilities are differentiated according the processing characteristics, objectives and base hardware among other indexes. In fact, all throughout the present thesis, this type of parallel computing platform will be referred as "local networks" and also as "NOW" - which is apparently one of the most used names in the bibliography. However, it is important to note that local networks that were not constructed with the specific objective of parallel computing (like in the previous cases of Beowulf systems and clusters) have cost and processing characteristics that are not present in the other ones. This thesis has as its main objective identifying the ways of parallelizing applications or the parallel applications characteristics in order to use to the utmost and / or in an optimal fashion the installed local networks that can be used up for parallel computing.

The definitions and the scope of each term or expression are not well defined. Only as example, note that in [48] [49], every network of network interconnected computers is called cluster and are divided into two general types:

- NOW (Network of Workstations): each computer has one or more users that allow the use of the computer during idle periods.
- PMMPP ("Poor Man's" MPP): cluster dedicated to running parallel applications with high performance requirements.

However, in [12] for instance, it is asserted that all the network computers used for the parallel computing are clusters and are classified according:

- Objective (high performance or availability).
- Users' ownership relation (each computer is dedicated exclusively to parallel computing or not).
- Each node or computer hardware (PC, SMP, etc.).
- Each node or computer operating system.
- Each computer configuration (Homogeneous or Heterogeneous).
- "Clustering" levels.

Thus, the terms hardware, software and the whole network configuration are in some way combined for each characterization. Notwithstanding the use of the terms "local networks" and "NOW", in the present thesis the installed local networks used for parallel computing are considered as:

- Clusters, since they are a set of network interconnected computers used for parallel computation.
- NOW, in the sense that each computer has a user or set of users who transfer it for parallel computation.
- PMMPP, since the computer is completely available for the time it is used. Even in the case of having a fraction of each computer, this fraction is always available.

- Heterogeneous, since it is really difficult that, in the installed local networks, all the computers have the same hardware and the same base software. This is mainly caused by:
  - ✗ The objective of each computer or the usual applications that it solves. The existence of each machine is caused by a user or a set of applications that have to be solved more or less periodically. Neither the users nor the applications have the same computing, storage, etc. requirements. Thus, the existence of each computer is not generally related to or has little to do with the others (at least, as regards physical characteristics)
  - ✗ The network installation time. The addition of new machines to the local network as well as the repair and up-date of existing machines -which can be called network evolution- makes it really difficult to keep homogeneity. Local network evolution is as unquestionable as the relation between evolution and heterogeneity. In general, the longer the local network installation time, the greater the heterogeneity. Only as example, we can mention that the availability of a type of computer is quite limited. In the case of PCs, this availability time is usually counted on months and, may be, on more than two years at the most.

It is really interesting to estimate the homogeneous clusters evolution, which have been installed relatively for a short period of time and are currently in production. Although the applications and users are well controlled and managed, failures likelihood in the hardware is still present. And, the larger the quantity of computers and hardware in general, the higher the likelihood of failure. Thus, the repair and / or replacement is not something very unusual nor it will be in homogenous clusters. When it is impossible to keep a cluster homogeneous due to the availability of hardware to be repaired / replaced, there are two alternatives (discarding manufacture ad hoc, whose cost is extremely high):

- Keeping homogeneity and not making any repair nor replacement. It is valid only if the applications are still being solved. Considering that the applications generally tend to increase their requirements, this alternative seems to be of little use.
- Repairing and/or replacing with available software. The cluster will automatically turn into heterogeneous. Heterogeneity is directly translated into different values for the performance metrics when we are dealing with processors and / or memory in the case of intensive computing tasks.

Other reason why a cluster cannot be necessarily kept homogeneous is the need to solve applications with greater computing and / or storage requirements. In this sense, as time is running, hardware availability is lower and, thus, if the homogeneity is to be kept or if the applications are to be solved with homogeneous hardware, a new cluster should be installed. The cost is that of the acquisition and installation of a whole cluster, in addition to a decision to be made regarding the homogeneous cluster no longer capable of solving the applications. This alternative seems to be the most logic (i.e., the installation of the necessary hardware in order to increase the cluster total capacity) generally implies heterogeneity.

As from the fact that it is really difficult to keep homogeneity even of the clusters dedicated to parallel computing, every contribution made in the context of the heterogeneous computers networks has a very wide scope of use. In fact, each time that - whichever the reason - a cluster "turns" heterogeneous, the applications and users will have to be

compliant to -in some way or another- the hardware in order to obtain an optimal performance (that, as stated before, is the main reason for the very existence of the parallel computing).

1.2 Parallel Computing Costs in Installed Local Area Networks

As previously stated, installed local networks are the "zero-cost" parallel computing platform as regards hardware. In fact,

- installation
- managing and monitoring
- maintenance

costs of each computer are not related to parallel computing, and thus they are not to be assumed by those who use them for parallel computing since

- local networks are already installed and functioning, and the parallel computing objective is not among their objectives (at least, among most of the installed local networks). The fact that they are used for parallel computation does not change this reality, though it is in some way added as one of the uses of local networks;
- each computer and the very local network have at least an administrator in charge of the configuration, interconnection and a minimum performance control;
- each computer has at least a user or set of users in charge of the maintenance. Also, these users are usually in charge of the hardware management and/or up-date, thus having hardware updating zero-cost.

But hardware cost does not represent all the cost related to the installed local networks processing that can be used for parallel computing. Costs to be assumed in this contexts are the ones related to:

- local network administration tools for parallel computing and for parallel programs development and execution;
- the availability of local network computers and the same local network;
- applications parallelization.

Parallel programs development and execution tools are essential to utilize local networks for parallel computing. What was first developed in this sense were message-passing libraries such as PVM (Parallel Virtual Machine) [44] - established as a standard de facto in this area. Then, the standard MPI (Message Passing Interface) [88] [107] [92] was proposed and they are both currently used, with the tendency of a new parallel applications development in MPI implementations available for the parallel architecture being used. Both libraries are implemented specifically for computers networks, their use is free [PVM] [LAM/MPI] [MPICH], and they are obtained via Internet. In this sense, the only cost of these tools is that of their installation. Though not so focused or standardized as PVM and MPI, multiple computer networks administration tools have been developed for parallel computing, and they are also available in Internet. Beowulf installations are one of the main tool sources for the networks administration used for parallel computing and that can be used up. However, there exists a consensus in that the administration cost of a

computers network used for parallel computing is relatively higher than in the case of classical parallel computers [17].

The installed local network computers availability for parallel computing is not complete. In spite of the fact that this cost is very difficult to quantify, the use of local networks has several alternatives, two of which are:

- Almost idle periods, such as nights or holidays. Notwithstanding that it depends on the specific characteristics in each local network - even during rush hours - computers are not necessarily used to the utmost [90].
- Determining *a priori* a percentage of each computer in order to be used for parallel application processes. In this case, it is like having the same quantity of computers though with less capability.

From the parallel applications point of view, both alternatives are similar: there exists a set of available resources. This is the "availability" context of the computers that will be used. More specifically, the experimentation will be carried out during periods in which local networks are not used with no other objective, and thus, the availability will be total (during the execution of parallel programs).

The most considerable cost - or at least the most unknown one - might be that of the parallelization and/or development of parallel programs for installed local networks whose resources can be used up. The greatest problem in this context is the very same parallelization. It has always been considered as one of the greatest problems (and with its associated cost) since there does not exist general methods. One of the main reasons for this is the raison d'être of parallel computing: the performance. Although syntactic, semantic and stylistic parallel algorithms can be very well designed, they are not useful when they do not get an acceptable performance. In general, it is really difficult to quantify the term "acceptable", but two possible senses can be mentioned:

- They use the available resources to the utmost. In general, the importance is focused on the use of available processors.
- Acceptable response time. In this case, it is independent on the application. In the case of parallelizing the time-state prediction task for a certain day, it would be clearly inadequate to obtain the prediction response one or two days later.

Several performance metrics of parallel systems are focused on the use of available resources since:

- It is independent of the applications and, in this sense, the metrics are general.
- If the available resources are used to the utmost, it is assumed that nothing better can be achieved at the very least in the parallel machine with the parallel algorithm used.

However, as stated before, several of the linear algebra problems have been successfully solved with parallel computers. Then, one of the first tasks to be carried out is the review of the already developed parallel algorithms in order to use them up in the computers networks context. At this point, we must not oversight the fact that computers networks, and more specifically each computer that can be connected to a network (with a network interface), has not been - and is not - designed for parallel computing distributed in multiple machines. Thus,

- at the very least, the proposed parallel algorithms and their efficiency must be analyzed

in terms of their performance in the computer networks;
- if there is no parallel algorithm apt for parallel computing associated to a problem in the network, another should at least be proposed in order to consider the efficiency of this type of parallel architectures.

Although the proposed parallel algorithms analysis must be carried out exhaustively and, maybe, "case by case" (or at least, by area of applications or processing characteristics), there exists a drawback from the very beginning: parallel machines have been and are designed for parallel computation and computer networks are not. Thus, it is evident that parallel algorithms are not directly usable in local networks that have been installed and are used with several purposes - among which parallel computing is not found or is not one of the most important ones.

Application parallelization for this type of parallel architecture have several drawbacks, or at least, several characteristics unknown in traditional parallel machines. The most important drawbacks are:
- Computing nodes heterogeneity.
- Interconnection network characteristics.

Both traditional parallel machines and homogenous clusters (as characterized before: built for parallel computing) have homogeneous processing elements (processors). This considerably simplifies the computing load distribution, since in order to achieve processing load balance, it is simply necessary to distribute the same quantity of operations to be executed by each computing element. In general, the definition of the term simply is not trivial though, in the context of linear algebra applications, it normally implies the distribution of the same data quantity (or equal portions of matrices data) between the processors. Thus, the parallelization of the applications arising from the linear algebra fields has not had any problem associated to the load balance in homogeneous parallel computers. In local networks used for parallel computation, it will be evidently necessary to solve the problem caused by the differences between the computing capabilities of the various machines used.

The most extended interconnection network as regards installed computers local networks is - without doubts - that defined by Ethernet standard [73] of 10 Mb/s and of 100 Mb/s. In fact, 10 Mb/s Ethernet network is assumed to be not apt for parallel computing [91]. Apart from maximum performance characteristics, Ethernet networks have several drawbacks due to its own definition and dependence on CSMA/CD protocol (Carrier Sense-Multiple Access/Collision Detect), which implies a performance generally dependent on the individual traffic of each of the interconnected machines. However, the huge computing power installed in the local networks turns worthy any contribution with this respect. On the other hand, the applications arising from linear algebra have a relatively large quantity of potential users and of local networks available for parallel computing. From the cost point of view, Ethernet networks not only are the most spread - and thus useful as regards current installations-, but also have great inertia as regards new installations since
- there exists a large quantity of qualified and experienced technical staff that can keep on installing these networks. Changing the technology generally implies higher costs of, at least, technical staff's training;
- there exists a great quantity of developed software that generally tends to be stable and

used;
- Ethernet standard has been defined with higher data transference capabilities, such as 1 Gb/s (10 6 bits per second) [76] and 10 Gb/s (10 7 bits per second) [110].

From another point of view, there are other associated costs that are not very much related to the strictly technical ones - such as the already mentioned. As stated in [18] (in the context of providing high throughput) the use of local networks is a technological and sociological problem. In this sense, beyond the existence of an "evangelist" (as denominated in [18]) that develops and creates with his own resources and with those of "allies" a high throughput computing cluster, and with it "generates demand" towards a wider set of potential users that, in turn, will contribute with their own individual resources, it is true indeed that all local network or set of local networks must have an explicit support of an organization in order to enable its use for parallel computation. However, this thesis will not deal with this/these cost/s.

1.3 Summary of the Thesis Objectives and Contributions and Organization

The main objective of the present thesis is the evaluation of problems and solutions for the parallel computing on the installed workstations networks, with their computing and communication characteristics. The problem through which the evaluation is carried out is matrix multiplication for several reasons:
- Similarity of the processing type as regards the rest of the operations (and applications) arising from linear algebra.
- Similarity in terms of computing and storage requirements, specifically in relation to the routines included in BLAS level 3.
- Quantity of publications identifying both parallel algorithms proposed and performance attained.

All the thesis aims at the possible maximum attainable performance, thus, although we will initially use standard tools highly employed in this context (such as PVM), the tendency will be upgrading and / or replacing everything that may impose an excessive penalty in the performance. This encompasses from the way to carry out local computing to the used communications routines monitoring and evaluation.

In principle, the method used to carry out the evaluation consists of two parts:
- Analysis of already proposed algorithms, methods and tools in order to carry out parallel processing. If this analysis implies an *a priori* clear penalty as regards throughput, at least a suitable alternative will be proposed.
- Exhaustive experimentation. A set of experiments to be carried out will be defined and the validity or not of the obtained performance will be analyzed.
- In case the obtained performance is not satisfactory, the source of the performance penalty will be identified, and a solution alternative will be proposed, with which the proposed experiments will be analyzed once again.

The contributions can be summarizes as follows:

- Identification of the utilization level of the parallel algorithms specifically proposed for matrix multiplication.
- Identification of the parallel processing characteristics with which the matrix multiplication are to be specifically solved.
- Identification of the parallel processing characteristics with which applications arising from linear algebra in general are to be solved.
- Identification and proposals of solutions for potential performance problems caused by both each computer local performance and the communications routines used.

The use of these contributions tends to

- utilize the available computing capability in installed local networks, despite not having parallel processing as objective.
- utilize current installation of homogeneous clusters used for parallel computing when, due to the evolution in the hardware and /or applications, are heterogeneous.

More specifically, throughout this thesis the following points are described:

1. From the analysis of matrix multiplication parallel algorithms, it will be clear that they are not directly usable in the parallel computing platform which represents computers local networks.
2. Even when a specific algorithm of matrix multiplication is proposed in order to use to the maximum the characteristics of the parallel computing architecture provided by local networks, the optimized performance is not assured. By way of experimenting and monitoring (via instrumentation) the algorithm performance, it will be shown that the PVM message passing library imposes an unacceptable performance penalization.
3. The reasons for which general purpose message passing libraries (among which the very PVM library and MPI implementations are included, for instance) cannot assure an optimized performance in those local networks of computers used for parallel computation are analyzed and briefly explained.
4. A single message passing operation, which is directly oriented to the use of the Ethernet networks characteristics used for computers interconnection in local networks, is proposed and implemented.
5. The performance of the proposed algorithms is evaluated by means of experimentation, including the proposed message passing operation, showing that the performance can be considered as acceptable or optimized for the local networks of those computers used for parallel computation.
6. It is also shown how one of the proposed algorithms for matrix multiplication can be used to evaluate the computers local network machines' capacity of carrying out background communications (in an overlapped fashion) while the local computation is being carried out.
7. Finally, the algorithmic proposal of this thesis is compared to the algorithms particular to ScaLAPACK library, which is considered as the one that implements the best algorithms (in terms of performance and scalability) of parallel computation in distributed memory parallel architectures. In this specific comparison case, only the homogeneous computers networks are taken into account, since ScaLAPACK does not have any estimate for the case of architectures with heterogeneous computing elements. This comparison also renders favorable performance and scalability results, at least up to what was evaluated in terms of quantity of machines used for parallel computation.

The following chapter, **Chapter 2: Matrix Multiplication**, addresses the analysis of the matrix multiplication in general - in the context of linear algebra operations - and the analysis of parallel algorithms proposed.

In **Chapter 3: Matrix Multiplication in Heterogeneous NOW**, the installed local networks characteristics are analyzed in detail from the parallel computing point of view. According to this analysis, we identify the principal characteristics of parallel computation in this processing platform and two specific algorithms are proposed for multiplying matrices in parallel.

Chapter 4: Experimentation, presents in detail the experiments carried out and analyzes the results obtained with the PVM message-passing library. It shows in detail the performance problems generated by this library in particular, and presents some comments on the expectations regarding the other message-passing libraries for computer networks. A communications routine is proposed; this routine, when remaking the experiments, shows how it is possible to achieve the maximum or optimized use both of the computing and communications available resources, this combination thus providing a satisfactory performance.

In **Chapter 5: Comparison with ScaLAPACK**, two important aspects in terms of the validity and use of this thesis' contributions are presented: 1) Application of parallelization principles in homogeneous environments dedicated to parallel computing for two specific cases: matrix multiplication and matrix LU factorization, and 2) Comparison of the results obtained by experimentation in terms of performance with respect to the ScaLAPACK library. This library is specifically dedicated to homogeneous parallel computing platforms with distributed memory, and it is generally accepted that implements the best existing parallel algorithms in terms of performance optimization and scalability.

Chapter 6: Conclusions and Further Work, summarizes the principal conclusions as from the analysis and experimentation task carried out. It also presents an estimation of the principal research lines followed from the work of the present thesis.

Next, reference details are presented in the **References**, and Appendixes are included. In general, the idea of the appendices is that they are self-contained, and for this reason they include their own bibliographical reference list.

Appendix A: Local Network Characteristics, shows in detail computer and local network wiring hardware used for the experimentation.

Appendix B: Computer Sequential Processing Performance, shows the impact of the processing code optimization levels in each of the computers, its maximum performance (utilized in the parallel processing) as well as some comments on the code that is and should be used in general in the experimentation with/in the parallel programs production.

Appendix C: Communications in the CeTAD Local Network, shows point-to-point performance of the communication in the CeTAD local network (one of the three used) and also of the communications with broadcast routine provided by PVM library between

parallel application processes. Some comments are also mentioned as regards the communications performance in the other local network used, with some references to other message-passing libraries available for parallel computing in computer networks.