The IST Cluster: an integrated infrastructure for parallel applications in Physics and Engineering

M. Marti, L. Gargaté, R. A. Fonseca[1], L. L. Alves, J. P. S. Bizarro, P. Fernandes,
J. P. M. Almeida, H. Pina, F. M. Silva, L. O. Silva

Instituto Superior Técnico, Lisboa, Portugal
michael.marti@ist.utl.pt

Abstract. The infrastructure to support advanced computing applications at Instituto Superior Técnico is presented, including a detailed description of the hardware, system software, and benchmarks, which show an HPL performance of 1.6 Tflops. Due to its decentralized administrative basis, a discussion of the usage policy and administration is also given. The in-house codes running in production are also presented.

1 Introduction

Instituto Superior Técnico (IST) has a long tradition in the deployment of computers to support scientific computing. Amongst the very first computers installed at IST was the mainframe IBM 360/44 [4]. This machine was fully dedicated to scientific computing and was still in use as late as 1981. Between 1983-1985 a first computer cluster of four VAX 11/750 nodes was installed, with the first network on campus [27, 24]. In 1987, IST received the first network connection to the "outside world" enabling communication with other research institutes throughout Portugal. With e-mail since the late 80s and internet access since early 90s the internet era commenced enabling advanced computational concepts, such as distributed computing and grid computing.

Currently the most powerful computer installed on campus is the IST Cluster [23]. Resulting from a partnership between Instituto de Engenharia Mecânica (IDMEC), Instituto de Plasmas e Fusão Nuclear (IPFN), Instituto de Engenharia de Estruturas, Território e Construção (ICIST), Centro de Estudos de Hidrossistemas (CEHIDRO), and Centro de Informática do IST (CIIST), this system supports research at IST, taking advantage of the expertise and the smaller clusters deployed all over campus.

This paper reports on the key points of the installation, and operation of the IST Cluster, as well as the main applications/problems being presently pursued with the resources of the IST Cluster. In Section 2, an overview of the hardware architecture of the cluster is given, including the computation nodes, network interconnect and support machines. Section 3 is dedicated to system software. Section 4 describes the usage policy and the administrative model in detail, which takes a very decentralized approach. Section 5 introduces the most important production

[1] Also at ISCTE, Lisboa, Portugal

software currently deployed on the system. The several benchmarks that were conducted to test the machines functionality and performance are discussed in Section 6. Section 7 describes the planned integration of the IST cluster with other three large high performance computing (HPC) infrastructures in Portugal in the framework of the Rede National de Computação Avançada (RNCA). Finally, Section 8 gives an overview of planned activities and planned upgrades for the IST Cluster.

2 Hardware

The IST Cluster is based on the IBM power architecture in blade form factor. At its core, it consists of 5 blade center chassis with 14 dual-CPU, dual-core JS21 blades each. These computing nodes are interconnected by two Gigabit Ethernet fabrics. Three main servers are also present for user access and management, file access and web portal.

Each of the JS21 [16] computing nodes hosts two dual-core PowerPC 970 CPUs clocked at 2.5 GHz with 32 KB (data) + 64 KB (instructions) L1 cache and 2 MB L2 cache. The frontside bus is 2x32 bit at 1.25 GHz per CPU. The memory controller, contained in the north-bridge, has two 400 MHz DDR2 memory channels, allowing for a maximum memory bandwidth of 6400 MB/s. Each node is equipped with 8 GB PC2-3200 CL3 ECC DDR2 of memory and 73 GB of serial attached SCSI (SAS) disk storage.

There are two separate Ethernet fabrics connecting with the two interfaces on each node. One fabric is used for inter-process communication and the other fabric is used for disk I/O and management traffic. The communication network has a flat structure: each of the 70 blades connects directly to the central switch which consists of two stacked Force10-S50 switches [6]. The management network has a two-layer topology: each blade center contains a 14-to-6 Nortel switch [15], reducing the 14 Ethernet channels from the 14 blades to a bundle of 6 channels. The 6 channels on the secondary side of the Nortel switches are bundled via channel bonding and connect to the remaining ports of the second Force10-S50 switch (virtual LAN). With channel bonding of six Gigabit Ethernet channels, one can expect at least an overall bandwidth of 6 Gb/s which is sufficient for disk I/O.

The three auxiliary machines available for the cluster are used as master (front-end) node, file-server, and web front-end. The master node is an IBM p505 [19] with a dual core 1.9 GHz POWER5 CPU, 36 MB L3 Cache, and 1 GB of memory. This machine is used as a management node. It runs the resource-manager, scheduler, and monitoring systems. User log into that master node via ssh to have Unix-level access to the cluster.

The storage node is an IBM p510 [20] with a dual core 1.9 GHz POWER5 CPU, 36 MB L3 cache, and 2 GB of memory. Further, it contains two fiber channel host-adaptors (FSC 4 Gbps) to connect to the storage system. This machine acts as a file server for both the terabyte-scale storage system and the parallel filesystem. Users do not have direct access to this node.

The server for the web front-end is an IBM x306 [22] with a dual-core Intel Pentium 830 CPU at 3.0 GHz and 1 GB of memory. It hosts the general webpage, an interface to manage jobs online as well as the Ganglia [25] status pages.

The cluster contains an expandable FSC to SATA raid system from IBM called DS4700 [17]. It currently contains 16 drives of 500 GB each, allowing for a total net storage of 7 TB. This storage is used as temporary transit space for job output with a strict system of disk quota and automatic removal of old files.

3 System Software

The operational software manages the computing resources and presents them accessible to the end user. Operational software includes operating system, hardware-management software, monitoring tools, compilers and debuggers, HPC libraries, and job queueing and scheduling systems. Table 1 lists the major software packages and the version numbers available on the IST Cluster. A brief explanation of the key options is given below.

One of the fundamental decisions is what implementation of clustering tools should be used to deploy and to manage the system. Several well-known solutions are available, such as Beowulf [31], Rocks [26], and Gluster [32]. In the case of the IST Cluster, it appears consistent to choose the IBM solution that is available free of charge under the IBM Academic Initiative. This has the advantage that it tightly integrates with the IBM blade center facilities and the other hardware components of the cluster. In addition, it leaves the free choice between Linux and AIX and it allows the two operating systems to run at the same time. The actual packages used to manage our cluster are IBM Network installation manager (NIM) and IBM cluster systems manager (CSM). These software solutions contain utilities for automated remote installation of the computation nodes, parallel execution shells for synchronized node administration, config-file management tools, and more.

For computing nodes with local hard disk, it is possible to install and to boot the operating system locally. This is preferable due to the reduced network traffic during boot and system disk access, while not posing an extra burden on the administration thanks to CSM and NIM. We tested the two main choices for the

Cluster management systems	
IBM Cluster Systems Management	1.6
IBM Network Installation Manager	2r1
operating systems	
IBM AIX	5.3
Suse linux enterprise server	10.1
queuing system	
Moab	5.1.0-p8
TORQUE	2.1.9
system monitoring	
Ganglia	3.0.5
IBM Director	5.10 u2

compilers	
XLC	8.0
XLF	10.1
gcc	4.0.0
parallel libraries	
IBM POE	4.3
MPICH2	1.0.6p1
LAM	7.1.4
OpenMPI	1.2

Table 1. Major software packages and version numbers installed on IST Cluster (February 2008).

operating system (AIX vs. Linux) using the production code OSIRIS and found that AIX exhibits a performance penalty lower than 4%, while providing high security and stability, leading to the decision to install AIX 5.3 (00003222D100) on the majority of the nodes. In addition, four nodes are running SuSe Linux enterprise server 10.1 (SLES, 2.6.16.21-0.8-ppc64) to run third-party applications that do not support AIX on Power. The head node and the storage node run the same version of AIX, whereas the web-fronted node uses SLES 9.

As a directory service, we are using LDAP v3 with IBM's LDAP schema. This allows for a centralized management of user accounts, mount-points, and many other aspects of the machines administration. The choice of LDAP over other directory facilities, was made to guarantee the future integration of the IST Cluster into RNCA, and into the central authentication system of IST.

As with any shared resources, it is crucial to have the right tools to regulate and facilitate the coexistence of users. Several approaches for resource management and job queuing were considered, including Sun Grid Engine, Maui / PBS, and Moab / TORQUE. The IST Cluster is currently running Moab [2] as the scheduler and TORQUE [3] as the resource manager. This approach has two major advantages over the other solutions: (i) the scheduler has many advanced features like backfill, fair share, extended reservations and accounting, etc, and (ii) Moab allows for simplified grid-like integration of several Moab or even Globus [7] driven nodes.

For HPC tasks on the IST Cluster, the IBM XL compiler suite is available - comprising the XLC c/c++ compiler, the XLF Fortran 77/90/05, and the AIX binary tool chain. This compiler produces the best optimized binaries for the Power architecture (although at the expense of a rather slow compiling). Alternatively, the gnu compilers and tool-chain are available as well [10].

IBM has its own implementation of MPI as part of the IBM Parallel Operating Environment (POE) [21]. It is tightly integrated with the XL compiler suite. For test purposes and compatibility reasons, we also installed and tested other MPI flavors (cf. Section 6) MPICH2 [11], LAM (legacy) [1], and OpenMPI (currently unsupported on AIX) [8].

The widespread network file system NFS has well-known scaling problems when used in conjunction with a large number of nodes (clients). To overcome this issue, all networked filesystems that are used for disk I/O of the production jobs are mounted via IBM's general parallel filesystem (GPFS)[18].

The local hard disk in each computing node is used as a system disk for the operating system and as a local, temporary scratch space for the production jobs. In addition, each of the 66 local AIX disks has a certain amount of disk storage reserved for a global storage pool. GPFS is capable to merge all these blocks of distributed disk storage into one large virtual filesystem which in turn can be mounted on each node. This approach has two advantages: (i) it helps harvesting unused disk space and, (ii) it combines the flexibility of a networked filesystem with the performance of local disks.

4 Usage Policy and Administration

The IST Cluster is shared, in equal parts, by four partners IDMEC, IPFN, ICIST / CEHIDRO, and CIIST . This flat, non-hierarchical structure must be reflected in the usage policy and in the administrative structure.

As a first measure, there are four Unix-level user groups which serve as primary groups for user accounts. These groups correspond directly to the four partners of the IST Cluster project. Each user account is associated with one of the four Unix groups, depending on the partner the corresponding user is associated with. Much of the usage policy and administration measures described in this section are based in this flat hierarchical structure.

One of the most important issues related to usage policies of HPC systems is the distribution of available CPU hours amongst the users. Each of the usergroups of the IST Cluster is associated with similar shares of CPU time. Organization and further distribution inside a user group is done via logical entities called projects: CPU time belonging to a group is subsequently distributed to these projects. Each user can belong to one or more projects. When running jobs on the cluster, the user must specify which project should be used to charge the consumed CPU hours. While possible and readily available, we do not currently deploy a system to regulate resource consumption among users within the same project.

There are several job queues avaliable in the queuing system of the cluster. These queues are different from each other in terms of job start priority and relative cost factor. The job start priority controls the relative order with which jobs from different queues will be assigned to computing resources. The relative cost factor maps ”effectively consumed CPU time” to ”charged CPU time”. This way, a job in a high priority queue will run earlier, but will consume more CPU time from the associated project. To ensure that the different users have the same chances at getting their jobs done within a given time a fair share system is in place. This system dynamically reduces the job start priority of jobs for users and groups that have a high values of effectively consumed CPU hours for the past 14 days. This prevents users or groups from temporarily harvesting all the resources by submitting a large number of jobs to the queue. Once a project consumes all its CPU hours, users submitting jobs under that project can only submit to a special queue, the drain queue. Jobs in the drain queue have no priority associated, meaning that such a job can only run if all other queues are empty or if the job requested can run without affecting start time for all the jobs currently in the queue. This gives projects with no CPU hours left the opportunity to still run jobs, while reducing the chance to have the system idle, thus guaranteeing an efficient use of the resources.

Mediation among the users is also required for disk storage occupation. Each user has access to a relatively small amount of storage on his home folder, to store codes and related information, but not simulation results or bulk data. There is a soft and hard quota in place for all file systems accessible by the end user of the cluster. This is particularly important for the two large file systems intended for simulation results and bulk data. These file systems are protected with group level quotas. It is also possible to define quotas for these file systems on the user level; however, we have not found this to be necessary so far.

Strictly enforced usage policy rules have the advantage of a fair resource distribution among all the users, but can inhibit users from doing certain tasks like testing, monitoring and run time debugging of their codes. To overcome this limitation, it was decided to reserve two of the 70 blades for such tasks. These two nodes have a more permissive usage policy, enabling the users to do tasks that are not possible on the other nodes, while not disturbing the normal operation of the rest of the machine. This gives the user an authentic computing node environment for compilation of custom codes, and for testing and debugging purposes. While it is still possible to access these two nodes via the compilation queue, it is also allowed to directly login to these nodes. Multiple users can use these nodes at the same time.

In order to be able to delegate the several administrative Unix operations, in a decentralized environment, we have created a set of Unix users which have the necessary privileges to do certain administrative tasks. Each user group and all the major software packages have an associated administrative user. These administrative users do not have a password and, therefore, it is not possible to login directly using the credentials of such a user. Nominated persons are allowed to change user privileges to such an administrative user via sudo facility. The administrative user, in turn, does have the rights to execute certain scripts with root privileges enabling that person to do administrative activity. This two level authorization model used for user, software, and job administration, has a decentralized structure, while avoiding sharing sensitive passwords.

We have created a large number of scripts for the group administrators to take over most of the daily tasks. For user administration there are scripts to create, to modify, and to remove user accounts as well as scripts to move files belonging to users under that group. For the queuing system and job administration there are scripts to create, modify, and remove projects, scripts to increase and to remove CPU hours from projects and all the necessary tools to manage jobs belonging to users in the group.

To manage software installations in a decentralized manner there is a Unix group for each software package in addition to the software administrative account. Each software package is installed in its own specific folder. The privileges of this software folder and all files beneath it are such that the software administrator has read, write, and execution permissions, while only members of the associated software group have read and execute permission. That way it is possible to control who has access to a certain software package by managing the membership of the associated software group. The administrative users for a software package have the means to add and to remove users from their software group.

5 Production Software and Research

The IST Cluster was first open to the public in January 2007. Since that time there has been a large number of users working with the system. Most of the users are affiliated with IST, but there are also accounts of researchers from Universidade

Nova de Lisboa, Universidade da Beira Interior, and Universidade Federal de Santa Catarina, Brasil. Currently, there are a total of 51 active users (February 2008).

Many of the parallel, scientific codes employed by the users are their own, in-house developments. Figure 1 shows some samples of plots and visualization of calculations currently running on the IST Cluster. Some of the codes on the IST Cluster are:

dHybrid — a 3D massively parallel kinetic ion, massless fluid electron code [9]. The code is used in problems where the kinetics of ions is of importance and where large time scale and large spatial scales need to be resolved, as is the case of many space plasma physics problems.

HOMS (Hierarchical Optimization for Materials and Structures) — three-dimensional model for bone remodeling developed taking into account the hierarchical structure of bone. On a global scale the bone is assumed to be a continuum material characterized by equivalent mechanical properties and, on a local scale, a locally periodic cellular material model approaches the bone trabecular architecture in terms of its mechanical properties. For each scale there is a material distribution problem governed by density based design variables which at the global level can be identified with the bone relative density.

OSIRIS — State of the art, massively parallel, electromagnetic fully relativistic Particle-In-Cell (EM-PIC) code for kinetic plasma simulations [5]. The particles are pushed using the interpolated electric and magnetic fields at particle positions, and the full set of Maxwell's equations are solved using a FDTD algorithm using the electrical current from particle motion. Applications include astrophysical shocks, ultra-intense laser plasma interactions, and fast ignition of fusion targets.

pjet — pseudo-spectral code for direct numerical simulations of plane jets. Its parallel strategy is based on MPI.

PPrimProjAlgD_RandInitCond — code to numerically solve a very large set of randomly generated Pareto eigenvalue problems of the type $\mathbb{R}^n_+ \ni x \perp (A - \lambda B)x \in \mathbb{R}^n_+$ by the primal projection algorithm. It is used to characterize the solution of contact problems.

QuickPIC — a fully relativistic, massively parallel PIC code, specially adapted for the modeling of plasma and laser wakefield acceleration [13, 14]. In QuickPIC, the plasma response is solved through a simplified set of Maxwell's equations, which are written under the quasi-static approximation. In addition, the laser evolution is determined through the ponderomotive guiding center approximation.

SUBLIM-3d (Strict Upper Bound Limit Analysis Code) — code for the computation of strict upper bounds of the collapse loads of rigid-perfectly plastic mechanical systems. It is applied in soil mechanics to study the stability of slopes and excavations.

TRI (Turbulence-radiation interaction) — code for the calculation of radiative heat transfer in turbulent flows, which allows to obtain fundamental insight on the TRI, by using data given by direct numerical simulation or large eddy simulation of turbulent flows. The physical analysis of the results obtained with this code gives important information for the physical understanding of TRI and for modeling purposes.

Furthermore, there are several commercial codes installed and running on the IST Cluster, including ABAQUS and ANSYS.

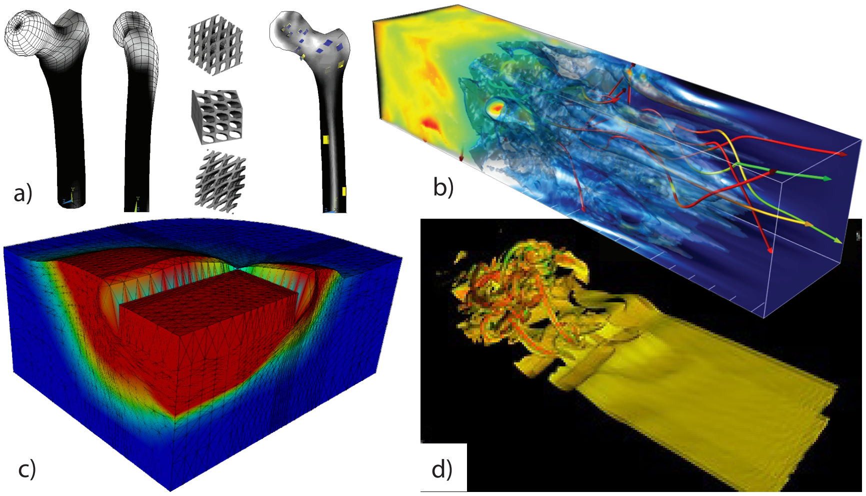

Fig. 1. Collections of diagrams and visualisations of data obtained from codes developed at IST and deployed on the IST Cluster: a) HOMS, b) OSIRIS, c) SUBLIM-3d and d) pjet.

6 Benchmarks

System tests and benchmarks were very important in two different stages of the IST Cluster project. Benchmarks conducted on test systems of the different hardware vendors helped us to compare real live performance of proposed systems. In the early deployment phase of the machine, benchmarks were crucial to test the correct functioning of all systems.

We used the **network benchmark** utility NetPIPE [30] to measure the performance of the network cards on the nodes. NetPIPE is a ping pong type of network test, which can leverage on different protocols, (TCP, MPI etc.) to measure network performance between 2 peers. For the management network we determined the latency to be $63\mu S$ (small packages) and the bandwidth to go up to 978 Mbps. For the communication network we determined these values to be $58\mu S$ and 980 Mbps respectively. It is worth noting that the second stage of switches in the management network increases latency minimally while yielding almost the same bandwidth for point-to-point communications.

In order to compare the **different MPI flavors** and to determine what flavor should be used for production, we ran NetPIPE in MPI mode for all available flavors. We found that the IBM implementation of MPI yields the best results, superseded only by the OpenMPI implementation thanks to the capability of

OpenMPI of channel bonding. In Figure 2, we present the NetPIPE results for latency and bandwidth comparing POE MPI with OpenMPI. It is clearly visible that OpenMPI uses a threshold for channel bonding allowing for high bandwidth for large packages and low latency for small packages. In the short message limit one can see that POE MPI still gives better performance, such that POE MPI should be the flavor of choice for latency limited codes.

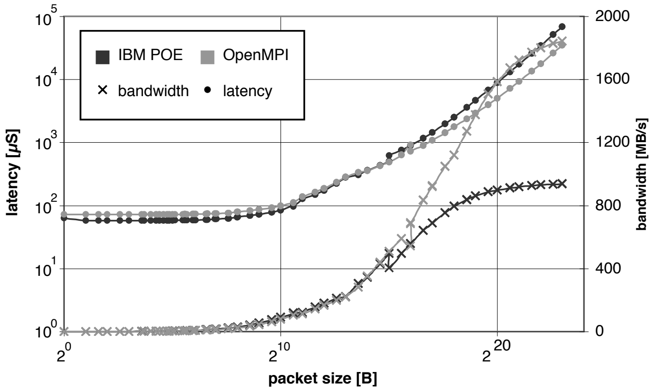

Fig. 2. MPI benchmarks comparing IBM POE (dark gray) with OpenMPI (light gray). Markers for bandwidth values are crosses with corresponding axis on the right side and markers for latency are points with corresponding axis on the left side.

To measure the impact of the different topology of the management and communication network we have used the INTEL MPI benchmark (IMB). This suite of benchmarks determines time consumption and bandwidth for the different MPI operations. Performance is determined while scanning different values for package size and number of nodes involved in the communication. We have found that the single transfer test and the parallel transfer test are not able to exhibit the differences between the two network topologies. This is due to the fact that, in these tests, communication always happens with neighboring nodes such that there is very little bandwidth requirement on the second stage of the management network (between the different blade centers). In contrast, the collective benchmarks did show a clear difference between the two networks. For instance, an Altoallv operation with a message size of 4194304 Byte takes 63.17 seconds, in average, on the management network while the same operation only takes 9.75 seconds in the communication network. Overall the IMB benchmarks exhibited an aggregate network bandwidth of more than 6 GB/s.

As described in Section 3, a parallel filesystem aggregates disk storage on the computing nodes into one large pool of storage space. From a logical point of view this storage space is nothing else than a networked file system accessible from all the nodes belonging to the cluster. Due to its internal structure, it is expected that this file system exhibits better performance than a normal networked file system.

Two different benchmarks were used to test the **performance of the parallel file system**. To get an estimate of the available aggregate write performance, all the nodes had to write a separate file in the same file system, thus measuring the aggregate write throughput to the system. We observed a write throughput of more than 240MB/s, an impressive value for a file system connected via a single gigabit fabric. As a second test, we determined the performance in writing one single HDF5 [12] file in parallel. Using an MPI test program, each one of the nodes opens the same HDF5 file and writes a data slab to it. Aggregate bandwidth obtained in this test lies at 58 MB/s. We are currently investigating why the HDF5 performance is lower than expected.

The **High Performance Linpack Benchmark** (HPL) [28] is a standard performance measurement used for supercomputing systems. For the IST cluster we considered a large set of different parameters to get the best HPL value. The final value was obtained in 64bit, with large memory pages, IBM POE and the XL compiler suite with full optimizations. The obtained HPL value of 1.6 Tflops places the IST Cluster at the top of this benchmark for Portugal as of Spring 2007. Considering the total power consumption of the system (25 kW) we obtain a weighted power consumption of 15.7 kW/Tflops.

7 Future integration in the RNCA grid

The IST Cluster is the first node of an integrated national effort to deploy a network of advanced computing facilities, under the RNCA [29]. The other founding three nodes are installed in Instituto de Engenharia Mecânica - Faculdade de Engenharia da Universidade do Porto, Laboratório Nacional de Engenharia Civil, and Universidade do Minho. The integration of the four nodes in a national grid is now in progress, with the main goal of providing an integrated national infrastructure for advanced computing.

The integrated approach to the design/configuration of each node led to a uniform management solution based on Moab, while retaining the ability to implement different architectures in each node, capable of responding not only to the local requests of the infrastructure but also to the diverse requirements of the community. Since all four participating nodes are using Moab as a workload manager the nodes will be combined with a minimal effort, while remaining in a well familiarized environment for job management. This integration will be performed in a decentralized manner, such that each node manages its part of the grid. Moab provides facilities for grid accounting, resource mapping, credential mapping, file staging, authentication and others. The integration of these four nodes will provide a baseline for the integration of computing resources with heterogeneity in hardware and operating system, thus providing a clear path and the seed for the further development of the network with the aggregation of additional infrastructures.

8 Ongoing work and future plans

The IST Cluster is now running in production mode, with many scientific results already acknowledging its use [23]. Several hardware extensions are being planned.

The different options under study are: (i) the increase of the number of computing nodes to a total of 392 CPU cores (estimated HPL = 2.4 GFlops), (ii) the installation of an Infiniband high speed interconnect (estimated HPL = 1.9 GFlops), (iii) the increase of the number of computing nodes to a total of 392 CPU cores and the installation of an Infiniband interconnect (estimated HPL = 2.7 GFlops), and (iv) the increase of RAID storage. On the software side - as a general guideline - a strong effort to reduce the number of commercial software packages used on the IST Cluster will be taken, moving towards software with academic licensing and, preferably, towards public open-source solutions. Future efforts will also be directed towards the improvement of the support for scientific computing, increased throughput and performance optimization while remaining open for new ideas and further developments, within national grid efforts.

Acknowledgements

This work is partially supported by Fundação para a Ciência e a Tecnologia, through grant REDE/1515/RCA/2005. We would like to thank the staff of CIIST for the local support. We would like to thank Eng. J. Pereira, Prof. J. D. Domingos, Eng. J. Prates and Prof. J. P. Matos for our interesting discussions on the history of computers at IST. We also would like to thank the IBM project team for their comprehensive and competent collaboration on the IST Cluster project.

References

1. G. Burns, R Daoud and J. Vaigl. ”LAM: An Open Cluster Environment for MPI”, *Proceedings of Supercomputing Symposium*, pp. 379–386, (1994).
2. Cluster resources, Moab scheduler, `http://www.clusterresources.com/pages/products/moab-grid-suite.php`.
3. Cluster resources, TORQUE resource manager, `http://www.clusterresources.com/pages/products/torque-resource-manager.php`.
4. J. D. Domingos. ”A introdução dos computadores no ensino da Engenharia: a aquisição do IBM 360/44 do IST”, `http://www3.dsi.uminho.pt/memtsi/mesas/4_sessao/mesa4%20-%20Delgado%20Domingues.pdf`.
5. R. A. Fonseca, L. O. Silva, R. G. Hemker, F. S. Tsung, et al. ”OSIRIS: A Three-Dimensional, Fully Relativistic Particle in Cell Code for Modeling Plasma Based Accelerators. *Lec. Not. Comp. Sci.*, **2331**, pp. 342-351, (2002).
6. Force10 Data center switch, `http://www.force10networks.com/products/s50.asp`.
7. I. Foster. ”Globus Toolkit Version 4: Software for Service-Oriented Systems”, *IFIP International Conference on Network and Parallel Computing, Springer-Verlag LNCS 3779*, pp. 2-13, (2006).
8. E Gabriel, G. E. Fagg, G. Bosilca, T. Angskun et all. ”Open MPI: Goals, Concept, and Design of a Next Generation MPI Implementation”, *Proceedings, 11th European PVM/MPI Users' Group Meeting*, pp. 97–104, (2004).
9. L. Gargate, R. Bingham, R. A. Fonseca, L. O. Silva et al. ”dHybrid: A massively parallel code for hybrid simulations of space plasmas”, *Computer Physics Communications*, **176**, pp. 419-425, (2007).
10. Gnu compiler, `http://gcc.gnu.org/`.

11. W. Group, N. Doss, and A. Skjellum. *MPICH Model MPI Implementation Reference Manual*, Argonne national laboratory, Argonne, (2003). http://www.compsci.wm.edu/SciClone/documentation/software/communication/MPICH-1.2.5/mpiman.ps
12. The HDF group, http://hdf.ncsa.uiuc.edu/.
13. C. Huang et al. "QuickPIC: A highly efficient particle-in-cell code for modeling wakefield acceleration in plasma", *J. Comp. Phys.*, **217**, pp. 658-679 , (2006).
14. C. Huang et al. "QuickPIC: a highly efficient fully parallelized PIC code for plasma-based acceleration", *J. Phys.: Conf. Ser.*, **46**, pp. 190-199 , (2006).
15. IBM Blade center open fabric, http://www-03.ibm.com/systems/bladecenter/hardware/openfabric/ethernet.html
16. IBM Blade JS21, http://www-03.ibm.com/systems/bladecenter/hardware/servers/js21e/
17. IBM DS4700, http://www-03.ibm.com/systems/storage/disk/ds4000/ds4700/
18. IBM General parallel filesystem, http://www-03.ibm.com/systems/clusters/software/gpfs/
19. IBM p505, http://www-03.ibm.com/systems/p/hardware/entry/505/
20. IBM p510, http://www-03.ibm.com/systems/p/hardware/entry/510/
21. IBM Parallel operating environment, http://www-01.ibm.com/cgi-bin/common/ssi/ssialias?infotype=an&subtype=ca&htmlfid=897/ENUS201-331&appname=usn
22. IBM x306, http://www-07.ibm.com/systems/hk/x/rack/x306/
23. IST Cluster webpage, http://istcluster3.ist.utl.pt
24. J. Martins. "ISTória de Bits", http://diferencial.ist.utl.pt/edicao/18/istoria.htm.
25. M. L. Massie, B. N. Chun, and D. E. Culler. "The Ganglia Distributed Monitoring System: Design, Implementation, and Experience", *Parallel Computing*, **30**, pp. 817–840, (2004).
26. P. M. Papadopoulos, M J. Katz, G Bruno. "NPACI Rocks: Tools and Techniques for Easily Deploying Manageable Linux Clusters", *Cluster 2001: IEEE International Conference on Cluster Computing*, (2001).
27. J. Pereira, J. D. Domingos, J. Prates and J. P. Matos. Private communications.
28. A. Petitet, C. Whaley, J. Dongarra and A. Cleary. "HPL - A Portable Implementation of the High-Performance Linpack, Benchmark for Distributed-Memory Computers", http://www.netlib.org/benchmark/hpl/ (2004).
29. Rede Nacional de Computação Avançada, http://www.fct.mctes.pt/pt/pnrc/default.asp?opt=7&lang=p.
30. Q. O. Snell, A. R. Mikler and J. L. Gustafson. "NetPIPE: A Network Protocol Independent Performance Evaluator", *IASTED conference*, http://www.scl.ameslab.gov/netpipe/paper/netpipe.ps.
31. T. L. Sterling, J. Salmon, D. J. Becker and D. F. Savarese. *How to Build a Beowulf*, The MIT Press, Cambridge, Massachusetts, (1999).
32. Z research Inc, Gluster, http://www.gluster.org/.