Performance of a Dynamic Small-size HPC Platform

D. Kehagias, M. Grivas, G. Pantziou

Abstract - **The dynamic cluster that we proposed in a previous work, matured to a full-scale research tool, that proved its abilities and its potentials. In order to justify its existence, we produced a qualitative performance examination, based on simple techniques and benchmarking tools and programs. The measurements demonstrated that the proposed platform indeed provides a powerful highperformance computing environment, capable of supporting quite large problem solving. It also showed that the advantages of the NoW as a dynamic pool of processors, provides extra on-demand power .**

Keywords - **Beowulf Clusters, NoW, dynamic Cluster, Linux Cluster performance enhancement.**

I. INTRODUCTION

Our previous work on clusters and NoWs concluded to a dynamic, high-performance, versatile, multi-computer complex, that exhibited dynamically adapted performance, ensured cluster-level minimal performance and availability, as well as an interesting Grid resemblance. The complex was used primarily as an educational platform ([1]). Nevertheless, as a dynamic platform, it provides high-performance that can be used for research purposes, too. Its potentials are valuable, since the minimum cluster ensures availability, but the dynamic, ondemand or when available expansion of performance over the NoW, offers a significant power for several kind of scientific problems.

In this paper we present the results of various performancewise test run on the aforementioned platform. It proves its ability to produce high-performance computing, with simple configuration and reuse of existing equipment. To justify that, we ran a set of tests on the platform, in many phases of available nodes. We measured how performance scales within the cluster and when the NoW nodes become available.

The following section describes the platform as used. Next, we explain the set of tests we used and follows the section with the results. We conclude with next steps in this research.

II. CONFIGURATION

The setup we use in the original form of the dynamic cluster is depicted in fig. 1. It includes a small Beowulf-class cluster and a NoW. The cluster consists of 8 PCs, employing Pentium 4 and 512 MB RAM, and a dedicated 100 Mbps Ethernet. No swap is used. The NoW configuration is based on the PC equipment of the Microcomputers laboratory, consisting of 18 PCs connected to a 10/100 Mbps Ethernet. Through a gateway, the laboratory LAN connects to the Institution (TEI) backbone and to the Internet.

The cluster PCs are dedicated to the parallel processing. On the other side, workstations that are members of the NoW may be used by students. Such use may consist of laboratory operation that can be light, moderate or heavily burdening nodes processors, such as compiling, graphics etc. In addition, students may overload the network (downloading large files) or the workstation (complex software, games etc). Last but not least, workstations may stack or reboot any time, without prior notice. Hence, NoW nodes cannot be considered totally available. Instead, they can assist dynamically to increase computing power, especially when no workshop sessions take place.

The parallel processing media is the message passing interface (MPI) and specifically LAM, an open-source implementation of MPI. One of the cluster PCs plays the role of the central controller for both the cluster and the NoW. This PC carries two NIC cards, one for the cluster s LAN and one that connects to the laboratory LAN. As explained in [2], this configuration ensures high security, availability and dynamic high performance.

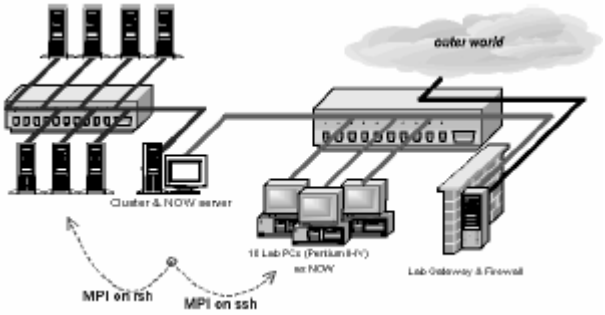

Fig.1. The dynamic cluster.

The cluster LAN is physically isolated from the rest; only the central controller may communicate with the cluster nodes. Hence, their communication happens without any security measurements. Packets encoding or cryptography would impose an impediment to performance, introducing higher latencies and limited bandwidth, as well as wasting CPU cycles. For that, there is no secure shell or SSL installed in cluster nodes. On the other side, security is of prime importance on the NoW, where all systems are exposed to Internet communications, that imply external and internal attacks. A multi-layered protection scheme in the Institution (TEI) enforces protection and therefore external attacks are hard to happen. However, students that work on the NoW nodes may, intentionally or unintentionally, expose the system to attacks, over their workstations. Therefore, security on this

D.Kehagias, M.Grivas and G.Pantziou are with the Department of Informatics, Technological Educational Institution (TEI) of Athens 122 10 - Athens, Greece.
Emails: dkehayas@teiath.gr, mgrivas@cs.teiath.gr pantziou@teiath.gr

side has to be enforced. Using ssh has been proved a feasible and adequate solution that does not degrade severely nodes and network performance.

III. TESTS

The sole objective in this work is to measure -and ensurethe performance of our proposed platform. At the moment, the possible jobs are limited to computational tasks, although both NoW and cluster nodes employee hard disk and could assist in other distributed computing facilities, such as databases, file management etc.

For the measurements of performance in clusters, many researchers have proposed different techniques and approaches. Some study in details specific factors and their influence to overall performance. Such specifics include network details ([3]), computational or communication needs of tasks ([4]), individual systems performance and heterogeneity ([5][6]), MPI performance and others. Because of the multiple factors that get into account, it is not always possible -or at least easyto identify the factors that affect more overall performance or the way they interact with other factors. It seems that with a large-scale statistical analysis in user-level programs, one cannot identify immediately such factors but may have a very clear picture of the system s performance and the circumstances that affect it.

Many researchers propose simple, user-level tasks as measuring facilities that shows the overall system performance in specific kinds of tasks ([7],[8]). From the differentiation among different kinds and several factors, one can better isolate the influencing factors. Such program metric programs include simple tasks like array multiplication, or more complex sets of measuring tasks that extract immediately sets of results for several values, such as NAS, HPL, HINT and the Pallas Benchmarks.

Our work has been guided by the research interests of influencing colleagues. Specifically, our current research emphasises the use of cluster for load balancing studies, image analyses and information retrieval. Thus, we ensured that performance distribution among different values will depict an analogy for our real tasks. Another important factor, due mainly to the educational perspective, was simplicity in con- figuration and results extract. We concluded in using the NAS benchmarks.

The NAS benchmarks suite was developed by NASA to measure the performance of parallel systems. It consists of 8 tests, namely the Block Tridiagonial (BT), the Conjugate Gradient (CG), the Embarrasingly Parallel (EP), the Fourier Transform (FT), the Integer Sort (IS), the LU Decomposition (LU), the Multi-Grid (MG) and the Scalar Pentadiagonal (SP). Details, articles and information can be seeked at the official site: http://www.nas.nasa.gov/NAS/NPB.

A crucial issue is the scalability of the platform in the form of gain by adding more processors from the NoW. Since the Beowulf-class cluster is a very well known -and very much studied- platform, we do not emphasize the cluster benchmarks, apart from a regular test that runs for measuring and configuration validity. Then, any test focuses on the merits of the NoW to the overall performance, proportionally to the cluster performance.

For our studies, the most interesting tests are EP and FT, that are more compute-intensive. Some other tests (i.e. LU, CG, MG) are used only for informational purposes, because they focus on issues that are not of interest, like blocking communication. BT, SP, IS were not eligible at all, since their requirements are beyond the form or scope of our platform. Those test run as-is, without any optimization or adaptation; neither did any optimization happen to the cluster software. Regarding the size of problems, it is recommended that size B or higher should be used accurate, exact performance comparisons. A is considered very small and may be affected by other factors that are irrelevant to ones study. Size W is a workstations (single machine) version, that is very small and not properly parallelized, mainly for comparison reasons. We used B size since C is very demanding and may not run in all of our equipment.

The common procedure for measuring performance is through the speedup value, that is the amount that the program will run faster than if it was running in a single machine. The most famous approach comes from the computer scientist Amdahl, back in 1967 ([9]). Apart from the speedup factor, he also introduced a formula that produces the upper limit of speedup, known as Amdahl s Law :

$$Speedup_N = \frac{T_{ser} + T_{par}}{T_{ser} + \frac{T_{par}}{N}} = \frac{1}{T_{ser} + \frac{1-T_{ser}}{N}}$$

where T_{ser} is the time to complete the serial portion of the program, T_{par} is the time for the parallel portion and N is the number of processors. If f is the serial fraction of the program then $T_{ser} = fT_1$, where T_1 is the time to execute in a single processor, and

$$Speedup_N = \frac{\mathrm{N}}{\mathrm{fN} + (1-f)}$$

Triggered by results on MPP that surpassed the limits of Amdahl Law, a newer study, [10], concluded with a different formula named Gustafson-Baris Law:

$$ScaledSp_N = \frac{T_{ser} + NT_{par}}{T_{ser} + T_{par}} \text{ or } ScaledSp_N = N + (1-N)T_{ser}$$

However, those laws were made for parallel systems and a long time ago, ignoring communication latencies between processors and other elements, operating system buffering, memory management and other delays. This is by far not the case in distributed systems, like clusters. Several attempts have been made to embed communication and other latencies into the above laws (for example [11]).Clusters perform better under specific circumstances, with a major factor being the computational effort needed for each parallel task. After all, in both the aforementioned laws, latencies can be considered as part of the serial portion. If the parallel portion is by far bigger (i.e. the parallelizable tasks are very big), then $f << 1 - f$ and practically Amdahl s law reaches the optimal N. Thoughts and information on behaviour of clusters can be seen in [12].

IV. RESULTS

The standard metrics is the time it takes for the systems to finish with a specific task. Since the nodes are homogeneous, we can compare their performance to a single computer and specifically the central controller. It should be noticed that each task tried is vastly parallelizable, since our research tasks later will be parallel, too. There was no meaning for a comparison against a simplified, serial algorithm for each measuring program. A single machine setup means that a single PC runs the same program, not that the program is single-threaded or serial. The performance penalty regarding scheduling, multi-tasking and local latencies is out of the scope of this work, which does not include different algorithms. In general, such issues do not affect a beowulf cluster s performance, since communication latencies of commodity networks are many orders of magnitude bigger.

In table I, we compare the cluster against a single machine, for 2 different tests (EP and FT) and for 3 problem sizes each. Our results ensure the validity of our configuration and study the merits in performance to the whole system when adding NoW nodes. The cluster, apart from the validity check, is considered as one step above the single system. The FT test in B size is very large and this is likely the reason that it could not run in the single machine configuration, since we do not use any swap space. However, the most important is that FT exhibits large communication overhead in one of its stages and therefore does not earn as much from the usage of the cluster. As seen in other works, FT can have superlinear scaling on platforms with very fast communication infrastructures, such as InfiBand, Myrinet or SCI. However this is not our case and, hence, FT performance faces a severe bottleneck in the data exchange phase. In addition, since our tasks are more process-oriented, we emphasize EP test.

TABLE I
CLUSTER PERFORMANCE RESULTS FOR DIFFERENT TESTS AND SIZES

Test & size	Single - Mops	Cluster - Mops
FT (W)	34	62
FT (A)	7	32
FT (B)	-	59
EP (W)	1.3	10.4
EP (A)	1.3	10.3
EP (B)	1.1	8.9

Another validity test is the scalability of performance within the cluster (table II). Although there are many studies about scalability within a Beowulf-class cluster, before going to the larger and more unpredictable situation we should ensure that the basis of our configuration follows the norm.

TABLE II
PERFORMANCE SCALABILITY IN THE CLUSTER

# of nodes	EP (B)	FT (A)
Single	1.1	7
2	2.2	10
4	4.5	18
8	8.9	56

Indeed, table II shows that performance within the cluster increases as expected. We should notice that our primary type of tasks (EP) increase linearly and do not exhibit any anomalies, like superlinearity etc.

Then, we study scalability when new computers are introduced, over the existing cluster. We consider the new PCs as available (idle), with a lightly loaded network as it happens off working periods. In table III, after cluster results, each row shows how many more PCs participate. The maximum number (16) is the number of workstations in the laboratory. In future versions, we will have the opportunity to include other laboratories, increasing the number significantly.

After all, the way of connection, i.e. over a separate LAN, using encoding etc, does not allow for an analogous linear scaling of performance. The most interesting detail is that additional nodes exhibit linearity as an independent part, meaning that NoW presents a function of independent cluster that works in parallel to -and complementing- the basic cluster.

TABLE III
PERFORMANCE SCALABILITY WITH NOW

# of nodes	EP (B)
single	1.1
cluster (8)	8.9
+2	9.1
+4	11.4
+8	14.3
+16	20.1

The performance increment is significant. Especially in large problems the merits in perormance can be more than visible. The only drawback is that processes should minimise information exchange, like EP that does not perform any interprocess communication. For such kind of tasks, time of execution can be significantly reduced. The following table IV shows the merits in terms of time gained for the EP test.

TABLE IV
TIME NEEDED TO COMPLETE TASK EP, SIZE B

# of nodes	EP (B)
single	682
cluster (8)	86
+2	80
+4	53
+8	35
+16	22

The following figures depict the improvement in the platform s performance: fig 2 shows the performance in Megaoperations per second (Mops) as a function of the number of nodes within the cluster for both FT (size A) and EP (size B), fig. 3 shows the performance in Mops for EP (size B) test within the cluster and beyond, including NoW nodes, as well as the time in sec.

V. CONCLUSION

In this paper we have studied the performance of our proposed complex, dynamic, multi-computer platform. These first results are very encouraging. Apart from an educational

tool, it exhibits real merits to research related to parallel algorithms and HPC studies. They enable us to continue investigation about the performance and the potentials of this platform.

Forthcoming investigation includes the extension of the platform by adding more computers, more laboratories and other individual clusters that will break the homogeneity and will introduce new problems. In a another direction, parallel algorithms regarding Information Retrieval, cluster and Grid scheduling and other parallel tasks will be produced to the platform. This will allow studies on performance of parallel programs over unstable connections and unpredictable latencies. Finally, more fine-grained and optimized tests will run to produce detailed results, specific to each participant element of the cluster, i.e. communication infrastructure, processing elements, memory usage, nodes performance and others.

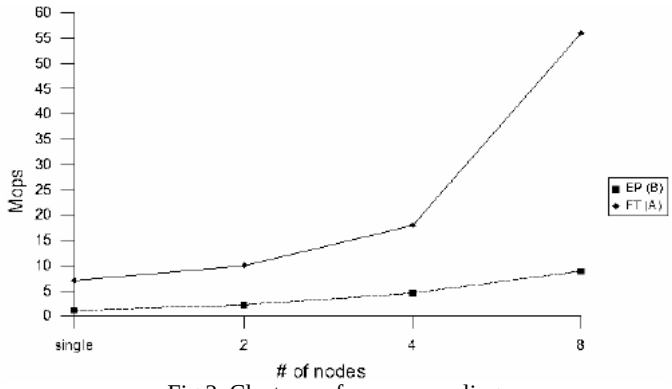

Fig.2. Cluster performance scaling.

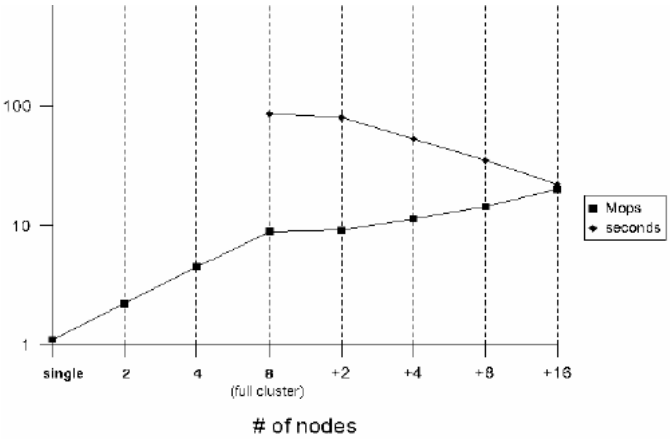

Fig.3. NoW - improving performance, minimizing time.

ACKNOWLEDGEMENTS

This work has been partly supported by the 3rd European Support Framework: Operational Programme in Education and Initial Vocational Training II, programme Archimedes.

REFERENCES

[1] D. Kehagias, M. Grivas, G. Meletiou, G. Pantziou, B. Sakellarios, D. Sterpis, and D. Ximerakis, " low-cost dynamic clustering system for education and research", in *Proceedings of TEMPUS Workshop*, Jan. 2004.

[2] D. Kehagias, M. Grivas, G. Meletiou, G. Pantziou, B. Sakellariou, D. Sterpis, and D. Ximerakis, "Building a low-cost high-performance dynamic clustering system", in *6th International Conference on Telecommunications in Modern Satelite, Cable and Broadcasting Services (TELSIKS 2003)* (B. D. Milovanovic, ed.), vol. 2, pp. 608 613, IEEE and Faculty of Electronic Engineering, Nis, Serbia, October 1-3 2003.

[3] J. Mache and V. Lo, "Dispersal metrics for non-contiguous processor allocation", *tech. rep*., University of Oregon, 1996.

[4] D. G. Feitelson, "The forgotten factor: Facts on performance evaluation and its dependence on workloads"

[5] S. Ali, J.-K. Kim, Y. Yu, S. B. Gundala, S. Gertphol, H. J. Siegel, A. A. Maciejewski, and V. Prasanna, "Utilization-based heuristics for statically mapping real-time applications onto the HiPer-D heterogeneous computing system", in *11th IEEE Heterogeneous Computing Workshop (HCW 2002)*, Apr 2002.

[6] F. D. Berman, RichWolski, S. Figueira, J. Schopf, and G. Shao, "Application-level scheduling on distributed heterogeneous networks", in *Supercomputing 96 Conference Proceedings, New York*, pp. 17 22, ACM Press and IEEE Computer Society Press, 1996.

[7] H. Dail, H. Casanova, and F. Berman, "A decoupled scheduling approach for the grads program development environment", 2002.

[8] A. S. Carsten Ernemann, Volker Hamscher and R. Yahyapour, "Enhanced algorithms for multi-site scheduling", in *Proceedings of 3rd IEEE/ACM International Workshop on Grid Computing (Grid 2002) at Supercomputing 2002, Baltimore*, USA, 2002.

[9] G. Amdahl, "Validity of the single processor approach to achieving large scale computing capability", in *Proceedings of the on AFIPS Spring Joint Computer Conference (Reston, Va.)*, pp. 483 485, AFIPS Press, Arlington, VA, 1967.

[10] R. Benner, J. Gustafson, and G. Montry, "Development and analysis of scientific application programs on a 1024-processor hypercube", *Tech. Rep. SAND 88-0317*, Sandia National Laboratories, Feb. 1988.

[11] K.-J. Andersson, D. Aronsson, and P. Karlsson, "An evaluation of the system performance of a Beowulf cluster", *Tech. Rep. 2001:4*, Dept. of Scientific Computing, Uppsala Uni., 2001.

[12] R. S. Morrison, "Cluster computing architectures, operating systems, parallel processing and programming languages", 2 April 2003.