Parallel Computing Case Studies and Performance Assessment of the IFIN-HH Parallel SIMFAP Cluster

Gh. Adam[1,2], S. Adam[1,2], A. Ayriyan[1], E. Dushanov[1], E. Hayryan[1], V. Korenkov[1], A. Lutsenko[1], V. Mitsyn[1], T. Sapozhnikova[1], A. Sapozhnikov[1], O. Streltsova[1], F. Buzatu[2], M. Dulea[2], I. Vasile[2], A. Sima[2], C. Visan[2], E.E. Donets[3], J. Busa[4], I. Pokorny[4]

[1] *Laboratory of Information Technologies, JINR*

[2] *Horia Hulubei National Institute of Physics and Nuclear Engineering, Romania*

[3] *Veksler and Baldin Laboratory of High Energies, JINR*

[4] *Technical University, Kosice, Slovakia*

Abstract

The paper describes the present status of the SIMFAP parallel computing cluster at IFIN-HH Bucharest. The peak performance assessment and outputs of parallel computing case studies are reported. They point to an effective and reliable operation of the open MPI version for high performance parallel computing implemented on the cluster within the present collaboration.

1. Peak performance

The SIMFAP parallel cluster in IFIN-HH was built using open software, both as it concerns the operation system (which is a freely-available Linux distribution that is based on Red Hat's commercial product: Red Hat Enterprise Linux) and the high performance parallel computing message passing interface (MPI) standard. Its main characteristics can be summarized as follows:

Processor	Intel 2x Xeon
Clock frequency	3000 MHz
2-level cash memory	2 MB
Cores within one processor	1
Processors within one node	2
RAM on node	4 GB
Nodes within cluster	8
Overall number of processors	16
Overall number of cores	16
Overall RAM	32 GB
Operation system	CentOS 5
Peak theoretical performance	96 GFlops
Network	Myrinet 2000
MPI	Version 1.2.7

To assess the cluster performance, we used the High Performance LINPACK (HPL) Benchmark, developed at the Innovative Computing Laboratory, University of Tennessee [1], and currently used to provide a reliable basis for tracking and detecting trends in high-performance computing. Twice a year, a list of the sites operating the 500 most powerful computer systems is assembled and released [2]. The same benchmark was used while testing the performance of the Central Information and Computing Complex cluster of JINR [3].

The HPL benchmark measures the computing time T needed for solving systems of linear algebraic equations, based on the LU-decomposition. Given a system of order N, the number of elementary floating point operations required for solving it is $N_{op} = (2/3) \cdot N^3 + 2 \cdot N^2$. The cluster performance is calculated from the formula $P = N_{op}/T$.

The benchmark output is presented in Figure 1. The corresponding numerical data are given in Table 1.

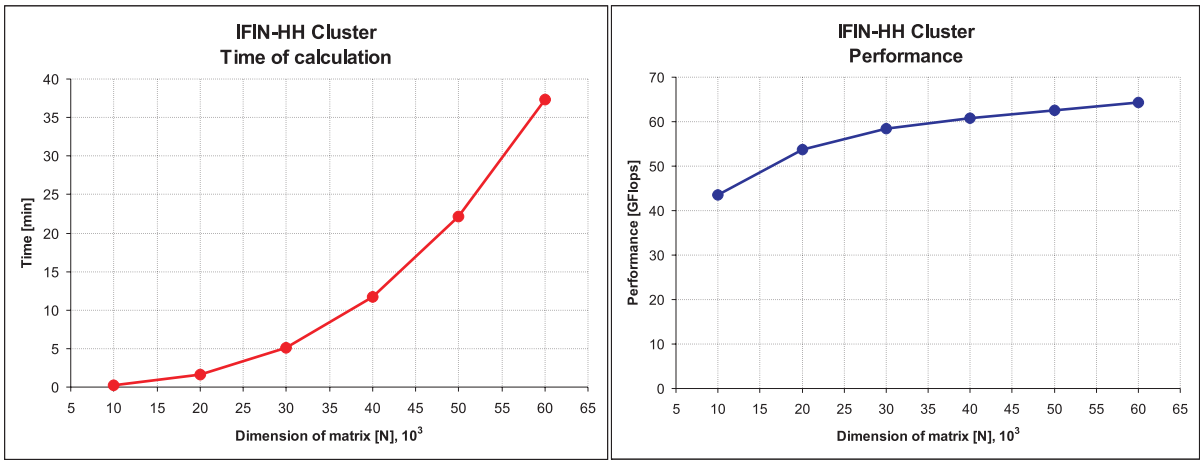

Fig. 1: HPL benchmark output: computing time and performance dependence on the order N of the system of linear algebraic equations

Table 1: Computing time T and performance P vs. the order N of the system

N	T (seconds)	P (GFlops)
10 000	15.31	43.56
20 000	99.29	53.72
30 000	308.29	58.39
40 000	702.15	60.77
50 000	1331.92	62.57
60 000	2241.66	64.24

The data reported in Figure 1 correspond to *the peak performance of the parallel cluster*. Over the reported range of N values, the overwhelming part of each on-node computations are using data transfer operations to the RAM only. At larger N, the RAM capacity is *exceeded*, such that a more and more important fraction of the needed data transfers make use of the disk swap memory, with the consequence that the cluster performance will markedly deteriorate under increasing N.

Under decreasing N values, the cluster performance decreases as a consequence of the increasing weight of the inter-processor communication processes.

The peak experimental value of 64.24 GFlops is very good as compared to the peak theoretical value of 96 GFlops ideally following from the cluster structure under the assumption of the occurrence of operations inside processors only, without any transfer operations, neither to the cache nor to the RAM.

The occurrence of a range of N values (30000 to 60000) over which the cluster performance shows a near flat behaviour points to the very good, N-independent, inter-processor communication secured by the Myrinet 2000 network.

A finite difference analysis of the times of calculation points to the fact that, over the range of the linear small slope peak performance of the parallel cluster, the parallelization of the solving code damps down significantly the N^3 one-processor code complexity to essentially a N^2 cluster code complexity. Detailed analysis will be reported elsewhere.

2. Parallel computing case studies on SIMFAP cluster

Another useful approach to the cluster performance assessment is the investigation of the dependence of the output parameters on the number of processors involved in the solution of given fixed-size problems. In Figure 2 we report the output of two such case studies: (i) matrix multiplication ($N = 1000$); (ii) the calculation of interaction electrostatic potentials in "protein-liquid" systems.

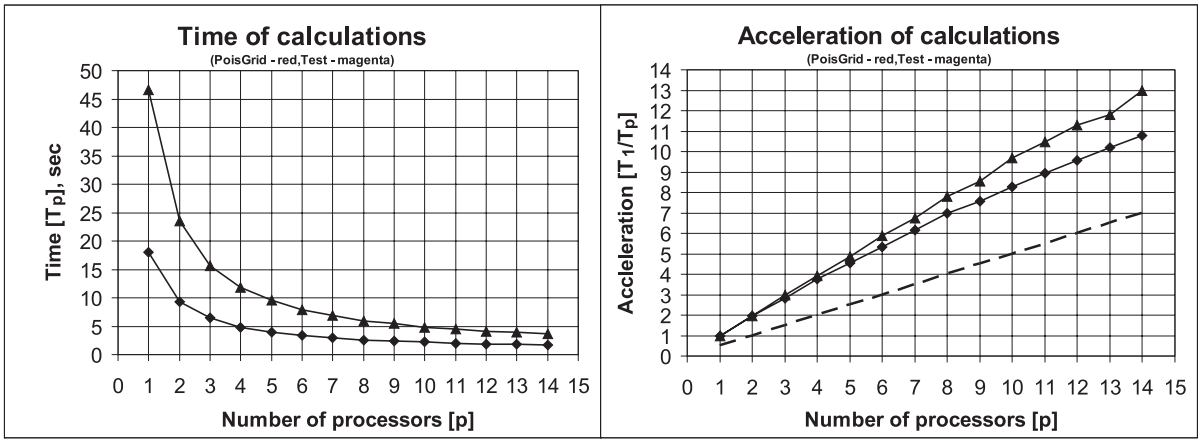

Fig. 2: Computing time (left) and Acceleration (right) of parallel computer solution of two problems: (i) matrix multiplication (rhombs); (ii) the calculation of interaction electrostatic potentials in "protein-liquid" systems (triangles). The dashed straight line provides the $p/2$ plot, above which the experimental data are placed

The scrutiny of these data points to the following features of the cluster operation:

- Under proper code implementation, the computing time needed for getting parallel code solutions *monotonically decreases with* p, the number of parallel processors involved in the parallel solution. Indeed, the addition of further processors to the problem solution replaces the disk swap memory-transfer of temporary data in favour of the much more effective RAM-transfer.

- The fractional gain in the speed of execution gradually decreases with the increase of p, eventually resulting in a plateau-like shape (with a small negative slope) behaviour of $T(p)$. This simply shows that the weight of the disk swap memory data transfer needing longer computation interrupts at the processor level is decreased towards zero, such that the weight of the inter-processor communications becomes sizable.

- At given p, the execution time depends, of course, on the inner problem complexity. The solution of the case study (ii) is simply more demanding than that of the case study (i).

- The *acceleration of the calculations*, defined as the ratio $T(1)/T(p)$, depends on the nature of the solved problem. In the case study (i), the larger weight of the inter-processor communications slows down the acceleration of the calculations.

3. Concluding remarks

The operation system of the IFIN-HH SIMFAP cluster is functioning reliably and efficiently.

The investigation points to a very good measured peak performance which exceeds two-thirds of the peak theoretical performance.

The investigated case studies point to the explicit cluster performance dependence on the inner nature of the solved problem.

Acknowledgments The present work was done within the Hulubei-Meshcheryakov programme of cooperation LIT-JINR–Romania. Romanian authors acknowledge partial financial support from the Romanian Authority for Scientific Research (Project 11/2006 – SIMFAP). A. Ayriyan, E. Hayryan and O. Streltsova acknowledge partial financial support from RFBR 05-01-00645 and RFBR-BRFBR 06-01-81014 Grants. E. Dushanov acknowledges partial financial support from "DREAMS-ASIA" (Development of gRid EnAbling technology in Medicine&Science for Central ASIA) NATO Grant EAP.NIG 982956.

References

[1] A. Petitet, R.C. Whaley, J. Dongarra, A. Cleary, *HPL – A Portable Implementation of the High-Performance Linpack Benchmark for Distributed-Memory Computers* http://www.netlib.org/benchmark/hpl/

[2] http://www.top500.org/

[3] V.V. Korenkov *et al.*, *Testing the new CICC-JINR supercomputing cluster*, this volume, pp.43-46.