Cluster paralel de tip Rocks CECMI, USM

Caracteristici tehnice

HP Proliant DL385G1 Server (x3)
CPU: AMD Dual-Core 2x2.4GHz
RAM: 4GB DDR
HDD: SmartArray 6i, 2x146GB
Network: 2x1Gbps LAN

HP ProLiant DL380G5 Server
CPU: Intel Xeon 5150 4x2.7GHz
RAM: 8GB DDR2
HDD: Hot Plug SAS 730GB (5x146GB)
Network: 2x1Gbps LAN

HP ProLiant DL145R02 Nodes (x12)
(noduri utilizate pentru cluster)
CPU: AMD 275 Dual-Core 2x2.2GHz
RAM: 4GB DDR
HDD: 80GB SATA
Network: 2x1Gbps LAN

HP dx5150 (x12)
(statii de lucru in sala)
CPU: Athlon64 3200+
RAM: 512MB DDR
HDD: 80GB SATA
Network: 1Gbps LAN
Monitor: HP L1706 LCD 17”

HP dx5150 (x3)

(statii de administrare)

CPU: Athlon64 3200+

RAM: 1GB DDR

HDD: 80GB SATA

Network: 1Gbps LAN

Monitor: HP L1940 LCD 19”

HP ProCurve 2824 Switch (x2)

(24x1Gbps ports)

Switch HP ProCurve 2650 (48 ports)

HP R3000 UPS (x2)

HP UniversalRack 10642G2

HP ProCurve AccesPoint 530WW

(antena Wi-Fi)

Storage HP SmartArray 6402

Enclosure: HP StorageWorks MSA20

HDD: 2TB (4x500GB) SATA

Configuratie soft a serverelor:

Frontend Server (pentru gestionarea clusterului, interfata web al clusterului)

Model: HP Proliant DL385G1
OS: Rocks 5.4 (CentOS 5.5)
Soft: Rocks Toolkit, Ganglia, SGE
Programming: OpenMPI, OpenMP, c, c++, f77
Servicii: ssh, nfs, dhcp, apache, php, mysql

FTP Server (pentru stocarea imaginilor de instalare, pachetelor, cartilor/tutorialelor etc)

Model: HP Proliant DL385G1
OS: CentOS 5.5
Servicii: ftp, nfs, smb
Storage: 1.5TB

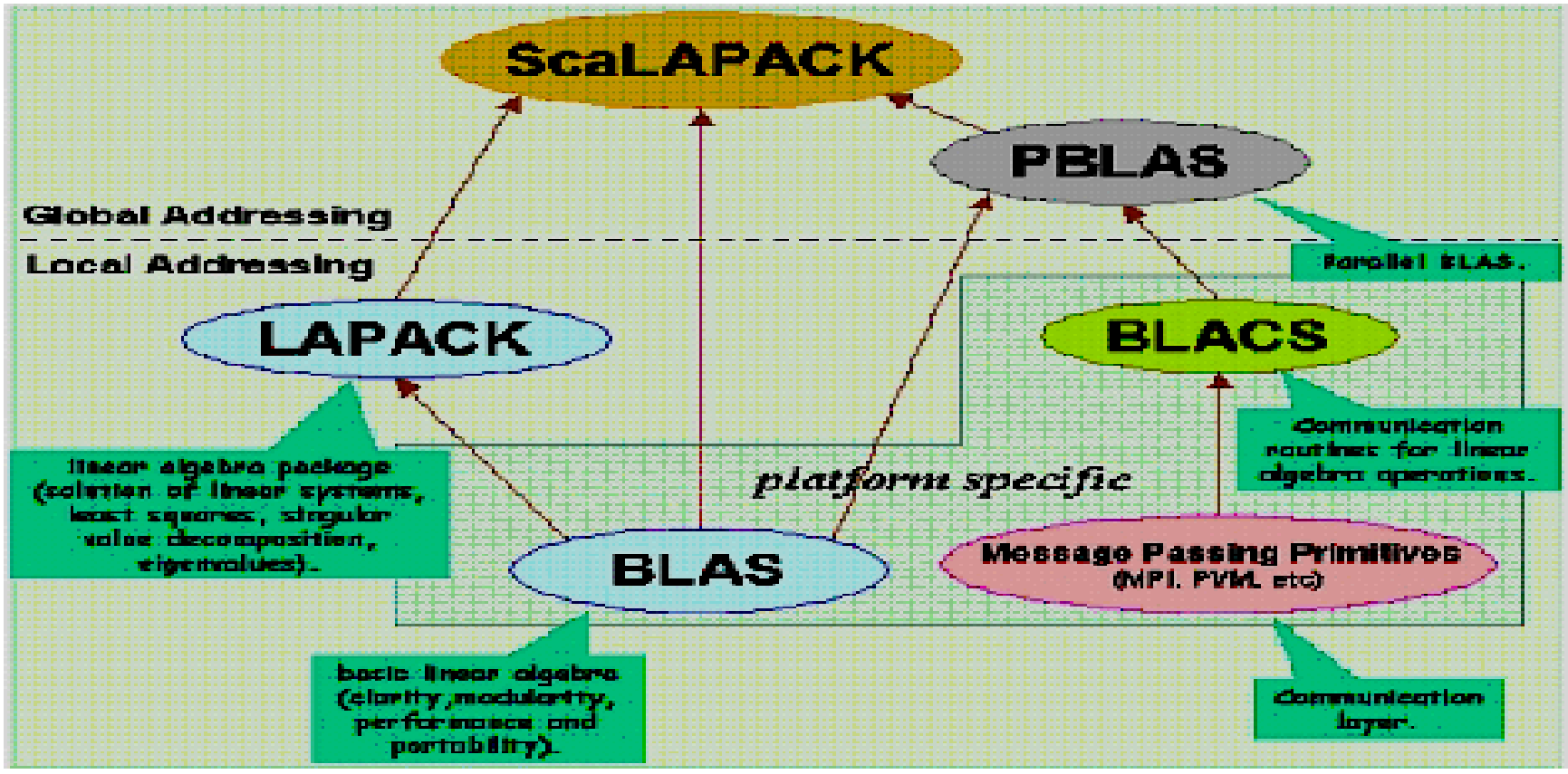
BD Server (pentru gestionarea bazelor de date distribuite)
Model: HP Proliant DL385G1
OS: Windows 2003 Server

ESXi Server (pentru gestionarea masinilor virtuale Web server, Proxy Server, Contester Server, Clone Server, PXE Server)
Model: HP ProLiant DL380G5
OS: ESXi 4.1
Storage: 547GB

Web Server (pentru gazduirea site-ului centrului, a conferintelor si cursurilor de instruire organizate etc)
OS: CentOS 5.5
Servicii: ssh, apache, php, mysql, ftp, nfs

Pachete de programa implementate si utilizate pe claster

Parallel Numerical Libraries ScaLAPACK for linear algebra calculations



ScaLAPACK contains
d**river routines** for solving standard types of problems**,**
computational routines to perform a distinct computational task,
auxiliary routines to perform a certain subtask or common low-level computation**.**

LAPACK is a collection of routines for solving **linear systems, least squares problems, eigenproblems, and singular problems**. High performance is attained by using algorithms that do most of their work in calls to the BLAS, with an emphasis on matrix-matrix multiplication. The LAPACK routines are written as a single thread of execution. LAPACK can accommodate shared-memory machines, provided parallel BLAS are available (in other words, the only parallelism is implicit in calls to BLAS).

The **BLAS** include subroutines for common linear algebra computations such as **dot-products, matrix-vecto multiplication, and matrix-matrix multiplication.** An important aim of the BLAS is to provide a portability layer for computation.

P**BLAS** is a parallel set of BLAS, called perform message-passing and whose interface is as similar to the BLAS as possible. This permitted the ScaLAPACK code to be quite similar, and sometimes nearly identical, to the analogous LAPACK code. The PBLAS will provide a distributed memory standard, just as the BLAS have provided a shared memory standard. This would simplify and encourage the development of high performance and portable parallel numerical software, as well as providing manufacturers with a small set of routines to be optimized.

The **BLACS** are a message-passing library designed for linear algebra. The computational model consists of a one- or two-dimensional **process grid,** where each process stores pieces of the matrices and vectors. The BLACS include **synchronous send/receive routines** to communicate a matrix or submatrix from one process to another, **to broadcast submatrices** to many processes, or **to compute global reductions** (sums, maxima and minima).

SIESTA (**S**panish **I**nitiative for **E**lectronic **S**imulations with **T**housands of **A**toms)

Is both a method and its computer program implementation, to perform efficient electronic structure calculations and *ab initio* molecular dynamics simulations of molecules and solids.
SIESTA can be compiled for serial or parallel execution (under MPI), and can provide :
Total and partial energies.
Atomic forces.
Stress tensor.
Electric dipole moment.
Atomic, orbital and bond populations (Mulliken).
Electron density.
Geometry relaxation, fixed or variable cell.
Constant-temperature molecular dynamics (Nose thermostat).
Variable cell dynamics (Parrinello-Rahman).
Spin polarized calculations (collinear or not).
k-sampling of the Brillouin zone.
Local and orbital-projected density of states.
COOP and COHP curves for chemical bonding analysis.
Dielectric polarization.
Vibrations (phonons).
Band structure.
The parallel version of SIESTA requires BLACS and ScalaPack, plus a MPI message passing library.
Here we build SIESTA with OpenMPI.

Schrödinger Software

Is the scientific leader in computational chemistry, providing software solutions and services for life sciences and materials research.

The Schoredinger Software package includes:
J**aguar**- an *ab initio* quantum mechanics application.
P**rime** - a protein structure prediction package.
G**lide** - software for rapid ligand-receptor docking.
L**iaison** - program for predicting binding affinities.
Q**Site** - QM/MM software for studying enzymatic reaction mechanisms.
P**hase** - for ligand-based pharmacophore modeling.
Q**ikProp** - for ADME properties prediction of drug candidates.
L**igPrep** - a rapid 2D to 3D conversion program that can prepare ligand libraries.
C**ombiGlide** - program for focused library design.
E**pik** - for accurate enumeration of ligand protonation states in biological conditions.
M**acroModel** - for molecular modeling.
S**trike** - a statistical package for examining structure-property relationships.
M**aestro** - the graphical user interface for all of Schrödinger's computational programs.

PETSc Software

The **P**ortable, **E**xtensible **T**oolkit for **S**cientific **C**omputation (PETSc) is a suite of data structures and routines that provide the building blocks for the implementation of large-scale application codes on parallel (and serial) computers. PETSc uses the MPI standard for all message-passing communication.

PETSc includes an expanding suite of parallel linear, nonlinear equation solvers and time integrators that may be used in application codes written in Fortran, C, C++, Python, and MATLAB (sequential). PETSc provides many of the mechanisms needed within parallel application codes, such as parallel matrix and vector assembly routines. The library is organized hierarchically, enabling users to employ the level of abstraction that is most appropriate for a particular problem. By using techniques of object-oriented programming, PETSc provides enormous flexibility for users.

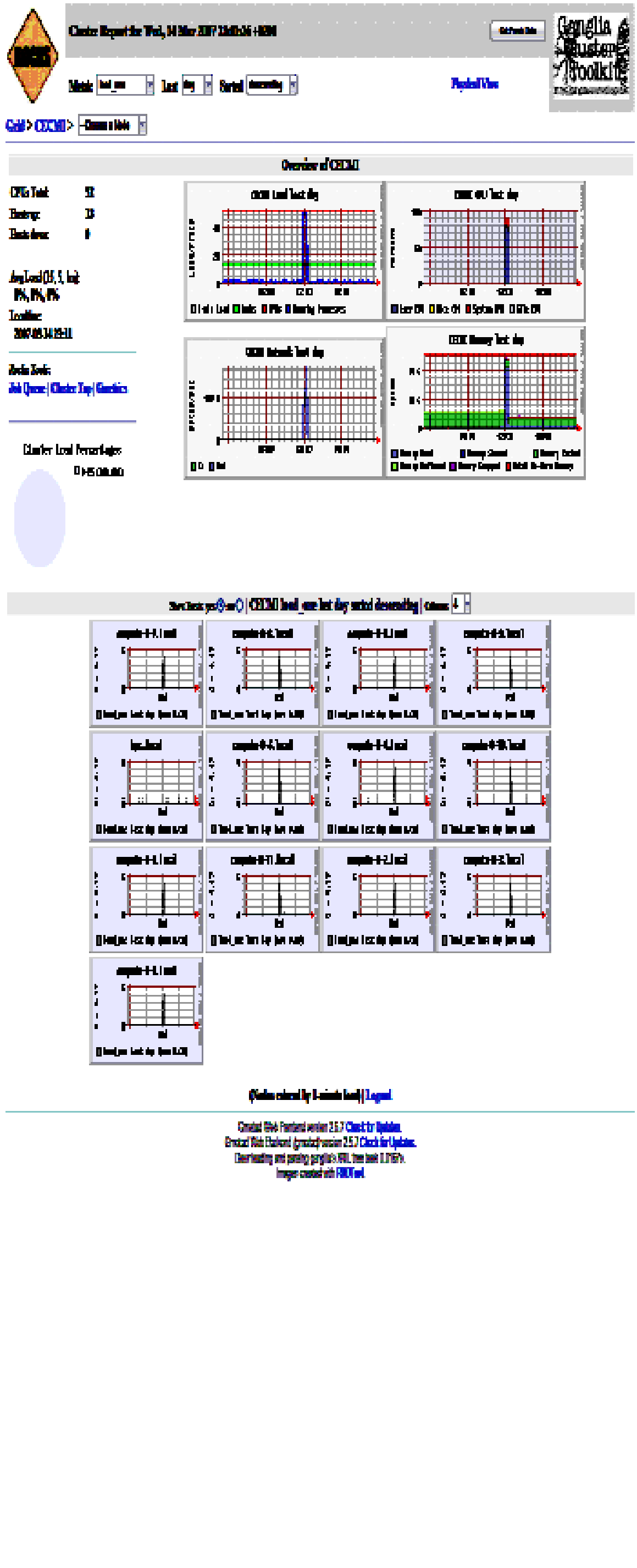
CRYSTALSoftware

CRYSTAL is a quantum chemistry ab initio program, designed primarily for calculations on crystals (3 dimensions), slabs (2 dimensions) and polymers (1 dimension) using translational symmetry, but it can also be used for single molecules. It is written by V.R. Saunders, R. Dovesi, C. Roetti, R. Orlando, C.M. Zicovich-Wilson, N.M. Harrison, K. Doll, B. Civalleri, I.J. Bush, Ph. D'Arco, and M. Llunell from Theoretical Chemistry Group at the University of Torino and the Computational Materials Science Group at the Daresbury Laboratory near Warrington in Cheshire England.

M**ath Kernel Library Software**

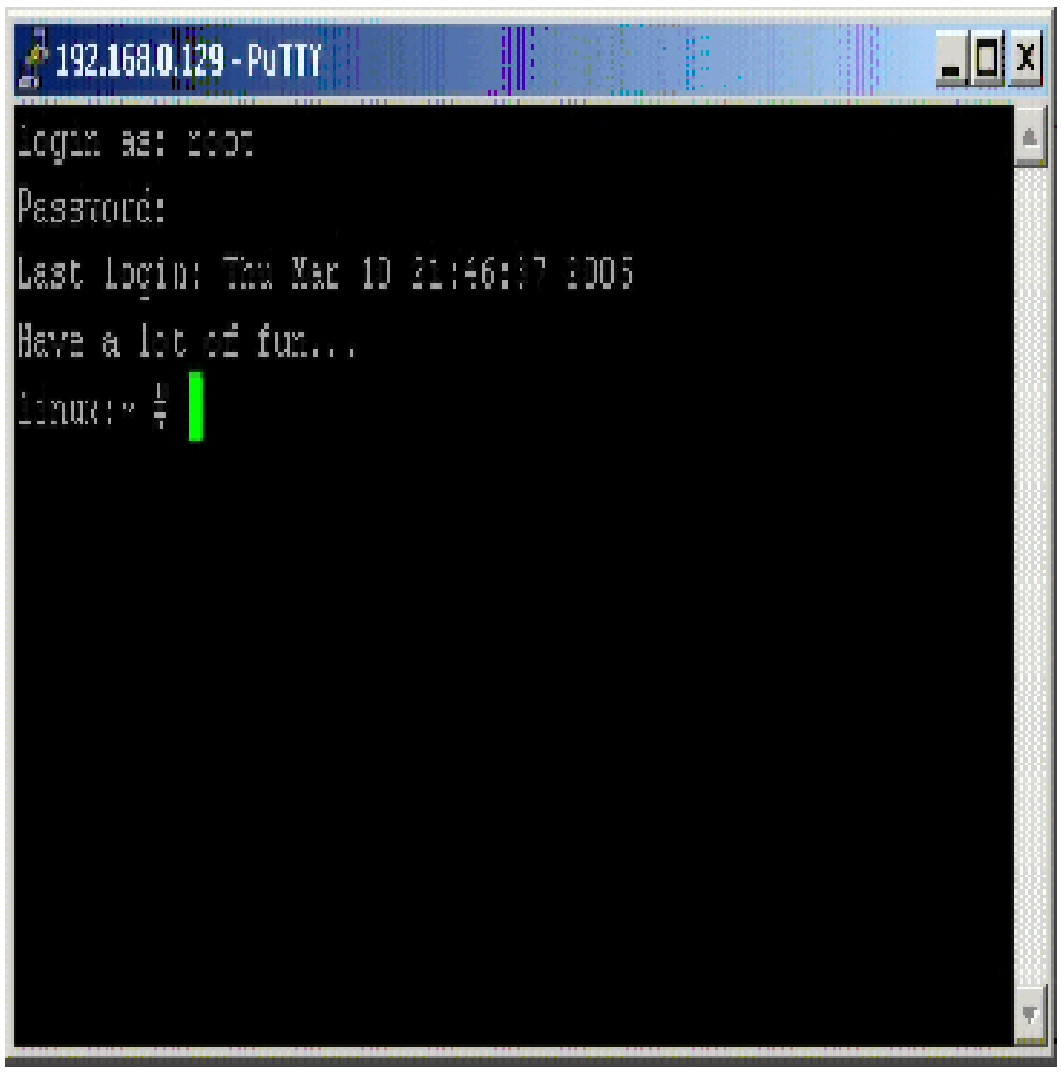
Intel 's **Math Kernel Library (MKL)** is a library of optimized math routines for science, engineering, and financial applications. MKL provides routines in the following areas
BLAS and Sparse BLAS
LAPACK
PBLAS
ScaLAPACK
Sparse Solver routines
Vector Mathematical Library functions
Vector Statistical Library functions•Fourier Transform functions (FFT)
Cluster FFT
Trigonometric Transform routines
Poisson, Laplace, and Helmholtz Solver routines
Optimization (Trust-Region) Solver routines
GMP arithmetic functionsOptimization (Trust-Region) Solver routines
GMP* arithmetic functions

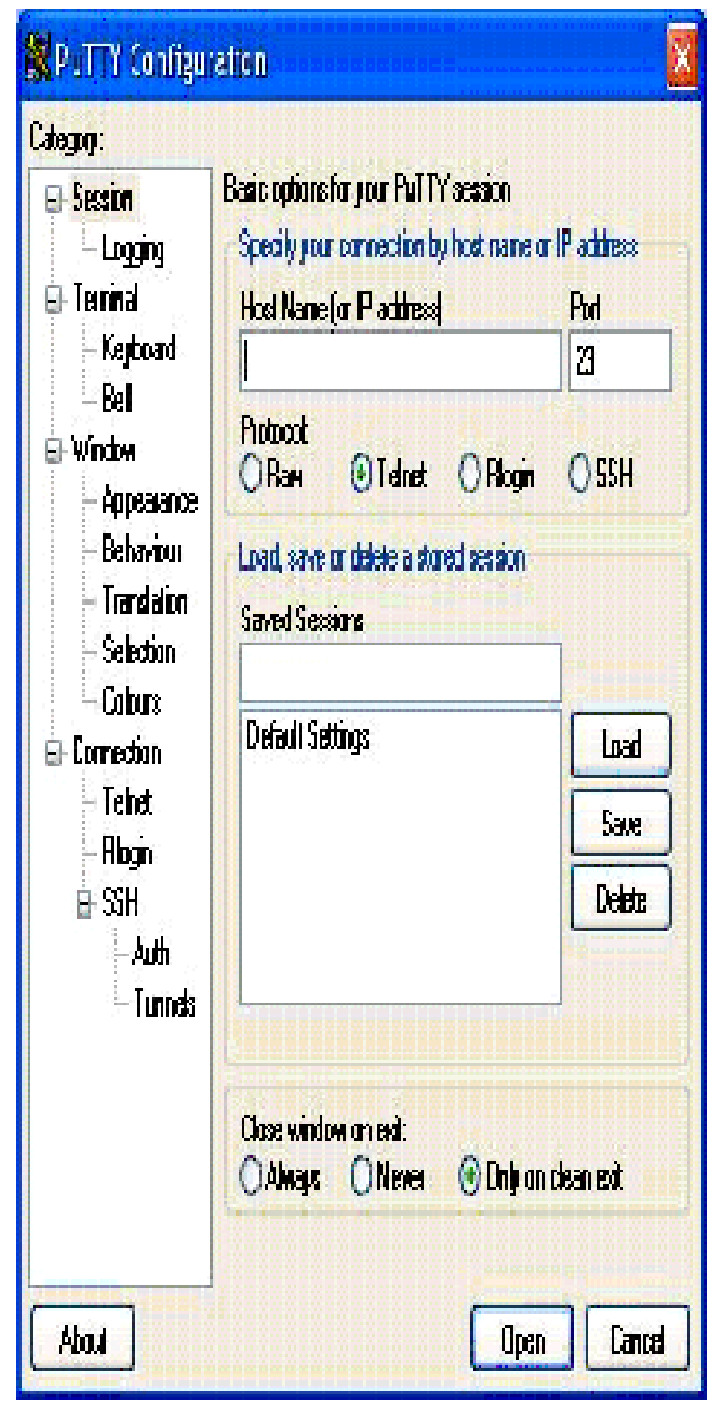
Aplicaţia web de monitorizare a activităţii clusterului “Ganglia” (http://hpc.usm.md/)



Utilizarea protocolului SSH pentru a executa programe pe cluster (din Windows)

PuTTY: a free telnet/ssh client <http://www.putty.nl/>





Rezutatul testării cluster-ului folosind pachetul specializat Linpack Benchmark <http://www.netlib.org/benchmark/linpackjava/>

```
[hancu@hpc ~]$ /opt/openmpi/bin/mpirun -np 52 -machinefile nodes   /opt/hpl/openmpi-hpl/bin/xhpl
[hancu@hpc ~]$ cat hpl.out
=============================================================================
HPLinpack 1.0  --  High-Performance Linpack benchmark  --  September 27, 2000
Written by A. Petitet and R. Clint Whaley,  Innovative Computing Labs.,  UTK
=============================================================================
An explanation of the input/output parameters follows:
T/V    : Wall time / encoded variant.
N      : The order of the coefficient matrix A.
NB     : The partitioning blocking factor.
P      : The number of process rows.
Q      : The number of process columns.
Time   : Time in seconds to solve the linear system.
Gflops : Rate of execution for solving the linear system.
The following parameter values will be used:

N      :  69000
NB     :    200
P      :      2
Q      :     26
PFACT  :   Crout    Right
NBMIN  :      2
NDIV   :      2
RFACT  :   Right
BCAST  :   1ring
DEPTH  :      0
SWAP   : Mix (threshold = 64)
L1     : transposed form
U      : transposed form
EQUIL  : yes
ALIGN  : 8 double precision words
----------------------------------------------------------------------------
- The matrix A is randomly generated for each test.
- The following scaled residual checks will be computed:
   1) ||Ax-b||_oo / ( eps * ||A||_1  * N        )
   2) ||Ax-b||_oo / ( eps * ||A||_1  * ||x||_1  )
   3) ||Ax-b||_oo / ( eps * ||A||_oo * ||x||_oo )
- The relative machine precision (eps) is taken to be         1.110223e-16
- Computational tests pass if scaled residuals are less than          16.0
=============================================================================
T/V                N    NB     P     Q               Time          Gflops
----------------------------------------------------------------------------
W00R2C2         69000   200     2    26       1783.07(30min)     1.228e+02
----------------------------------------------------------------------------
||Ax-b||_oo / ( eps * ||A||_1  * N        ) =        0.0154313 ..... PASSED
||Ax-b||_oo / ( eps * ||A||_1  * ||x||_1  ) =        0.0114427 ..... PASSED
||Ax-b||_oo / ( eps * ||A||_oo * ||x||_oo ) =        0.0020272 ..... PASSED
=============================================================================
Finished      2 tests with the following results:
              2 tests completed and passed residual checks,
              0 tests completed and failed residual checks,
              0 tests skipped because of illegal input values.
----------------------------------------------------------------------------

End of Tests.
```

Utilizarea clasterului in procesul de instruire

Ciclul I Studii de licenta

“*Programere paralela“,* specialitatile Matematica Aplicata, Informatica si Tehnologii informationale:

Cursul „Programare paralela” î-şi propune să acopere minimul de noţiuni necesare înţelegerii modalitatilor de implementare soft-ware a algoritmilor paraleli pe diverse sisteme paralele de calcul de tip claster. Cursul va contribui esenţial la dezvoltarea aptitudinilor şi capacităţilor de construire, studiere a algoritmilor paraleli şi implementarea lor în diferite sisteme paralele de calcul. Drept rezultat al cunoştinţelor acumulate la orele de curs studentul trebuie să poată aplica cele mai importante metode şi rezultate expuse în curs, pentru implementarea celor mai moderne realizări din domeniul informaticii şi tehnologiilor informaţionale în practică.

Cursul contine urmatoarele doua compartimente:

Elaborarea programelor pentru sistemele paralele de calcul cu memorie distribuită folosind standartul MPI (Message Passing Interface).

Elaborarea programelor pentru sistemele paralele de calcul cu memorie partajata folosind standartul Open MP .

Lucrarile de laborator :

Elaborarea programelor paralele pentru sisteme mixte cu memorie distrubuita si partajata.

Ciclul II Studii de masterat

C*alcul paralel* (Programul de master: „"Tehnologiile Produselor Soft-ware")

C*ontinutul cursului*: Sisteme si modele de calcul paralel. Performanţele algoritmului paralel. Utilizarea strategiilori de calcul de tipul Partitioning ,Pipelininla si Devide and Conquer elaborarea algoritmilori paraleli. Utilizarea pachetului de programe ScaLAPACK pentru inflementarea soft a algoritmilor paraleli pentru sisteme liniare de calcul.

l*ucrari de laborator:* Implementarea soft a algoritmilor paraleli pentru sisteme de calcul paralel cu memorie distribuita si partajata folosind tehnologii de programare MPI si OpenMP. Utilitati practice ale pachetulu

M*odelare matematică şi calcul performant* (Programul de master: Analiza statistică a datelor şi modelarea proceselor economico-financiare)

C*ontinutul cursului:* Conceputul al paralelismului nelimitat: principiul dublării, exemple de algoritmi cu înălţimea mică. Conceputul al paralelismului interior: proprietăţile şi avantajale principale, decompoziţia unui algoritm. Indicii de eficienţă. Algoritmii paraleli pentru rezolvarea numerică a problemelor din algebra liniară. Diferite posibilităţi de efectuare în mod paralel a operaţiilor de bază: suma matricelor, înmulţirea matricea-vector, înmulţirea matricea - matricea . *ijk* - forme.
Metode directe de rezolvare a sistemelor de ecuaţii algebrice liniare şi posibilităţile de realizare a algoritmilor în mod paralel (sisteme triunghiulare, Metoda Gauss, *LU* descompunere. *ijk* - forme, sisteme simetrice, sisteme tridiagonale.) Paralelismul în metode iterative.

Utilizarea clasterului in procesul de cercetare

The DAAD School „Parallel Programming in Mathematical Modeling” (3 courses, 40 hours), was organized by the Center for Education and Research in Mathematics and Computer Science (CECMI) of the Moldova State University in the framework of the DAAD (Germany) *Project “Center of Excellence for Applications of Mathematics”* during September 10-16, 2009.

In order to collaborate with Academy of Science of Moldova the Université Claude Bernard Lyon 1 (Laboratoire des Multimatériaux et Interfaces, Professor Dominique Luneau) have bought the quantum chemical programs such as Jaguar and Crystal. These ones and Siesta code were installed on the cluster of CECMI to perform the quantum chemical calculations. 2009-2010

MULTUMESC PENTRU ATENTIE !

Pentru informatii suplimentare despre modalitatile de utilizare a clasterului:

tel. 57-76-76 (239 bl. 4 USM)
e-mail: *boris.hancu@gmail.com*