

Test Funzionali e di scalabilità

Si richiede una serie di prove di prestazioni basate su un insieme di codici considerati rappresentativi dell'utilizzo della struttura HPC di CRESCO. L'insieme comprende:

- CPMD
- Fluent
- MCNP
- GROMACS

Si richiedono delle prove di scalabilità di esecuzione del codice a partire dal singolo nodo a numero crescente di nodi. Le prove dovranno essere a carico costante i.e. stesso numero di "problemi" per singolo processore.

Lo scopo dei presenti test è la verifica del corretto funzionamento dell'intero sistema e la corretta installazione dei codici. L'efficienza dei codici sul sistema non è oggetto del test. Il test si intenderà superato quando i casi proposti saranno eseguiti sul sistema HPC nel suo complesso e termineranno senza errori in tutte le configurazioni previste.

Test di scalabilità di CPMD

Il codice CPMD (www.cpmc.org) utilizza una base di onde piane e implementa il metodo Car Parrinello. Calcola la struttura elettronica dei materiali, effettua il rilassamento di una struttura atomica, esegue dinamica molecolare, calcola stati eccitati ed innumerevoli altre proprietà. È scritto in Fortran90 e utilizza MPI oppure 'mixed MPI/SMP' per la versione su macchine multiprocessori.

La parallelizzazione di CPMD può essere fatta su almeno due livelli. Il primo, basato su architetture a memoria distribuita, usa le MPI e raggiunge una buona efficienza su architetture fino a circa 200 CPU. Ad un livello superiore si può implementare una parallelizzazione a memoria condivisa (utilizzando le OpenMPI). I due metodi di parallelizzazione sono indipendenti e possono essere utilizzati insieme. In quest'ultimo caso si può raggiungere una buona efficienza su architetture a memoria distribuita ma con memoria condivisa su singolo nodo e con migliaia di CPU.

Lo scopo del presente benchmark è quello di accertare la corretta ed efficiente compilazione di CPMD sulla macchina da acquistare verificandone la scalabilità (verificando i tempi di esecuzione del job, i tempi per le comunicazioni internodo, i tempi di scrittura su disco, ecc.).

La società vincitrice della gara dovrà fornire tutte le istruzioni necessarie alla compilazione di CPMD nonché le librerie utilizzate.

Descrizione del benchmark

Il benchmark è costituito nel modo seguente.

Si lancia lo script `submit.sh` che a sua volta lancia lo script `cpmd.sh` (questo lancia l'eseguibile con l'opportuno file di input).

Si verifica la scalabilità della macchina rispetto a: 1) numero di processori impiegati; 2) dimensioni del sistema da studiare (cioè in questo caso il numero di atomi).

1) Per variare il numero di processori basta variare il comando 'procs' in `submit.sh` (sono preimpostate 5 righe per i casi procs=32, 256, 512, 1024, 2000).

2) Per variare le dimensioni del sistema bisogna solo selezionare l'opportuno file `diamond*.in` nello script `cpmd.sh` (cioè basta togliere il commento dalla riga che interessa).

Il test consiste nel far eseguire a CPMD 300 passi di dinamica molecolare per tre supercelle di diamante con i seguenti numeri di atomi: 216, 512, 1000.

I file di input sono

`diamond216WF.in`

`diamond216MD.in`

`diamond512WF.in`

`diamond512MD.in`

`diamond1000WF.in`

`diamond1000MD.in`

Prima di lanciare il job per la dinamica molecolare (file di input `diamond*MD.in`) bisogna sempre lanciare prima un job che prepari le funzioni d'onda (files di input `diamond*WF.in` selezionati in `cpmd.sh`).

Dopodiché si lanciano i job con i files `diamond*MD.in` (selezionati in `cpmd.sh`) e selezionando il numero di processori in `submit.sh`.

Nel caso si debbano eseguire più run sullo stesso sistema materiale ma variando il numero di processori, conviene salvare il file `RESTART.1` creato con il file di input `diamond*WF.in` in modo da riutilizzarlo per i successivi lanci di CPMD con il file di input `diamond*MD.in`.

Nei files `diamond*.out` sono riportate le informazioni sui tempi di esecuzione.

I casi da analizzare sono riportati nella tabella 1.

TABELLA 1

Numero Atomi	Numero Processori
216	16
216	32
216	64
216	128
216	256
216	512
512	256
512	512
512	1024
1000	512
1000	1024
1000	2000

I file di input (*.in) per ogni caso da analizzare ed i files submit.sh, cpmd.sh e C_MT_GIA_BLYP.psp (questo file contiene gli pseudopotenziali ed è necessario per ogni caso da analizzare) saranno forniti da ENEA in fase di collaudo.

Test di scalabilità di Fluent

Nove applicazioni industriali CFD con dimensioni che oscillano da 32,000 a 10,000,000 celle sono state scelte per dimostrare le prestazioni del codice su una varietà di piattaforme hardware. Le prestazioni di un codice CFD dipendono da numerosi fattori tra cui dimensione e topologia della griglia di calcolo, i modelli implementati, la partizione della mesh, i compilatori; oltre che da caratteristiche dell'hardware su cui è eseguita la simulazione. I problemi scelti rappresentano una gamma di simulazioni tipiche di quelle che potrebbero essere trovati nelle applicazioni industriali.

A causa di grandi richieste computazionali di molti dei problemi applicativi, è importante fissare un numero d'iterazioni per effettuare le valutazioni; inoltre per le simulazioni in parallelo, è stata predefinita la partizione della griglia utilizzata

Al fine di rendere omogenee le valutazioni tra i diversi casi analizzati sono stati definiti alcuni parametri principali:

il "Rating", definito come il numero di benchmarks che possono essere eseguiti in sequenza su una macchina nel periodo di 24 ore. È calcolato dividendo il numero di secondi in un giorno (86400) per il numero di secondi richiesti per eseguire il benchmark. Un elevato Rating valutazione vuole dire prestazioni più elevate.

“Speedup”: è il rapporto tra “wall-clock-time” richiesto per completare un calcolo prefissato utilizzando un singolo processore, e quello del calcolo equivalente eseguito su macchine simultanee. Il suo valore varia da 0 al numero di processori utilizzati per l’esecuzione parallela. Quando lo speedup è uguale al numero di processori utilizzati, lo speedup si dice perfetto o lineare. Lo speedup può superare il numero di processori, in questo abbiamo un super-linear speedup che è giustificato da un surplus di memoria disponibile.

“Efficiency”: rappresenta lo “speedup” normalizzato con il numero di processori utilizzati ed è espresso in percentuale. Essa indica l’utilizzazione della CPU durante un calcolo parallelo. Un’efficienza del 100% indica che ogni CPU è completamente occupata dal calcolo in atto e corrisponde ad uno speedup lineare. Un’efficienza del 60% indica che ogni CPU viene utilizzata per il calcolo solamente per il 60% del tempo; il tempo restante è utilizzato per altre funzioni come per esempio la comunicazione parallela.

“Redzone”: è definita come la regione con un’efficienza maggiore del 66%.

“Redzone Rating”: è definito come il massimo rating raggiunto nella Redzone, fissati il benchmark e la macchina, attraverso una simulazione in parallelo.

“Peak Rating”: è definito come il massimo rating raggiunto, fissati il benchmark e la macchina, attraverso una simulazione in parallelo con numero qualsiasi di processori.

Una volta installata la versione del codice scelta, (<http://www.fluent.com/>) sarà necessario leggere con Fluent i files .scm (files “scheme”) che contengono le istruzioni per la lettura dei casi da testare.

La tabella che segue illustra le caratteristiche principali dei 9 casi su cui eseguire i benchmark, disponibili sul sito di FLUENT:

Class	Benchmark	Cells	Mesh	Models	Solver	Description
small	FL5S1	32,000	hexahedral	ke	segregated implicit	turbulent flow in a bend
	FL5S2	32,000	hexahedral	ke	coupled implicit	turbulent flow in a bend
	FL5S3	89,856	hexahedral	ke	coupled implicit	flow in a compressor, rotor 37

medium	FL5M1	155,188	tetrahedral	ke 6spe react DPM P1	segregated implicit	coal combustion in a boiler, with particle tracking
	FL5M2	242,782	hybrid, hanging-node	ke	segregated implicit	turbulent flow in an engine valveport
	FL5M3	352,800	hexahedral	ke 6spe react	segregated implicit	combustion in a high velocity burner
large	FL5L1	847,746	hexahedral	ke	coupled explicit	transonic flow around a fighter
	FL5L2	3,618,080	hybrid	RNG ke	segregated implicit	external aerodynamics around a car body
	FL5L3	9,792,512	hexahedral	RSM	segregated implicit	turbulent flow in a transition duct

Tabella delle prestazioni da compilare a cura del fornitore. La selezione dei casi test da effettuare sarà concordata con ENEA, ma al minimo dovrà essere eseguito un test per tutte le configurazioni riportate nella tabella seguente:

Sistema	Caso FL_XXYY					
	FLUENT Version	CPUs or Cores	Machines or Blocks	Rating	Speedup	Efficiency
Sezione 1 da ripetere su tutti e 32 I nodi		serial				
		1				
		4				
		16				
Sezione 2		serial				

Sezione 2		1				
		8				
		32				
		64				
		128				
		512				
		2048				

Test di scalabilità di MCNP5 (versione 5)

MCNP è un codice di uso generale che tratta il trasporto singolo e accoppiato per neutroni, elettroni e fotoni . Il codice descrive una configurazione tridimensionale arbitraria per mezzo di celle geometriche definite da superfici di primo, secondo e tori ellittici del quarto ordine. Fra le varie caratteristiche, nell'interesse di questa specifica, vanno citate la capacità di includere una sorgente di particelle definita dall'utente e la capacità di una raccolta estesa di dati di sezioni d'urto. La versione corrente è la **5**. Il codice è sviluppato dal gruppo **X-5 Monte Carlo Team - Diagnostics Applications Group** dei **Los Alamos National Laboratory**. Il codice è commerciale e può essere acquistato presso la stessa istituzione.

Le configurazioni geometriche usualmente descritte sono abbastanza complesse e possono presentarsi non completamente ottimizzate. Nella maggior parte delle simulazioni si fa uso di sorgenti di particelle con distribuzione spaziale, angolare ed energetica definite dall'utente. Anche il database delle sezioni d'urto può variare in dimensioni a seconda della applicazione.

Per questi motivi e allo scopo di omogeneità si chiede di utilizzare come insieme di casi per la determinazione delle funzionalità della configurazione HPC un sottoinsieme (11) dei casi (42) di regressione che il software di installazione di MCNP5 esegue al completamento per misurarne la bontà. I casi sono individuati dai nomi di file da **inp01** a **inp06** e da **inp10** a **inp14**. Si richiede di scalare i problemi a **1, 10, 200, 2000 e 2500** processori. Per ottenere una legge di scala a carico costante il numero di particelle (parametro **nps** nei file di input) va aumentato proporzionalmente al numero di processori. I test dovranno essere effettuati sulla Sezione 1, Sezione 2 e Sezione 1+ Sezione 2 del sistema HPC. La sequenza dettagliata dei test verrà definita in fase di collaudo tra la ditta fornitrice ed i responsabili ENEA della fornitura.

Test di scalabilità di GROMACS

GROMACS è un software per il modelling molecolare, sviluppato originariamente all'interno di una collaborazione scientifica tra l'Università di Groningen (prof. Berendsen) e l'ETH di Zurigo (prof. Van Gunsteren). GROMACS è attualmente divenuto un codice di riferimento di questa classe di attività, sia in ambito di ricerca scientifica che di ricerca industriale. ENEA è stata coinvolta, all'interno di un progetto dell'FP3 della UE, nella parallelizzazione del codice seriale Gromos, proposto all'inizio degli anni 80.

GROMACS è un software che utilizza propri force fields per la descrizione delle forze interatomiche; tale force field, particolarmente accurato per la descrizione dei legami all'interno di sistemi molecolari complessi, lo rendono particolarmente accurato per l'analisi delle strutture delle proteine.

Il Dipartimento Biotecnologie è particolarmente interessato all'utilizzo di strumenti di modelling molecolare che supportano le sue attività di proteomica, di progettazione di anticorpi e di immunocongiugati.

GROMACS is a versatile package to perform molecular dynamics, i.e. simulate the Newtonian equations of motion for systems with hundreds to millions of particles.

Nel seguito si dettagliano (in inglese) alcune caratteristiche del codice GROMACS, evinti dal sito di download del codice (www.gromacs.org)

GROMACS supports all the usual algorithms you expect from a modern molecular dynamics implementation, (check the online reference or manual for details), but there are also quite a few features that make it stand out from the competition:

- GROMACS provides *extremely high performance* compared to all other programs. A lot of algorithmic optimizations have been introduced in the code; we have for instance extracted the calculation of the virial from the innermost loops over pairwise interactions, and we use our own software routines to calculate the inverse square root. The innermost loops are generated automatically in either C or Fortran at compile time, with optimizations adopted to your architecture. Assembly loops using SSE and 3DNow! multimedia instructions are provided for x86 processors, separate versions using all x86-64 registers are used on Opteron x86-64 and Xeon EM64t machines. This results in exceptional performance on inexpensive PC workstations, and for Pentium IV/Opteron processors there are also SSE2 double precision assembly loops. There are new manually tuned assembly loops for ia64 (both single and double precision), and last but certainly not least we have written Altivec assembly loops both for Linux and Mac OS X. Gromacs is normally 3-10 times faster than any other program; check the article in Journal of Molecular Modeling (reference can be found under resources) for a comparison benchmark.
- GROMACS is user-friendly, with topologies and parameter files written in clear text format. There is a lot of consistency checking, and clear error messages are issued when something is wrong. Since the C preprocessor is used, you can have conditional parts in your topologies and include other files. You can even compress most files and GROMACS will automatically pipe them through gzip upon reading.

There is no scripting language - all programs use a simple interface with command line options for input and output files. You can always get help on the options by using the **-h** option, or use the extensive manuals provided free of charge in electronic or paper format. There is also an integrated graphical user interface available for all programs.



As the simulation is proceeding, GROMACS will continuously tell you how far it has come, and what time and date it expects to be finished.

Both run input files and trajectories are independent of hardware endian and can thus be read by any version GROMACS, even if it was compiled using a different floating-point precision. All files from GROMACS 2.0 can further be used in the new version 3.

GROMACS can write coordinates using lossy compression, which provides a very compact way of storing trajectory data. The accuracy can be selected by the user.

GROMACS comes with a large selection of flexible tools for trajectory analysis - you won't have to write any code to perform routine analyses. The output is further provided in the form of finished Xmgr/Grace graphs, with axis labels, legends, etc. already in place!

A basic trajectory viewer that only requires standard X libraries is included, and several external visualization tools can read the GROMACS file formats.

- GROMACS can be run in parallel, using standard MPI communication.
- GROMACS contains several state-of-the-art algorithms that make it possible to extend the time steps in simulations significantly, and thereby further enhance performance without sacrificing accuracy or detail.

The package includes a fully automated topology builder for proteins, even multimeric structures. Building blocks are available for the 20 standard aminoacid residues as well as some modified ones, the 4 nucleotide and 4 deoxynucleotide residues, several sugars and lipids, and some special groups like hemes and several small molecules.

There is ongoing development to extend GROMACS with interfaces both to Quantum Chemistry and Bioinformatics/databases.

GROMACS is *Free Software*, available under the **GNU** General Public License.

The site where the code can be downloaded is www.gromacs.org

Test-cases

I test cases inclusi nella prova si riferiscono alla simulazione di una specifica proteina, la malato-hydrogenase. Sono stati preparati due input relativi alla stessa molecola e saranno forniti dall'ENEA in fase di collaudo. Il primo (T1) il sistema è composto da una sola molecola di malato-hydrogenase con una sfera di solvatazione di molecole d'acqua. Il secondo (T2) le molecole di malato-hydrogenase sono due e la shell di solvatazione di molecole d'acqua è notevolmente più ampia. Nella Tabella sono riportate le caratteristiche dei due inputs (numero totale di atomi, range del numero dei core dell'architettura all'interno del quale effettuare i tests di scalabilità).

Test	Number of residues (number of water molecules)	Number of cores	Simulation time-period
T1	327 (6369)	1-128-512	1 nsec
T2	654 (61404)	128-512-2048	1 nsec

