

### **Algoritmi di simulazione di dinamica molecolare per l'era exascale**

Guardando la lista top500 dei computer più potenti al mondo si può vedere che esistono computer di centinaia di migliaia di cores che sviluppano una potenza di calcolo di qualche petaflop ([www.top500.org](http://www.top500.org)). In Europa è appena partito il progetto PRACE con un primo supercomputer già operativo in Germania (IBM, No. 9 nella top500), un secondo pronto in Francia (Bull, No. 6 nella top500) e un terzo in via di definizione al Cineca a Bologna. E intanto anche se siamo appena entrati nel petaflop, si parla già di exascale, computer 1000 volte più potenti di quelli oggi disponibili, quindi con diversi milioni di cores.



Parallelamente all'arrivo di nuovi supercomputer sono ancora pochi i software che riescono a utilizzare al meglio queste infrastrutture di supercalcolo, che riescono cioè a utilizzare decine di migliaia di cores per ogni singolo calcolo con un guadagno vicino al numero di cores utilizzati. Tra questi ci sono software di fluidodinamica e struttura elettronica, come ad esempio il CPMD che scala linearmente sino a più di 16000 cores (test condotti

a Juelich).

Invece nel campo della dinamica molecolare classica, ad esempio per le simulazioni di proteine, non si riescono ancora a utilizzare migliaia di cores. Utilizzando anche i migliori software disponibili (GROMACS e NAMD), si riesce a scalare bene con sistemi di massimo 1000/2000 atomi per core, precludendo la possibilità di utilizzare appunto migliaia di core per i sistemi di maggior interesse biologico, quelli sino a 100.000 atomi.

Una concreta possibilità per queste simulazioni è quella di sviluppare degli algoritmi cosiddetti "embarrassingly parallelized": algoritmi Monte Carlo che lavorano su decine di repliche e che scambiano pochi dati e quindi scalano in maniera lineare. Questi algoritmi sono facilmente implementabili una volta che si possiede un codice di dinamica molecolare che scali bene su poche decine di cores (<32). E' necessario implementare un interfaccia che controlli il programma generando centinaia di repliche e implementi degli scambi sulle repliche. Se si ha un codice che scala su 32 cores, volendo usare 10000 cores si possono lanciare più di 300 repliche.

Lo sviluppo di questi algoritmi, benché semplice dal punto di vista teorico, prevede la soluzione di alcuni problemi pratici quali:

1. il controllo dell'implementazione dei calcoli quando si ha a che fare con centinaia di repliche
2. l'uso di generatori di numeri casuali efficaci
3. il trattamento di una mole di dati enorme per l'analisi

Delle tesi di **laurea magistrale** sono disponibili per risolvere i problemi sopra elencati. Si richiede la conoscenza dell'inglese e almeno di un linguaggio di programmazione (C,C++,fortran95). Abbiamo un codice MD che scala su poche decine di cores e il nostro obiettivo è quello di chiedere un progetto PRACE-fase preparatoria per adattare il programma ai supercomputer e nel giro di 2 anni presentare un progetto PRACE per simulare la dinamica di proteine.

Contattare: Matteo Ceccarelli c/o Dipartimento di Fisica ([matteo.ceccarelli@dsf.unica.it](mailto:matteo.ceccarelli@dsf.unica.it))

Andrea Bosin c/o Dipartimento di Matematica e Informatica ([andrea.bosin@dsf.unica.it](mailto:andrea.bosin@dsf.unica.it))

**Bibliografia:** *J. Chem. Theory Comput.* **2010**, 6, 2804–2808; *J. Chem. Theory Comput.* **2009**, 5, 1624–1631