

Valutazione delle prestazioni del software scientifico parallelo in ambienti virtualizzati

Prof. Paolo Cremonesi

Andrea Sansottera

Introduzione

Il progetto ha lo scopo di valutare le prestazioni di alcune routine tipicamente utilizzate in ambito scientifico quando eseguite in macchine virtuali. In particolare, consideriamo routine per le quali sono disponibili versioni parallele, implementate secondo il paradigma *shared memory* oppure secondo il paradigma *message passing*. Le prestazioni misurate in assenza di virtualizzazione saranno considerate come *baseline* e dovranno essere comparate con quelle ottenute utilizzando diversi hypervisor di tipo 1 (*bare metal*) sul quale gireranno, tramite una tecnica nota come *virtual SMP* (symmetric multi-processing), macchine virtuali dotate di più virtual CPUs. Nel caso di routine per le quali è disponibile un'implementazione basata su *message passing*, andranno comparate le prestazioni ottenute con una sola virtual machine e quelle ottenute distribuendo il carico su più virtual machines.

Ambiente di test

I test verranno effettuati su una macchina presente nel laboratorio NECST, dotata di un processore Core i7 920 e 12 GB di memoria RAM. La CPU dispone di quattro core fisici e otto processori logici, grazie all'implementazione del simultaneous multi-threading (*hyper-threading* nella terminologia di Intel). Lo studente avrà a disposizione un disco rigido sul quale dovrà installare tutti i software necessari all'esecuzione dei test.

Le routine implementate con paradigma shared-memory andranno testate eseguendo una sola macchina virtuale con a disposizione 8 virtual CPU.

Le routine implementate con paradigma message-passing andranno testate eseguendo:

- 1 virtual machine con 8 virtual CPU e 10 GB di RAM
- 2 virtual machine con 4 virtual CPU e 5 GB di RAM
- 4 virtual machine con 2 virtual CPU e 2.5 GB di RAM

Poiché le librerie di algebra lineare non beneficiano in genere del simultaneous multi-threading, il numero totale di processi da utilizzare è 4 (4x1 VM, 2x2 VM e 1x4 VM). La comunicazione tra le macchine virtuali dovrà essere effettuata su una rete virtuale tra le VM (non attraverso un'interfaccia virtuale bridged).

Il sistema operativo di riferimento è Linux Debian 7 (noto anche come Debian Wheezy), la versione "stable" della distribuzione Debian. Questo sistema operativo verrà utilizzato sia per stabilire le prestazioni native (la baseline), sia come sistema operativo guest installato nelle macchine virtuali. La libreria MPICH2, disponibile tra i pacchetti standard di Linux Debian 7, verrà utilizzato per l'esecuzione delle routine implementate con paradigma message-passing. Nel caso di esecuzione su più macchine virtuali (2 o 4) può essere usato il comando `mpiexec.hydra`, che effettua l'autenticazione tramite SSH (deve essere configurata l'autenticazione tramite chiave privata/pubblica anziché tramite password).

Gli hypervisor considerati sono le ultime versioni stabili di:

- Microsoft Hyper-V

- VMware ESX
- Citrix Xen Server

Versioni gratuite di questi hypervisor sono disponibili direttamente dal sito del venditore oppure tramite il programma Microsoft DreamSpark a cui aderisce il Politecnico di Milano.

Benchmarks

BLAS e LAPACK sono due librerie standard per l'algebra lineare, utilizzate in molti software scientifici. BLAS implementa le operazioni algebriche tra scalari, vettori e matrici. Le routine BLAS sono classificate in base alla *memory-intensity* (rapporto tra operazioni di memoria e operazioni floating point): le routine *level-1* eseguono operazioni tra scalari e vettori (memory bound), le routine *level-2* eseguono operazioni vettore-matrice (memory bound sulla maggior parte delle architetture), le routine *level-3* eseguono operazioni matrice-matrice (CPU bound se le cache sono usate in modo appropriato). LAPACK è basata su BLAS e implementa le fattorizzazioni di matrice, la soluzione di sistemi lineari e il calcolo di auto-valori e auto-vettori. Le routine di LAPACK sono classificate in quelle di più basso livello (*computational routines*) e in quelle di più alto livello (*driver routines*). Sono disponibili diverse implementazioni compatibili di BLAS e LAPACK, ottimizzate per diverse architetture. La maggior parte dei fornitori offrono sia una versione single-thread che una versione multi-thread, implementata in genera utilizzando il multi-threading nelle routine BLAS level-3. Poiché le implementazioni usano la stessa interfaccia, è possibile cambiare implementazione semplicemente facendo il linking ad una diversa implementazione. Il codice C e C++ può usare le interfacce CBLAS e LAPACKE per chiamare le routine BLAS e LAPACK, le cui interfacce sono pensate per il linguaggio Fortran.

PLASMA è un progetto di ricerca il cui scopo è realizzare libreria di algebra lineare ad alte prestazioni per le architetture shared-memory. Si basa su una implementazione single-thread di BLAS. Offre routine con funzionalità equivalenti a quelle di BLAS level-3 e LAPACK. L'interfaccia è leggermente diversa dall'interfaccia standard.

PBLAS e ScaLAPACK sono librerie parallele implementate tramite il paradigma message-passing che offrono funzionalità equivalenti a BLAS e LAPACK, rispettivamente, ma con un'interfaccia diversa. Si basano su una libreria MPI (interfaccia standard per programmazione con il paradigma message-passing) e una implementazione single-thread o multi-thread di BLAS e LAPACK.

Dovranno essere testate diverse routine BLAS e LAPACK e gli equivalenti forniti da PLASMA e PBLAS/ScaLAPACK. Le routine dovranno essere chiamate da un codice C++, relazato ad-hoc dallo studente, partendo da un template funzionante da noi fornito.

- dgemm (general matrix-matrix multiply)
 - Cambiando le dimensioni delle matrici
- dgetrf (LU factorization of a general matrix)
 - Cambiando la dimensione della matrice (quadrata)
- dpotrf (Cholesky factorization of a symmetric positive matrix)
 - Cambiando la dimensione della matrice (quadrata)
- dgeqrf (QR factorization of a general matrix)
 - Cambiando le dimensioni della matrice
- dgesv (system solving with general matrix using LU factorization)
 - Cambiando la dimensione della matrice (quadrata)
 - Cambiando il numero di vettori sul lato destro

- dposv (system solving with symmetric positive matrix using Cholesky factorization)
 - Cambiando la dimensione della matrice (quadrata)
 - Cambiando il numero di vettori sul lato destro
- dgels (system solving with general matrix using QR factorization)
 - Cambiando le dimensioni della matrice (ovvero il numero di equazioni e di variabili)
 - Cambiando il numero di vettori sul lato destro
- dgesvd (SVD factorization of a general matrix)
 - Cambiando le dimensioni della matrice

La documentazione di queste routine è fornita con la libreria Intel MKL.

Le routine di PLASMA equivalenti hanno il prefisso PLASMA_*. La documentazione è disponibile sul sito della libreria¹.

Le routine di PBLAS e ScaLAPACK equivalenti hanno il prefisso "p" (per esempio, pdgemm, pdgesv, ecc.). La documentazione è fornita anche con la libreria Intel MKL, che implementa anche PBLAS e ScaLAPACK.

Requisiti richiesti

Lo studente dovrà avere una certa dimestichezza con il sistema operativo Linux. Sarà inoltre necessaria una conoscenza di base dei linguaggi C/C++ e degli strumenti di sviluppo disponibili su Linux, necessarie per apportare piccole modifiche al codice dei benchmark e compilarli. Infine sarà necessario utilizzare uno strumento come SVN o GIT in modo da condividere il repository del codice benchmark con il revisore. Le conoscenze specifiche necessarie per l'utilizzo degli hypervisor devono essere apprese autonomamente durante la realizzazione del progetto.

Impegno richiesto

Sarà richiesta una breve presentazione settimanale nel quale verranno chiaramente riportati i progressi fatti dallo studente e discussi eventuali dubbi sul proseguire dei lavori. Per l'esecuzione dei test, sarà necessaria la presenza nel laboratorio NECST.

¹ <http://icl.cs.utk.edu/plasma/docs/index.html>