Corso di Algoritmi Avanzati—modulo 2

Secondo progetto di programmazione—Anno Accademico 2014/2015

Moreno Marzolla

Versione 1.0 del 26/4/2015

Prima versione di questo documento

Descrizione del progetto

Scopo del progetto è l'implementazione di un algoritmo parallelo per architetture a memoria distribuita, usando il linguaggio C con MPI, per il calcolo della distribuzione della distanza angolare tra coppie di punti nello spazio. Questa metrica trova applicazioni in campo astronomico, dato che lo studio della distanza angolare tra galassie fornisce informazioni sulla natura e origine della materia oscura.

Consideriamo un insieme di N punti aventi coordinate (x_i, y_i, z_i), relativamente ad una origine arbitraria, $i = 0, \ldots, N-1$. Scelti due punti i e j con $i < j$, la loro distanza angolare θ_{ij} è definita come:

$$\theta_{ij} = \arccos\left(\frac{x_i x_j + y_i y_j + z_i z_j}{\sqrt{(x_i^2 + y_i^2 + z_i^2)(x_j^2 + y_j^2 + z_j^2)}}\right)$$

dove arccos() è l'inversa della funzione coseno (in linguaggio C è la funzione `acos()`, per usare la quale è necessario linkare il programma con l'opzione `-lm` per includere la libreria matematica).

θ_{ij} assume valori compresi tra 0 e π. Vogliamo determinare la distribuzione di θ_{ij} per $0 \le i < j < N$, che consiste in un istogramma da calcolare come segue. Si partiziona l'intervallo $[0, \pi)$ (aperto a destra) in K sottointervalli di uguale ampiezza $I_k = [\pi k/K, \ \pi(k+1)/K)$. Per ogni $k = 0, \ldots K-1$ si determina il numero F_k di valori θ_{ij} che cadono all'interno di I_k. Quindi F_0 sarà il numero di coppie di punti aventi distanza angolare in $[0, \pi/K)$, F_1 è il numero di coppie di punti aventi distanza angolare in $[\pi/K, \ 2\pi/K)$, e F_{K-1} è il numero di coppie di punti aventi distanza angolare in $[\pi(K-1)/K, \ \pi)$. Prestare attenzione al fatto che $\theta_{ij} = \theta_{ji}$ (la distanza angolare tra i punti i e j è uguale alla distanza angolare tra i punti j e i), ma ogni coppia deve essere contata una sola volta, per cui è necessario considerare solo i valori θ_{ij} con $i < j$. L'array $F_0, \ldots F_{K-1}$ è il risultato della computazione.

Dato che ci sono in tutto $N(N-1)/2$ distanze angolari, si deve avere

$$\sum_{k=0}^{K} F_k = \frac{N(N-1)}{2}$$

Il programma da realizzare riceve tramite riga di comando tre parametri:

1. il primo parametro è il valore K (intero positivo) che indica il numero di sottointervalli in cui si divide l'intervallo $[0, \pi)$. Indicativamente si puo' usare $K = 100$, ma il programma deve poter funzionare con qualsiasi valore di K.
2. il secondo parametro indica il numero N di punti. Si puo' assumere il vincolo che N sia (molto) maggiore del numero P di processi MPI.
3. Il terzo parametro è il nome di un file di testo contenente le coordinate dei punti. Il file ha almeno N righe, ciascuna delle quali contiene i valori delle tre coordinate x, y, z, separati da spazi o tabulazioni; le coordinate sono numeri reali arbitrari. Sulla pagina web del corso è presente un file di esempio, contenente le coordinate di 119613 stelle. Il valore di N passato sulla riga di comando puo' essere minore del numero di righe del file; in tal caso il

programma legge le prime N righe e ignora le altre.

Al termine dell'esecuzione, il programma stampa a video K valori interi, uno per riga, che rappresentano i valori F_0, … F_{K-1} calcolati. Chi lo desidera puo' inserire nella relazione un plot dei valori ottenuti sotto forma di istogramma; cio' non è comunque obbligatorio.

Modalità di svolgimento del progetto

- Il progetto deve essere svolto individualmente. Non è consentito condividere codice o relazione con altri studenti, né utilizzare software disponibile in rete. Non è consentito discutere il progetto con altri.
- Il progetto deve essere realizzato in un singolo file sorgente chiamato `angdist.c`. Il sorgente deve essere adeguatamente commentato. All'inizio del file sorgente deve essere incluso un commento che riporta cognome, nome e numero di matricola dell'autore/autrice, nonché il comando da usare per la compilazione (vedi punto successivo).
- Il progetto deve essere implementato in linguaggio C come applicazione a riga di comando, facendo uso delle librerie MPI. Il progetto deve essere compilabile ed eseguibile senza errori su una delle macchine Linux dei laboratori studenti (sistema operativo Debian GNU/Linux 7.7), oppure sull'immagine dei laboratori virtuali (http://www.virtlab.unibo.it/index.html) mediante il comando

  `mpicc -Wall angdist.c -o angdist` *`[eventuali flag di compilazione]`*

  Indicare nel commento iniziale del sorgente e/o nella relazione la riga di comando corretta per la compilazione nel caso siano richiesti altri flag, ad esempio nel caso in cui sia necessario includere librerie di sistema. Nota: sulle macchine del laboratorio è installato OpenMPI (pacchetti `openmpi-bin`, `libopenmpi-dev` ed `openmpi-doc` per la documentazione).
- Per verificarne la correttezza, il programma verrà eseguito mediante il comando

  `mpirun -n` *`P`* `./angdist` *`K N inputfile`*

  dove P indica il numero di processi MPI da creare, K indica il numero di intervalli (bin) dell'istogramma, N il numero di stelle da considerare, e `inputfile` rappresenta il nome di un file di input contenente le coordinate di almeno N punti; solo le prime N terne di coordinate verranno considerate. Tutte le istanze saranno lanciate localmente, in modo da evitare ogni problema con la rete.
- Si puo' assumere che il numero N di punti sia sempre (molto) maggiore del numero di processi P. Si puo' assumere che N sia multiplo di P, ma in questo caso il progetto verrà penalizzato rispetto a soluzioni che funzionano correttamente anche nel caso in cui N non sia multiplo di P.
- Assieme al sorgente deve essere consegnare una relazione in formato pdf in un file chiamato `relazione.pdf`. La relazione non deve superare la lunghezza di **cinque facciate** in formato A4 (font non inferiore a 10 punti). **Nel computo delle cinque facciate rientrano TUTTE le facciate del documento consegnato** (pagina del titolo, eventuale indice o altro); suggerisco di non sprecare spazio dedicando una facciata al titolo o all'indice, che in un documento così corto sarebbero comunque superflui. Indicare il proprio cognome, nome, e numero di matricola all'inizio della relazione.
- La relazione deve descrivere ad alto livello l'implementazione con particolare riguardo alla strategia di parallelizzazione adottata e gli eventuali limiti e/o potenzialità di tale strategia.

La relazione **deve** includere una sintetica analisi sperimentale della scalabilità dell'algoritmo realizzato, mostrando i grafici di speedup ed efficienza. Non è richiesto che l'analisi di scalabilità venga effettuata sulle macchine del laboratorio. È possibile effettuare i test lanciando tutte le istanze localmente (anche in numero superiore al numero di core disponibili); ovviamente non terrò conto dei tempi di esecuzione in termini assoluti, ma solo del fatto che l'analisi di speedup ed efficienza sia svolta correttamente.

Scadenza e modalità di consegna

Il progetto e la relazione devono essere consegnati entro le ore **12:00 (mezzogiorno)** di **martedì 26 maggio 2015**.

Ricordo che è necessario ottenere una valutazione positiva del progetto per poter partecipare alla prova scritta del modulo 1. La valutazione verrà comunicata via mail, e se positiva resterà valida per l'intero anno accademico 2014/2015, ossia fino all'appello di esami di gennaio/febbraio 2016 (compreso).

La consegna deve avvenire inviando una mail dal proprio indirizzo istituzionale @studio.unibo.it a moreno.marzolla@unibo.it con subject

Consegna Progetto Algoritmi Avanzati 2014/2015

Nella mail vanno indicati cognome, nome, numero di matricola del mittente. Alla mail deve essere allegato un archivio in formato `.zip` oppure `.tar.gz` contenente il sorgente e la relazione in formato pdf; chi lo desidera può includere altri files, ad esempio un `Makefile` per la compilazione (non obbligatorio). L'allegato sarà denominato con il cognome dell'autore (es., `Rossi.zip` oppure `Rossi.tar.gz`), e conterrà una directory con lo stesso nome (es., `Rossi/`) contenente a sua volta il sorgente e la relazione in formato pdf.

Nota per gli utenti Apple: gli archivi .zip prodotti sotto MacOSX spesso includono una directory `.DS_Store/`. L'archivio che viene consegnato deve **essere privo** di tale directory.

Valutazione dei progetti

Per ottenere una valutazione positiva è necessario che il programma compili sulle macchine Linux del laboratorio studenti e funzioni correttamente secondo le specifiche indicate in questo documento. Ulteriori elementi di cui si terrà conto:

- Efficienza della strategia di parallelizzazione adottata;
- Generalità della soluzione implementata;
- Qualità della relazione, in termini di chiarezza, correttezza e presentazione del contenuto;
- Qualità del codice, in termini di semplicità e chiarezza. Verrà penalizzato il ricorso a micro-ottimizzazioni inutili, nonché codice ridondante o inutilmente complesso da capire.