

Specifiche di presentazione dei progetti

Premessa

Questo documento introduce le modalità di sviluppo dei progetti assegnati nell'ambito del corso di Calcolo Parallelo e Distribuito del Corso di Laurea Magistrale in Ingegneria Informatica dell'Ateneo Roma Tre nell'anno accademico 2021/22.

In tale anno sono stati assegnati a 12 gruppi di due o tre studenti 9 progetti di ottimizzazione e parallelizzazione del codice Julia nel contesto del prototipo, già sviluppato, di una pipeline complessa di algoritmi.

Obiettivo generale dei progetti

L'obiettivo generale dell'insieme di progetti è (1) il calcolo dell'arrangement (partizione) dello spazio euclideo indotta da una collezione di complessi cellulari (ambito disciplinare: geometric modeling), e (2) la conseguente generazione di un'algebra finita di atomi corrispondenti alle colonne delle matrici degli operatori di bordo e cobordo di grado massimo in 2D e/o in 3D.

Utilizzo: Algebre Solide in contesti CAD e BIM

Calcolati i vettori booleani corrispondenti agli insiemi di atomi che definiscono la struttura dei complessi generatori dell'arrangement, qualunque espressione booleana (CSG - Constructive Solid Geometry), tra gli oggetti solidi di ingresso (descritti da rappresentazioni al contorno), può essere scritta come espressione booleana tra i corrispondenti vettori binari, e risolta per componenti dal compilatore stesso del linguaggio Julia.

Modalità e tempistica della presentazione dei progetti.

Ogni progetto sarà sviluppato e presentato attraverso tre consegne in sequenza, la cui consegna dovrà avvenire entro le date specificate di seguito.

- Consegnna Studio preliminare: Venerdì 29-04-2022
- Consegnna Studio esecutivo: Venerdì 20-05-2022
- Consegnna Studio definitivo: Venerdì 10-06-2022

Ciascuna consegna consisterà di: (A) una breve relazione in formato markdown (pandoc); (B) tre slides grafiche esportate in pdf; (C) uno o più notebook julia.

Tipicamente ogni consegna successiva modificherà la precedente ovunque necessario, correggendo e/o introducendo argomenti e paragrafi preesistenti, e precisando con maggiore dettaglio il lavoro implementativo e/o il codice presentato nei notebook.

Studio preliminare

- A. **Illustrazione del problema** che il codice Julia oggetto del progetto deve ottimizzare, utilizzando il livello di dettaglio preferito, ma mantenendosi entro il limite di 2000 caratteri;
- B. **Illustrazione del grafo delle dipendenze** della base di codice oggetto di studio. Questo è un grafo orientato disegnato sul piano, dove ogni arco orientato rappresenta una chiamata di funzione (v_1, v_2) , dove v_1 è la funzione chiamante, e v_2 è la funzione chiamata. I nodi avranno un'etichetta (label) corrispondente al nome della funzione. gli archi avranno una etichetta numerica consecutiva (es: 1, 2, 3, ...) corrispondente all'ordine delle chiamate dal nodo loro origine. Se necessario, potete disegnare anche una versione compressa o semplificata del grafo.
- C. Il notebook dovrà contenere:

URL della base di codice del proprio progetto.

Questa sarà ottenuta:

1. clonando localmente il repo di riferimento *LinearAlgebraicRepresentation.jl*
2. rimuovendo tutti i file non utilizzati dal progetto (in `src/`, `temp/`, `test/`, `doc/`, etc)
3. non rimuovendo alcuni pochi esempi in grado di girare nella bse di codice rimasta.

La metodologia da seguire è semplice:

1. scegliere due/tre esempi (parametrici, in modo da girare producendo dati molto diversi per dimensione);
2. verificare che siano eseguibili nel repo iniziale;
3. creare una copia locale del repo iniziale, da utilizzare per la rimozione dei file
4. ogni poche rimozioni, verificare che gli esempi continuino a girare nel repo semplificato. tornare indietro quando non girano.
5. verificare sempre graficamente (ViewerGL) i risultati di ogni operazione sul codice e sui dati

Studio esecutivo

Studio definitivo