

Politecnico di Milano
Facoltà di Ingegneria dell'Informazione

Estratto della tesi di Laurea Specialistica in Ingegneria Informatica “Sperimentazione e valutazione di OpenCL su piattaforme parallele per un metodo a volumi finiti per le acque basse ” di Stramare Roberto (Matricola: 718754)

Relatore: prof. Luca Oddone Breveglieri
Correlatore: prof. Edie Miglio

Anno accademico: 2010/2011 - Appello di aprile 2012

La simulazione di problemi scientifici e ingegneristici richiede grandi capacità computazionali, questa tesi mostra l'uso dello standard OpenCL (Open Computing Language) per il calcolo su hardware parallelo di diverso tipo (quindi anche dispositivi come GPU programmabili oltre a CPU multicore) in un'implementazione parallela sviluppata a questo scopo per la soluzione del problema di simulazione noto come “equazioni delle acque basse” o, in inglese, “shallow water equations”. CPU parallele e GPU programmabili sono entrambe dotate di parallelismo a livello hardware, ed entrambe sono sempre più diffuse, e in questa tesi si mostra quindi come è stato utilizzato OpenCL per migliorare le prestazioni tramite la parallelizzazione del codice (e l'esecuzione su tale hardware) della soluzione dello schema di simulazione bidimensionale delle equazioni delle acque basse (denominate in inglese “2D shallow water equations”), questo in particolar modo su GPU effettuando quindi quello che viene definito GPU computing.

Dopo la spiegazione del modello utilizzato per la simulazione, dell'algoritmo per la sua risoluzione, dell'origine del calcolo parallelo, del significato di GPU computing, della sua storia, dello standard OpenCL e delle differenze tra CPU e GPU si presentano diverse versioni equivalenti utilizzando calcolo parallelo tramite OpenCL con la spiegazione delle peculiarità di ognuna. In seguito sono presentati i risultati sperimentali dell'esecuzione su diversi dispositivi (sia CPU sia GPU, sia su personal computer sia su server) e infine possibili sviluppi dell'utilizzo di OpenCL in questo problema di simulazione.

Negli ultimi anni la facilità di accedere (anche in ambito domestico) a macchine che permettano un reale calcolo parallelo è aumentata, in particolare ora è facile disporre di macchine con processori multicore e con schede grafiche programmabili, CPU multicore sono poi contenute anche in personal computer a basso costo e dispositivi mobili e le GPU non sono più da tempo relegate al solo calcolo dell'output grafico e sono anch'esse presenti su dispositivi mobili. L'uso delle GPU per effettuare calcoli paralleli viene chiamato GPGPU (General Purpose GPU) oppure GPU computing. La sigla GPU sta per Graphics Processing Units ed indica il componente centrale di una scheda video, è da notare che questi componenti eseguono normalmente operazioni in modo pesantemente parallelo, tuttavia limitate alla grafica, anche se esistevano comunque modi di utilizzare questa potenza di calcolo nascosta per altri scopi.

Entrambi i maggiori produttori di GPU (ATI ed NVIDIA) hanno sviluppato soluzioni per il GPU computing (rispettivamente con i linguaggi Brook e C for CUDA) che sono state poi dirette verso la definizione di OpenCL, uno standard (inizialmente proposto da Apple Inc.) per utilizzare dispositivi paralleli (in particolare le GPU ma anche le CPU) utilizzando principalmente il linguaggio di programmazione C in modo indipendente dal produttore. La soluzione precedente di ATI è considerata deprecata mentre NVIDIA supporta ancora C for CUDA.

Si fa notare che l'architettura delle schede video è molto adatta al calcolo parallelo, infatti ATI ed NVIDIA oltre a offrire schede video offrono anche schede con architetture simili alle controparti

video utilizzabili esclusivamente per il calcolo parallelo ad alte prestazioni (la serie Tesla di Nvidia e la serie Firestream di AMD).

Lo scopo dell'attività di tesi è stato quello di utilizzare OpenCL per lo sviluppo di un algoritmo parallelo per la soluzione del problema di simulazione noto come "equazioni delle acque basse" o, in inglese, "shallow water equations". In questa tesi si mostra come si può utilizzare OpenCL per realizzare tramite parallelizzazione un'implementazione più efficiente, in particolar modo su GPU, della soluzione dello schema di simulazione bidimensionale delle equazioni delle acque basse (denominate in inglese "2D shallow water equations").

Approcci simili erano già stati utilizzati tuttavia il GPU computing non era ancora supportato (e spinto dalle aziende produttrici di GPU) come oggi e i programmi erano basati sull'utilizzo di OpenGL. L'impulso a questa tesi è stato dato dalla volontà di migliorare, grazie alla parallelizzazione offerta da OpenCL, il codice Fortran completamente sequenziale realizzato dal prof. Edie Miglio che implementa uno schema di soluzione per le equazioni delle acque basse.

Questa tesi si apre nel Capitolo 1 "Modellistica" con una panoramica sui modelli per fluidi incomprimibili a superficie libera e presenta poi le equazioni di Navier-Stokes e come da esse è ricavato il modello bidimensionale delle acque basse. La tesi poi nel Capitolo 2 "Metodi di risoluzione" descrive prima la forma generale delle equazioni di conservazione e forme generali per la loro soluzione e poi il metodo Lax-Friedrichs e come è stato qui applicato. Nel Capitolo 3 "Calcolo parallelo, GPU computing e OpenCL" la tesi prosegue con la descrizione del calcolo parallelo, del GPU computing e dello standard OpenCL (con anche la loro storia e citando anche diverse piattaforme hardware utilizzate o utilizzabili per il GPU computing, con o senza OpenCL) ed esempi di applicazione; il capitolo termina spiegando i diversi modelli di calcolo parallelo utilizzabili in OpenCL e mostrando le differenze architetturali tra CPU e GPU (con anche la descrizione di due GPU specifiche). Nel Capitolo 4 "Implementazioni con OpenCL" si presentano diverse versioni equivalenti utilizzando calcolo parallelo tramite OpenCL con la spiegazione delle peculiarità di ognuna partendo da un'implementazione basilare e descrivendo poi i miglioramenti apportati, si accennano infine ad ulteriori miglioramenti apportabili alle prestazioni e all'output. Nel Capitolo 5 "Risultati dei test" sono presentati i risultati sperimentali dell'esecuzione su diversi dispositivi (sia CPU sia GPU, sia su personal computer sia su server). Infine nel Capitolo 6 "Conclusioni" si traggono le conclusioni valutando positivamente l'utilizzo di OpenCL per problemi di questo tipo facendo anche il confronto con la scarsa letteratura esistente e si descrivono inoltre possibili sviluppi effettuabili a partire dalle implementazioni descritte in questa tesi.