

Parallelizzazione di un decoder H.264 su piattaforma Cell, con paradigma a scambio di messaggi

Scenario

Le architetture MPSoC (Multi-Processor System-on-Chip) sono ormai ampiamente diffuse. Avendo a disposizione piattaforme molto potenti, la vera sfida è scrivere del software in grado di sfruttarle al meglio. Essendo disponibile in un prodotto accessibile a costi non elevati (PlayStation 3), Cell è il processore ideale per sperimentare nuovi approcci alla programmazione parallela su multiprocessore.

Un classico modo per parallelizzare l'applicazione è suddividere la mole di dati da elaborare in parti uguali, una per ogni processore a disposizione. Un approccio molto più flessibile è quello di suddividere l'algoritmo in tanti task, che comunicano tra loro tramite scambio di messaggi, per poi mappare in maniera ottima tali task sui processori disponibili.

Un decoder H.264 rappresenta una delle applicazioni che operano su stream di massima diffusione, basti pensare al suo utilizzo su lettori DVD, HDTV, cellulari.

Obiettivo del progetto

Scopo del progetto è parallelizzare un decoder H.264 open source. L'algoritmo dovrà essere suddiviso in task, i quali potranno essere mappati su diversi core. I task potranno comunicare tra loro utilizzando un paradigma a scambio di messaggi.

Passi Implementativi

La realizzazione del progetto attraversa diversi passi, che possono essere considerati punti di verifica dell'andamento del lavoro al termine dei quali (ma non solo) è consigliato un confronto con i responsabili del progetto.

- a) Familiarizzare con l'ambiente di sviluppo per Cell.
- b) Capire come utilizzare la libreria di Message-Passing esistente.
- c) Parallelizzazione del decoder H.264.
- d) Effettuare confronti di prestazioni con altre soluzioni.

Scheduling e Logistica

Partenza del progetto: ASAP

Il lavoro dovrà essere svolto in laboratorio almeno per la fase di esecuzione e debugging (è necessario utilizzare la PlayStation 3) e per un periodo iniziale che prevede l'apprendimento dell'ambiente di sviluppo software.

Lo studio e l'implementazione dell'algoritmo possono essere svolti a casa (è richiesto Linux, preferibilmente Fedora Core).

Aspetti Formativi

La scelta di questi progetti è consigliata a chi vuole acquisire esperienza nella progettazione di software parallelo per architetture multicore.

Sarà utilizzato il linguaggio C.

Referenti:

Ing. Marco Mantovani
Ing. Martino Ruggiero

marco.mantovani8@unibo.it
martino.ruggiero@unibo.it