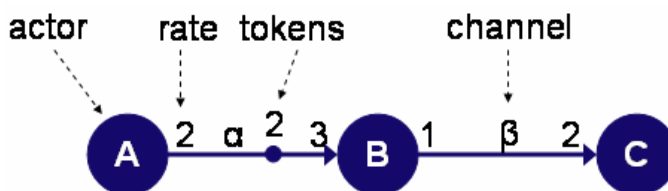


Modellazione e test in ambiente MPSoC di un'applicazione di streaming multimediale sviluppata mediante il modello computazionale Synchronous Dataflow

Scenario

Il modello computazionale *Synchronous Dataflow* (SDF) appare particolarmente adatto allo sviluppo di applicazioni multimediali che elaborano flussi di dati in input (*streams*) in ambiente multiprocessore su singolo chip (MPSoC).

Infatti, il paradigma SDF permette di esprimere esplicitamente il parallelismo e le dipendenze produttore consumatore. Inoltre, dopo aver profilato i tempi di esecuzione dei nodi computazionali, è possibile effettuare una serie di analisi statistiche per controllare il soddisfacimento di vincoli sullo throughput desiderato, al variare del mapping degli *actors* sulle differenti unità computazionali dell'architettura MPSoC.



Obiettivo del progetto

Scopo del progetto è modellare un'applicazione di video streaming H263 sotto forma di *dataflow application*. Bisognerà quindi trasformare il codice C/C++ (di cui comunque sono già state evidenziati le parti che presentano parallelismo sui dati e sui task) in un *Dataflow graph*. Il numero opportuno di nodi computazionali (actors) andrà individuato e a ciascuno di essi andrà associata un'opportuna sezione di codice C/C++ (che l'actor eseguirà ogni qualvolta riscontrerà soddisfatta la propria *firing condition*).

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.

- Analisi e sperimentazione del codice C/C++ su piattaforma virtuale MPSoC (tramite cross-compilazione e successiva simulazione). *Profiling* dei tempi di esecuzione dei task principali.
- Modellazione di un Dataflow graph che rispecchi il comportamento dell'applicazione, scegliendo opportunamente: numero di actors, dipendenze, dimensione dei token spediti, *rates* sugli archi, *delays* (numero di token già presenti sugli archi, che successivamente circoleranno sulla rete).
- Analisi statistiche sul grafo ottenuto e calcolo dello throughput al variare di *mapping* (raggruppamento degli actors in *cluster* da assegnare ai differenti core computazionali) e *scheduling* (sequenze di esecuzione all'interno di un cluster).
- Validazione complessiva di correttezza. valutazione dell'efficienza e del rispetto dei vincoli di throughput di tipo *soft real time* (vincoli temporali che è opportuno soddisfare, per esempio, per non avere ritardi nella decodifica e riproduzione di un flusso video compresso).

Scheduling e Logistica

Partenza del progetto: dalla seconda metà di marzo.

Il lavoro dovrà essere svolto in laboratorio almeno per un periodo iniziale che prevede l'apprendimento dei tool software utilizzati.

La successiva fase può essere svolta a casa utilizzando il software di progettazione o in laboratorio utilizzando postazioni condivise con altri studenti.

Aspetti Formativi

La scelta di questi progetti è consigliata a chi vuole acquisire esperienza nella progettazione di applicazioni di streaming multimediale per sistemi embedded e sia interessato allo studio di nuovi modelli computazionali.

Le competenze di base richieste/ da acquisire (anche in maniera assistita nella fase di training iniziale) riguardano:

- Conoscenza del linguaggio C/C++ per la comprensione di algoritmi codificati in tale linguaggio.
- Comprensione del concetto di cross-compilazione e simulazione su Instruction Set Simulator (ISS).
- Comprensione del MoC (Model of Computation) Synchronous Dataflow.
- Conoscenza di meta-linguaggi (in particolare sarà utilizzato l'XML).
- Familiarità con i principali codec audio/video.

Referenti:

Ing. Alessandro Dalla Torre, disponibilità dalla seconda metà di marzo

alessandro.dallatorre@unibo.it