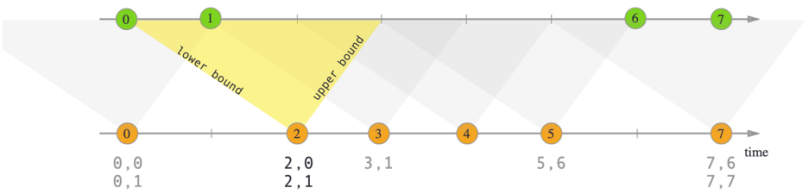
Titolo: Approcci ibridi di parallelismo per operatori di data stream processing

Proposta

Il paradigma del Data Stream Processing studia l'elaborazione efficiente di computazioni che operano su sequenze infinite di dati trasmessi ad alta velocità (es. dati finanziari, reti di sensori, social media). Dal punto di vista generale, le applicazioni di streaming vengono descritte come grafi dataflow, in cui i nodi rappresentano operatori eseguenti un calcolo stabilito dagli utenti sugli input ricevuti, e in grado di produrre output trasmessi ad altri operatori del grafo nella forma di risultati intermedi.

L'obiettivo di un motore di stream processing è di consentire l'esecuzione del grafo con la banda di elaborazione (*throughput*) più alta possibile e la *latenza* più bassa consentita. In questo contesto la tesi si concentra su operatori con stato tra cui in modo particolare l'operatore di *Online Interval Join* (OIJ).

L'OIJ è un operatore che processa due stream di dati chiamati rispettivamente stream *base* e stream *probe*. La computazione ha come obiettivo determinare tutte le coppie (x, y), dove *x* è un input ricevuto dallo stream *base* e *y* dallo stream *probe*, che rispettano la condizione di join. Questa è determinata dal fatto che *x* e *y* devono avere lo stesso attributo chiave, ed il timestamp *y.timestamp* di deve ricadere nell'intervallo temporale *[x.timestamp + lowerBound; x.timestamp + upperBound]*, dove *lowerBound* e *upperBound* sono dei limiti temporali dettati dall'utente durante la configurazione iniziale dell'operatore. La figura seguente mostra l'idea della computazione:



Come si può vedere, la coppia (2,0) rispetta la condizione di join mentre (3, 0) no (stream arancione *base*, verde di *probe*).

Strumenti

La tesi richiede di prendere dimestichezza con il linguaggio di programmazione C/C++ in riferimento alla libreria parallela per il data stream processing WindFlow, sviluppata dal Dipartimento di Informatica. Il candidato avrà a disposizione anche un pool di applicazioni di benchmark per i sopracitati strumenti, già disponibili in un repository pubblico GitHub.

Obiettivo

L'OIJ è disponibile in WindFlow con due approcci di parallelismo in cui in entrambi l'operatore è implementato da più thread concorrenti. Nel primo approccio cosìddetto *Key Parallelism* (KP), i dati dei due stream sono distribuiti sui thread in modo da assegnare dati della stessa chiave allo stesso thread, che eseguirà per intero il calcolo della OIJ con i dati finora ricevuti di quella chiave. Nell'approccio *Data Parallelism* (DP) invece, lo storico dei dati ricevuti di ciascuna chiave, utili per il calcolo dei risultati della OIJ, è diviso equamente tra i thread che lavorano su una partizione dei dati. Mentre KP fornisce in generale throughput maggiore, DP risulta migliore in termini di latenza.

Il lavoro ha come obiettivo aggiungere al sistema un meccanismo di monitoraggio in tempo reale delle chiavi e degli input finora ricevuti per chiave al fine di capire eventuali sblianciamento del carico e mitigarlo mediate un approccio ibrido KP+DP, atto a cercare di rendere il throughput più alto possibile con la latenza più bassa possibile (obiettivo non raggiunto con il semplice utilizzo di KP o DP in alternativa).

Prerequisiti

- Architetture degli Elaboratori e Sistemi Operativi
- Ingegneria del Software
- Una buona base di programmazione di sistema in C