Titolo: Integration of Data Compression Techniques in Stateful Stream Processing at the Edge

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). Le elaborazioni compiute possono andare dal calcolo di statistiche (streaming analytics) ma anche elaborazioni più complesse come il processo di inferenza su modelli precedentemente addestrati oppure tecniche di learning continuo. Dal punto di vista generale, le applicazioni di streaming vengono descritte come grafi dataflow, in cui i nodi rappresentano operatori che eseguono 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.

Le applicazioni di stream processing sono inerentemente parallele, essendo ogni nodo del grafo un thread indipendente. L'obiettivo di un buon motore di stream processing è quello di consentire l'esecuzione del grafo con la banda di elaborazione (throughput) più alta possibile e la latenza più bassa consentita. Tuttavia un aspetto critico, soprattutto nell'elaborazione su dispositivi con capacità limitate, è quello di gestire lo stato della computazione, il quale potrebbe diventare più grande della memoria a disposizione. In quel caso un approccio è quello di trasferire parte dello stato su memoria secondaria (direttamente o tramite strumenti come Key-Value Store). Un approccio alternativo, o complementare, è quello invece di ricorrere ad algoritmi di compressione da applicare sullo stato della computazione al fine di mantenerlo nella dimensione consentita. Tale approccio richiede la decompressione dei dati al fine di aggiornare lo stato e poterlo utilizzare, ed è auspicabile capire come organizzarlo tenendo conto delle specifiche tipologie di stato che sono spesso presenti nel paradigma dello stream processing (esempio uno stato rappresentato da una finestra temporale di dati ricevuti di recente mantenuti in ordine di arrivo).

Strumenti

La tesi è completamente incentrata sulla libreria di streaming open source WindFlow (https://github.com/ParaGroup/WindFlow), una libreria C++17 per lo streaming efficiente su multicore sviluppata dal Dipartimento di Informatica. L'obiettivo del lavoro è l'estensione del suo supporto a runtime per gli scopi definiti in questa proposta. Lo studente apprenderà quindi gli aspetti essenziali del linguaggio di programmazione C++.

Obiettivo

L'obiettivo della tesi è quello di inquadrare il problema con specifica applicazione al calcolo di aggregati su finestre temporali, che rappresenta un tipico operatore con stato. La libreria WindFlow è già dotata di un sofisticato supporto che consente di migrare parte dello stato su memoria secondaria. L'obiettivo sarebbe capire che trade-off si possono ottenere confrontando la soluzione che usa memoria secondaria con una soluzione, da prototipare, che utilizza compressione dei dati (evitando o rendendo meno frequente l'interazione con la memoria secondaria).

Prerequisiti

- Architetture degli Elaboratori e Sistemi Operativi
- Ingegneria del Software
- Una buona base di programmazione (con linguaggi C/C++, Java)