

Titolo: Progettazione ed Implementazione di una API ad Alto Livello Tabellare per WindFlow

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 eseguanti 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.

Lo sviluppo di applicazioni di streaming può essere compiuto usando API con diversi livelli di astrazione. Alcune risultano più *user-friendly*, nascondendo però molti aspetti di configurazione a basso livello che possono risultare importanti al fine delle prestazioni. Altre sono di più a basso livello, consentono un alto livello di configurabilità e ottimizzazione, ma risultano più ostiche per un programmatore meno esperto. Per esempio Apache [Flink](#), un motore di stream processing molto popolare, fornisce diverse astrazioni: una API chiamata *DataStream* più a basso livello, una API di complessità intermedia e molto dichiarativa chiamata *TableAPI*, e una API ad altissimo livello di fatto basata su dialetti dell'SQL.

Strumenti

La tesi è completamente incentrata sulla libreria open-source di streaming [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 prevede l'estensione del suo supporto a runtime e della API esistente per gli scopi definiti in questa proposta. Lo studente apprenderà quindi gli aspetti essenziali del linguaggio di programmazione C++ e dei suoi più recenti standard (C++17).

Obiettivo

L'obiettivo è dotare WindFlow di una API molto dichiarativa e user-friendly come la [TableAPI](#) di Flink. Essa consente di definire applicazioni di stream processing che eseguono calcoli relazionali (esempio filtering, projection, join, aggregati) in modo semplice utilizzando l'API di livello sottostante chiamata *DataStream*. Fare una cosa simile, seppur limitata per la tipologia di operatori offerti da WindFlow (che sono un sottoinsieme di quelli disponibili in Flink) richiede di imparare a padroneggiare il linguaggio di programmazione C++ nei suoi ultimi standard e lavorare a livello di programmazione generica (template) e meta-programmazione.

Prerequisiti

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