

**Titolo:** Analisi e Studio di Meccanismi di Microbatching Adattivo nel Contesto del 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). 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 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.

Lo sviluppo di sistemi di streaming per architetture multicore è un argomento di ricerca particolarmente "caldo", dove lo studio di supporti a runtime efficienti dal punto di vista della banda di elaborazione e della latenza rappresenta un obiettivo importante e non di semplice ottenimento. Una ottimizzazione per migliorare la banda è quella consentire lo scambio di dati tra operatori in forma di piccoli batch, in modo da ammortizzare gli overhead del runtime. Tipicamente batch di decine/centinaia di dati sono sufficienti per ottimizzare la banda su sistemi multicore. Tuttavia, l'uso del batching si scontra con la latenza di elaborazione, più alta in caso di batch inutilmente grandi. Il lavoro verte sullo studio di strategie ed euristiche per determinare il livello di batching migliore per le applicazioni di streaming.

### **Strumenti**

La tesi è completamente incentrata sulla libreria di streaming open source [WindFlow](https://github.com/ParaGroup/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**

Determinare la dimensione ottimale dei batch è un problema complesso. Batch inutilmente grandi aumentano la latenza senza un apprezzabile incremento della banda. Per contro, batch piccoli consentono di minimizzare la latenza a fronte di una banda di elaborazione in generale peggiore a causa degli overhead (costi fissi) pagati per singolo dato scambiato. Lo scopo della tesi è quello di studiare euristiche e strategie per controllare dinamicamente ed in modo autonomo la dimensione dei batch, in modo da stabilizzarsi sui livelli ottimali per ciascuna applicazione. Tale aspetto richiede un approccio decentralizzato per ciascuna coppia di operatori interconnessi, che decideranno in autonomia il grado di batching migliore. Tale idea consente di avere livelli di batching diversi nell'intero grafo dataflow, in cui percorsi non-critici (con traffico ridotto) avranno livelli di batching inferiori rispetto a percorsi critici, che attraversano cioè operatori che più spesso comunicano nell'applicazione. La base di partenza rappresenta il supporto di batching già presente in WindFlow, che tuttavia richiede di specificare la dimensione dei batch in modo statico e per coppia di operatori comunicanti.

### **Prerequisiti**

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