

**Titolo:** Realizzazione di un Sistema di Eventi Temporizzati in una Libreria di Data Stream Processing in C++ (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. L'elaborazione di computazioni di streaming avviene mediante sistemi in cui le applicazioni sono sviluppate come grafi dataflow di operatori, ciascuno rappresentante uno stadio intermedio di trasformazione dei dati.

Il lavoro di tesi verte nella realizzazione di un sistema di eventi temporizzati nel framework di ricerca per il Data Stream Processing WindFlow (<https://github.com/ParaGroup/WindFlow>), una libreria C++17 per lo streaming efficiente su multicore sviluppata dal Dipartimento di Informatica.

### **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**

Gli operatori di un grafo applicativo WindFlow ricevono un flusso di dati ed eseguono il codice fornito dall'utente per calcolare output indirizzati verso gli operatori successivi del grafo. Solitamente il codice dell'operatore può operare non solo sugli input, ma anche su strutture di stato interno che sono usate e aggiornate per la computazione. Nell'approccio standard, la logica dell'operatore è innescata esclusivamente alla ricezione di un nuovo input (*data-driven*), e richiede l'esecuzione della logica utente e la produzione dei corrispondenti risultati. Può capitare in molte applicazioni reali la necessità di eseguire, da parte degli operatori, callback innescate ad intervalli regolari di tempo (*timeout*), in modo da operare sullo stato interno dell'operatore ed eventualmente produrre output. In sostanza, tali callback verranno eseguite allo scadere di un timer e non alla ricezione di un nuovo dato. Tale approccio può essere integrato con meccanismi per eseguire callback non solo ad intervalli regolari, ma anche al verificarsi di determinati eventi (*event-driven*).

L'obiettivo della tesi è progettare una nuova versione di WindFlow che consideri la possibilità di avere eventi temporizzati in almeno un sottoinsieme degli operatori forniti dalla libreria, e valutare il progetto proposto con un prototipo applicato ad alcuni casi d'uso applicativi disponibili o da progettare.

### **Prerequisiti**

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