# Paper Review - FineStream: Fine-Grained Window-Based Stream Processing on CPU-GPU Integrated Architectures

Sai Venkat Malreddy

15 oktober 2024

## 1 Idea and Contribution

FineStream addresses the performance limitations of stream processing on traditional discrete CPU-GPU architectures, where data transmission via PCI-e causes high latency and reduces throughput. This issue arises because separate CPU and GPU memory requires frequent data transfers, bottlenecking performance. FineStream leverages CPU-GPU integrated architectures, which use shared memory between CPUs and GPUs on the same chip, eliminating PCI-e overhead. The proposed methodology includes fine-grained workload scheduling between CPU and GPU, a performance model, and dynamic query adjustment, allowing optimized execution of SQL-based stream queries. This approach significantly enhances throughput and reduces latency compared to existing engines designed for discrete architectures. Main contributions of the paper is leveraging shared memory to optimize query execution and features a performance model and lightweight scheduler for dynamic workload adjustments, achieving significant improvements in throughput, latency, and energy efficiency over existing methods

## 2 Positive Comments

- **Application to many use cases**
  The paper demonstrates FineStream across diverse datasets and queries, including anomaly detection in smart grids, vehicle tracking, and cluster monitoring, showcasing its adaptability to various applications.
- **Online Profiling**
  The profiling component helps adjust resource allocation and optimize task scheduling dynamically, ensuring that the system can adapt to changes in workload or data patterns. Authors mentioned, it enables FineStream to make real-time decisions about how to distribute workloads between the CPU and GPU, helping to maintain optimal performance throughout changing conditions.
- **Role of Shared Memory in Performance**
  The theoretical exploration of shared memory's role in performance optimization contributes to a better understanding of memory architectures in modern computing systems

## 3 Negative Comments

- **utilization**
  The framework assumes a relatively balanced workload between CPU and GPU, which may not hold true in all real-world scenarios, potentially leading to under utilization of one component.

- **Fault tolerance**
  The paper does not adequately address fault tolerance and recovery strategies, which are essential for maintaining reliability in production systems, particularly in streaming applications.

- **Reactive vs Proactive Profiling**
  The current approach relies on reacting to data patterns rather than anticipating them. Non-deterministic behavior often means that changes can be abrupt and unpredictable. The dynamic resource allocation may not have enough foresight to allocate resources effectively before a problem arises.

## 4 Questions unanswered about the paper

- **What are the key parameters considered in the performance model, and how do they impact scheduling decisions?**