Improving Financial Systems and Software Performance by Utilizing the GPU

Jack Imburgia  
Department of Computer Science

Syracuse University  
Syracuse, NY USA  
joimburg@syr.edu

***Abstract* – Alternative computational methodologies are often researched when looking for potential improvements in performance and efficiency. Each subsequent year sees the demand for computational power increase, the amount of data to be analyzed increase, and the problems to be solved more mathematically and computationally complex. This is compounded by efforts to better utilize power consumption. It is critical that organizations with great computational needs have the most efficient and fastest processing available.**

**Scheduling methodologies in operating systems are often researched when looking for potential improvements in performance and efficiency. With the proliferation of embedded and internet of things (IOT) hardware and systems, the increased complexity of these systems, and the resulting heighted demands for resources, it is critical to maximize efficiencies in processing time, energy usage, and the capacity of throughput.**

**Financial services have traditionally relied on CPU based systems to perform computations and data processing.**

**This paper proposes that financial services companies adopt a GPU focused architecture and software designs. This can lead to decreased processing time presently and open doors as the products and services in finance and the capital markets become increasingly complex.**

**Operating systems have traditionally relied on Fixed Priority (FP) scheduling systems. The reasons for are many including ease of implementation and predictability of output. However, a reasonable alternative to this approach is the scheduling method Earliest Deadline First (EDF). In contrast to FP, EDF offers dynamic priority scheduling where tasks are continuously reevaluated and scheduled next according to their next deadline. This can lead to greater CPU utilization which will result increased efficiency.**

Keywords—gpu, finance, software

# Introduction and Background

An operating system manages hardware, network activity, system I/O, persisting files to disk storage and, in the interest of this paper, scheduling processes. Real Time Operating Systems (RTOS) must behave in a way that seems immediate and responsive to end users. Operating systems used in embedded system devices such as smart phones are an example. Real time operating systems accomplish this through scheduling methodologies that allocate resources so that multiple processes can run concurrently which results in responsive experiences for users. Data is processed as it arrives and ideally there are no buffer delays.

## GPU

There are two components to determine the performance of a task in a real time operating system. First, the output of task must compute the desired result correctly. Second, the task must be completed in an expected time frame.

But the accuracy of a task cannot be assessed solely on the output. There are also two additional components to determining the correctness of a real time operating system. First, it must be determined if a process was completed in a time frame that meets its requirements. Second, the difference between the expected and actual timing must be calculated to determine the variability. An important job of a real time operating system is to have a mechanism in place to deal with the worst case amounts of time each task takes to complete

## Financial services

Tasks is real time operating systems can generally be generally be put into two categories.

* **Banks** -
* **Hedge funds** -
* **Asset managers** -

# Analysis: Challenges faced by financial Services softwares and systems

An alternative to traditional fixed priority scheduling systems used in most real time operating systems is Earliest Deadline First (EDF). This introduces the ability of tasks to have periodic deadlines which are not seen in fixed priority.

## Massive amounts of data processing

The concept of priority remains in earliest deadline first. However, is not fixed. Instead, it is dynamically determined based on the closest

.

## Complex financial calculations

Tasks in earliest deadline first are periodic tasks. Each task will repeatedly perform a process over a span of time. The periods are also referred to cycles and the tasks as cyclic tasks.

## Increasing reliance on data science

The tasks that are eligible to be run are maintained in the ready queue. Tasks are with the highest priority tasks at the beginning of the queue. These are the tasks have the closest deadline. The lowest priority tasks are at the end of the queue and have

# Research

In researching the application of GPUs in financial services, three papers are cited for their relevance:

1. **Dimitri Dojchinovski; Marjan Gusev; Vladimir Zdraveski, "Efficiently Running SQL Queries on GPU"**

An early attempt to demonstrate that earliest deadline first can be implemented in a Unix environment.

## Background

There are two components to determine the performance of a task in a real

## Implementation

There are two components to determine the performance of a task in a real

## Results

There are two components to determine the performance of a task in a real

1. **Steven Solomon; Ruppa K. Thulasiram; Parimala Thulasiraman, "Option Pricing on the GPU"**

A proposal for an alternative implementation of the earliest deadline scheduler ready queue with the hope of making it more popular in industry.

## Options

There are two components to determine the performance of a task in a real

## Implementation

There are two components to determine the performance of a task in a real

## Results

There are two components to determine the performance of a task in a real

Fig xxxx. Option pricing performance for CPU vs GPU

A picture containing text, line, screenshot, plot

Description automatically generated

*https://ieeexplore.ieee.org/document/5581462/footnotes#footnotes-id-fn1*

1. **Gow-Hsing King; Zong-You Cai; Yan-Ying Lu; Jan-Jan Wu; Hung-Pin Shih; Chao-Rui Chang, "A High-Performance Multi-user Service System for Financial Analytics Based on Web Service and GPU Computation"**

Suggested improvements for standard earliest deadline first algorithms with the goal of reducing complexity.

## Overview

There are two components to determine the performance of a task in a real

## Proposal

There are two components to determine the performance of a task in a real

1. *Service Oriented Architecture (SOA):* Put something here…

* **Interoperability** -
* **Loose coupling** -
* **Abstraction** -
* **Granularity** -

Fig xxxx. SOA architecture with GPU

A picture containing text, screenshot, diagram, design

Description automatically generated

1. *High performance financial analytics with GPU*: Put something here..

1. *Web services*: Put something here..

# Proposed GPU Solutions to Fiancial Services Challenges

Table 1 shows three tasks with the relevant attributes needed to perform earliest deadline first scheduling.

## CUDA centric software

Put something here…

Fig xxxx. CUDA architecture

A picture containing text, screenshot, diagram, rectangle

Description automatically generated

*https://en.wikipedia.org/wiki/CUDA*

1. *Numba CUDA:* Put something here…

Benchmark

1. *BlazingSQL*: Put something here.. explain rdbms

Benchmark:

1. *cuDF*: Put something here… dataframes

Benchmark:

Fig xxxx. cuDF vs Pandas – mean calculation

A screenshot of a computer program

Description automatically generated with medium confidence

Fig xxxx. cuDF vs Pandas – merge operation

A screenshot of a computer code

Description automatically generated with medium confidence

## GPU enhanced software architecture

Put something here…

# Proof of Concept implementation

Earliest deadline first can be implemented relatively simply using C#. Fig. 2 displays a basic data structure to store the task related attributes. The State attribute stores the count of times that task received the CPU during this scheduling pattern.

The source code discussed in the proof of concept is located at https://github.com/jackimburgia/earliest-deadline-first .

# Conclusion

Earliest Deadline First offers many benefits that make it a viable and reasonable approach to scheduling in real time operating systems. The increased CPU utilization will lead to a better experience for end users, less power consumption, and free up resources or other activities. While not presently popular in commercial operating systems mostly due to the complexities of the its implementation, its benefits are worth the effort of the difficult implementation.

# References

1. Dimitri Dojchinovski; Marjan Gusev; Vladimir Zdraveski, "Efficiently Running SQL Queries on GPU", 2018 26th Telecommunications Forum (TELFOR)
2. Steven Solomon; Ruppa K. Thulasiram; Parimala Thulasiraman, "Option Pricing on the GPU", 2010 IEEE 12th International Conference on High Performance Computing and Communications (HPCC)
3. Gow-Hsing King; Zong-You Cai; Yan-Ying Lu; Jan-Jan Wu; Hung-Pin Shih; Chao-Rui Chang, "A High-Performance Multi-user Service System for Financial Analytics Based on Web Service and GPU Computation", International Symposium on Parallel and Distributed Processing with Applications
4. Nolan Foster, "How GPU Computing is Improving Financial Analytics", https://www.acecloudhosting.com/blog/how-gpu-computing-improve-financial-analytics/
5. "CUDA", https://en.wikipedia.org/wiki/CUDA
6. Mantas Levinas, "Everything You Need to Know About GPU Architecture and How It Has Evolved", https://www.cherryservers.com/blog/everything-you-need-to-know-about-gpu-architecture#:~:text=While%20the%20specific%20components%20will,data%20(SIMD)%20stream%20architecture.
7. TechTarget Contributor, "GPGPU (general purpose graphics processing unit)", https://www.techtarget.com/whatis/definition/GPGPU-general-purpose-graphics-processing-unit
8. Tristan Yates, "Quants: The Rocket Scientists of Wall Street", https://www.investopedia.com/articles/financialcareers/08/quants-quantitative-analyst.asp
9. Jacob Lee, "6 Ways GPU Acceleration is Disrupting Financial Analytics", https://www.kinetica.com/blog/6-ways-gpu-acceleration-disrupting-financial-analytics/
10. Spin, "Black-Scholes Model for Dummies", https://spin-fi.medium.com/black-scholes-model-for-dummies-372e08d7398f
11. Pawel Lachowicz, "GPU-Accelerated Finance in Python with NumbaPro Library. Really?", https://quantatrisk.com/2014/12/06/gpu-accelerated-finance-in-python-with-numbapro/
12. Martin Heller, "BlazingSQL review: Fast ETL for GPU-based data science", https://www.infoworld.com/article/3610188/blazingsql-review-fast-etl-for-gpu-based-data-science.html
13. Mike Beaumont, "Unlock the potential of Google Colab with RAPIDS AI + BlazingSQL", https://medium.com/rapids-ai/experience-blazingsql-running-100x-faster-than-apache-spark-on-google-colab-4ad579dd98b4
14. Scott Grauer-Gray, William Killian, Robert Searles, John Cavazos, “Accelerating Financial Applications on the GPU”, https://cavazos-lab.github.io/FinanceBench/resources/AcceleratingFinancialApplicationsOnTheGPU-paper.pdf
15. George Seif, "Here’s how you can speedup Pandas with cuDF and GPUs", https://towardsdatascience.com/heres-how-you-can-speedup-pandas-with-cudf-and-gpus-9ddc1716d5f2