

Nome do Aluno: Eiki Luis Yamashiro Batista dos Santos

Nome do Orientador: Rafael Corsi Ferrão

Título: Aceleração de Algoritmos de *High-Frequency Trading* com FPGA

Palavras Chave: FPGA, *High-Frequency Trading*, Aceleração de Algoritmos, Bolsa de Valores, Operações em Alta Frequência.

1. Descrição do Problema e Revisão de Literatura

Desde seu surgimento, a bolsa de valores oferece diversos benefícios, para investidores, com a renda proveniente de um certo nível de risco e a abertura para realização de investimentos, e para empresas, possibilitando a arrecadação de fundos, além de bancos ou financeiras. Dessa forma, a bolsa de valores promove o aquecimento de toda cadeia produtiva e econômica dos países.

Segundo Peter Gomber [1], a dinâmica competitiva do mercado transformou o panorama de processos de compra e venda de ativos, o que resultou em uma nova maneira de execução de conceitos e de infraestrutura acerca da segurança das operações. O processo inteiramente manual, exposto a erros humanos, foi se automatizando e a utilização de ferramentas da computação tornou-se cada vez mais recorrente. Dessa forma, o mercado iniciou sua modernização a partir do uso de *Electronic communication networks (ECNs)*.

Atualmente, o mercado financeiro movimenta bilhões de dólares diariamente, e cada vez mais, se torna acessível para novos investidores. Assim, a importância e quantidade de análises demandadas pelo mercado aumentou. O *High-Frequency Trading* ou operações em alta frequência permite a análise de uma grande quantidade de informação, de forma a otimizar o desempenho dos modelos, priorizando as tomadas de decisões eficientes, que são essenciais para a obtenção do lucro.

A quantidade de dados que um algoritmo é capaz de processar pode impactar na velocidade, portanto, é necessário balancear e identificar a prioridade, visto que a quantidade pode aumentar a eficiência, enquanto a velocidade garante menos variação entre o valor real e o esperado. Segundo Sabedra [4], o tempo de espera na tomada de ação do algoritmo de alta frequência permite o classificar em 4 tipos diferentes de estratégias de *HFT*. A Tabela 1 apresenta a descrição de cada tipo e seu tempo de espera.

	Automated Liquidity Provision	Market Microstructure Trading	Event Trading	Deviations Arbitrage
Descrição	Algoritmos quantitativos, priorizam a velocidade e são otimizados para executar da maneira mais rápida possível	Algoritmos de engenharia reversa que analisa dados de um curto período de tempo do passado para realizar uma ação	Algoritmos específicos para eventos sazonais do mercado	Utiliza estatística e modelos matemáticos
Tempo de Espera	1 minuto	10 minutos	1 hora	1 dia

Tabela 1 - Diferentes tipos de estratégias de *HFT* e seus respectivos tempos de espera (tempo de atuação das ordens)

As ordens enviadas para a bolsa de valores são computadas em ordem de chegada. Portanto, para todas as estratégias apresentadas na Tabela 1, além da eficiência, o tempo também é considerado para mensurar o desempenho, cada milissegundo é importante e impacta diretamente no lucro sobre os outros investidores do mercado. Assim, os feeds de dados de mercado destinados às operações de baixa latência são disponíveis via UDP, permitindo que uma única fonte seja compartilhada para vários usuários [2].

Entre o tempo da requisição até a recepção da ordem, o preço pode variar, ou seja, o valor esperado, pode ser diferente do valor real, visto que cada operação possui o potencial de alterar o preço de compra ou venda. Mediante a essa situação, um método para acelerar os algoritmos através de *FPGAs* é proposto por Ricardo Sabedra [4] para aumentar a velocidade que a ordem é feita, pois quanto menor o tempo entre a requisição e a recepção, menor é a chance de ocorrer uma grande variação entre o valor real e o esperado.

Segundo Leber [3] uma decodificação eficiente do fluxo de dados via UDP é essencial, assim a decisão de emitir o pedido deve ser realizada com a menor latência possível. Para atingir esses objetivos, Leber apresentou um mecanismo implementado em *FPGA*, para acelerar as negociações. Assim, as tarefas de decodificação são descarregadas para blocos de um hardware otimizado. Esse sistema proposto implementa uma pilha de processamentos completa.

2. Objetivo

O atual projeto tem como objetivo acelerar um algoritmo de *High-Frequency Trading* através de uma *FPGA*, visando diminuir a variação entre o valor esperado e o valor real. Portanto, a proposta integra a plataforma *Metatrader 5* com o algoritmo implementado e a *FPGA*.

3. Metodologia

Primeiro será feito um estudo da literatura a fim de entender o algoritmo, uma implementação em python será realizada a fim de entendimento e comparação futura dos resultados, devemos também entender com realizar a comunicação com o sistema de compra e venda de ações, para então atacar o problema e acelerar o algoritmo em hardware. Uma pesquisa será realizada a fim de decidir a melhor maneira de realizar a implementação e quais impactos ela terá no sistema final.

3.1 Algoritmo

Aldrige [6] apresenta um algoritmo clássico para o *High-Frequency Trading*, conhecido como *Bandas de Bollinger*, esse modelo indica valores máximos e mínimos dos preços de uma posição específica, para qualquer ponto no tempo. Assim, o cálculo fornece duas bandas, a superior e a inferior, que determinam uma área onde, com base no histórico do ativo, é possível identificar se o preço sofrerá uma valorização ou uma queda. Além das bandas é possível determinar a média móvel dos valores de uma ação.

A Figura 2 ilustra o funcionamento do algoritmo. Se a variação do preço corrente (linha vermelha) cruzar a borda superior, de baixo para cima, é enviada uma ordem de venda, caso contrário (variação do preço cruza a banda inferior) é enviada uma ordem de compra.

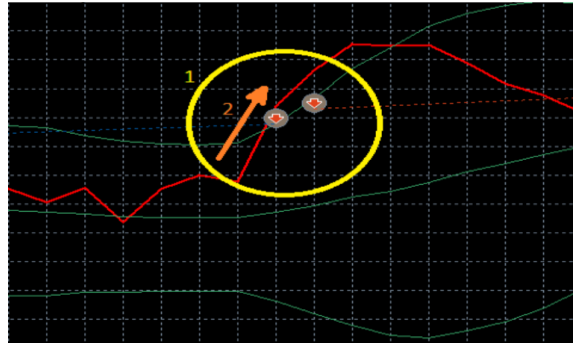


Figura 1 - [3] Borda de subida - Bandas de Bollinger

As Equações da média móvel (\bar{X}) e do desvio padrão (σ) a seguir, são utilizadas para o cálculo das Bandas de Bollinger:

$$\bar{X} = \frac{\sum_{j=1}^N \bar{X}}{N}$$

Equação 1 – Cálculo da média móvel (\bar{X})

$$\sigma = \sqrt{\frac{\sum_{j=1}^N (X_j - \bar{X})^2}{N}}$$

Equação 2 – Cálculo do desvio padrão (σ)

A partir do cálculo do desvio padrão e da média móvel, pode-se determinar as bandas superior, central e inferior:

$$\text{Banda superior} = \bar{X} + 2\sigma$$

Equação 3 – Cálculo da Banda Superior

$$\text{Banda central} = \bar{X}$$

Equação 4 – Cálculo da Banda Central

$$\text{Banda inferior} = \bar{X} - 2\sigma$$

Equação 5 – Cálculo da Banda Inferior

Para permitir a comparação da variação do preço corrente, o armazenamento desses valores é feito em filas do tipo *FIFO* (*First in, First out*).

Segundo Babbar [8], o algoritmo das Bandas de Bollinger apresenta uma alta precisão na tomada de decisões de comprar ou vender ações. Assim, esse será o algoritmo escolhido para ser implementado na *FPGA*.

3.2 Implementação na *FPGA*

Existem diversas formas de implementar o algoritmo de Bandas de Bollinger na *FPGA*. É possível utilizar descrição de hardware (VHDL), ferramentas como *High Level Synthesis (HLS)* ou *OpenCL*, faz parte do estudo definir qual ferramenta será utilizada, ponderando a eficiência da implementação.

O *HLS* ajuda a solucionar os problemas no desenvolvimento de programas em *FPGA*, ele permite que cientistas que não estão familiarizados com hardware, executem seus aplicativos na *FPGA* (Fujita [7]). Nesse método, a interface é conectada diretamente com a *FPGA*, o que permite a comunicação com baixa latência.

Bonato [9] apresenta em seu artigo, uma proposta para realizar a integração do *Metatrader 5* e *FPGA*, com auxílio do *OpenCL*. Essa ferramenta é utilizada por desenvolvedores de hardwares heterogêneos, como unidades de processamento gráfico (*GPUs*) ou *FPGAs*. Para Fujita [7], o desempenho absoluto de *floating-points* em *FPGAs* não é o suficiente para superar *GPUs*, especialmente em casos de cálculos de ponto de precisão dupla, utilizados em clusters *HPC (High Performance Computing)* como aceleradores. Logo, é importante que as aplicações da *FPGA* sejam aquelas em que a *GPU* não pode exercer uma alta eficiência. A Figura 2 ilustra a visão geral da proposta de Bonato. As setas em rosa representam a comunicação entre o Algoritmo e o *Metatrader 5* e as setas em verde representam a comunicação entre a *FPGA* e o algoritmo em C++. Note que é utilizado um protocolo de 32 bits.

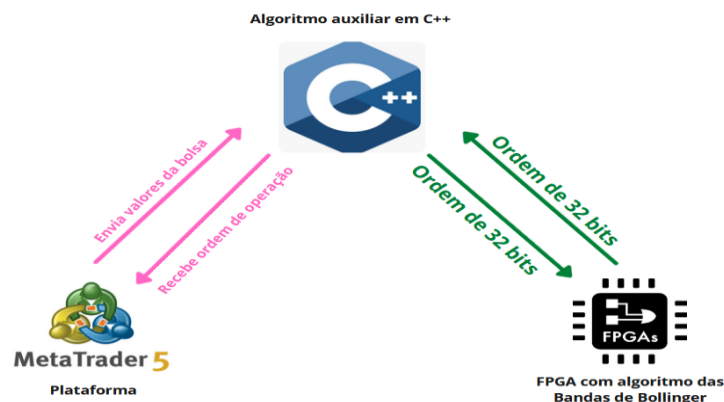


Figura 2 – Visão geral da integração *Metatrader 5* e *OpenCL*

A *FPGA* pode ser conectada ao computador via *PCI Express (PCIe)* para a transmissão de dados, isso permite uma menor latência de comunicação entre o computador e a placa *FPGA* que realiza o processamento.

3.3 Materiais

Para o desenvolvimento do projeto será utilizado a *FPGA DE5 a-NET-DDR4* disponível no laboratório de arquitetura de computadores do Insper, que possui uma *FPGA* de alta densidade (*Arria 10 GX FPGA*) comunicação *PCIe (x8)*. A *FPGA* será conectada na workstation do laboratório¹.

Pensando no trabalho durante a pandemia, um acesso remoto via *VPN* será configurado possibilitando o trabalho remoto.

3.4 Cronograma

Atividades	2021					2022						
	ago	set	out	nov	dez	jan	fev	mar	abr	mai	jun	jul
Estudar Metatrader 5	X	X										
Configurar <i>FPGA</i> + Workstation		X	X									
Interpretar pacote <i>UDP</i> e implementação em <i>C++</i>				X	X	X	X					
Estudar o Algoritmo						X	X	X				
Implementação do Algoritmo na <i>FPGA</i>								X	X	X		
Integração e Validação									X	X	X	
Artigo									X	X	X	X

¹ Devido a pandemia iremos configurar acesso remoto ao equipamento, possibilitando o desenvolvimento remoto.

4. Resultados Esperados

Com o intuito de permitir a análise do desempenho do método proposto, é necessário compreender os conceitos de *throughput* e latência. Medido em bits por segundo, o *throughput*, em termos gerais, pode ser entendido como a taxa de transferência em redes de comunicação (como a *Ethernet*), sendo a razão entre a quantidade de dados transferido pelo tempo em que essa transferência levou para ser feita. Já a latência pode ser definida como a quantidade de tempo necessária para um pacote de dados ir de um ponto designado para outro.

É esperado obter os *throughput* e a latência de uma implementação do algoritmo acelerado com *FPGA* e comparar os resultados com uma implementação em python que funciona integralmente na CPU, além de dados como o tempo de espera do hardware em função do tamanho da média móvel. Dessa forma, será possível avaliar o desempenho da performance da implementação em hardware.

5. Referências Bibliográficas

- [1] P. Gomber, B. Arndt, M. Lutat, T. Uhle, *High Frequency Trading. E-Finance Lab*. Abril, 2011.
- [2] G.W Morris, D.B. Thomas, and W. Luk, *FPGA Accelerated Low-Latency Market Data Feed Processing*, 2009 17th IEEE Symposium on High Performance Interconnects. 2009.
- [3] C. Leber, B. Geib, H. Litz, *High Frequency Trading Acceleration using FPGAs*, *University of Heidelberg*. 2011.
- [4] R. S. Sabedra, Implementação em FPGA de algoritmo para análise de ativos financeiros na bolsa de valores. Universidade Federal do Rio Grande do Sul, Instituto de Informática, 2020.
- [5] J. A. Brogaard, *High Frequency Trading and Its Impact on Market Quality*. Northwestern University School of Law. Julho, 2010.
- [6] Aldrige, I. *High-Frequency Trading – A Practical Guide to Algorithmic Strategies and Trading Systems*. [S.l.]: John Wiley Sons, Inc., 2010.

[7] N. Fujita, *Parallel Processing on FPGA Combining Computation and Communication in OpenCL Programming*. University of Tsukuba, Graduate School of Systems and Information Engineering, Tsukuba, Japan. 2019.

[8] BABBAR, R. *A study of performance of moving average, Bollinger bands and relative strength index in selected stocks and stock indices*. 2011.

[9] COSTA, C. R.; ROSA, L. d. S.; BONATO, V. Integrando o metatrader5 com aceleradores fpga via opencl named pipes. In: Workshop em Computação Heterogênea – WCH.[S.I.]: SBC, 2018

[10] A. Wilen, J. Schade, R. Thornburg. *Introduction to PCI Express, A Hardware and Software Developer's Guide*. Intel Press. 2003.