



PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO

**Estudo sobre a Utilização de *Clusters*
de GPU na Resolução de Problemas
de Computação Financeira**

Luiz Marcio Faria de Aquino Viana

TRABALHO DE CONCLUSÃO DE CURSO

CENTRO DE CIÊNCIAS SOCIAIS - CCS

DEPARTAMENTO DE ADMINISTRAÇÃO

Pós-graduação em Administração de Empresas

Rio de Janeiro, Novembro de 2011.



Luiz Marcio Faria de Aquino Viana

**Estudo sobre a Utilização de *Clusters* de GPU
na Resolução de Problemas de Computação Financeira**

Trabalho de Conclusão de Curso

Trabalho de Conclusão de Curso, apresentada ao programa de pós-graduação IAG Master em Desenvolvimento Gerencial – ONS da PUC-Rio como requisito parcial para a obtenção do título de especialista em Administração.

Orientador: Luiz E. Brandão

Rio de Janeiro
Novembro de 2011.

Resumo

Viana, Luiz Marcio F. A. Brandão, Luiz E. Estudo sobre a Utilização de *Clustes* de GPU na Resolução de Problemas de Computação Financeira. Rio de Janeiro, 2011. 36 p. Trabalho de Conclusão de Curso - IAG Master em Desenvolvimento Gerencial – ONS – Departamento de Administração. Pontifícia Universidade Católica do Rio de Janeiro.

Existe no mercado mundial uma crescente demanda por soluções de computação de alto desempenho que são voltadas para a área financeira. Esta necessidade levou alguns fabricantes de processadores de placas de gráficas a desenvolver soluções que usam a capacidade de processamento matemático existente nestas placas para oferecer uma plataforma de computação baseada em clusters de processadores gráficos a um custo acessível. O objetivo do trabalho é realizar um estudo inicial sobre os problemas de computação financeira que estão sendo resolvidos com o uso desta tecnologia e realizar uma avaliação do mercado para este tipo de solução.

Palavras- chave

Cluster; GPU; Árvore de Decisão; Opções Reais; NVIDIA

Abstract

Viana, Luiz Marcio F. A. Brandão, Luiz E. Study about the Use of GPU Clusters to Solve Financial Problems. Rio de Janeiro, 2011. 36 p. Trabalho de Conclusão de Curso - IAG Master em Desenvolvimento Gerencial – ONS – Departamento de Administração. Pontifícia Universidade Católica do Rio de Janeiro.

There is a growing world market demand for solutions for high performance computing geared towards the financial market. This necessity has led some manufacturers of graphics processors to develop solutions that leverage the power of existing mathematical processing these boards to provide a computing platform based on clusters of graphics processors at an affordable cost. The objective of this study is to conduct an initial study on the financial problems of computing that are being resolved with the use of this technology and make an assessment of the market for this type of solution.

Key-words

Cluster; GPU; Decision Tree; Real Options; NVIDIA

Sumário

1 O problema	1
1.1. Introdução	1
1.2. Objetivo Final	2
2 Conceitos sobre <i>Stream Multiprocessors</i>	4
2.1. A Especificação OpenCL	4
2.2. A Arquitetura Fermi	5
2.3. Recursos de Desenvolvimento	8
3. Problemas de Computação Financeira	12
3.1. Avaliação de Projetos Usando Opções Reais	12
3.2. Conceitos sobre Opções Reais	13
3.3. Método de Avaliação com Opções Reais	17
3.4. Exemplo de Avaliação de Opções Reais	18
4 Metodos de Avaliação de Risco	21
4.1. Árvores de Decisão	21
4.1.1. Cálculo da Árvore de Decisão	22
4.2. Simulação de Monte Carlo	23
4.2.1. Gerando Valores Aleatórios	24
5. Principais Fabricantes	26
5.1. NVIDIA	26
5.1.1. Estratégia de Mercado da NVIDIA	28
5.2. AMD	28
5.2.1. Estratégia de Mercado da AMD	30
5.3. Mercado de Computação Baseado em GPU	30
6 Conclusão	33
6.1. Sugestões e recomendações para novas pesquisas	34
7 Bibliografia	35

Lista de figuras

Figura 1: Arquitetura Composta por 16 <i>Streaming Multiprocessors</i>	7
Figura 2: A Arquitetura Fermi - <i>Streaming Multiprocessors</i>	7
Figura 3: Modelos de Computação Paralela que Suportam CUDA.....	8
Figura 4: Programa dividido em blocos de tarefas que são processados mais rapidamente em GPUs com múltiplos processadores	9
Figura 5: Modelo de processamento SIMT(<i>Single-Instruction, Multiple-Thread</i>)	10
Figura 6: Exemplo de multiplicação de matriz utilizando GPU	11
Figura 7: Exemplo de variação no preço de um ativo modelado por árvore	14
Figura 8: Exemplo de modelagem de opções usando uma árvore em grade	15
Figura 9: Exemplo de modelagem de opções usando uma árvore binária.....	15
Figura 10: Exemplo de árvore de decisão de um modelo de opção real.....	21
Figura 11: Distribuição de probabilidade acumulada de uma normal.....	25
Figura 12: Resultado das vendas de placas Tesla	31
Figura 13: Crescimento no uso de supercomputadores por indústria	32

Lista de Tabelas

Tabela 1: Melhorias introduzidas na arquitetura Fermi.	6
Tabela 2: Os passos para avaliação de projetos usando opções reais.....	18
Tabela 3: Sequencia de fases do projeto da planta de geração eólica.	19
Tabela 4: Estimativa do valor do projeto pelo fluxo de caixa.	19
Tabela 5: Estimativa de variação de custo do projeto.....	19
Tabela 6: Árvore de decisão com o cálculo da variação de custo do projeto.	20
Tabela 7: Passos necessários para a simulação de Monte Carlo.....	23
Tabela 8: Probabilidade de variação do início de um projeto.....	24
Tabela 9: Expectativa de receitas com vendas de placas Tesla em 2011.	32

1 O problema

Diversos problemas de computação financeira são resolvidos usando simulações e análises baseadas em árvores de decisão. Estes são métodos computacionais que trabalham com uma quantidade grande de dados e consomem muito processamento e a evolução na arquitetura dos processadores começa a disponibilizar alto poder de processamento por um custo baixo, criando novas possibilidades nesta área.

Atualmente, muito investimento tem sido feito na utilização da tecnologia de processadores gráficos na resolução de problemas de computação financeira, e isto vem sendo explorado por grandes empresas deste setor, como a NVIDIA e a AMD, que desenvolveram uma linha de produtos específica para este segmento e este trabalho realiza um estudo sobre um dos problemas de computação financeira que esta sendo resolvido com o uso desta tecnologia.

1.1.Introdução

Os computadores da atualidade possuem múltiplos núcleos de processamento, e são construídos para atender à todos os tipos de aplicações, sendo classificados como processadores de propósito geral. Estes processadores atendem a um conjunto grande de aplicações que vão desde aplicações de processadores de texto e planilhas eletrônicas a simulações e análises complexas.

Embora os processadores de propósito gerais sejam largamente utilizados, existe um conjunto de processadores muito utilizados, mas pouco conhecidos, que são desenvolvidos especificamente para atender a determinadas necessidades, sendo dedicados ao processamento de sinais, criptografia de dados, processamento gráfico e etc.

Os processadores gráficos denominados GPU (*Graphical Processing Unit*), são processadores específicos para processamento de imagens e são constituídos por um número grande de unidades de processamento matemático.

Estes processadores possuem facilidade para processar uma grande quantidade de dados, e estão sendo aplicados em outras áreas de pesquisa.

A combinação de alto desempenho e baixo custo destes processadores fornece a oportunidade ideal para o surgimento de computadores de alto desempenho com custos realmente acessíveis. Estes processadores podem ser combinados em uma arquitetura onde diversas unidades são agrupadas, aumentando a capacidade de processamento.

Atualmente estes sistemas de computação estão sendo explorados em diversos segmentos tais como no estudo climático, óleo e gás, financeiro, biologia, e outros. Com diversos fabricantes, tais como IBM, HP e Cray, oferecendo soluções de computação utilizando a arquitetura dos processadores gráficos (GPU).

Este trabalho está dividido em seis seções. Na **Seção 2**, apresentamos os conceitos sobre *streaming multiprocessors* que são os processadores gráficos que estão sendo usados para resolver problemas de computação financeira. Na seção seguinte, **Seção 3**, nós discutimos um dos problemas de computação financeira que demandam grande quantidade de processamento que é a avaliação de projetos através da análise de opções reais. Na **Seção 4**, apresentamos um estudo sobre árvores de decisão e simulação de Monte Carlo que são dois métodos usados em análise financeira. Na **Seção 5** analisamos o mercado que vem sendo explorada pela NVIDIA, que é um dos principais fabricantes de processadores gráficos, e na última seção, **Seção 6**, apresentamos as conclusões e sugestões de trabalhos futuros.

1.2.Objetivo Final

Este trabalho tem como objetivo identificar os problemas de computação financeira que precisam de grande poder de processamento, e quais são as expectativas deste mercado através da análise da estratégia da NVIDIA, que é uma das principais empresas de desenvolvimento nesta área.

Além de identificar as necessidades de computação de alto desempenho na modelagem de problemas de computação financeira, analisando como a

tecnologia de *stream multiprocessors* pode ser utilizada nesta área para fornecer soluções de baixo custo com grande capacidade de processamento e que permitem a exploração de modelos cada vez mais complexos e precisos.

Este trabalho também procura identificar o potencial do mercado que está sendo explorado pela NVIDIA, além de identificar soluções que possam explorar a capacidade de processamento dos *stream multiprocessors*.

2 Conceitos sobre *Stream Multiprocessors*

Com o objetivo de criar imagens computacionais cada vez mais reais, os fabricantes de *hardware* desenvolveram processadores gráficos com grande capacidade de processamento. Estes processadores possuem uma quantidade grande de unidades de processamento aritmético, o que possibilita a exploração destes processadores também em aplicações não gráficas e que requerem um grande volume de processamento matemático.

2.1. A Especificação OpenCL

A utilização de processadores gráficos para resolver problemas que demandam grande capacidade de processamento computacional, teve início em 2003 com estudos acadêmicos sobre este assunto. Neste período, empresas como a NVIDIA e a AMD disponibilizaram interfaces de programação para explorar a capacidade dos seus processadores gráficos, e em seguida, outras empresas iniciaram o desenvolvimento de interfaces de programação também proprietárias, o que gerou um grande problema de padronização na indústria e que tornava necessário o desenvolvimento de aplicações específicas para cada modelo de placa de vídeo o que aumentava a complexidade no desenvolvimento das aplicações.

Este cenário motivou os engenheiros da Apple a criar em 2008 a especificação OpenCL que definiu um padrão de linguagem dedicada à computação paralela baseada em GPU. O objetivo da especificação OpenCL é criar um padrão de interface de programação com suporte a múltiplas plataformas (Mac OS, Linux e Windows), e que tire vantagem de ambientes computacionais compostos por diferentes *hardwares*.

Os primeiros processadores gráficos suportados eram da NVIDIA que já possuía uma interface de programação bastante desenvolvida e que serviu de base para o desenvolvimento inicial da especificação OpenCL.

A primeira implementação da especificação OpenCL foi disponibilizada para o Mac OS em Agosto de 2009, e a Apple considerou esta tecnologia como essencial para os seus produtos e passou a incluí-la em todos os novos lançamentos, o que criou um monopólio de fornecimento de placas da NVIDIA para os computadores da Apple.

Atualmente, outros fabricantes de processadores gráficos como a AMD desenvolveram implementações da especificação OpenCL para seus produtos, mas aderiram mais tarde a tecnologia o que permitiu uma penetração maior dos produtos da NVIDIA.

2.2. A Arquitetura Fermi

Com o objetivo de facilitar o desenvolvimento de aplicações que utilizam o poder de processamento das GPUs, a NVIDIA desenvolveu uma plataforma composta por uma interface de programação de alto nível baseada na linguagem C/C++ e em uma arquitetura inovadora, denominada Fermi, que introduziu inúmeros recursos voltados para aplicações que necessitam de grande poder de processamento.

A arquitetura Fermi foi desenhada para trabalhar de forma extremamente eficiente com matemática de precisão dupla, que representam valores com precisão de até 16 casas decimais, sendo até 4x mais rápida nestas operações que a arquitetura anterior de placas gráficas.

Esta arquitetura também inclui melhorias em diversas áreas tais como no suporte à memória com detecção e correção de erros, na hierarquia de cachê, na maior capacidade de memória compartilhada, e na troca rápida de contexto entre tarefas. Estas melhorias permitiram a NVIDIA produzir um processador com excelente desempenho e com uma interface de programação simples. A **Tabela 1** apresenta um resumo sobre cada uma das melhorias introduzidas na arquitetura Fermi.

<p>Memória com detecção e correção de erros. As memórias com detecção e correção de erros são extremamente importantes quando pensamos em computação paralela realizada em <i>clusters</i>, pela combinação de múltiplas placas de vídeo, onde a probabilidade de erro de memória e falha do sistema aumenta significativamente.</p>
<p>Memória compartilhada. O conceito de memória compartilhada no próprio <i>hardware</i> permite que as tarefas que estão em execução no processador compartilhem uma mesma área de memória, reduzindo significativamente a comunicação entre a memória e o processador.</p>
<p>Hierarquia de cachê. O uso de uma hierarquia de cachê de memória facilita o desenvolvimento de aplicações que resolvem problemas computacionais que não são resolvidos facilmente com o uso de memória compartilhada. A hierarquia de cachê possui dois níveis. Um primeiro nível, denominado cachê L1, que fica dentro de um <i>streaming processor</i> e que aproveita a localidade dos dados endereçados no mesmo processador. E um segundo nível de cachê, denominado cachê L2, que aproveita a localidade dos dados acessados por todos os <i>streamings processors</i> existentes no processador gráfico.</p>
<p>Troca rápida de contexto entre tarefas. Assim como as CPUs que suportam multitarefa, os processadores gráficos atuais também são desenvolvidos para executar em um ambiente multitarefa, onde diversas aplicações compartilham um tempo do processador. Neste ambiente, o processador perde muito tempo sempre que ocorre uma troca de contexto entre aplicações e a arquitetura Fermi é 10 vezes mais rápida que as arquiteturas anteriores.</p>

Tabela 1: Melhorias introduzidas na arquitetura Fermi.

O primeiro processador gráfico baseado na arquitetura Fermi possuía aproximadamente 3 bilhões de transistores e 512 núcleos de processamento CUDA (CUDA é o nome dado pela NVIDIA para a sua arquitetura de computação paralela baseada em GPU), e um núcleo CUDA é uma unidade de processamento capaz de executar uma instrução de precisão dupla ou com número inteiro por ciclo de processamento. Isto é, em um processador que trabalha com uma velocidade de 1 GHz o ciclo de processamento é da ordem de 10^{-9} segundos.

Os 512 núcleos são agrupados em 16 *Streaming Processors*, cada um com 32 núcleos CUDA. O processador possui uma interface de 384 bit (ou 6 valores de precisão dupla) para se comunicar com uma memória de até 6 GB através da interface PCI Express comum aos computadores atuais. A **Figura 1** abaixo apresenta a arquitetura Fermi.

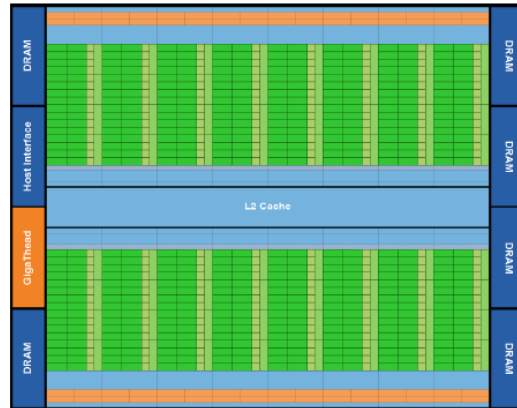


Figura 1: Arquitetura composta por 16 *Streaming Multiprocessors*.

Fonte: [NVIDIA's Next Generation CUDA Computer Architecture: Fermi 2010]

Cada *Streaming Processor* possui até 32 núcleos CUDA, **Figura 2**, que são compostos por uma unidade de precisão dupla e de números inteiros, além de 16 unidades de leitura e escrita em memória que permitem que o acesso a memória seja realizado por até 16 tarefas em um mesmo ciclo de processamento. Além disso, existem 4 unidades para processamento de funções especiais para cálculos de seno, cosseno e raiz quadrada. Cada unidade especial possui um fluxo de processamento separado do fluxo normal e executa as instruções em um ciclo.

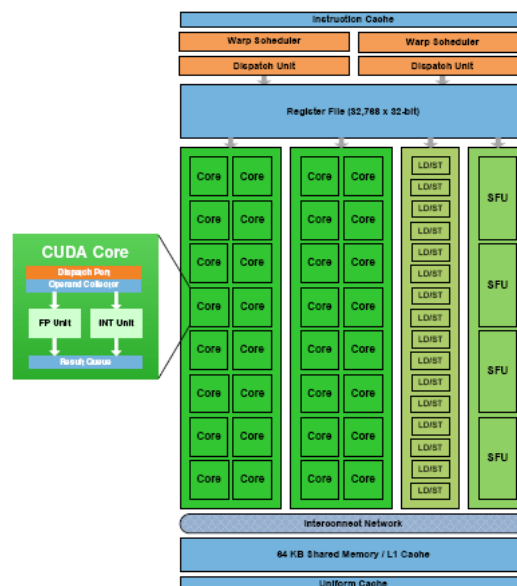


Figura 2: A arquitetura Fermi - *Streaming Multiprocessor*.

Fonte: [NVIDIA's Next Generation CUDA Computer Architecture: Fermi 2010]

2.3. Recursos de Desenvolvimento

A arquitetura Fermi introduzida pela NVIDIA apresentou um novo conceito de computação paralela de propósito geral denominado CUDA, que permitiu a utilização da arquitetura Fermi na resolução de inúmeros problemas de computação paralela. Servindo de base para o surgimento de outros padrões da indústria como o OpenCL que é uma especificação de modelo de programação paralela que procura abstrair a complexidade da arquitetura através de uma camada de desenvolvimento. A **Figura 3** apresenta alguns dos modelos de computação paralela que utilizam CUDA.

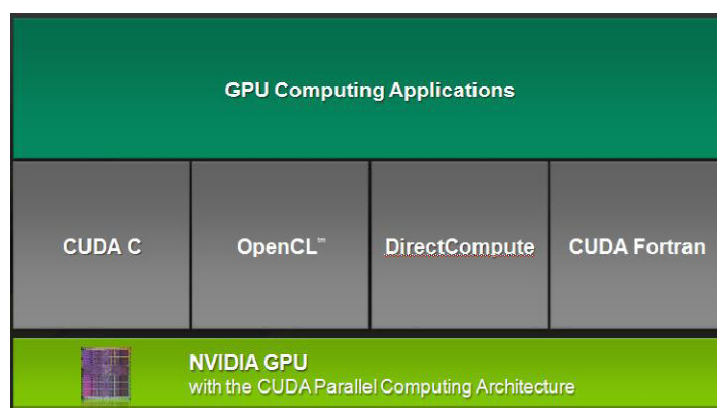


Figura 3: Modelos de computação paralela que suportam CUDA.

Fonte: [OpenCL Programming Guide for CUDA Architecture 2010]

O modelo de computação paralela definido pelo conceito CUDA foi criado para permitir que as aplicações consigam obter o melhor rendimento das arquiteturas paralelas independentemente das características da placa gráfica.

Esta abstração utiliza um paralelismo no nível de tarefas que possibilita que o programa seja subdividido em blocos de tarefas que podem ser executados em paralelo (**Figura 4**). O processamento de cada bloco de tarefas é independente e executado sobre uma massa grande de dados que é dividida para ser processada pelas diversas unidades de processamento existentes.

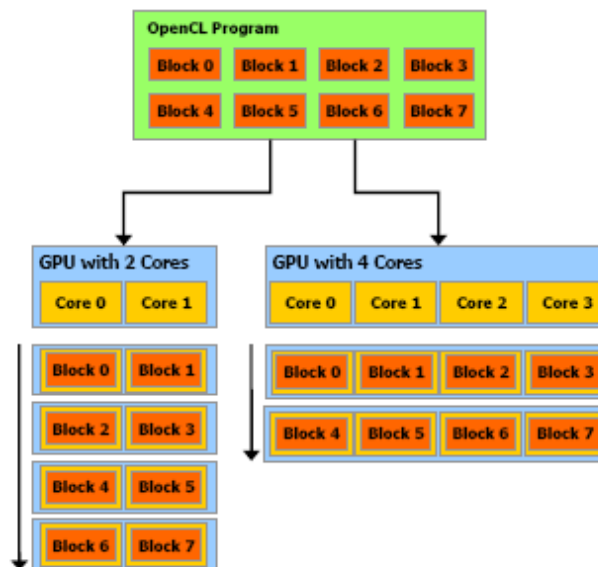


Figura 4: Programa dividido em blocos de tarefas que são processados mais rapidamente em GPUs com múltiplos processadores.

Fonte: [OpenCL Programming Guide for CUDA Architecture 2010]

Quando um programa OpenCL dispara um processo os blocos são numerados e distribuídos como blocos de tarefas para os diversos núcleos de processamento que estão disponíveis. As tarefas destes blocos são executadas concorrentemente no multiprocessador e quando um bloco termina outro entra imediatamente em processamento.

Um multiprocessador é projetado para executar centenas de tarefas simultaneamente e para gerenciar uma grande quantidade de tarefas ele utiliza o conceito de arquitetura SIMT (*Single-Instruction, Multiple-Thread*) que maximiza a utilização das unidades matemáticas da GPU (**Figura 5**).

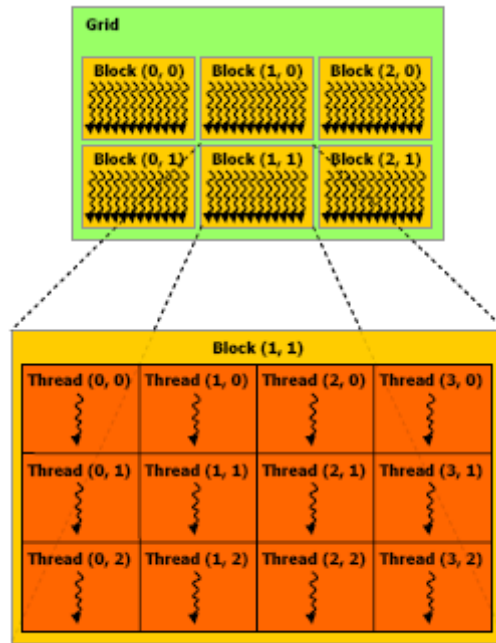


Figura 5: Modelo de processamento SIMT (*Single-Instruction, Multiple-Thread*).

Fonte: [OpenCL Programming Guide for CUDA Architecture 2010]

A arquitetura SIMT possui muitas similaridades com a arquitetura SIMD (*Single-Instruction, Multiple-Data*), pois trabalham com o processamento de grandes volumes de dados em paralelo. Porém estas arquiteturas diferem de forma significativa uma das outras. A principal diferença é que na arquitetura SIMD o processamento paralelo ocorre no nível de instruções que operam com múltiplos dados de entrada, enquanto em uma arquitetura SIMT múltiplas operações são executadas em cada tarefa e o paralelismo ocorre no nível de tarefas simplificando muito o desenvolvimento de aplicações que exploram esta arquitetura. Além disso, nas arquiteturas SIMD não é possível abstrair as características do *hardware* e o desenvolvimento de algoritmos que exploram de forma eficiente esta arquitetura se torna muito complexo e com pouca capacidade para escalar com o aumento no número de unidades de processamento.

A **Figura 6** apresenta um exemplo de multiplicação de matriz utilizando o processamento realizado por uma arquitetura SIMT. Neste exemplo cada bloco de tarefa é responsável por processar uma matriz quadrada contida nas matrizes A e B, e o resultado da computação de cada matriz quadrada produz uma sub-matriz da matriz de resultado C.

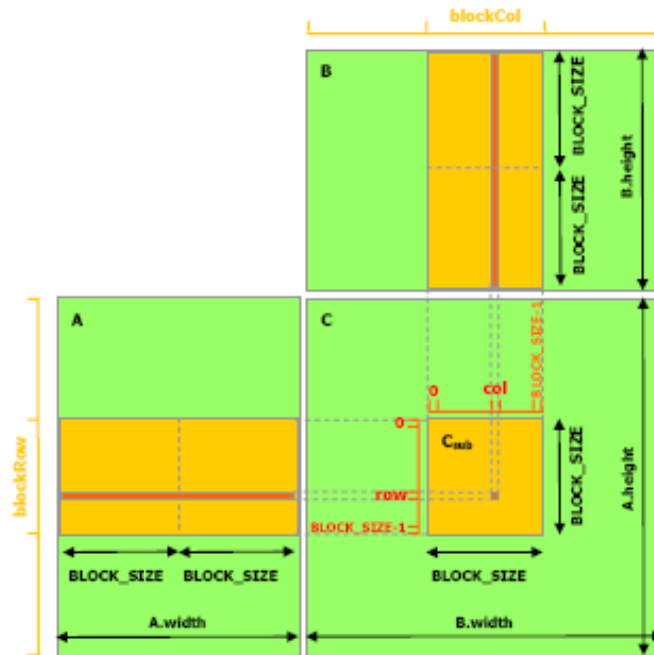


Figura 6: Exemplo de multiplicação de matriz utilizando GPU.

Fonte: [OpenCL Programming Guide for CUDA Architecture 2010]

Para o problema caber nos recursos do *hardware*, cada matriz retangular, A e B, é dividida em várias matrizes quadradas de dimensão igual ao tamanho do bloco de processamento, e cada sub-matriz de C é computada pela soma dos produtos destas matrizes quadradas, onde cada um destes produtos é calculado carregando as sub-matrizes quadradas de A e B e processando cada produto utilizando uma tarefa do bloco, e o resultado do processamento é acumulado e utilizado para produzir as sub-matrizes de C.

Nesta seção, apresentamos os conceitos básicos sobre as arquiteturas das GPUs e como estes processadores exploram o processamento paralelo. Na próxima seção examinaremos alguns problemas de computação financeira que podem obter vantagens com o uso desta tecnologia.

3 Problemas de Computação Financeira

Um dos problemas de análise financeira que podem ser resolvidos com o uso da tecnologia de multiprocessadores baseados em GPUs envolve a avaliação de projetos usando análise de opções reais.

Os problemas de análise de opções reais são típicos problemas que demandam grande capacidade de processamento, pois a complexidade da solução aumenta significativamente com o aumento no número de decisões que serão tomadas durante o projeto, e a avaliação é baseada em árvores de decisões que exigem o processamento de uma quantidade grande de dados.

3.1.Avaliação de Projetos Usando Opções Reais

O processo usualmente adotado para avaliar investimentos em projetos é o método de análise do fluxo de caixa descontado. O problema com esta metodologia é que ela considera que todas as fases do projeto serão executadas e que não serão tomadas decisões que possam alterar o custo do projeto.

Porém, para projetos grandes que envolvem decisões de investimento que podem afetar significativamente o custo do projeto, é necessário avaliar o investimento através de uma perspectiva que leve em consideração as diferentes opções que podem ser tomadas, e nesta situação uma avaliação utilizando uma metodologia baseada em opções reais se torna mais adequada.

O conceito de opções reais é aplicado na avaliação de projetos que envolvem a tomada de decisão em diferentes fases e para estes casos o uso de uma análise de opção real produz uma estimativa melhor para avaliação do projeto do que o processo de avaliação utilizando fluxo de caixa descontado do projeto.

Como exemplo, suponha que uma empresa esteja estudando a construção de uma nova planta industrial e que a decisão de construção da planta envolve um gasto inicial de \$60 milhões para a elaboração do projeto e

estudos iniciais de viabilidade, e que após este período se a viabilidade do projeto for aceita, o projeto passa para uma segunda fase onde a empresa pode optar por construir a nova planta industrial ao custo de \$400 milhões, e após um período de 2 anos a empresa irá reavaliar a demanda do mercado e poderá optar por expandir a fábrica ao custo de mais \$800 milhões.

Este é um típico cenário para aplicação de uma análise utilizando opções reais na avaliação de projetos. Nesta análise o custo final do projeto depende das decisões tomadas pela empresa nas diversas etapas, e envolve tomada de decisões apoiadas em uma análise muito mais elaborada dos custos do projeto.

3.2. Conceitos sobre Opções Reais

Devido à necessidade de melhorar a estimativa dos custos dos projetos, muitos estudos foram conduzidos com o objetivo de construir uma metodologia de avaliação de projetos usando opções reais com formulações matemática seguindo modelos semelhantes aos adotados em opções financeiras, porém devido à grande quantidade de incertezas e de decisões que podem ser tomadas em um projeto este problema se torna muito complexo do ponto de vista matemático e as soluções adotadas são freqüentemente obtidas por simulações.

Por exemplo, quando avaliamos opções financeiras onde a opção pode ser exercida apenas na data de maturidade é possível estimar o valor de uma opção com base em uma formulação matemática levando em consideração o valor atual da ação, o tempo até a sua maturidade e algumas constantes determinadas, como foi apresentado em [Black, Scholes 1973], sendo que isto é possível quando são assumidas algumas condições ideais.

- i) A taxa de juros tem que ser constante ao longo do tempo;
- ii) O preço da ação segue um processo aleatório contínuo do tipo *Random Walk* com variância que cresce linearmente no tempo;
- iii) A ação não paga dividendos ou outras distribuições;
- iv) A opção é do tipo Européia e só pode ser exercida na data de maturidade;
- v) Não existe custo de transação na compra ou venda da ação ou opção;

Entretanto estas formulações matemáticas são possíveis apenas em modelos onde a opção pode ser exercida na maturidade como é o caso de opções européias. Quando analisamos modelos mais complexos como no caso das opções americanas, onde a opção pode ser exercida a qualquer momento até a data de maturidade, ainda não existe uma formulação matemática capaz de estimar o valor de uma opção e o processo de estimativa é baseado em simulações.

No caso de opções reais existem muitas incertezas e uma grande quantidade de opções que podem aumentar significativamente a complexidade do modelo, e uma forma de simplificar matematicamente o processo de avaliação é usando uma aproximação binomial [COX, ROSS, RUBINSTEIN 1979] e neste caso existem duas considerações. A primeira utiliza uma aproximação através de uma árvore em grade e a outra através de uma árvore binária.

Em ambos os casos, as árvores são construídas de modo a representar a probabilidade de variação para cima ou para baixo no preço de um ativo. Assim, considerando S como o preço de um ativo em um tempo t_0 , então em um tempo t posterior este ativo tem probabilidade q de assumir um valor igual à Su e probabilidade $1 - q$ de assumir um valor igual à Sd , sendo que u e d representam a variação proporcional do preço do ativo no tempo t , conforme mostra a **Figura 7** abaixo.

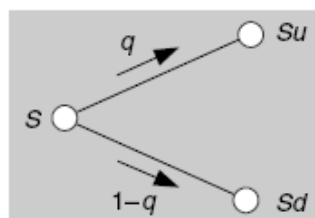


Figura 7: Exemplo de variação no preço de um ativo modelado por árvore.

Fonte: [Brandão, Dyer, Hahn 2005]

Em uma árvore em grade o resultado de um movimento para cima e posteriormente para baixo no preço de um ativo é igual ao resultado de um movimento para baixo e em seguida para cima, conforme apresenta a **Figura 8**. Deste modo, o uso de uma árvore em grade é um processo pouco intuitivo principalmente para problemas de modelagem mais complexos porque se

assume que uma árvore em grade é uma árvore completa o que não é verdadeiro para a maioria dos modelos.

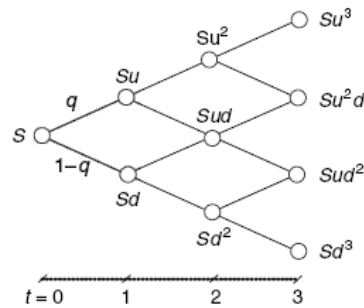


Figura 8: Exemplo de modelagem de opções usando árvore em grade.

Fonte: [Brandão, Dyer, Hahn 2005]

Desta forma, outra simplificação matemática pode ser obtida utilizando uma árvore binária onde cada nó produz um par de novos nós. A **Figura 9** a seguir, apresenta um exemplo de uso de uma árvore binária para resolver problemas de avaliação de projetos usando opções reais.

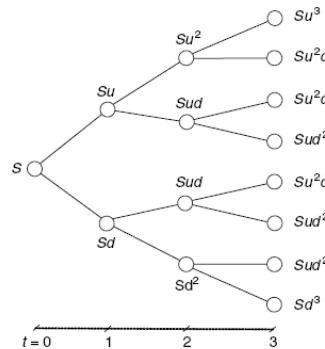


Figura 9: Exemplo de modelagem de opções usando uma árvore binária.

Fonte: [Brandão, Dyer, Hahn 2005]

O uso de uma árvore binária para modelar opções simplifica o processo de modelagem já que é possível incluir os nós de decisão mais facilmente utilizando esta forma de representação. Porém, quando avaliamos o problema do ponto de vista computacional a solução de um problema utilizando uma árvore em grade oferece uma complexidade menor para a solução do problema, $O(n^2)$, enquanto que em uma árvore binária a quantidade de nós aumenta em complexidade na proporção de $O(2^n)$, onde n é o número de níveis da árvore.

Para calcularmos os valores dos parâmetros **u** e **d** que representam as taxas de variação que o preço do ativo pode receber no intervalo de tempo **dt**, nós podemos usar um valor de portfólio que replique o valor do ativo como foi apresentado por [BRANDÃO, DYER, HAHN 2005]. Isto é obtido considerando que o valor do ativo pode ser representado por uma quantidade **A** de ações de valor **S** e um valor **B** investido a uma taxa livre de risco **r** como apresenta a **Equação (i)** abaixo.

$$(i) V = AS + B$$

Como em um período de tempo **dt** o valor do portfólio tem probabilidade **q** de variar o valor das ações para **Su** e probabilidade **1 - q** de variar o valor para **Sd**, onde **u > 1** reflete uma taxa de crescimento no valor da ação e **d = 1 / u < 1** reflete uma redução do valor da ação. Deste modo, após um tempo **dt** o valor do ativo será dado pelas **Equações (ii) e (iii)** apresentadas a seguir.

$$(ii) V_u = AS_u + B(1 + r)$$

$$(iii) V_d = AS_d + B(1 + r)$$

A partir destas equações podemos calcular a quantidade de ações **A** e o valor **B** que compõem o valor do portfólio que representa o valor do ativo no instante inicial, conforme é apresentado nas **Equações (iv) e (v)** abaixo.

$$(iv) A = (V_u - V_d) / ((u - d) S)$$

$$(v) B = (uV_d - dV_u) / ((u - d) (1 + r))$$

Para calcular o valor de **p** substituímos os valores de **A** e **B** definidos nas **Equações (iv) e (v)** na **Equação (i)** e obtemos às **Equações (vi) e (vii)** apresentadas a seguir.

$$(vi) V = [pV_u + (1 - p) V_d] / (1 + r)$$

$$(vii) p = (1 + r - d) / (u - d)$$

A probabilidade **p** é denominada de probabilidade livre de risco porque representa a probabilidade de crescimento no valor do ativo em função da taxa livre de risco **r**. Este valor é usado para construir a árvore em grade ou a árvore

binária, que são usadas para calcular o valor esperado do projeto em função dos pagamentos futuros e de uma taxa de desconto livre de risco.

3.3. Método de Avaliação com Opções Reais

Na avaliação de um projeto usando opções reais, os gestores precisam identificar inicialmente as decisões que deverão ser tomadas ao longo do desenvolvimento do projeto. Esta seqüência de decisões será usada na avaliação do projeto considerando as opções e constitui o primeiro passo do processo de avaliação usando esta metodologia.

O passo seguinte requer o calculo do valor presente do projeto desconsiderando as opções para cada fase da análise e esta avaliação pode ser obtida usando o método de avaliação pelo fluxo de caixa descontado do projeto, que representa a melhor estimativa para o valor do projeto desconsiderando as opções.

No passo seguinte, são calculados os parâmetros da aproximação binomial, isto é, os valores de **u**, **d**, e **p**, usando como estimativa a volatilidade no preço do ativo e uma taxa livre de risco de interesse.

Com base nestes parâmetros construímos uma árvore binária que possui como características uma probabilidade constante **p** e **1 - p** de variação para cima e para baixo em todas as fases, e que representa um processo de difusão do tipo *Geometric Browning Motion* (GBM), que é um processo aleatório do tipo *Random Walk* usado para descrever preços de ativos onde o próximo valor é determinado pelo valor atual somado a uma variável aleatória que segue uma distribuição fixa para o processo.

Após a modelagem do projeto utilizamos uma árvore em grade e atribuímos as opções para construir a árvore de decisão do projeto [COPELAND, TUFANO 2004]. A **Tabela 2** apresenta os passos necessários para avaliar um projeto utilizando análise de opções reais.

1) Identificar as flexibilidades gerenciais e as seqüência de decisões.
2) Estimar o valor do projeto desconsiderando as opções e usando o método do fluxo de caixa descontado.
3) Calcular os parâmetros de uma aproximação binomial para o processo.
4) Construir uma árvore em grade.
5) Adicionar as opções e resolver a árvore de decisão usando uma taxa livre de risco.

Tabela 2: Os passos para avaliação de projeto usando opções reais.

Neste processo são realizadas diversas iterações para obter estimativas de valores e com o aumento no número de decisões que precisam ser tomadas ao longo do projeto a complexidade da árvore se torna muito grande exigindo um maior esforço computacional para a sua resolução.

Deste modo, estes modelos exigem cada vez mais recursos de processamento para efetuar as análises em um tempo satisfatório, e para este tipo de aplicação o uso de processadores com capacidade de processamento paralelo para operações matemáticas pode ser bastante útil.

3.4.Exemplo de Avaliação de Opções Reais

Neste exemplo, considere que uma empresa deseja avaliar o investimento necessário para construir uma planta de geração eólica de 600 MW que pode ser ampliada para 1000 MW após entrar em operação.

No primeiro ano, serão gastos \$40 milhões para preparar um estudo de viabilidade, desenvolver os projetos e obter as licenças necessárias à obra. Após esta fase, a empresa tem a opção de gastar mais \$200 milhões com obras de infra-estrutura necessárias para a construção da usina. Em seguida, a empresa tem a opção de gastar mais \$400 milhões para construir a planta de geração eólica, e após a planta entrar em operação, a empresa poderá optar por ampliar a capacidade de produção para 1000 MW pelo custo de mais \$200 milhões. A **Tabela 3** apresenta cada fase da obra com os respectivos custos e prazos de construção.

Fase #	Descrição	Custo	Prazo
1	Estudo de viabilidade, desenvolvimento de projetos e obtenção das licenças necessárias	\$40 milhões	1 ano
2	Obras de infra-estrutura	\$200 milhões	2 anos
3	Construção da planta de geração eólica de 600 MW	\$400 milhões	2 anos
4	Ampliação da capacidade para 1000 MW	\$200 milhões	2 anos

Tabela 3: Seqüência de fases do projeto da planta de geração eólica

Os principais parâmetros necessários para resolver este problema são o valor inicial do projeto que foi estimado em \$564.53 milhões usando o método de calculo do fluxo de caixa descontado (**Tabela 4**). A taxa livre de risco que foi assumida como sendo de 6% ao ano, e a volatilidade, que é o percentual anualizado dos desvios padrões dos retornos do projeto que foi calculado em 36% para este projeto.

Ano 0	Ano 1	Ano 2	Ano 3	Ano 4	Ano 5
\$40 milhões	\$188.68 milhões	-	\$335.85 milhões	-	\$149.45 milhões

Tabela 4: Estimativa do valor do projeto pelo fluxo de caixa

No passo seguinte, calculamos os parâmetros para uma variação anual da aproximação binomial com taxa livre de risco e volatilidades de 6% e 36% respectivamente.

$$(i) u = e^{\delta\sqrt{dt}} = e^{0,36} = 1,43$$

$$(ii) d = 1 / u = 0,70$$

$$(iii) p = (1 + r - d) / (u - d) = 0,49$$

$$(iv) q = (1 - p) = 0,51$$

Usamos os parâmetros calculados na construção de uma árvore em grade que traduz a variação do valor do projeto seguindo um modelo GBM. O resultado desta árvore está apresentado na **Tabela 5** abaixo.

t_0	t_1	t_2	t_3	t_4	t_5
564.53	809.15	1159.78	1662.35	2382.70	3415.19
	393.86	564.53	809.15	1159.78	1662.35
		274.78	393.86	564.53	809.15
			191.71	274.78	393.86
				133.75	191.71
					93.32

Tabela 5: Estimativa de variação de custo do projeto

Em seguida, inserimos as opções e usamos uma árvore de decisão para calcular os possíveis valores do projeto, encontrando os valores apresentados na **Tabela 6** abaixo.

t_0	t_1	t_2	t_3	t_4	t_5
Investe \$40	Investe \$200		Investe \$400		Investe \$200
63.6 - 40 =23.6	336.9 - 200 =136.9	614.5	1484.4 - 400 =1084.4	2194.0	3415.2 - 200 =3215.2
	49.9 < 200 (não_investe)	107.4	631.2 - 400 =231.2	971.1	1662.4 - 200 =1462.4
		0.0 (não_investe)	217.8 < 400 (não_investe)	375.85	809.2 - 200 =609.2
			41.85 < 400 (não_investe)	90.1	393.9 - 200 =193.9
				0.0 (não_investe)	191.7 < 200 (não_investe)
					93.3 < 200 (não_investe)

Tabela 6: Árvore de decisão com o cálculo da variação de custo do projeto

O valor presente do projeto foi estimado em \$23,63 milhões e os resultados sugerem que o investimento seja tomado e que seja assumida a opção de ampliar a capacidade da planta de geração eólica para 1000 MW.

Nesta seção apresentamos uma das metodologias de avaliação de projetos através de análise de opções reais, que é um dos problemas de computação financeira que demandam grande capacidade de processamento. Na próxima seção (**Seção 4**) apresentaremos os problemas de avaliação de árvores de decisão e como o uso de *streaming processors* podem reduzir o tempo de análise dos projetos.

4 Métodos de Avaliação de Risco

Esta seção apresenta dois métodos de avaliação usados na análise de projetos usando opções reais, o uso de árvores de decisão e simulações de Monte Carlo.

4.1. Árvore de Decisão

As árvores de decisão são metodologias largamente utilizadas em análise financeira para avaliar opções, e constituem uma grande fonte de estudo do ramo da inteligência artificial e banco de dados, sendo usada em soluções para sistemas especialistas e em soluções de mineração de dados.

Uma árvore de decisão é constituída por nós de decisão, representados graficamente por quadrados, e por nós de incertezas que possuem probabilidades atribuídas a cada uma das possibilidades e são representados por círculos, como pode ser visto na **Figura 10**.

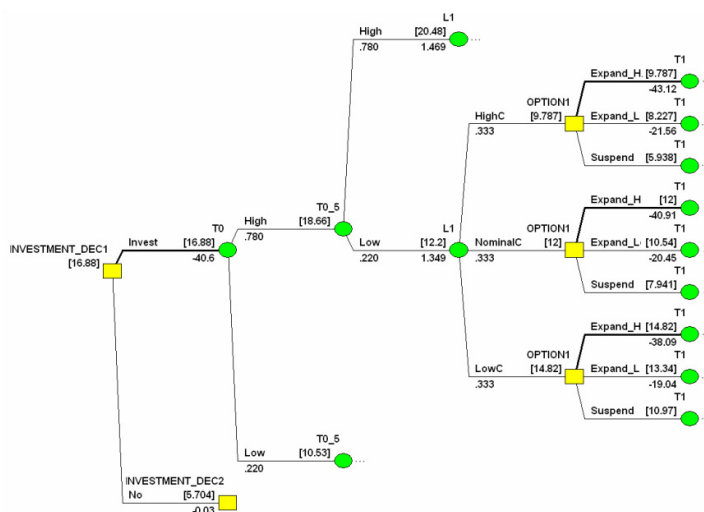


Figura 10: Exemplo de árvore de decisão de um modelo de opção real.

Fonte: [Tan, Anderson, Dyer, Parker 2009]

Para resolver um problema baseado em árvores de decisão os algoritmos seguem um caminho que vai das folhas até o nó raiz, atribuindo os valores com base no valor esperado para cada nó de decisão.

Em uma árvore de decisão os resultados são independentes e estão em um espaço completo, isto é, não existe intercessão entre dois resultados quaisquer, e a soma das probabilidades de ocorrência de cada resultado é igual ao espaço de respostas.

Os algoritmos utilizados para resolver problemas baseados em árvores de decisão são algoritmos gulosos que exigem grande capacidade de processamento e memória, sendo problemas do tipo NP- completos e que muitas vezes são resolvidos por meio de heurísticas.

O principal problema com as árvores de decisão esta na dificuldade em modelar problemas complexos, pois para problemas com 10 níveis de decisões a árvore binária chega a ter cerca de $2^{10} - 1 = 1023$ nós, dificultando bastante a atribuição das decisões.

4.1.1. Calculo da Árvore de Decisão

Para calcular uma árvore de decisão os algoritmos devem seguir das folhas até o nó raiz, onde para cada nó intermediário é calculado o valor esperado em função dos valores dos nós filhos seguindo três regras.

(i) **Nós de incerteza.** Para os nós de incerteza são calculados o valor esperado com base nas probabilidades das opções.

(ii) **Nós de decisão.** Substitua os nós de decisão pela opção que resolve o problema para o valor esperado. Isto é, se o valor esperado é maior que o valor de corte do nó de decisão, este valor é assumido para o nó, e em caso contrário é assumido o valor zero para a opção e a decisão não satisfaz a condição.

(iii) **Custos.** Quando um tronco da árvore possuir um custo, subtraia o custo do valor do nó de decisão.

Esta regra de resolução de árvore de decisão permite chegar ao valor estimado do projeto com opções que é uma medida mais adequada que a

aplicação direta do fluxo de caixa descontado na avaliação do projeto e inclui as opções que podem ser aplicadas e as incertezas.

O método de análise por árvore de decisão é um método estatístico largamente utilizado na resolução de problemas de análise de opções reais, e analisaremos a seguir a simulação de Monte Carlo que também é uma metodologia empregada na solução de problemas de avaliação de projetos.

4.2. Simulação de Monte Carlo

Embora possamos resolver inúmeros problemas de avaliação de projetos usando árvores de decisão, esta metodologia traz alguns problemas. Em alguns modelos a quantidade de nós da árvore pode crescer muito e tornar o problema inviável do ponto de vista computacional e para estes modelos a utilização de uma solução baseada em simulação seria mais adequado.

A simulação de Monte Carlo é um método estatístico que realiza diversas interações com valores aleatórios para obter o valor esperado do modelo, e possui algumas vantagens sobre o uso de árvores de decisão.

O método de simulação de Monte Carlo possui dois componentes, um modelo que representa o projeto e que é utilizado na avaliação do resultado, e as variáveis aleatórias que seguem uma determinada distribuição e que são utilizadas como parâmetros de entrada do modelo para gerar o resultado.

O principal objetivo da simulação é gerar um grande número de resultados para obter uma melhor perspectiva do resultado do projeto. Para isso é necessário realizar diversas iterações, usando diferentes valores de entrada para obter um resultado que melhor represente a distribuição do valor esperado para o projeto. A **Tabela 7** apresenta os passos necessários no processo de simulação.

(i) Obter amostras de variáveis aleatórias que representam as incertezas do modelo.
(ii) Substituir os valores das variáveis aleatórias no modelo e obter o resultado.
(iii) Armazenar os resultados em um arquivo de saída.
(iv) Repetir as iterações (i) a (iii) um número suficiente de vezes.
(v) Analisar a distribuição e obter os valores de saída.

Tabela 7: Passos necessários para a simulação de Monte Carlo

Quando o processo de modelagem é finalizado obtemos a aproximação do valor esperado do projeto e a frequência de distribuição. A média dos valores presentes calculados se aproxima do valor esperado com o aumento no numero de iterações que melhora a precisão da simulação.

4.2.1. Gerando Valores Aleatórios

Um dos principais fatores que influenciam o resultado da simulação é a representação das incertezas utilizando variáveis aleatórias. Estas variáveis podem representar distribuições discretas ou contínuas, e os geradores de números aleatórios fornecidos nos programas de computador geram valores que representam uma distribuição uniforme.

Para transformar uma distribuição uniforme com valores entre 0 e 1 em uma distribuição discreta, podemos criar condições que garantam esta ocorrência. A **Tabela 8** apresenta as probabilidades de variação do inicio de um projeto e representam valores discretos.

Atrasos para o Inicio das Obras	Probabilidade
3 meses	0,30
4 meses	0,50
5 meses	0,20

Tabela 8: Probabilidades de variação do inicio de um projeto

Existem outras variáveis que seguem distribuições contínuas como, por exemplo, o tempo esperado para finalizar uma atividade que pode ser concluída em até duas semanas. Esta variável segue uma distribuição aleatória que pode ser modelada através de uma distribuição normal.

Para transformar os valores aleatórios de uma distribuição uniforme em valores que representam outra forma de distribuição, nós podemos usar a função de probabilidade acumulada e mapear os valores 0 e 1 fornecidos pela distribuição uniforme em valores aleatórios que representam a outra distribuição. A **Figura 11** abaixo apresenta a distribuições de probabilidade acumulada para uma distribuição normal com diferentes médias e desvios padrões.

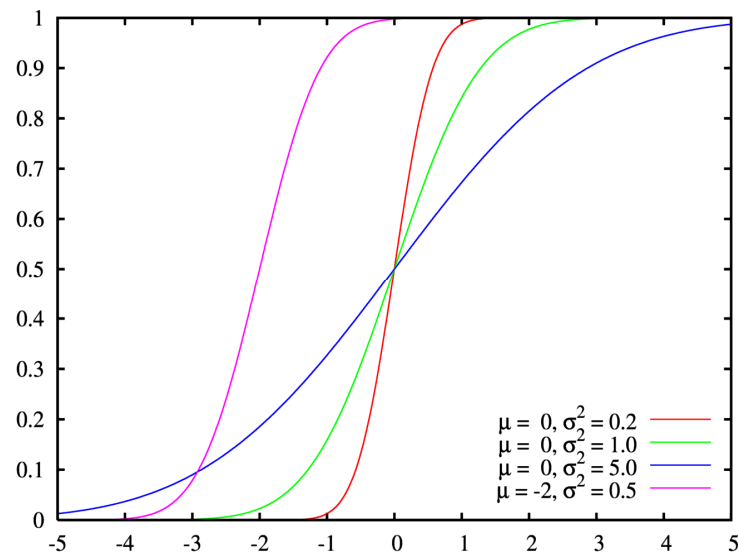


Figura 11: Distribuição de probabilidade acumulada de uma normal.

Nesta seção apresentamos algumas das principais ferramentas utilizadas na resolução de problemas de computação financeira e avaliação de risco, o uso de árvores de decisão e o método de simulação de Monte Carlo, na próxima seção estudarão os principais fabricantes de *stream multiprocessors*.

5 Principais Fabricantes

Os principais fabricantes de soluções de computação de alto desempenho utilizando GPU são NVIDIA, AMD, S3 Graphics e SGI, e nesta seção descreveremos o portfólio de produtos e a estratégia da NVIDIA e AMD para este mercado.

5.1. NVIDIA

A NVIDIA possui uma linha de placas gráficas dedicadas a diversas funcionalidades, para utilização em jogos e em programas de engenharia, e atualmente eles desenvolveram uma nova arquitetura de *stream multiprocessors* que está sendo utilizada para o desenvolvimento de aplicações que exigem grande capacidade de processamento.

O portfólio de produtos da NVIDIA inclui três linhas de placas gráficas, as placas da linha GeForce que são dedicadas a jogos e entretenimento, a linha Quadro que é voltada para uso profissional e a linha Tesla que foi desenvolvida para atender o segmento de computação paralela de alto desempenho.

GeForce

A GeForce é uma linha de placas gráficas desenvolvidas pela NVIDIA para atender ao segmento de entretenimento e jogos. Esta linha de placas está dividida em três categorias. A principal categoria voltada para jogadores que desejam obter a melhor experiência nos jogos é a GeForce GTX, que possuem excelente desempenho chegando a ter até 512 núcleos de processamento CUDA e cerca de 1,5 GB de memória.

Em seguida, a NVIDIA possui a categoria de placas GeForce GTS que possui um desempenho intermediário e voltado para o mercado de jogos, e uma categoria de entrada que é utilizada por quem deseja ter uma placa de vídeo com bom desempenho para assistir a filmes e brincar com os jogos atuais obtendo uma boa experiência. Esta linha de placas é voltada para o usuário

doméstico e são utilizadas em computadores de mesa e também nos computadores portáteis, onde a NVIDIA possui uma linha com baixo consumo de energia.

Quadro

A linha de placas de vídeo Quadro é voltada para uso profissional, são placas com grande poder de processamento e que possuem grande capacidade de memória que permite o uso mais profissional desta linha. As placas da linha Quadro estão separadas em quatro grupos com objetivos de atender diferentes segmentos.

A principal categoria é a Ultra High-End que é voltada para usuários profissionais que trabalham com processamento de imagens em tempo real, estas placas possuem 448 núcleos de processamento CUDA e 6 GB de memória que possibilitam o seu uso em aplicações que precisam de grande poder de processamento.

As placas Quadro possuem ainda uma categoria High-End que também é voltada para as aplicações que trabalham com um grande volume de processamento. Nesta categoria estão placas com até 384 núcleos de processamento CUDA e cerca de 2 GB de memória, e uma linha Mid-Range composta pelo modelo Quadro 2000 que possui 192 núcleos de processamento e 1 GB de memória. As placas Quadro também possuem uma linha de entrada voltada para aplicações de engenharia.

Tesla

As placas da linha Tesla são dedicadas ao mercado de computação de alto desempenho e podem ser combinadas com as placas Quadro para aumentar o poder de processamento das Workstations.

Com o lançamento das placas Tesla a NVIDIA entrou no mercado de computação de alto desempenho com o lançamento de uma arquitetura composta por hardware e software, que possui cerca de 510 núcleos de processamento e 6 GB de memória.

5.1.1.Estratégias de Mercado

As placas da linha GeForce são comercializadas em lojas de varejo e são largamente adotadas por jogadores e pelo público doméstico devido ao grande poder de processamento e ao baixo custo.

A linha GeForce também serve de linha de entrada para os usuários profissionais que após utilizarem os produtos NVIDIA para jogos e entretenimento, procuram seus produtos também no meio profissional.

Os profissionais das áreas de engenharia e entretenimento precisam de soluções profissionais que são atendidas pelas placas da linha Quadro, estas placas são vendidas através de um canal de parceiros especializados que fornecem soluções completas com Workstations dimensionadas adequadamente para o tipo de trabalho que se deseja realizar, enquanto as placas Tesla são comercializadas por um canal seletivo de revendedores que fornecem soluções completas de computação de alto desempenho. Entre os revendedores desta linha de produtos estão nomes como IBM e HP.

A venda de placas da linha Tesla envolve a comercialização de uma solução completa de aplicações que fazem uso da capacidade de processamento destas placas e possuem soluções de servidores com clusters de placas gráficas dedicados a este tipo de processamento.

5.2.AMD

A AMD possui uma linha de placas gráficas que atendem a diversas necessidades, tais como na utilização em jogos, em programas de engenharia e atualmente foi desenvolvida uma arquitetura de *stream multiprocessors* que concorre com a solução da NVIDIA para processamento de alto desempenho e utiliza a especificação OpenCL para desenvolvimento de soluções de computação em GPU.

A AMD possui duas linhas de placas gráficas, a linha ATI Radeon que é voltada para jogos e entretenimento, e a linha ATI Fire que é indicada para uso profissional.

ATI Radeon

A linha ATI Radeon é uma linha de placas voltadas para uso doméstico por quem deseja obter uma boa experiência no uso de placas gráficas para atividades de jogos e entretenimento.

Estas placas também são encontradas em soluções on-board integradas nas arquiteturas das placas principais dos computadores que possuem processadores da AMD, o que fornece ao usuário doméstico uma boa experiência por um custo baixo quando comparado com os concorrentes.

As placas da linha ATI Radeon possuem quatro categorias, desde placas para uso comum que exigem pouco recurso gráfico até uma poderosa linha de placas 3D com grande capacidade de processamento, e que pode ser utilizada também de forma profissional.

ATI Fire

As placas da linha ATI Fire são voltadas para o público profissional que precisa de desempenho e de grande capacidade de memória para processamento gráfico.

Esta linha de placas está dividida em três categorias, uma linha de entrada para usuários de aplicações de engenharia que fazem pouco uso de placas gráficas, uma linha intermediária para aplicações que necessitam de grande capacidade de processamento e melhor desempenho na criação de imagens foto realísticas, além de uma terceira linha profissional voltada para os usuários que precisam de grande desempenho na criação de imagens foto realísticas em tempo real.

Estas placas ATI Fire Pro também são utilizadas em soluções de computação de alto desempenho onde são oferecidas de forma integrada a arquiteturas de servidores com múltiplos processadores. Nestas soluções são oferecidos os computadores e a solução integrada para gerenciar o processamento na GPU.

5.2.1. Estratégias de Mercado

As placas da linha ATI Radeon são facilmente encontradas em lojas de varejo, atendendo a uma grande quantidade de aplicações para uso doméstico. Estas placas também são encontradas como produtos integrados com a placa principal do computador, oferecendo uma poderosa solução por um preço significativamente inferior as soluções equivalentes encontradas no mercado.

Esta estratégia da ATI favorece a adoção das placas gráficas e servem como uma plataforma de entrada para soluções corporativas. Neste segmento a ATI oferece soluções baseadas nas placas ATI Fire que são fornecidas por revendedores qualificados que oferecem estas placas como parte de uma solução gráfica de Workstation para trabalhos gráficos profissionais.

As placas ATI Fire Pro são as placas de alto desempenho da ATI e são utilizadas em Workstations e em servidores para computação de alto desempenho, onde estas placas compõem os recursos de processamento de *stream* para cluster de servidores e são adotadas por nomes como IBM e Cray.

5.3. Mercado de Computação Baseada em GPU

No mercado atual de computação de alto desempenho, fabricantes com IBM, DELL, HP, BULL, SGI e CRAY já possuem soluções de supercomputadores baseados em GPU, e possuem também soluções prontas para atender ao segmento corporativo.

As principais soluções exploram o elevado grau de processamento paralelo encontrado nas GPUs e estão utilizando *clusters* de GPUs em supercomputadores. A vantagem de utilização das GPUs está na alta capacidade de processamento com um consumo baixo de energia, sendo este um fator determinante na sua adoção em supercomputadores.

Um dos principais supercomputadores atuais o Tianhe-1A que possui 7168 placas NVIDIA Tesla M2050, poder de processamento de 2.5 Petaflops, e alta eficiência quando comparamos a relação entre a capacidade de processamento e o consumo de energia. As placas NVIDIA Tesla estão presentes nas principais soluções HPC oferecendo a melhor eficiência.

Outra tendência que vem sendo explorada pela Amazon Web Services é a disponibilidade de soluções de supercomputadores na nuvem, com o compartilhamento de processamento de *clusters* de GPU através da Internet. Nestas soluções as empresas contratam capacidade de processamento para execução de aplicações. Atualmente, diversos fornecedores de soluções de renderização oferecem serviços na nuvem que exploram a capacidade de processamento dos *clusters* de GPU. Um dos principais serviços é o projeto NEON da Autodesk que possui servidores nas nuvens que são utilizados por clientes de soluções de mídia e entretenimento quando desejam renderizar cenas de filmes.

A receita obtida pela NVIDIA com o fornecimento das placas de vídeo Tesla para uso em sistema de supercomputadores vem crescendo de forma significativa desde o lançamento desta arquitetura em 2008, chegando ao valor de \$100 milhões em 2011, um valor superior ao resultado financeiro obtido com as vendas de placas Quadra que são utilizadas em Workstations, mostrando uma tendência de crescimento na utilização desta tecnologia, **Figura 12**.

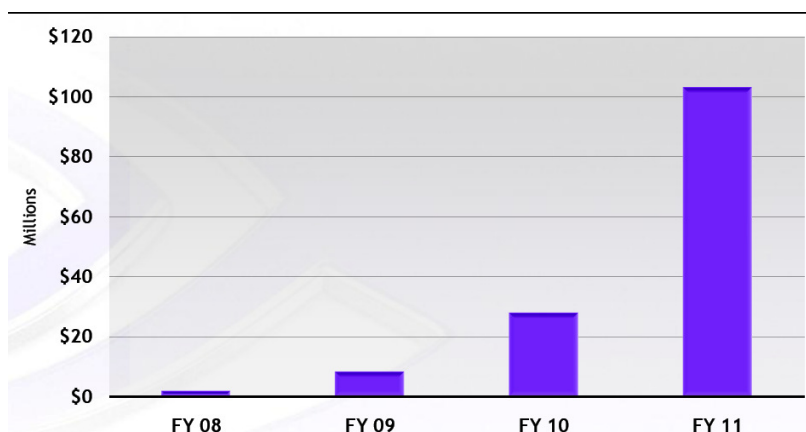


Figura 12: Resultado das vendas de placas Tesla.

Fonte: [UBS Global Technology and Services Conference]

Existe uma grande oportunidade no segmento corporativo que está crescendo com a popularidade de uso das placas NVIDIA Tesla neste segmento, **Figura 13**.

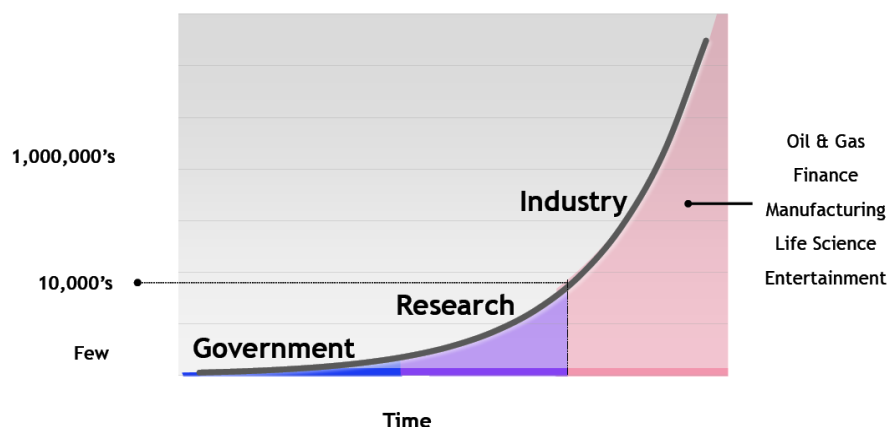


Figura 13: Crescimento no uso de supercomputadores por indústria.

Fonte: [UBS Global Technology and Services Conference]

As oportunidades em cada mercado chegam a \$500 milhões em soluções para aplicações financeiras. Com grande perspectiva na área sísmica onde cerca de 20000 GPUs devem ser vendidas em 2011 para atender este segmento.

Aplicação Científica	Análise Sísmica	Governo / Defesa	Biologia e Química	Indústria Metalúrgica	Finança
\$600 milhões	\$300 milhões	\$600 milhões	\$300 milhões	\$300 milhões	\$500 milhões

Tabela 9: Expectativa de receitas com vendas de placas Tesla em 2011

Atualmente existem 200.000.000 de placas gráficas com suporte a tecnologia CUDA usada pela NVIDIA no mercado o que fornece uma base boa para o crescimento de soluções que utilizam esta tecnologia

Nesta seção apresentamos uma perspectiva do mercado que está sendo explorado pela NVIDIA em computação baseada em GPU para o mercado corporativo, e na próxima seção apresentaremos as conclusões e trabalhos futuros.

6 Conclusão

O mercado de computação baseada em GPU é um mercado recente, mas que apresenta um grande potencial de crescimento e está sendo fortemente adotado por diversas empresas que necessitam de soluções de computação de alto desempenho.

Neste mercado tanto a NVIDIA quanto a AMD possuem soluções extremamente eficientes e compatíveis com a especificação OpenCL que fornece um padrão para desenvolvimento de aplicação baseada em GPU.

Neste mercado a NVIDIA está mais bem posicionada com uma ligeira vantagem devido ao grande investimento que está sendo realizado no desenvolvimento de novas soluções que utilizam o poder das GPU para resolver problemas que não são necessariamente de computação gráfica.

Em computação financeira os recursos computacionais das GPUs são de grande valor, pois permitem que simulações sejam realizadas em um menor tempo, além de permitir a criação de modelos cada vez mais complexos e que podem ser analisados com maior frequência.

Neste trabalho estudamos problemas relacionados com opções reais onde as soluções se baseiam em árvores de decisão e o grau de complexidade é da ordem de $O(2^n)$, e que dependendo do período de análise, pode exigir máquinas com grande capacidade de processamento para realizar a tarefa em um tempo razoável.

Podemos desta forma, concluir que os novos conceitos que estão sendo introduzidos pelas tecnologias de processamento de GPU estão fornecendo uma nova dimensão para as análises e simulações que podem ser realizadas no campo da computação financeira.

6.1. Sugestão e Recomendações para Novas Pesquisas

Este trabalho apresentou uma nova oportunidade de mercado para o desenvolvimento de soluções para o mercado financeiro que demandam grande poder de processamento usando a arquitetura dos *stream multiprocessors*.

Este estudo mostra a crescente necessidade de processadores rápidos para resolver problemas de computação financeira com o aumento na complexidade do modelo e abre novas perspectivas de utilização destas metodologias com maior frequência no estudo de viabilidade de projetos.

O surgimento de computadores com grande capacidade de processamento por um custo baixo fornece uma grande oportunidade de desenvolvimento de ferramentas de análise para modelos mais complexos que representam resultados esperados mais precisos.

Este trabalho foi apenas o início de um estudo desta tecnologia e do mercado que está sendo explorado pela NVIDIA, e os próximos passos serão a complementação deste estudo com o desenvolvimento de aplicações que aproveitam estes recursos.

7 REFERÊNCIAS BIBLIOGRÁFICAS

BEN-HAIM, Yael; TOM-TOV, Elad. A Streaming Parallel Decision Tree Algorithm. Journal of Machine Learning Research 11th, 2010

BLACK, Fisher; SCHOLES, Myron. The Pricing of Options and Corporate Liabilities. Journal of Political Economy 81th, 1973

BRANDÃO, Luiz E.; DYER, James S.; HAHN, Warren J. Using Binomial Decision Tree to Solve Real-Option Valuation Problem. Decision Analysis - Vol. 2, No 2, 2005

BRANDÃO, Luiz E.; DYER, James S. Projetos de Opções Reais com Incertezas Correlacionadas. BASE - Revista de Administração e Contabilidade da Unisinos, Ano 6, Vol 1, 2009

BRANDÃO, Luiz E.; DYER, James S. Decision Analysis and Real Options: A discrete Time Approach to Real Option Valuation. Annals of Operations Research, 135, 2005

COPELAND, Tom; TUFANO, Peter. A Real-World Way to Manage Real Options - Harvard Business Review 82th, 2004

COX, John C.; ROSS, Stephen A.; RUBINSTEIN, Mark. Option Pricing: A Simplified Approach. Journal of Financial Economics 7, 1979

NVIDIA's Next Generation CUDA Computer Architecture: Fermi. (2010)

OpenCL Programming Guide for the CUDA Architecture. Version 3.2 (2010)

ONODA, Mauricio; EBECKEN, Nelson F. Implementando em Java um Algoritmo de Árvore de Decisão Acoplado a um SGBD Relacional. XVI Simpósio Brasileiro de Banco de Dados, 2001

PROTTER, Philip; LAMBERTON, Damien; CLÉMENT, Emmanuele. An Analysis of the Longstaff-Schwartz Algorithm for American Option Pricing. Finance Stock 6th, 2002

QUILAN, J. R. Simplifying Decision Trees. AI Memos, 1986

SCHUYLER, John. Risk and Decision Analysis in Projects - 2nd Edition, 2001

SMITH, James E. Alternative Approaches for Solving Real-Options Problems. Decision Analysis - Vol. 2, No. 2, 2005

TAN, Burcu; ANDERSON, Edward; DYER, James; PARKER, Geoffrey. Using Binomial Decision Trees and Real Options Theory to Evaluate System Dynamics Models of Risky Projects. Proceedings of the 27th International Conference of the System Dynamics Society, 2009