

Projeto de Pesquisa para Iniciação Científica

Comparando a utilização de soluções livres e proprietárias para programação geral de GPUs

Isabella Basso do Amaral

Orientador: Alfredo Goldman

Instituto de Matemática e Estatística

Universidade de São Paulo

São Paulo, Brasil

gold@ime.usp

Resumo

A adoção, por parte da comunidade científica, de *software* com intuito de aproximar soluções para problemas relevantes no cenário contemporâneo traz consigo desafios além dos técnicos, principalmente, no que diz respeito à utilização de soluções livres para a execução de tais pesquisas. Esta é essencial para que não violemos princípios básicos da pesquisa científica como a entendemos nos dias de hoje, como por exemplo o princípio da reprodutibilidade. No presente projeto, então, verificamos a distribuição de pesquisas em relação à sua utilização de soluções livres ou proprietárias, especificamente no contexto de *GPU General Programming* (GPGPU), assim como as disparidades entre opções livres e proprietárias que dificultam o cenário ideal. A partir deste ponto, exploramos os passos necessários para facilitar a adoção de soluções livres, assim como o seu desenvolvimento.

Palavras-chave

GPU, CUDA, Vulkan, Compute, OpenCL, software, livre, aberto, proprietário

I. INTRODUÇÃO

O computador tornou-se parte indispensável da pesquisa acadêmica, seja facilitando a leitura de artigos e livros ou “resolvendo” (aproximando) problemas numéricos, os quais são, em grande maioria, insolúveis sob o olhar analítico.

A aproximação de soluções para problemas numéricos em *software*, embora não seja uma questão nova ou sequer teórica, demanda a utilização de técnicas avançadas de programação exigindo, por vezes, o uso de *software* extremamente particular, complexo e que demanda conhecimento técnico específico, tanto da teoria quanto do *software* em questão.

A. Otimização

Também no que diz respeito aos problemas numéricos, gostaríamos de ressaltar, em especial, a possibilidade de paralelizá-los, agilizando (por vezes desproporcionalmente) a produção de resultados.

Em diversos casos, tais otimizações são necessárias para que possamos produzir resultados oportunamente, levando em conta simplesmente a capacidade de paralelização do *hardware* moderno *versus* suas capacidades de execução linear, notamos que frequentemente ele é pelo menos milhares de vezes mais capaz na primeira categoria do que na segunda.

A GPU (*Graphics Processing Unit*) é um componente especializado em tarefas de paralelização, de tal forma que, atualmente, é utilizada extensivamente para auxiliar na resolução de problemas nas áreas de física, biologia, química e até mesmo da matemática moderna sendo, por vezes, mais relevante do que a CPU (*Central Processing Unit*) de um dado sistema em aplicações como aquelas da ciência de dados.

B. Sobre o uso de software livre em aplicações científicas e sua importância

No que tange a utilização de *software* para aplicações científicas, nos preocupamos principalmente com a utilização de *software* livre, que, estritamente falando, é aquele que se alinha perfeitamente com os princípios da pesquisa científica.

Chamamos de ***software* livre** aquele que é aberto para ser lido e auditado, e que não depende de corporações (embora possa ser auxiliado por estas) para que se mantenha. Possuindo comunidades autônomas de usuários e desenvolvedores que o mantêm de acordo com interesses pessoais e plurais, porém não necessariamente sem o envolvimento de capital.

O *software* livre permeia todo o contexto computacional moderno, sendo utilizado extensivamente na internet (e.g. ≈ 80 dos servidores utilizam *Linux*), no contexto do desenvolvimento de *software* e, especialmente, na pesquisa científica.

Contrastamos tal com o ***software* proprietário**, onde a contribuição de um indivíduo sempre será atrelada com interesses corporativos pois pertence à uma empresa e é secreto sendo, portanto, impossível auditá-lo.

Existe ainda o *superset* de *software* livre, que é o ***software* aberto**, onde pode existir uma empresa que o mantêm de acordo com seus interesses, porém este não é secreto, podendo ser auditado e aceitando contribuições individuais¹.

Note, então, que a utilização de *software* proprietário não cabe no contexto científico, pois demanda desmedida confiança à interesses corporativos, indo totalmente de encontro a princípios fundacionais da ciência como a temos hoje, onde reprodutibilidade e transparência são essenciais.

Com isso em mente, advogamos pelo uso exclusivo do *software* livre nesse contexto o que, infelizmente, ainda não é totalmente plausível como será explorado mais a frente.

II. JUSTIFICATIVA

Haja vista que a GPU é um componente de *hardware* sendo, então, necessário comandá-la de forma específica e precisa, idealmente gostaríamos de abstrair detalhes do *hardware* específico, de tal forma que tenhamos fino controle sobre sua utilização, sem comprometer, porém, seu desempenho (i.e. a abstração deve otimizar o que quer que seja tendo em mente o *hardware* específico).

¹No presente projeto, no entanto, vamos nos dedicar exclusivamente à problemática do *software* livre X *software* proprietário.

A. Compiladores

Classicamente, utilizamos *shaders* para a programação de GPUs, sendo mais ou menos análogos à programação de CPUs em sua sintaxe, mas não necessariamente em sua lógica, o que torna seu uso não-trivial para alguém sem o conhecimento específico. O código do *shader* é compilado em etapas, sendo primeiramente convertido para uma linguagem intermediária, onde é otimizado com base em conceitos primitivos próprios para abordagens de paralelização. Depois, sendo convertido para um binário otimizado para a GPU específica onde será executado, e então enviado para o *hardware*.

B. Comparando o software livre com o proprietário para aplicações científicas

No contexto específico de aplicações numéricas, no presente projeto nosso interesse é voltado à **programação geral** com GPUs (GPGPU), onde temos utilização extensiva da linguagem **CUDA**, desenvolvida pela empresa NVIDIA – uma pioneira em computação gráfica. Dada sua simplicidade e abrangência, o CUDA segue invicto em aplicações científicas que utilizam GPUs.

Gostaríamos de notar, no entanto, que essa é uma tecnologia proprietária, e que sua adoção supera a de alternativas livres por uma grande margem sendo, então, de grande importância para a comunidade científica contemporânea que existam alternativas viáveis e livres para essas aplicações, como dito em I-B.

III. OBJETIVOS

Dada a utilização desproporcional do CUDA em relação à alternativas livres, possuímos diversos questionamentos que devem ser abordados no presente projeto, como:

- 1) Qual a figura exata da adoção de *software* livre para aplicações científicas?
- 2) Onde o CUDA se sobressai em comparação à essas alternativas?
- 3) Como podemos diminuir essa diferença e como facilitar a migração para soluções que usam exclusivamente código livre?
- 4) O cenário ideal é factível?

Como alternativas (livres) ao CUDA, destacamos o **OpenCL** e o **Vulkan**, os quais serão utilizados durante o projeto para aperfeiçoamento e comparações.

IV. METODOLOGIA

V. CRONOGRAMA

REFERÊNCIAS

- [1]
- [2]
- [3]
- [4]
- [5]
- [6]
- [7]