inesc id lisboa

Recursos Humanos

RELATÓRIO DE ACTIVIDADES DE BOLSA DE INVESTIGAÇÃO

(A preencher pelo Bolseiro)

RELATÓRIO INTERCALAR ☐ RELATÓRIO FINAL ☑

Identificação do bolseiro

Nome completo: Pedro Guilherme Antunes Diogo

Identificação da bolsa

Tipo de bolsa: Bolsa de Introdução à Investigação Referência: BII_2008

Período: De: 2008 - 10 - 31 a: 2009 - 10 - 31

Nome do projecto: Desenvolvimento e Análise de Algoritmos Para Modelação de Sistemas Dinâmicos Não Lineares

Área de trabalho: Engenharia Electrotécnica e de Computadores

Orientador científico: Luís Miguel d'Ávila Silveira

Actividades desenvolvidas

(Descreva sucintamente as principais actividades desenvolvidas e resultados obtidos, usando como referência o Plano de Trabalhos aprovado para o período).

Este trabalho tem como objectivo implementar algoritmos de regressão na arquitectura de processamento paralelo CUDA e comparar essas mesmas implementações com as alternativas existentes para CPUs.

Numa primeira fase, o trabalho desenvolvido teve como objectivo a introdução de conceitos e aprendizagem da arquitecura de processamento paralelo CUDA. Para tal foi criado um programa com vista a implementar o método dos mínimos quadrados, usando a arquitectura de processamento paralelo CUDA em conjunto com o programa MATLAB. No decorrer desta fase e depois de criado o programa foi dado foco à decomposição de Cholesky para a resolução de sistemas lineares do tipo A*x=b, tendo sido efectuados alguns testes e melhorias no código em estudo.

Numa segunda fase, o trabalho desenvolvido teve em vista o estudo de Support Vector Machines na arquitectura CUDA. Para tal foi estudado um projecto existente - cuSVM - uma implementação de SVMs na arquitectura CUDA, baseada na biblioteca em CPU - LibSVM. De momento o software cuSVM só possibilita o uso de uma função kernel, sendo um dos objectivos implementar outras funções kernel, e o teste das mesmas, comparando os resultados obtidos com os da biblioteca em CPU - LibSVM.

(continuação na folha em anexo)

(continuar em folhas adicionais, se necessário)

Desvios em relação ao planeado e respectiva justificação

(Deverão ser mencionadas as circunstâncias que tenham influenciado positiva ou negativamente o cumprimento do plano de trabalhos)

Numa fase inicial não estava previsto que iria ser estudada a decomposição de Cholesky para a resolução de sistemas lineares do tipo A*x=b, contudo tal foi necessário pois a resolução deste tipo de sistemas ainda não estava disponível na biblioteca de operações básicas de algebra linear da aquitectura CUDA - cuBLAS. Tal desvio foi devido ao facto deste tipo de sistemas ser essencial para a implementação do método dos mínimos quadrados.

Publicações e trabalhos elaborados no âmbito da bolsa

(Caso se trate de uma bolsa para obtenção de grau ou diploma académico, juntar a este relatório uma cópia do respectivo trabalho final)

Os resultados dos trabalhos elaborados foram:

1. Implementação do método dos mínimos quadrados, disponível em:
   http://algos.inesc-id.pt/wikicuda/index.php/Código_Lançado:Método_dos_Mínimos_Quadrados
2. Implementação da decomposição de cholesky, com mais informações em:
   http://algos.inesc-id.pt/wikicuda/index.php/Código_Lançado:Decomposição_de_Cholesky
3. Comparação dos resultados obtidos entre cuSVM e LibSVM (por concluir)

(continuar em folhas adicionais, se necessário)

Bolseiro

Pedro Guilherme Antunes Diogo

Assinatura: Pedro Guilherme Antunes Diogo Data: 24 - 11 - 2009

Orientador Científico

Luís Miguel d'Ávila Silveira

Assinatura: Luís Silveira Data: 24 - 11 - 2009

CUDA, acrónimo de **C**ompute **U**nified **D**evice **A**rchitecture, é uma arquitectura de processamento paralela usando placas gráficas, desenvolvida nos laboratórios da empresa NVIDIA. A tecnologia CUDA foi introduzida em todas as placas gráficas da NVIDIA desde a sua linha GeForce 8.

Para que se possa dar uso à tecnologia CUDA será necessário programar numa linguagem suportada. Actualmente estão disponíveis as linguagens C, C++, Python, Fortran, Java, Matlab e a framework .NET da Microsoft.

A principal vantagem do processamento em placas gráficas, face ao processamento em CPUs será a velocidade. Como demonstrado no **Gráfico 1,** nos últimos anos houve uma explosão das GFLOPS (Operações de Vírgula Flutuante por Segundo, em Inglês **FL**oating point **O**perations **P**er **S**econd) nas placas gráficas, muito superior ao verificado nos processadores, sendo assim bastante aliciante para situações que envolvam grande quantidade de cálculos numéricos. Outra grande vantagem será o factor preço. Para este tipo de aplicações uma placa gráfica avaliada em 200€ poderá facilmente ser mais eficiente que um processador dedicado, avaliado em 1000€.

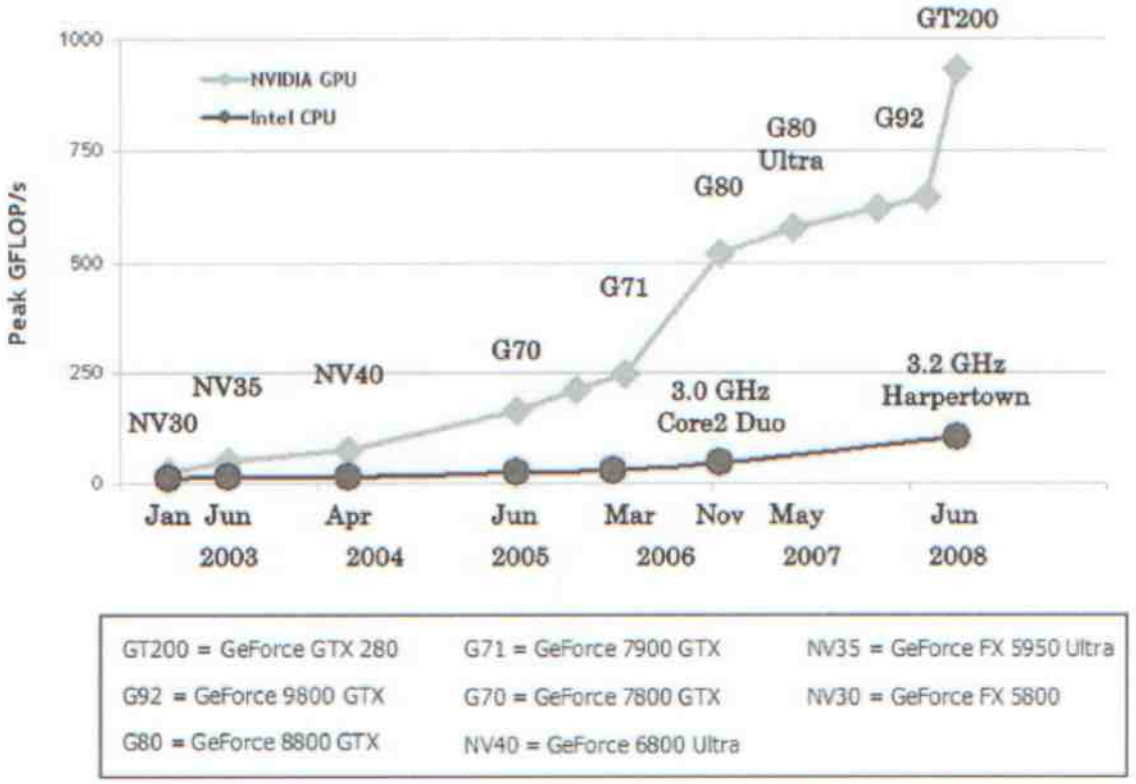

Gráfico 1 - Comparação de GFLOPS em diversos dispositivos. A azul CPUs, a verde placas gráficas

Existem contudo desvantagens. Apesar do suporte por parte da NVIDIA ser muito bom, existe o inconveniente desta tecnologia ser proprietária, e com um nicho de aplicações, reduzindo assim o suporte externo. Outra potencial desvantagem será a dificuldade de programação no meio. Existem ainda diversas bibliotecas por lançar que poderão vir a reduzir esta dificuldade.

Alguns projectos estão em relevo na secção da tecnologia na página da NVIDIA, e poderão dar um exemplo do que se encontra desenvolvido, ou em desenvolvimento, actualmente.

Método dos Mínimos Quadrados

O método dos mínimos quadrados é uma técnica que procura encontrar o melhor ajustamento de um conjunto de pontos dado, tentando minimizar a soma do quadrado das distâncias entre a curva a ser ajustada e os pontos dados.

Este método foi útil como introdução para técnicas de regressão mais avançadas, tais como as Support Vector Machines, discutidas de seguida.

No decorrer da implementação deste método foi verificado que grande parte da complexidade do algoritmo se resumia somente à resolução de um sistema do tipo A*x=b, desta forma foram concentrados os esforços para estudar uma forma de resolver este tipo de sistemas em placas gráficas. Tal deu origem ao estudo do desenvolvimento de Cholesky.

Implementação da Factorização de Cholesky

Como anteriormente referido a decomposição de Cholesky foi útil na resolução de sistemas lineares do tipo A*x=b. Este método decompõe uma matriz simétrica definida positiva, numa matriz triangulal, L e L' a sua transposta, tal que A=L*L'. Esta decomposição torna assim fácil a resolução do sistema A*x=b, pois transforma-o na resolução de dois sistemas lineares de matriz triangular, operação suportada pela biblioteca de algebra linear da arquitectura CUDA - cuBLAS.

O software usado foi uma modificação do software encontrado em [1] . A modificação teve em vista tornar possível a decomposição de Cholesky para matrizes quadradas de dimensão superior a 22 linhas e colunas.

Foram realizados alguns testes ao código modificado, comparando diversos parâmetros de um algoritmo em CPU, com a implementação na arquitectura CUDA tanto em precisão simples (float), como em precisão dupla (double).

Para os erros foi comparado o módulo da diferença entre o resultado da computação A=L*L', tendo sido L calculado a partir da decomposição de Cholesky, com a matriz original A que deu origem a L. Analisando os erros médios e máximos podemos concluir que as implementações double em CUDA e CPU são bastante idênticas, pois ambas usam precisão dupla. A precisão simples em CUDA tem erros superiores derivados da menor precisão. Analisando a performance verificamos que só é proveitoso a utilização de CUDA para matrizes de dimensão superior a 4000 linhas e colunas. A implementação em precisão dupla, à partida mais lenta que a implementação simples, revelou-se mais rápida para matrizes de grande dimensão.

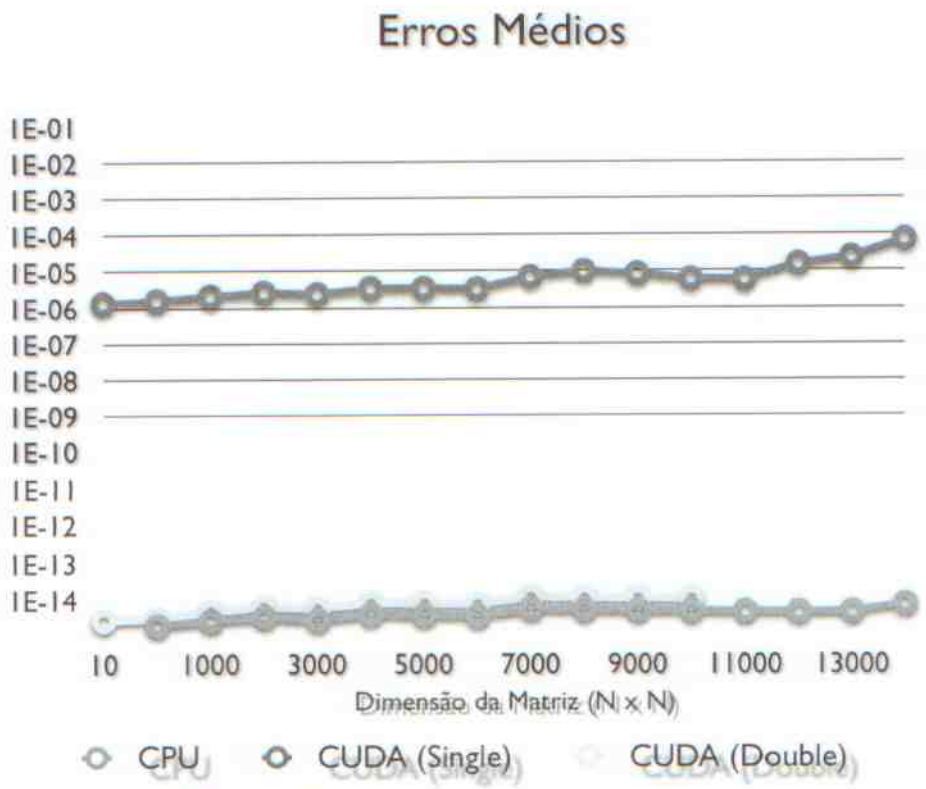

Gráfico 2 - Comparação dos erros médios dos resultados obtidos

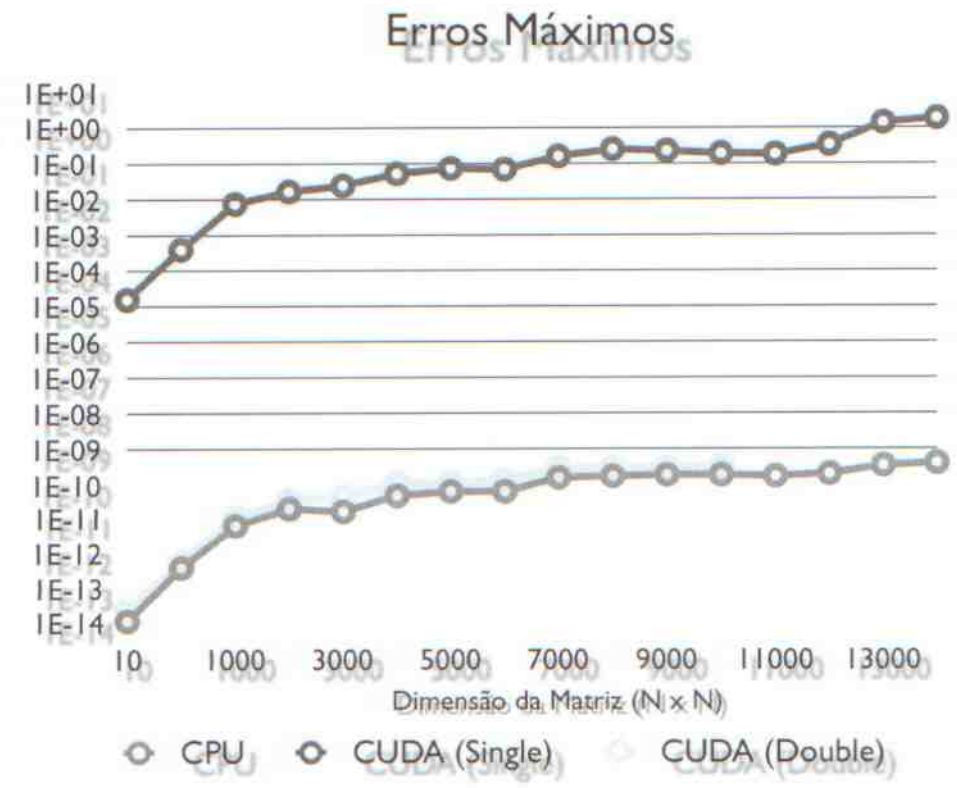

Gráfico 3 - Comparação dos erros máximos dos resultados obtidos

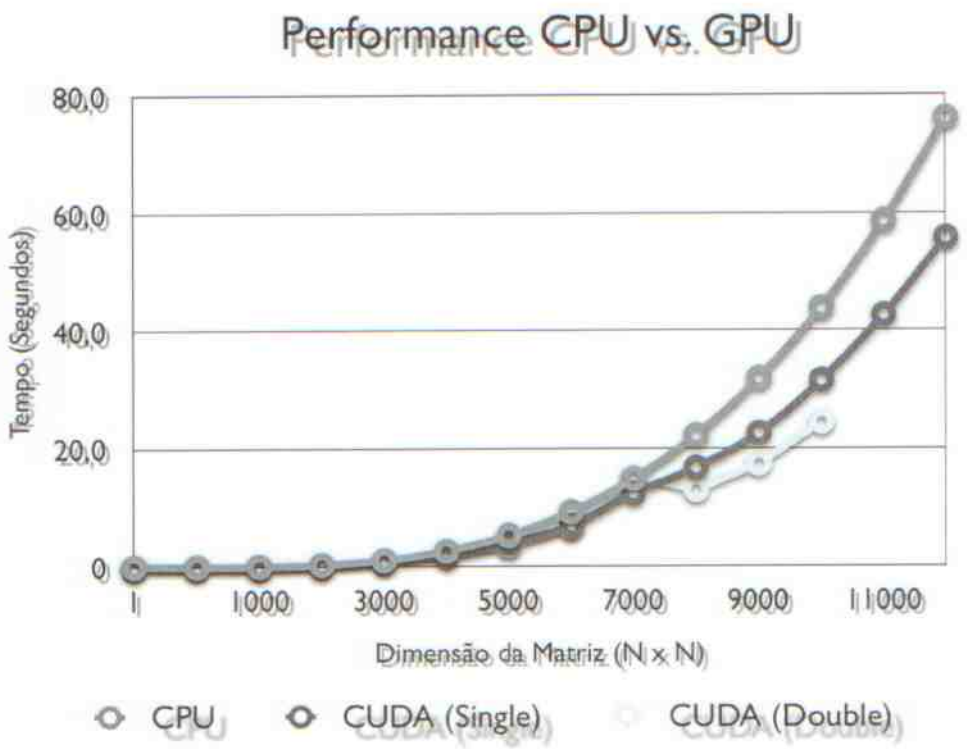

Gráfico 4 - Comparação da performance das implementações em CUDA e em CPU

Support Vector Machines

Support Vector Machines (SVMs), são um método de regressão e classificação que, baseado numa aprendizagem inicial - treino - classifica ou faz a regressão de determinados pontos num processo posterior.

Na fase de aprendizagem é utilizado um conjunto de pontos inicial - pontos de treino - para gerar um modelo válido para ser usado na classificação ou regressão de outro conjunto de pontos. Os pontos de treino consistem num conjunto de vectores em Rn aos quais estão associados valores contínuos ou discretos, caso sejam um conjunto de pontos para regressão ou classificação, respectivamente. No final pretende-se encontrar um hiperplano que separe da melhor forma os pontos de diferentes classes, caso seja pretendido classificação, ou o melhor ajuste aos mesmos, caso seja pretendida a regressão. Este processo depende de uma função - kernel - que ao variar também variará o modelo em que o hiperplano depende.

Após alguma investigação foi descoberto um projecto existente que implementava SVMs na arquitectura CUDA - cuSVM [2] - tendo sido decidido basear a nossa abordagem às SVM, estudando este software. Para comparação em CPUs foi usada uma biblioteca muito comum em SVMs - LibSVM [3] - de onde é baseado o software cuSVM.

Actividades desenvolvidas *(continuação)*

Actualmente os nossos esforços assentam na tentativa de implementar novos kernels de treino no software cuSVM, visto este só ter disponível um kernel Radial Basis Function, tendo já sido implementado um kernel polinomial.

Já foram também feitas algumas comparações entre o software cuSVM e a LibSVM, porém os resultados do software cuSVM revelaram-se muito diferentes aos esperados por outros estudos, logo não serão válidos de momento.

[1] - http://www.ast.cam.ac.uk/~stg20/cuda/cholesky/

[2] - http://patternsonascreen.net/cuSVM.html

[3] - http://www.csie.ntu.edu.tw/~cjlin/libsvm/