Resolução de Problemas Numéricos com Alto Desempenho: alternativas de implementação

Leila Diane Wentz, Andriele Busatto do Carmo,
Marcos José Brusso, Carlos Amaral Hölbig

Curso Ciência da Computação - Universidade de Passo Fundo
Campus I – BR 285 – Bairro São José CEP 91501-970 – Passo Fundo – RS - Brasil
{62408, 62384}@lci.upf.br , {brusso, holbig}@upf.br

Introdução

A resolução de problemas numéricos da computação científica envolve, em uma grande quantidade de casos, a manipulação de matrizes e vetores. Estes problemas são muito aplicados na análise de estruturas elétricas, na análise de sistemas mecânicos e no cálculo de parâmetros na construção civil, entre outros problemas na área da Física e Engenharia.

A manipulação de matrizes e vetores é composta, basicamente, de duas funções: somatório e produto escalar. Pensando em obter um melhor desempenho no tempo de resolução desse tipo de aplicações, pode-se fazer uso de algumas alternativas de implementação como, por exemplo, o uso de instruções ao nível de processador (SSE1, SSE2 e SSE3), uso de funções em *Assembly* e de diretivas de otimização dos compiladores.

O presente artigo visa apresentar e demonstrar essas alternativas de implementação que poderão ser utilizadas para solucionar problemas reais que demandam um alto desempenho. Conforme descrito no parágrafo anterior, as alternativas escolhidas visarão implementações utilizando o compilador *gcc* (usando diretivas de otimização), funções em *Assembly*, a linguagem C/C++ com a biblioteca BLAS e as instruções ao nível de processador SSE.

Instruções SSE

As instruções SSE (Streaming SIMD Extensions) são instruções ao nível de processador, especificamente de processadores da família Intel, utilizando a tecnologia NetBurst.

Microarquitetura NetBurst

A microarquitetura NetBurst oferece vários recursos, onde pode-se destacar:

- Tecnologia *Hyper Pipelined*: *pipeline* maior, composto por 20 estágios no caso do Pentium 4 e de 32 estágios no Pentium Prescott.
- Barramento de sistema de 400 MHz.

- *Execution Trace Cache Level 1*: cache que armazena até 12K de instruções decodificadas, reduzido o tempo total necessário para a recuperação das previsões de desvio incorretas.
- REE (*Rapid Execution Engine*): O clock das unidades lógica-aritméticas (ALU – *Arithmetic Logic Units*) é acionado com o dobro da freqüência do núcleo do processador, permitindo que instruções de inteiros básicas, como *Add* (adicionar), *Sub* (subtrair), *Logical* AND (E lógico) e *Logical* OR (OU lógico), executem utilizando metade de um ciclo de clock. Por exemplo, o REE em um processador Pentium 4 de 1,5 GHz executa a 3 GHz.

Essa microarquitetura também oferece as instruções SSE nível 2 e, mais recentemente, de nível 3:

- **SSE:** Foram as primeiras instruções a serem lançadas. Elas reduzem a quantidade geral de instruções necessárias para executar uma tarefa específica de um programa. Como resultado, elas podem melhorar o desempenho, acelerando uma ampla gama de aplicativos, incluindo aplicativos de processamento de vídeo, voz, foto e imagem, de criptografia, financeiros, de engenharia e científicos.
- **SSE2:** Com a introdução da microarquitetura NetBurst adicionou-se 144 novas instruções SSE, conhecidas como SSE2. Com esse aprimoramento, tornou-se possível a exibição e manipulação de imagens com resolução mais alta, áudio de alta qualidade, vídeo MPEG2, codificação/decodificação simultâneas de MPEG2, redução do uso da CPU para o reconhecimento de voz, precisão mais alta e tempos de respostas mais ágeis.
- **SSE3:** São 13 novas instruções do tipo MMX que agiliza funções de software tais como codificação de vídeo e conversão de números de ponto flutuante em inteiros.

```
void somatorio(float *a, float *b, float *c)
{
asm{
      mov eax, a
      mov edx, b
      mov ecx, c
      movaps xmm0, XMMWORD PTR [eax]
      addps xmm0, XMMWORD PTR[edx]
      movaps XMMWORD PRT [ecx], xmm0
     }
}
```

Figura 1 – Função de somatório utilizando as instruções SSE

As instruções MMX são SIMD para inteiros, enquanto as instruções SSE3 foram desenvolvidas como SIMD para números de ponto flutuante com precisão simples. As instruções MMX operam sobre dois números inteiros de 32 bits simultaneamente, enquanto as instruções de SSE3 operarem sobre quatro números de ponto flutuante de 32 bits cada. Para ilustrar o uso das funções SSE, a figura 1 descreve uma função para o cálculo do somatório.

Diretivas de Otimização do Compilador *gcc/g++*

Outra alternativa para tentar se obter uma melhora no desempenho é o uso de diretivas de otimização do compilador *gcc/g++* (como essa proposta visa a implementação de programas utilizando a linguagem de programação C/C++ optou-se pelo uso do compilador *gcc/g++* e, por este motivo, estará sendo abordada as diretivas de otimização desse compilador). A otimização do código de um programa é uma tentativa de melhorar seu desempenho, ao custo de um aumento no tempo de compilação, uso de memória e, em alguns casos, um aumento no tamanho do arquivo binário resultante. O *gcc* trabalha basicamente com três níveis de otimização. Cada nível é formado por um conjunto de *flags* de otimização, onde cada *flag* tem a função de informar ao compilador que um determinado tipo de otimização deve ser feito no código do programa. Cada um dos três níveis pode ser ativado ou desativado individualmente.

As otimizações efetuadas no nível 1 (-O1, ou -O) implicam na tentativa de redução do código do programa e do seu tempo de execução, sem efetuar nenhuma otimização que cause um aumento significativo no tempo de compilação do programa. O nível 2 (-O2) aplica a maioria das otimizações disponíveis, inclusive as do nível 1. Neste nível, há um aumento no tempo de compilação, e também no desempenho do programa, se comparado com o nível 1, e realiza otimizações que tentam evitar que o processador perca ciclos de *clock* esperando por operações ociosas. Já o nível 3 (-O3) habilita todas as otimizações do nível 2 (e, consequentemente, também as do nível 1). Nota-se um aumento no tamanho do arquivo binário resultante, se aplicado esse nível de otimização.

BLAS - Basic Linear Algebra Subprograms

Como exemplo de uma biblioteca numérica que utiliza os recursos mencionados nos itens anteriores pode-se citar a biblioteca BLAS.

A BLAS é uma biblioteca que reúne sub-rotinas de operações básicas da álgebra linear. Seu principal intuito é a obtenção do resultado no menor tempo possível, sendo assim uma biblioteca voltada para o alto desempenho. No momento da instalação da BLAS, é identificado qual o processador da máquina e, conseqüentemente, quais as instruções ao nível de processador estão disponíveis. Com esta propriedade, a BLAS possibilita um considerável aumento no desempenho dos programas que a utilizam.

Conclusões

Através de alguns resultados já obtidos pode-se observar a importância da correta escolha de ferramentas e técnicas de programas que podem ser utilizados na resolução de problemas de computação científica. Deve-se considerar o fato de que essas ferramentas podem ter abordagens diferentes, ou seja, podem visar o ganho do desempenho ou o ganho da exatidão. Por isso é fundamental que o programador/usuário saiba avaliar as reais necessidades de seu problema possibilitando, com isso, a correta escolha da ferramenta mais adequada para a sua resolução.

Referências

[CON 03] CONTESSA, Diego Fraga. **Uso Eficiente da Biblioteca de Rotinas Básicas de Álgebra Linear no Ambiente de Programação Paralela por Troca de Mensagens em Cluster**. Porto Alegre, Universidade Federal do Rio Grande do Sul. 2003.

[DEN 04] DENARDI, Rúbia Medianeira. **Uso da BLAS em Ambiente de Cluster.** Trabalho Individual. PPGC da UFRGS. Porto Alegre. 2004.

[KAR 03] KARNIADAKIS, G.; KIRBY II, R. **Parallel Scientific Computing in C++ and MPI: A Seamless Approach to Parallel Algorithms and Their Implementation.** 2th.ed. Cambridge: Cambridge University Press, 2003.

[GNU 04] Using the GNU Compiler Collection. Programming Languages Supported by GCC. Disponível em http://gcc.gnu.org/onlinedocs/gcc-4.0.1/gcc/G_002b_002b-and-GCC.html. Agosto. 2005.

[WAL 04] WALL, K. Using the GNU Compiler Collection. Disponível em http://www.samspublishing.com/articles/article.asp?p=130865&seqNum=5&rl=1. Agosto. 2005.

[PAR 05] **Paralelismo ao Nível da Instrução.** Disponível em: http://gec.di.uminho.pt/lesi/ac20405/ApontamentosILPv3.pdf. Setembro 2005

[TEC 05] **Tecnologia Pessoal Intel - Glossário**. Disponível em: http://www.intel.com/portugues/personal/resources/glossary/body.htm#int_str_sim. Julho 2005

[PEN 05] **Visão Geral Intel Pentium 4**. Disponível em: http://www.intel.co.jp/support/pt/processors/pentium4/sb/CS-007989.htm. Junho 2005

[INT 05] **Site oficial Intel**. Disponível em: http://www.intel.com. Agosto 2005