Otimização de Código usando o compilador gcc/g++ - um estudo sobre as diretivas de otimização

Gabriel Rech, Dimas Gabriel, Prof. Dr. Carlos Amaral Hölbig

Universidade de Passo Fundo
CAMPUS I - Km 171 - BR 285, Bairro São José
Caixa Postal 611 - CEP 99001-970 – Passo Fundo/RS PABX (54) 3316-8100
75973@lci.upf.br, digabriel@gmail.com, holbig@upf.br

Introdução

A perda de desempenho na execução de programas pode ser gerada por vários fatores como, por exemplo, a escolha inadequada dos métodos numéricos que serão utilizados para a resolução deste problema. Além disso, é de fundamental importância que o programador leve em consideração aspectos relacionados a otimização do código que está sendo implementado. Este fator pode ser abordado tanto na elaboração do programa como no uso de técnicas e/ou ferramentas que propiciem que os programas implementados sejam verificados e otimizados com o objetivo de melhorar seu desempenho. No caso do uso de diretivas de otimização dos compiladores, os programas poderão ser modificados (otimizados) ainda em tempo de compilação. A crescente necessidade de programas mais eficientes demonstra a importância de pesquisas que objetivem códigos mais otimizados [AHO 95]. Esta pesquisa tem como objetivo fazer um estudo sobre as diretivas de otimização do compilador gcc/g++, com foco nas linguagens C e C++.

Metodologia

A realização desta pesquisa está focada no compilador gcc/g++ (compilador free para as linguagens C e C++) para sistemas operacionais baseados em Linux. Sobre este compilador estão sendo estudadas as diretivas de otimização de código O0, O1, O2 e O3 (conjunto de rotinas de otimização) [STA 05]. Com base nas características específicas de cada diretiva será analisado que tipo de otimização que essa diretiva realiza e sobre qual tipo de instrução no programa é mais adequado o seu uso. Para tanto será realizada uma série de testes onde serão aplicados cada um destes conjuntos de rotinas de otimização e serão verificados o resultado do desempenho obtidos com a sua execução. Através deste estudo será possível desenvolver programas e compilá-los de forma mais eficiente do ponto de vista de desempenho computacional [STA 04].

Resultados

Inicialmente a pesquisa está trabalhando em duas frentes: a primeira refere-se a implementação de alguns métodos numéricos para a resolução de sistemas lineares (Eliminação de Gauss, Gauss-Seidel e Gauss-Jacobi) onde pode-se observar, de forma geral, que os programas compilados sem o uso de diretivas de otimização obtiveram um tempo de execução em torno de 100% superior aos programas compilados com o uso das diretivas de otimização. Já a segunda frente do trabalho está voltada para o uso de ferramentas de análise de desempenho. Estas ferramentas analisam o código do programa que foi implementado e, através de uma série de testes, indicam quais as opções de otimização que são mais e menos adequadas para serem utilizadas no programa que se está testando. Um exemplo deste tipo de ferramenta é a Acovea (http://www.coyotegulch.com/products/acovea/index.html) que possibilita a análise das opções de compilação através do uso de algoritmos baseados em técnicas da Inteligência Artificial. Estas ferramentas possibilitam uma grande ajuda ao programador pois podem indicar de uma forma mais eficiente quais as diretivas de otimização que poderão ser utilizadas ou pelo menos testadas nos programas.

- Gráfico gerado a partir de testes de otimização com o método de Gauss-Seidel;
- Implementação utilizando um matriz de ordem 1024;
- Tempo de execução do programa gerado sem nenhuma otimização(O0);
- Tempo de execução do programa gerado utilizando os conjuntos de diretivas de otimização O1, O2, O3, Os.

Conclusões

A elaboração de programas que visem obter um melhor desempenho computacional deve ser levado em consideração no momento da implementação de sistemas computacionais, principalmente os relacionados a aplicações de grande porte. Para tanto o estudo e utilização de técnicas de otimização podem contribuir de maneira significativa para essa melhora.

Referências

[STA 05] STALLMAN, Richard M. **Using the GNU Compiler Collection.** Boston: GNU Press, 2005.

[STA 04] STALLINGS, William. **Arquitetura e organização de computadores: projeto para o desempenho. 5.ed.** São Paulo: Pearson Education, 2004.

[AHO 95] AHO, Alfred V.. **Compiladores: princípios, técnicas e ferramentas.** Rio de Janeiro: Guanabara Koogan, 1995.