

技術報告：最適化コンパイラ

Experimental/Enhanced GNU C Compiler System and Pentium GCC

濱崎雅弘 (知的システムデザイン研究室)

Masahiro HAMASAKI (Intelligent System Design Laboratory)

谷村勇輔 Yusuke TANIMURA

Abstract This document is report of EGCS and PGCC. EGCS stands for Experimental/Enhanced Gnu Compiler System. It is a descendant of gcc which attempts to improve the all-round performance of the compiler. PGCC stands for Pentium GCC. The PGCC is a version of the GNU C compiler (gcc), which provides special optimizations for the Pentium processor.

1 はじめに

最適化コンパイラによって生成されたプログラムは、通常使用しているコンパイラで作るものよりも高速になる。UNIX上のコンパイラとして広く普及している gcc にも最適化オプションや、より最適化された gcc がある。今回は、それらの最適化コンパイラの性能を通常のコンパイラと比較し、その性能を検証する。

2 最適化コンパイラ

コンパイラにおける最適化には2種類ある。一つはソース言語から解析部によって解析された中間言語を論理的に最適化する手法。もう一つが、その中間言語をハードウェアに適した形式に変換することである。今回紹介する egcs 及び pgcc は後者の最適化を強化した gcc と言える。

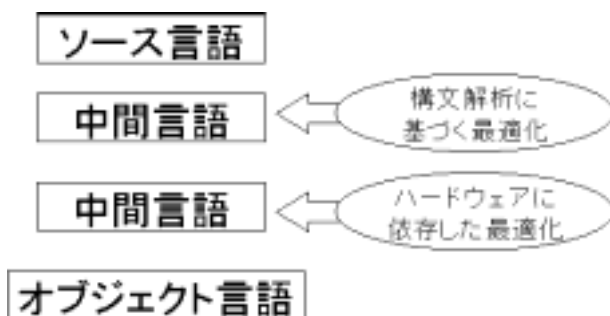


図 1: コンパイラにおける最適化

2.1 egcs

egcs(エッグス)は Experimental/Enhanced Gnu Compiler System の略で、米 Cygnus Solutions 社¹によって開発されたコンパイラである。gcc-2.8.x をベースに数値演算関係の最適化能力が強化され、様々なプロセッサ・アーキテクチャに対する最適化強化及び bug fix も積極的に行なわれ、i586, i686 系や PowerPC750 の最適化も追加されている。2% - 30% の速度向上が得られ、実際に gzip-1.24 では展開速度が 20% ~ 35% 向上したとの報告もある。

EGCS の開発チームは今年の5月に GCC のメンテナンスを引き継ぐことが正式に決まり、GCC Steering Committee と名称を変更した。これにより EGCS の次のバージョンは gcc-2.95 としてメジャーリリースすることになった。²その結果 gcc は "GNU C Compiler" ではなく "GNU C Collection" の略称となる。

2.2 pgcc

pgcc は Pentium gcc の略で egcs にパッチされたものである。対象を Pentium Processor やその互換チップに絞ってさらなる最適化が施されている。

そもそも、Pentium チップのハードウェア設計は、一般的な最適化方法ではなく、使用頻度の高い命令やその組合せに対して特に最適化されているという特徴を持つ。そのため、ソフトウェアをこの設計向きに最適化するのはかなり複雑で、多くの例外がある。しかし、同時に相当な性能向上が可能であるとも考えられる。

¹<http://www.cygnus.com/>

²<http://egcs.cygnus.com/gcc-2.95/gcc-2.95.html> にて入手可能

3 性能比較

最適化コンパイラを用いることによって、実際にどのくらい性能が向上するかを検証してみる。利用したコンパイラのバージョンは以下の通りである。

gcc version 2.7.2.3

gcc version egcs-2.91.60 Debian 2.1 (egcs-1.1.1 release)

gcc version pgcc-2.91.66 19990314 (egcs-1.1.2 release)

gcc と egcc には最適化オプションが 3 段階あり、今回は最適化を行わないものと、2, 3 段階目を用いた。2 段階目は生成後の関数のインライン化等のファイルサイズを大きくするものを除いた最適化で、3 段階目はそれらを含む最適化である。pgcc には 6 段階の最適化オプションと、さらに intel 互換の中でのアーキテクチャに対応した最適化がある。今回利用したハード (PentiumII) にあわせて、6 段階目のオプションとさらに PentiumII 用の最適化を加えたものを用いた。

3.1 アセンブラレベルでの比較

0 から 1000 まで足し算をする、という簡単な短いプログラムを用いて、アセンブラコードファイルを出力させてみた。それぞれの比較結果を下に示す。jump 命令はプログラムの速度低下の一因であり、その数は最適化の目安となる。また、file size はアセンブラコードファイルのファイルサイズで、小さいほどプログラムとしてスリム化 (つまり最適化) されていると言える。

表 1: コンパイラによるアセンブラコードの違い

	gcc	gcc -O3	egcc -O3	pgcc -O6
jump 命令	3 回	1 回	1 回	1 回
file size	429byte	283byte	323byte	241byte

表から、egcc および pgcc が最適化を行っていることが容易にわかる。また、jump 命令の数とファイルサイズの関係から、pgcc が egcc と同様の最適化 + ペンティアム用の最適化を行っているということが推測できる。

3.2 実行速度での比較

次に、実際にプログラムを動かして時間を計測してみる。CPU は PentiumII である。プログラムは、ループを使った単純な整数計算の繰り返しである Program A、グレイコードと整数の変換を行う Program B、そしてマルコフ連鎖の計算を Program C とする。

グラフは gcc の最適化オプション無しでコンパイルした実行ファイルの実行速度を 1 として、その速度比で表す。ProgramC の gcc -O3 のグラフが無いのは、コン

パイルしたプログラムがうまく動作しなかったからである。

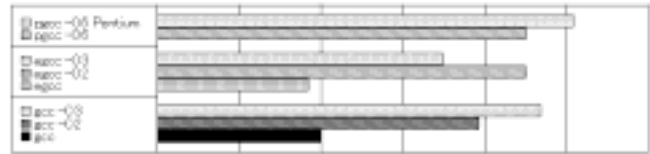


図 2: ProgramA:整数演算

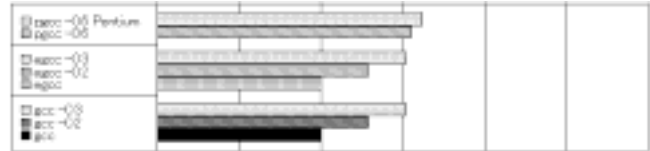


図 3: ProgramB:グレイコード

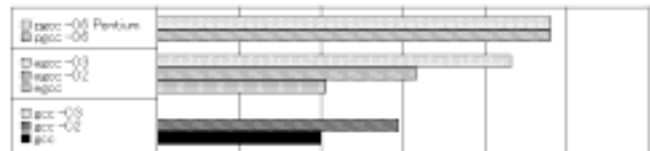


図 4: ProgramC:マルコフ連鎖

グラフから分かるように、pgcc は全てにおいて優れた成績を収めている。とくに浮動小数点の演算が多くあるマルコフ連鎖については顕著である。

4 おわりに

以上の結果より、pgcc は Pentium 上での最適化コンパイラとして非常に優秀な性能を持っているということと、どの環境においても最適化を行うことが効果的であることがわかった。最適化には幾つかの問題もあり、その程度は問題視しなければならないが、少なくともコンパイラが推奨しているレベルでの最適化オプション (gcc, egcc でいう -O2、pgcc の -O6) は、プログラムを作成する際には必須であると言える。

参考文献

- [1] 遠藤俊徳 『GNU C Compiler』 (秀和システム, 1998)
- [2] Pentium Compiler Group 『PGCC』 (http://www.goof.com/pgc/, 1999)