

線形モデルによる
アプリケーション性能予測の一手法

5Y-5

境 隆二, 森本 展行, 遠藤 浩太郎

(株) 東芝 情報・通信システム技術研究所

1. はじめに

演算中心のアプリケーション性能を予測するには、公表されたベンチマーク値の比率を計算して、現行マシンでの性能にこの比率を乗じるのが最も簡単である。しかし、この方法ではプロセッサの種類やアプリケーション特性のばらつきのために、十分な精度で予測出来ない。本稿では、標準ベンチマークの公表値の線形式によって、アプリケーションの性能予測を行う方法について述べる。

2. 計算機性能とアプリケーション性能

計算機性能には、クロックスピード、パイプライン構成、並列度、キャッシュ方式(H/W)や、コンパイラの最適化、OSの種類(S/W)などの様々な要素が関係している。従って、特定のアプリケーション性能が、どの計算機で一番速くなるかを判定するのは難しい。図1に、SPECfp92に含まれるそれぞれのプログラムの性能の分布を示す。図1の縦軸は計算機の種類であり、横軸はコンパイラの最適化レベルである。各チャートの方位はSPECfp92の各プログラムに対応し、値は全プログラムの性能の平均値(SPEC値)を1としたときの相対値である。それぞれのチャートの形が異なるということは、単純にSPEC値のみで性能を正確に予測出来ないことを意味している。

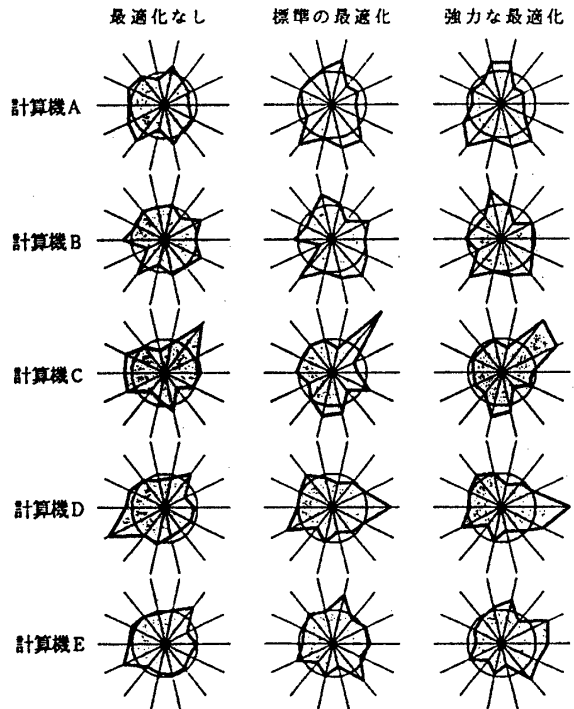


図1 SPECfp92の各プログラムの性能分布

3. 線形モデル

計算機とその計算機でサポートされているコンパイラの最適化の組をプラットフォームと呼び、その集合を $\{P_k\}$ で表す。対象とするアプリケーションの集合を $\{A_i\}$ とし、 P_k における A_i の実行時間を $t_k(A_i)$ で表し、 $\{A_i\} \rightarrow R^M$ への写像

$$T: A_i \rightarrow (t_1(A_i), t_2(A_i), \dots, t_M(A_i)) \quad (3.1)$$

を考える。 A_1+A_2 は A_1 と A_2 を逐次実行するアプリケーション、 $a \cdot A_1$ は A_1 を a 回実行するアプリケーションとすると、写像 T には

$$T(A_1+A_2) \cong T(A_1) + T(A_2) \quad (3.2)$$

$$T(a \cdot A_1) \cong a \cdot T(A_1) \quad (3.3)$$

という構造がある。したがって写像 T の像 $\{T(A_i)\}$ は、ある線形部分空間に近い形をしている。 $\{T(A_i)\}$ をアプリケーション性能の線形モデルと呼び、 $T(A_i)$ を A_i の特性ベクトルと呼ぶことにする。

4. 予測方法

線形モデルにもとづいて、アプリケーション性能を予測する方法を考える。 R^+ を 0 より大きい実数の集合とすると $T(A_i) \in R^+M$ である。 R^+M の距離 d を $d(x, y) = \max_k \{|\log x_k - \log y_k|\}$ により定め

る。距離として対数をとるのは、予測誤差を実行時間の比で評価するためで、予測時間を t' 、実測時間を t としたときに、予測誤差の最大値が $\exp(d(t', t))$ となるからである。予測対象となるアプリケーションの集合 $\{A_i\}$ に対して、線形部分空間 H が誤差 r で $\{T(A_i)\}$ に近いとは、 $R = \log(1+r)$ としたとき、

$$\forall A_i, T(A_i) \in U_R(H) = \{y \in R^+M \mid \exists x \in H^+, d(y, x) < R\} \quad (4.1)$$

という条件を満たすものである (図 2)。ただし、 $H^+ = H \cap R^+M$ である。いま、 N 個のベンチマークプログラム $B_1 \sim B_N$ について、 $T(B_1) \sim T(B_N)$ が張る線形部分空間 H が、許容される予測誤差 r に対して条件 (4.1) を満たすとき、アプリケーション A_i の特性ベクトルは次のように書ける。

$$T(A_i) \approx \sum_j a_j(j) * T(B_j) \quad (4.2)$$

この a_j を求めることによって $t_k(A_i)$ を予測しようというのが基本的なアイデアである。実際には、 $T(B_j)$ には測定誤差があり、この誤差によって H の次元が増大し、 $a_j(j)$ の絶対値が非常に大きくなる可能性があるため、 a_j の決定に際してはこの問題を考慮する必要がある。

我々は、ベンチマークとして SPECfp92 を使って、この予測方法の評価と、SPECfp92 が線形モデルの中でどのような部分空間を埋めるのか等の評価を行った。評価手順は以下の通りである。

それぞれ異なるプロセッサを搭載した数種類の計算機を用意し、SPECfp92 のベンチマークプログラムをそれぞれ、最適化なし、最適化オプション、より高度な最適化オプションでコンパイルし実行時間を求める。次に、予測したいアプリケーションを、同じようにそれぞれの計算機で各オプションについて実行時間を求め、各プラットフォームにおける (4.2) による予測誤差が最小になるように a_j を決定する。この a_j を用いて、目的の計算機での性能を線形式 (4.2) で予測する。また、予測に使う SPECfp92 のベンチマークプログラムを適当に選んだときの予測誤差も評価する (モデル化の方法が少し異なるが、線形モデルを用いたコンパイラの最適化の評価に関しては [1] がある)。

5. 評価結果

図 3 に、線形モデルで予測した場合と、SPEC 値の比を単純に乗ずる従来の方法で予測した場合の予測誤差を示す。この図から、従来の方法に比べて本予測方法の方が、予測精度が優れていることがわかる。しかし、アプリケーションによっては、十分な予測精度が得られていない。このことは SPECfp92 のベンチマークセットが張る部分空間が、条件 (4.1) を満たしていないことを示している。図 4 は、予測に使う SPECfp92 のプログラムをいくつか選んで、特性の良いものから順番に増やしていった場合の予測誤差を示したグラフである。このグラフの縦軸は予測誤差、横軸は選んだベンチマークプログラムの個数である。このグラフから、SPECfp92 の 14 個のプログラムのうち 8 個のプログラムである程度の精度で予測ができることがわかる。逆に、我々が評価した環境では、その他のプログラムは余分 (線形従属) であることがわかる。

6. まとめ

公表されたベンチマーク性能値から、アプリケーションやベンチマークプログラムの詳細な内部処理を調べずに、高い精度でアプリケーション性能を予測する方法を示し、その効果を確認した。また、ベンチマークセットが満たすべき特性を定式化し、この観点から SPECfp92 の評価を行った。今後は、SPEC95 についても同様の評価を行う予定である。

参考文献 [1] R.H.Saavedra, A.J.Smith, "Performance Characterization of Optimizing Compilers," IEEE Trans. Software Engineering, vol. 21, no. 7, July. 1995

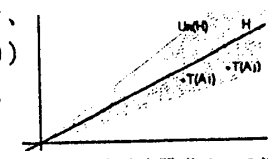


図 2 線形部分空間 H とその近傍

	係数 a_j	予測誤差	
		線形予測	従来方法
77*アプリケーション1	(2.9, 1.7, 1.0, -8.6, -1.8, -0.8, -1.1, -0.2)	26.0%	70.0%
77*アプリケーション2	(0.3, 1.5, -1.1, -1.1, -0.2, -0.5, -1.0, -0.1)	51.5%	65.5%
77*アプリケーション3	(0.1, 0.1, 0.3, 0.9, 0.2, -0.1, -0.4, -0.1)	7.9%	24.0%

図 3 予測誤差率

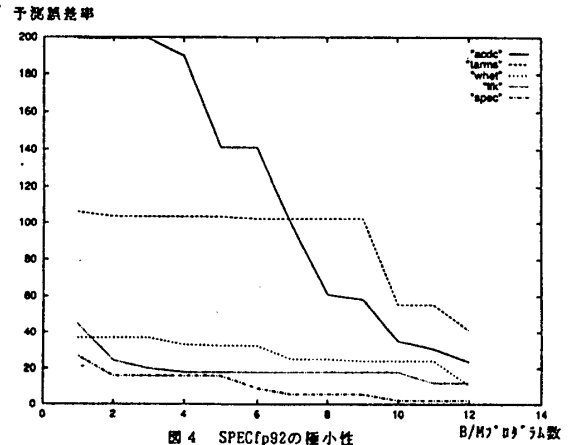


図 4 SPECfp92 の種小性