

メモリ使用量最小化を目的とした コンパイラの最適化オプションセットの探索

Automatic Selection of Compiler Optimization Option Set for Memory Usage Minimization

大城亮
Ryo Ooki

石浦菜岐佐
Nagisa Ishiura

関西学院大学理工学部
School of Science and Technology, Kwansai Gakuin University

1 はじめに

コンパイラには多くの最適化処理が実装されているが、与えられたプログラムに適した最適化処理の組み合わせを手動で調節することは困難なため、その探索を自動的に行う研究が行われている。最も多く研究されているのは実行時間の短縮を目的とした探索 [1] だが、組み込みシステムではコードサイズの最小化も重要になる。文献 [2] ではアセンブリ命令数の最小化する探索を行っているが、実際のメモリ使用量は考慮していない。

本研究では、メモリ使用量最小化を目的としたコンパイラの最適化オプションセットの自動探索手法を提案する。探索の評価値は、コードサイズだけでなく静的データ領域やスタックサイズも考慮したメモリ使用量を使用し、また探索アルゴリズムには適応温度並列シミュレーション (以下 PSA/AT) [3] を利用する。

2 コンパイラの最適化オプションセット

多くのコンパイラでは、各最適化処理の ON/OFF をオプションによって制御可能であり、対象プログラムに適切な最適化オプションセット (最適化オプションの組み合わせ) を指定すれば、生成コードの性能を向上できる。しかし、例えば GCC 4.5.1 には 172 個の最適化オプションがあり、 2^{172} 通りの組み合わせを手動で調整することは困難である。そこで、メタヒューリスティックを利用した探索が研究されている [1]。

3 メモリ使用量最小化を目的とした探索手法

本手法では、メモリ使用量の最小化を探索目標とする。メモリ使用量は ROM と RAM の合計値とする。ROM には text, data, 及び rodata セクションを、RAM には data, bss セクションとスタックを配置するものとする。各セクションサイズは生成コードのヘッダから取得し、スタックサイズはアセンブリ命令列の解析により算出する。

探索アルゴリズムには、PSA/AT [3] を用いる。本手法では、最適化オプションセットの一つを PSA/AT の一状態に対応付け、あるオプションの切り替えを状態遷移に対応付ける。状態遷移の受理/非受理は、状態に対応するコンパイル結果のメモリ使用量を用いたメトロポリス基準に基づいて決定する。

4 実験結果

ARM 用 GCC 4.5.1 に対して、探索プログラムを Perl で実装し、評価実験を行った。実行環境は、CPU Intel Core i7 2.9GHz、メモリ 4GB、OS Ubuntu10.10、並列実行数

8、探索周期 30 である。表 1 は、各プログラムについて、GCC の `-O0` (最適化無し) と `-Os` (コードサイズ優先の最適化) オプション、及び本手法で求めたオプションセットによる各セクション及びメモリ使用量をまとめたものである。「cpu」は探索に要した CPU 時間である。本手法は、`-Os` オプションと比較してメモリ使用量を平均約 7.2% 削減している。

表 1 実験結果

program	option	サイズ (byte)							合計	cpu(s)
		text	data	bss	stack	ROM	RAM			
bzip2	-O0	26384	32	60	80	26416	172	26588	-	
	-Os	18833	32	60	132	18865	224	19089	-	
	本手法	18549	32	60	104	18581	196	18777	1242.1	
parser	-O0	7680	0	0	40	7680	40	7720	-	
	-Os	4908	0	0	68	4908	68	4976	-	
	本手法	4720	0	16	68	4720	84	4804	3523.2	
enhex	-O0	1144	68	0	20	1212	88	1300	-	
	-Os	841	68	0	32	909	100	1009	-	
	本手法	885	0	0	32	885	32	917	174.8	
jikespg	-O0	402568	2308	2116	144	404876	4568	409444	-	
	-Os	223138	1388	2148	280	224526	3816	228342	-	
	本手法	209696	1276	3232	292	210972	4800	215772	11906.2	
vfprintf	-O0	9440	0	0	60	9440	60	9500	-	
	-Os	6864	0	0	36	6864	36	6900	-	
	本手法	5996	0	0	52	5996	52	6048	1543.0	
sha	-O0	2492	0	0	16	2492	16	2508	-	
	-Os	1212	0	0	56	1212	56	1268	-	
	本手法	1104	0	0	52	1104	52	1156	325.2	
rijndael (AES)	-O0	43712	8	12	28	43720	48	43768	-	
	-Os	33346	8	12	84	33354	104	33458	-	
	本手法	32438	8	12	72	32446	92	32538	1580.9	
blowfish	-O0	10616	4172	0	48	14788	4220	19008	-	
	-Os	9768	4	0	40	9772	44	9816	-	
	本手法	2480	0	0	16	2480	16	2496	425.6	

5 むすび

本稿では、メモリ使用量を最小化する最適化オプションセットの自動探索手法を提案した。今後の課題としては、ARM 以外のアーキテクチャへの適用が挙げられる。

参考文献

- [1] S. C. Lin, et.al: "Automatic selection of GCC optimization options using a gene weighted genetic algorithm," in *Proc. ACSAC*, pp. 1-8 (Aug. 2008).
- [2] M. Haneda, et.al: "Code size reduction by compiler tuning," in *Proc. SAMOS*, pp. 186-195 (July 2006).
- [3] 三木光範他: "適応的温度調節機能を持つ温度並列シミュレーション," 65 回情報処大会, 5Y-6 (Mar. 2003).