

2パス限定投機方式における動的最適化適用可能性の検討

津田 翼[†] 大津 金光[†] 横田 隆史[†] 馬場 敬信[†]

[†]宇都宮大学大学院工学研究科情報システム科学専攻

1 はじめに

プログラムの高速化には、プログラム中で頻繁に実行されるループの高速化が重要である。ループ中の実行経路 (パス) も 2 本のパスがほとんどの実行割合を占めており、この 2 本のパスに対して投機的マルチスレッド実行をすることでループの高速化が期待できる。そこで、本研究室では 2 パス限定投機方式の提案をしている [1]。2 パス限定投機方式では、ループ中で最も実行頻度の高いパスと 2 番目に実行頻度の高いパス (以降、#1、#2 パス) に特化した投機コードを生成し、2 つの投機コードのうち、どちらが実行されるか予測しながら投機的にマルチスレッド実行を行う。

また、プログラム実行時にプロファイル情報を取得し最適化を行う動的最適化が注目されている。プログラム実行中において集中して実行される部分とその挙動を把握できれば、プログラムの挙動を反映した最適化が可能となる。特に、頻繁に実行されるパスが高々 2 本であり、それが入れ替わっているならば 2 パス限定投機方式に動的最適化を適用することで挙動に沿った最適化ができると考えられる。

本稿では、ループ中のパスに着目してプログラム実行時の挙動の変化を把握し動的最適化に反映することを目的として、2 パス限定投機方式において動的最適化が有効となる条件を明らかにする。そして、その条件に合致するループがどの程度存在するかを調べ、動的最適化への適用可能性を検討する。

2 2パス限定投機における動的最適化

従来研究 [1] では、プログラム実行全体の頻度から #1、#2 パスを特定しているため、プログラム実行時のパスの振る舞いを反映しているわけではない。プログラム実行全体の頻度が小さいパスでも、プログラム実行時では集中して実行されているケースもある。プログラム実行時において、#1、#2 パスとして実行されるパスが入れ替わっている挙動を示すループならば、2 パス限定投機方式に動的最適化を適用することでプログラムの挙動に即した最適化が可能になると考えられる。

動的最適化では、プロファイル→最適化→実行のサイクルをプログラム実行時に行うことによって、プログラムの高速化を目指す。プロファイルにより、ループ中の #1、#2 パスを特定し、それらのプロファイル情報を基にそれぞれのパスに沿った形でプログラムコー

ドを最適化する。実行では、2 パス限定投機方式を実現する PALS[2] による実行を想定する。

3 実行時におけるパス挙動の解析

3.1 解析方法

パスプロファイラ [3] により、ループ中のパスを一定本数検出するごとに、ループとそのループ中のパスの実行頻度を算出する。これにより、実行時においてループ中での実行パスの挙動変化を調べ、#1、#2 パスとして実行されるパスの違いについて着目する。

調査対象には、SPEC CINT2000 より 164.gzip, 175.vpr-place, 175.vpr-route, 176.gcc, 181.mcf, 197.parser, 254.gap, 255.vortex, 256.bzzip2 の 9 つのプログラムを用いる。最適化オプションは-O3 を適用し、データセットは train を使用する。

3.2 解析結果

一例として、181.mcf の関数 primal_bea_mpp 内に含まれるループの解析結果について述べる。参考として、このループの全て実行を終えたときの実行頻度を表 1 に示す。表中のパス ID とは、ループ内のパスを検出した順に割り振られる通し番号で、パスの種類の識別に用いる。

表 1: 全実行終了時のパス実行頻度

順位 (実行頻度 (%))	パス ID
#1(58.024)	P31
#2(26.580)	P33
#3(12.707)	P32
#4(1.390)	P75
#5(1.160)	P59
#6(0.139)	P76

プログラム実行中におけるパスの実行頻度の推移を図 1 に示す。181.mcf 全体で検出したパスの延べ総本数は 402,830,463 本で、図 1(a)~(d) はプログラム全体の中で解析したループが実行されている部分の実行頻度の推移を表している。横軸はパスの延べ検出総本数で、縦軸は実行頻度を表す。図 1(a) では、主に P31 と P32 が #1、#2 パスとして実行されている。図 1(b) や図 1(c) では、#1 パスに P31、#2 パスに P32 や P33 が現れている。図 1(d) では、#1 パスが P33、#2 パスが P31 で実行されている。図 1 より、実行の局面によって #1、#2 パスが違うパスに入れ替わっていることが分かる。このようなループであれば動的最適化が有効であると考えられる。

3.3 動的最適化が有効となり得る条件

図 1 から、動的最適化が有効となり得るループは、複数本 (3 本以上) パスが存在する中で、実行時でも #1、#2 パスが違うパスと入れ替わるループである。また、#1、#2 パスの両方で支配的な実行頻度を占め

A Consideration of Applicability of Dynamic Optimization in Two-Path Limited Speculation Method

[†]Tsubasa Tsuda, Kanemitsu Ootsu, Takashi Yokota and Takanobu Baba

Department of Information Systems Science, Graduate School of Engineering, Utsunomiya University (†)

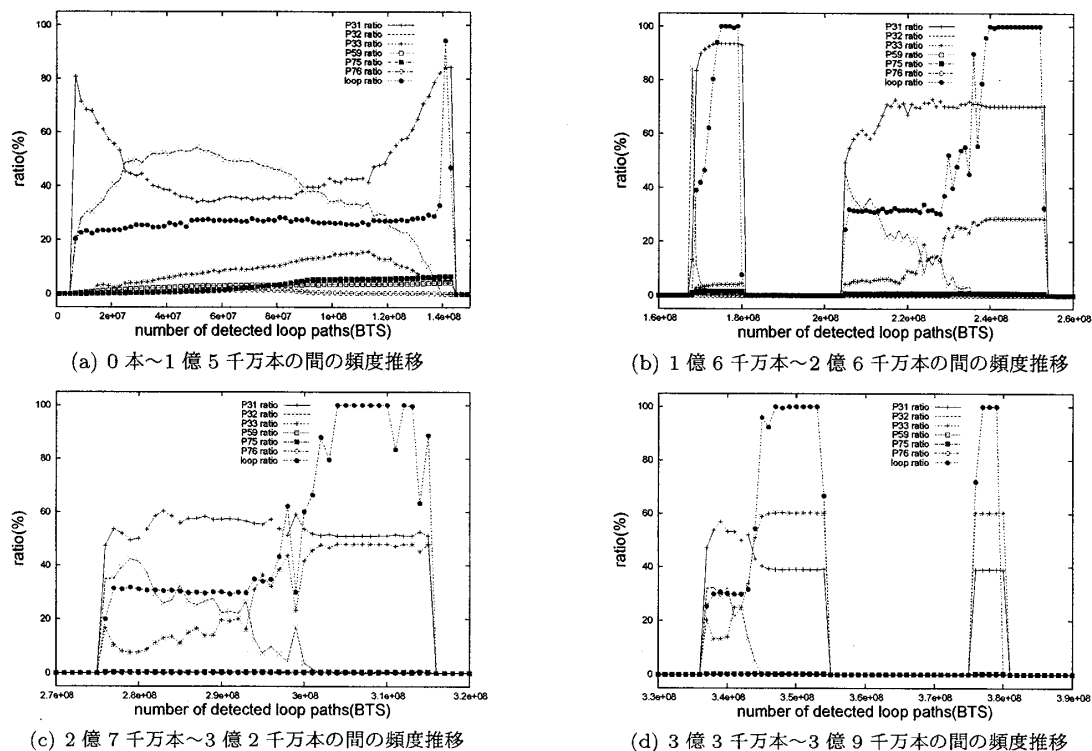


図 1: パスの実行頻度の推移

ていれば、2パス限定投機方式により速度向上が期待できると考えられる。2レベルパス予測手法 [1] によるパス予測の容易性も考慮し、#1、#2パスが違うパスと入れ替わった後もしばらくの間はその順位のまま実行が続いている必要がある。

以上の点についてプログラム実行中に着目し、調査対象のプログラムについて、全ループの種類数、1、2本のみしかパスが実行されないループの種類数、3本以上パスが実行されるループの種類数、動的最適化が有効となり得る条件に当てはまるループの種類数について調査した結果を表 2 に示す。パスを 100 本検出するごとに実行頻度を算出し、そのとき求めた #1、#2パスが 10^5 本分続いたとき、実行時における #1、#2パスとしてカウントするようにした。

表 2: 動的最適化が有効となり得るループの個数

プログラム名	全ループ	1,2本	3本以上	条件適合
164.gzip	100	86	14	11
175.vpr-place	127	108	19	8
175.vpr-route	235	202	33	7
176.gcc	1097	887	210	90
181.mcf	69	59	10	8
197.parser	467	374	93	42
254.gap	404	360	44	35
255.vortex	154	140	14	7
256.bzip2	133	109	24	14
total	2786	2325	461	222

どのプログラムも 1 本、もしくは 2 本しかパスが実行されないループが多数存在している。高々 2 本程度のパスしか実行されないループでは、2パス限定投機方式で対応可能であるので、動的最適化を行う必要はない。3 本以上のパスが実行されるループの数に対し

て、条件に適合したループの数はプログラムによってばらつきはあるもののトータルで約半数であった。3 本以上パスが実行される複雑な構造を持つループについては、動的最適化の適用によって速度向上が期待できる可能性が十分にあると言える。

4 おわりに

本稿では、2パス限定投機方式での実行において動的最適化が有効となり得るループの条件を示し、その適用可能性について検討した。1、2本しか実行されないループは動的最適化の必要はないが、3本以上パスが実行されるループに対しては、動的最適化を適用することで高速化の可能性があると考えられる。今後の予定としては、動的最適化の適用による速度向上率を評価することが挙げられる。

謝辞

本研究は、一部日本学術振興会科学研究費補助金 (基盤研究 (C)20500047, 同 (C)21500049, 同 (C)21500050) および宇都宮大学重点推進研究プロジェクトの援助による。

参考文献

- [1] 横田隆史ほか: “2パス限定投機方式の提案”, 情報処理学会論文誌: コンピューティングシステム, Vol.46, No.SIG16(ACS-12), pp.1-13, 2005.
- [2] 十鳥弘泰ほか: “2パス限定投機方式を実現するマルチコアプロセッサ PALS の提案”, 信学技報, Vol.109, No.319(CPSY2009-46), pp.19-24, 2009.
- [3] 矢野目秀人ほか: “パスに含まれる命令数を考慮したハードウェアホットループパス検出機構”, 情報処理学会第 70 回全国大会論文講演集, pp.1-137~pp.1-138, 2008.