メモリ階層性能シミュレータを用いたCPU単体性能チューニング

佐藤 幸紀、佐藤 真平 (東京工業大学/ JST CREST)

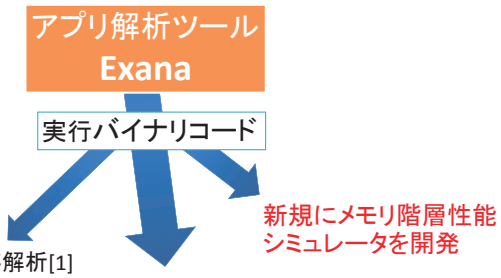
CPU単体性能のチューニング

- 大規模並列処理をする上でも実行効率を決める鍵
- キャッシュを意識することが重要

大量かつ複雑なメモリアクセスを生じるHPCアプリから深化するメモリ階層向けにチューニングを支援するツールが必要

ツールの支援によりメモリアクセスに関する最適化の実施
(パディング、データ構造変換、キャッシュブロッキング、プリフェッチ等)

将来的には自動化・半自動化を目指す



(NICAM-DCのプロファイル結果)

メモリ階層性能シミュレータ

HW性能カウンタによる解析の問題点

- プログラム全体のキャッシュの統計値ではチューニングするべき箇所が特定できない
- 一般的なキャッシュシミュレータも同様

例) HimenoBMT Dynamic版Orig. シミュレーション結果
Cache miss counts and rates:

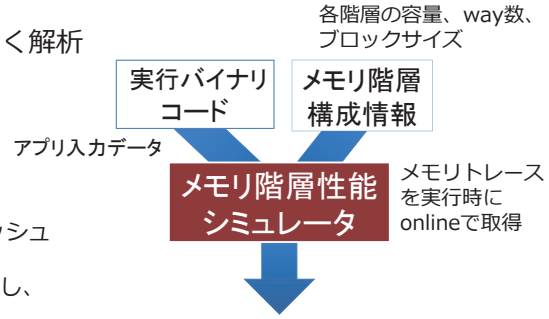
L1	3184561124	37.52 % of Accesses
L2	225119549	7.07 % of Accesses
L3	179983240	79.95 % of Accesses

今回開発したメモリ階層性能シミュレータの特徴

- HPCアプリをソースコードやMakefileを修正することなく解析
- 各階層のCacheのミス率を命令毎に出力
- **競合ミス**の回数を取得
  - 競合ミスは性能に大きな影響を与える
  - メモリレイアウトやアクセスパタンに由来

競合ミスの判定方法

- セット毎に32エントリのFIFOを用意しアクセスしたキャッシュのタグの履歴を保持。FIFOに記録されるタグは重複を許容
- LRUポリシにてリプレースされるものとFIFOのタグを比較し、一致した場合、競合ミスと判定



評価と検証

ベンチマーク HimenoBMT(C言語, Dynamic版)
S データセット、400反復
キャッシュパラメータ (Intel Xeon SB世代をシミュレーション)
L1=32kB,8way L2=256kB,8way L3=10MB,20way
block size=64、ライトスルー方式、厳密なLRU、Inclusive Cache
現在、マルチスレッド、プリフェッチ、レーテンシの見積りは未実装

(1.1) Orig版(Paddingなし) プログラム全体のキャッシュの挙動

Cache miss: L1=37.52 %、L2=7.07 %、L3=79.95 %
Conflict miss: L1=92.86 %、L2=11.20 %、L3=3.52 %

L1のキャッシュミスが非常に多く原因が競合ミスと分かる
どの部分をチューニングすればよいかは不明
↓
命令毎にデータを取得し、メモリアクセスが多い順に並べる

src	instAddr	memAccess	miss%	conf%	miss%	conf%	miss%	conf%
himenoBMTxpa.c:292	402e6f	12418042	100.00	93.62	6.36	0.95	100.00	0.95
himenoBMTxpa.c:293	402e7d	12418042	100.00	93.50	6.36	1.48	100.00	1.48
himenoBMTxpa.c:294	402eba	12418042	100.00	93.44	6.36	1.78	100.00	1.78
himenoBMTxpa.c:297	402f0b	12418042	100.00	93.44	6.36	20.70	100.00	20.70
himenoBMTxpa.c:300	402f1f	12418042	100.00	92.49	6.36	1.51	100.00	1.51
⋮								

(1.2) 命令ごとにキャッシュミスとコンフリクトミス率を取得すると、ステンシル計算時のメモリアクセスが競合の原因と分かる

(2) Paddingの実施:
ステンシルのIndexingのマクロ、データ領域確保の部分

- #define MR(mt,n,r,c,d) mt->m[(n) * mt->mrows * mt->mcols * mt->mdeps + (r) * mt->mcols* mt->mdeps + (c) * mt->mdeps + (d)]
- malloc(mnums * mrows * mcols * mdeps * sizeof(float));

変更後

- (mrows+1), (mcols+1), (mdeps+1) に変更
- malloc(mnums * (mrows+1) * (mcols+1) * (mdeps+1) * sizeof(float));

(3) チューニング版(Paddingあり) キャッシュの挙動

Cache miss: L1=2.70 %、L2=98.97 %、L3=82.27 %
Conflict miss: L1=1.03 %、L2=11.21 %、L3=8.19 %

src	instAddr	memAccess	miss%	conf%	miss%	conf%	miss%	conf%
himenoBMTxpa.c:292	40323c	12508717	6.41	0.00	100.00	6.17	100.00	6.17
himenoBMTxpa.c:297	4032d4	12508717	6.41	0.00	100.00	10.16	100.00	10.16
himenoBMTxpa.c:307	40335a	12508717	6.41	0.00	100.00	3.29	100.00	3.29
himenoBMTxpa.c:309	40336d	12508717	6.41	0.00	100.00	4.59	100.00	4.59
himenoBMTxpa.c:293	403235	12508314	6.41	0.00	100.00	19.67	100.00	19.67
⋮								

L1の競合ミスが解消され、L2の容量ミスとなったことが分かる
容量ミスを改善するためにはアクセスパタンの改善が必要

(4) 実機での性能検証 Intel Xeon E5-2680 v2 @ 2.80GHz にて検証

Paddingなし	1837 MFLOPS
Paddingあり	2318 MFLOPS

メモリレイアウトのチューニングにより26%のスコア向上を確認

[1] Yukinori Sato, Yasushi Inoguchi, Tadao Nakamura, Whole Program Data Dependence Profiling to Unveil Parallel Regions in the Dynamic Execution, 2012 IEEE International Symposium on Workload Characterization, pp.69-80, 2012/11/5

[2] Yuki Matsubara and Yukinori Sato. Online memory access pattern analysis on an application profiling tool. In Proceedings of 2014 Second International Symposium on Computing and Networking, pp.602-604, December 2014. . (WANC2014)