C-028

SR11000 コンパイラにおけるデータキャッシュ向け最適化

Optimization for Data Cache on SR11000 Compiler

本川 敬子† 伊藤信一‡ 橋本 博幸† 久島 伊知郎†
Keiko Motokawa Shinichi Itou Hiroyuki Hashimoto Ichiro Kyushima

1．はじめに

科学技術計算においてプログラムの実行速度を上げるためには，実行頻度の高いループに対するデータキャッシュを意識した最適化が特に重要である．キャッシュ向け最適化としては，キャッシュミス時のメモリレイテンシを隠蔽するプリフェッチ最適化や，キャッシュミス回数を削減するデータ局所化を目的としたループ交換・ループタイリングなどのループ変換が知られている．

本稿では，日立スーパテクニカルサーバ SR11000 モデル H1 に搭載された日立最適化 FORTRAN90 コンパイラにおけるデータキャッシュ向け最適化のうち，ロードストリームに対するソフトウェアプリフェッチ，最適化指示文を用いた単一ループ向けおよび複数ループ向けのタイリング最適化にについて述べる．SR11000 モデル H1 のキャッシュ構成は L1:32KB/1CPU, L2:1.5MB/2CPU, L3:256MB/16CPU であり，最適化 FORTRAN90 では主として L2 キャッシュを考慮した最適化を適用している．

2．プリフェッチ最適化

SR11000 モデル H1 の CPU である POWER4+では，ハードウェアによるデータプリフェッチエンジンを搭載し，最大 8 個のロードストリームを検出してキャッシュへのプリフェッチを自動的に行う[1]．ループ内のロードストリームが 8 個を超える場合には，プリフェッチエンジンが全てのストリームをプリフェッチすることはできないため，実効メモリバンド幅が低下する．dcbt(Data Cache Block Touch) 命令を発行する，ソフトウェアプリフェッチ(SWPF)を適用すれば，ハードウェアによるストリーム数の上限を解消することができる．

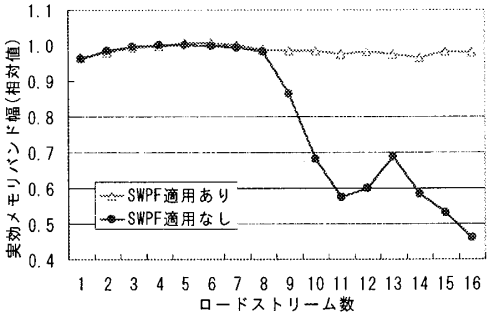

図 1　実効メモリバンド幅

図 1 は，ロードストリームを 1 から 16 まで変化させたループについて実効メモリバンド幅を測定した結果である（16CPU で並列実行）．SWPF 適用なしではロードストリームが 8 を超えると性能が低下するのに対して，SWPF を適用した場合には安定して高い性能が得られる．最適化 FORTRAN90 では，ループ中の全ストリームに対して dcbt 命令を発行するプリフェッチ最適化を自動的に行っている[2].

3．ループタイリング最適化

ループタイリングは，キャッシュサイズより大きいデータを参照する多重ループネストに対して，参照データがキャッシュサイズ以下となるようにループのイタレーション空間をグループ化することにより，データ局所性を高めるループ変換である[3].

最適化 FORTRAN90 ではタイリング最適化の適用方法を指定する指示文を備えている．図 2 に行列積のループに対する指示文指定とタイリングの適用を示す．指示文 tiling はタイリング最適化の対象ネストを示し，tilesize はタイル分割の対象ループとタイルサイズを指定する．s1,s2 には実際には 1 以上の整定数を指定する．

```
*soption tiling
    do k=1,N
*soption tilesize(s1)
      do j=1,N
*soption tilesize(s2)
        do i=1,N
          C(i,k) = C(i,k)+A(i,j)*B(j,k)
        enddo
      enddo
    enddo
```

(a) 行列積ループの指示文入りソースプログラム

```
do jj=1,N,s1
  do ii=1,N,s2
    do k=1,N
      do j=jj,min(jj+s1-1,N)
        do i=ii,min(ii+s2-1,N)
          C(i,k) = C(i,k)+A(i,j)*B(j,k)
        enddo
      enddo
    enddo
  enddo
enddo
```

(b) 適用後のコードイメージ

図 2　タイリング最適化の例

図 2 の例で，元のループでは配列 A は i, j ループの繰り返しにより N×N 個の要素を参照し，k ループの繰り返しによりこれらの要素を繰り返し参照する（再利用するという）．タイリング適用後は i, j ループの繰り返しによる参照が s1×s2 個の要素となり，再利用間のデータ参照量が削減されるため，キャッシュヒットの可能性が高まりデータ局所性が向上する．

† （株）日立製作所システム開発研究所
‡ （株）日立製作所ソフトウェア事業部

最適化 FORTRAN90 では，コンパイラによる自動的なタイリング最適化の機能も備えている．ループネスト内の参照データを解析して，オリジナルのループにおける再利用間のデータ参照量がキャッシュサイズより大きいループネストを最適化対象とし，再利用間での参照データサイズがキャッシュサイズを超えない最大のタイルサイズを 2 のべき乗の値から選択する．タイルサイズはループネスト内の全ループに対して同一（図 2 の s1,s2 は同じ値をとる）としている．

4. 複数ループに跨るストリップマイニング

最適化 FORTRAN90 では stripmine 指示文により複数ループネストに跨ったタイリング変換（ストリップマイニング最適化と呼ぶ）を適用できる．図 3(a)は２個の 2 重ループを最適化対象とする指示文指定の例である．stripmine～end stripmine が最適化範囲を示し，s（1 以上の整定数）がタイルサイズを示す．stripmine の第２パラメタは分割対象のループを最内側からのレベルで指定するもので，本例ではパラメタ 1 の指定により最内側ループを指定している．適用後のコードは図 3(b)のようになる．第 1 の 2 重ループで参照される配列 A の要素数は，元のループでは M1×N だったのに対して，ストリップマイニング適用後は j ループがサイズ s で分割されているので M1×s に縮小され，第 2 の 2 重ループの実行時に第 1 の 2 重ループで参照した配列 A のデータがキャッシュ上に残っている可能性が高まる．

このようにストリップマイニング最適化は，複数のループネスト間でデータを再利用する場合に有効であり，ループに跨ってデータ局所性を向上させるループ変換により，キャッシュミスを削減する．

```
*soption stripmine(s,1)
  do i=1,M1
    do j=1,N
      ... A(j,i) ...
    enddo
  enddo
  do i=1,M2
    do j=1,N
      ... A(j,i) ...
    enddo
  enddo
*soption end stripmine
```

(a) 指示文入りソースプログラム

```
do jj=1,N,s
  do i=1,M1
    do j=jj,min(jj+s-1,N)
      ... A(j,i) ...
    enddo
  enddo
  do i=1,M2
    do j=jj,min(jj+s-1,N)
      ... A(j,i) ...
    enddo
  enddo
enddo
```

(b) 適用後のコードイメージ

図３　ストリップマイニング最適化変換

5. 最適化の効果

本稿で述べた最適化のうちプリフェッチ最適化およびタイリング最適化の効果を調べた．

(1) プリフェッチ最適化の効果

SPECfp2000 および NAS Parallel Benchmarks(NPB) の Fortran プログラムを対象にソフトウェアプリフェッチを適用した場合の性能向上を調べた結果を図 4 に示す．SPECfp2000 は 1CPU 上で実行したもの，NPB(Class C)は 16CPU 上で並列実行した結果である．swim, mgrid, CG, SP でソフトウェアプリフェッチによる効果が得られた．

(2) タイリング最適化の効果

図 2 に示した行列積のプログラム（データ型は倍精度浮動小数）でループ長 N が 3000, 5000, 10000 の場合に対して，タイルサイズ(S1,S2)を変化させてタイリングを適用した場合の性能向上（タイリングなしの性能を 1 とする）を図 5 に示す．タイルサイズ 200 のときの性能が最もよく，16%から 60%の性能向上が得られた．図 5 からタイルサイズが小さすぎても大きすぎても効果がなく，適切なタイルサイズの指定が重要であることがわかる．コンパイラによる自動最適化ではタイルサイズ 256 を選択し，この場合の性能向上は 10%から 27%である．

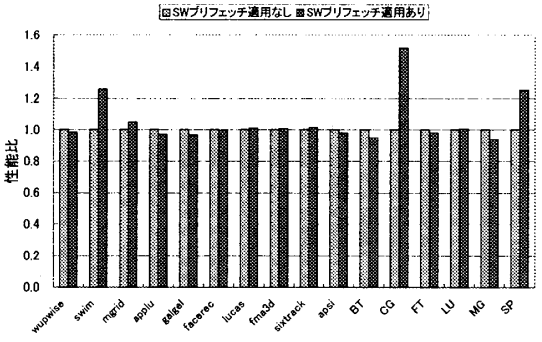

図４　プリフェッチ最適化の効果

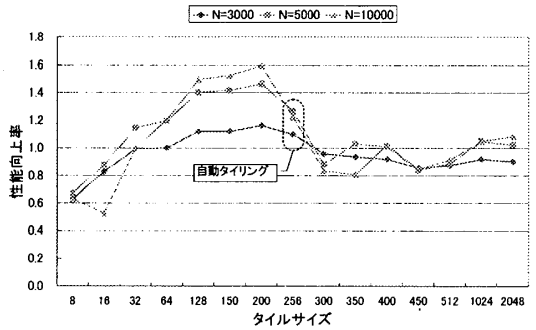

図５　タイリング最適化の効果

6. おわりに

SR11000 モデル H1 に搭載された日立最適化 FORTRAN90 コンパイラにおけるデータキャッシュ向け最適化のうち，ソフトウェアプリフェッチ，タイリング最適化，ストリップマイニング最適化について述べた．ソフトウェアプリフェッチは，多ストリームのループで効果を発揮する．タイリング最適化，ストリップマイニング最適化では指示文を用いたチューニング手段を提供しており，データ局所性の向上によりキャッシュミスの削減を図る．

参考文献

[1] J.M.Tendler, et al.: “POWER4 system microarchitecture”, IBM Journal of Research and Development, Vol. 46, No.1, pp.5-25, 2002.

[2] 青木,處,本川,五百木,石原：“SR11000 モデル H1 のソフトウェアプリフェッチ手法”，電子情報通信学会 2004 年総合大会，D-6-11，2004.

[3] M.E.Wolfe: “Improving Locality and Parallelism in Nested Loops”, Stanford University, CSL-TR-92-538, 1992.