計算科学技術 特論B



第5回 アプリケーションの性能最適化の実例2

2018年5月



講義の概要

- ・スーパーコンピュータとアプリケーションの性能
- ・アプリケーションの性能最適化1(高並列性能最適化)
- ・アプリケーションの性能最適化2(CPU単体性能最適化)
- ・アプリケーションの性能最適化の実例1
- ・アプリケーションの性能最適化の実例2



内容

- ・理研で進めた性能最適化
- ・Seism3Dの性能最適化
- ・FrontFlow/blueの性能最適化
- ・NPB MGのチューニング事例

本資料は,理化学研究所AICS運用技術部門ソフトウェア技術チーム,井上俊介氏(現所属富士通),熊畑清氏の発表データを使用して作成しています.

2018年5月17日 計算科学技術特論B

3



理研で進めた性能最適化



理研で進めた性能最適化

6本のターゲットアプリ

	プログラム名	分野	アプリケーション概要	期待される成果	手法
	NICAM	地球 科学	全球雲解像大気大循環 モデル	大気大循環のエンジンとなる熱帯積雲対流活動を精 織に表現することでシミュレーションを飛躍的に進化さ せ,現時点では再現が難しい大気現象の解明が可能 となる.(開発 東京大学,JAMSTEC,RIKEN AICS)	FDM (大気)
	Seism3D	地球 科学	地震波伝播・強震動 シミュレーション	既存の計算機では不可能な短い周期の地震波動の解析・予測が可能となり、木造建築およびコンクリート構造物の耐震評価などに応用できる。(開発東京大学地震研究所)	FDM (波動)
	PHASE	ナノ	平面波展開第一原理 電子状態解析	第一原理計算により、ポスト35nm世代ナノデバイス、 非シリコン系デバイスの探索を行う.(開発 物質・材料 研究機構)	平面波 DFT
	FrontFlow/Blue	工学	Large Eddy Simulation (LES)に基づく非定常流 体解析	LES解析により、エンジニアリング上重要な乱流境界 層の挙動予測を含めた高精度な流れの予測が実現で きる.(開発 東京大学生産技術研究所)	FEM (流体)
	RSDFT	ナノ	実空間第一原理電子状 態解析	大規模第一原理計算により、10nm以下の基本ナノ素 子(量子細線、分子、電極、ゲート、基盤など)の特性解 析およびデバイス開発を行う.(開発 東京大学)	実空間 DFT
	LatticeQCD	物理	格子QCDシミュレーショ ンによる素粒子・原子核 研究	モンテカルロ法およびCG法により、物質と宇宙の起源 を解明する.(開発 筑波大)	QCD
2018年5月17日 計算科学技術 特論B 555 555 555 555 555 555 555 555 555 55					K computer

理研で進めた性能最適化

コラボレーション



K computer

2018年5月17日 計算科学技術 特論B

筑波大 RIKEN AICS

東京大学生産技術研究所

理研で進めた性能最適化

6本のターゲットアプリの計算機科学的な位置づけ



理研で進めた性能最適化

6本のターゲットアプリの計算機科学的な位置づけ



Seism3Dの性能最適化



2018年5月17日 計算科学技術 特論B

Seism3D



Seism3Dの並列化



2018年5月17日 計算科学技術特論B

11

Seism3Dの並列特性分析





Seism3Dの並列特性分析

当初の評価アプリについて





Seism3Dの並列化





- 今後計算体系の上面のみで実行される津
 波の計算を鑑み2次元分割が計画された
- ・ 2次元分割での通信時間を評価
- 通信量は4-5倍になるが通信回数は2/3に なると評価
- ・ 通信のパッキングは2次元が有利
- ・ 通信時間は演算時間の数%ですむと評価

新しい評価アプリ





Seism3Dの単体性能最適化 (性能見積りとチューニング) Seism3Dは要求B/F値が大きいアプリケーション CPU単体性能チューニング手順 (5)改良コーディングの性能予測 改良 コーディン: の予測性能 (要求B/Fが高いアプリについて) (6)更なる性能チュー ング (3)性能推定(要求B/Fが高いアプリについて) オリジナル コーディンク の予測性能 存続 (4)性能チューニング (2)プロファイラー測定結果を 使った問題の発見 2.08+00 1.86+00 1.86+00 1.46+00 1.26+00 1.06+00 8.06-01 8.06-01 2.06-01 0.06+00 オリジナル 性能 低

2018年5月17日 計算科学技術 特論B

ルーフラインモデル

・(1)プロファイラーを使った測定

15



S. Williams, A. Waterman, and D. Patterson: Roofline: an insightful visual performance model for multicore architectures. Commun. ACM, 52:65–76, 2009.



• ハードウェアの実効的なメモリバンド幅:B

- アプリケーションの演算強度:X=f/b
 - アプリケーションの要求フロップス値:f
 - ・アプリケーションの要求バイト値:b

アプリケーションの性能: min(F, B*X)

. ACM, 52.05-70, 2007.

B*X=B*(f/b)=B*F*f/(b*F)=(B/F)/(b/f)*F

アプリケーションの性能 :min(F, (B/F)/(b/f)*F) アプリケーションのピーク性能比 :min(1.0, (B/F)/(b/f))

ルーフラインモデルの別の見方

メモリ

Mdata

- ・メモリデータ転送量:Mdata Mband[.] 実効メモリバンド幅: Mband, • メモリデータ移動時間: Mtime L2\$ 演算器 Mtime=Mdata/Mband (2)演算 ・プログラム演算量:Ca
 ・演算ピーク性能:Ppeak • 最小演算時間: Ctime Ctime=Ca/Ppeak 演算器の処理とメモリのデータ (3)実行時間:Ex time の処理が同時に終わるモデル Ex time=max(Mtime. Ctime) 17 2018年5月17日 計算科学技術 特論B
 - メモリとキャッシュアクセス(1)





(1)メモリ

空間微分Z方向の計算a)b)(1次元目の差分)

do J = 1, NY	
do I = 1, N	Х
do K = 3	, NZ-1
DZV (k	(I,J) = (V(k,I,J) - V(k-1,I,J))*R40 &
	- (V(<mark>k+1</mark> ,I,J)-V(<mark>k-2</mark> ,I,J))*R41
end do	
end do	
end do	

要求B/F	12/5 = 2.4
性能予測	0.36/2.4 = 0.15
実測値	0.153

アプリケーションのピーク性能比の予測値 :ハードウェアのBF値/アプリの要求BF値

2018年5月17日 計算科学技術特論B

19

ハードのB/F値の算出:

実効メモリバンド幅/ピーク性能 =46(GB/sec)/128(GFlops) =0.36

要求Byteの算出:

1store,2loadと考える

4x3 = 12byte

要求flop:

add : 3 mult : 2 = 5



空間微分X方向の計算a)b)(2次元目の差分)

do J = 1, NY do I = 1, NX do K = 1, NZ DXV (k,I,J) = (V(k,I,J) -V(k,I-1,J))*R40& - (V(k,I+1,J)-V(k,I-2,J))*R41 end do end do end do

要求B/F	12/5 = 2.4
性能予測	0.36/2.4 = 0.15
実測値	0.151

- 第2軸(I軸)が差分
- 1ストリームでその他の3配列は \$L1or\$L2に載っており再利用できる
- 従って1次元目が差分のパターンと同じ性能になる

要求Byteの算出:

P12より、メモリコストだけを考慮する。

1sore,2loadと考える

4x3 = 12byte

要求flop:

add : 3 mult : 2 = 5



空間微分Y方向の計算a)b)(3次元目の差分)

最外ループをブロックでスレッド並列			
do J = 1, NY		第3軸が差分 →	再利用性なし
do I = 1, NX			
do K = 1, NZ			
DYV $(k,l,J) = (V(k,l,J) - V(k,l,J-1))*R40 \&$			
- (V(k,l,J+1)-V(k,l,J-2))*R41			
end do			
end do			
end do	要又	ҟflop:	

要求flop:

add : 3 mult : 2 = 5

要求B/F	24/5 = 4.8		
性能予測	0.36/4.8= 0.075		
実測値	0.076		

要求Byteの算出:

1store/5loadより

(5+1) * 4byte = 24



2018年5月17日 計算科学技術 特論B



21

要求するB/Fが大きいアプリケーションについて



空間微分Y方向の計算a)b) (Tuning1:3次元目をcyclicでスレッド並列化)

!\$OMP DO SCHEDULE(static,1),PRIVATE(I,J,K)

do J = 1, NY do I = 1, NX do K = 1, NZ DYV (k,I,J) = (V(k,I,J) - V(k,I,J-1))*R40 & - (V(k,I,J+1)-V(k,I,J-2))*R41 end do end do end do

	要求B/F	12/5 = 2.4		
	性能予測	0.36/2.4 = 0.15		
	実測値	0.136		
2018年5月17日 計算科学技術 特論 B 23				

キャッシュに載せる

- 第3軸をcyclic分割 → 1ストリーム
 で3配列がL2に乗る(説明次項)
- 性能が2倍になる

要求Byteの算出:

1sore,2loadと考える

4x3 = 12byte **要求flop:**

add : 3 mult : 2 = 5



(cyclic分割スレッド並列の説明)



<u>空間微分Y方向の計算a)b)</u> (Tuning2:ZXYループ融合cyclicスレッド並列)

	要求B/F	28/15 = 1.86		
	性能予測	0.36/1.86 = 0.19		
	実測値	0.177		
2018年5月17日 計算科学技術 特論 B 25				

要求B/F値を下げる キャッシュに載せる

 K,I,J軸差分のループを融合する ことにより、V(K,I,J)のロードを 共通化でき、プログラムの要求 B/F比を下げる。

要求Byteの算出:

Store 3 +4 load と考えると、

 $(3+4)^*4 = 28$ byte

要求flop:

add : 9 mult : 6 = 15

Tuning3:XFILL指示行によるチューニング

- · 下記ループのdzvはストアのみの配列
- しかし通常はロードとストアが発生するためメモリアクセスは2と計算 する
- · 京のコンパイラではXFILL指示行の機能をもつ
- ・ この指示行を指定することにより余計なロードが発生しなくなる
- これにより要求B/F値は2.4から1.6に減少し推定性能が15%から22.5%に
 向上する
- ・ 実測値は21.2%

```
\begin{array}{l} \textbf{!OCL XFILL} \\ \text{do } j = 1, \text{ NYP} \\ \text{do } i = 1, \text{ NXP} \\ \text{do } k = 3, \text{ NZ-1} \\ \text{dzv } (k,i,j) = ( v(k,i,j) - v(k-1,i,j) ) * \text{ R40 & } \\ - ( v(k+1,i,j) - v(k-2,i,j) ) * \text{ R41} \\ \text{end do} \\ \text{end do} \\ \text{end do} \\ \text{end do} \end{array}
```



キャッシュの有効利用



キャッシュの有効利用



キャッシュスラッシングの解消

- 連続アクセスのキャッシュミスの基準値はseism3Dは単精度であるため 3.125%となる(1回/32回=1/(4B/128B))
- L1D\$ミス率が基準値を超えL1D\$ミスdm率が20%を超えるとL1キャッシュ スラッシングの可能性が高い
- L1キャッシュスラッシングはループ内の配列ストリームアクセスが多い
 場合に起きる確率が高くなる
- ・ このような場合はループ分割や配列融合が効果的

境界条件応力場更新	Org	Tune	+1155+11.
L1Dミス率	3.54%	2.71%	・ L1Dミス率:3.54%(>3.125%)
L1Dミスdm率	49.93%	11.83%	 L1Dミスdm 49.93%(>20%) → L1キャッシュ競合により、メモリスループットが
L1Dミスhwpf率	25.99%	88.17%	+分でない.
L1Dミスswpf率	24.08%	0.00%	ループ分割
L2ミス率	2.11%	1.97%	
L2スループット	42.32GB/s	35.79GB/s	
メモリスループット	39.86GB/s	42.87GB/s	・ ループ分割の結果, 最内ストリームが減少し, hwnflによるミスが支配的
Peak Ratio	8.80%	10.05%	 ・メモリスループットおよび性能が向上

2018年5月17日 計算科学技術 特論B

29

速度時間積分の計算 e)

オリジナルコードの結果

要求B/F	72/52=1.38		
性能予測	0.36/1.38 = 0.26		
実測値	0.240		



K computer

CPU単体性能チューニングの結果



Seism3Dの総合性能

(*)通信を含む性能

The number of node	Elapse time(sec)	Ratio to peak performance
16	48.8	17.1%
256	48.8	17.6%
4096	48.9	17.7%
16384	48.8	17.8%
36864	48.6	17.9%
64512	48.5	17.9%
82944	48.5	17.9%

■ 8万ノードまでの良好なウィークスケーラビリティを得られた
 ■ フルノードでトータル性能1.9Pflopsを達成



FrontFlow/blueの性能最適化



2018年5月17日 計算科学技術 特論B

33



FrontFlow/blue (FFb)



- 全体剛性マトリクスを構築するタイプ
- 全体構成マトリクスを構築せずに要素剛性マトリクスのみで計算を進めるタイプ(エレメント・バイ・エレメント法)
- ■FFBは新バージョンにおいて両方のソルバに対応

全体剛性マトリクスを使用する方法(陰解法の場合)

要素剛性マトリクス



FFbのタイムステップループ構造



FFbの主要部の構造



FFbの主要部の構造



FFbの主要計算



- 10万個の4面体要素+α
- 14.11s(30%)が4面体要素の勾配計算
- 13.04s(27%)が疎行列ベクトル積
- 8.32s(17%)が4面体要素の発散計算



FFbの並列化

3次元の領域分割(非構造格子)





2018年5月17日 計算科学技術特論B

39



FFbの並列特性分析





規模を拡大しても、1コア当たりの計算規模は同等としている



FFbの並列特性分析

初期のウィークスケール評価



FFbの並列特性分析

問題設定を簡素化

- ・ 下左図のようなキャビティの問題がある
- その問題を下右図のように横に連結してそれぞれのノードが同じ問題 を解く仮想的な問題を作成
- こうすることにより各ノードは全く同じ問題を計算する完全なウィー クスケールの問題となる
- これでシンプルにアプリケーションの問題のみをあぶり出す事が可能
 となる



FFbの並列特性分析

- ・ ここでは1000プロセスと8000プロセスの結果を示す
- ・ 演算は完全にウィークスケールしておりロードインバランスもない
- ・ 隣接通信の増大もないが大域通信 (スカラ値のallreduce) のみが問題と分かった



FFbの単体性能最適化 (性能見積りとチューニング)

FFbは要求B/F値が大きく、かつリストアクセスを使用 するアプリケーション



オリジナルコード(疎行列とベクトルの積)

ICRS=0 DO 110 IP= BUF=0.01 DO 100 k ICRS= IP2=IF BUF=1 100 CONTII	:1,NP E0 URS+1 CRS+1 CRS(ICRS) BUF+A(ICRS)*S(IP2)	CRS格納形式の行列ベクトル積 ベクトルアクセスがリストアクセス となる ベクトルの部分がL1キャッシュに 載っていると仮定した場合 ベクトルのメモリへのアクセスを 全く無視してよい メモリからのロードは行列とリスト のみ
110 CONTUI	NE 行列	単精度:2 load なので
		2*4 = 8byte
要求B/F	8/2 = 4	要求flop:
性能予測	0.36/4 = 0.09	~3oh.

(スレッド並列を仮定しピーク性能128Gflopsに対して) add: 1 mult: 1 = 2

2018年5月17日 計算科学技術特論B

45

節点圧力カーネル (calaxc相当)の実測性能

(スレッド並列なし:1コア)

- オリジナルコードはスレッド並列されていなかった
- この状態の推定性能を前頁と同様な方法で見積もる
- メモリバンド幅を1コアで占有する場合のSTREAMベンチマークの結果は 20GB/秒
- 1コアの理論ピーク性能は16GFLOPS
- 従って理論的なB/F値は20GB/16GFLOPで1.25

要求Byteの算出: 2loadより 2* 4byte = 8

要求flop:1(add)+1(mult) = 2

要求B/F	8/2 = 4
性能予測	1.25/4= 0.313
実測値	0.059(六面体)
	0.024(四面体)

(スレッド並列なしピーク性能16Gflopsに対して)

- ベクトルがリストアクセス
- 連続アクセスでないためプリフェッチが 効きにくい
- メモリアクセスのレイテンシが見える
- 最悪1キャッシュラインのうち1要素しか 使用できない事による大きなペナルティ が発生
- 著しい性能低下が発生
- L2オンキャッシュでも同様のペナルティ が発生



チューニング1: フルアンロール



2018年5月17日 計算科学技術特論B

47

チューニング2: リオーダリング(1/4)



チューニング2: リオーダリング(2/4)

節点番号のリオーダリング: ■オリジナルデータを各軸分割しブロックを作成 ■各ブロックを外と内に分割し物理座標に基づき内側・外側の順 にナンバリング



チューニング2: リオーダリング(3/4)



4面体リオーダリング結果 6面体リオーダリング結果











節点圧力カーネル (calaxc相当) チューニング結果

	6 面体	4 面体
フルアンロール (8core)	5.4%	3. 0%
フルアンロール + リオーダリング (8core)	8.1%	7. 7%
L1 オンゴ	Fャッシュである時 ある9%に近い性能	の理論性能値で 値を実現



節点圧力カーネル (callap) のチューニング

カーネル概要 1/2

四面体要素について ∇p の有限要素近似 $\nabla p = \frac{\partial p}{\partial x_i} \cong \sum_{i=1}^{\partial N_j} \frac{\partial p}{\partial x_i} p_e$ を計算する



節点圧力カーネル (callap) のチューニング

カーネル概要 2/2



- ・要求バイト 16要素×4B×(9/4)+1要素×4B=148B
- ・浮動小数点演算数 24FLOP
- ・要求B/F値は148/24=6.17
- STREAMベンチマークによる「京」の実効メモリバンド 幅は46.6GB/s
- ・「京」の実効B/F値は46.6/128=0.36
- ・メモリバンド幅がネックとなり実効上の性能上限は 128GFLOPSの0.36/6.17=5.83%

2018年5月17日 計算科学技術 特論B

節点圧力カーネル (callap) の性能

- ・カラーリングのみを実施しスレッド並列した時点で1.6%のピーク性能比
- ・さらにブロックカラーリングの手法を適用
- ・calaxcと同様の節点のオーダリングを行う事でピーク性能比:3.78%を達成
- ・さらに配列融合を実施し4.41%まで向上

パターンNo	内容
1	DNX, DNY, DNZをDNXYZに融合
2	FX, FY, FZをFXYZに融合
3	パターン2+演算淳子変更
4	パターン3+パターン1

パターンNo	ピーク 性能比%	L1D キャッシュ ミス率%	メモリ スルー プット GB/s
1	3.95	3.98	35.70
2	4.05	3.69	35.91
4	4.05	3.69	35.88
5	4.41	3.37	38.80

・加えてブロック間の要素数のバランスを改善することによりピーク性能 比:4.58%,メモリスループット:41.22GB/secまで到達



FFbのウィークスケール性能(Ver5)



FFbの総合性能(Ver7)

現状の総合性能は80000ノードでピーク性能比:3.16%







59



NPB MGを使用した解析例

NPB MG (マルチグリッド法により連立一次方程式 を解くベンチマークテストプログラム)

Vサイクルマルチグリッド





NPB MGを使用した解析例

NPB MG (マルチグリッド法により連立一次方程式 を解くベンチマークテストプログラム)



プロファイラーによるコスト分析

Procedures profile

**************************************	**************************************	**************** }\$ ******************	·*************************************	***************************************	**************************************	******		
Cost	. %	Barrier	%	MPI	%	Start	End	
1367	100.0000	1	0. 0732	8	0. 5852			Application
620 277 140	45. 3548 20. 2633 10. 2414	0 0 0	0.0000 0.0000 0.0000	0 0 0	0.0000 0.0000 0.0000	736 664 826	 758 686 858	residOMP_1_ psinvOMP_1_ rpri3OMP_1
131 80	9.5830 5.8522	0	0.0000	0	0.0000	913 1241	948 1243	interpOMP_1_ readyPRL_1_
20 19 17	1. 3899 1. 2436	0 0 0	0.0000 0.0000 0.0000	2 6 0	10.0000 31.5789 0.0000	1398 1263 2290	1493 1393 2296	take3_ give3_ zran3. PRL 1
15 14	1.0973 1.0241	0 0	0.0000 0.0000	0	0.0000	2115	2310	jwe_etbf zran3_

rprj3:制限補完ルーチン





ループのB/Fから推定性能を算出する

			Mst	Mid	L2st	L2Id	L1st	L1Id	市地省	垂笛		
		回転数	配列	配列	配列	配列	配列	配列	111 894-94	**	要求B/F	推定性能
rp	nj3	(最大値)	1	7	0	12	2	7	20	4	2.666667	0.135
1	do j3=2,m3j-1	256										
	i3 = 2*j3-d3											
2	do j2=2,m2j-1	256										
3	i2 = 2*j2-d2											
4	do j1=2,m1j	514										
5	i1 = 2*j1-d1											
6	x1(i1-1) = r(i1-1,i2-1,i3) + r(i1-1,i2+1,i3)											
	> + r(i1-1,i2, i3-1) + r(i1-1,i2, i3+1)			3		1	1	1	3			
7	y1(i1-1) = r(i1-1,i2-1,i3-1) + r(i1-1,i2-1,i3+1)											
	> + r(i1-1,i2+1,i3-1) + r(i1-1,i2+1,i3+1)					4	1	1	3			
8	enddo											
9	do j1=2,m1j-1	512										
10	i1 = 2*j1-d1											
11	$y^2 = r(i1, i2-1,i3-1) + r(i1, i2-1,i3+1)$											
	> $+ r(i1, i2+1,i3-1) + r(i1, i2+1,i3+1)$			2		2			3			
12	$x^2 = r(i1, i2-1,i3) + r(i1, i2+1,i3)$											
>	+ r(i1, i2, i3-1) + r(i1, i2, i3+1)			1		3			3			
13	s(j1,j2,j3) =											
	> 0.5D0 * r(i1,i2,i3)											
	> $+ 0.25D0 * (r(i1-1,i2,i3) + r(i1+1,i2,i3) + x2)$											
	> + 0.125D0 * (x1(i1-1) + x1(i1+1) + y2)											
	> + 0.0625D0 * (y1(i1-1) + y1(i1+1))		1	1		2		5	8	4		
14	enddo									<u> </u>		
15	endo			_			~ ~	_				_
16	***** 性能の自然のない。 メモリ律速と仮定すると、	5 女 期待	B/F され	= 6 ,る性	54/2 E能低	24= 直は	2.6	(36/	2.5	7 =	13.5	%
		(20									ē

ループrprj3の分析(1)



✔ メモリスループット

最大值: 46G/s

メモリビジー率: メモリスループットの最大値(46G/s)に 対する割合

Ксон

Memory Cache

メモリスループット (GB/sec)	L2エンジン スループット (GB/sec)	L2 スループット (GB/sec)	メモリビジー率	L2エンジンビジー率	L1エンジンビジー率
3.37	15.76	11.17			44%
3.34	15.71	11.18			44%
3.34	15.72	11.16			44%
3.35	15.69	11.14			44%
3.35	15.66	11.09	58%	70%	44%
3.33	15.72	11.18			44%
3.37	15.80	11.22			44%
3.35	15.73	11.17			44%
26.78	125.70	89.23			44%

B/Fが高いループであるが、メモリスループットが低い。。



ループrprj3の分析(2)

√ キャッシュ状況

L1D、L2ミス率 倍精度であれば6.25%が基準値 それを上回り、L1Dミスdm率が20%を超えている場合は L1キャッシュスラッシングが発生していると考える。

040110													
	L1I ミス率 (/有効総命 令数)	L1D ミス率 (/ロード・ス トア数)	ロード・スト ア数	L1D ミス数	L1D ミス dm 率 (/L1D ミス 数)	L1D ミス hwpf 率 (/L1D ミス 数)	L1D ミス swpf 率 (/L1D ミス 数)	L2 ミス率 (/ロード・ス トア数)	L2 ミス数	L2 ミス dm 率 (/L2 ミス数)	L2 ミス pf 率 (/L2 ミス数)	μDTLB ミ ス率 (/ロード・ス トア数)	mDTLB ミ ス率 (/ロード・ス トア数)
Thread 0	0.03%	12.52%	2.99E+08	3.75E+07	41.47%	58.53%	0.00%	3.43%	1.03E+07	9.62%	90.38%	0.00599%	0.00017%
Thread 1	0.03%	12.58%	2.99E+08	3.76E+07	41.52%	58.48%	0.00%	3.40%	1.02E+07	8.57%	91.43%	0.00449%	0.00010%
Thread 2	0.03%	12.57%	2.99E+08	3.76E+07	41.81%	58.19%	0.00%	3.42%	1.02E+07	8.04%	91.96%	0.00431%	0.00005%
Thread 3	0.03%	12.53%	2.99E+08	3.75E+07	41.36%	58.64%	0.00%	3.41%	1.02E+07	9.05%	90.95%	0.00417%	0.00004%
Thread 4	0.03%	12.48%	2.99E+08	3.73E+07	41.78%	58.22%	0.00%	3.43%	1.03E+07	9.20%	90.80%	0.00414%	0.00003%
Thread 5	0.03%	12.59%	2.99E+08	3.77E+07	41.66%	58.33%	0.00%	3.41%	1.02E+07	8.31%	91.69%	0.00430%	0.00006%
Thread 6	0.03%	12.63%	2.99E+08	3.78E+07	41.84%	58.16%	0.00%	3.44%	1.03E+07	7.97%	92.03%	0.00415%	0.00004%
Thread 7	0.03%	12.57%	2.99E+08	3.76E+07	41.56%	58.44%	0.00%	3.43%	1.03E+07	8.96%	91.04%	0.00494%	0.00004%
Process	0.03%	12.56%	2.39E+09	3.01E+08	41.62%	58.37%	0.00%	3.42%	8.19E+07	8.71%	91.29%	0.00456%	0.00007%

L1キャッシュスラッシングが発生していると判定。



2018年5月17日 計算科学技術 特論B

65

ループrprj3の分析結果

- ・ し1キャッシュスラッシングが発生している。
- ・ 演算系はSIMD化率が低い

⇒ 本ループは、演算の比率が低い(B/Fが高い)ため、 まずはスラッシングを回避する処置が必要と判断する。

スラッシングの回避策:

・使用する配列数を減らす ・しープ分割、配列融合、スカラ化 ・使用する配列のアドレスをずらす ・使用する配列のアドレスをすらす ・Common化



ループrprj3のチューニング(1)

テンポラリ配列x,yの利用をやめることによ ループ融合を実施し、無駄なロードを発生さ	り、配列数を少なくするのと同時に させないチューニングを採用する。
$ \begin{array}{cccccccccccccccccccccccccccccccccccc$	$ \begin{cases} $$ OMP PARALLEL DO PRIVATE(xy1,y2,x2,i3,i2,i1)$ do j3=2,m3j-1$ i3 = 2*i3-d3$ do j2=2,m2j-1$ i2 = 2*j2-d2$ do j1 = 2*j1-d1$ C i1 = 2*j1-d1$ C i1 = 2*j1-d1$ y11m = r(i1-1,i2-1,i3) + r(i1-1,i2,i3+1)$ xy11p = r(i1+1,i2,i3) + r(i1+1,i2,i3+1)$ xy12p = r(i1+1,i2-1,i3) + r(i1+1,i2,i3+1)$ xy12p = r(i1+1,i2-1,i3-1) + r(i1+1,i2-1,i3+1)$ xy12p = r(i1+1,i2-1,i3-1) + r(i1+1,i2+1,i3+1)$ xy12p = r(i1+1,i2-1,i3-1) + r(i1,i2-1,i3+1)$ xy12p = r(i1,i2,i3-1) + r(i1,i2-1,i3+1)$ x + r(i1,i2-1,i3-1) + r(i1,i2-1,i3+1)$ x + r(i1,i2-1,i3) + r(i1,i2-1,i3+1)$ x + r(i1,i2-1,i3+1)$ x + r(i1,i2-1,i3) + r(i1,i2-1,i3) + r(i1,i2-1,i3+1)$ x + r(i1,i2-1,i2-1,i3+1)$ x + r(i1-1,i2-1,i3+1)$ x + r$
enddo	

2018年5月17日 計算科学技術 特論B

67

K computer

ループrprj3のチューニング(2)

配列rをpaddingすることにより、アド	レスを明にずらす。
(rは1次元配列を3次元配列の引数で受	愛け取るため、様々なルーチンに手が入る)
$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$	$ \begin{array}{ c c c c c c c c c c c c c c c c c c c$
enddo	enddo
enddo	enddo



ループrprj3のチューニング(3)

	Before	After
L1Dミス率	12.56%	6.45%
L1Dミスdm率	41.62%	15.44%
メモリスループット	26.78G/s	41.33G/s
Peak性能	4.94%	10.96%



2018年5月17日 計算科学技術 特論B

69



まとめ

- ・理研で進めた性能最適化
- ・Seism3Dの性能最適化
- ・FrontFlow/blueの性能最適化
- ・NPB MGのチューニング事例



Hasegawa, Y., Iwata, J.I., Tsuji, M., Takahashi, D., Oshiyama, A., Minami, K., Boku, T., Shoji, F., Uno, A., Kurokawa, M., Inoue, H., Miyoshi, I., Yokokawa, M.:

"First-principles calculations of electron states of a silicon nanowire with 100,000 atoms on the K computer.", Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis. pp. 1:1–1:11. SC '11, ACM, New York, NY, USA (2011)

黒田 明義, 長谷川 幸弘, 寺井 優晃, 井上 俊介, 市川 真一, 小松 秀実, 大井 憲行, 安藤 琢也, 山崎 隆浩, 大野 隆央, 南 一生. :"ナノ材料第一原理分子動力学プログラムPHASE の京速コンピュータ「京」上の計算性能最適化", ハ イパフォーマンスコンピューティングと計算科学シンポジウム論文集, pp.144-152 (2012)

南 一生, 井上 俊介, 堤 重信, 前田 拓人, 長谷川 幸弘, 黒田 明義, 寺井 優晃, 横川 三津夫.:"「京」コン ピュータにおける疎行列とベクトル積の性能チューニングと性能評価"ハイパフォーマンスコンピューティング と計算科学シンポジウム論文集, pp.23-31 (2012)

Kiyoshi Kumahata, Shunsuke Inoue, Kazuo Minami.: Kernel performance improvement for the FEM-based fluid analysis code on the K computer, Procedia Computer Science, Volume 18, 2013, Pages 2496–2499, 2013 International Conference on Computational Science

井上俊介, 堤重信,前田拓人,南一生.:スーパーコンピュータ「京」におけるメモリインテンシブな アプリケーションの評価および高性能化,先進的計算基盤システムシンポジウム SACSIS2013論文集,pp,123(2013)

Masaaki Terai, Eiji Tomiyama, Hitoshi Murai, Kazuo Minami and Mitsuo Yokokawa.: "K-scope: a Java-based Fortran Source Code Analyzer with Graphical User Interface for Performance Improvement", Third International Workshop on Parallel Software Tools and Tool Infrastructures (PSTI2012).

南一生, 井上俊介, 千葉修一, 熊畑清, & 横川三津夫. (2016). キャッシュの効果を考慮したルーフラインモデルの拡張によるプロ グラムの性能予測. 情報処理学会論文誌コンピューティングシステム (ACS), 9(2), 1-14.

K-scope http://www.r-ccs.riken.jp/jp/k/software/

ebt-hpc/cca <u>https://github.com/ebt-hpc/cca</u>



71