QCD計算と宇宙流体力学計算の高速化

近藤　正章　　宮崎　高志

東京大学先端科学技術研究センター
中村研究室

QCDの概要

- QCD (Quantum ChromoDynamics: 量子色力学)
  - 4次元格子空間におけるグルオン場の計算
- 現行のCP-PACSにおけるQCDシミュレーション
  - 24^3 x 48の四次元格子空間(クエンチ近似)
- 将来的な計算要求
  - 48^3 x 96の四次元格子空間(フルQCD)
  - 16 GFLOPS x 4096 PU のマシンを仮定すると
    → 1 PUあたり6^3 x 12 の計算
- BiCGStabと呼ばれるiterativeアルゴリズムを解く

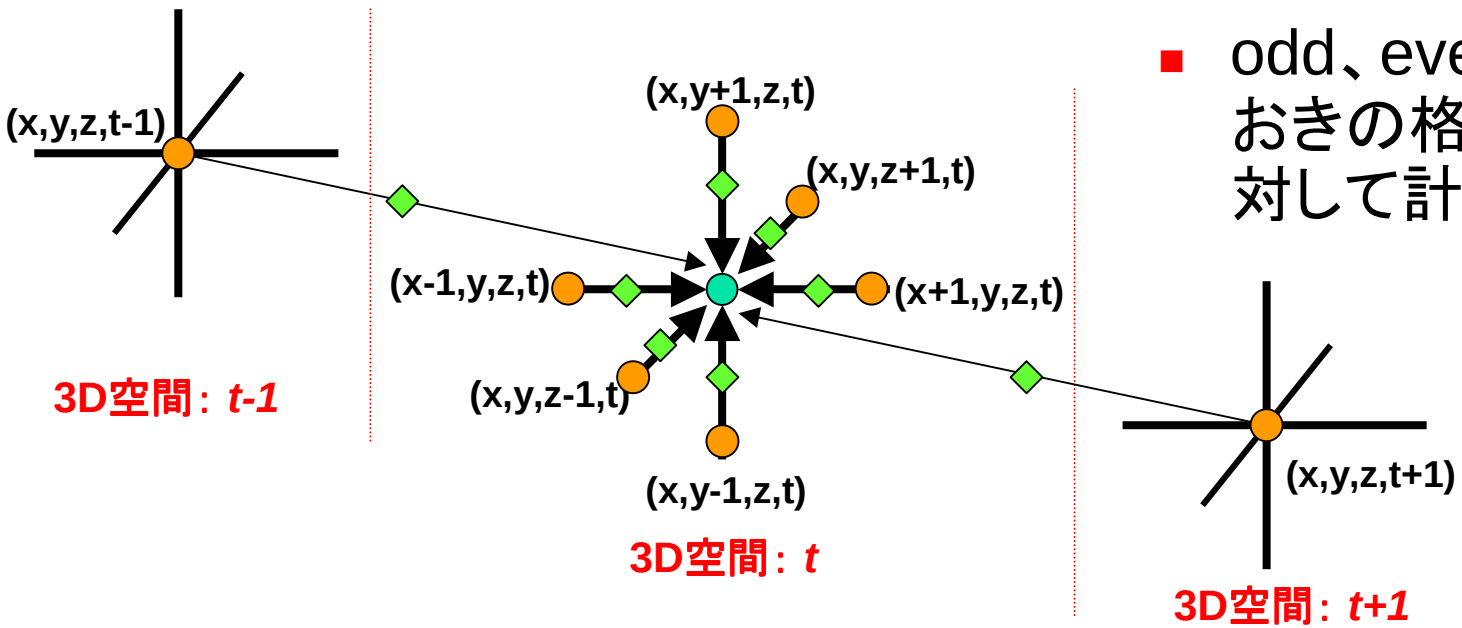
イテレーションループの構造

Operation	source data	=>	destination data
1 continue			
inter-MULT 1			
call RBMULT	U(1.5MB),	B_e(0.5MB)	=> G_o(0.25MB)
call localmult	M_o(1.5MB),	G_o(0.25MB)	=> G_o(0.25MB)
call RBMULT	U(1.5MB),	G_o(0.5MB)	=> V_e(0.25MB)
call localmult	M_e(1.5MB),	V_e(0.25MB)	=> V_e(0.25MB)
inter-MULT 2			
call RBMULT	U(1.5MB),	R_e(0.5MB)	=> G_o(0.25MB)
call localmult	M_o(1.5MB),	G_o(0.25MB)	=> G_o(0.25MB)
call RBMULT	U(1.5MB),	G_o(0.5MB)	=> T_e(0.25MB)
call localmult	M_e(1.5MB),	T_e(0.25MB)	=> T_e(0.25MB)
inter-MULT 3			
goto 1			

→ RBMULTが最も処理時間のかかるルーチン

データアクセスの概要

- RBMULTにおいてアクセスされる配列
  - G: クオークの自由度を表す(3×4複素行列) ➔ ●
    - odd/evenという1つおきの格子点ごとにアクセス
  - U:グルオンの自由度を表す(3×3複素行列) ➔ ◆



- odd、evenと呼ぶ1つおきの格子点空間に対して計算

アクセスされる配列の特徴

- G (R,B,V,T): 非常に再利用性が高い
- U: 4回 / iteration
- M: 2回 / iteration←(localmultルーチンでアクセスされる)
- 各配列のデータサイズ

	宣言	アクセス
G,R,B,V,T	各 1.36MB	最大1.0MB
U	6.8MB	2.5MB
M	3.0MB	3.0MB

- 再利用性: G >> U > M
- アクセスの時間間隔: G << U = M

キャッシュの問題点

- キャッシュのみでは:
  (G、U、Mがすべてキャッシュ経由でアクセス)
  - 再利用性の高いGなどの配列がキャッシュから追い出されてしまう

→ 同じreplacement algorithm がすべてのデータに対して適用されるため

↓

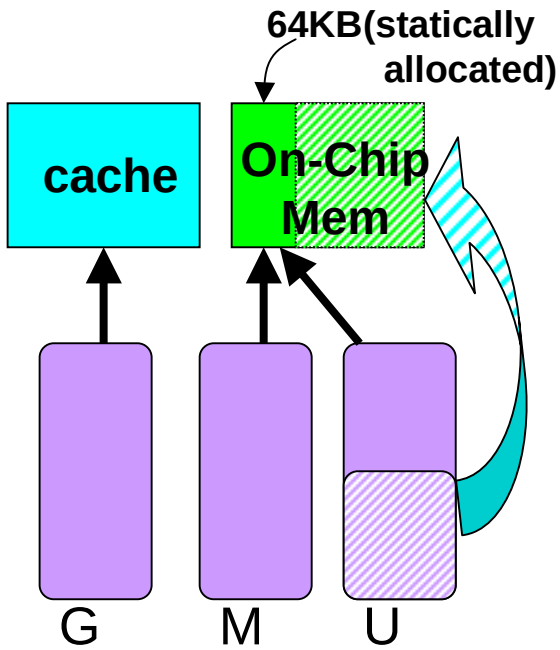
オンチップメモリを用いることで回避可能

性能評価

- ２種類の性能評価を示す
  - 比較的チップ内メモリの容量が大きい場合
    （チップ内メモリ容量：2MB）
    - RBMULTルーチンにおいて、ブロッキング必要なし
  - 小容量のチップ内メモリの場合
    （チップ内メモリ容量：32KB）
    - RBMULTルーチンにおいて、ブロッキング必要

オンチップメモリを用いた戦略

（2MBオンチップメモリ）



- 配列G (2.5MB)
  - 再利用性、時間的局所性高い
    →キャッシュ経由でアクセス
- 配列U (1.5MB), 配列M (3MB) :
  - Gとのコンフリクトを避けるため
    →オンチップメモリ経由でアクセス
  - 64KBのtemporary bufferを設ける
    (再内ループでのアクセスサイズが64KBであるため)
  - 残りのオンチップ領域にはできる限りUを固定して載せる
    (オンチップメモリサイズにより制限される)
  - 載りきらない配列Uについてはbuffer経由でアクセス

評価条件1

- 評価の仮定
  - parameters
    - registers: Int=32,FP=32
    - execution units: Int=4, FP(madd)=4, FP(div,sqrt)=1
    - load/store throughput: 4double precision words / cycle
    - multiply-add operation latency: 4cycle
    - load/store latency: 2cycle
    - Off-Chip Memory throughput: 1double precision word

    throughput
    on-chip:off-chip = 4:1
  - instruction cache: all hit
  - branch prediction: perfect
  - data cache structure: non-blocking L1 cache
  - out-of-order execution with reservation station

評価条件2

- キャッシュとオンチップメモリの仮定
  - reconfigurable On-Chip Memoryを用いる
    - 合計2MB, 4way set associativeのハードウェア

	cache size (assoc.)	On-Chip Mem size
cache	2MB (4way)	0MB
SCIMA-1	1.5MB (3way)	0.5MB
SCIMA-2	1MB (2way)	1MB

  - キャッシュラインサイズ: 32B or 64B
- オフチップメモリレーテンシ
  - 0 cycle, 10 cycle, 40 cycle

評価尺度

- 実行サイクル数のbreakdown
  - *Cnormal*: 合計サイクル
  - Cinf:　オフチップメモリスループット 無限大
  - Cperf: オフチップメモリスループット 無限大　&
    レーテンシ0 cycle

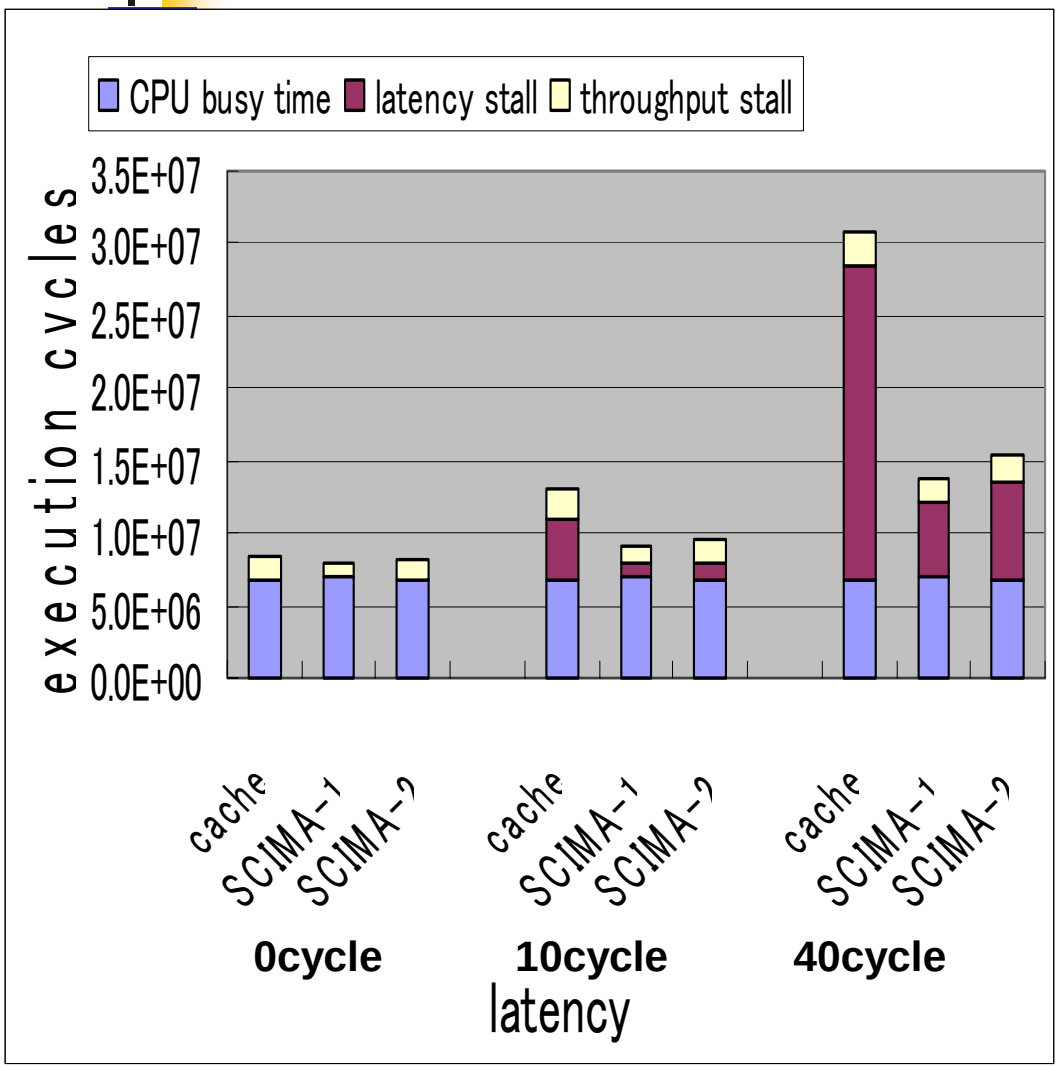
throughput-stall = Cnormal – Cinf

latency-stall = Cinf – Cperf

CPU-busytime = Cperf

- オフチップメモリトラフィック
  - キャッシュ/オンチップメモリとオフチップメモリ間の
    データ転送量

評価結果 (32B line)



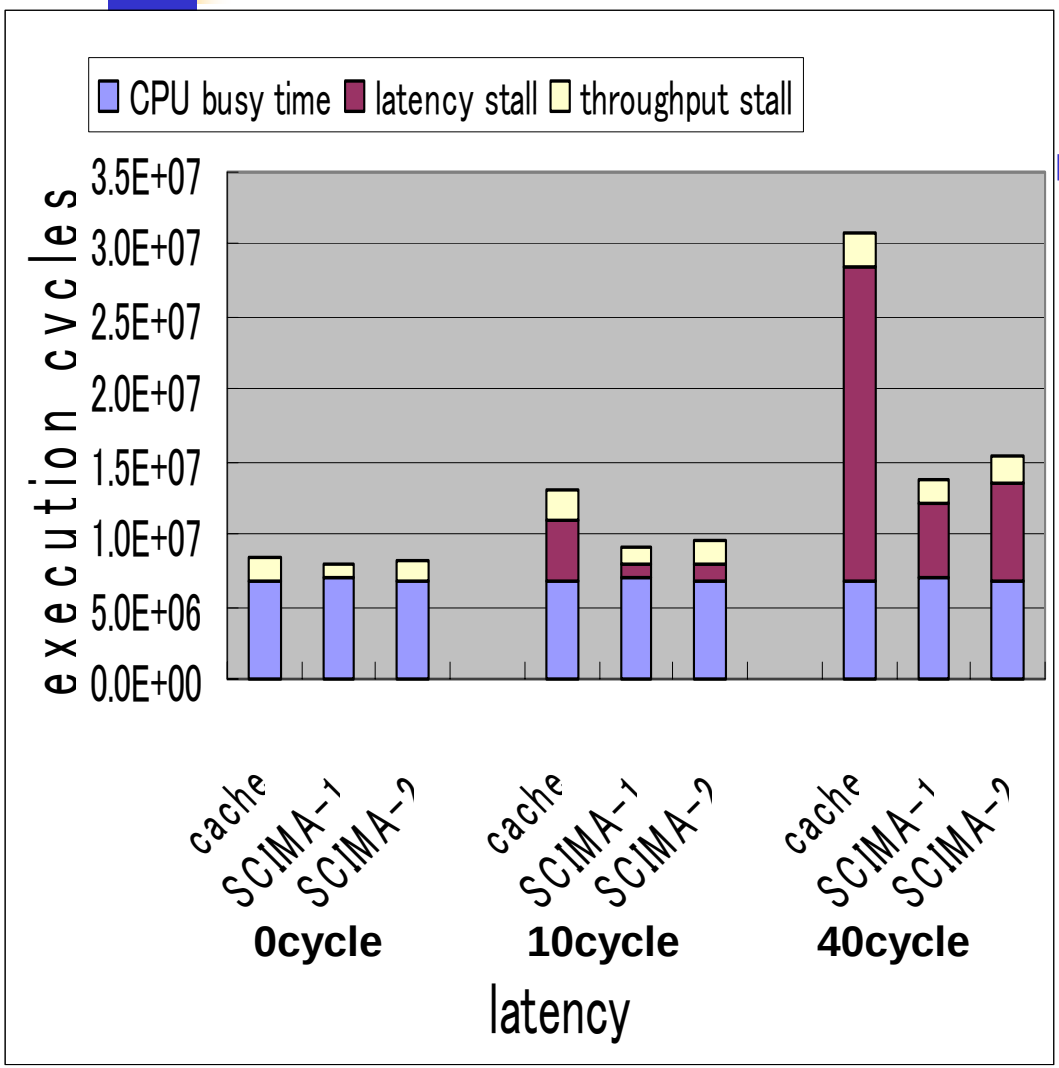
- レーテンシに関して
  - レーテンシ0cycleでは同じ性能
  - レーテンシ大ではSCIMA-1/2はcacheに比べ良い性能を達成

↓

データ転送サイズ:

cache line << page-load/page-store

評価結果 (32B line)



- トラフィックに関して
  - レーテンシ0cycleにおいてもSCIMA-1/2はcacheよりも高性能
  - SCIMA-1/2ではトラフィックが78%削減

↓

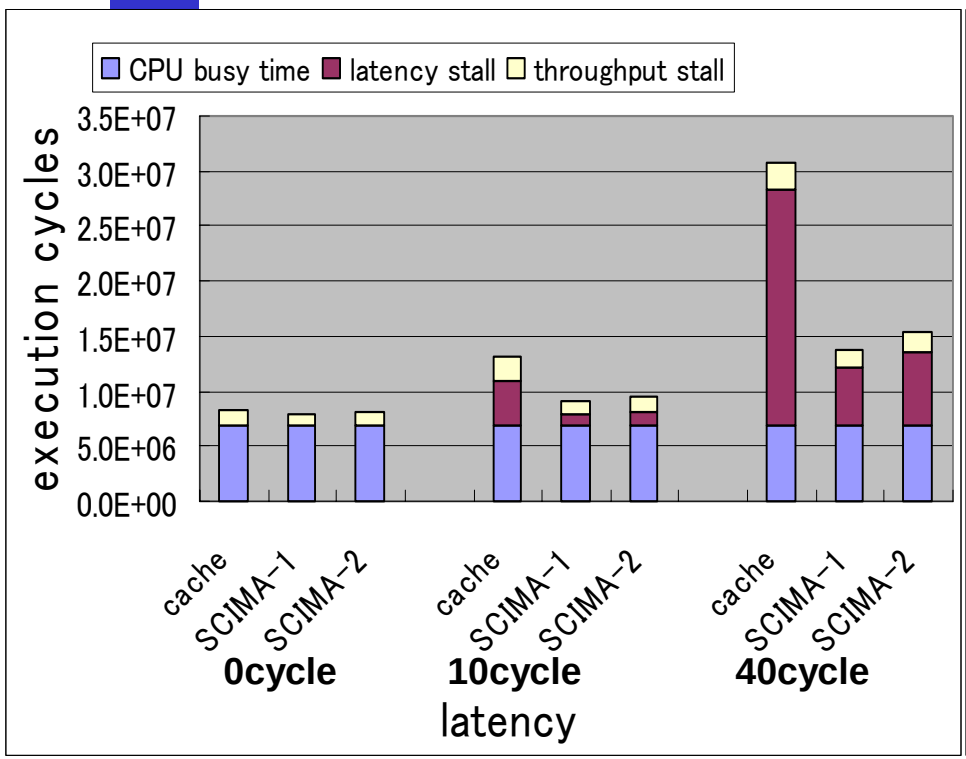
配列GとU/M間のコンフリクト削減の効果

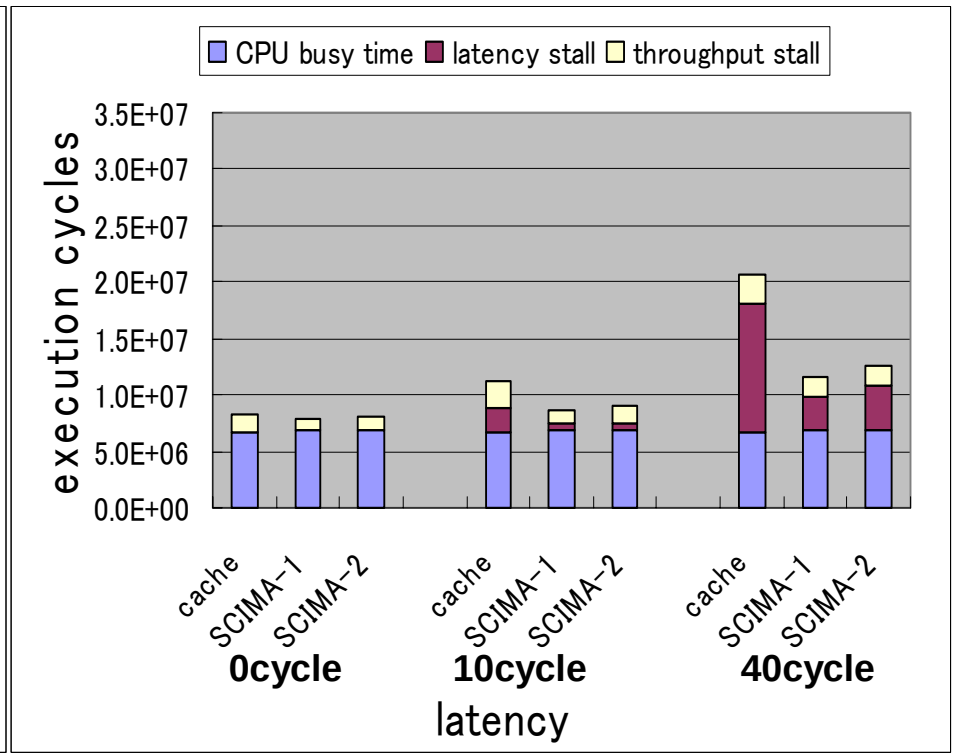
オフチップメモリトラフィック

	line size	cache	On-Chip Memory	total
cache	32B	18.6MB	0MB	18.6MB
	64B	20.4MB	0MB	20.4MB
SCIMA-1	32B	4.7MB	9.9MB	14.6MB
	64B	5.6MB	9.9MB	15.5MB
SCIMA-2	32B	6.1MB	8.2MB	14.3MB
	64B	7.5MB	8.2MB	15.7MB

- cacheに比べSCIMAのトラフィックは75%(64B/line)から78%(32B/line)も削減
- SCIMA-2 (1.0MB memory & 1.0MB cache)はSCIMA-1 (1.5MB memory & 0.5MB cache) よりわずかにトラフィック小

評価結果 (32B to 64B line)





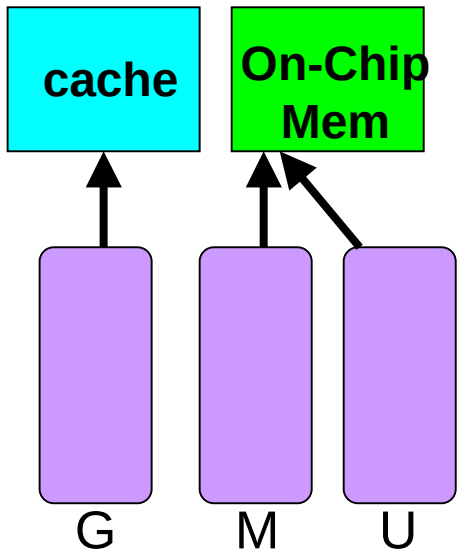
- 64B lineではレーテンシストールが減少
- スループットストール(トラフィック)が増加

→ 大きいキャッシュラインが常に良いとは限らない

オンチップメモリを用いた戦略

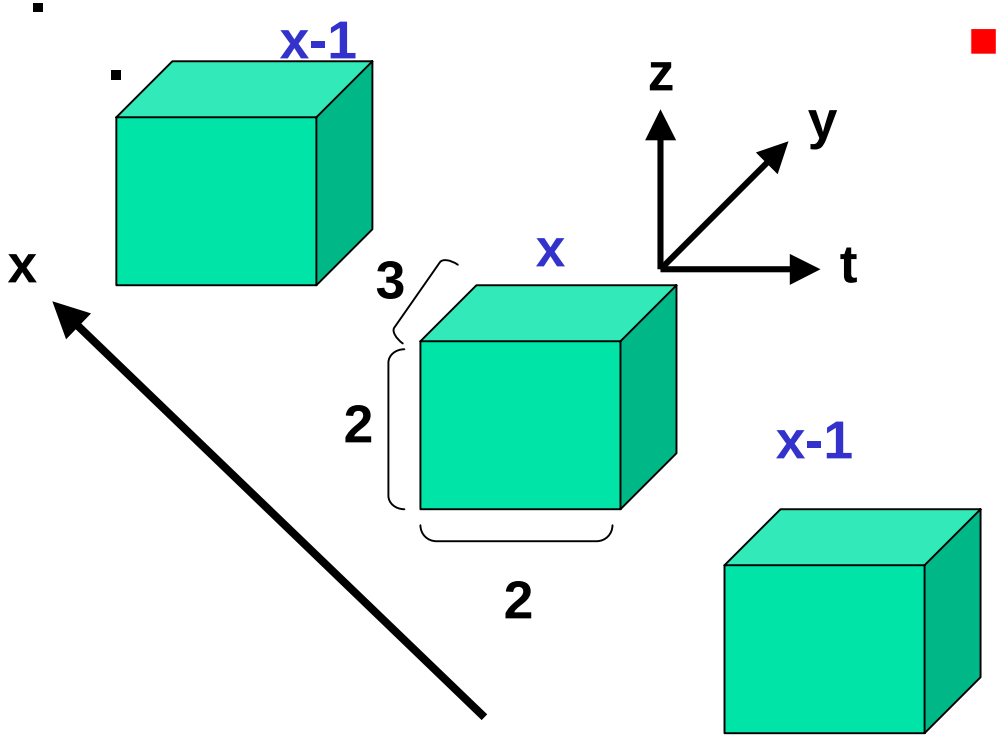
（32KBオンチップメモリ）

- 配列G (2.5MB)
  - 再利用性、時間的局所性高い
  - キャッシュブロッキングを行う
    → キャッシュ経由でアクセス
- 配列U (1.5MB), 配列M (3MB) :
  - Gとのコンフリクトを避けるため
    → オンチップメモリ経由でアクセス
  - 8KBのtemporary buffer
    (ブロッキング後の再内ループでのアクセスサイズが7KBであるため)
  - 常に再内ループで必要なUをオンチップメモリに転送



ブロッキング

- 配列Gなどのブロッキング
  - 3次元(y,z,t方向)のブロックをx方向に進める



- すべての(y,z,t)空間について行う

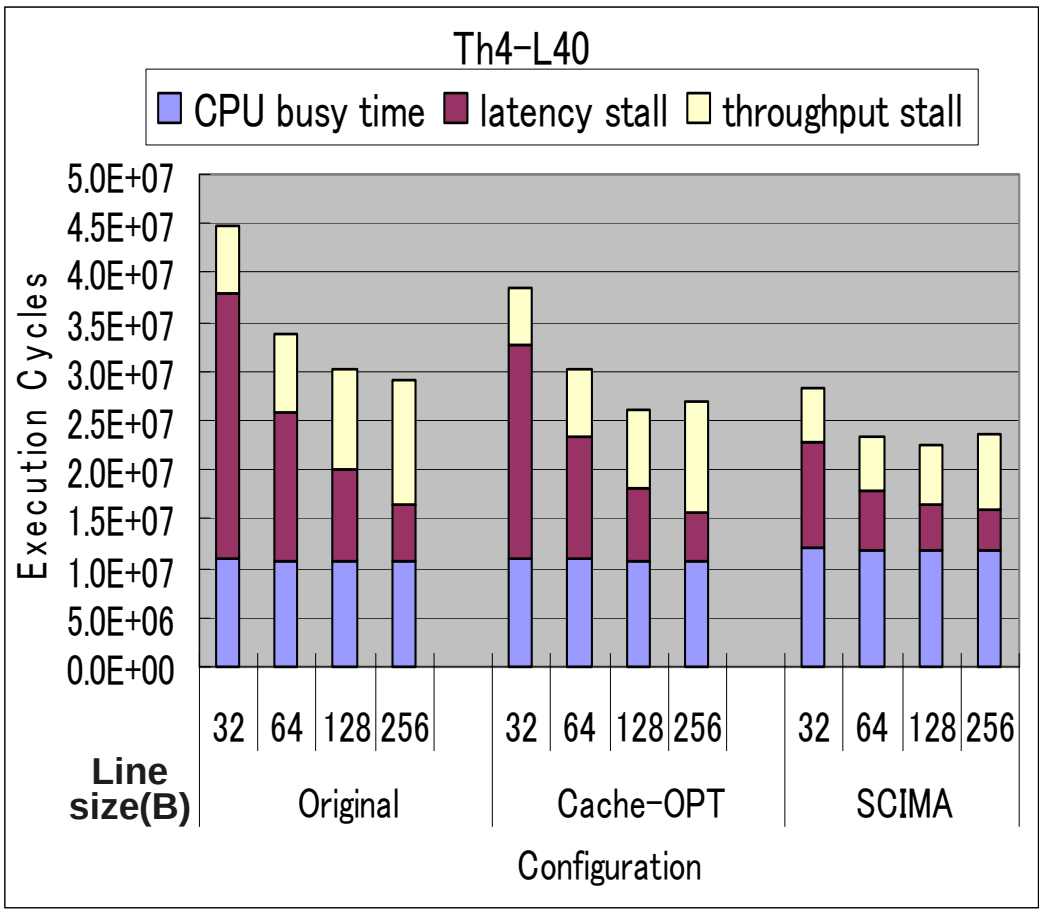
評価条件

- キャッシュとオンチップメモリの仮定
  - reconfigurable On-Chip Memory
    - 合計32KB, 4way set associativeのハードウェア

	cache size (assoc.)	On-Chip Mem size
original	32KB (4way)	0KB
Cache blocking	32KB (4way)	0KB
SCIMA	24KB (3way)	8KB

  - キャッシュラインサイズ: 32B、64B、128B、256B
- オフチップメモリレーテンシ
  - 40 cycle

評価結果



- 最も性能の良いもの同士を比べるとSCIMAは1.2倍ほどキャッシュに比べ高速
  - latency stall:37%削減
  - throughput stall: 25%削減
  - CPU busy timeが1割ほどSCIMAで多い

オフチップメモリトラフィック

（32B、128B）

	line size	cache	On-Chip Memory	total
original	32B	26.8MB	0MB	26.8MB
	128B	41.7MB	0MB	41.7MB
cache blocking	32B	21.8MB	0MB	21.8MB
	128B	33.2MB	0MB	33.2MB
SCIMA	32B	10.0MB	11.4MB	21.4MB
	128B	14.5MB	11.4MB	25.9MB

- 32Bラインではcache blockingとSCIMAで、それほど差がない
- 128BラインではSCIMAで22%ほどトラフィックが削減できる

QCDの評価結果の考察

- キャッシュとSCIMAの比較 (latency 40cycleのとき)
  - オンチップ大：キャッシュに比べ2倍の性能向上
  - オンチップ小：キャッシュに比べ1．2倍の性能向上

↓

- QCDではGなどの配列が載る程度のチップ内メモリ容量があることで良い性能が得られる
- 今後の課題
  - Gなどの配列をオンチップメモリに載せる（実験中）

→ 配列G同士のコンフリクトを防ぐことができ、更なるトラフィック削減が期待できる

宇宙流体力学計算(AFD)の概要1

- 非粘性圧縮性流体
  - 1次元オイラー方程式: $\frac{\partial \mathbf{Q}}{\partial t} + \frac{\partial \mathbf{E}}{\partial x} = 0$

$$\mathbf{Q} = \begin{bmatrix} \rho \\ m \\ e \end{bmatrix} \quad \mathbf{E} = \begin{bmatrix} m \\ m^2/\rho + p \\ m(e+p)/\rho \end{bmatrix}$$

ρ : 密度
m : 運動量
e : 全エネルギー

- 3次元オイラー方程式を差分法を用いて解く
- 適用する差分法はMUSCL内挿法によって2次精度化されたAUSMDV法

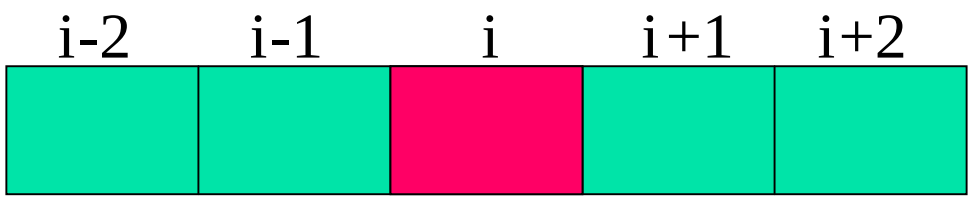
宇宙流体力学計算（AFD）の概要2

- 分ステップ法を用いて3次元問題に対処

x軸→y軸→z軸→z軸→y軸→x軸

First step　　Second step

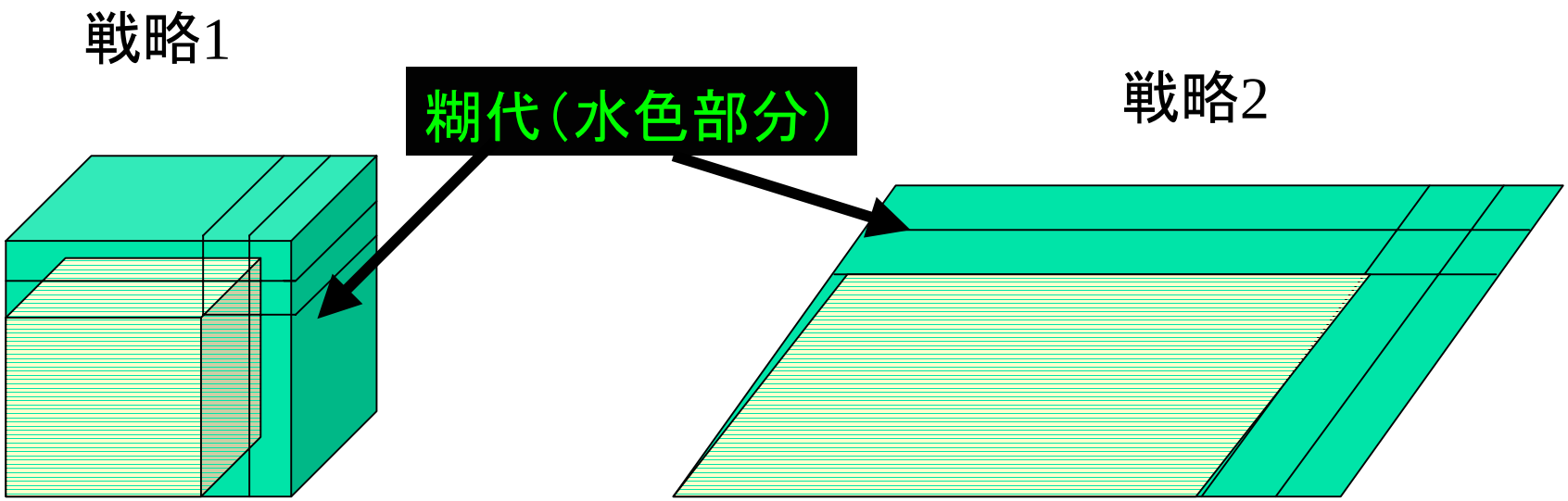
- 1つの要素を更新するために、自身と前後2つずつの要素が必要



高速化の戦略1

- キャッシュ、SCIMAの評価ともにブロッキングを適用する
- ブロックに区切るとブロックの境界部分に無駄(糊代)が生じる
- ブロックの区切り方によって2つの戦略
  - 戦略1:3次元ブロッキング
    → 3次元空間でのデータ再利用が可能
  - 戦略2:2次元ブロッキング
    → 平面で区切ることで糊代が減少

高速化の戦略2



Mem size	戦略1	戦略2
64KB	10x10x8	24x32
256KB	18x18x10	64x32

評価条件

- キャッシュとオンチップメモリの仮定
  - チップ内メモリサイズ：64KB、256KB
  - 4way set associativeのハードウェア

64KB

	cache (assoc.)	On-Chip
cache	64KB (4way)	0MB
戦略1	32KB (2way)	32KB
戦略2	32KB (2way)	32KB

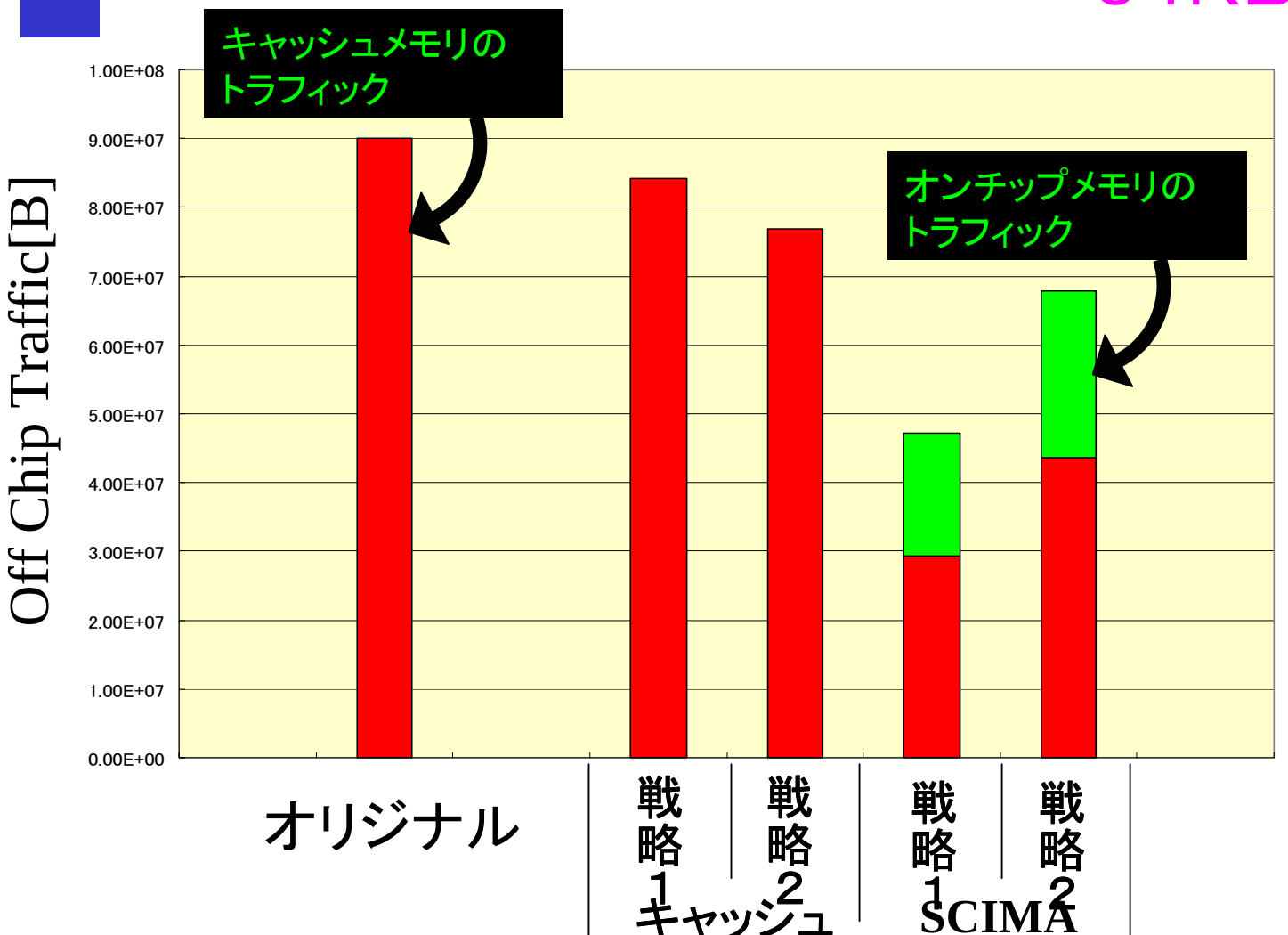
256KB

	cache (assoc.)	On-Chip
cache	256KB (4way)	0MB
戦略1	128KB (2way)	128KB
戦略2	128KB (2way)	128KB

  - キャッシュラインサイズ: 32B、64B、128B、256B
- オフチップメモリレーテンシ
  - 40 cycle
- 問題サイズ：64×32×32

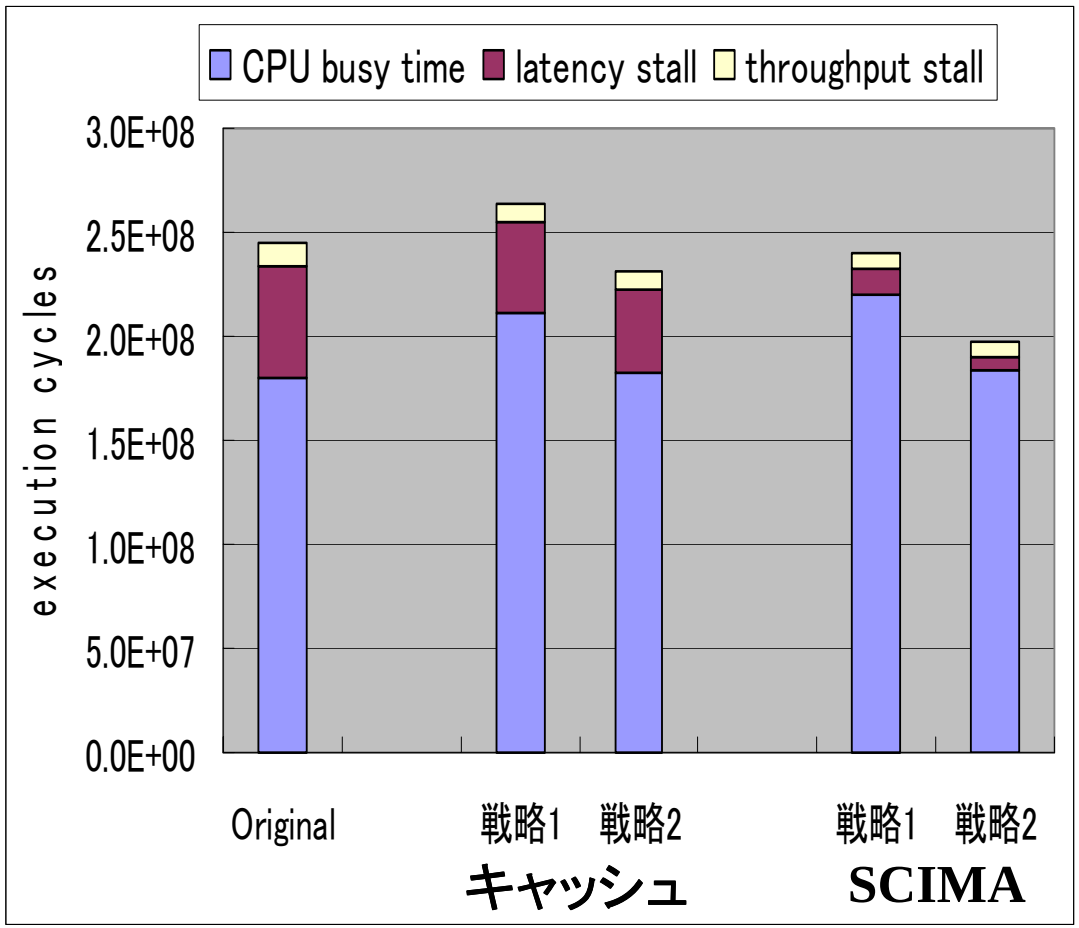
評価結果(トラフィック)

64KBメモリ



評価結果(実行サイクル数)

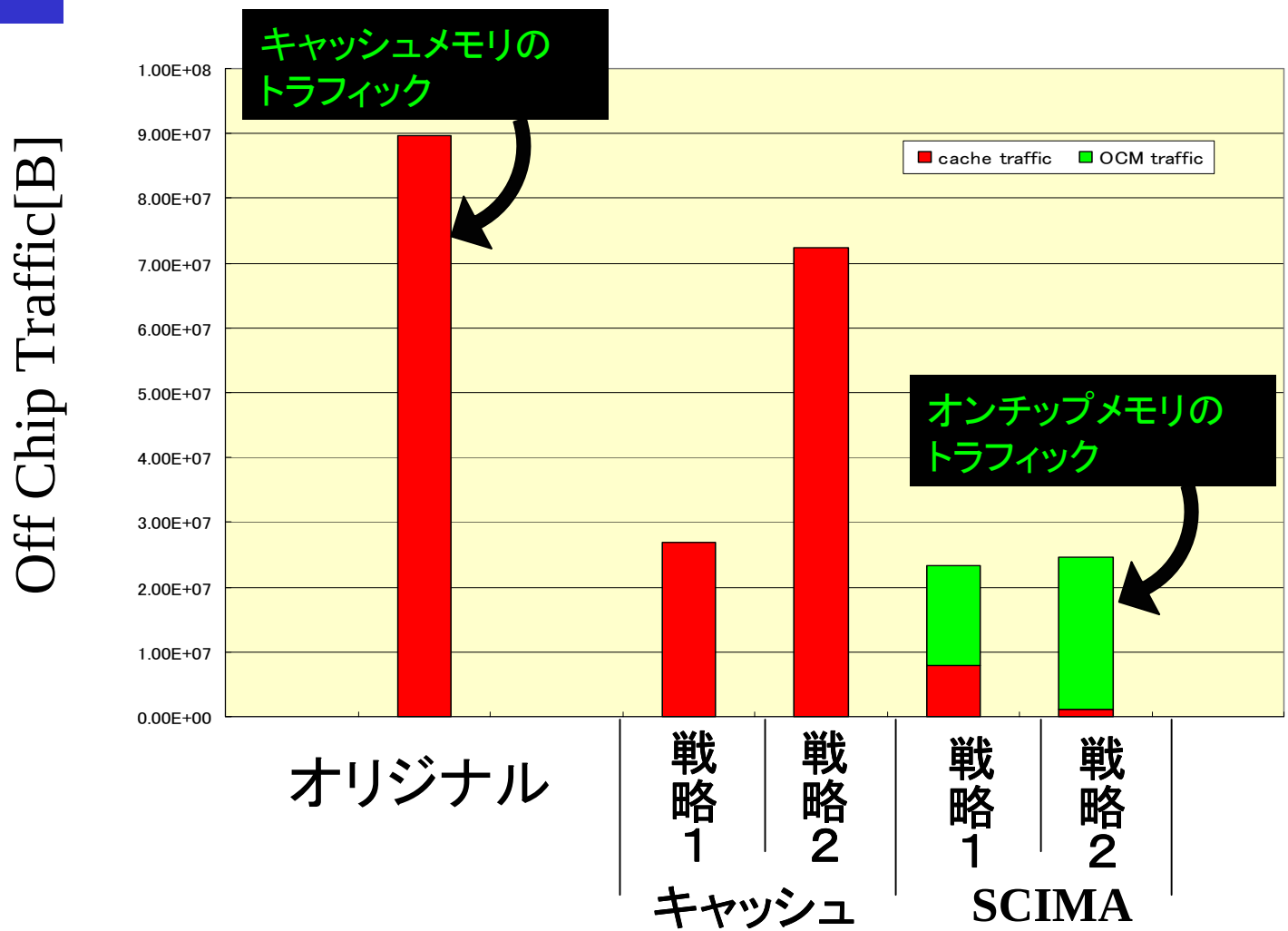
64KBメモリ



- ブロッキングによりCPU busy time増
  - ブロックの境界面での演算、コピー回数が増える
  - 戦略1は境界面が多いため戦略2に比べCPU busy time大
- メモリによるストール時間は減少する

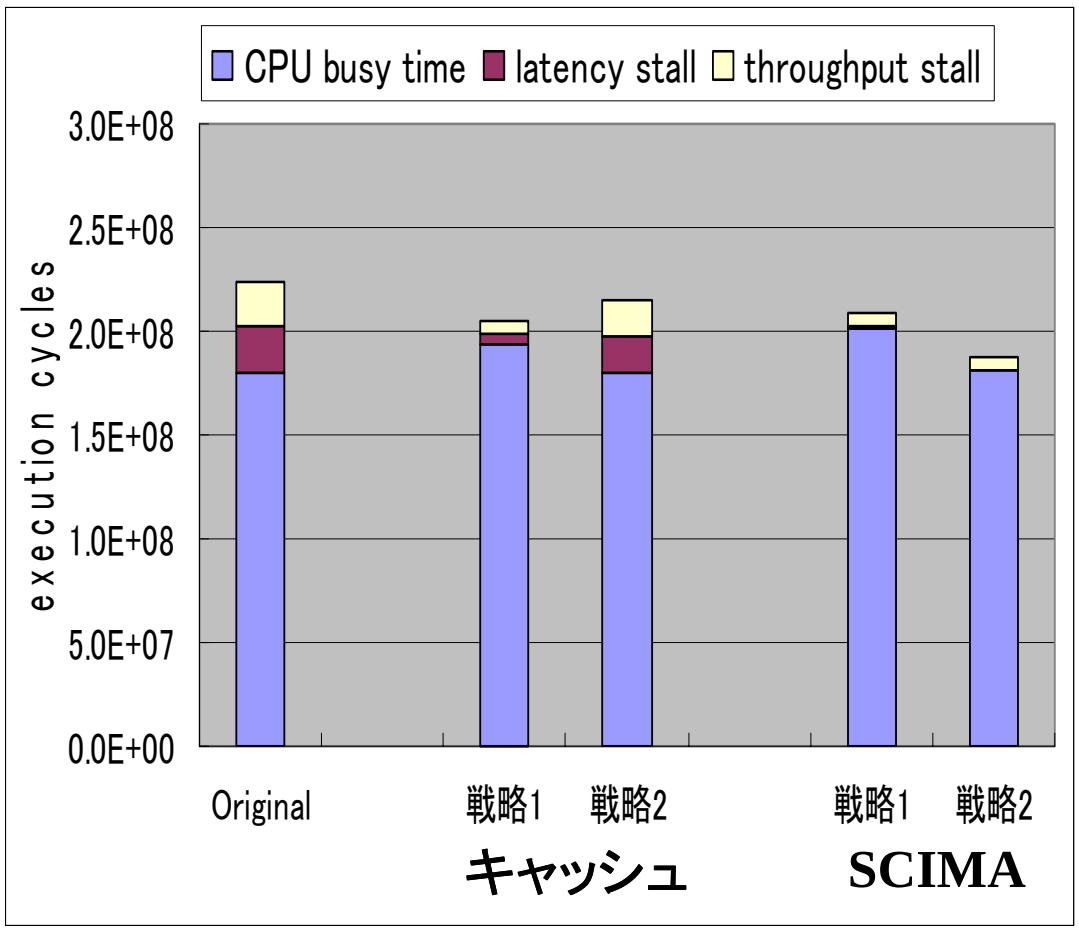
評価結果(トラフィック)

256KBメモリ



評価結果(実行サイクル数)

256KBメモリ



- ブロッキングによりCPU busy time増
  - ブロックの境界面での演算、コピー回数が増える
  - 戦略1は境界面が多いため戦略2に比べCPU busy time大
- メモリによるストール時間は減少する

AFD評価結果の考察

- 戦略１と戦略２の比較
  - オフチップメモリトラフィックは戦略１が少ない
    - 再利用性をより活用（３次元でブロッキングしているため）
  - 性能については戦略２が高速
    - 戦略１では糊代が多くなり演算量が増える → CPU busy time増加
- チップ内メモリ容量が大きくなると:
  - ブロックサイズを大きくできるため相対的に糊代の部分の割合が減少し、戦略１が高速となる可能性も考えられる
- 今後の課題
  - 更なる最適化の模索
  - CPU busy timeが長い理由の調査

まとめ

- QCD、およびAFDにおいてSCIMAはキャッシュに比べ高性能が得られる
- 将来的な半導体技術のトレンド:
  - 相対的に、オフチップメモリレーテンシ大
    オフチップメモリバンド幅　小

  - オフチップメモリトラフィックの削減
  - 大粒度なデータ転送

  が重要

- SCIMAは両者に解を与えることができるアーキテクチャである