プログレスレポート

森健一郎
配布対象：安藤先生

2020 年 4 月 28 日

1　研究目的

プロセッサ・チップ上には，ホット・スポットと呼ばれる単位面積あたりの電力が大きい場所が存在する．ホット・スポットは，そうでない場所と比べて温度上昇が激しいため，プロセッサの故障を引き起こす可能性が高い．[1–5] 従って，ホット・スポットを生成する回路の消費電力を低下させる必要がある．

ホット・スポットを生成する回路の 1 つに，発行キュー (IQ:issue queue) がある．IQ のサイズはプロセッサの世代が進むごとに大きくなっており，より深刻なホット・スポットとなっている．従って，IQ の電力削減に対する要求は非常に大きい．

IQ の中で最も電力を消費するのは，タグ比較の回路である．タグ比較は，発行幅分のディスティネーション・タグとすべてのソース・タグで行われるため，電力効率が非常に悪い．そこで本研究では，ディスティネーション・タグとソース・タグの下位ビットが等しい命令についてのみ比較器を動作させることにより，動作する比較器の数を減少させ電力を削減する方法を提案する．

提案手法は，次のように実現する．IQ を複数のセグメントに分割し，第 n セグメントには，第 1 ソース・タグの下位ビットが n である命令のみをディスパッチする．そして，ウェイクアップのタグ比較の際には，ディスティネーション・タグの下位ビットが，自身に割り当てられた命令の第 1 ソース・タグの下位ビットと等しいセグメントのみ，比較器を動作させてタグ比較を行う．この方法により，第 1 ソース・タグについての比較器の動作回数を「1/セグメント数」に減少させることができる．

提案手法における欠点として，セグメントが詰まることによる性能低下が挙げられる．あるセグメントに空きがない状態で，そのセグメントにディスパッチされる命令が現れた場合を考える．この場合，他のセグメントにディスパッチすることはできないため，該当のセグメントに空きが出るまでディスパッチを停止する必要があり，これは性能低下に繋がる．本研究では，この欠点に対する対応策を考え，性能低下が許容できる範囲内に収まるようにする必要がある．

また，その他の欠点として，第 2 ソース・タグの比較器の動作回数は削減できないことなどが挙げられる．これらの欠点に対しても十分に検討し，提案手法における電力削減及び性能の変化について評価を行う．

2　経過

2.1　前回の経過

- Last Tag Prediction(LTP) に GHB 方式を実装
- セグメントを第 2 ソースタグに基づき更に分割するサブ・セグメントを実装

2.2　今回の経過

よりタグ比較を削減することを重視した SWAP_AGGRESIVE と，より容量効率の低下を防ぐことを意識した SWAP_CONSERVATIVE 方式を適切に切り替えて使用する SWITCH 方式を実装

3 活動報告

3.1 ベンチマークの分類

今回のレポートの主題に入る前に，今後の説明のために SPEC CPU 2017 ベンチマークを特徴によって分類する．分類する種類としては，命令レベル並列性の高い high-ILP，メモリレベル並列性の高い high-MLP，そのどちらでもない others に分類する．

この分類を行うために，提案手法を使用しないモデルにおける， IPC と，L2 キャッシュの MPKI(L2C MPKI) の測定を行った．評価環境および評価結果を以下に示す．

表 1　プロセッサの基本構成

Pipeline width	8 instructions wide for each of fetch, decode, issue, and commit
Reorder buffer	320 entries
IQ	128 entries
Load/Store queue	128 entries
Physical registers	256(int) + 256(fp)
Branch prediction	12-bit history 4K-entry PHT gshare 2K-set 4-way BTB 10-cycle misprediction penalty
Function unit	4 iALU, 2 iMULT, 3 FPU, 2 LSU
L1 D-cache	32KB, 8-way, 64B line 2-cycle hit latency
L1 I-cache	32KB, 8-way, 64B line 2-cycle hit latency
L2 cache	2MB, 16-way, 64B line 12-cycle hit latency
Main memory	300-cycle latency 8B/cycle bandwidth
Prefetch	stream-based,32-stream tracked, 16-line distance, 2-line degree, prefetch to L2 cache

3.1.1 評価環境

評価環境について説明する．シミュレータには SimpleScalar をベースに修正を加えたものを使用した．表 1 にプロセッサ構成を示す．測定ベンチマークには，SPEC CPU 2017 ベンチマークのうち，int 系 9 本と fp 系 9 本の計 18 本を使用した．ベンチマークの測定区間は，プログラムの先頭から 16G 命令をスキップした後の 100M 命令である．

3.1.2 評価結果

評価結果を図 1，2 に示す．横軸は各ベンチマークおよびその平均，縦軸は IPC と L2C MPKI の値を示す．

■ILP による分類　図 1 をもとに，high-ILP に属するベンチマークを決定する．本研究では，IPC が 3.5 以上のベンチマークを high-ILP とする．

図より，high-ILP となるベンチマークは，xz，bwaves，cactuBSSN，cam4，imagick，roms の 6 つとなる．

■MLP による分類　図 2 をもとに，high-MLP に属するベンチマークを決定する．本研究では，L2C MPKI が 2 以上のベンチマークを high-MLP とする．

図より，high-MLP に分類されるベンチマークは，omnetpp，xalancbmk，lbm の 3 つとなる．

表 2　ベンチマークの分類

high-ILP	xz，bwaves，cactuBSSN， cam4，imagick，roms
high-MLP	omnetpp，xalancbmk，lbm
others	deepsjeng，exchange2，leela，mcf， perlbench，x264，fotonik3d，nab，pop2

■分類のまとめ　各ベンチマークを分類した結果を表 2 に示す．

なお，今回の分類に用いた IPC と L2C MPKI の基準値は，論文 [6] において ILP(MLP) が高いと分類されているベンチマークにおける IPC や L2C MPKI を基準に決定した．

3.2 提案手法による性能低下に関する考察

ベンチマークの ILP や MLP と，提案手法による性能低下との関係に関して考える．さらに，一部のベンチマークで提案手法によって性能が工場する理由についても考える．

3.2.1 性能低下に関する考察

図 3 に提案手法の SWAP_AGGRESIVE 方式における性能低下を示す．横軸はベンチマークを，縦軸は提案手法を使用しない通常の発行キュー (以下 BASE) に対する性能低下率を示す．図中の ILP および MLP のマークは，そのベンチマークが high-

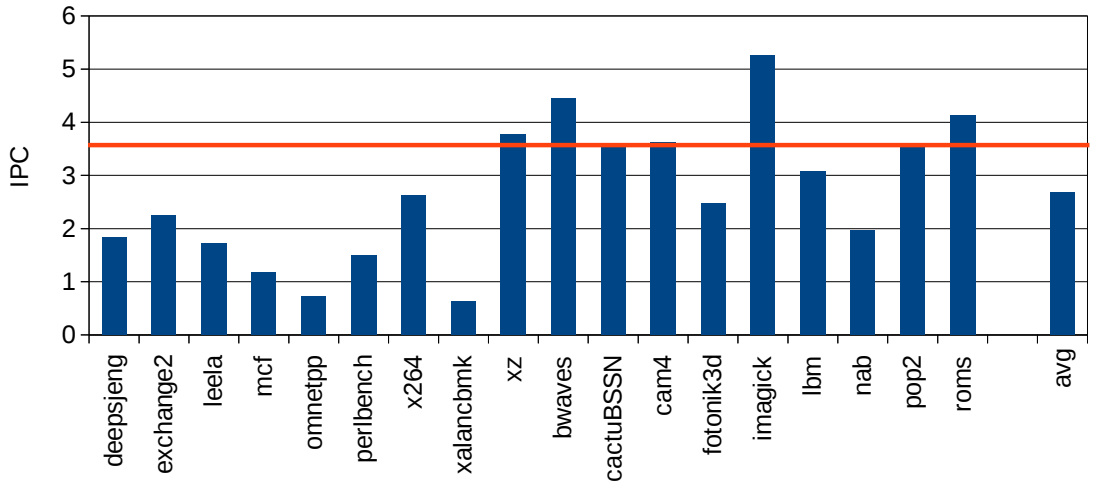

図 1　ベンチマークごとの IPC

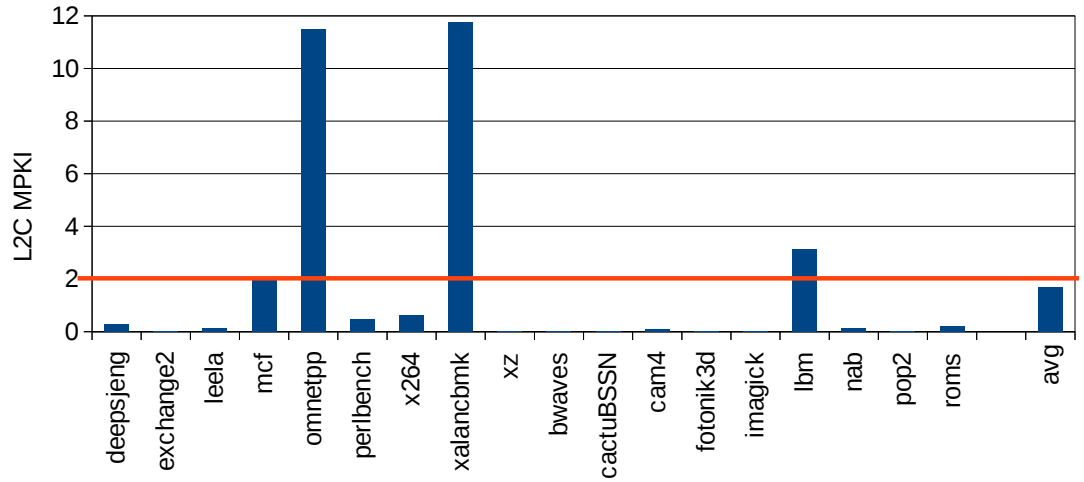

図 2　ベンチマークごとの L2C MPKI

ILP(MLP) に属していることを示す．セグメントの分割数が 16 の場合と 32 の場合に関して測定を行った．なお，測定環境は 3.1.1 節で示したものと同様である．

同図より，セグメント数が 32 の場合で性能が低下しているベンチマークをリストアップすると，omnetpp，x264，xalancbmk，xz，bwaves，cactuBSSN，imagick，lbm，roms となる．これらのベンチマークは，x264 を除いてすべて high-ILP か high-MLP に属するベンチマークである．したがって，ILP が高い，もしくは MLP が高いベンチマークにおいては，発行キューの容量効率が重要であり，提案手法により容量効率が低下することによって，性能が低下しているといえる．

3.2.2　性能向上に関する考察

ILP や MLP が高いベンチマークにおいて，提案手法により性能が低下しているのに対して，いくつかのベンチマークでは性能が向上している．図 3 を見ると，deepsjeng，exchange2，leela，mcf，cam4，pop2 などが大きく性能向上していることがわかる．この原因に関して考える．

提案手法により性能が向上するベンチマークの特

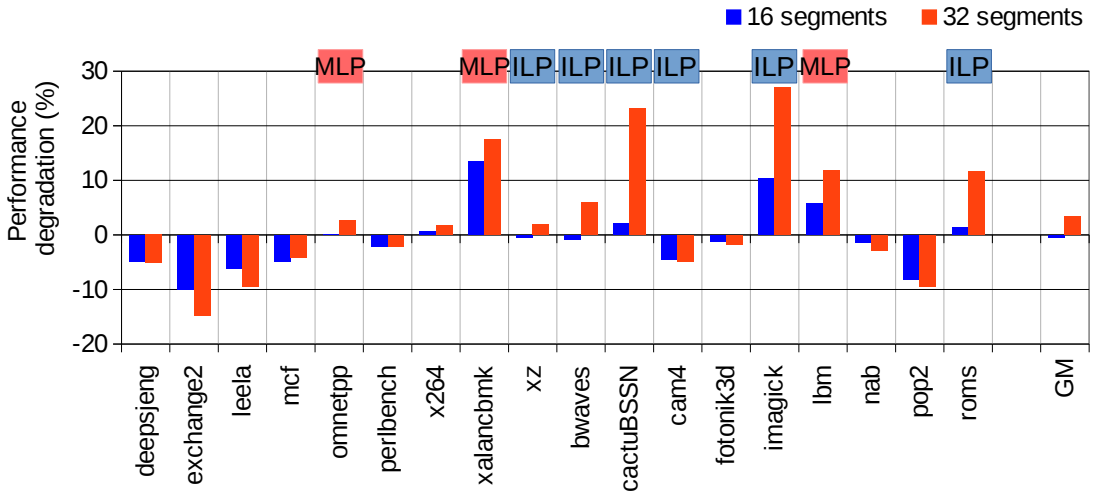

図 3　提案手法 (SWAP_AGGRESIVE) による性能低下

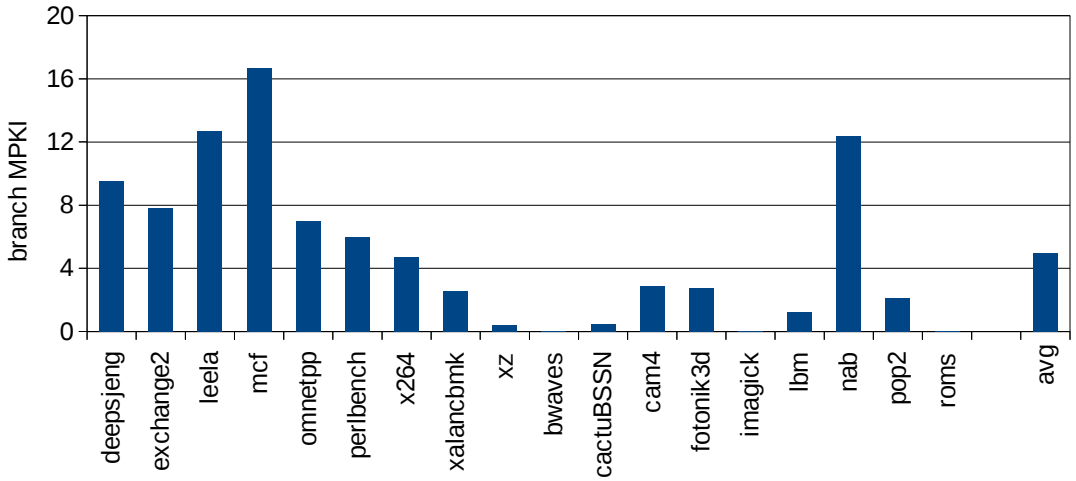

図 4　ベンチマークごとの分岐予測の MPKI

徴として，分岐予測ミスの発生頻度が関係していると考えられる．図 4 に，各ベンチマークの分岐予測 MPKI を示す．同図より，提案手法によって性能が向上しているベンチマークのうち deepsjeng，exchange2，leela，mcf では分岐予測ミスの発生頻度が高くなっていることがわかる．

分岐予測ミスの発生頻度が高いベンチマークにおいて，提案手法を用いた場合に性能が向上する理由としては，容量効率の低下によって，分岐予測ミスが判明するタイミングが早まっているためであると考えられる．ランダム・キューでは，分岐命令などの優先的に実行すべき命令が，ディスパッチされたエントリ位置の関係で発行優先度が低下してしまうことがある．このような場合には優先的に実行すべき命令の発行が遅れ，性能低下につながってしまう．

提案手法が原因で発行キューの容量効率が低下することにより，発行キュー内の命令数が減少し，結果として優先的に発行されるべき命令が従来よりも早期に発行され，結果として性能が向上すると考えられる．そして，分岐予測ミスが頻繁に発生する場合，分岐命令の優先度は非常に高くなるため，提案

手法における性能向上が大きくなっていると考えられる．

なお，cam4，pop2 に関しては，分岐予測ミスの発生頻度はそこまで高くないにもかかわらず，性能が大きく向上している．これに関しては，分岐命令以外の優先度の高い命令が優先的に発行されているためと考えられるが，確定はできない．

表 3　ベンチマークごとの特徴

benchmark	ILP	MLP	Br MPKI	Performance
deepsjeng			H	U
exchange2			H	U
leela			H	U
mcf			H	u
omnetpp		○		d
perlbench				u
x264				d
xalancbmk		○		D
xz	○			d
bwaves	○			d
cactuBSSN	○			D
cam4	○			u
fotonik3d				u
imagick	○			D
lbm		○		D
nab			H	u
pop2				U
roms	○			D

3.2.3　提案手法による性能変化に関するまとめ

提案手法による性能の変化について，表 3 にまとめる．ILP，MLP の ○ 印は，それぞれ high-ILP，high-MLP に属することを表す． Br MPKI は分岐予測ミスの発生頻度が多いことを示す． Performance の項目は提案手法による性能変化を表す．D が大きく性能が低下，d が小さく性能が低下， u が小さく性能が向上，U が大きく性能が向上していることを表す．

表より，これまで述べてきたように，ILP および MLP が高いベンチマークは性能が低下し，分岐予測ミスの発生頻度が高いベンチマークは性能が向上するという傾向が確認できる．

3.3　SWITCH 方式

ここまで述べてきたように，ILP や MLP が高い場合には発行キューの容量効率が必要で，そうでない場合には発行キューの容量効率は重要ではない．また，提案手法において，よりタグ比較を削減する SWAP_AGGRESIVE 方式 (以下 AGG) と，容量効率の低下を防ぐことをを重視した SWAP_CONSERVATIVE 方式 (以下 CONS) には，タグ比較の削減と容量効率の間にトレードオフの関係がある．

そこで， 実行中にプログラムの ILP や MLP を評価し，その評価に応じて AGG と CONS を切り替えて使用する SWITCH 方式を提案する．本方式では，ILP もしくは MLP が高い場合には CONS で実行し，そうでない場合には AGG で実行する．この手法により容量効率の低下による性能低下を防ぎつつ，容量効率が不要な時には積極的なタグ比較の削減が行えると期待できる．

本手法において重要となるのは，ILP 及び MLP の評価方法である．以下で詳しく説明する．

3.3.1　ILP の評価方法

ILP を実行中に評価する指標として，以下の 2 つが有効であると考えられる．

- IPC
- Issue Stalled Rate(ISR)

■IPC　IPC は最もわかりやすい ILP の評価指標であるといえる． IPC を用いた評価方法と方式の切り替えについて説明する．

1. プログラムを一定サイクルごとのインターバルに区切り，各インターバルにおいて期間内の IPC を測定する
2. 測定した IPC を設定したしきい値と比較し，しきい値よりも大きい場合には ILP が高いと評価する
3. ILP が高いと評価された場合には，次のインターバルを CONS で実行する．そうでなければ，AGG で実行する

なお， AGG と CONS の切り替えは，ディスパッチのアルゴリズムを変更するだけである．

IPC で評価する方法は，単純で直観的であるが，問題点もある．それは， AGG において性能が大き

く低下する場合，本来なら CONS へ切り替えを行う必要があるが，AGG での IPC で評価するため，切り替えが行えない可能性があるという点である．

たとえば，CONS で実行した場合 IPC が 3.8 で，AGG で実行した場合の IPC が 3.4 であり，切り替えのしきい値が 3.5 であるとする．現在 AGG で実行しているとすると，CONS で実行した場合と比較して 10% 以上性能が低下しているため，本来であれば次のインターバルで CONS に切り替えを行うべきである．しかし，現在の IPC 3.4 としきい値の 3.5 を比較すると，現在の IPC のほうが低いため，ILP が高いとは評価されず，CONS への切り替えは行われない．

このように，IPC を使用する方法では，AGG と CONS の IPC がしきい値付近である場合に，正しい切り替えが行えない可能性がある．

■Issue Stalled Rate(ISR)　ISR は，発行がストールする (発行命令数が 0 である) サイクルの割合を表す．発行がストールする場合は，発行キューに発行できる命令がないことを表すため ILP は低い．逆に発行がストールしない場合には，発行キューに発行できる命令が存在することを表すため ILP は高い．従って，ISR が高い場合には ILP は低く，ISR が低い場合には ILP が高いと評価できる．

ISR による ILP の評価方法と，方式の切り替えについて説明する．

1. プログラムを一定サイクルごとのインターバルに区切り，各インターバルにおいて発行がストールしたサイクル数を測定し，ISR を計算する
2. 測定した ISR を設定したしきい値と比較し，しきい値よりも低い場合には ILP が高いと評価する
3. ILP が高いと評価された場合には，次のインターバルを CONS で実行する．そうでなければ，AGG で実行する

なお，実際に ISR が ILP の良い評価値となるかに関しては，3.3.3 節で評価を行う．

3.3.2　MLP の評価方法

MLP を評価して切り替えを行う方法として，以下の 2 つが有効であると考えられる．

- L2C MPKI
- Dynamic Switching by Cache Miss(DSCM)

■L2C MPKI　L2C MPKI は，最もわかりやすい MLP の評価指標であるといえる．L2C MPKI を用いた評価方法と方式の切り替えについて説明する．

1. プログラムを一定サイクルごとのインターバルに区切り，各インターバルにおいて期間内の L2 キャッシュ・ミス数を測定し，MPKI を計算する
2. 測定した MPKI を設定したしきい値と比較し，しきい値よりも大きい場合には MLP が高いと評価する
3. MLP が高いと評価された場合には，次のインターバルを CONS で実行する．そうでなければ，AGG で実行する

■Dynamic Switching by Cache Miss(DSCM)　論文 [7] では，L2 キャッシュ・ミスの発生をきっかけに発行キューのサイズを拡大する手法を提案している．この論文によると，MLP が高くなる (すなわち発行キューの容量効率が重要となる) のは L2 キャッシュ・ミスが生じた直後である．

したがって，L2C ミスが発生したときに CONS に切り替え，一定期間が終了したら AGG に戻すという方法が有効であると考えられる．この方式を Dynamic Switching by Cache Miss(DSCM) と呼ぶこととする．

なお，今回はこの方法における一定期間をメモリ・アクセスのレイテンシ (300 cycle) とした．

3.3.3　ISR の有効性の検証

ISR が実際に ILP のよき評価値となるかを検証する．図 5 にベンチマークごとの発行命令数の分布を示す．横軸はベンチマークとその平均，縦軸は発行命令数の割合をパーセント表記で示している．最も下の青い部分が発行命令数が 0 の割合，すなわち

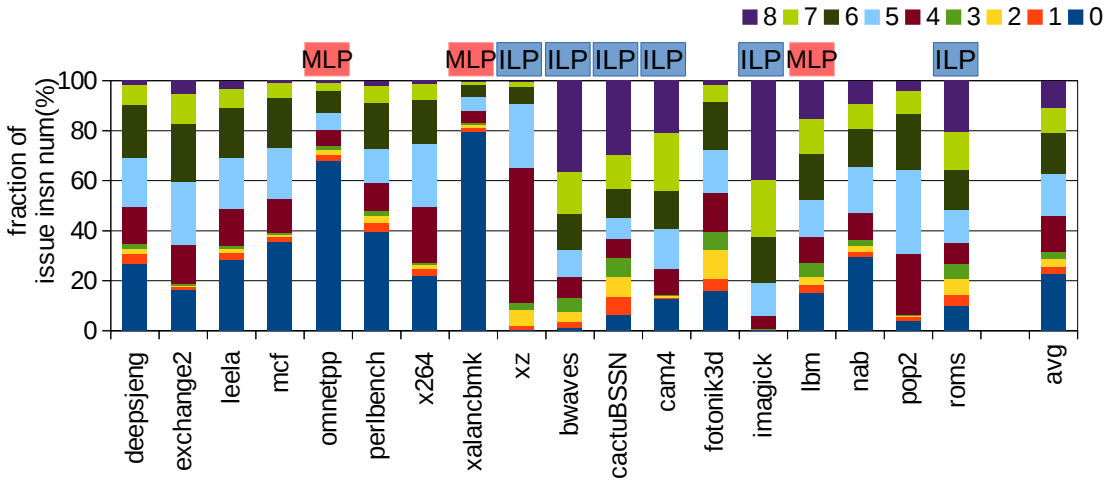

図 5　ベンチマークごとの 発行命令数の分布

発行がストールした割合を表す．

図より，ILP の高いとされるベンチマークは平均の値よりも発行がストールした割合が低いことがわかる．全体の平均が 22 % に対し， high-ILP のベンチマークでの平均は 5% となっている．したがって，ILP と ISR の間には相関があり， ISR を用いて ILP を評価できるといえる．

そのほかのベンチマークに関しても考察する．特徴的なのは，MLP が極めて高い omnetpp と xalancbmk において， ISR が非常に高くなっていることである．これは，キャッシュ・ミスによってプロセッサが停止し，結果として発行がストールするケースが頻繁に発生するためであると考えられる．

その他，deepsjeng や mcf といった分岐予測ミスの発生頻度が高いベンチマークでも発行がストールする割合が比較的高いといえる．これは，分岐予測ミスのリカバーが頻繁に発生するためであると考えられる．分岐予測ミスのリカバー時には，当然発行はストールする．

以上により，ISR は ILP の良い評価値となる可能性が高いと分かった．

3.3.4　提案手法による ISR の変動

ISR により ILP が評価できることはわかったが，これだけでは SWITCH 方式で有効であるとは言えない．AGG と CONS とで ISR の値が大きく変動してしまっては，IPC の時に述べたのと同様の問題が生じて，AGG から CONS へ切り替えるべきタイミングで適切な切り替えが行えない可能性があるためである．

そこで，提案手法の AGG および CONS を適応した際の ISR の測定を行った．測定結果を図 6 に示す．横軸はベンチマークとその平均，縦軸は ISR を表す．図より，全体的に AGG は BASE や CONS よりも 5%p 弱ほど ISR が増加する傾向にあることがわかる．これは容量効率の低下により利用できる ILP が低下しているためであると考えられる．

ただし，IPC では複数のベンチマークにおいて CONS から AGG に変更すると， 10% 以上の性能低下が見られていたのに対して，ISR は 5%p 程度の低下にとどまっている．また， high-ILP に属するベンチマークの AGG での ISR の平均が 10% 程度であるのに対して，high-ILP 以外の便意マークの平均は 35% 程度となっており，high-ILP とそうでないベンチマークとでの ISR の差は大きいことがわかる．

したがって， ISR は IPC よりも提案手法の方式による影響 (ブレ) が小さい評価指標であるといえる．つまり， high-ILP の ISR とそれ以外のベン

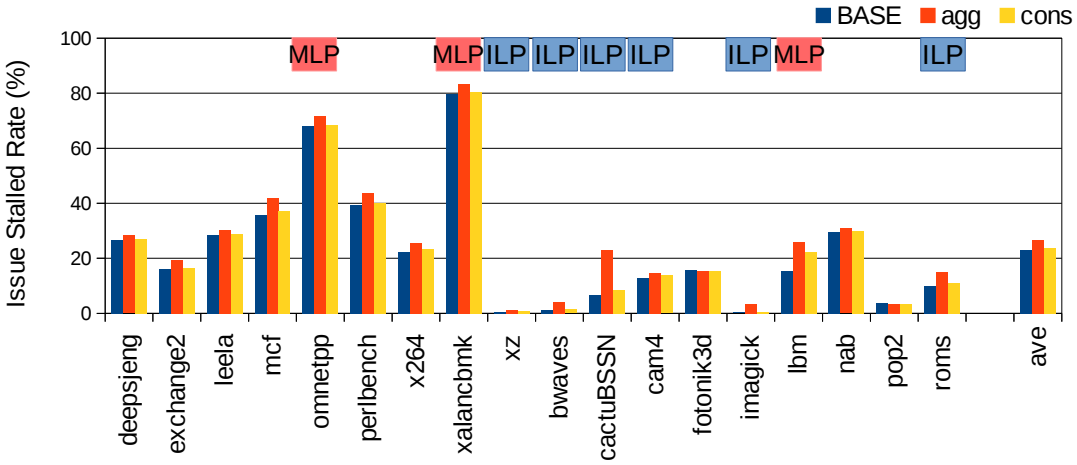

図 6　提案手法による ISR の変化

チマークの ISR との間に，適切なしきい値を設定できれば，適切なタイミングでの切り替えが行えると考えられる．

唯一， cactuBSSN のみは AGG と CONS で 10%p 以上の差が見られるため， SWITCH 方式において適切な切り替えが行えない可能性がある．

3.3.5　SWITCH 方式の実装と評価

SWITCH 方式をシミュレータに実装し，評価を行った．

ILP を評価して切り替える方法として

- IPC
- Issue Stalled Rate(ISR)

を使用する方法に関してそれぞれ評価を行う．その後， MLP を評価して切り替える方法として

- L2C　 MPKI
- Dynamic Switching by Cache Miss(DSCM)

を使用する方法に関してもそれぞれ評価を行う．

測定時のプロセッサ構成は 3.1.1 節で述べたものと同様である．また，セグメントの分割数は 32 に固定している． また，IPC や ISR L2C MPKI を測定するインターバルの期間はは 10K cycle としている．

ここで，SWITCH 方式の評価基準に関して再確認しておく．SWITCH 方式は

- 性能低下を最小限に抑え
- かつ，タグ比較の削減が AGG に近い

場合に有効性が高いと評価される．

3.4　SWITCH 方式の評価：IPC による切り替え

まず，IPC による切り替えを行う SWITCH 方式に関して評価を行う．図 7,8,9 に測定結果を示す．

図 7 は性能変化を表すグラフである．横軸はベンチマークとその平均を，縦軸が BASE に対する性能低下率を示す．各凡例は CONS が CONS 方式で実行したモデルを， Thre = x は SWITCH 方式において CONS と AGG を切り替える IPC のしきい値を x としたモデルを表す．AGG は AGG モデルで実行した方式を表す．

図 8 は提案手法によるタグ比較の削減率を表すグラフである．横軸はベンチマークとその平均，縦軸がタグ比較の削減率を示す．各ラベルは性能変化のグラフと同様である．

図 9 は SWITCH 方式において AGG モードであったサイクルの割合を表すグラフである．この値が 100 % の場合，すべてのサイクルが AGG モードで実行されたことを表し， 0% の場合はすべてのサイクルが CONS モードで実行されたことを表す． (本来は AGG と CONS の割合を積み上げグ

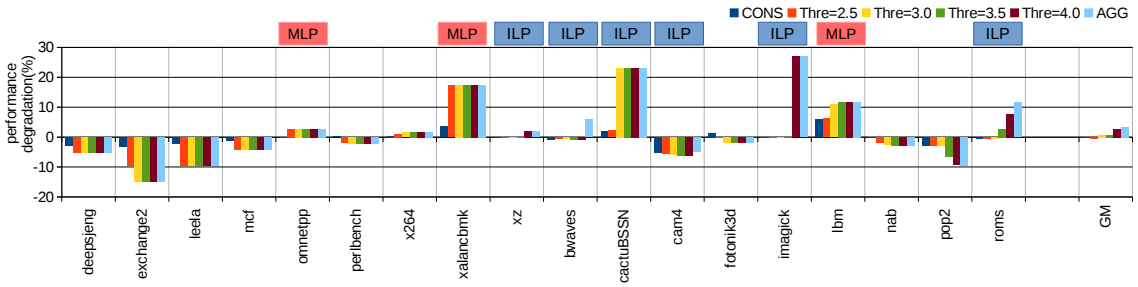

図 7　SWITCH 方式による性能変化 (IPC を用いた方式)

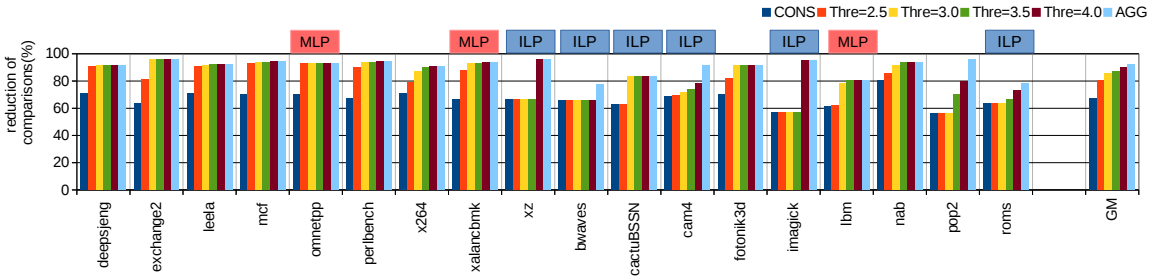

図 8　SWITCH 方式によるタグ比較の削減 (IPC を用いた方式)

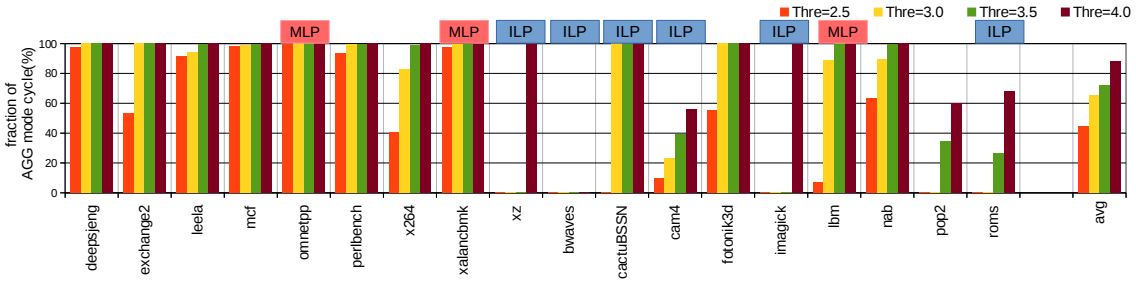

図 9　AGG モードである割合 (IPC を用いた方式)

ラフで表現するのが望ましいが， calc の表示がおかしく上手くグラフを作れなかったため，このようなグラフの形式となっています)

■AGG モードの割合に関する考察　IPC による切り替えは ILP を評価して行うものであるため，評価は high-MLP に属するベンチマークを中心に行う．図 9 を見ると，どのベンチマークでもしきい値が増加するに連れて AGG の割合が大きくなっていることがわかる．これは，しきい値が大きくなるにつれて，ILP が高いと評価する IPC の条件が厳しくなっているためである．

特に注目すべきであるのは， xz や cactuBSSN，imagick である．これらのベンチマークでは，しきい値がある値を超えると AGG の割合が急激に増加 (ほぼ 0% から 100%) している．今回の測定では全体的な性質を知るために，IPC のしきい値は大きな間隔で変化させてあるため，このような結果となったと考えられる．

また，図より，IPC を使用する方式の欠点として，ベンチマークによって適切な IPC のしきい値が異なるという点が上げられる．例えば， xz や imagick は しきい値が 3.5 から 4.0 に上がる際に

AGG の割合が急激に増加している．したがって適切なしきい値は 3.5 と 4.0 の間にあると考えられる．

その一方で cactuBSSN では，しきい値が 2.5 から 3.0 に上がる際に AGG の割合が急激に増加している．したがって，適切なしきい値は 2.5 と 3.5 の間にあると考えられる．

このように，ベンチマークによって適切であるしきい値が異なるため，単純に固定のしきい値を用いるだけではうまく制御できないと考えられる．

■性能変化に関する評価　今回の測定では high-ILP のベンチマークにおいて，AGG の割合が 0% か 100% となってているため，性能は CONS で実行した場合もしくは AGG で実行した場合と同じとなっていることが多い．細かなしきい値の調整を行い，適切なバランスの場合を評価する必要がある．

ここで，high-ILP のなかで roms に関して考える．roms は，しきい値が 3.5 の場合で AGG の割合が 20% 程度，しきい値が 4.0 の場合で 60% 程度となっている．この場合について性能低下を見てみると，しきい値が 3.5 の場合で 2% 程度，しきい値が 4.0 の場合で 7% 程度性能が低下していることがわかる．

これは，AGG での実行により性能が低下していることを示す．つまり，本来は CONS で実行すべきタイミングで切り替えがうまく行われず AGG にとどまっているため性能が低下してしまっている．したがって，3.3.1 節で述べた IPC で ILP を評価する際の欠点が現れている．

■タグ比較の削減に関する評価　提案手法のタグ比較の削減は， AGG の割合が大きいほど多く，AGG の割合が小さいほど少なくなっている．これは AGG がよりタグ比較を削減できるという性質によるものである．

■IPC による切り替えのまとめ　IPC による切り替えには，以下の 2 点の問題があることが分かった

- ベンチマークによって適切なしきい値が異なる
- AGG では CONS よりも性能が低下するため，本来 CONS に切り替えるべき状況でも AGG にとどまってしまう場合がある

IPC を ILP の評価指標として用いる場合には，この 2 点の欠点に対応する必要がある．

3.5　SWITCH 方式の評価：ISR による切り替え

Issue Stalled Rate(ISR) を使用して切り替えを行う方式について評価を行う．測定結果を図 10, 11 に示す．図の形式は IPC を用いた方式と同じである．なお，タグ比較削減率のグラフに関しては，IPC を用いた方式の時と同様で，図 9 の AGG と CONS の比によって決まるものであるため省略している (AGG の割合が高いほど削減率が大きい)．

ラベルの Thre = x は，ILP が高いと判定する ISR のしきい値として x(%) を用いたことを表す．ISR は発行がストールした割合を表す評価値であるため， ISR がしきい値以下の場合に， ILP が高いと判断される．

■AGG モードである割合に関する考察　high-ILP に属する命令に関してみてみると， xz，bwaves，imagick はどのしきい値においてもほぼ 100% CONS の状態で実行されていることがわかる．これらのベンチマークは，ISR が極端に低いため，常に 発行キューの容量効率が重要であると判断されていると考えられる．

AGG で実行した際の性能低下が大きい roms に関しては，しきい値が 15% もしくは 10% で実行すれば，性能低下を十分に抑えつつ (3% 程度)，過半数のサイクルを CONS モードで実行できるため，有効な切り替えができているといえる．

これらのことから，ISR を用いた場合，しきい値の値を 15% ～ 10% の間に設定すれば，high-ILP に属するベンチマークにおいて性能低下を抑制しつつ，可能であれば AGG で積極的なタグ比較の削減が行えることがわかった．したがって，ILP を評価する評価値として，ISR は IPC よりも有効性が高いと判断できる．

唯一，cactuBSSN に関しては，ISR のしきい値を 20% 程度まで上げなければ性能低下を引き起こしてしまう．これは，図 6 で示した通り，cactuBSSN のみ AGG と CONS で ISR の差が 10%p 程度あ

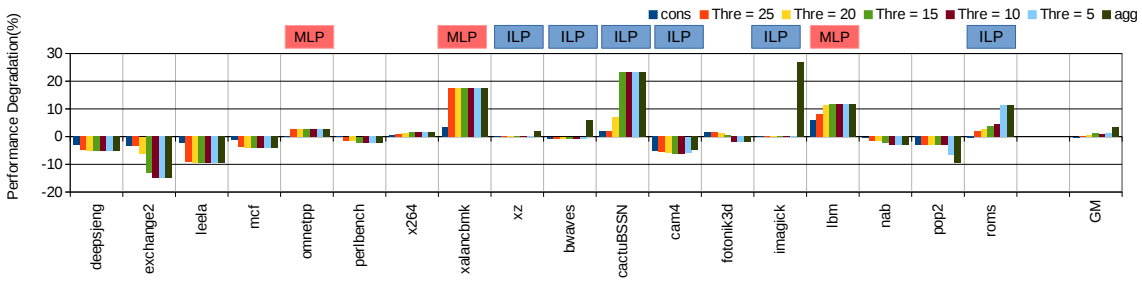

図 10　SWITCH 方式による性能変化 (ISR を用いた方式)

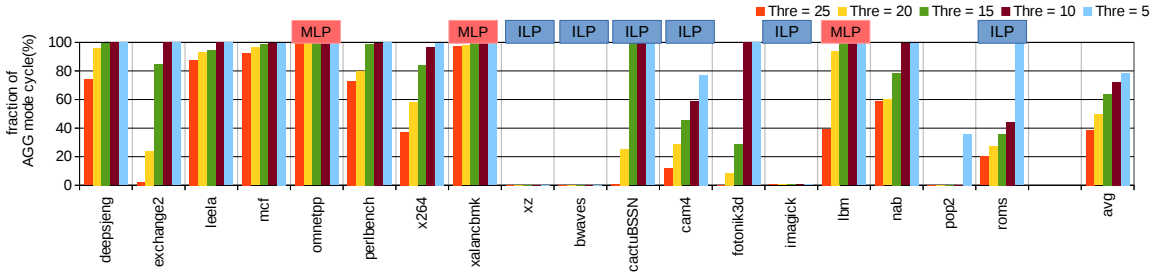

図 11　AGG モードである割合 (ISR を用いた方式)

るためである．すなわち， cactuBSSN では AGG モードにした際に ISR が大きく上昇してしまうため，CONS へ切り替える必要がある際に正しく切り替えが行えないという現象が生じている．これに関しては，今後の課題であり，解決する必要がある．

なお，cactuBSSN に合わせて ISR のしきい値を 20% 程度とするという方法も考えらる．しかしこの方法では，本来 AGG で実行すべきであるベンチマーク (deepsjeng や x264) において AGG の割合が低下してしまうという現象が生じるため，好ましくないといえる．

3.6　SWITCH 方式の評価：L2C MPKI による切り替えと DSCM

MLP を評価して切り替えを行う方法として，L2C MPKI にもとづき切り替える方法と，キャッシュ・ミスが発生した時に一定期間切り替える DSMC に関して評価を行った．評価結果を図 12, 13 に示す．図の形式はこれまでと同様である．ラベルの Thre = x は， L2C MPKI によって切り替える方式において MLP が高いと判断するしきい値を x とした場合を表す．

まず，L2C MPKI によって切り替えを行う方法について考える．図 13 より， high-MLP のベンチマークにおいては，AGG の割合が他よりも低いことがわかる．これは， MLP が高い場合に CONS モードで実行できていることを表す．その結果として，high-MLP のベンチマークでの性能低下が抑えられていることが，図 12 より確認できる．

次に，しきい値の大きさを変化させた場合について考える．MPKI のしきい値が大きことは， MLP が高いと評価する基準が厳しくなることを意味する．これは，図 13 においてしきい値が大きいほど AGG の割合が高くなっているという結果と一致する．つまり，しきい値が大きいほど， MLP が高いと判断される可能性が低くなるため，結果として AGG で実行する割合が増加する．

しきい値が 6 の場合は， high-MLP のいずれのベンチマークにおいても性能低下は 5% 以下に抑えられている．その一方で，しきい値が 8 になると，xalancbmk において性能低下が 10% 弱と高い値を示している．したがって，現在の評価環境においては L2C MPKI のしきい値は 6 前後が妥当である

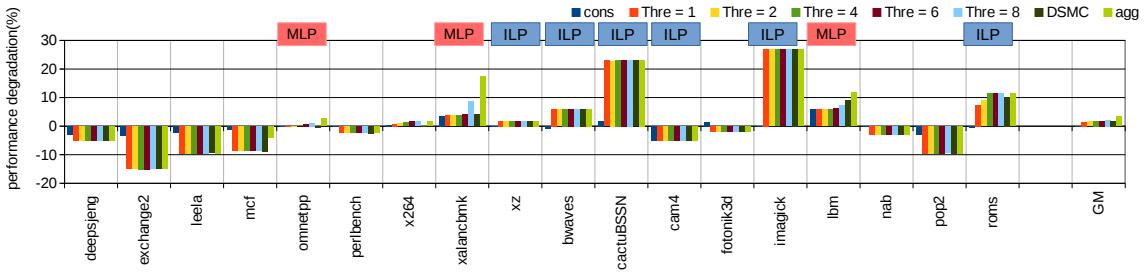

図 12　SWITCH 方式による性能変化 (MLP を用いた方式)

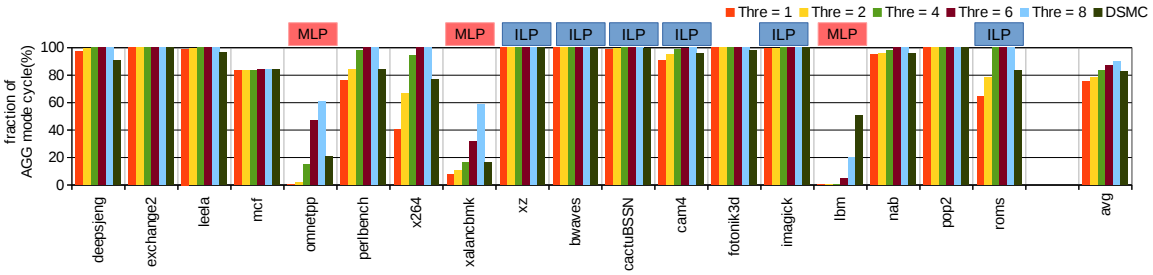

図 13　AGG モードである割合 (MLP を用いた方式)

といえる．

続いて， DSMC の評価に移る．図 13 より high-MLP の中でも，omnetpp と xalancbmk においては AGG モードである割合が 20% 程度と低いのに対して， lbm では 50% と高い割合を示している．これは，lbm は high-MLP の中では比較的 L2C MPKI が低いため，ほかの 2 つと比べて CONS に切り替わる頻度が低いためであると考えられる．

また，図 12 を見ると，lbm では DSMC で 10% 程度と大きく性能が低下している．この結果は．キャッシュ・ミス発生からメモリ・アクセスのレイテンシだけ CONS にしておくのでは不十分である可能性を示している．したがって， CONS に切り替えておくサイクル数を増やして検証する必要があり，これは今後の課題である．

最後に， L2C MPKI を利用する方法と DSMC とを比較する．現状では， DSMC が L2C MPKI を利用する方法に対して有効性が高いとは言えない．また，しきい値を適切に選択できた場合 (今回でいう 6 前後) 性能を大きく低下させず，なおかつ一定の割合で AGG での実行ができている．このことから，先述した DSMC の課題が解決できない限り， M2C MPKI を利用した方法で良いのではないかと考える．

3.7　評価のまとめと課題

ここまでの SWITCH 方式に関する評価をまとめ，課題をリストアップする．

3.7.1　IPC を用いた方式

IPC をもとに ILP を評価して切り替えを行う方式に関してまとめる． この方式には，次の 2 つの問題点があることが分かった．

- ベンチマークによって適切なしきい値が異なる
- AGG では CONS よりも性能が低下するため，本来 CONS に切り替えるべき状況でも AGG にとどまってしまう場合があり，性能が低下する

3.7.2　ISR を用いた方式

ISR をもとに ILP を評価して切り替えを行う方式に関してまとめる．ISR を用いた方式は，IPC を用いた方式と比較して有効性が高いことがわかった．この理由は以下の 2 点であると考えられる．

- AGG 方式と CONS 方式で ISR の差が (IPC の差と比較して) 小さい
- high-ILP に属するベンチマークとそうでないベンチマークとで ISR の差が大きい

性能低下とタグ比較削減のバランスを考えると，しきい値は 10% ～ 15% 程度が適切である．

一方で， ISR を用いた方式は cactuBSSN において性能が低下してしまうという問題がある．これは， cactuBSSN が唯一 AGG 方式と CONS 方式とで ISR が大きく変化することが原因である．

3.7.3 L2C MPKI を用いた方式

L2C MPKI をもとに MLP を評価して切り替えを行う方式に関してまとめる．本方式は high-MLP に属するベンチマークに対して有効であることがわかった．性能低下とタグ比較の削減のバランスを考えると，しきい値は 6 前後が適切である．

3.7.4 DSMC 方式

キャッシュ・ミスが発生したときに AGG から CONS に切り替え，一定期間が経過したら AGG に戻す，Dynamic Switching by Cache Miss(DSMC) 方式にかんしてまとめる．本方式では，high-MLP のうち lbm において性能の低下が大きかった．この理由としては，今回設定した一定期間 (メモリ・アクセス・レイテンシ：300cycle) が短すぎた可能性が考えられる．したがって，性能が低下しないような適切な期間を調べる必要がある．

3.8 SWITCH 方式：ILP と MLP の併用

ILP が高い場合と MLP が高い場合に同時に対応するため，ILP を評価して切り替えを行う手法と， MLP を 評価して切り替えを行う手法を同時に利用する方式に関して評価する．

これまでの評価で，ILP の評価指標には ISR が，MLP の評価指標には L2C MPKI が有効であることがわかった．そこで今回は ISR と L2C MPKI による評価を同時に行い，「ILP が高いと判断された」もしくは「MLP が高いと判断された」場合に次のインターバルを CONS モードで実行し，「ILP と MLP がいずれも低い」と判断された場合には，次のインターバルを AGG モードで実行するようにした．

なお，これまでの評価から，ISR のしきい値は 15%，L2C MPKI のしきい値は 6 と設定している．

3.8.1 評価結果

評価結果を図 14(性能変化)，15(タグ比較の削減率)，16 に示す．図 16 は SWITCH 方式で実行したときの AGG(青) と CONS(赤) の割合を表している．

図 14 より，cactuBSSN を除くすべてのベンチマークで，SWITCH 方式では性能低下が CONS と同程度まで抑制できていることがわかる．したがって，ILP と MLP の評価に応じて適切に切り替えができているといえる．

唯一 cactuBSSN に関しては性能低下が AGG と同程度となっている．これは， cactuBSSN では ISR による制御が上手くいっていなかったことが原因である．図 16 を見ても，cactuBSSN は high-ILP であるにもかかわらず，ほぼ 100% AGG の状態になっており，切り替えが上手くできていないことがわかる．

次に 図 15 のタグ比較の削減率について考える．図 16 において AGG の割合が多いベンチマークは，SWITCH 方式での削減率が AGG に近い高い値を示している．反対に CONS の割合が高いベンチマークにおいては SWITCH 方式での削減率は CONS に近い値となっている．

以上の考察から， ILP と MLP を両方利用する SWITCH 方式においては，ほとんどのベンチマークにおいて性能を低下させず，かつ積極的にタグ比較を削減できる場合には AGG モードに切り替えて積極的な削減ができているといえる．ただし，cactuBSSN においては ISR による ILP の評価制度が不十分であるため，性能が低下してしまっている．

このことから，ILP を評価する方法に関してはさらなる改善が必要であり，これは今後の課題といえる．

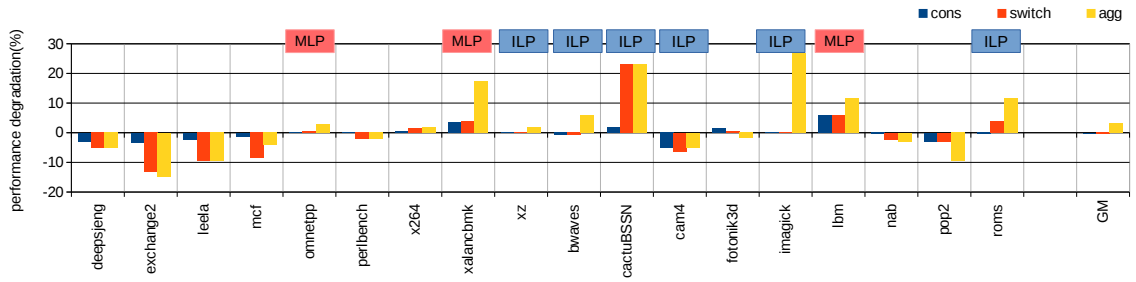

図 14　SWITCH 方式による性能変化

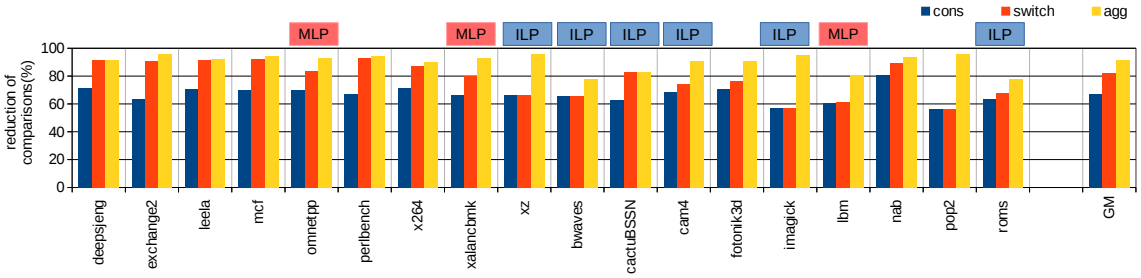

図 15　SWITCH 方式によるタグ比較の削減

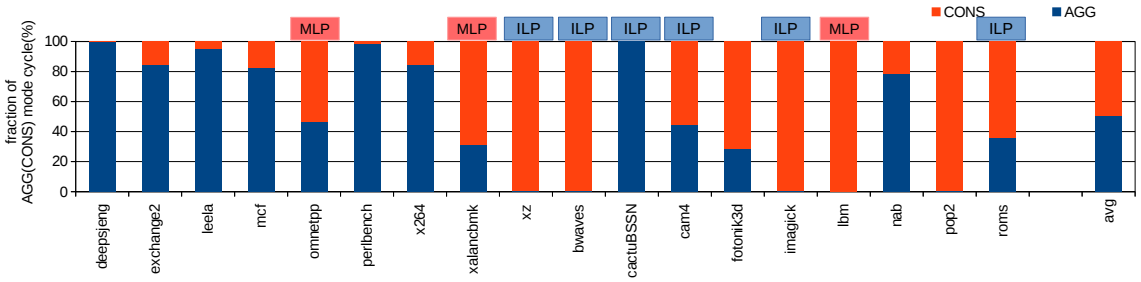

図 16　AGG モードである割合

4　研究計画

- しきい値の細かいチューニング：今回の測定ではおおまかな性質を調べることを目的としていたため，SWITCH に用いるしきい値は荒くしか評価できていない．そのため最もバランスのとれるしきい値をチューニングする必要がある
- ILP の評価方法の改善：ISR による ILP の評価では一部不十分であるため，より精度の高い評価方法及び切り替えアルゴリズムを模索する必要がある
- ウェイクアップの電力測定に関して，松田さんからもらったコードを理解する

5　関連文献：Content-Addressable Memory(CAM) Circuits and Architectures: A Tutorial and Survay [8]

本論文は CAM の回路設計及びアーキテクチャに関する Survay となっている．この中で，自分の提案手法と根本のアイデアが同じである研究が紹介されているため，ここでまとめておく．

本論文ではアーキテクチャの観点から提案された

CAM の電力削減方法が 3 点紹介されている．

5.1 CAM のバンク化

CAM をバンク化することによって回路面積の削減と消費電力の削減が可能である．ここで CAM のバンク化とは，CAM を複数に分割し，各バンクにはデータの下位ビットが一致するものを入れることを指す．これによって，同一のバンクに入っているデータは下位ビットが同一であるためその分のデータ領域を削減できる．さらには，比較が行われる際に，下位ビットが一致するバンクのみ比較を行えばよいため，電力削減も可能である．

本手法の欠点として，バンクのオーバーフローが挙げられる．（ルータの CAM では，オーバーフローが容易におこると記述されていた）この対策として本論文では「定期的なバンクの再分割」を行うことによりバンク間の命令数のバランスをとるという方法が紹介されていた．

これはおそらく，物理的には分割されていない CAM に関してパーティションを定期的にしなおすことにより，特定のバンクでオーバーフローが起きることを防ぐ，ということだと思う．その際の比較方法などは不明であるため，関連論文より調べる必要がある．

5.2 ビット・カウントによる電力削減

この方法では，CAM 内のデータにおいてビットが 1 である数を数えておき，その値をテーブルに保存しておく．そして比較の際には 1 のビット数が一致する CAM のエントリのみ比較気を動作させることによって電力を削減する．

5.3 エンコードの変更による削減

保存するデータのエンコードを変えることによって回路面積を削減する．

参考文献

[1] N. H. E. Weste and D. M. Harris, *CMOS VLSI Design: A Circuits and Systems Perspective, 4th edition*, Addition Wesley, 2010.

[2] F. Monsieur, E. Vincent, D. Roy, S. Bruyre, G. Pananakakis, and G. Ghibaudo, Time to breakdown and voltage to breakdown modeling for ultra-thin oxides (Tox<32Å), In *Proceedings of the 2001 IEEE International Integrated Reliability Workshop*, pp. 20–25, October 2001.

[3] S. Khan and S. Hamdioui, Temperature dependence of NBTI induced delay, In *Proceedings of the 2010 IEEE 16th International On-Line Testing Symposium*, pp. 15–20, July 2010.

[4] J.R. Black, Electromigration–a brief survey and some recent results, *IEEE Transactions on Electron Devices*, Vol. ED-16, No. 4, pp. 338–347., April 1969.

[5] R. Viswanath, V. Wakharkar, A. Watwe, and V. Lebonheur, Thermal performance challenges from silicon to systems, *Intel Technology Journal*, Vol. 4, No. 3, p. 116, August 2000.

[6] Hideki Ando, Swque: A mode switching issue queue with priority-correcting circular queue, In *Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture*, MICRO ’52, p. 506518, New York, NY, USA, 2019. Association for Computing Machinery.

[7] Y. Kora, K. Yamaguchi, and H. Ando, Mlp-aware dynamic instruction window resizing for adaptively exploiting both ilp and mlp, In *In Proceedings of the 46th Annual International Symposium on Microarchitecture, pp.37-48*, December 2013.

[8] K. Pagiamtzis and A. Sheikholeslami, Content-addressable memory (cam) circuits and architectures: a tutorial and survey, *IEEE Journal of Solid-State Circuits*, Vol. 41, No. 3, pp. 712–727, 2006.