2008－ARC－179 (9)
2008／8／6

可変パイプライン段数プロセッサの段数切替えスケジューラの設計と評価

野村和正　佐々木敬泰
大野和彦　近藤利夫

近年，モバイルプロセッサでは，プロセッサの性能向上に伴う消費エネルギーの増加が問題となってきており，低消費エネルギーと高性能の両立が要求されている．そのため，様々な低電力化手法が提案されている．その一つとして著者らは可変パイプライン段数アーキテクチャ (VSP:Variable Stages Pipeline) を提案している．VSP はパイプラインステージ統合 (PSU:Pipeline Stage Unification) と同様，パイプライン段数を動的に変更するアーキテクチャであり，現在主流となっている消費電力削減手法である Dynamic Voltage Scaling (DVS) よりも消費エネルギーを削減可能であることが示されている．しかし，従来の VSP は実行時に最適なパイプライン段数を決めるスケジューラが実装されていないという問題があった．そこで，本稿では VSP を実用化する上で不可欠となるパイプライン段数切換えスケジューラを作成し，評価を行った．本稿で実装したスケジューラは，プロセッサの負荷に応じて，低消費エネルギーかつ高性能を両立するようにパイプライン段数を変更するよう制御を行うものである．このスケジューラを VSP に組み込み評価を行った結果，パイプライン段数の少ない低消費電力向けプロセッサと比較し，電力遅延積において約 2 割の改善率が得られた．

Design and Evaluation of Mechanism Changing Pipeline Stage for Variable Stages Pipelne

Kazumasa Nomura, Takahiro Sasaki, Kazuhiko Ohno and Toshio Kondo

In late years, with the mobile processor, the increase of the consumption energy with the performance enhancement of the processor becomes the problem, and low consumption energy and high-performance coexistence are demanded. Therefore it is suggested various low power technique. we suggest Variable Stages Pipeline (VSP). VSP, is similar to Pipeline Stage Unification (PSU), is the architecture to change the number of the pipeline stages in dynamically, and can reduce the consumption energy than Dynamic Voltage Scaling (DVS). However, there was a problem that the scheduler of VSP decided the number of the pipeline stages where were most suitable for run time was not implemented. Therefore, in this paper, for VSP widely used, I implemented the number of the pipeline stages change scheduler and evaluated it. The scheduler which I implemented controls it depending on the load of the processor to change the number of the pipeline stages to low consumption energy and high efficiency. As a result of evaluated by incorporated this scheduler in VSP, proposal scheduler is about 20% better than low consumption energy processor with Energy-Delay Product.

1. はじめに

近年，モバイルプロセッサでは，低消費エネルギーと高性能の両立が要求されている．この要求に応えるため DVS (Dynamic Voltage Scaling)[1)2)] が提案され広く用いられている．DVS はプロセッサ負荷やユーザから指定されたバッテリ持続時間要求に応じて，動的に電源電圧とクロック周波数を変更することで，低消費エネルギーかつ高性能を実現している．DVS は消費電力を削減する手法としては有効であるが，近年のプロセス技術の進歩により，電源電圧を変更できる幅が小さくなってきており，将来 DVS の有効性は減少していくものと考えられる．

そこで，我々は DVS に変わる低消費電力手法とし，パイプライン段数を動的に変更する可変パイプライン段数アーキテクチャ (VSP:Variable Stages Pipeline)[3)] を提案している．VSP は，DVS のように電源電圧とクロック周波数を変更することで消費電力を削減するのではなく，パイプラインステージを統合することにより，パイプライン段数を減らし，同時にクロック周波数を低下させることで消費電力を削減している．そのため，VSP は将来の超低電力 CMOS プロセスにおいても変わらない効果が期待できる．しかし，従来の VSP は実行時に最適なパイプライン段数を決めるス

ケジューラが実装されていないという問題がある．

そこで，本稿では VSP を実用化する上で不可欠となるパイプライン段数切換えスケジューラを作成し，評価を行った．本稿で実装したスケジューラは，コンピュータの負荷に応じて，低消費エネルギーかつ高性能を両立するようにパイプライン段数を変更するよう制御を行うものである．著者らの行った評価によると，メインメモリへのアクセス数とプロセッサの負荷は密接に関係していることがわかったため，提案スケジューラは負荷を測定するためにメインメモリアクセス頻度に着目し，動的にパイプライン段数変更を行うよう設計した．同時に，本稿のスケジューラは細粒度にパイプライン段数を切り替えるよう設計したものであるが，細粒度な段数切換えを行った場合，段数切換えにかかるオーバーヘッドを無視できないためパイプライン段数変更のオーバヘッド削減手法を提案し，評価によりその有効性を明らかにする．

提案スケジューラを VSP に組み込み評価を行った結果，パイプライン段数の少ない低消費電力向けプロセッサと比較し，電力遅延積 (ED 積) において約 2 割の改善率が得られた．また，パイプライン段数の多い高性能向けプロセッサと比較し，今回評価のために使用したベンチマークのすべてに対し，ED 積において良い結果が得られた．

2. VSP (Variable Stages Pipeline)

本節ではまずベースとなる VSP について述べる．VSP は周波数とパイプライン段数を負荷に応じて動的に切り替え，低消費エネルギーと高性能の両立を実現する手法である．具体的には，多段パイプライン構成で周波数を高くし高速に動作させる HS (High Speed) モードと，少段パイプライン構成で周波数を低くし低速で動作させるが消費電力を低減する LE (Low Energy) モードを用意し，この 2 つのモードを負荷に応じて切り替えることで低消費エネルギーかつ高性能を実現しようとしている．VSP アーキテクチャは図 1 のような構成となる．また，パイプライン段数を変更する方法は，汎用プロセッサのパイプラインレジスタの部分を図 2 のような構成（以下，D-FF+MUX）に変更することで実現できる．図 2 において HS モード時は D-FF を選択し，LE モード時は D-FF を選択しないことでパイプライン段数変更を実現できる．ただし，このままではパイプラインレジスタを結合した部分で巨大な組み合わせ回路ができてしまい，グリッチによる電力増加が発生してしまう．ここで，グリッチとは回路にあらわれる無駄な電気信号の変動のことであり，パルス周期の突然の変化や，ゲート遅延・配線遅延のばらつきなどで生じてしまう無駄な電力のことである．また，グリッチは一度発生すると次の回路へ伝播され，後段の回路ではさらにグリッチが発生する．このグリッチの伝播はラッチを挿入することで緩和できる．そこで VSP では，統合するパイプラインレジスタをすべて D-FF+MUX に置換するのではなく，一部にパイプラインレジスタとラッチの動作を切り替えられる LDS-cell という回路を挿入することでグリッチの緩和を行っている．LDS-cell は図 3 のような構成になっており，HS モード時では D-FF として動作させるが，LE モード時はラッチとして動作させることにより，グリッチの伝播を解決している．

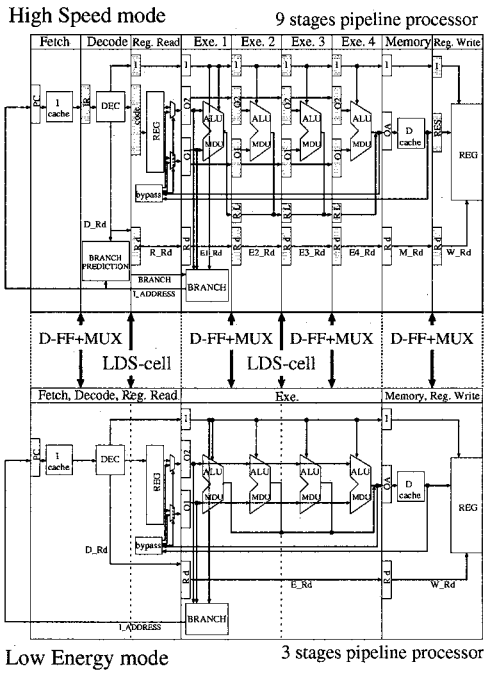

図 1　VSP プロセッサ

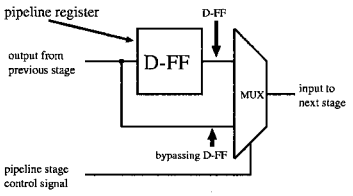

図 2　D-FF+MUX

また，HS モードと LE モードの 2 つのモードにおいての特徴は以下に示す通りである．

- HS モード

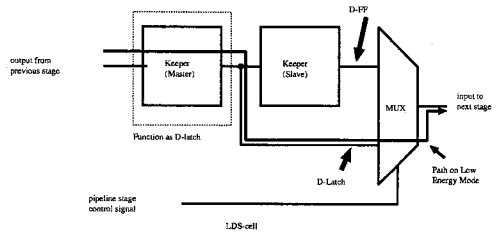

図 3 LDS-cell

- 多段パイプライン構成であり，LDS-cell はパイプラインレジスタとして動作する．
- インターロックと演算結果のフォワーディング機構を搭載している．

- LE モード
  - 少段パイプラインであり，LDS-cell はグリッチの緩和を行う D-ラッチとして動作する．また，DFF+MUX では D-FF を使用せず，パイプラインを統合する．
  - 周波数を低下させること，バイパスされて使用しなくなったパイプラインレジスタのクロックを止めることから消費電力を削減することが出来る．
  - 分岐予測ミスペナルティの低下，データ依存による待ちサイクルの低減により，HS モードに比べ IPC が向上する．

また，VSP と同様にパイプライン段数と周波数を変更するアーキテクチャとして，パイプラインステージ統合（PSU:Pipeline Stage Unification）[4)5)] が提案されている．VSP と PSU の主な違いは，PSU はグリッチの削減を行っていないが，VSP では LDS-cell を追加し，グリッチの削減を行っている点である．

2.1 VSP と DVS, PSU の比較

文献 3) において，著者らは VSP と DVS, PSU の実行時間と消費電力の評価を行った．文献 3) より，LE モードにおいて，VSP は DVS に比べ約半分の実行時間で処理が終了し，消費エネルギーも DVS と比較し約半分程度に削減できることを示した．同様に，VSP と PSU の比較の結果では，グリッチの削減効果により，VSP は PSU と比べ 3 %ほど電力が削減されることを示した．

3. VSP スケジューラの必要性

VSP は消費電力の低減において有効性があると示したが，スケジューラは VSP の性能に大きく影響を与えてしまうため，性能の良いスケジューラの作成は不可欠である．

現在幅広く用いられている DVS においても電源電圧を切り替える制御を与えるため，スケジューラが用意されている．DVS スケジューラの仕組みは OS が負荷を監視し，負荷に応じて電源電圧と周波数を変更している．しかし，電源電圧変更には多くのオーバーヘッドが発生するため，DVS スケジューラは長い期間で電圧を切り替えるよう設計されている．しかし，周波数変更やパイプライン段数変更のためのオーバーヘッドは電源電圧変更に比べるとかなり小さいため，DVS スケジューラを VSP にそのまま組み込むことは適切ではないと考えられる．

PSU においても，文献 6) のスケジューラが提案されている．しかし，文献 6) のスケジューラは複雑な仕組みになっており，ハードウェア量の増加や，実現の困難さが懸念されている．また，VSP は数百命令単位くらいの細粒度のモード切換えを想定しているが，文献 6) のスケジューラでは，10 万命令に 1 度の割合でモード切換えを行っており，VSP にそのまま適用することは困難である．

このような理由から，本稿では細粒度でハードウェア量の小さい VSP スケジューラを提案し，評価を行う．

4. 提案 VSP スケジューラの仕組み

本節では提案 VSP スケジューラの仕組みについて述べる．

本稿で提案するスケジューラはプロセッサの負荷に応じてモードを切り替える手法である．つまり，負荷が高く高速な処理が必要だと考えられる場合には HS モードにし，負荷が低く高速な処理が必要ない場合には低電力である LE モードで動かすことにより，効率の良い処理が行うことができる．負荷の取得方法が誤っていた場合，効率の良くないモードへ切換えを行う可能性があるため，スケジューラを設計する上で、負荷の動的検出方法は重要である．単位時間当たりにどのくらい命令が実行されているかということが負荷の指標の一つであるが，命令の依存関係やメインメモリアクセス速度などの問題から，命令はスムーズに流れず，プログラムの特性によって負荷にはばらつきが生じる．負荷を OS 側で見ることもできるが，細粒度なスケジューラを設計することから考えると，ハードウェアでの実現が必要である．プロセッサの負荷を判断する指標を決定するため，様々なシミュレーション実験を行った結果，メインメモリアクセスが特に命令の流れを遮っていることがわかった．現在プロセッサの処理速度に比べメインメモリアクセス速度の遅さが問題となっているが，今後もメインメモリアクセス速

度の遅さは問題となると考えられる．そのため，今回load命令とstore命令の頻度からモード切り替える制御を与えることとした．load命令とstore命令の回数をカウントする場合，キャッシュヒットとキャッシュミスを区別なくカウントするが，キャッシュミス回数をカウントする手法も実装し，比較したところ，評価結果の変化は見られなかった．そのため，実装が容易だと考えられるload命令とstore命令の回数をカウントする手法を選択した．また，モード切換えを行うしきい値は，いくつか考えられるパターンにおいてシミュレーションの評価を行い，よりよい結果が得られた値を用いた．

4.1 モード切換えオーバーヘッド

本稿ではモード切換えを細粒度に行うVSPスケジューラを作成するが，細粒度な設計の場合，モード切換えを行うためのオーバヘッドの影響を多く受けてしまう．VSPにおいてモード切換えを行うためには以下の2つのオーバヘッドが発生する．

- 周波数変更を行うためのオーバヘッド
- パイプラインレジスタ数が減少する場合に，パイプラインフラッシュをしなければならないため発生するオーバヘッド

モード切換えを行う場合，前者のオーバヘッドは必ず発生してしまうため，回避することができない．しかし，周波数変更にかかるオーバヘッドはたかだか2，3サイクルであるため無視できる．後者は，HSモードからLEモードへ切り替える場合に発生する．VSPのパイプラインレジスタの中にはHSモードでは使用されるが，LEモードでは使用されないものがある．このようなパイプラインレジスタでは，HSモード動作時ではデータが保存されるが，HSモードからLEモードへモード切換えを行った場合，LEモード時ではこのパイプラインレジスタを使用できないため，HSモード時に保存したはずのデータを呼び出すことができなくなってしまい処理が正常に行うことができなくなってしまう．この問題を解決するために，HSモードからLEモードへ切換えを行う場合，パイプラインレジスタを一度フラッシュさせ，パイプラインレジスタ内に残っていた命令を初めからやり直すという手法を行う必要があると考えられていた．しかし，この方法では処理が進んでいたはずの命令を初めからやり直さなければならないので効率が悪くなってしまう．これが後者のオーバヘッドである．このオーバヘッドは数十サイクル以上のオーバヘッドがかかってしまうため，細粒度にモード切換えを行う場合には，このオーバヘッドが積み重なってしまい，性能を悪化させる原因となってしまう．

4.2 モード切換えオーバーヘッド低減手法

そこで，HSモードからLEモードへの切換えを行うタイミングを分岐予測ミス発生時に限定することで，モード切換えのオーバヘッドを隠蔽する手法を提案する．

一般的なアプリケーションにおいて，分岐命令の出現頻度は11 %～17 %であり，分岐予測ミス率は分岐予測を行ったとしても4 %程度は発生してしまう．このことから，200命令に一度くらいは分岐予測ミスが発生することになる．従って，分岐予測ミスをきっかけにしても細粒度なモード切換えを実現できるため，本稿では分岐予測ミスをモード切換えオーバヘッド削減のために活用する．

図4を用いてオーバーヘッド削減手法を説明する．図4に示すように，あるサイクルにおいて，HSモードの状態で命令がいくつか流れているとする．ここで，分岐予測ミスが発生した場合，データが残るのはプロセッサの命令処理ステージのうち実行ステージ以降のパイプライン内だけである（図4のB）．実行ステージ以前のデータはフラッシュされ，パイプラインレジスタ内は空になる（図4のA）．モード切換え時において実行ステージ以降の処理は高周波数でHSモードとして動かし，実行ステージ以前の処理はパイプライン段数を変更し低周波数でLEモードとして動かすという期間（以下，MDモード）を設ける．MDモード中は実行ステージ以降の命令の方が速く処理されるため，新しくフェッチされた命令がリザベーションステーションに登録される頃には実行ステージ以降の命令はすべて実行され，実行ステージ以降のパイプラインレジスタ内が空になる（図4のC）．その時，LEモードに初めて完全に切り替えることができる．このようにすることで，モード切換えのためにかかっていたパイプラインフラッシュによるオーバヘッドの削減を目指す．

5. 実　　装

本節では提案手法であるVSPスケジューラの実装について述べる．図5にVSPスケジューラのブロック図を示す．

図5の32 entry queueは最近に行われたload, store命令の回数を記憶しておく部分である．Instruction Decoderから現在のサイクルのload, store命令の命令数が32 entry queueに渡される．32 entry queueは32個の3bitD-FFで構成されるキュー構造になっており，3bitD-FFに1サイクル中のload, store命令

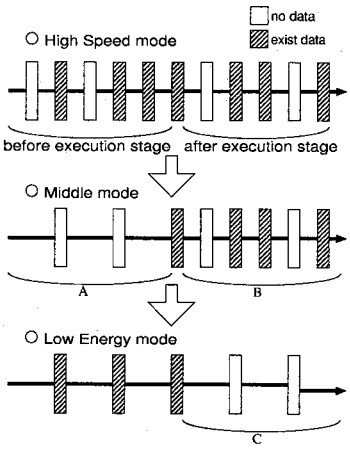

図 4　オーバヘッド低減手法

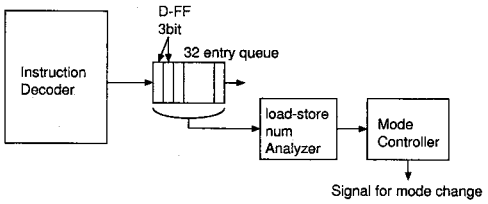

図 5　提案スケジューラに必要なハードウェア

の命令数が保存される．このようにして，最近の 32 サイクルの load, store 命令の命令数を保存する．load-store num Analyzer は 32 entry queue のデータを受け取るが，これまで示したように 32 entry queue には load, store 命令の命令数がそのまま格納されているわけではない．そのため，load-store num Analyzer は命令数の解析を行う．現在，load-store num Analyzer の機能は確立しているが，ハードウェアでの実現方法は確立していないが，今回はシミュレーションでの評価のため，ハードウェアを考慮せず評価を行った．しかし，load-store num Analyzer の仕組みは複雑ではないと予想される．Mode Controller は load, store 命令数と，あらかじめ設定しておいたしきい値とを比較し，モード切換えを行うかを制御する部分である．Mode Controller は簡単な比較を行うだけである．

このように，提案 VSP スケジューラは図 5 における回路を追加するだけであり，ハードウェア量の増加はわずかであると考えられる．

6. 性能評価

本稿では SimpleScalar Tool Set 内の out-of-order 実行シミュレータをベースに提案手法を組み込み，評価を行った．また，消費電力を見積もるツールである wattch も SimpleScalar 上に組み込み電力の評価を行った．命令セットは SimpleScalar PISA であり，表 1 に示すように，SPECint2000 の 10 本をベンチマーク・プログラムとして用いた．実行は最初の 1G 命令をスキップした後，1G 命令を測定に用いた．

表 1　ベンチマーク

ベンチマーク名	処理内容
ammp	計算化学
art	画像認識，ニューラルネットワーク
bzip2	圧縮
equake	地震波伝播シミュレーション
gcc	C 言語コンパイラ
gzip	圧縮
mcf	組み合わせ最適化
parser	文字列処理
vpr	FPGA の配置配線
vortex	オブジェクト指向データベース

表 2　プロセッサ構成

processor	issue width 8, RUU 64 entry, LSQ 32 entry, int ALU 8, int mult/div 4, fp ALU 8, fp mult/div 4, memory port 8
branch prediction	PHT 8K entry history width 6 の gshare BTB 2K entry PAS 16 entry
L1 instruction cache L1 data cache L2 cache	64KB/32B line/1-way 64KB/32B line/1-way 2MB/64B line/4-way
memory	first reference 64 cycle
TLB	forward interval 2 cycle

6.1 評価環境

表 2 にシミュレーションにおいて仮定したプロセッサの構成を示す．同時に，LE モードでは周波数の関係上，表 3 に示すようにキャッシュアクセスやメインメモリアクセスのために必要なサイクル数を変更した．LE モード時の場合，メモリに与える周波数を変更しメモリに対する消費電力も低減する方法もあるが，今回はメモリの周波数を変更せずシミュレーションを行った．また，本稿ではパイプライン段数を HS モードは 20 段と仮定し，LE モードは 5 段とした．近年のプロセッサと比較するとかなり段数の多い設定となっているが，これは 20 段以外の段数では SimpleScalar の設計が困難であるためである．

6.2 評価結果

このような評価環境で，1)HS モードのみで動作，2)LE モードのみで動作，3) 提案手法の 3 種類の手法を Simple Scalar 上に実現し評価を行った．また，これらの手法における実行時間，消費エネルギー，ED

表 3 アクセスレイテンシ

モード	HS モード	LE モード
L1	1 cycle	1 cycle
L2	6 cycle	2 cycle
memory	18 cycle	12 cycle
TLB miss	30 cycle	10 cycle

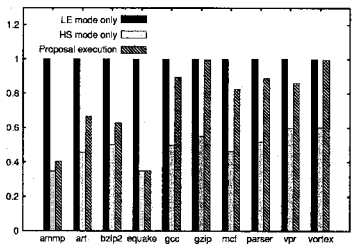

図 6 実行時間

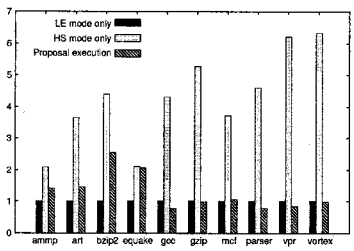

図 7 消費エネルギー

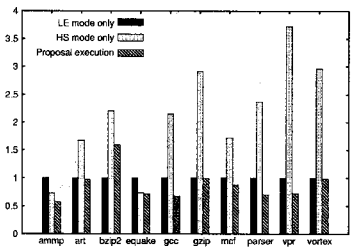

図 8 ED 積

積の評価結果をそれぞれ図 6，図 7，図 8 に示す．

図 6〜図 8 を見るとわかるように，提案手法は HS モードと LE モードをうまく切り替えているため，実行時間は LE モードほどかからず，消費エネルギーは HS モードよりも良いという評価が得られた．その結果，ED 積において，提案手法は LE モードよりも 2 割ほど良い結果が得られた．また，どのベンチマークにおいても，提案手法は HS モードのみの動作と比べて ED 積において良い結果が得られた．

7. まとめと今後の課題

本稿では，VSP を最大限に活かすための細粒度なスケジューリング手法を示した．結果，消費電力が小さいプロセッサを想定している LE モードと比較し，ED 積において約 2 割程度の改善率が見られた．

今後の課題として，文献 6) スケジューラを実装し，本稿の提案手法と比較し評価を行う必要がある．また，スケジューラの効果をより良くするために，さらなるスケジューリング手法を考える，リーク電流などの評価を厳密に行うためにハードウェアで実装し評価を行う，SimpleScalar の設計の困難さから HS モードのパイプライン段数を 20 段に設定しているが，20 段ではパイプライン段数が多いと思われるため，まずは 16 段に変更する．これらの課題を今後改善し評価を行う必要がある．

謝　辞

本研究の一部は科研費補助金（19700042）の援助を受けている．

参 考 文 献

1) Min Yeol Lim, Vincent W.Freeh, “Determining the Minimum Energy Consumption using Dynamic Voltage and Frequency Scaling”, IEEE, 2007
2) Johan Pouwelse, Koen Langendoen, Henk Sips, “Dynamic Voltage Scaling on a Low-Power Microprocessor“, 7th ACM Int, 2001
3) 市川 裕三，佐々木 敬泰，弘中 哲夫，北村 俊明，近藤 利夫，“可変パイプラインを用いた低消費エネルギープロセッサの設計と評価 “, ACS, 2006
4) 嶋田 創，安藤 秀樹，島田 俊夫，“低消費電力化のためのパイプライン”，情報処理学会研究報告，ARC, 2001
5) 嶋田 創，安藤 秀樹，島田 俊夫，“パイプラインステージ統合：将来のモバイルプロセッサのための消費エネルギー削減技術”，先進的計算基盤システムシンポジウム SACSI2003, 2003
6) Jun YAO, Shinobu MIWA, Hajime SHIMADA, Members, and Shinji TOMITA, “A Dynamic Control Mechanism for Pipeline Stage Unification by Identifying Program Phases“, IEICE TRANS. INF. & SYST., Vol. E91-D, pp.1010-1022, APRIL 2008
7) Jinson J Koppanalil, Prakash Ramrakhyani, Sameer Desai, Eric Rotenbergm, Anu Vaidyanathan, “A Case for Dynamic Pipeline Scaling“, ACM, 2002