2004－EVA－9　(5)
2004／6／11

システムシミュレータSimicsを用いたLinux Super Pageの評価

飯田　佳洋†　　清水　尚彦††

† 東海大学大学院工学研究科 〒 259-1292 神奈川県平塚市北金目 1117
†† 東海大学電子情報学部コミュニケーション工学科 〒 259-1292 神奈川県平塚市北金目 1117
E-mail: †{3aepm001,nshimizu}@keyaki.cc.u-tokai.ac.jp

あらまし　Linux Super Page はプロセッサの TLB におけるページ粒度可変機構を Linux に実装したものであり、標準カーネルと比べ TLB ミスを大幅に減少させることでパフォーマンスの向上を目指している。これまでに行ってきたベンチマーク評価とは別の手法として、今回我々はシステムシミュレータ Simics を用いた実アプリケーションにおける TLB ミスの評価、及びカーネルチューニングを行った。本論文では Simics を用いた Linux Super Page 環境の構築と評価方法について述べる。

キーワード　Linux Super Page, TLB, Simics

Evaluation of Linux Super Page with Simics System Simulator

Yoshihiro IIDA† and Naohiko SHIMIZU††

† Graduate School of Engineering, Tokai University, 1117 Kitakaname, Hiratsuka, Kanagawa, 259-1292 Japan
†† Dept. Communications Engineering, Tokai University, 1117 Kitakaname, Hiratsuka, Kanagawa, 259-1292 Japan
E-mail: †{3aepm001,nshimizu}@keyaki.cc.u-tokai.ac.jp

Abstract　Linux Super Page is the implementation of variable page size on TLB. It is expected improving performance by reducing TLB misses compared with normal kernel. We have evaluated Linux Super Page by benchmarks, and verified that Linux Super Page improves performance. In this paper, we evaluated Linux Super Page by Simics, full system simulator, and we describe the details of evaluation and environment on Simics.

Key words　Linux Super Page, TLB, Simics

1. はじめに

近年のコンピュータシステムでは、キャッシュミスと並び TLB ミスもパフォーマンスの低下に大きく影響している。これは TLB が扱うページサイズの小ささとエントリの少なさというハードウェア実装上の問題に起因している。TLB ミスが発生するとオペレーティングシステムが逐一アドレス変換・管理を行うため、TLB ミスのオーバーヘッドは非常に大きいものとなる。

近年の RISC プロセッサではページサイズを可変に管理する機構を実装しており、IA32 においては 4K、4MB、Alpha、Sparc64 では 4K、64K、256K、4MB ページサイズの指定をサポートしている。しかし Linux では単一ページサイズしかサポートしていないため恒常的に TLB ミスが頻発しており、大きなサイズのデータを扱うアプリケーションでは TLB ミスによる性能の低下が無視できない問題となっている。

Linux におけるページング機構の実装はシンプルさを基本理念としており、プロセッサ依存の記述を抑えた設計となっている。現在では Linux はデスクトップ、サーバ、メインフレームでも動作するほど普及しており、インターネットの普及により x86 プロセッサ + Linux + Apache という構成が多く見られるようになってきた。上記のような環境ではページサイズよりも大きなデータを大量に扱うためにキャッシュミス・TLB ミスが頻発することが想定され、キャッシュミス・TLB ミスの抑制によって大きなパフォーマンスの向上が期待できる。

Linux Super Page は標準の Linux Kernel に対してページ粒度可変機構を実装し、TLB ミスの発生を抑制することでパフォーマンスの向上を目指すものである。既に Linux-2.4.19 において Alpha、Sparc64、i386 プロセッサ上で Super Page が動いている [1] [2] [3] [4] [5]。

2. Linux Super Page

Linux Super Page による TLB ミスの減少が顕著に見られるベンチマークの一つに行列転置がある。表 1 に示す環境において Super Page カーネル、標準カーネルの行列転置の比較を行うと図 1、2 に示す通りとなる。行列の大きさが大きいほど標準カーネルでは 4K ページ境界をまたぐアクセス

が多発し、TLB ミスによるパフォーマンス低下が起こっている。

図 2 は Stream ベンチマークを実行したときの比較である。Stream ベンチマークはメモリ転送スループットを計測するものであり、連続アクセスを行ったときに TLB ミスが発生する割合は大きくない。そのため Array size が 2000000 のとき (転送量 45.8MB) のときは性能差は誤差を考慮すると大きくならない。Linux が管理するページサイズを超えるプログラムは数多く存在するため、カーネル・ユーザプログラムを含めて TLB ミスを正確に予測することは困難であるため、ベンチマークだけでは Super Page による TLB ミスの減少を正確に評価することが難しい。そこでシステムシミュレータを用いてアプリケーションレベルの動作を調べ、正確な TLB ミスの評価を試みた。

表 1　ベンチマーク環境

OS	Linux-2.4.19
Processor	Intel(R) Pentium(R) 4 CPU 2.80GHz
L2 Cache	512 KB
TLB	Inst. 64, Data 64

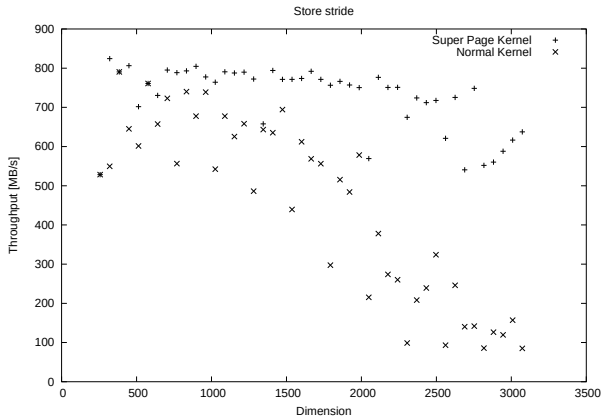

図 1　行列転置 - Store stride 転送

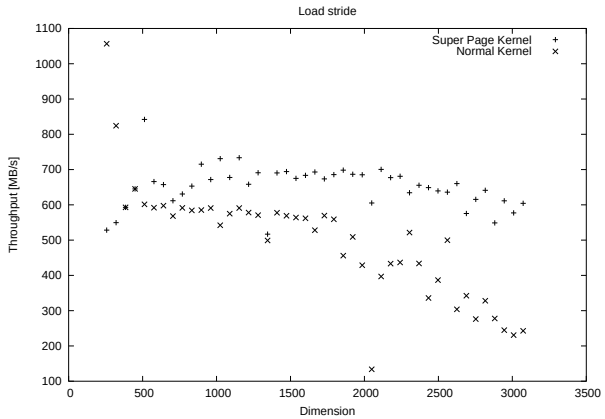

図 2　行列転置 - Load stride 転送

表 2　Stream ベンチマーク

Function	Normal Kernel Rate (MB/s)	Super Page Kernel Rate (MB/s)
Copy:	2487.1944	2618.0047
Scale:	2497.4809	2711.4166
Add:	3106.3805	2999.0493
Triad:	3100.9868	3052.6734

3. システムシミュレータ Simics による評価

Simics [6] [7] は i386、Alpha、Sparc やその他主要マイクロプロセッサ ISA・マイクロアーキテクチャをサポートし、PC/AT アーキテクチャ等の環境が標準で用意されている。Simics は Windows、Linux 等 OS が動作するまでのシミュレーションが可能であり、プロセッサやメモリリソースの量を指定したり、マルチプロセッサ環境、NUMA 環境等を構築することが可能である。Simics では API を使うことによってアーキテクチャの変更・トレース等を行うことができ、ここでは TLB ミスの評価を行う API を利用した。

我々は Linux Super Page カーネルを導入した Linux システムを Simics 環境で用意し、TLB ミスをトレースする環境を構築した。シミュレーションを行うターゲットの環境は一般的な PC/AT 互換の IA32 + Linux + Apache を組み込んだものである。

3.1 Linux Super Page 環境の構築

Simics 上でシステムを動作させるにはシステムのディスクイメージが必要となる。Linux 上では `dd` コマンドを用いて Super Page カーネルを導入したディスクをダンプし、ダンプした領域のヘッド・セクタ・シリンダ数を取得する。ブートローダが格納されている領域や SWAP 領域もイメージ内になければならない。Simics で使われている `craff` というディスクイメージ圧縮ツールは大きなディスクイメージを使用するとき等に必要に応じて使用する。

3.2 TLB ミスの評価

Simcs では `haps` と呼ばれるイベントビューアが用意されており、最も簡単な TLB のトレース方法を以下に示す。

```
simics> trace-hap hap = TLB_*
```

hap を指定することで TLB に関するイベントを見ることができ、自分で作成したトレースプログラムに API を通して組み込むことで TLB ミスの発生をトレースすることができる。haps で見ることができる TLB に関するイベントは表 3 に示す通りである。

また、`tlb0.info`、`tlb0.status` を見ることで実装している TLB エントリ数、アソシエイティブ、及び現在使われている TLB エントリのステータスを見ることができる。

まずはじめに Linux Super Page を実装した linux-2.4.19 における Page Fault の発生について調べた。一般的なウェブサーバを想定した環境では様々なデーモンが稼働しており恒常的に Page Fault が発生するため、その全ての TLB ミスから httpd に関するものだけを抽出するのは効率的では

表 3　TLB に関するイベント

haps の種類	イベントの意味
TLB_Replace_Instruction	Instruction TLB の上書きが発生
TLB_Replace_Data	Data TLB の上書きが発生
TLB_Invalidate_Instruction	Instruction TLB の 1 エントリ無効化
TLB_Invalidate_Data	Data TLB エントリの 1 エントリ無効化
TLB_Fill_Instruction	Instruction TLB の全エントリが有効
TLB_Fill_Data	Data TLB の全エントリが有効
TLB_Miss_Instruction	Instruction TLB ミス
TLB_Miss_Data	Data TLB ミス

ない。ここでは Linux をシングルモードで起動し、init 及び/bin/sh プロセスが稼働している環境を用意した。Simics 上で以下のトレースを実行し Page Fault 割り込みの状況を確認した。

```
trace-exception name = Page_Fault_Exception
```

Linux を最小限のプロセスで起動した状態では Page Fault が発生せず、ユーザイベントに起因するものによってのみ Fault が発生する。このときの TLB 使用状況を図 3 に示す。

httpd を動かす上で最小限のネットワークサービスの構成で TLB の利用について調べると、httpd リクエストが無い状態では TLB の使用率が低く Page Fault も起こらないが、TLB_Replace_Instruction、TLB_Replace_Data、TLB_Invalidate_Instruction および TLB_Invalidate_Data が発生しており、Linux の TLB 管理によって発生している。

httpd リクエストが起こると上記のイベントに加えて TLB_Fill_Instruction 及び TLB_Fill_Data が発生し、TLB が足りずに Page Fault が頻発した。httpd リクエストの処理中のある一点での測定では、TLB の Instruction、Data 共に 100%がユーザプロセスによって使用されており、TLB ミス一つにつき約 300 ステップ程のアドレス変換のオーバーヘッドが生じた。

3.3　Apache Benchmark による評価

以上の環境にて Apache Benchmark を利用して httpd 動作時の TLB ミスの評価を試みた。Simics 上で httpd を起動し、リクエストを処理するときの TLB ミスをトレースすると、haps を用いただけのトレースではどこで TLB ミスが発生したかを命令レベルでトレースすることになる。

命令レベルによるトレースでは TLB ミスの数を評価するだけであり、その結果を Apache や使用した module、Linux Kernel のチューニングにフィードバックすることが難しい。よって Apache Benchmark を実行したときの TLB ミスをトレースする環境について検討した。

haps や API を用いることによって TLB ミス等のイベントの発生数をカウントすることは容易である。API を通して実行中のプロセスを取得することは可能であるため、特定のプロセスにおける TLB ミスの評価は可能である。また、TLB ミスハンドラのカーネルルーチンの実行をイベントとして捉えることができるため、全体の実行時間に占める TLB ミスハンドラの割り合いを評価することも可能である。

```
Instruction 4 Mb (1/64 entries used)
------- LA ------- ------- PA ------- U/S R/W G PAT MTRR
0x00000000c0000000 0x0000000000000000 supr ro G WB WB
Instruction 4 kb (3/64 entries used)
------- LA ------- ------- PA ------- U/S R/W G PAT MTRR
0x0000000040102000 0x000000000139c000 user ro - WB WB
0x0000000008048000 0x000000000bbfa000 user ro - WB WB
0x00000000400fc000 0x000000000133b000 user ro - WB WB
Data 4 Mb (8/64 entries used)
------- LA ------- ------- PA ------- U/S R/W G PAT MTRR
0x00000000c0000000 0x0000000000000000 supr rw G WB WB
0x00000000c1000000 0x0000000001000000 supr rw G WB WB
0x00000000c9000000 0x0000000009000000 supr rw G WB WB
0x00000000cac00000 0x000000000ac00000 supr rw G WB WB
0x00000000cb000000 0x000000000b000000 supr rw G WB WB
0x00000000cb400000 0x000000000b400000 supr rw G WB WB
0x00000000cb800000 0x000000000b800000 supr rw G WB WB
0x00000000cbc00000 0x000000000bc00000 supr rw G WB WB
Data 4 kb (4/64 entries used)
------- LA ------- ------- PA ------- U/S R/W G PAT MTRR
0x00000000cc828000 0x000000000acd8000 supr ro G WB WB
0x0000000008049000 0x000000000bbe8000 user ro - WB WB
0x000000000804a000 0x000000000bcd9000 user ro - WB WB
0x00000000bffff000 0x000000000b9da000 user rw - WB WB
```

図 3　Linux の最小限の稼働時における TLB の使用状況

3.4　シミュレーション実行速度

Simics を用いたシミュレーションはエミュレータレベルであり、評価するイベント数によって速度は異なるが今回の検証では実機の数分の 1 から数十分の 1 のスピードでの動作であった。Simics は x86-64 の開発用エミュレータ VirtuHammer [8] としても知られ、開発・評価用としての実績がある。

4.　ま　と　め

Linux Super Page は単一ページサイズしかサポートしていない Linux カーネルに対して複数のページサイズをサポートするものである。これは HP-UNIX や IRIX と比べてユーザプログラムに変更を加えること無く動的にラージページを割り当てることができる点で異なる。TLB ミスの発生を抑えることによって数倍の性能向上を得ることができ、科学技術計算、Web サーバ、データベースや画像処理といったページサイズを大幅に超えるサイズを持つデータを扱う分野では

適用可能である。

今回は Linux Super Page による TLB ミスの削減について、定量的な評価を行う環境を検証し、システムシミュレータ Simics を用いたシミュレーションについて検討を行った。Simics は様々なプロセッサ ISA・マイクロアーキテクチャ、マルチプロセッサやメモリモデルをサポートし、Windows や Linux といった OS の動作レベルでのシミュレーションが可能である。

Simics API を使った TLB 評価を行うための環境を構築し、Linux Super Page の定量的な評価に用いることができるものと確認した。TLB ミスの定量的な評価として、プロセスの中での TLB ミスの発生数、TLB ミスハンドラの実行時間を計測できることを確認した。

5. 今後の課題

今回は Simics を用いた Linux Super Page の評価環境の構築と評価方法の検討を行った。今後この方法により Apache Benchmark 等実アプリケーションに近いベンチマークを行い、TLB ミスに関する評価を行う。並びにこれによって得られた評価から Linux Kernel のチューニングへフィードバックし、TLB 管理に関する新たなアプローチを検証する予定である。

文　献

[1] 高取,“IA32 版 Linux Super Page の開発”, 東海大学工学部卒業研究論文, 2001

[2] 早坂, 高取, 清水,“IA32 版 Linux Super Page の実現と評価”, コンピュータシステムシンポジウム, 2002

[3] 早坂, 清水,“Linux Super Page とページカラーリングによるメモリ性能の評価”, IPSJ-SIG EVA-5, 2002

[4] 早坂, 清水,“IA32 版 Linux Super Page の実現と評価”, ACS 論文誌第 2 号, 2003

[5] http://shimizu-lab.dt.u-tokai.ac.jp/lsp.html

[6] Peter P. et al.,“Simics: A Full System Simulation Platform”, IEEE Computer, pp.50-58, Feb 2002

[7] http://www.simics.net/

[8] http://www.virtutech.com/