5K-9

省電力 OS のための QEMU を用いたシステム評価環境の実現*

松尾 和弥 †
東京農工大学工学部
情報コミュニケーション工学科

並木 美太郎 ‡
東京農工大学大学院
共生科学技術研究院

1 はじめに

近年、マシンの高性能化に伴う消費電力・発熱量の増大が問題となりつつある。そのため省電力への要求が組込み・ハイエンドの別を問わず高まってきている。そんな中ハードウェア技術的にはかなり細粒度の電力制御が可能になりつつある。例えばメモリであれば、メモリ全体をいくつかのブロックに区切ってブロック単位での電源 ON・OFF を管理するといったことが可能になると考えられる。

2 評価環境の必要性

電力制御技術を活かすためには次の 2 点を分析・考察する必要がある。(1) 具体的にどのようなハードウェア省電力機能が効果的か (2) その機能を OS などソフトウェア側からどのように利用するのが効果的か。これらを考察する上で、実際にマシンがどのように動作したかという『生のデータ』は非常に重要な設計指針となる。また生のデータは設計だけでなく考案した機能による省電力効果の試算、省電力機能どうしの比較といった用途にも応用できる。

3 本研究の目的・方針

本研究ではこのような生のデータを採取するための環境の構築を目指した。特に今回はメモリの省電力化に着目し、X86・ARM など各種アーキテクチャでのメモリアクセスをトレースできる環境を構築した。環境の構築はフリー、オープンソースの CPU エミュレータである QEMU を改造することで実現した。また構築した環境の利用法を示す意味もこめて、実際に改造した QEMU を用いてメモリアクセスのトレースを行い、細粒度の省電力機能の有用性について簡単にではあるが検証を行った。

4 QEMU を選んだ理由

QEMU はフリー、オープンソースの CPU エミュレータである。オープンソースであるため目的に応じた改造・自由度の高いデータ採取が可能である。オープンソース以外にも他のエミュレータや実機と比べて次のような特長がある。(1) エミュレータとしては比較的高速である (2)X86 以外にも ARM、MIPS など多様なアーキテクチャのエミュレートに対応している (3) 実機を導入するよりも低コストである。これらの特長を鑑みて本研究では QEMU を用いた。QEMU のソース、本研究で用いたテスト用の Linux イメージ、各種ドキュメントなどは http://fabrice.bellard.free.fr/qemu/から入手できる。

5 環境の構築

5.1 改造の概要

メモリアクセスには (1) 命令フェッチによるものと (2) メモリが命令のオペランドに含まれていることによるものがある。命令フェッチによるメモリアクセスをトレースするために `translate.c` の関数 `cpu_exec()` 内に必要な情報を採取するための関数を追加した。オペランドとしてのメモリアクセスを採取するために `softmmu_header.h` 内の関数に適宜情報を採取するための関数を追加した。

性能についてだが、X86 の場合、改造した QEMU では実機の 70 倍程度の速度低下があった。単純に比較できるものではないが数千倍の速度低下も珍しくない [7] トレーサとしては十分な速度であると考えている。

5.2 採取した情報

まず、1 回のメモリアクセスごとの物理アドレスと仮想アドレスは命令フェッチ・オペランドともに採取した。オペランドによるアクセスの場合にはアドレス以外に書込み・読込みの種別を採取した。命令フェッチによるアクセスの場合はアドレス以外にフェッチ時の CPU の特権レベルを採取した。また Linux 系 OS をエミュレートする場合に限るが、カーネルの改造と連携することでアクセスを行ったプロセスの PID も取得できる。オペランドによるアクセスの場合、1 アクセスのトレースにつき 12 バイトを要する。命令フェッチのトレースの場合、20 バイトで何個かのフェッチをまとめてトレースする。これは QEMU の仕様 (エミュレート対象の命令を逐次ではなく複数個まとめて処理する) に合わせたためである。参考までに第 6 章で行ったトレースによるファイルサイズとメモリアクセス回数を表 1 に示す。実行速度同様、メモリトレース環境としては許容範囲だと考えている。

表 1 アクセス回数とトレース結果サイズの関係

	フェッチ	その他	サイズ
X86	10266270	50040956	約 630MB
ARM	72801305	273789993	約 3.6GB

6 構築した環境を用いての検証

本章では構築した環境の利用法を示す意味をこめて、改造した QEMU を用いて実際にメモリトレースを行った。128MB

* Development of System Evaluation Environment using QEMU for Low Power Computer System

† Kazuya Matsuo, Department of Computer and Infomation Science, Tokyo University of Agriculture and Technology

‡ Mitaro Namiki, Institute of Symbiotic Science and Technology,Tokyo University of Agriculture and Technology

の物理メモリを搭載した x86 と ARM アーキテクチャ上で、電源投入から QEMU テスト用の Linux イメージが起動完了するまでの間のアクセスをトレースした。トレース結果から時間経過とメモリ利用状況の関係を視覚化したものを図 1、図 2 に示す。いずれの図も X 軸は経過時間に準ずるものとして実行命令数、Y 軸は物理メモリのアドレスとなっている。X 軸 Y 軸ともに 10 進数表記である。プロット箇所は X 座標の時刻に Y 座標の物理アドレスへのアクセス (読込・書込は区別せず) があったことを意味している。

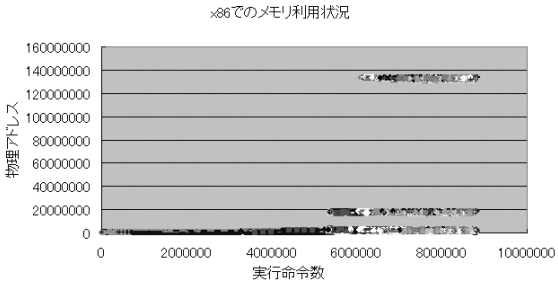

図 1　x86 でのメモリ利用状況

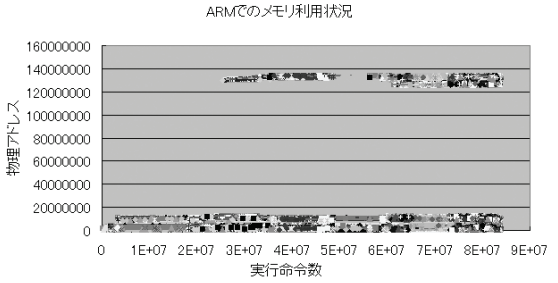

図 2　ARM でのメモリ利用状況

これらの図から分かるように使用されていないアドレスがかなり存在している。これら未使用部分の電源を OFF にすることで電力消費を削減できると考えられる。

そこで次の 2 つの省電力手法を仮定し、実際に消費電力の改善を試算してみた。(1) メモリはいくつかのブロックに等分され、ブロック単位で電源を ON・OFF できる (2) 各ブロックの電源は最初のアクセス時に ON となり、それ以降は ON のままである。また、定量的な指標を得るために文献 [4] を参考に大まかではあるが次のメモリ電力量モデル式を構築した。

$$
\begin{aligned}
\text{電力量} &= 3.3 \times \sum_{i=0}^{N-1}(5945 n_{irw} + \frac{301.7 f_i}{N_B}) \\
&= 3.3 \times (5945 N_{rw} + \sum_{i=0}^{N-1} \frac{301.7 f_i}{N_B}) \qquad (1)
\end{aligned}
$$

n_{irw} は i 番目のブロックへの読込み回数 + 書込み回数である。f_i は i 番目のブロックの電源 ON 期間をメモリクロック数に換算したものである。N_{rw} はメモリ全体での読込み回数と書込み回数の合計、N_B はブロックの数である。読込み・書込みともにメモリサイクルで 6 クロック、バースト長 4、ACT コマンドの実行回数=(読込み回数)+(書込み回数) と仮定し計算した結果である。単位は $V \cdot ns \cdot mA$ である。ただし今回は改善前後の比率に注目するので単位情報は用いない。

更に次の条件を仮定する。(1) データキャッシュ無し (2) データアクセス無しの命令は 1CPU クロック、アクセス有りの命令は 60CPU クロックで実行 (3)CPU:メモリの速度比は 1:10 (4) 命令キャッシュのキャッシュミスは無視。これらの仮定のもと式 (1) から省電力機能有りの場合と無しの場合で図 2 の ARM の電力量がどう変化するかを試算した。その結果としてメモリの消費電力が改善前の 82.3% に抑えられるという結果が得られた。図 2 と比べると物足りない結果と感じられるかもしれない。これは未使用部分の待機電力よりもメモリアクセスに要する電力のほうが影響力が大きかったためである。また、OS 側でハードウェア機能を意識したメモリ管理 (使用メモリより少ないブロックにまとめる割り当てアルゴリズムなど) を行えば動的なメモリサイズの最適化・更なる省電力化が期待できる。

7　問題点と今後の課題

本研究の今後の課題について述べる。まず、トレースした情報を正しく有効に活用するための環境づくりが急務である。データキャッシュの考慮・より正確な消費電力モデル式の考案など計算の精度を高めていく必要がある。また、メモリの利用状況はプログラムによって異なる。様々なプログラムからメモリトレースを行い、どのような省電力技法が有効なのかを更に考察していく必要がある。

また、QEMU と実機の差異というものも考えなければならない。いずれは QEMU から得られた結果を実機の結果と比較する必要もある。

8　まとめ

本稿では QEMU を用いたメモリアクセストレース環境の構築、および細粒度の電力制御の有用性について述べた。高まり続ける省電力への要求に応えるために今後も研究を進めていく。

参考文献

[1] 佐々木 広、浅井 雅司、池田 佳路、近藤 正章、中村 宏『統計情報に基づく動的電源電圧制御手法』SACSIS2006

[2] Kiran Puttaswamy, Kyu-Won Choi, Jun Cheol Park, Vincent J. Mooney ,Abhijit Chatterjee and Peeter Ellervee『System Level Power-Performance Trade-Offs in Embedded Systems Using Voltage and Frequency Scaling of Off-Chip Buses and Memory』ISSS2002, http://portal.acm.org/citation.cfm?id=581249&coll=portal&dl=ACM&CFID=11103101&CFTOKEN=69544047

[3]『新世代マイクロプロセッサアーキテクチャ (前・後編)』情報処理 Vol.46 No.10 pp1099-pp1143, Vol.46 No.11 pp1211-pp1265 社団法人 情報処理学会 2005

[4]『SDRAM の使い方』Elpida Memory,Inc. 2005 http://www.elpida.com/pdfs/J0123N60.pdf

[5] Daniel P. Bovert, Marco Cesati 著 高橋 浩和監訳『詳解 Linux カーネル 第 2 版』株式会社オライリー・ジャパン, 2003

[6] QEMU のソース・バイナリ・ドキュメントなど http://fabrice.bellard.free.fr/qemu/

[7] 杉村 康、伊土 誠一『Pentium 系 Linux の詳細動作を解析するスーパートレーサとアナライザ (STDB)』http://lc.linux.or.jp/lc2001/papers/stdb-paper.pdf