

電力ベースサンプリングシステム PARITS の提案

平井 聡 三輪 真弘 佐藤 充 久門 耕一

(株)富士通研究所

1. はじめに

PARITS (Power Aware Regulated IT System) は、PC サーバのプロセッサ、主記憶等の消費電力を累積カウンタにより計測する装置、および制御ソフトウェアで構成される電力ベースサンプリングシステムである。PARITS により、PC サーバ上で動作する OS、アプリケーションプログラムの関数単位の電力消費を時系列で高精度に分析することが可能となる。本発表では、この PARITS のしくみおよび特徴について説明する。

2. 背景

近年、様々な機器の低消費電力化が注目されており、コンピュータシステムにおいても、ソフトウェアの挙動と消費電力の関連を分析する手法が求められている。

これまでに、組込み系システムではシミュレーションや実測を通じてプログラム動作と消費電力を関連付ける取り組みが行われてきた[1]。

しかし、PC サーバシステムにおいては、CPU、メモリ、チップセット以外にも多数の I/O デバイスから構成されており、部品点数の多さや複雑さから、単純に外部の電力を測定しても内部動作との関係を結びつけることが難しかった。

そこで我々は、PC サーバシステム上でプログラム動作と消費電力を直接関連付け、高精度の解析を行うため、マザーボード上の複数ポイントの消費電力をトリガーとした実行命令アドレスサンプリングを実現する PARITS の開発を行った。

3. PARITS の構成

PARITS は電流測定ボードとそれを制御しプログラム動作と結び付けるソフトウェア部から成る。

3.1 電流測定ボード

電流測定ボードはサーバ上に流れる電流に応じてカウンタを制御するハードウェアである。電流測定ボード内部にはカウンタが設けられており、測定された電流量に比例してカウンタがカウントアップされる仕組みが用意されている。

電流はマザーボード上の電源ラインに微小抵抗 (数 mΩ) を挿入することで検出し、電流測定ボードに入力される。電流測定ボードは PCI Express スロットに接続される拡張カード形式で実装されている。電流測定ボード上には FPGA が搭載されており、PCI Express のインタフェースやカウンタの制御を行っている。電流測定ボードは CPU からは PCI デバイスとして認識され、カウンタはレジスタとしてアクセス可能である。

電流測定ボードには、カウンタがオーバーフローすると CPU へ対して割り込みを発生する機能が備えられており、CPU 内部の PMC (Performance Monitoring Counter) と同様の使い方が可能となっている。

なお、電流測定ボードは、複数のカウンタを用意し、CPU、メモリなど複数のポイントからの入力に対応している。

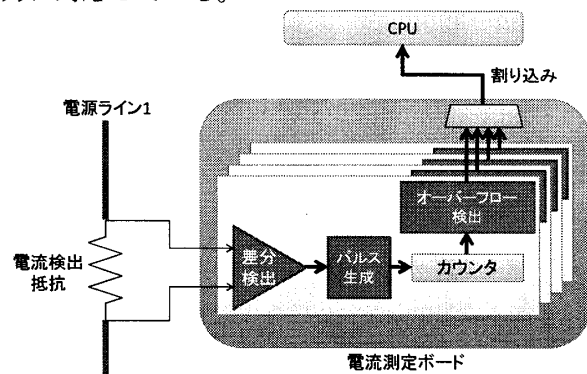


図1: 電流測定ボード概念図

3.2 制御ソフトウェア

制御ソフトウェアは電流測定ボードの初期化、カウンタ値の読み書き、割り込み制御などを行う Linux 用 OS ドライバとして実装している。

計測時は、本ドライバを通じて、カウンタの選択、カウンタがオーバーフローするまでの回数 (サンプリングレート) 設定を行う。

カウンタのオーバーフロー時には図2のように電流測定ボードより割り込みが発生し、本ドライバの割り込みハンドラで、実行箇所を特定するためのプロセス ID (PID)、命令アドレス (PC)、および時刻 (TSC: Time Stamp Counter) を記録する。

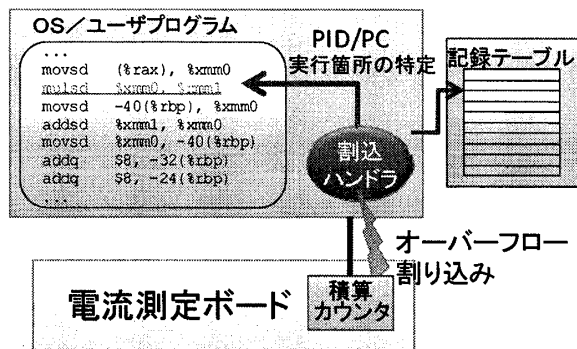


図 2: 制御ソフトウェア概念図

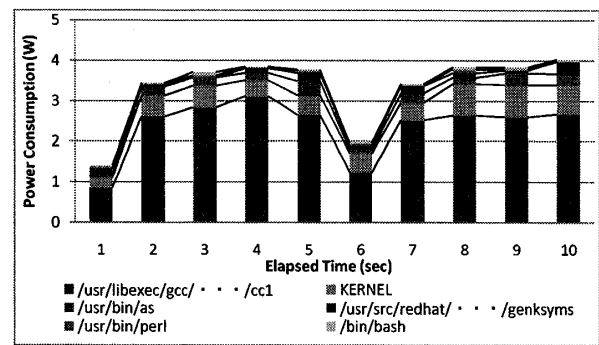


図 3: 時系列分析例(カーネルコンパイル)

4. PARITS の特徴

PARITS は、データ採取の精度を上げるため、またより詳細なプログラム動作との関連付けを行うために、以下のような手法を取り入れている。

(1) NMI (Non-Maskable Interrupt) の使用

PARITS 電流測定ボードは、CPU への割り込みとして、マスク不可能な割り込み (NMI) を使用している。これは、CPU への割り込みの通知方法として、通常の IO APIC 割り込み線経由ではなく、MSI (Message Signaled Interrupt) を使用し、かつ割り込みタイプとして NMI を指定することで実現している。NMI の使用により、割り込み発生時の応答性を向上させると共に、OS やドライバ内の割り込み禁止区間でのデータ採取を可能としている。

(2) 時系列データによる挙動解析

制御ソフトウェアでは、通常のプロファイラと同様に割り込み発生時の走行アドレスに対してプロセス/関数単位での集計を行うことが可能である。PARITS では、これに加え、割り込み発生時のデータを時刻情報 (TSC) と共にシーケンシャルに蓄積することで、測定期間内の任意の時間範囲での分析も可能としている。

図 3 は、Linux カーネルのコンパイル実行時に PARITS を用いて採取した CPU 消費電力プロファイルである。10 秒間データ採取を行い、1 秒毎にプロセス単位での消費電力を集計している (表示した上位 6 プロセスで CPU 電力の 98% を消費)。コンパイラ本体 (cc1) が約 7 割の CPU 電力を消費し、Linux カーネル部 (KERNEL) も 1~2 割消費していることが分かる。なお、グラフの谷間は CPU Idle が多い処理部分であった。

このように、PARITS を用いることにより、システムのプログラム挙動変化と消費電力の関係を容易に分析することが可能である。

5. PARITS の応用

現在、我々は PARITS を用い、CPU 消費電力をベースとしたサンプリングの精度検証を行っている

る段階 [2] であるが、今後以下のような応用を検討している。

(1) 複数ポイントによるプログラム消費電力分析
CPU 以外にメモリ、チップセット、HDD 等の電流測定ポイントを付加しプロファイリングを行うことで、メモリアクセス、I/O アクセスを含めたシステム全体の詳細な電力消費情報を採取、分析する。

(2) CPU の PMC との関連分析

プログラム動作と消費電力の関連分析において、CPU 内部に持つ PMC を利用した取組みが行われているが [3]、PARITS を使用し、カウンタオーバーフロー割り込み時に PMC を併せて読出すことにより、PMC イベントと消費電力の相関を高精度で得ることが可能となる。さらに、複数の PMC イベントを利用した電力推定のモデル化を行い、OS レベルでのプロセスマイグレーションやアプリケーションレベルでのジョブスケジューリングへの適用を検討している。

6. おわりに

PARITS は、PC サーバ上で動作する OS、アプリケーションの電力消費状況を、高精度で解析可能とするシステムである。PARITS を用いることで、消費電力の高いプロセス/関数の抽出や、時間軸での消費電力の変化を容易に分析することが可能となる。今後は、PARITS を用いて PMC イベントと消費電力の関連分析を行い、電力視点でのスケジューラへの適用検討を行う予定である。

参考文献

- [1] The XTREM power and performance simulator for the Intel XScale core: Design and experiences, Jinzhang Peng, et al, ACM TECS 2007.
- [2] 三輪 真弘, 平井 聡, 佐藤 充, 久門 耕一, "電力ベースサンプリングシステム PARITS の評価", 第 72 回全国大会 4A-2, 2010.
- [3] Complete System Power Estimation: A Trickle-Down Approach Based on Performance Events, W. Lloyd Bircher and Lizy K. John, ISPASS 2007.