



Linux/Alpha活用講座

清水 尚彦 <nshimizu@keyaki.cc.u-tokai.ac.jp>

第15回

Alphaでのパフォーマンス測定

Sunが日本語Solarisを無償で提供したり、StarOfficeを無償にしたりと何かと話題を集めています。だんだんとソフトウェアのビジネスモデルが変わりつつあるのか、変えようとしているのかわかりませんが、ずいぶんと積極的な攻勢です。OSとOfficeソフトを無償にして、ハードウェアとエンドユーザーのアプリケーションで利益を得ようとするSunの戦略が、何をターゲットにしているかは明らかです。また、CompaqもLinuxへの各種ツールの提供だけでなく、彼らのUNIXであるTru64 UNIXを99ドルに値下げしたりと、あわただしい日々でした。これが無償になったらどうだろうと思ったりします。それでも私は、自分の使うアプリケーションはLinuxで十分なので乗り換える必要は感じませんが、Netscapeなどのブラウザが安定して提供される環境で64ビットマシンを楽しみたいという人も多いかもしれませんね。

Microsoftが独占企業として訴えられていた訴訟で、Microsoftの敗色が出ているという話はさらに面白く、これからの動向に目が離せません。Sunの動きと合わせて時代の変革に直面しているのかもしれない。もちろん、すでにデファクトスタンダードの地位を確保してしまったMicrosoftは、この程度ではびくともしない可能性の方が高いのも事実です。

一方でメモリの値段のように、ちょっとした事故や天災で価格が大きく変動したりと、ハードウェアのビジネスは相変わらず安定とは程遠く、競争も厳しい世界にあります。特定のプロバイダとの契約を前提にすればハードウェア価格無償というものさえ出始め、まるで日本の携帯電話やPHSのビジネスのようになってきました。体力勝負のこういった世界では投資余力のあるIntelやIBMが当然有利なわけですが、その他のメーカーはこれらの大手が手を出さないニッチな市場でおとなしく利益を追求するか、技術的な優位性を武器に打っ

て出るかの選択を迫られています。Intelの一人勝ちという構図はぜんぜん面白くないので、責任のない立場としては、できる限りほかのメーカーに頑張ってほしいものです。その意味では先月号でお伝えしたAMDのAthlonは、かなりの頑張りをもって健闘していると思います。投資余力の差でIntelが使える製造設備は常に少し先をいくので、これに真っ向から勝負するAMDは相当大変だとは思いますが、ぜひ頑張ってほしいものです。特に数値計算のためにFPUの構造すら変えようという動きは、オープンソースでチャンスをつかみたいコンパイラ研究者などにとっても手放しで喜んでいいことではないでしょうか。枯れたアーキテクチャでは競争は厳しくても、新しい袋には新鮮なアイデアで勝負できます。



Windows CEとLinuxの接続

Alphaと直接関係がないのですが、最近Windows CEを入手し、Linuxと連携して便利に使えたのでちょっとだけ脱線して(いつものことだっ?)この話をします。

携帯メール端末としてシャープのアイゲッティを入手したという話は前に書きました。メール程度の長さの文ならタッチペンでの入力でも耐えられますが、原稿はちょっと無理があります。携帯機器としてはHPの100LX/200LXをずっと使っていてPIMとしては十分満足していたのですが、Vimを導入しても、私には長い文章の入力はやっぱり難しく、何かないものかと探していました。そんな折、ふとしたことでWindows CEはネットワークカードも使えて、しかもTSEのクライアントにもなるらしいと聞いて、わかにCEに興味を持って調べていました。すると新しいモデルが出たモバイルギア の旧

モデル、R320が安売りしていたので、せっかちな私はTSEクライアントとして使えるかはともかく、携帯エディタとして欲しくなって入手しました。モバイルギアはキーボードが大きく(つまり外形も大きいのですが) タッチタイプは問題なく高速に入力できるので、同じような価格帯のほかのマシンと比較すると原稿作成という面では圧倒的に有利です。さらにHP200LXで長時間の電池駆動になれている私には、普通のノートパソコンではとてもやっつけられないので、単3電池2本で25時間というスペックには魅力を感じています。CEならLANへの接続も簡単なので、面倒なことが嫌いな私にも使えるだろうという思惑もあります。さっそくこの原稿はR320で書いていますが、思惑どおりネットワークに接続できないと被害は大きいので多少心配しながら書いています。

なんて書いたそばからネットワークへの接続は成功しました。CEといえどもMicrosoftネットワークへの接続方法はNTや95や98と同じなのですね。その点は評価できます。CEはPCコンパニオンという位置付けがMicrosoftではなされて、PCの母艦が存在することが前提とされています。マニュアルを見てもいろいろと調べても普通のネットワークマシンとして母艦のPCなしで使う方法は分からなかったのですが、やってみれば簡単なことでした。結局、「CEサービス」なんていう付属のCD-ROMは一度も使わずにすみそうです。CEをLinuxを中心としたネットワークで有効に使うためには次のものを用意しました。

ネットワークカード

3000円くらいの安いカードでCE対応をうたったものを買いました。モバイルギアのマニュアルにはNEC製のLANカードしか使えないように書いてありましたが、NE2000相当のカードであればOSのドライバが対応しているようで、新たにドライバなどをインストールする必要はありませんでした。カードを入れると自動的に認識して必要な設定をするように促されます。私の職場の環境ではDHCPサーバーがあるので、TCP/IPの設定でアドレスをサーバーから取得とするだけでOKでした。家ではマシンの数が少ないので固定IPとしているので、家でLANに接続するときには別途設定するか、DHCPサーバーを立ち上げておく必要がありますが、当家でLANにつなぐ必要は感じていません。始めは消費電力の小さなカードを買おうと探していましたが、結局LANにつなぐ環境ではACアダプタが使えないわけがなく、カードの消費電力は実は大した問題ではないものだと分かり、割り切ってしまうと値段の安さだけで選んでよさそうです。カードを買うために店頭の商品を比較していて気がついたのですが、最近のカードではLinux対応を明示したものも多いですね。ちょっと驚きでした。

SambaをインストールしたLinuxマシン

CEのデバイス名の設定がデフォルトのHandPCだとネットワークに接続できずちょっと戸惑いましたが、固有の名前を入力することで解決しました。Sambaの設定でホームディレクトリを共有できるようにして、エクスプローラから

```
\\Server\username
```

のように自分のログイン名称の共有名をディレクトリとして指定するとパスワードを聞いてきて、きちんとSambaのディレクトリが読み込めました。もちろん、コピー、ペーストによってファイルの移動も簡単にできます。95や98でマウスのドラッグによるファイルの移動になれている目からはいちいちコピーしてコピー先でペーストするという動作はまだちょっといい感じがするかもしれませんが。私ば「コマンドラインが使えればもっと早いになぁ」と思っています。

寺崎氏のTeraTermCE

これを入れることで普段使っているマシンにtelnetで接続できて、ほとんど完璧な携帯端末になっています。このソフトのインストールはLinuxマシンの自分のホームディレクトリに<http://www.vector.co.jp/>から入手したパッケージを展開して行きます。付属のREADMEはとりあえず無視して次のようにしました。エクスプローラからSamba経由でホームディレクトリにネットワーク接続します。プロセッサ種別ごとに分かれて展開したパッケージの中から自分のCEマシンのプロセッサに合わせたディレクトリが問題のものです。そのディレクトリから該当するプロセッサのteraterm.exeだけを選択し、エクスプローラの編集メニューからコピーを行います。次にCEのProgram Filesディレクトリにおいてペーストするとこのファイルがコピーされます。また、深いディレクトリにおいてあると不便なのでデスクトップにTeratermのショートカットを作っておきます。ショートカットはもう一箇所、Windowsディレクトリのプログラムのフォルダにも作っておきます。私はプログラムの下の通信のところで作りました。これによってスタートメニューからTeratermが呼び出せるようになります。このようにネットワークに接続されていれば簡単にファイルのコピーが可能なのでWindowsを使わなくてもLinuxとの連携が面倒なことはありません。もちろん、これは私がCEをPIMとして使っておらず、単なる文字エントリーマシンとしてしかみなしていないからで、PIMの同期などを考えればそれなりに必要なことも出てくるのでしょう。

R320のモノクロの小さな画面ではXは実用的でないのですが、それは求めすぎというものでしょう。もっと画面サイズの大きなCEではTSEのクライアントソフトを導入すること

で、TSEのXサーバーを実用的に利用する道もありそうです。
TCO削減にCEはいかが？

A パフォーマンスモニタの 利用方法

プログラムの性能を上げるためにはまず第一にプログラムの振る舞いを正確に把握する必要があります。よく行われる方法は時間の測定です。プログラムのどこにどれだけの時間がかかっているかは、プロファイルを取れば大体のところは分かります。ところが、時間の内訳についてはプロファイルとソースコードから推定するしかありません。正しく推定できれば良いのですが、キャッシュやパイプラインの複雑な振る舞いがあると、プログラムの字面だけではなかなか何が起きているかを知ることは難しくなります。

そういったところで活躍するものとして、ハードウェアモニタという回路がプロセッサに組み込まれていることがあります。Alphaでは初代の21064からパフォーマンスカウンタという名前のカウンタが複数用意されています。これらのカウンタはプロセッサ内で発生した、プログラム動作に重要なイベントの回数やクロック数をソフトウェアの制御によって選択してカウントすることができます。ハードウェアだけあってもそれを利用する方法がなければ仕方ありません。このカウンタはPALコードからは直接参照できますが、一般のプログラムからは直接は参照できないのです。これを有効に使うためにCompaqのUNIXでは「wrperfmon」という特別な命令をPALコードに用意しています。この命令はLinuxでも使えるものなのですが、21164のMILOでは長らくサポートされていなかったのが有効には使われていませんでした。しかし、このところSRMを使ってLinuxをインストールする人も多くなってきたようですし、Mikitaさんの新しいMILOが発表されたりと徐々に環境も整ってきたので、一度きちんと使い方をまとめておこうと思います。

特権状態でしか利用できないこの命令は、カーネルのモジュールとして実装してシステムコールで呼び出すのが正しい使い方です。21164のパフォーマンスカウンタは長いほうでも16ビットしかなく、あっという間にオーバーフローしてしまうので、普通は割り込みを併用してもっと多くのイベントを計数できるようにします。Compaqからリリースされている性能解析ツールIprobeも、カーネルモジュールにモニタ命令の呼び出しと割り込みの処理を任せています。割り込みをサポートするにはカーネルの割り込み部分をよく知らないといけないため、このモジュールをそのまま使って性能解析を行うのが簡単そうです。このカーネルモジュールの使い方を説

明する前に、性能モニタに使うことができる命令の動作を知っておく方が後の理解も容易になるので、性能カウンタの動作の簡単な説明の後、性能モニタ命令の説明をします。

性能カウンタの実装がプロセッサの世代によって異なるので、性能モニタの命令はこれに準じてプロセッサの世代によって異なるものとなっています。Alphaのアーキテクチャリファレンスマニュアルは、性能モニタの命令の詳細仕様まで出ていますが、付録にあるとはいっても、新しいプロセッサが出るごとにマニュアルが改定される必要があります。本来、こういった実装依存のものをアーキテクチャのマニュアルに載せるのはどうかと思います。もう少し、共通項だけを載せて、プロセッサ依存の部分はプロセッサの固有マニュアルに任せれば良いのと思います。もっとも一冊になっていた方が便利がいいのは確かですから、それが悪いとは言い切れません。ただ、一世代古いマニュアルを買った私には差分だけをWebで取れた方が嬉しいのです。

wrperfmon命令の概要

個別のプロセッサの話に入る前に、wrperfmon命令の使い方について概要を説明します。この命令は、特権モードでしか動作しません。ユーザーモードでこの命令が発行されると、命令デコードの例外になります。命令に対する入力データは「r16」、「r17 (a0, a1) のレジスタによって受け渡されます。命令の結果は「r0 (v0)」のレジスタに返ってきます。この命令の動作によって「t0」と「t8~t11 (r1, r22~r25) の各レジスタは値を保証されません。そこで、これらのレジスタを使用するプログラムでは前後に値を保存しておく必要があります。ちなみにPALのオペコードは16進で「39」となります。

余談ですが、PALには特権でない命令もあり、その場合には、オペコードは80以上の値となっています。代表的な非特権のPAL命令には「システムコール(83)」があります。これなどは一般ユーザーから出せないとい何の意味もないコールになってしまいます。wrperfmonを一般ユーザーで出せないのは、プログラムのデバッグに使いたい向きにはちょっと不便ですが、一般にはIprobeのドライバ経由で利用することになります。PALコードを書き換えて一般ユーザー用のwrperfmonのエントリを作ればもっと便利になりますが、SRMのPALの書き換えは一般にはできません。もちろん、MILOのように、独自のPALを用意しているインターフェイスがあれば、これを修正することはできるのですが、SRMで使うユーザーとの互換性が取れなくなるので、なるべくオフィシャルなインターフェイスを守るべきでしょう。Iprobeのドライバの中では、wrperfmonのPALコールをリスト1のようにインラインアセンブラで記述しています。

EV4におけるwrperfmon命令

機能コードの異なる4種類の命令があります(表1)。

機能コード2と3には引数がつきます。引数はどちらもa1のレジスタに設定します。機能コード2の引数は、レジスタのビットごとに表2のように割り当てられます。

機能コード3の引数は同じくa1に設定しますが、a1が0の時

リスト1

```
static inline unsigned long
wrperfmon(unsigned long perf_fun,
           unsigned long arg) {
register unsigned long __r0 __asm__("$0");
register unsigned long __r16 __asm__("$16");
register unsigned long __r17 __asm__("$17");
__r16 = perf_fun;
__r17 = arg;
__asm__ __volatile__(
"call_pal %1"
: "=r"(__r0)
: "i"(PAL_wrperfmon), "r"(__r16), "r"(__r17)
: "$1", "$22", "$23", "$24", "$25", "$26");
return __r0; }
```

表1

機能コード	説明
0	性能モニタのディスエーブル
1	性能モニタのイネーブル
2	モニタイベントの選択
3	モニタプロセスの制御

表2

34-32 カウンタ1のイベント選択	
値	機能
0	Dキャッシュミス
1	Iキャッシュミス
2	2命令発行サイクル
3	分岐予測失敗
4	浮動小数点演算
5	整数演算
6	ストア命令
7	外部イベント
11-8 カウンタ0のイベント選択	
値	機能
0	命令発行回数 / 2
2	パイプラインドライ
4	ロード命令
6	パイプラインフローズン
8	分岐命令
10	トータルサイクル
11	PALモードサイクル
12	命令発行数のないスロット / 2
14	外部イベント
3 カウンタ0の割り込み頻度設定	
値	機能
0	65536回毎割り込み
1	4096回毎割り込み
0 カウンタ1の割り込み頻度設定	
値	機能
0	4096回毎割り込み
1	256回毎割り込み

にはすべてのプロセスをモニタの対象にするのに対して、1の時にはプロセスコントロールブロック中のPMEビットが立っているプロセスのみがモニタの対象とされます。これによって特定プロセスのみ性能を観測することができるようになっています。

EV5/EV56におけるwrperfmon命令

機能コードの異なる8種類の命令があります(表3)。

機能コード0、1、2の引数はカウンタの番号を引数のビットに対応させたビットマップになっています(表4)。

となっています。

機能コード2の引数はビットごとにフィールドに分かれていて表5のようになります。

機能コード3の引数はカウンタが動作するプロセスを絞り込むために使います。ビットマップであらわされます。

機能コード4はカウンタの割り込み頻度を制御します。引数は3つのフィールドに分かれています(表7)。

機能コード5と6はカウンタのリードライトの命令ですが、どちらもカウンタの本体は以下のビットを用います。リードだけは読み込みの成功失敗をやるためのSTATUSというビットがビット0にあります。これを除けば同じ並びを用います。

63-48 CTR0
47-32 CTR1
29-16 CTR2

lprドライバ

前述のように特権命令を使えない一般ユーザーが性能カウンタを使うには、ドライバを呼び出すしか方法がありません。キャラクタの特殊デバイス「lpr」のドライバを使うlprobeでは、ドライバの呼び出しはすべてioct1のシステムコールにて行います。このioct1のコードは27種類用意されています。サンプルコードについての解説は、次号でしたいと思います。

表3

機能コード	説明
0	性能モニタのディスエーブル
1	性能モニタのイネーブル
2	モニタイベントの選択
3	モニタプロセスの制御
4	割り込み頻度制御
5	カウンタ読み出し
6	カウンタ書き込み
7	カウンタクリアつきモニタイネーブル

表4

ビット	カウンタ番号
0	カウンタ0
1	カウンタ1
2	カウンタ2

表5

31 カウンタ0のイベント選択	
値	機能
0	サイクル数
1	命令発行数
7-4 カウンタ1のイベント選択	
値	機能
0	命令発行なしのサイクル
1	一部のスロットのみ命令発行
2	パイプラインドライ
3	リプレイトラップ
4	1命令のみ発行のサイクル
5	2命令のみ発行のサイクル
6	3命令のみ発行のサイクル
7	4命令発行のサイクル
8	命令フローの変更、ただし、カウンタ2の選択が分岐予測失敗の時には条件分岐、同PC予測失敗の時にはサブルーチンリターンを計数
9	整数演算命令
10	浮動小数点演算命令
11	ロード命令
12	ストア命令
13	命令キャッシュアクセス
14	データキャッシュアクセス
15	CBOX1の選択結果の利用
3-0 カウンタ2のイベント選択	
値	機能
0	長期間ストール(13サイクル以上)
2	PC予測失敗
3	分岐予測失敗
4	命令キャッシュミス
5	命令アドレス変換バッファ(ITB)ミス
6	データキャッシュミス
7	データ変換バッファ(DTB)ミス
8	ロードマージ(MAFにマージ)
9	LDUリプレイ
10	WB/MAFフルリプレイ
11	外部イベント
12	サイクル数
13	メモリバリア
14	ロックつきロード命令
15	CBOX2に従う
21-19 CBOX1	
値	機能
0	Sキャッシュミス
1	Sキャッシュ読み込み
2	Sキャッシュ書き込み
3	Sキャッシュビクティム
5	Bキャッシュヒット
6	Bキャッシュビクティム
7	システムリクエスト
24-22 CBOX2	
値	機能
0	Sキャッシュミス
1	Sキャッシュリードミス
2	Sキャッシュライトミス
3	Sキャッシュ共有ライト
4	Sキャッシュライト
5	Bキャッシュミス
6	システムインバリデート
7	システムリードリクエスト

表6

ビット	動作
0	0の時全プロセスを対象、1の時PMEが立つプロセスのみを対象とする
8	カーネルモードでカウンタストップ
9	PALモードでカウンタストップ
30	ユーザーモードでカウンタストップ

表7

9-8 カウンタ0の頻度制御	
値	機能
0	割り込み無し
2	65536イベント毎割り込み
3	256イベント毎割り込み
7-6 カウンタ1の頻度制御	
値	機能
0	割り込み無し
2	65536イベント毎割り込み
3	256イベント毎割り込み
5-4 カウンタ2の頻度制御	
値	機能
0	割り込み無し
2	16384イベント毎割り込み
3	256イベント毎割り込み

A おわりに

Compaqがベータリリースしたコンパイラについて私に質問をしてきた方が何人いらっしゃいましたが、このベータ版については問題点を発見したらCompaqに報告することになっています。まずはCompaqにコンタクトをとっていただきたいと思います。

初めてWindows CEを導入して慣れない端末で打った割には、比較的短時間で、それなりの速さで入力できるようになりました。Linuxとの連携も悪くなくCEはもっと注目されてもいい機械だと思っています。

使い方のターゲットを絞れば、CEでなくてもLinuxでも同じくらいの軽いマシンはできる気がします。しかし、vi系のエディタがないと私は若干苦しいので、慣れてきたとは言ってもCEだけでは厳しいものがあります。Vim + SKKの環境がCE並みのマシンとして出てくれば触手が動くかもしれません。だれか作ってくれませんか？