Linux/Alpha活用講座

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

第20回
Alphaのネットワークブート

まず始めにお詫びしなくてはならないのですが、6月号でベンチマークのグラフを出したのですが、編集部との行き違いからグラフの機種情報や横軸の意味が間違っていました。校正時に修正をお願いしたのですが、記事に反映されていないのでわけの分からないグラフになってしまいました。各グラフの情報は、正しくは次のようになります(編集部注：読者のみなさん、および筆者の清水先生へ、この場を借りてお詫び申し上げます。正しいグラフを再掲します)。

グラフ1機種情報:
上からXP1000(LAPACK)、XP1000(CXML)、VT5-600(LAPACK)、VT5-600(CXML)になります。

グラフ2、3横軸:
単位は「倍」になります。実時間の何倍速でエンコードできたかということを表します。ですから数値が大きいほど高速ということになります。

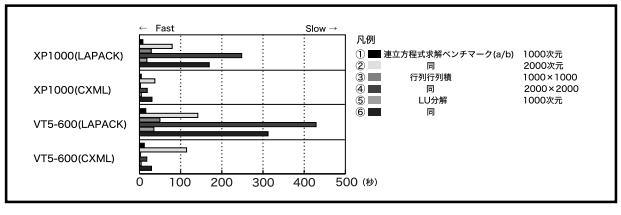
もう一点、こちらは私の勘違いですが、SHMMAXの大きさが24bitに制限されていると書きましたが、実は制限されているのはＳＨＭＭＡＸの上限を決めるための別の変数で、SHMMAX自身はもっと大きな値まで大丈夫でした。現在の設定値ではSHMMAXは256MBytes程度取れるはずです。カーネルをリコンパイルしなくても/proc/sys/kernel/shmmaxに数値を書き込めば上限は大きくなります。ただし、ここへの書き込みは値をチェックしていないようなので、無茶な値を書かないように注意が必要です。

目まぐるしく変わっていくこの分野ではちょっと油断するとすぐに過去の人になってしまいます。やりたいと思ったことを後回しにしていると、誰かに先を越されてしまい悔しい思いをしますね。Windows CEのマシンを持ち運び用の端末として使っている私は、これにLinuxが移植できたらいいなあと漠然と思っていましたが、LinuxやNetBSDがしっかり移植されているのには驚きました。昔は手さぐりでLSIの仕様を探りながらデバイスドライバをチューニングしたものですが、今ではWebの普及もあって資料の入手は容易になって結果的にデバイスドライバの開発なども簡単になりつつあります。Linuxの多くの機種への移植が進んでいくとプロセッサごとの個別の部分が整理されて、ソースコード自体はすっきりしてくることでしょう。

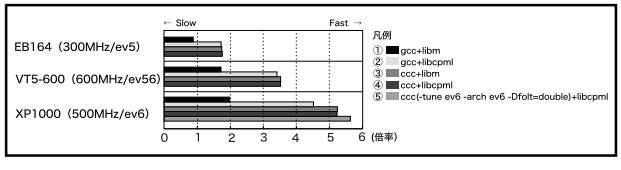
ぜひやってみたいと思っていることに、自分の考えたアーキテクチャにLinuxを移植してワークステーションに仕立てることがあります。こういった対象としてもLinuxは面白い素材で

2000年6月号のグラフ(再掲)

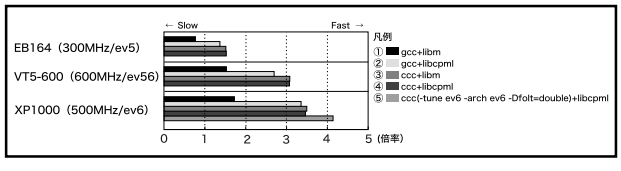
グラフ1　ベンチマークテスト



グラフ2　MP3エンコード テスト①(倍率は「実演奏時間÷エンコード時間」)



グラフ3　MP3エンコード テスト②(倍率は「実演奏時間÷エンコード時間」)



す。昔なら箱を作ってもソフトがないから動かせないなんてことになったのですが、今ならちょっと(?)の努力できちんと動くものを作り上げられる技術的な素地ができています。

A 最近のLinux/Alphaのニュース

さて、6月号では最近のニュースを書くことをすっかり失念していました。というわけで今月は2カ月分のニュースをお送りします。

・1Uと呼ばれる、薄型のラックマウントのサーバータイプのマシンがアナウンスされました。これはSlateと呼ばれています。薄型のラックマウントのサーバーを積み重ねてVirtualServerで負荷分散することで高性能な分散システムが簡単に構成できます。

・MILOの新版がリリースされました。従来のMILOは2.0.xのカーネルをベースにしていたのですが新しいMILOは2.2.xのカーネルをベースに構成し直されています。

・COMPAQのC++コンパイラのβリリースが出ています。おそらく今まで出てきたCやFORTRANと同等のコード生成のロジックを使っていると思われるので性能的にも楽しみですね。とはいっても私自身はC++はあまり使っていないのでベンチマークをするほどのネタを持っていないのが残念です。

・RTLinuxのAlphaへの移植が計画されているようです。高性能な数値計算ができるマシンとリアルタイムOSの組み合わせはなかなか面白い応用を生むかもしれませんね。

・SRM Howtoの新版が出ました。SRMのマシンをインストールする機会はどんどん増えていくと思いますので情報が適宜アップデートされるのは嬉しいですね。

・Jetと呼ばれる大規模なクラスタが作られています。今はXP1000を276台接続していますが、今年の夏の終わりにはクラスタの規模を倍にするようです。

・Windows NTのバイナリ最適化プログラムであるSpikeのLinux/Alphaへの移植が始まりました。バイナリコンパイラを手がけてきたCOMPAQの研究者達にとって、コンパイルできるなら最適化もできるというのは自然な流れだと思います。もっとも多くのプログラムのソースコードの入手が容易なLinuxの世界では、バイナリの最適化がどこまで役に立つのかは良く分からないのですが、技術的には面白い試みですね。

・Samsungが今年中に1.4GHzから1.6GHzのAlphaプロセッサを出すという記事が出ています(画面1)。1GHz競争に思わず不覚をとったAlpha陣営としてはインパクトのある周波数を早く出さないといけないのでしょうね。クロック周波数だけではなくトータルの性能として考えないと意味がないのですが、ユーザーは見た目の数値に惑わされやすいものです。

A Kondara MNU/Linux

せっかく購入したパッケージだから使ってみない手はないと、Kondara MNU/Linuxのインストールを始めましたが、これが一筋縄ではいきません。そもそも私が古いマシンしか持っていないからなのかと疑っていますが、Cabrioletと呼ばれるAlphaPC64ではMILOからロードしたインストーラがKonadara MNU/LinuxのCD-ROMを認識してくれません。SRMに乗せ換えたEB164ではCD-ROMがブータブルとなっていないように見えます。これではちっとも記事が進まないのでブータブルなCD-Rを焼くことにしました。まずはKondaraのCD-ROMをイメージとして読み出します。

```
# dd if=/dev/scd0 of=kondara.img
```

次にこのパッケージのabootがどこにあるかをマウントして確認しておきます。

```
# mount /dev/scd0 /cdrom -o iso9660,ro
# find /cdrom -name bootlx -print
```

すると/boot/bootlxが見つかります。次にCD-ROMをブート可能にするために次のコマンドを打ちます。

```
# isomarkboot kondara.img /boot/bootlx
```

ここまでくればしめたもので、このイメージをCD-Rに焼いてしまいます。こうして作ったCDはちゃんとブートできます。CD-ROMからブートさせた後で、実行例1のようなコマンドを

画面1

SRMから入力しましょう。ここではCD-ROMのSCSI IDが4のシステムを想定しています。デバイス名が分からない人はSRMから“show dev”コマンドで確認できます。

ここまでは順調でしたが、ルートをマウントできずブートが失敗しました。原因を探ると、どうやらCD-ROM上の/devのディレクトリが空になっていることが問題だろうと見当をつけましたが、具体的な対策はまだ行っていません。もしかしたらオプションが違うだけなのかもしれません。それではブートフロッピーディスクからブートできるかとやってみたら、こちらもSCSIのCD-ROMを認識できず失敗するようです。カーネルにSCSI CD-ROMのサポートが入っていないのだろうと思います。せっかく購入したKondaraのパッケージだからなんとかしようということで、CD-ROMの中身をすべてハードディスクの1パーティションにコピーして、ハードディスクからインストールしました。始めからIDEのCD-ROMを搭載している最近のシステムではこのようなことはないと思います。

さて、インストールはできたのですが、とりあえずKondaraを使う必然性はないので、インストールしただけで日常的には使ってはいません。ここまで書いたところで気がついたのですが、CD-ROMやブートフロッピーディスクの中にあるカーネルを書き換えることで、通常のインストール手順で行けるはずですね。もう一度やり直す元気はでないので、今回は奮戦記だけとさせください。

AEV4での注意点

EV4のユーザーはもうほとんどいないかもしれませんが、私のところでは現役で使用しています。パッケージを入れ替えて以降、EV4のマシンがどうも不安定になっているのでいろいろと調査中ですが、カーネルのバグがあることがlinux-kernelメーリングリストで報告されています。EV4で実装されていない命令をエミュレートするときに、プログラムカウンタの更新を誤って無限ループになる現象が発生するのです。取りあえずパッチをあてて様子を見ていますが、執筆時点では大丈夫そうです。このパッチを使ってみたい方は次のURL

```
http://boudicca.tux.org/hypermail/linux-kernel/
2000week09/0381.html
```

を参照してください。2.2.14のカーネルでは/usr/src/linux/arch/alpha/kernel/traps.cの420行目付近になります。EV4以外のシステムには悪さをしないはずなので、カーネルの修正をしても構わないでしょう。2.2.15にはこのパッチが取りいれられることになりそうです。EV4はMultiaやAXPPci33やJensen、Cabrioletなどといったマシンですが、現役で使っている人は少ないかもしれません。今のx86に比べても性能は良くないので、今さら使う人もいないと思いますが、とにかくAlphaのバイナリが動作する環境がほしい人などにはリースバック品が安く手に入ると思われるため、結構需要があると思います。特にAlphaStation 200 4/100などは安価に新品が出回ったため、私ですら計3台も所有しています。そのうちの2台はNFS、NIS、WWWサーバーとそのバックアップシステムとして使っています。

Aネットワークブートの実験

SRMにはネットワークブートの機能があるので、これを使えばディスクレスのクラスタが簡単にできます。BIOSがネットワークブートに対応していないx86のマシンでは、ディスクレスを実現するのにEthernetカードにブートROMを搭載する必要があるのですが、SRMには標準でついてくるのでお得な感じですね。これを実現するには、サーバーとなるマシンを用意して、bootpとtftpの設定をする必要があります。今のネットワークは高速だし、スイッチングハブも信じられないほど安価になっているので、クラスタ以外の通常のマシンの利用にもメンテナンスが容易なネットワークブートを活用しない手はないですね。

bootpの設定はサーバーとするマシンによるので詳細は省かせていただきますが、リスト1のようなbootptabを使いました。ただし、アドレス等はこのまま打ちこまないで各自のマシンと環境に合わせてください。このbootptabでMultiaも

リスト1

```
.default:\
    :td=/tftpboot:bf=bootpfile:\
    :ht=ethernet:hn:vm=rfc1048:\
    :ds=xx.xx.xx.ss:sm=255.255.255.0:\
    :dn=my.domain.gr.jp:gw=xx.xx.xx.tt:bs=auto:

myhost:ha=yyyyyyyyyyyy:ip=xx.xx.xx.uu:tc=.default:
```

実行例1

```
boot dka400 -fi kernels/generic.gz -fl "root=/dev/scd0"
```

EB164もブートできたのですが、Multiaはカーネルのバージョンかコンパイルオプションに気を付けないと、ブートの途中で失敗してリブートするようです。今のところ、SRMの環境変数でブートファイルを指定してMultiaだけ別のイメージをブートしていますが、もう少し調査が進めばジェネリックなカーネルでブートする方法も見つかると思います。

ネットワークブートのためのカーネルはabootのパッケージで

```
# make netboot
```

として作成しますが、Debianユーザーでabootをインストールしてあればもっと簡単に

```
# cat netboot.nh vmlinux.gz net.pad > vmlinuz.bootp
```

のように作成できます。もしくはカーネルのソースコンパイル時に

```
# make bootpfile
```

とすると圧縮していないので大きめになりますが、ネットワークブートに対応したカーネルが

```
/usr/src/linux/arch/alpha/boot/bootpfile
```

としてできます。どちらでもお好きなほうを使ってください。

カーネルの構成時に注意するべき点として、Multiaではフレームバッファを用いるTGAがグラフィックカードとして実装されているということです。ですから、コンソールにVGAだけでなく、必ずフレームバッファの項目をチェックして、TGAを使えるようにしてください。私はこれを忘れてしばらく悩んでいました。

また、常時インターネットに接続している環境で完全にネットワークからシステムをインストールしたい場合には、MILOの改良に精力的に取り組んできたNikita Schmidtさんの作成したネットワークインストール用のブートイメージを使うことができます。私はフロッピーディスクの壊れたMultiaをこの方法でインストールしてみました。Debian/のpotatoでは、一切、フロッピーディスクもCD-ROMも使わずにインストールができます。どうせなら本当のディスクレスにしても良いのですが、そこまではまだ必要ないので、取りあえず内蔵ハードディスクドライブにネットワークからインストールしてみました。ブートイメージは次のURLからダウンロードできます。

```
http://www.debian.org/Lists-Archives/debian-alpha-0003/msg00056.html
```

まだテスト版なので、インストールのネットワークインストールの途中で不安定になるところがあったりしましたが、NFSと併用して、無事Multiaのネットワークブートを立ち上げました。340MBytesしかないMultiaの標準ディスクは、まともにインストールするとすぐにいっぱいになってしまいますから、必要最低限のものだけしかインストールしていません。このように、カーネルだけをネットワークブートすることにどういうメリットがあるのだろうと疑問に思われる方もいるかもしれませんね。実は結構多くのメリットがあるのです。

まずは以前の記事に書いたようにWindows NTとの共存を図るために、ハードディスクをFAT形式のパーティションとしていたことがありました。MultiaやEB164のようにMILOが供給されているマシンでは、FAT形式でもあまり困らないのですが、XP1000のようにMILOがない場合には、Linuxはどうしてもsrmから立ち上げる必要があったのです。以前に記事に書いたときには、この対策として、フロッピーディスクベースのブートをするようにしていましたが、信頼性の低いフロッピーディスクに頼るのもなんとなく不安ですし、第一、使い勝手が良くありません。一時カーネルをブータブルなCD-Rに書き込んで使っていくという手もありますが、CD-Rでは、カーネル再構築のたびにいらないCD-Rができるわけで、貧乏症の私には精神衛生上あまりよくありませんでした。

こんな場合は、ネットワークブートならカーネルの更新も楽々できるわけですし、フロッピーディスクの信頼性やCD-Rの扱いの不便さに煩わされることもありません。

次に、これもMultiaのように古いマシンを使うと顕著な点です。マザーボード上にIDEのコネクタがついていてIDEのドライブを付けられるようになっているマシンが多いのですが、せっかく接続してもIDEからはブートできないマシンが多いのです。そんなときにもLinuxのカーネルさえ何らかの方法で立ちあがってしまえば、カーネルにはIDEのドライバを組み込めるわけで、普通に利用することができます。SRMやARCのコンソールからブートできてMultiaに内蔵できるのは2.5インチのSCSIドライブです。これはPowerBookで使っていたタイプなのですが、こんなマイナーなドライブはすでに入手が難しいし、あっても値段が高くなっているはずです。ところが、SRMからは認識されないものの、IDEのドライブも物理的には実装できるのです。IDEならばノートパソコン向きに大容量の2.5インチドライブもいろいろと入手できます。さらにMultia以外でもフロッピーディスクの調子が今一つというマシンをお持ちの方も多いのではないでしょうか？

何年も前に、ニュースで、コンピュータのコストダウンのために、あまり利用されていないフロッピーディスクの信頼性の

基準を各メーカーが下げているという記事を読んだことがあります。そのときには日本ではまだまだフロッピーディスクの利用は多かったはずなので違和感を感じましたが、やはりネットワークの時代にはフロッピーディスクは不要なのですね。

余談ですが、サスペンス小説の「ハンニバル」にはZIPドライブの話が出てきます。日本だったらフロッピーディスクかMOになるのでしょうね。CD-ROMもATAPIのものなら安価でもSCSIのものは高価だったりします。なんて私の個人的な状況ばかりですね。今どきの皆さんは、こんなマシンはさっさと買い換えているのかもしれません。

もう1つのメリットは、こちらも以前記事にしましたが、IprobeなどのMILOでは対応していないPALコードを用いたプログラムを使うことが可能ということがあります。

そのうちそのうちと先延ばしにしてなかなか性能測定のデバイスドライバの開発が進みませんが、SRMのPALに定義されているwrperfmon命令を利用すると、プログラムのチューニングが容易になります。測定手段が良くなることによって技術が進歩するというのはどの分野でも同じですね。

さて、次にネットワークブートするためのSRMのコマンドです。ネットワークインストールの場合と通常のLinuxの立ち上げの場合がありますが、どちらも基本は同じです。まずはブートデバイスをネットワークカードに設定します。SRMが認識できるカードはtulipと呼ばれるDEC 21x40系統のものです。Multiaなどはオンボードのインターフェイスをそのまま使います。新しい世代のSRMではIntel EtherExpress Pro100/Bもサポートされているようです。ネットワークカードが1枚だけで、そのカードからブートする場合のブートデバイスは`ewa0`となります。ブートデバイスの環境変数にこのカードを指定しましょう。

```
set bootdef_dev ewa0
```

次にブートプロトコルにBOOTPを指定します。デフォルトは異なるものになっています。

```
set ewa0_protocol bootp
```

自動ブートをする場合には、ブートオプションを環境変数で設定してしまいます。たとえばルートデバイスを`/dev/sda2`としてBOOTPで与えられたファイルをブートする場合には

```
set boot_osflags "root=/dev/sda2"
```

とします。ブートファイル名を明示的に指定したいときにはこれに加えて

```
set boot_osfile "bootpfile"
```

のように指定します。自動ブートでなければこれらの指定はコマンドラインからできます。デバッグ中は勝手に指定されると困ったりするので、環境変数を"`""`"にしてしまってすべてコマンドラインから指定にした方がいいでしょう。また、前記のブータブルなインストール環境を用いるときには

```
b ewa0 -fi "tftpboot.img" -fl ""
```

としました。ブートフラグは勝手に"`root=/dev/ram`"が追加されるようになっていますから、これでブートするといきなりインストーラが動き出します。

内蔵ハードディスクを持つマシンで、カーネルだけネットワークブートすることで、ディスクレスのための大枠は経験したことになります。そこで、この一連の作業で一通りの経験を積んだら、次は完全ディスクレスへ挑戦してみましょう。完全ディスクレスのメリットは、カーネルだけリモートブートするよりもっといろいろあります。

- ディスクを一カ所に集中することでメンテナンスを容易にできます。最近のLinuxではマシンごとの個別の設定が必要なディレクトリやマシンごとに別々に書きこみが発生するようなディレクトリはまとめられています。なので、/usrを始め、かなりの部分を共有するすることで、1つのマシンを設定したら他のマシンはそれを参照するだけでよくなります。
- トータルの投資額を低くすることができます。当然ですが、同じファイルを格納するためにディスクを用意する必要がなくなります。
- これもメンテナンスに類することですが、不用意なユーザーにいきなり電源を切られるようなことがあっても平気になります。自分だけが使うときには気にならなくても多くの人と共有するといろいろな人がいるものです。

様々な利点がある完全ディスクレスのLinuxですが、長年にわたって雑多なマシンが入り乱れている私の環境に導入するためには慎重な計画を立てなくてはならないので、設定の詳細は次号に回したいと思います。冒頭のニュース記事で紹介したSlateをラックに組んで、ディスクレスのクラスタを作れば、相当大規模なクラスタもコンパクトに実装できますね。AlphaSC相当の性能をEthernetだけで出すのは大変だと思いますが、応用によっては十分な分野もあるはずです。これからAlphaのクラスタを立ち上げようとしている人にも役に立つことを願っています。