Linux/Alpha活用講座

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

第12回
XP1000導入

私の大学では、毎年夏にメンテナンスのために停電があって、研究室のサーバー(AS200)もそのときには停止します。今年もshutdownをatコマンドで入れておいて安心して休みを取っていたら、電源が回復したはずの日になってもサーバーにアクセスできません。大学に来てみてびっくり、サーバーの電源が壊れていました。やはりファンが止まるくらいの年月がたっていれば電源だって寿命がきているのでしょう。もしかしたら、メンテナンスで電源におかしな波形が入力されていたのかもしれません。いずれにせよサーバーが動かないと困るので、学生が主として使っていた同じ機種から電源を抜いて付け替えました。無事動いたのはいいのですが、早急に電源を手配しなくてはいけませんね。

さて、CompaqからLinux用のC言語コンパイラと拡張数値計算ライブラリ(CXML)のリリースが案内されています。Cのβテストは執筆時点では始まっていませんが、CXMLはβテストが始まっていて、Fortranと同じページhttp://www.digital.com/fortran/linuxからダウンロードできます。そこからたどって登録するとダウンロードのURLをメールで送り返してくるのはFortranコンパイラと同じです。CXMLにはLAPACK互換の行列演算ライブラリやFFTなどの信号処理ライブラリなどが入っており、実用的なプログラムを作りたいユーザーにとって便利なライブラリだと思います。

LAPACKについてですが、もっと詳しく知りたい人は以下のURLを見てください。

http://www.netlib.org/lapack/lug/lapack_lug.html

XP1000がやってきた

Compaq社の御協力で私の所にもAlpha21264を搭載したマシン「XP1000」が導入できました。到着したマシンの仕様は次の通りです。

CPU: 21264-500MHz
RAM: 256MB
HDD: 9GB(4.5GBはNTをプリインストール)
Video: ELSA GLoria Synergy
Network: 21143 100Base-T

HDDはSCSI接続なのですが、SCSIの外部コネクタはなく、外部に機器を増設するにはカードも増設しないとだめのようです。NTがインストールされていることからも分かるようにDOSタイプのパーティションの区切りになっています。

到着したXP1000を飾っておいても仕方がないので、例によってDebianのインストールをします。今回は出始めのマシンなので、いろいろなOSで十分な対応が取られているわけではなく、Debianのインストール時にも多少問題がありました。分かってしまえばなんだそんなことだったのかと思えるほど単純な問題ですが、インストール時の手順がやや複雑になります。XP1000だけでなく多くのマシンにも同じ手順でインストールできるケースが多いので、何度も出て来て申し訳ありませんが、今回もインストールの話を簡単にします。そのかわり、今回はXP1000に特徴的な所だけに絞って話をしたいと思ってい

ますので、ご容赦ください。グラフィックカードの「ELSA GLoria Synergy」は今のところ商用Xサーバでしかサポートされていません。METRO-Xのサイトでオンライン発注をしようとしたのですが、クレジットカードの利用期限がホームページから入力できる範囲になかったので発注が失敗してしまいました。時間がないので今回はXなしのワークステーションとして利用します。

●コンソールBIOSの切替え

XP1000を始めとして、いくつかのAlphaプロセッサ搭載ワークステーションには、NTをブートするためのBIOSである「ARC」か「AlphaBIOS」が搭載されています。NTであればCD-ROMから簡単に導入できますが、LinuxではTru64 UNIXと同等のファームウェアを必要としているため、ちょっと特殊な方法が必要となります。特にファームウェアであるPALコードはマシン毎に異なるので、インストールの準備段階が一番難しいのです。マシンによっては「MILO」という独自のPALコードを搭載したローダーが用意されていて、ARCやAlphaBIOSのOSのセレクションメニューからLinuxを選択できるようになっています。しかし、XP1000は新しい機種のため、公式にはMILOは対応していません。そこで、「MILOなし」でのLinuxの起動に挑戦します。

MILOは独自のPALコードを搭載していると書きましたが、実はこのPALコードの提供する機能そのものは、CompaqのTru64 UNIXの要求と同じです。つまり、MILOによってARCやAlphaBIOSのようにNT用のサービスしかないマシンにもLinuxを導入できるようになったと言えるでしょう。

さて、XP1000に対応したMILOがないので、NTと同居させるのは難しいと思われた方もいるかもしれません。実にその通りで、NTのBIOSでは、MILOがないことには最初のブートディスクから起動させることさえ困難です。ところが、XP1000はTru64 UNIX用のSRM BIOSも搭載している点が他機種とちょっと違います。このSRMならTru64 UNIXのみならず、適切なオプションでコンパイルされていればLinuxもブートすることが可能となります。SRMのPALコードが機種毎の違いを吸収してくれるので、Linuxのカーネルはgeneric用のものをそのまま使うことができるのです。

そこで、インストールの第一段階はBIOSをSRMに切替えることから始まります。しかし、通常のマシンでは、メニューを検索してもボードのメニューからBIOSの切替え設定の項目は見付かりません。実は、この項目はAlphaBIOSの奥深くに隠れています。SRMへのBIOSの切替えを行うには次のようにします。

1. BIOSがメモリカウントを終るころに、F2キーでセットアップモードに入る。
2. CMOSセットアップのメニューを開く
3. F6キーでAdvanced setupのページを開く
4. ブートBIOSのタイプをSRM(Tru64)に設定し直す。
5. F10キーで設定をセーブして電源を入れ直します。

これで無事にSRMのBIOS画面になると思います。ならない場合はもう一度上記の手順を見直してみてください。

●Debianのインストール

SRMへの切替えが出来れば次のステップはLinuxのインストールになります。これには2枚のFDを用意します。1枚はDebianプロジェクトからroot1440.binをダウンロードしてこれをフロッピーディスクに書き込みます。Windowsしか使えない人はRAWRITE.EXEというユーティリティで作成してください。Linuxではddコマンドで作成すればよいでしょう。次の1枚は、次のURLからダウンロードしたイメージを使います。

```
ftp://gatekeeper.dec.com/pub/Digital/Linux-Alpha/Images/generic-up-223.img
```

こちらも同様にrawriteかddコマンドなどでFDに書き込んでください。こちらのイメージがブート用のカーネルとなります。これらの準備が整ったら、ブートディスクをドライブに入れて、次のように環境変数を設定してブートしましょう。

```
>>>set bootdef_dev dva0
>>>set boot_osflags "root=/dev/fd0 load_ramdisk=1"
>>>set boot_file vmlinux.gz
>>>boot
```

RAMディスク用のFDを要求してくるのでroot1440.binをコピーしたFDに入れ換えてEnterキーを押してください。これで、Debianのインストール画面まで立ち上がって

くるはずです。

後は通常のDebianのインストールと同じとなればいいのですが、実はまだ少し工夫が必要です。というのは、Debianではresc1440.binというイメージにカーネルとモジュールが入っているのですが、これがインストール時に要求されます。今回のインストールでは、これらは使わないし、SRMはNTタイプのパーティションは扱えないので、もし使えたとしてもこちらを使ってのインストールはあまり意味をなしません。そこで、カーネルとモジュールのインストールをスキップして、メニューから1つずつ手動でインストールを継続することになります。

以上で、Debianのベースシステムのインストールは完了します。後はネットワークから適当なパッケージをダウンロードして自分なりのワークステーションに仕立ててください。

最後にもう一度シャットダウンしてSRMに戻って次のように環境変数を設定しましょう。

```
>>>set boot_osflags "root=/dev/sda?"
>>>set auto_action boot
```

「sda?」のところはインストールの方法によって変わります。私は「sda6」にしていますが、自分のシステムに合わせてください。

実はSRMを使うと、カーネルをネットワークからロードすることができるので、適当なファイルサーバーさえ用意できれば、ディスクレスの構成が簡単に作れます。SRM経由のディスクレスブートは稼働部分の全くないノードを作れる点で優れています。後で紹介するCPlantはこの機能を利用しています。

●Compaq Fortran β版のインストール

例によってCompaq Fortranのβ版をインストールします。EV5用のCPMLはシェアードライブラリを自動生成するように変更されていたのですが、なぜか私のダウンロードしたEV6用のCPMLでは、シェアードライブラリの自動生成が出来なかったので手動で行います。それ以外のインストール方法に付いては本誌10月号を参照してください。EV6(21264)用のCPMLのパッケージを展開すると/usr/lib/compaq/cpml-0.2のディレクトリの下にlibcpml_ev6.aというファイルが作成されます。ところが、ここに作られるべきlibcpml_ev6.soは作成されていませんでした。前に問題にしていたインストールの失敗は、このファイルがないことが原因でした。要するにこのファイルさえできればいいので、手動で作ることに挑戦です。次の手順を踏めば作ることができます。

1. アーカイブの展開

```
# mkdir obj
# cd obj
# ar x ../libcpml_ev6.a
```

2. シェアードライブラリの作成

```
# gcc -shared -o ../libcpml_ev6.so -Wl,
 -soname,libcpml.so *.o
```

3. 一時ファイルの削除

```
# cd ..
# rm -f -r obj
```

これで無事に必要なライブラリが揃いました。ここで、10月号の方法に従ってCPML以外のFortranのパッケージをインストールしてください。それが出来たらfortコマンドでコンパイルしたバイナリが正しく動作することを確かめてください。

●XP1000の性能

性能については、後でもう少しまとまった記事を書きますが、10月号の続きとしてヒメノベンチの結果だけ簡単に報告しておきます(リスト1)。

コンパイラにもよりますが、37%程度の性能向上が得られています。これを10月号で報告した21164A-600MHzと比較するとCompaq Fortranで約5割向上、G77で4割強の性能向上になります。21264が多くのアプリケーションで21164Aの2倍近い性能を出すと言われていることと比較すると若干控えめな性能ですが、これはヒメノベンチがメモリのスループットにものすごく高い性能を要求していて、XP1000がCompaqの21264のシリーズの中で

リスト1

```
--------------- Compaq Fortran ----------------------
  mimax= 129  mjmax= 65  mkmax= 65
  imax= 128  jmax= 64  kmax= 64
 cpu :    1.331055     sec.
  Loop executed for           13  times
  Gosa :  3.0001516E-03
  MFLOPS measured :   160.8349
  Score based on MMX Pentium 200MHz :   4.984037


---------------- G77 Fortran ------------------------
  mimax= 129 mjmax= 65 mkmax= 65
  imax= 128 jmax= 64 kmax= 64
 cpu :   1.83300805sec.
  Loop executed for  13 times
  Gosa :  0.0030001516
  MFLOPS measured :  116.791656
  Score based on MMX Pentium 200MHz :  3.61920214

-----------------------------------------------------
```

は最もプアなメモリスループットを持っているということが大きな原因でしょう。DS20やES40で測定すればまた違った結果が得られると思います。

Aクラスターと通信ライブラリ

1台ではとても出せないような高い性能を出すために、複数のマシンをクラスタ接続して利用することが一般的になってきました。その例として、以前(1999年1月号)Avalonの簡単な紹介をしました。スーパーコンピュータサイトTop500にランキングされているLinux/Alphaのマシンは2台あるわけですが、今回はAvalonと並ぶもう1つの雄である「CPlant」の紹介と、クラスタを作成するための基本ソフトである通信ライブラリの話をしたいと思います。

●CPlant

このシステムはアメリカのサンディア国立研究所で開発されているマシンです。1つのプログラムを安価なシステムで高速に動作させることを狙ったAvalonと違って、目的は計算センターのように多人数(数百人)で共有して利用する大規模システムを構築することにあります。そのため構成はかなり贅沢になっていて、1998年版ではマシンの計算ノードとして500MHzの21164Aを搭載するPWS500aを400ノード用意し、ノード間はMyrinetという1Gbpsクラスの高速ネットワークで接続するという構成でした。スーパーコンピュータサイトTop500の6月のリストでは400台のうち150台を使って129番目だったのですが、その後350台を使ったベンチマークで125.2GFLOPSをマークし、53番目に相当する順位に位置しています。1999年の計画では、ノードをXP1000にバージョンアップした上で800ノードに増やすことになっています。

CPlantでは、プログラムのコンパイルのためにTru64 UNIXのサーバを用意して、Compaq製のコンパイラを利用するようにしています。Linux/AlphaとTru64 UNIXはスタティックリンクされたバイナリには互換性があるので、このような対応も可能になります。

CPlantのノードはいくつかのパーティションに分かれています。パーティションの区切りは関連するノードをリブートするか再構成することによって変更できます。主なパーティションは次のとおりです。

- **サービスパーティション**:負荷分散後ユーザーのログオン先はこのパーティションのノードから選ばれます。
- **計算パーティション**:計算ジョブはすべてこのパーティションに登録されています。1998年版の計算ノードの構成は次のようになっています。
  - 500MHz 21164A
  - 192MB SDRAM
  - 2MB L3cache
  - 10/100BaseTx
  - Myricom Myrinet SAN PCI card
- **IOパーティション**:外部記憶を持ち計算ノードの入出力を受け持ちます。
- **コンパイルパーティション**:Tru64 UNIXで動作し、バイナリを作成します。

サービスパーティションのノードはシステムにログオンするユーザーごとに割り当てられます。そこで、システムのユーザー数によってこのパーティションの最適サイズは変わります。小さいジョブを行う人が多ければ、その分サービスパーティションのノード数は多く必要になります。逆に大きなジョブを数人が行うだけならサービスパーティションのノード数は少なくてすみます。

CPlantについての詳しい情報は、以下のURLから入手することができます。

```
http://www.cs.sandia.gov/cplant/
```

●通信ライブラリ

PVMやMPIなどのネットワークライブラリは、ネットワーク上のデータ形式やソケット番号、IPアドレスなどという数値計算の本質とは外れた瑣末な事項を、メモリに常駐するライブラリのルーチンが受け持ってくれます。 そのためユーザーは本来の計算部分に注力してプログラムを作成できます。もちろん、HPFのようなデータパラレルを言語上でサポートする物ではなく、単なるライブラリなので、容易といってもユーザーがデータの配置や通信タイミング等をすべて決めなくてはなりません。しかし日常使っているFORTRAN77やCなどの言語を使って並列プログラムが簡単に組めるようになるのでその恩恵は計り知れないものがあります。

PVMとMPIの関係ですが、オークリッジ国立研究所で作られたPVMに対して、MPIは並列計算機のベンダや研究者が集まって仕様を作成したという出所の違いが大きくその性格を決めているといえましょう。PVMは利用者が個々の計算機に個別にプログラムを組まなくても済むように作られましたが、通信の実装に関するベンダー個別の事情を無視しています。これに対してMPIはベンダが仕様制定に深くかかわっているため効率や汎用性には十分な注意が払われてきました。しかし、ベンダ間で調整がつかないものに関しては汎用性がないとして有用な機能であっても仕様から削除されてしまいました。そこで、PVMではできてもMPIではできないといった機能上の差が生じることになって、せっかくMPIが制定されてもPVMのユーザーの移行は簡単には進んでいません。一番の違いは並列プロセスや並列マシンの管理に関するものです。管理には汎用的な解決方法はあまりないのでMPIではこれらの管理を仕様から落しました。ですが、プロセスの生成や消滅はそれなりに需要があるので、MPI-2において導入されることになっています。しかしながら、一般の数値計算ではどちらもほとんどコーディング上の違いはないといえます。そこで、これから並列計算を始められる方は、MPIを使ってみることをお勧めします。今後はMPIがベンダのスタンダードとして定着して、MPI向けに性能や仕様を調整していく可能性が高いと思っているからです。

私自身はすでに自分で並列プログラムを作るときには100% MPIを使っています。MPIの有利な点をまとめたページが次のURLにあります。

```
http://www.mpi.nd.edu/lam/mpi/mpi_top10.php3
```

このページは英語なので、書いてある内容を以下に簡単に説明します。

- MPIには複数の無償の実装がある
- MPIは第3者によるプロファイルの方式を定義している
- MPIには完全な非同期通信が提供される
- MPIのグループの概念は堅牢で効率的
- MPIはメッセージバッファを効率的に利用する
- MPIの同期方式はユーザーを他のソフトから保護する
- MPIはMPPやクラスタで効率良く利用可能
- MPIは移植性が良い
- MPIは形式的に定義されている
- MPIは標準である

Aまとめ

今回は入手したXP1000のインストールの話と並列システムについてお話ししました。いろいろなベンチマークによる性能比較の企画も用意しつつありますので期待してください。私のXP1000は動作周波数が500MHzのマシンですが、クロックの高いマシンが続々と登場しております。ですが、コンピュータの性能は面白いもので、クロックのスピードと性能は必ずしも比例しないし、各種のパラメータさえきちんと把握できれば、性能予測も同じアーキテクチャのマシン同士なら比較的容易にできます。機会をみて性能の検証の話もきちんと取り上げていきたいと思います。