



# Linux / Alpha 活用講座

清水尚彦 nshimizu@keyaki.cc.u-tokai.ac.jp

## Java 2 SDK v1.3.1 版の評価

しばらくの間、CompaqのホームページからダウンロードしたNetscapeを使っていたのですが、画像が必要なことは日常の生活であまりないので、それならw3mでも十分ではないかと思ひ、最近はおっぱらw3mユーザーとなっています。必要な情報のほとんどは文字情報なので、外部ビューワを起動して画像を表示することもほとんどなく、結構快適に過ごしています。「こんなことならさっさとw3mを使っておけば良かった」といまさらながら思っています。今のところ、w3mで唯一困るのはYahoo!のクッキーが受け取れない」ということですが、致命傷ではないのでそのまま使っています。

ただ、w3mは最低限のキーバインドを覚えておかないと、使い始めには苦勞しますね。テキストベースでよいから、メニューを簡単に表示するモードがあっても良いかもしれません。思えばその昔、「CP/M」というOSで「ワードスター」という英文ワープロを使ったとき、ユーザーがメニューのレベルを3段階位にカスタマイズでき、キーバインドを覚えたらメニューの表示を制限する設定があったのに驚いたものですが(古い!)、同じように、初心者でも中級者でも気持ちよく使えるようなユーザーインターフェイスがあるといいですね。

以前は、せいぜいPDFがTeXかPSが使えるれば、ほとんどの仕事は済んだのですが、このところ大学でも「Webからシラバスや業績記録を入力しろ」という要求がくるので、ブラウザも使わざるを得ないような環境になってきています。しかし、Compaqの提供するLinux用のNetscape環境では、国際化ライブラリが不足していて日本語が十分動かずに困ることがありました。しかしw3mではそういうこともなさそうなので、ちょっとだけ安心して使っています。

誤解のないように付け加えておきますが、「Compaqの提供するNetscapeがダメ」というわけではありません。添付されているTru64のシェアードライブラリにi18n(Internationalizationの略)の

ものが抜けているのが問題なのです。ですから、Tru64もしくはそれに添付されるライブラリを使えば問題なく日本語も使えると思います(私は使っていないので確認はしていません)。

## A LaTeX環境のトラブル

4月号で、Compaq EB164上のDebian GNU/Linux potatoのアップグレード時、ptex-binのバージョン管理に問題があったことについて書きましたが、その後dvipsk-jaにも問題があることが分かり、修正しています。こちらは、バージョン管理でなく元のパッケージのconfigureのコマンドのアーキテクチャの扱いが悪さをしているため、Alpha以外のプロセッサでは大丈夫だと思います。

今回の問題はこのパッケージの「config.sub」のアーキテクチャ判定の項目に「alphaev6」が入っていないという初歩的なトラブルで、configure.subの該当箇所にこのアーキテクチャを追加すれば普通にビルドできます(alphaev56までは入っていますから、エディタで検索すればすぐに分かります)。どうしてこのようにチップモデルまで含めて評価をしているのか、configureの動作を良く理解していないので釈然としませんが、プロセッサのモデルが見つからなかったら、汎用のオプションを選択するという仕組みがあってもよさそうですね。

## A 21264/EV68Aの ハードウェアマニュアル

さて、本誌2000年4月号で「0.18umのAlphaであるEV68は出

ないんじゃないか」と書いたのですが、どうやら間違っていたようです。Compaqのホームページから、21264/EV68Aのハードウェアリファレンスマニュアルが入手できるようになっていました(記事末のRESOURCE [1]を参照)。これによると、940MHzのときの消費電力は75Wと、相変わらず相当電力食いのプロセッサですね。

ハードウェアリファレンスマニュアルは、高速な専用ライブラリを作成する際には「座右の銘」とも言うべき重要な役割をしてくれます。これを詳しく読むと、どのようなときに性能上のペナルティが発生するかが明確に分かります。もちろん、何でそんなことが起きるかが分かった方がいいのですが、読みこなすためには、プロセッサ構造に対するちょっとした想像力が必要になることもあります。

私が後藤氏とBLASや初等関数のライブラリを作成したときにも、EV5のハードウェアリファレンスマニュアルを参考にしながら、パイプラインストール条件や、ストアバッファ、ミスアドレスファイルの動作を元にして、プログラムの枠組を調整して性能を上げていきました。

21264/EV68Aのハードウェアマニュアルは、データブリッジ関係の説明が充実しているので、インラインアセンブラを用いたプログラムの高速化に役立つようになっています。

実を言うと、初代のEV6のハードウェアリファレンスマニュアルには、性能関係の情報がすっかり落ちていたのです。私はこれではっかりしてしまったものです。そのときには、仕方がないのでコンパイラ製作者へのガイドブックを使って情報を補っていました。

今回のハードウェアリファレンスマニュアルには、再び性能情報も統合されて分かりやすくなったと思います。ただ、EV68Aというからには、EV67とプロセス以外でも何か違いがあるのではないかと思うのですが、このマニュアルからは何も読み取ることはできませんでした。



## クレイの新クラスタ製品

さて、「T3E」などで並列コンピュータ分野で実力を見せているクレイですが、APIの「CS20」という1Uのデュアルプロセッササーバをベースに、Alpha/Linuxの並列スーパーコンピュータを作成するらしいという情報が入ってきました。

「The Cray SuperCluster Series」と名付けられた新しいクラスタ製品は、既存の製品を組み合わせたクラスタによって名機「T3E」を再構成するものです。その技術は前述のCS20のほかに、Myrinetのネットワーク、Linux、UNICOS環境と、実

績のあるものが寄せ集められています。これで専用並列コンピュータの系列はまた1つ衰退していくようですね。もっともIBMの「SP/1」の時代から、利幅のあまり厚くないHPC分野の製品にはこの方式が適していることが分かっていたので、ビジネスとしては正しい方向に進んだといえるでしょう。

この製品は、アマチュアが単にマシンをつないだだけのクラスタと明らかに差別化している特徴として、

- ・動的再構成
- ・グローバルチェックポイント/リスタート
- ・グローバルリソースマネージメント
- ・シングルシステムイメージ
- ・効率的ジョブスケジューリング

などを挙げています。大規模な長時間計算を行うときには、2番目の特徴が大変ありがたいものですね。計画停止の前にチェックポイントを保存しておけば、後で同じところから実行を再開できるのですから。クレイはSuperClusterの開発に当たって、「Linuxのクラスタ環境であるScyld OSを研究した」としていますから、これらの特徴のいくつかはScyldの技術を使っているのでしょう。



## Java 2 SDK v1.3.1 版

Compaqは、Linux/Alpha用のJava 2 SDK v1.3.1の 版を発表しました[2]。執筆時点では、Tru64版よりLinux版の方がバージョンが新しくなっています。MotifがLesstifが必要になりますが、Lesstifでは0.91.4-1でテストされているとなっています。

このバージョンでも若干問題があるようですので、安定したシステムをお望みの方は、Motifをお使いになることをお勧めします。Motifも、オープンソースコミュニティに対してはロイヤルティフリーとなっているので、Linuxでも無償で使えます。Open MotifのRPMファイルはCompaqのサイトから入手できます([3])。

Javaを使うには、そのほかにもglibc-2.1.3とCompaqの配布するライブラリlibots-2.2.7-2とcpml 5.1.0-2が必要になります。Compaq配布のライブラリはJavaのキットに入っていますが、FORTRANなどをインストールされている方は、すでに同じものが入っているはずです。

RPMをインストールすると、「/usr/java/jdk1.3.1/」というディレクトリの下にファイルが展開されます。ここには通常はパスが通っていないので、必要なパスを通してご利用く

## 実行例1 Java 2 SDK v1.3.1 版で行った100次元のLinpackベンチマーク

```
$ /usr/java/jdk1.3.1/bin/javac -O Linpack.java
$ /usr/java/jdk1.3.1/bin/java Linpack
Mflops/s: 49.048 Time: 0.01 secs Norm Res: 1.43 Precision: 2.220446049250313E-16
```

## 実行例2 Java 2 SDK v1.3.1 版で行った500次元のLinpackベンチマーク

```
Mflops/s: 37.226 Time: 2.25 secs Norm Res: 5.17 Precision: 2.220446049250313E-16
```

## 実行例3 CACAOで行った100次元のLinpackベンチマーク

```
Mflops/s: 62.424 Time: 0.01 secs Norm Res: 1.43 Precision: 2.22045e-16
```

ださい。

「/usr/java/jdk1.3.1/bin/java -showversion」と実行すると、次のようなバージョン情報が表示されます。

```
java version "1.3.1"
Java(TM) 2 Runtime Environment, Standard Edition
Classic VM (build 1.3.1-beta, native threads, jit)
```

「JIT」という情報がありますね。JITが動作しているなら高速になっていると期待が持てます。ということで、Linpack.javaのベンチマークを実行してみましょう。

まずは、従来より実績のある100次元の行列によるベンチマークからです。動作環境はAthlonチップセットを使ったマザーボードUP1100に600MHzのAlpha 21264を搭載した低価格マシンです。実行例1のように、結果は49Mflops/s程度となりました。実行時間を見ると分かるように、極めて短い時間で終わっているので、性能測定という面からは問題があります。そこで、500次元に拡張して測定すると実行例2のような結果になりました。同等の性能のベンチマーク結果をNetlibのページ([4])から抜き出してみると、表1のような報告がなされています。また、執筆時点の上位には表2のようなマシンが並んで

います。Pentium 4の検討ぶりが際立っていますね。これから見ると、AlphaのJDKのJITはまだ改良の余地がたっぷりあると言えるでしょう。

それでは、Alpha用の高速JITとして2000年4月号で紹介した「CACAO」でも試してみましょう([5])。その結果ですが、100次元の行列の方は実行例3のようになりました。これは、2000年4月号で報告したXP1000の500MHz

のマシンの値「57Mflops/s」に比べて、ほぼクロック周波数の性能の向上と同程度の向上となっています。これを500次元に拡張すると、若干性能が落ちます。CACAO自体はバージョンアップされておらず、古いJITしか動かない状況ですので、結果は実行例4のようになりました。Netlibのベンチマーク報告から対応するマシンを検索してみると表3のようになります。ほとんどの仕事をJavaでこなすという方は、もう少しJITの性能が上がるか、何かを待つのがよさそうですね。

もう1つの選択肢はgcjによってNativeなオブジェクトにするという方法です。マルチプラットフォームのロードモジュールが必要なケースは少ないかもしれないので、Javaのネイティブコンパイラというのも1つの進み方だと思ってます。Debianならパッケージを入れるだけで済むので簡単なのですが、Vine Linuxにはgcjのパッケージが用意されていないので、自分でビルドしなくてははいけません。

自分でビルドする場合にはgccのスナップショットを用いることとなります。2000年12月11日以降のスナップショットには、ライブラリを含めたリリースが統合されているので、扱いやすくなっていると思います。もちろん、スナップショットを用いることの不安はあるのですが、当面は仕方ないですね。

表1 Java Linpack List of Timingsより抜粋(実行例2と同等の結果)

Mflops/s	実行環境		
37.7	Intel PIII Windows2000 Dual 733 MHz PIII	8/31/00	Dan Zwick
37.6	Intel PIII Windows2000 PIII 800EB, 133MHz FSB, 384MB Micron PC133 SDRAM	7/18/00	Damien Heiser
37.5	Intel PIII Other OS/2 Warp 4.5 Java 1.3, Netscape 4.61	8/31/00	Duane Chamblee
37.2	Other Windows95/98 toghjgj	1/21/01	toghjgj
37	Other Windows95/98 duron 700	1/20/01	Kim Jin-Suk
37	Intel PIII Windows2000 MSIE 5.0 PIII 1.0 Ghz	2/1/01	Zeger Knops

表2 Java Linpack List of Timingsより抜粋(上位機種)

Mflops/s	テスト環境		
152	Other Other Dell 1.5 GHz, Windows ME	1/27/01	John Hebley
137	Other Windows95/98 pentium 4 1,500Mhz	1/22/01	Iars Klingman
118	Other Windows95/98 1.5ghz	1/23/01	bill
72.8	Other Windows95/98 AMD Athlon Thunderbird 1.36 GHz	2/11/01	Tim Devlin
66.9	Other Windows2000 Athlon 1200 266FSB 256MB PC2100	1/5/01	Sean K. McKannay

ちなみに500MHzのAlpha 21264を搭載したXP1000には、Debianをインストールしているのでgcjをパッケージから入れて試すことができました。500次元のLinpack.javaのベンチマークは実行例5のようになりました。が、Timeの値がどうも不安定で、何となく信頼性のない結果に感じられます。gcjが正式にリリースされたら、きちんと計り直してみたいところです。スナップショットのgcjをビルドしてみたい人は「Howto build and run libgcj/gcj snapshots or cvs」[[6]]に手順が示されています。新しいスナップショットなら、gcc-20010205.tar.gzなどのファイルを用意するだけでgcjのビルドができるはずです。

この原稿のために、20010205バージョンのスナップショットをダウンロードして試してみたのですが、コンパイルはできるものの、実行時にライブラリのモジュールが見つからないなど、スナップショットならではの環境整備が必要なようで、残念ながら記事には間に合いませんでした。

## A UP1100を SPECベンチマークで評価

さて、4月号で評価をあきらめたUP1100搭載マシンですが、やはり64Mbytesでは何もできないので、256MbytesのDIMMを購入しました。メモリの価格が下がっていることにもびっくりしますが、要求メモリ量が大きくなっていることにも驚きますね。昔メインフレームコンピュータの開発を行っていたころに「32Mbytesのメモリを標準にしたい」と言うと、「32Mbytesはいらないという顧客が値下げ要求を出すからあまり基本単位を大きくするな」と言われたのが、たった10

### 実行例4 CACAOで行った500次元のLinpackベンチマーク

```
$ ./cacao -old Linpack
Mflops/s: 45.487 Time: 1.84 secs Norm Res: 5.17 Precision: 2.22045e-16
```

### 実行例5 gcjで行った500次元のLinpackベンチマーク

```
$ gcj -02 -funroll-loops --main=Linpack -o lin Linpack.java
Mflops/s: 80.0 Time: 1.3200038 secs Norm Res: 7.0 Precision: 2.4E-16
```

表3 Java Linpack List of Timings

Mflops/s	テスト環境		
45.4	Intel PIII Windows2000 Dual PIII 800Mhz	9/8/00	Dzung Do
45.2	Other Windows2000 AMD Duron 950 mhz (overclocked from 600) in IE 5.5	11/4/00	Jared Richardson
45	Intel Pentium Windows2000 JIT	11/21/00	Robert Pigeon
45	Other Windows95/98 AMD Athlon 650(Classic),128RAM	9/28/00	Luka Bajzer
45	Other Windows95/98 amd athlon 800Mhz	8/25/00	henkjan

年前のことなのですが、次昔のことのようです。これは、かつてメインフレームのメモリがめっちゃくちゃ高価だったという理由があるので、価格決定権を持たない開発の一スタッフでは、いかんともしがたいもどかしさがあったものです。

PCの世界においても、CP/Mの時代は、たった64Kbytesのメモリでワープロやスプレッドシートやデータベースまで動作させていたのに……。64Mbytesでは何もできないという時代と比較すると、恐ろしいものがありますね。

昔話ということで出てきたCP/Mですが、これを試してみたい方は「The Online Software Museum」[[7]]を参照してみてください。MBASICやdBASE IIなど、一世を風靡したプログラムを試すことができます。MBASICってご存知ですか？ マイクロソフトが初めて売ったプログラムの末裔です。CP/Mのソフトはi8080用ですから、初期のMBASICの原型を留めていると思われます。CP/MそのものもCaldera社がNovel社から買い取った中からソースともども公開されています。

また、「Welcome to the unofficial CP/M web site.」[[8]]は、「unofficial」と銘打ったページではありますが、Caldera社から正式に許可を受けて公開しているもののようです。ここで公開されているバイナリとソースファイルは、あくまでもコマースソフトで商用には使えませんが、懐古趣味の実験にはちょうど適しています。yazeなどのエミュレータと合わせて利用すれば、趣味のZ80の組み込みアプリケーションの開発にも使えると思います。

CP/Mはともかくとして、UP1100をSPECベンチマークで評価してみました。同じチップセットを用いているAthlonのマシンも合わせて評価したかったのですが、Linuxで動作する

Fortran 90のコンパイラが入手できず断念しました。実は、Pacific-Sierra ResearchのVAST/F90という非商用には無償のコンパイラがあったのですが、コンパイラドライバがちょっとおバカで、SPECのベンチマークのコンパイルに失敗して動作させることができませんでした。原因は分かっているのですが、バイナリ供給されるコンパ

イラドライバは手を入れる訳にもいかず、かといって、シェルスクリプトで新しいドライバを書くほどの元気もなかったのが今回は見合わせです。元気が出たらスクリプトを書いて再挑戦したいと思いますが、あんまり当てにしないでください。Pacific-Sierra Researchのコンパイラは Veridian Pacific-Sierra Research High Performance Computing ([ 9 ])を参照してください。これはF77へのプリプロセッサになっています。このほかにも富士通がコンパイラを出していたのですが、商用配布を中止しているようですね。

比較のために、以前評価したDP264のデータとその性能比を表4にまとめました。ただし、UP1100の方が少しコンパイラのバージョンが上がっているので、この性能比がそのまま本来のプロセッサの能力比を表しているものではないことに注意が必要です。コンパイラが良くなったためと思われる変化は178.galgelで従来の低い性能値が見違えるほど、良くなっています。ただし、全体としてはメモリデータ幅が半分になり、バスクロックが2/3になってプロセッサクロック周波数も若干下がったので、かなりの性能差が出ています。プロセッサクロック周波数自体は1割ほどしか変わらないのでバス構造がいかにSPECの性能にインパクトがあるかが分かります。

## COLUMN

### 情報処理学会システム評価研究会 (SIGEVA)の紹介

ベンチマークなどのシステム評価は、私はかなり重要なことだと思っています。本来は自分のアプリケーションさえ速ければそれで良いのですが、システム構築やシステムの構築の際に、あらかじめ自分のアプリケーションでどれだけの性能が得られるかを判断するのは難しいものがあります。さらに、自分のアプリケーションやシステムの性能を把握してシステムチューニングをすることも要求されるため、評価に関してはこの面でも話題の尽きることはありません。

この「システム評価」を対象とした研究会ができました。その名も「情報処理学会システム評価研究会(SIGEVA)」([ 10 ])です。私もメンバーとして参加していますが、この研究会にはBSD系で活躍されている上原哲太郎氏もメンバーになっています。メインフレーム系の人達やオープンシステム系の人達など、いろいろな人達で和気あいあいとシステム評価の研究ディスカッションを行う会ですので、興味のある方はぜひ御参加ください(一説には「研究会より懇親会が大事だ」という情報もありますが、未確認です:-)。

(清水尚彦)

表4 DP264のデータと性能比

Name	UP1100	DP264	Ratio
168.wupwise	306	429	0.713
171.swim	397	654	0.607
172.mgrid	213	362	0.588
173.applu	251	393	0.639
177.mesa	397	470	0.845
178.galgel	456	286	1.594
179.art	275	540	0.509
183.quake	164	216	0.759
187.facerec	323	528	0.612
188.amp	244	341	0.716
189.lucas	256	410	0.624
191.fma3d	271	409	0.663
200.sixtrack	220	273	0.806
301.apsi	209	384	0.544
SPECfp_base2000	274	384	0.714

DP264 : Compaq Fortran-1.0, CXML-4.0.0, Compaq C6.2.9  
UP1100 : Compaq Fortran-1.1, CXML-5.0.0, Compaq C6.2.9

## R E S O U R C E

- [ 1 ] Alpha Technical Documentation Library  
<http://www.support.compaq.com/alpha-tools/documentation/current/chip-docs.html>
- [ 2 ] Linux/Alpha用のJava 2 SDK v1.3.1 版  
<http://www.support.compaq.com/alpha-tools/software/java/java.html>
- [ 3 ] Open MotifのRPMファイルの入手先  
<http://www.support.compaq.com/alpha-tools/software/index.html>
- [ 4 ] Java Linpack List of Timings  
[http://www.netlib.org/benchmark/linpackjava/timings\\_list.html](http://www.netlib.org/benchmark/linpackjava/timings_list.html)
- [ 5 ] CACAO Home Page  
<http://www.complang.tuwien.ac.at/java/cacao/>
- [ 6 ] Howto build and run libgcj/gcj snapshots or cvs  
<http://gcc.gnu.org/java/build-snapshot.html>
- [ 7 ] The Online Software Museum  
<http://elena.sysun.com/museum/exhibit.html>
- [ 8 ] Welcome to the unofficial CP/M web site.  
<http://deltasoft.fife.wa.us/cpm/archive/unofficial/index.html>
- [ 9 ] Veridian Pacific-Sierra Research High Performance Computing  
<http://www.psrv.com/>
- [ 10 ] 情報処理学会システム評価研究会(SIGEVA)  
<http://sigeva.aist-nara.ac.jp/>