



# Linux / Alpha 活用講座

清水尚彦 nshimizu@keyaki.cc.u-tokai.ac.jp

## 多粒度TLBカーネルでのSPEC計測

先月に引き続き、またバイクの話から始めます。バイクのチェーンがそろそろ寿命なので交換を依頼しました。ついでに点検をお願いしておいたのですが、カムチェーンを少し張っただけで、後はコンディションが良いから何もしていないとのことでした。カムチェーンはタベット音がしていたので以前自分でやろうとしたのですが、ピストンのマーカがどこにあるのかすっかり忘れていてできなかったため、この際だからサービスマニュアルを手に入れようと思い、バイク屋の親父さんに聞いてみました。すると、現物はもう手に入らないけれど、インターネットで探せば何とかできるのでは、と言われました。

年を取ったバイク屋の親父さんからインターネットという言葉が何の不思議もなく出て来るのに、新鮮な驚きを感じています。で、肝心のマニュアルの方は「YAMAHA」、「SR250」、「サービスマニュアル」のキーワードで検索して、複製版マニュアルを扱っているショップを見つけて入手できました。実は、このバイクを新車で買ったときに、サービスマニュアルと一緒に買おうとしたのですが、そのときのバイク屋は「買わなくても、いつでも見せてあげるから」と言って売ろうとはしてくれませんでしたので、ずっと、あのとき無理にでも入手しておけば良かったと後悔していました。

また、最近のテレビにはないと思いますが、私が購入した20年程前には、テレビには回路図が付いて来ました。LSIの時代になると代替部品は入手しにくいので困ってしまいますが、トランジスタのテレビなら、単品で部品を交換すれば簡単に直ったものです。回路図にしてもサービスマニュアルにしても、それを読み下してメンテナンスをするには、それなりの努力がいるのですが、努力すれば報われるというところが大事なのだと思います。人が使うものがどんどんブラックボックスになっていくのは寂しいものです。

ついでながら、バイク屋で整備をしてくれたお兄さんは、点検のついでに試乗していたらしく、乗りやすいし、ポジションもゆったりして長距離も楽そうなおバイクだとしきりに褒めていました。ヤマハからまた売り出せばいいのになんて言っていました。たまに高速を使うけれど、普段は一般道を使う私のようなユーザーには、車重が軽く取り回しが楽で、さらに燃費も良い250ccという排気量はちょうど良いのです。レースをやるわけでもなく、目立ちたいわけでもないで、私は「普通の」バイクが欲しいだけなのですが、税制や車検などの関係で人気のあるはずの250ccというカテゴリのバイクにメーカーが力をあまり入れていないのが残念ですね。

さて、最近読んだ本で面白かったものに、ダイヤモンド社から出ている「いつまでバグを買わされるのか」という本があります。

これは、「ソフトウェアにはバグはつきもの」というユーザーのあきらめの態度が、余計にバグを放置することを許すことになるという論旨で、正当な権利を主張しようとするユーザーを激励している本です。

複雑さの度合いが高くなってくると、ソフトウェアだけでなく、ハードウェアにも同じことが言えるようになると思いますが、幸いにして多くのハードウェアの製作メーカーは、品質管理のプロセスをきちんと持っているように思われます。その点から見ると、フリーソフトウェアは、ボランティアベースの開発が多いので、品質管理のためのプロセスが機能するとは言いがたく、品質向上のための何らかの働きかけが必要になってくるのではないのでしょうか。ユーザーベースが多いからバグの報告も修正も素早くできるというのは良く聞かれることですが、そもそも、設計の段階やリリースの段階でバグを大量に内包しているという状況が問題なのです。

この本で指摘されているようなことが対策されて、商用ソ

フトの品質が工業製品として極めて当たり前のレベルに達したときには、フリーソフトが基幹部分を担うのは困難になるのではないのでしょうか？

今は、有料でも品質の悪いソフトと、無料でありながら有料と同等以上の品質を持ったソフトという位置付けがされているわけですが、(当初そのようにフリーソフトウェアを位置付ける論調が多かったように)有料で品質の良いソフトと、無料で品質保証のないソフトという形に本当に変わってくると、ボランティアベースのフリーソフトウェアは、隙間市場に対応したものに限定されるのではないかと危惧しています。

もっとも、多くの会社はこの本で書かれているように、ユーザーが品質を望んでいないと思っているので、その間に基幹ソフトとしての地位を固めることで、共通なプラットフォームとしてのオープンなソフトが力を付けていく余地は十分あるはず。今後、開発プロセスにおよぶ議論がLinuxの世界でなされていくと面白いと思います。

ビジネス上の基幹ソフトといえばデータベースですが、SAPがデータベースソフトのADABASをオープンソースにするとという話を聞きました。ADABASは、確か階層型のデータベースではなかったかと思いますが、ビジネスで実績を持っているこのようなソフトが時代遅れになって利用者が減ったからだと思います)オープンソースの世界に来ることで、生きた教材としてさらなる進んだソフトウェアを構築するヒントとなればいいなと思っています。

ソフトウェアでもハードウェアでも、範となるものがあることで、新規の技術者の参入障壁を大変低くすると思っています。ただ、Mozillaのプロジェクトのように、いきなり大きなソースを見せられても新しく取り組む人は山のようなソースコードの何処から手を付けて良いのか分からないでしょう。良い解説書はどうしても必要になります。

その点、MINIXを教科書として出版したTanenbaum教授やPL/0のコンパイラのソースを載せて出版したWirth教授の功績は素晴らしいものだと思います。

以前読んだBorland社のインタビューの中で、コンパイラの開発のヒントはPL/0の中にすべてあったと書いてありました。MINIXもPL/0も、それほど大きなソフトウェアではありませんが、必要なものは大体備わっている完成したソフトウェアに、きちんと解説がついた教科書となっていることで、これらをベースに勉強する人達の助けになっています。学習効果をあげるためにさらに大切なことは、手に取って確かめられることだと思います。MINIXは、IBM PCという当時(今でも)一番普及していてどこでも手に入るハードウェアで動作するように設計されていて、誰でも手に取って使っ

みることができました。

こう考えると、IT講習券なんてものにお金を使うより、日本語で読める優れたLinuxベースのOS解説書を作ったり、GCCをベースとした優れたコンパイラ解説書を作ったり、Mozillaをベースにしたブラウザ解説書を出すとか、日本のIT技術を立ち上げるためにすべきことは山ほどあるように思えます。

こういって、すでにあるものを勉強しても仕方がないという人がいると思いますが、技術は現状の技術の歴史の上に重ねられるものであると私は信じています。歴史に学ぶというのは、何も人文関係の学問や政治だけのことではないのです。ただ、先ほど挙げたソフトウェアは、まだ技術的に熟成しているとは言いがたく、解説書が教科書のような説得力を持たない可能性があるのが懸念されることです。

その点、ADABASのように産業分野で実績を積み、問題点が枯れているソフトウェアをベースに優れたデータベース解説書ができると、その分野でのマイルストーンとして重要なものになるような気がします。

IT普及とは、アメリカのIT関連産業の市場となることだとしか思っていないような人たちが主導すると、つまらないことに我々の税金を投入されることになりそうです。

## A 私の成果

文句ばかりで何も自分でやっていないと思われるのも心外なので、こういったことに私自身も取り組んでいる(いた)ことを書いておきます。

1つは、プロセッサの論理設計です。Webページで教育用のパイラインプロセッサを公開しています(記事末RESOURCE[1]参照)。8bitの小さなプロセッサですが、キャッシュを持ち、ヒットアンダーミスの制御や分岐予測も備えた、学部学生の教育目的には十分と思われる内容を盛り込んだ設計になっています。解説書の出来は十分ではないのですが、このプロセッサを用意したことで、私の研究室の学生は、プロセッサを作ることに抵抗を感じなくなっていると思います。

このプロセッサは、海外からも広くダウンロードされますし、これを見たらしい海外のメーカーから、新プロセッサのプロジェクトへの参加を打診されたこともあります。ただ、ハードウェア記述言語の処理系がフリーのものを使っていないため、直接試すことができる人たちが少ないのが、私自身は不満に感じていて、なんとかハードウェア記述言語の

処理系も自前で用意したいと思っています。

そうそう、コンパイラの開発も重要ですね。このプロセッサ用にyacc / lexで小さなコンパイラを作ろうとされていて、そのための準備はしているのですが、ついつい先送りになっています。コンパイラも簡単にできることを学生には示してあげたいので、何とか時間を見付けたいものですね。

もう一つは、今は閉鎖されているLinux/Alphaの日本語メーリングリストのメンバーで行った、数値計算ライブラリの開発です。これは、メーカーの専用ライブラリ並、もしくはそれを超える性能のライブラリをオープンな開発プロジェクトが作成可能であることを示したものと、有意義だったと思います。libmの改良や、dgemmルーチンの開発を行いました、パイプラインやメモリレイテンシを意識したプログラム開発と、性能チューニングのためのハードウェアモニタの利用技術など、できたプログラムだけでなく、開発プロセスとしても注目してよいプロジェクトであったと思っています。

さらに、ここ数カ月取り組んでいるページ粒度の変換機構をLinuxに取り込むカーネル改造があります。大規模数値計算を行うための障害を観察すれば、必然的に要求される仕組みなのですが、SGI以外のワークステーションメーカーはあまり積極的に問題視してこなかったように思います。ハードウェアとOSとアプリケーションの3者の動きを知らないと取り組めないことが、その障害なのでしょうか？あるいは、別のセクションに口を出しにくい「部門最適」の罠に陥っているのかもしれない。

多粒度TLBをサポートしたカーネルで、代表的なベンチマークであるSPECベンチマークの結果を計測することができたので、報告しておきます(表1)。実は、整数ベンチマークは一部のアプリケーションのコンパイルに失敗して計測できなかったため、浮動小数点だけを報告することにします。SPEC CPU2000の浮動小数点ベンチマークは、表2のようなアプリケーションによって構成されています。これらの結果をSUNの「Ultra5 10」という、300MHzのUltra SPARCのワークステーションにおける性能(これを100とします)と比較したものがスコアとして扱われます。

ベンチマーク計測は、ビジュアルテクノロジー社の「DP264 (21264A-667MHz)」のマシンで行いました。Linux 2.2.16と、これに多粒度TLBパッチを施したもので計測しました。コンパイラは、「Compaq C-6.2.9」と「Compaq Fortran-1.0」です。表1、グラフ1)には、SPECのホームページ[2]からの引用として、Tru64の「DS20E(21264A-667MHz)」と、Athlon 1.1GHzのデータも合わせて載せておきます。

DS20Eはプロセッサは同じでもコンパイラが異なります。

特にFortranは、「KAP FORTRAN」というソースレベルの変換を行うものを用いています。おそらくキャッシュ容量も違うのではないのかと思いますが、自分の使うマシンの構成を正確には把握していないので、明らかではありません。Athlon

表1 SPECベンチマークの結果

	no GH	GH	Tru64	Athlon
168.wupwise	429	434	482	340
171.swim	654	727	1101	503
172.mgrid	362	361	545	256
173.applu	393	420	405	294
177.mesa	470	487	511	419
178.galgel	286	994	445	457
179.art	540	564	1245	211
183.equake	216	214	244	260
187.facerec	528	573	681	390
188.amp	341	344	414	291
189.lucas	410	513	662	230
191.fma3d	409	381	487	354
200.sixtrack	273	276	274	237
301.apsi	289	497	450	267
SPECfp_base2000	384	452	514	311

no GH : DP264(Alpha 21264-667MHz) Linux 2.2.16

GH : DP264(Alpha 21264-667MHz) 多粒度TLBサポートLinux

Tru64 : DS20E(Alpha 21264-667MHz) Tru64 UNIX

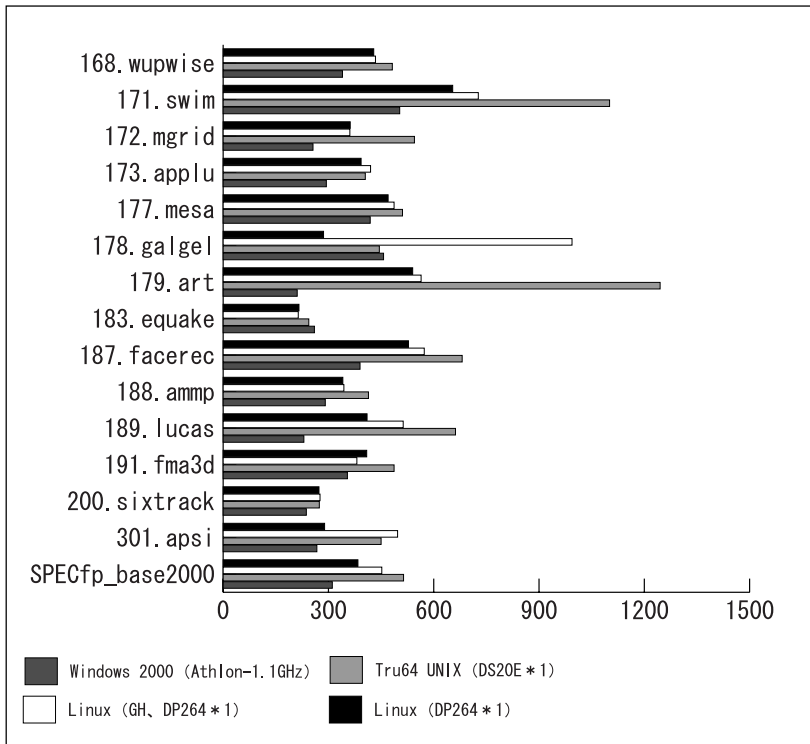
Athlon : Athlon-1.1GHz、Windows 2000

表2

ベンチマーク	言語	説明
168.wupwise	F77	Physics: Quantum chromodynamics
171.swim	F77	Shallow water modeling
172.mgrid	F77	Multigrid solver: 3D potential field
173.applu	F77	Partial differential equations
177.mesa	C	3DGraphic Library
178.galgel	F90	Computational fluid dynamics
179.art	C	Image recognition/neural networks
183.equake	C	Seismic wave propagation simulation
187.facerec	F90	Image processing: Face recognition
188.amp	C	Computational chemistry
189.lucas	F90	Number theory/primality testing
191.fma3d	F90	Finite-element crash simulation
200.sixtrack	F77	Nuclear physics accelerator design
301.apsi	F77	Meteorology: Pollutant distribution

表3

ベンチマーク	性能向上率
171.swim	11.2%
178.galgel	247.5%
189.lucas	25.1%
301.apsi	72.0%



グラフ1 SPECベンチマークの結果

はIntelのCコンパイラとCompaqの「Visual Fortran」を用いています。表にあげる数値は、すべて「CFP\_base」と呼ぶコンパイルオプションを共通化したデータになっています。

このデータの中で、粒度サポートで10%以上の性能差があったベンチマークは表3のもです。

これを見ると、「178.galgel」と呼ばれるベンチマークの性能向上率が著しいのが目立ちますね。

このベンチマークは、「CFD」と呼ばれる流体力学のアプリケーションですが、大量のメモリをアクセスすることから、多粒度TLBの効果がはっきり出ています(もっとも大量のメモリをアクセスするアプリケーションは他にも多くあり、このアプリケーションに効果が高かったのは、大量のメモリのアクセスのアクセスパターンが一般のRISCプロセッサにとって扱いの難しいものだったということです)。

この値は、KAP FORTRANを用いたDS20Eより2倍程性能が良くなっています。KAP FORTRANはソースレベルで最適化してメモリ参照パターンも最適化しているはずなので、KAPでは最適化できない本質的な参照パターンがあることを示していると思います。このように、多粒度TLBのサポートで性能を上げるものが多いのですが、DS20Eの結果と比較すると、多くのアプリケーションではKAPの威力が出ているので、ぜひLinux用にもKAP FORTRANをリリースして欲しいも

のですね。ただし、179.artだけは小さなアプリケーションで、4MBytesのキャッシュがあると、ほとんどすべてキャッシュに入ってしまうので、大きな差はキャッシュの差なのかもしれません。

この結果、集計値である「SPECfp\_base2000」は17.7%の向上となりました。基本的には性能が下がる要素はないはずなのですが、191.fma3dのように、若干下がったものもあるので原因を追求する必要があります。しかし、全体として期待した通りの性能になっていると思います(このパッチの開発当初は15%の性能向上を目指してしまっていたので、目標性能には達しています)。

今までは、自分が用意したアプリケーションで性能が上がると言っていたので、都合の良いものだけを集めて話をしていると懐疑的な意見もあったと思います。しかし、SPECのベンチマークを試すことができ、他のプロセッサなどとの比較も簡単に

できるようになりました。IBMやHPの高性能機と比較してもLinux/Alphaのマシンは十分競争力を持っていることが分かります。

Samsungの新しいプロセッサが市販されるようになれば、さらに大きく性能が向上する可能性があるのも楽しみです(これは「EV68」と置き換え可能と噂されていますから、場合によっては、今使っているマシンのプロセッサカードを入れ換えるだけでいいのかもしれませんが)。後は、プロセッサとシステムの価格がずっと下がってくれることを期待します。

## R E S O U R C E

- [ 1 ] 筆者のホームページ  
<http://shimizu-lab.et.u-tokai.ac.jp/>
- [ 2 ] SPEC(Standard Performance Evaluation Corporation)  
<http://www.specbench.org/>