

数学ソフト蒐集プロジェクト, project AP Library について (Extended Abstract)

沼田 泰英*

数学ソフト蒐集プロジェクト, project AP Library^{*1}について紹介する.

概要

1 縁起

Project AP Library とは, 一言で言うと “数学に関するソフトウェアを広く浅く集めようというプロジェクト” である. まず, なぜこのようなプロジェクトが始まったのかについて簡単に説明する.

近年, 純粋数学の研究にも計算機が欠かせなくなりつつある. すなわち, 計算機による具体例の計算の下で予想された様々な結果に対し, きちんとした理論付けを行うという研究方法が広がりつつある. 数学の研究を補助する様々なソフトウェアが存在しており使われている. 講演や集中講義などで, “この計算にはこのソフトウェアを使った” とか, 逆に “このソフトウェアでは計算が終らない” などという事が述べられる事もある. この様なコメントがあった後に, 実際にそのソフトウェアを使用したいと思う事がある. しかしながら, この様な場合にそのソフトウェアが自分の使っている環境にすでに入っている事は少ないと思われる. この場合, 実際に使うためには, 自分の環境にインストールする事が必要であるが, 共用の計算機であれば簡単にはいかず, 例えば次のうちのいずれかをする必要があると思われる^{*2}:

1. 自分のスペース (home directory) にインストールをする.
2. 管理者にお願いしてインストールしてもらう.
3. あきらめる.

しかしながら, いずれも難点を持っている. まず, 自分のスペースにインストールをする場合であるが, 通常ソフトウェアはマシン全体にインストールをする事を前提に作られている. そのため, 様々な設定を修正してからコンパイルをする必要がある. この作業には熟れや知識が必要である. また, 必要なライブラリなどがイモヅル式にでてくる事もありこれに対して全て対処しなければならない. これは, 試しに使ってみたいと思った当初のモチベーションに対してあまりにもハードルが高い.

次に, 管理者にお願いをしてインストールしてもらうという選択肢についてであるが, 管理者はそれなりにインストールになれてはいるであろうが, やはり手間には変わり無い. また, マシンにインストールするとな

* 北海道大学大学院理学研究科; nu@math.sci.hokudai.ac.jp

^{*1} <http://www.math.sci.hokudai.ac.jp/~g-staff/project/aplibrary/>, g-staff-aplibrary@math.sci.hokudai.ac.jp

^{*2} もちろん, 個人用の計算機を持っているなら, KNOPPIX/Math を使うなど選択肢は広がる.

るとインストール後も管理が必要である。実際にそのソフトウェアを使っていない管理者にとってそのソフトウェアがきちんと動いているかを把握しメンテナンスをするのは大変難しい事であると思われ、また、他のソフトウェアに影響がないか、セキュリティは大丈夫かなど様々な懸念もある。これは、いちユーザが試しに使ってみようというのにはハイコストである。

そこで、我々は、院生用の端末室の 10 数台あるマシンの内の 1 つを“試しに様々なソフトウェアを使ってみるためのマシン”を位置付け、(きちんと動作をする事は保証せず) 様々なソフトウェアをインストールすることにした。目的は、あくまで試しに使ってみるということであり、いざ誰かが試しに使いたくなった時のために、日頃時間に余裕がある時に様々なソフトウェアをインストールしようという事である。

2 目標と方針

ソフトウェアを広く浅く集めると言う事が目標である。動機は“試しに使ってみる”というのであるので、様々なソフトウェアをインストールしておき、いざという時に使えるようにしておく事が第一である。その目標のためにであれば、そのソフトウェアがきちんと動くかどうか、最新であるか、などという事は犠牲にしても、沢山のソフトウェアをインストールする事が必要である。

集めるという意味は、我々の端末にインストールをするという事の他に、我々の端末にインストールしやすい形に build しなおす (パッケージングする)、またパッケージングしたものをライセンスの許す限り公開し第三者が利用できるようにするという事も含む。

この目的のために次の方針の下で事を進めている:

- インストールされたものがきちんと動いているかには興味を持たない。
- 複数のバージョンがある場合、一番インストールしやすいものを採用する。
- インストールが困難な場合は潔く諦める。
- ライセンスに注意し、なるべく OpenSource なものを集める。
- RPM 形式で build し、APT-line を公開する。

この方針について、以下で簡単に説明する。

2.1 インストールされたものがきちんと動いているかには興味を持たない。

もちろん、きちんと動くか否かと言う事は重要な事であるが、数学の研究を補助するソフトウェアは高度に専門的であり素人にはきちんと動いているかどうかの判断は難しい。インストール時にエラーが出ず、取り合えず起動すればそのソフトウェアについては完了とする。

2.2 複数のバージョンがある場合、一番インストールしやすいものを採用する。

あくまで試しに使ってみることが目的であるので、最新のバージョンには拘らない。また、安定して動くバージョンである必要もない。取り合えず起動するという形でインストールするのに一番簡単なバージョンを採用し、少しでも多くのソフトウェアを集める。

2.3 インストールが困難な場合は潔く諦める.

とりたてて理由が無ければそのソフトウェアは諦めて、別のソフトウェアのインストールを試みる。広く集める事が目的なので、その方が目的にあっている。まだまだ多くのソフトウェアがあるわけで一つのソフトウェアに時間を費す理由は無い。もしかしたら、バージョンが上がりインストールが可能になる事も考えられるので、保留とし他のソフトウェアに取り掛かる。

2.4 ライセンスに注意し、なるべく OpenSource なものを集める.

もちろん不法行為を行うわけにはいかないでライセンスに十分注意する。また、パッケージングしなおしたい手前 Open Source なものがよい。

2.5 RPM 形式で build し、APT-line を公開する.

なるべく make install による直接のインストールは行わず、RPM 形式で build をしインストールする。なぜ make install ではなく RPM なのかというと、make install よりも RPM の方がアンインストールがしやすいからである。様々なソフトウェアをインストールするので不具合が出るかも知れない、また、ためしに試してみるという事であるので使わなかったらアンインストールをするであろう。ためしに使うという目的のためであれば、インストールのしやすさもさることながら、アンインストールのしやすさが重要になってくる。

なぜ、Debian のパッケージ deb では無く RPM なのかというと、我々の端末室の OS が Vine Linux であった事、deb を知っている人間がいなかった事が主な理由であり深い理由は無い。

作った RPM は、APT-line として公開する。この様な形で公開する事で、VineLinux であれば簡単にソフトウェアをインストールが出来るようになり、第三者が利用しやすい。

3 現状

ここでは、現状、システムについていくつかのべる。

3.1 現状

我々は、およそ 300 の数学に関係があると思われるソフトウェアの存在を知っている。これらの内、RPM の作製が終了しているものは、Vine に始めから存在するものも含めておよそ 60 である。我々の APT-Line にあるファイルのサイズの合計は、およそ 1.1G である (主に i386.rpm と srpm がほとんどを占める)。

現在、1 から RPM を作るのではなく、すでに他のディストリビューション様にカスタマイズされた RPM を Vine で build 出来るように作り直すと言う作業を中心に行っている。他のディストリビューション用の RPM が存在しているものは、まだ終わっていないものの内およそ 70 である。これらは rpm SEEK.com^{*3}、RPM pbone.net^{*4}、Rpmfind.Net^{*5}、FileWatcher^{*6} などを利用して調べた。

*3 <http://www.rpmseek.com/>

*4 <http://rpm.pbone.net/>

*5 <http://rpmfind.net/>

*6 <http://www.filewatcher.com/>

3.2 BTS: 影舞

我々は集めるべきソフトウェアの情報を, Bug Tracking System 影舞^{*7} を用いて管理している. BTS というのは, ソフトウェアの開発時に用いられ, どのようなバグが報告されているのか, 報告されたバグはどの程度解決したか, などという情報を管理するために, 用いられているものである. 我々は影舞を利用し, どのようなソフトウェアがあるのか, どのソフトウェアのパッケージングが終ったのかなどと言う情報を管理している. 全て web 上から入力が可能で, 新しいソフトウェアを見つけた場合はいつでも投稿出来るようになっている. 本来であれば一般に公開し新しいソフトウェアを知っている人が投稿できるようにした方が良いのであるが, SPAM を懸念し今のところ公開していない.

3.3 wiki: RandomNote

RPM を作製するにあたって, 様々な knowhow を蓄積するために, RandomNote^{*8} (wiki の一種) を用意した. この wiki に書き込まれている内容には第三者にとっても有益なものも含まれており公開した方が良いと思われるのではあるが, こちらも SPAM を懸念し今のところ公開していない.

Wiki というのは, web から誰でも書き込む事が出来る掲示板のようなシステムである. 様々, wiki があるが, 我々の選んだ RandomNote には次のような特徴がある: 通常の wiki にあるようなカテゴリ分類などの機能はない; また各ページのタイトルなどと言う概念も, 通常の wiki と異なり存在しない; その替わり, 検索機能があり, 検索する事で過去に書いた記事から欲しい記事を探すことになる. 小さなメモ書きを, 形式などに拘らずとりあえず保存しおき, 必要となった時に探し出すといった, 形態にマッチするシステムであったので, 我々は RPM 作製の際のコツのようなものを書き溜めるために採用した.

3.4 APT-line

jack.math.sci.hokudai.ac.jp というマシンと Apache を使い web server として運用している. この web server を使い, 作製した RPM を APT-line として公開している. 今のところ http のみで ftp によるアクセスは用意していないが, ftp サーバを設定するのが面倒なのだという消極的な理由による.

APT-line として, 北大以外からもアクセスされているようではあるが数は少ない (2 箇所からは定期的にアクセスがあるようである). APT-line としてではなく, ウェブブラウザなどを用いてダウンロードしていくケースが多い (2006 年 6 月のダウンロード件数: srpm 45 件, i386 用 rpm 62 件). これらのほとんどが google などの検索ページから (rpm2html による解説データベースのページを経由して) 来ている.

我々の公開している APT-line を利用する際の, /etc/apt/sources.list の書き方は <http://www.math.sci.hokudai.ac.jp/~g-staff/project/aplibrary/aptline/> を参照.

3.5 rpm2html

APT-line で公開している RPM の情報は rpm2html というソフトウェアを用いて html にし <http://jack.math.sci.hokudai.ac.jp/aptline/rpm2html/> で公開している.

^{*7} <http://www.daifukuya.com/kagemai/>

^{*8} <http://ninjinix.x0.com/rn/index.rb?AboutPage.txt>

4 問題となった点

ここでは我々が苦勞している点、苦勞した点についていくつか述べる。

4.1 RPM の作製

我々には、RPM の作製に対するノウハウが全く無かった。どのように RPM を作製すれば良いのか、どこを変更すべきかなど全くわからず試行錯誤が続いた。

我々は基本ディストリビューションを Vine としたが、Vine は良く言えば安定指向であり悪く言えばライブラリが古いものばかりである。そのため最新のライブラリを必要とするものはライブラリも build する必要があった。このようなプロジェクトを行う際には、Fedora、Debian などのディストリビューションの方が良かったのかも知れない。

他のディストリビューションのものを修正する際、ライブラリの名前を変更する、Vine には無いマクロを変更するという 2 点が一番多い。これらは、ノウハウが貯り変更方法がわかった今は問題ないが、当初は苦勞した。rpm SEEK.com などが必要とされるパッケージを探し、そのパッケージに含まれるファイルを調べ、そのファイルが Vine ではどのパッケージに含まれるかを rpm -qf などを用い探し出すということを繰り返した。

4.2 APT-line

当初 APT-line を公開する方法についてのドキュメントがあまり無かった。結局 genbasedir などのコマンドを使用すれば可能であったのであるが、このコマンドを知るまでに時間がかかった。またこの genbasedir についてもドキュメントが少なく使用方法を調べるまでに時間がかかった*9。

4.3 人手

人手が足りていない。新しいソフトウェアを見付ける人、RPM を探す人、RPM を作る人、出来た RPM を試す人、そのどれもが足りておらず作業が捗らない。

5 今後

今後、取り合えず、我々は更にソフトウェアを蒐集することを目標としたい。i386 以外のアーキテクチャ、特に ppc 用の RPM も作って行きたい。また、RPM だけでなく deb も作製したい。いまは“広く浅く”であるが“広く深く”したい、即ち、きちんと、最新版や安定版のパッケージを作製し追従して行きたい。

取り合えず、人的資源が足りないので、興味のある方の参加をお待ちしております。(例えば、あのパッケージはインストールしようとしたけど出来なかった、こんなソフトウェアもあるよ、などという報告だけでも助かります。)

*9 2 年程経ち、web にあるドキュメントもかなり増えた

付録 A 我々の知っているソフトウェアの列挙

以下に我々が存在を知っているソフトウェアを列挙する。ただし、BTS のタイトルとして投稿されたもののコピーであるので正式な名称とは限らない。

3D A2X ACE ACE/gr ACL2 ADE ALDOR ARIBAS Akapa Algae Algebra Alien Analysis Apfloat Axiom BLACS BLAS Boost CGAL CLICAL CLN CM2 CMAT Calcoo ChaosTimes Church Cinderella CoCoa Combinatorial ComputeL Computer DELORES DERIVE DPGraph Dataplot Diffpack DigiGraph DigitizeIt Dr. ECPP ESS Eukleides Euler FIST FORM FRISCO FastGEO Fermat FeynmanGraph FlexPDE Fracplanet Free Fundamental Fung-Calc GALP GANG GAP GB GENESIS GENREG GIAC GLgraph GMP GRAPE GRG GRPN GTS Galois Gandalf Ganith Gauche GeoSteiner GetData GiNaC Gmsh GnuBC GraphThing Graphlet Gretl Gri HULL HartMath IntBasis JACAL KANT, KNOT KSEG Kalamaris Kali Kan/sm1 Kenzo KnotPlot Knotenpunkt Knotscape LAPACK LEDA LINK: LIVEMATH LOWX Lapackpp LdeApprox LinAl LinPacker Linear Lisp-Stat MAGNUS MAS MAXIMA METIS MP: MaTX MacAnova Macaulay2 Mace4* Macsyma Magma MashTV MathCAD MathPad Mathomatic Mathscribe Matrix Mock-Mma MuPAD Mx Mxp NTL NZMATH NetSolve/GridSolve Ngraph Octave Online PAFF PARI/GP PDL PGPLOT PSPP Paradox Phaser Ploticus ProofGeneral QMG Qemu Qhull R REDTEN REDUCE RTP Reviver Ricci Rlab SACLIB SCURVE SF SNAC Schur SciGraghica Semigroupe Shape SimLab Simplicial Snap SnapPea Software Superficie Surface Symaxx SymbMath Symmetrica TeLa The Topaz Vega ViSta WIMS X-Bc XLISP-STAT XLOOP XMDS: XMaxima XTrain XaoS Xlife YACAS Yorick ZEN admesh apcalc arpack++ autotool bear bernina bibcursed bugsx casa cdd, cgtkcalc checkinstall cimg circlepack coq daVinci dalc dc ddd deal desing disk dome drawmap dstooltk edenmath.app epc epix eqc felix fftw3 freefem freefem3d frontman fung-calc g3data gTybalt gcalculator gambit gbase gcalctool geg genius geomview gfrac glito glsc gnuplot goctave grace graphviz gsl-progs gtkgraph guppi gyapp hexcalc hp67 java jgraph k2k kalc kbruch kdeedu kgeo kig kile kmatplot kmplot koctave3 kpercentage kpl2 kpncalc krelais kstars labplot lbt lidia lmp lp-solve lrs matgen mathplot matlab matrixss matwrap ming myphpmoney nauty numexp-core online pasu perisic pgcalc phox pi plantri plot plotmtv plotutils polymake porta pstoedit pychart pymathml qalculate qliss3d qmatplot qtzmanim rCalc regina-normal rgnuplot rkward rplot rrdtool rs sc scilab simath singular spice statist surf tcp/tc teddy tgeg tk tkmatrix vinci xfig xgobi xgrid xlistat xtoys yaplot yplot z88 zimg zmanim
S A M

付録 B APT-RPM

B.1 APT-line の公開方法について

RPM を置くためのディレクトリ:

- /localdirectory/version/arch/RPMS.category
- /localdirectory/version/SRPMS.category

を作る。ここで version は例えば Vine Linux 3.2 であれば 3.2 などとする。arch は取り合えず、i386 を用意しておけば良い。noarch, i486, i586, ppc など最適。category はすきに付けければ良い。SRPMS.category には *.src.rpm を arc/RPMS.category には *.arch.rpm を置く。

この状況で

```
genbasedir /localdirectory/ver/arch category
```

もし複数のカテゴリがある時は

```
genbasedir /localdirectory/ver/arch category1 category2
```

などとすれば良い。RPM を増やしたり更新する度に、この genbasedir を叩く必要がある。

次に、シンボリックリンクをはるなどして /localdirectory/version らを公開する。

もし /localdirectory/version を <http://yourdomain/parthtoaptline/version> という名前で公開するのであれば、`/etc/apt/sources.list` に

```
rpm http://yourdomain/parthtoaptline version/${ARCH} category
rpm-src http://yourdomain/parthtoaptline version/${ARCH} category
```

などと書いたマシンからであれば `apt-get install` でインストールできるようになる。

B.1.1 例

我々の場合を具体例を挙げておく（ただし、初期のディレクトリ構成であり現在の物とは異なる）。我々は、ライセンスごとにパッケージを分類する事にした。カテゴリはライセンス名からとり、

- gpl
- lgpl
- bsd

などとした。このカテゴリ名に合わせて、

- /privaterpms/3.0/i386/RPMS.gpl
- /privaterpms/3.0/i386/RPMS.lgpl
- /privaterpms/3.0/i386/RPMS.bsd
- /privaterpms/3.0/noarch/RPMS.gpl
- /privaterpms/3.0/noarch/RPMS.lgpl
- /privaterpms/3.0/noarch/RPMS.bsd
- /privaterpms/3.0/SRPMS.gpl
- /privaterpms/3.0/SRPMS.lgpl
- /privaterpms/3.0/SRPMS.bsd

というディレクトリを用意した。例えば GAP というソフトウェアは GPL で公開されているので `gap-4.4-3v11.src.rpm` は、`/privaterpms/3.0/SRPMS.gpl` へ、`gap-4.4-3v11.src.rpm` は、`/privaterpms/3.0/i386/RPMS.gpl` へ保存する。

```
genbasedir /privaterpms/3.0/i386 gpl lgpl bsd
```

というコマンドでパッケージリストを更新する。また、パッケージが増える度に

```
genbasedir /privaterpms/3.0/i386 gpl lgpl bsd
```

を叩きパッケージリストを更新する。

/home/httpd/html/priv/Vine から/privaterpms へシンボリックリンクをはり, <http://jack/priv/Vine/3.0> で/privaterpms/3.0 が見えるようにした。

サーバサイドの設定などは以上。

これで,

```
rpm http://jack/priv/Vine 3.0/$(ARCH) gpl lgpl bsd
```

```
rpm-src http://jack/priv/Vine 3.0/$(ARCH) gpl lgpl bsd
```

などと/etc/apt/sources.list へ書いてあるマシンからはこの APT-line が利用できるようになった。もし, GPL, LGPL のもののみを利用したいのであれば,

```
rpm http://jack/priv/Vine 3.0/$(ARCH) gpl lgpl
```

```
rpm-src http://jack/priv/Vine 3.0/$(ARCH) gpl lgpl
```

のように書いておけば良い。