

センター運用におけるオープンソース・ソフトウェアの役割

名古屋大学情報連携基盤センター

長谷川 明生

Roles of Open Source Software on IT Center Operation

Nagoya University Information Technology Center

Akiumi HASEGAWA

梗概

オープンソースと呼ばれるソフトウェアは、基本的に無償で提供されており、各種のIT関連サーバの構築にひろく利用されている。このような種類のソフトウェアは、サーバ用のオペレーティングシステムから個人的趣味の分野のプログラムまで多様に存在する。ひろく普及しているPC向けハードウェアと、このような種類のソフトウェアの組み合わせで、ほとんどのソフトウェアシステムが実現可能である。現在の計算センターやネットワーク管理センターの運用システムは、このようなソフトウェアに強く依存している。本論文では、この種のソフトウェアのセンター運営における役割およびその得失について議論する。

Abstract

Open source software is provided basically free of charge and widely accepted in building IT servers. This kind of software covers the wide range of programs from operating systems for servers to private hobby use programs. Almost all kind of software systems can be built with combinations of popular PC hardware and this kind of software systems. Operation tasks or operation tools of current computation centers or network operation centers are highly depends on such software. The author will discuss on the roles of this type of software on center operation. Another discussion is also made on the merits and demerits of this kind of software.

キーワード：オープンソース，フリーウェア，センター運用

はじめに

インターネット技術の普及と、それにとまなうオープン化の流れによるUNIXオペレーティングシステムの普及は、単体の汎用大型計算機やスーパーコンピュータ中心の大型計算機センターや情報処理センターのシステムのありかたを大きく変容させた。システムの持つべき機能への要求は多様化しており、もはやシステム製造者が利用者の要求するソフトウェアのすべてを提供することは不可能となった。

一方で、ハードウェアの一部として提供されていたオペレーティングシステム等の基本ソフトウェアの有料化が定着してきた。このような動きに対抗して、コンピュータシステムの構築や運用に必須の基本ソフトウェアは無償であるべきだというFSF¹の考え方がその提供するソフトウェアとともに広く普及してきた。FSFのような動きやオープンソースの考え方は、ソフトウェアの共通基盤としてのUNIX系オペレーティングシステムの普及およびネットワークの発展によるソフトウェア流通形態の変化と非常に密接な関係にある。

ネットワークを検索すれば、よほど特殊な分野でないかぎり、必要なソフトウェアを入手することが可能となっている。このような動きの中で、センターの運営において、オープンソースやフリーソフトウェア²を無視できない。ここで、そのセンターシステム運用における役割について論じる。

オープンソースのセンターでの利用

オープンソース・ソフトウェアと呼ばれる範疇のソフトウェアには、利用条件の非常にゆるやかなGPL³に代表されるものから、かなり厳格な基準に基づいてのみソースが公開される商用ソフトウェアまでであるが、ここでは、GPLもしくはGPLに準拠したソフトウェアを議論の対象とする。

表1 主要ソフトウェア一覧

| 分類 | 名称 |
|--------|------------|
| 編集系 | Emacs |
| | Xemacs |
| 言語処理系 | gcc |
| | scm |
| | gcl |
| | ruby |
| | Tcl/tk |
| ネットワーク | bind |
| | sendmail |
| | ncftp |
| | CML |
| | inn |
| | mh |
| | mnews |
| | openssh |
| | openssl |
| apache | |
| 文書処理 | xv |
| | freeWnn |
| | ptex |
| | groff |
| | Xpdf |
| セキュリティ | tcpwrapper |
| | chkrootkit |

表2 ネットワーク用

| 分類 | 名称 |
|----------|-----------------|
| 通信監視 | MRTG |
| | Analog |
| セキュリティ監視 | Snort |
| | Swatch |
| セキュリティ監査 | Nmap |
| | Nessus |
| | Crack |
| | John the Ripper |
| メールサービス | Postfix |

表1に、名古屋大学情報連携基盤センターのメールサービス用のシステム上で利用している主なオープンソースのソフトウェアを示す。表1に示したものは、運用上不可欠で主に利用しているものである。このホストでは、商用コンパイラの導入によるライセンスコストを削減するために、システム構築そのものをgccを用いておこなっている。

計算サービスを目的としたベクトルプロセッサVPP-5000システム上にも、プログラム開発や遠隔利用者のためにemacs環境およびSSHを移植した。

表2に、ネットワーク管理や全学ネットワークサービスのために使用しているソフトウェアを示す。

表1や2に示したものの他に、gzip等の圧縮ソフトウェアやpatch等の多数のユーティリティソフトウェアをセンター運用に利用している。なによりも運用管理用ソフトウェアや課金統計情報の管理およびWebのCGI作

成等にPerlやRubyといったスクリプト言語は必須である。研究上でも、FreeBSDやLinuxおよびそれらをもととしたSELinux等のオペレーティングシステムおよびGlobusツールキット等の大規模なソフトウェアも日常的に利用しており、研究の素材として使われている。

導入および運用コスト

オープンソース・ソフトウェアの入手は無償であることが多いが、これらのソフトウェアを使用するためには、最低限ネットワークダウンロード作業と「make install」程度の作業は必要である。また、個別のソフトウェアのインストールそのものは単純な作業であっても、ソフトウェアの種類が多量になれば、トータルの作業コストは増大する。さらに、運用方針の異なるシステムが複数存在する場合には、システム数に応じて加速度的にコストは増大する。ソフトウェアのインストールは、現実には、ソフトウェア間の依存関係やコンパイラ、共有ライブラリの互換性の問題があり「make install」だけでは済まないことが多い。そのような例を表3に示す。コンパイラとオペレーティングシステムおよびソフトウェアの組み合わせにより、コンパイルできない場合が、実際に存在する。また、ソフトウェアによっては、コンパイラの種類やバージョンまで指定されている場合がある。

さらに、導入しようとするシステムのオペレーティングシステムが

表3 ソフトウェアとコンパイラの相性

| 環境 | コンパイラ | Emacs | Mew |
|------|-----------|-------|------|
| | | 21.3版 | 3.3版 |
| Sol9 | Gcc-3.3.2 | × | ○ |
| | Forte7 | ○ | ○ |
| Sol7 | Gcc-3.1 | ○ | ○ |
| | Forte6 | ○ | × |

configureのリストにないスーパーコンピュータやリストに存在しても十分にテストされていないシステムの場合、ソースプログラムから調べてポーティングするという作業が必要である。

導入した後も、商用システムにクラスタパッチとよばれるような定期的なソフトウェア保守が存在すると同様に保守が欠かせない。保守を無視すれば、機能の陳腐化による利用者離れ、オペレーティングシステムの保守やバージョンアップに原因する動作不良の発生だけでなく、ソフトウェアによってはopenSSHやOpenSSLのバグの場合のようにセキュリティ上の危機を引き起こす。表4に、本センターの汎用計算サーバ上での、gccおよびemacsのバージョンの変遷を示す。Emacsのバージョンは、スーパーコンピュータ上のEmacsとバージョンをあわせるためである。Gccは、2.4のバイナリーパッケージを初期導入し、そのコンパイラで、2.95.2を作成し、その後、導入ソフトウェアの都合にあわせるために一度ダウングレードして、2001年に再度3.0.1にしている。

表4 ソフトウェアバージョンの変遷

| Gcc | | Emacs | |
|--------|-------|-------|-------|
| 版数 | 時期 | 版数 | 時期 |
| 2.95.2 | 99.10 | 20.5a | 99.12 |
| 2.8.1 | 00.01 | 20.6 | 00.03 |
| 3.0.1 | 01.09 | 20.7 | 00.06 |

これらのソフトウェアについて、著者の作業用ワークステーションには、ほぼ最新のバージョンが評価用としてインストールしてある。

これら一連の作業にかかるコストは無視できるものではない。このようなコストのうち最大の人的コストの評価尺度として、たとえば「人・日」がある。しかし、一般にソフトウェアの導入や保守作業では、作業者の熟練度が低いと人・日が高く出る傾向がある。この補正手段として、作業者に依存するファクターをかけてコストを算出することも行われているが、ファクターそのものの算出に困難がある。

人的コスト以外に、ディスク等のハードウェア資源の問題もある。多くの利用者へのサービスを前提としたシステムでは、一旦公開したソフトウェアは、利用者が存在するかぎり、代替なしで廃止することは困難である。一方で、商用ソフトウェアと異なり、オープンソース・ソフトウェアでは、ソフトウェアの改廃が激しく、ソフトウェア自身の変更やOSのバージョンアップによって、動作しなくなったり(Nemacs等)、場合によっては、開発者が不在となったりすることがある。ソフトウェアの作成時期が古くて、新

しいコンパイラではmakeできないが利用頻度の高いものや、コンパイラを更新しないとコンパイルできない新しいソフトウェアへの要求が常に存在している。多くのアプリケーションの要求するWidgetやライブラリ群についても、同様である。その結果として、常に版数の異なるだけの共有ライブラリを保持しておく必要がある。表4で示したように、名古屋大学の汎用計算サーバ上では、gccの異なるバージョン2.8, 2.95.2および3.0.1用のライブラリが残されている。

オープンソースの得失

オープンソース・ソフトウェアの特長を以下に列挙する。

- ソースプログラムが公開されており、改変が可能なこと。
- バグフィックスが早いこと。
- 入手が無償であること。
- 最新の技術やアルゴリズムが早期に導入されること。
- 技術的ノウハウがインターネット上に蓄積されている。

これに対して、問題点としては、以下のようなことが考えられる。

- ソースプログラムが公開されているため、セキュリティ上の弱点が発見されやすいこと。
- バージョンの改変が早すぎること。陳腐化が早いこと。
- 仕様の変更が激しいこと。常に開発中であること。
- ソフトウェアの継続性に問題が

あることがある。

- 品質にばらつきがあること。
- ドキュメントが整備されていない。

オープンソース・ソフトウェアの得失点は表裏一体である。センターでのサービスに利用する立場では、最新の技法やアルゴリズムを使ったソフトウェアを早期に利用できるという利点は、一方ではソフトウェアの寿命や継続性に問題をかかえることでもある。実際、NemacsからMuleへの移行や、かな漢字変換ソフトウェアの有償化への対応には苦慮した経験がある。

サービスの継続性ととも、センター運用で、もっとも気をつかうのはセキュリティに関する問題である。オープンソースの場合、原則として自己責任であるのに対して、商用のソフトウェアの場合、契約条項の問題もあるが、問題が発覚した場合に苦情先が存在するという利点は大きい。

オープンソースとセンター運用再考

センターの運用にとって、利用者の意向は最優先されるべきもので、いまどきUNIXといいながら、Emacsが動作しないシステムでは見向きもされない。また、TeXで旧字体とか、科研マクロを使いたいという要望があれば、オープンソースの山を探して、インストールするというのが日常であろう。

PCクラスタとなれば、Linuxは必須で、運用を担当する技術者はセキュリティと利便性の両立に頭を抱えるこ

とになる。が、汎用コンピュータ一台で、多くの要望が処理可能だった時代と異なり、あらゆる要求にこたえるにはオープンソース・ソフトウェアの利用は必然である。

残る問題は、それにかかるコストをどう処理するかにかかっている。その解決として、

1. 外部に任せてしまう。
2. 内部で抱える。

のどちらかしかない。

外部委託について

オープンソースソフトウェアに限らず、いろいろな点で外部委託が話題に上る。外部委託に関する問題点や疑問を以下に記述しておく。

- 本当に安いのか？ 運用のための作業量は、だれが実行しても同じではないか？
- 何を委託するのか、厳格な仕様をオープンソースのソフトウェアについて記述できるのか？ 無限責任は、発注側には楽ではあるが。
- だれが、委託の質や結果を判断し、評価するのか？ 正しい評価が可能か？
- 外部委託が長期化した場合、運用能力や作業の評価能力をどのように維持可能か？

運用コストを下げるためには、操作性の高いソフトウェアが必要である。一般に、操作性を要求すると商用ソフトウェアが優位に立つ。基幹業務に、運用コストを考えて商用ソフトウ

アを導入してしまうと、技術的な業者依存だけでなく、ライセンス料金の高騰といった問題も発生する。

技術力を保ちつつアウトソーシングが可能かどうかは十分に検討しなければならない問題である。

センターとして、技術力を保ち続けるための方策のひとつとして、オープンソースソフトウェアの活用が重要である。

おわりに

オープンソース・ソフトウェアのセンター運用での利用について、概観した。実際のセンター運用において、オープンソース・ソフトウェアの利用は、常識となっているが、その導入や保守についてのコストも無視できないものがある。

この種のソフトウェアの利用や流通のありかたについても、今後とも議論していく必要がある。

なお、ここで上げたソフトウェアについては、一般的に著作権等で保護されている。また、本論分で引用したソフトウェアは、著名なものばかりで、また数が多いこともあり逐一出典を引用することはしない。

参考文献

- 1) <http://www.fsf.org/philosophy/philosophy.html>
- 2) <http://www.fsf.org/philosophy/free-software-for-freedom.html>
- 3) <http://www.fsf.org/copyleft/copyleft.html>