

3. 教育用計算機環境に適したオープンソースデスクトップ



(有) ヴァインカーブ

松林 弘治
shaolin@vinecaves.com

やまだ あきら
akira@vinecaves.com

鈴木 大輔
daisuke@vinecaves.com

概要とねらい

大学等の高等教育機関では、コンピュータリテラシーの広範な修得を目的として、すべての学生を対象にした大規模な演習室の設置を行っていることが多い。近年、これらの教育用計算機システムとしてLinuxを始めとするオープンソースソフトウェアを中心に構築されたクライアント環境の導入事例が増えてきている。

本稿では、我々が大阪大学サイバーメディアセンターに納入したVine Linuxをベースとして構築した教育用クライアント環境の開発事例から得られた知見を紹介する。また、今回の作業を通じて明らかになった問題点、今後の課題など、大規模分散ネットワーク環境での教育用システム構築を構築する上で考慮しなければならない点について考察を加える。

教育用計算機環境とオープンソース

大学に代表される高等教育機関では、コンピュータリテラシーの修得を主な目的として、全学生を対象とした情報教育、あるいは計算機利用を取り入れたさまざまな授業が広く行われてきている。そのために用いられるコンピュータ環境の規模は教育機関ごとに異なるが数百台から数千台に及ぶところもある。

これだけ多くの台数ともなると、個々の計算機を維持

管理し、発生した障害への対処を行うために膨大な量の作業が生じることが多い。また、さまざまな分野の授業や実習で使われるため、それぞれで必要とされるソフトウェアや利用環境への要請も幅広く、柔軟かつ迅速な運用が求められる。

昨今、このような大規模な教育用計算機システムとして、Linuxを始めとするオープンソースソフトウェアの導入事例が増えてきている。コスト的なメリットのほか、もともと大学では以前よりUNIXが研究・教育に幅広く利用されてきた経緯があり、蓄積されてきた技術的ノウハウをUNIX互換OSであるLinuxへ比較的容易に適用できるというメリットもあろう。また、再頒布の自由やソースコードの公開といったオープンソースソフトウェアの特徴に対するシンパシーや、それらの教育的効果への期待もあるのではないかと考えられる。

大阪大学サイバーメディアセンターでの事例

大阪大学サイバーメディアセンター（以下CMC）^{☆1}では、学生向けの計算機教育環境として約1,000台のコンピュータが配備されており、共通教育や学部専門科目などで利用されている。

登録ユーザ数は2万人を超え、各ユーザのホームディレクトリは豊中・吹田の両キャンパスごとに設置されたファイルサーバ上に置かれている。その他、アカウント

^{☆1} <http://www.cmc.osaka-u.ac.jp/index.html>

情報を管理するサーバ、アプリケーションサーバ、メールサーバ、ネットニュースサーバ等が設置されている。

CMCの教育用計算機システムは、2000年度に大規模なリプレースが行われ、その際にLinuxをベースとしたクライアント環境が導入された。その後2003年にソフトウェア環境の更新が行われ、我々はVine Linux^{☆2}バージョン2.6r1をベースとした教育機関向け環境のパッケージであるVine Linux Educational Edition^{☆3}を納入し、現在までCMCのスタッフと連携しながらメンテナンス作業を行っている。

今回の事例においては、サーバ環境(CMC利用者のアカウント情報、ディスクスペース、メールサーバ等)はCMC既存のものを使ったため、我々はクライアント側環境の構築およびメンテナンスのみに従事した。

要求事項

CMCのスタッフとのミーティング等を通じて、クライアント環境の実装やカスタマイズにおいて前提となる事項が挙げられた。以下はその中の主だったものである。

• 全コンピュータの遠隔によるメンテナンスが可能であること

大規模教育用計算機環境においては、CMCでは1,000台規模であるように、対象となるコンピュータの台数が非常に多くなることから、実際に1台ずつコンピュータの前に行かずにメンテナンスが行えることが必須である。ハードウェアの物理的な障害の場合は直接計算機の前に行かなければいけないが、ログイン画面が出てこなくなった、プログラムが暴走して操作不可能になった、といった、ソフトウェア的なトラブルが起こったコンピュータの復旧作業を遠隔から行えることは重要である。また同時に、ハードウェアトラブルも含めた各コンピュータの状態を常に把握しておくために行うログ管理も遠隔から行える必要がある。

• 複数種類のハードウェアの存在を許容できる仕組みであること

大規模教育用計算機環境では、通常同一のハードウェア構成を持ったクライアントが大量に導入されることが多い。しかしながら、運用途中で別予算でハードウェアが追加されることも少なくないし、利用目的を異にした演習室単位でハードウェア構成を変えているケースもあり得る。そのため、単一のハードウェア構成を前提とし

た管理システムではなく、ある程度のバリエーションに対応可能な仕組みが必要となる。

• 全コンピュータの修正やアップデートが容易であること

突発的なトラブルではなく、各コンピュータのシステムを構成しているプログラム群のバグによる動作の不具合が明らかになった場合や、プログラムにセキュリティ上の問題が発見された場合は、それらの問題を修正した上で、全コンピュータに修正プログラムをインストールする必要がある。この時、1台ずつインストール作業を手作業で行うことになれば大変な労力になってしまう。そのため、修正済プログラムを全コンピュータに配布しインストールするまでを自動化する仕組みが必要となる。

• 全コンピュータの状態が基本的に同一に保たれること

どのコンピュータを使用しても、利用できるプログラムや使用感に差が生じないようにすることは、すべてのユーザに同一の環境を提供するために必要である。このようにユーザがどのコンピュータを利用するかにかかわらず同一の環境を提供するため、すべてのコンピュータが基本的に同じ設定であり、また同じプログラムが利用可能でなければならない。サイト/演習室単位で特定のプログラムを追加で入れておくといった運用も考えられるが、この場合も、サイト内の全コンピュータが同一の環境となるようにしなければならない。

• ユーザがシステムに変更を加えられないようにすること

上に関連して、多数のユーザが不特定のコンピュータを利用する教育用環境では、独自のプログラムをインストールする、せっかく整えてあるコンピュータ上の環境を勝手に変更されてしまうなど、コンピュータの状態を一般の利用者が変更できるようではメンテナンスの面からもセキュリティの観点からも好ましくない。そのため、管理者ではない一般のユーザがシステムに変更を加えられないような対策が必要となる。

• ユーザの個別設定と、サイト/ホスト依存設定を切り分けられること

各ユーザがログインした環境内で個別に施した変更内容と、管理するホストやサイトに依存する設定内容は、管理上の利便性から明確に区分されるようになっていなければならない。これはすなわち、各種設定やカスタマイズがどのレイヤで行われるべきかを明確にし、各レイヤでの設定が明確に区分されるような仕組みを構築しなければならない。

☆2 <http://vinelinux.org/>

☆3 <http://vinecaves.com/eduvine.html>

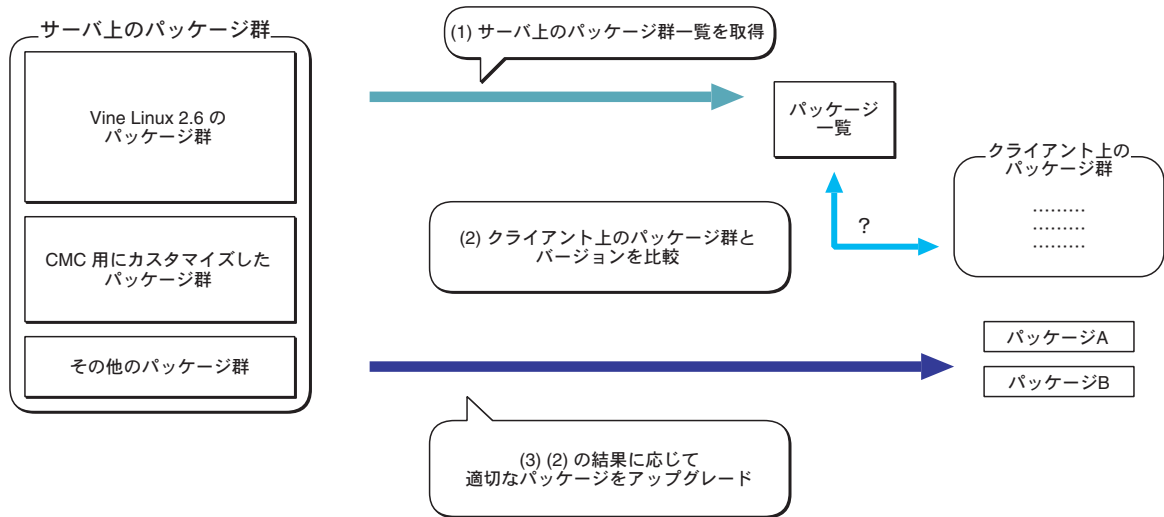


図-1 パッケージ管理システムの稼働イメージ

• GUI ツールを拡充すること

各ユーザ固有の設定項目を保持している初期設定ファイル群がなんらかの理由で破損してしまった場合、プログラムが正常に動かなくなったり、ログインがうまくできなくなってしまう可能性がある。このような場合、設定ファイル群の仕組みをよく知らないユーザでも簡単に初期状態に戻すことができる簡単な GUI ツールを提供することは、管理上の観点からも非常に有用である。

また、ログインパネル上に管理者からのお知らせを表示するパネルや、プリンタで印刷する際に簡単に用紙サイズや用紙の方向などを表示するためのダイアログウィンドウといった、現時点の Vine Linux には備わっていないが大規模教育用環境では必要とされる GUI ツール群も用意されなければならない。

今回の CMC における作業では、これらの GUI ツールのほとんどはすでに CMC で作られ利用されているものを引き続き使用した。

実装と運用

これらの点を踏まえ、我々は Vine Linux 2.6r1 をベースにクライアント環境の構築を行った。以下、実際に実装し運用しているシステムの中から一部を記す。

パッケージ管理システム

多くの Linux ディストリビューションでは、システム

全体を構成するプログラムや共有ライブラリ、付随ファイル群などを「パッケージ」という単位で管理している。あるプログラムをバージョンアップする際には、そのプログラムを含むパッケージをより新しいものと差し替えることで、システムの状態を更新する。今回導入した Vine Linux では、rpm (RPM Package Manager) ^{☆4} というパッケージシステムを採用している。

Vine Linux ではさらに、Debian GNU/Linux 由来の apt (Advanced Packaging Tool) ^{☆5} というパッケージ管理ツールも採用しており、各パッケージ間の依存関係を解消しつつパッケージのインストールやアップグレードを統一的行うことが可能である。

これらの仕組みをバックエンドに用いて、上述の「全コンピュータのアップデート」や「全コンピュータの状態を基本的に同一に保つ」ことを実現した。具体的には、クライアントの電源投入後の起動時に、各コンピュータはファイルサーバ上にあるパッケージの一覧を取得する。その後、自らにインストールされているパッケージの一覧と比較を行い、より新しいバージョンのパッケージが存在すればアップグレードを行う。また、特定のファイルにパッケージ名を記述しておくことで、新たなパッケージのインストールや、すでにインストールされているパッケージの削除を行うことができる。

また、システムが動作中にも定期的パッケージ情報を取得し、更新されたパッケージが見つければバックグラウンドでパッケージのアップグレードを行う (図-1)。この際、システムの再起動を行うが、現在口

☆4 <http://www.rpm.org/>
 ☆5 <http://www.debian.org/doc/manuals/apt-howto/index.en.html>

グインしているユーザが存在する場合には、再起動を遅延するようになっている。

システム設定ファイルの管理

システム全体の設定や各クライアント固有の設定を行うファイル群については、管理上のメリットから上のパッケージとは別に CVS と呼ばれるバージョン管理システムを用いて管理を行っている。クライアントはシステム起動時にサーバ上の設定ファイル群と同期を行い、最新の状態を取得する。この場合も、必要であれば再起動を行う。

システム起動時のインタラクションの抑制

Vine Linux を含めた Linux ディストリビューションでは、システム起動時に新たな周辺機器が認識された場合など、コンソールからの入力を促す仕掛けが用意されていることが多い。しかし、教育用大規模計算機環境においては、遠隔操作ができる状態になる前にシステムが一時停止した状態になってしまったり、たまたまコンソール近くにいたユーザに操作されてしまったりすることは避けなければならない。そのため、起動時にコンソールでの対話的操作を要求する類のスクリプトや該当プログラムは削除した。

代わりに、各クライアントに装着されている拡張カードや周辺機器などを自動設定する仕組みを導入している。一般的には完璧なハードウェア完全自動検出というのは難しいが、大規模教育用計算機環境では、ごく一部の例外的なマシンを除いて、ハードウェア構成のバリエーションが比較的限定されているため、複雑な自動検出等の仕組みを使わずに個々の設定を施すことが可能である。

ユーザのデフォルト値の自動設定、その他細かい作り込み

全ユーザが最初にログインした時のデフォルト設定値（デスクトップ環境の細かい設定項目や、各種プログラム上の設定内容）をどのようにするか、これについては一般的な法則や最適値というものがない。ただ、実際の教育現場での声を統合することにより、その現場特有の適切な初期設定を行うことは有用であるし、強く望まれていることでもある。

まず、通常ユーザが各種プログラムの初回利用時に手で設定する必要のある項目（電子メールアドレス、SMTP/IMAP サーバ、proxy サーバ等）を自動で設定するように作り込みを行った。これらの項目内容については、実際の利用環境で望まれる設定値を使った。

デスクトップ環境は Vine Linux 2.6 標準の GNOME 1.4 を採用したが、メニュー項目やランチャーパネル、デスクトップ上のアイコン配置等、以前のシステムと使用感をできるだけ共通化するために作り込みを行った。その

後、現場の教育スタッフや CMC スタッフからあがった要求に応じて初期画面の修正を行ったりもした。その他 CMC スタッフからの要求に応じて、各種プログラムの細かな変更等を行い、必要に応じて CMC 専用のパッケージの作成も行った。

サイトごとやユーザごとのデフォルト設定値を変更する作業の中で気になったことは、一部のプログラムでは、設定項目が別途ファイルに保管されているのではなく、コンパイル前のソース内に埋め込まれているものである。この場合、設定値を変更するたびにソースコードを修正した後コンパイルをし直し、全クライアントで作成した rpm パッケージのアップグレードを行う必要があり、非常に煩雑であった。

Web ブラウザの mozilla など一部のプログラムでは、二重起動を防止するために、起動中はロックファイルを作成するようになっている。ところが、正常終了したにもかかわらずロックファイルが残ったままになるケースがまれにみられたり、コンピュータの電源が突然落ちた時にロックファイルが残る場合が発生したりした。そのため、プログラムが起動していないにもかかわらずロックファイルが残っている場合にこれを安全に削除できるツールを作成している。

現状の考察と展望

今回の実装および運用、メンテナンスの作業を通じて、CMC のような大規模クライアント環境に特有な要件や、考慮すべき点に多数ゆきあたった。これらの中でより一般的だと思われる件について紹介し、現時点での対策、実装運用例や、今後の方向性について述べる。

ファイル共有環境での反応の遅延

ユーザの利用可能なディスクスペースはネットワークを介したサーバ上にあり、ユーザがログインした直後に自動でマウントされる。ログイン処理中、アプリケーションプログラムの起動中、ログアウト処理中など、ユーザのホームディレクトリ上のファイルは頻繁に参照され、更新される。CMC のように 1,000 台規模のクライアント台数となると、サーバへのアクセス集中やネットワークの混雑、あるいはファイル共有の仕組みそのものの問題などにより、クライアントの遅延が著しく増加してしまう場合がある。

ネットワーク環境のさらなる充実やクライアント/サーバ処理能力の向上をはかることにより、いくばくかは解消されようが、これは根本的な解決とはならない可能性がある。クライアントのコンピュータ側でできる対処としては、アプリケーションの使用設定ファイル群

が極力最小化されるように必要なら修正を施し、余計なディスクアクセスが生じないようにすることや、速度低下によりユーザが焦燥感を覚えたり誤操作を行ったりしないように配慮することである。

前者は、サイトの管理ポリシーとの摺り合わせ作業や、アプリケーションごとの子細な作り込みが必要となる。後者については、必要に応じてアプリケーション起動時のスプラッシュを表示させることでユーザに時間がかかることを知らせる仕組みや、アプリケーションの二重起動を防止する仕組みなどが挙げられる。

ユーザ初期設定ファイル群損傷時の対応

ユーザ領域にある初期設定ファイルがなんらかの理由で破損してしまった場合、アプリケーションの動作が不安定になったり、起動しなくなってしまうことがある。通常こういった場合には、設定ファイルを削除したり、特定の設定ファイルの内容を手動で修正してから、アプリケーションを再起動することになる。しかしながら、全学部さまざまな授業で利用されることが前提である教育用環境の場合、すべてのユーザに手動で修復を要求することは難しい。

そのため、ユーザ領域に置かれているさまざまなファイルのうち初期設定ファイル群については、簡単な操作でそれらを再初期化できる仕組みを提供しておくのが有効である。現状のCMCの環境では、(主にアプリケーション側の問題によって)過去に初期設定ファイル破損のトラブルが報告されたアプリケーションについては、各アプリケーションごとに設定ファイルを再初期化するためのメニューを提供している。

また今後の課題として、アプリケーション側のつくり方を工夫することも考えられる。ユーザ領域に置かれている初期設定ファイル群をいちいち初期化したもので置き換えるのではなく、削除した上で起動しただけで初期設定状態になるようにアプリケーション側に対応を含める、というアプローチは有効であると思われる。

パッケージ更新のテスト環境

現在、バグ修正やセキュリティ修正が施されたパッケージは、即座にファイルサーバ上に配置するのではなく、まず最初に我々やCMCスタッフの個別の環境で動作テストを行っている。それでもなお、パッケージを配布した後に、実際のCMCクライアント環境上で動作の不具合等が見つかることがある。この際、数多くのクライアント機でほぼ同時期に不具合が発生することになり、管理者の負荷が急激に増大してしまう。

この問題に対処するために、サーバ上に配置したパッケージ群のスナップショットのようなものを複数作成しておいて、必要に応じて各コンピュータの内容を指定し

たスナップショットに同期させるといった仕組みについて研究中である。これにより、新しくサーバ上に配置したパッケージに問題が発覚した場合でも、1つ前のスナップショットにロールバックすることにより、当座の混乱を回避することが可能となる。

またこの仕組みは、複数のクライアント環境を統一的に扱うためにも役立てることができると考えられる。たとえば、ある演習室と別の演習室で、インストールされているパッケージ群を違えたい場合など、異なるクライアント環境を複数用意したい場合に、それぞれのクライアント環境用にスナップショットを用意しておくことができる(図-2)。

バグの報告と対応の効率化

実際に多数のユーザが同時に利用する教育用計算機環境では、予想もつかないようなトラブルが発生したり、いままでも明らかにならなかったプログラムの不具合が顕在化することが予想される。そのため、トラブルの発生状況によっては、ユーザやティーチングアシスタント、教員、計算機施設のスタッフ、業者の間で情報が錯綜する恐れがあり、保守作業が十分にできない場合もあり得る。

今回のCMCでの作業においては、CMCのスタッフと我々が電子メールベースで情報を交換しながら作業を行ったが、複数の問題の発生やそれらの対応作業が並行して起こる今回のようなケースでは、情報のとりこぼしや対応の遅延などが発生してしまうことがあった。

このような問題を解消する手段として、近い将来にバグ管理システムの導入を予定している。バグ管理システムは主にオープンソースソフトウェアの開発現場で、複数の開発者やユーザとの間でソフトウェアのバグに関する情報を共有し、開発の効率化をはかる目的で広く使われている。バグ管理システムには数多くの実装が存在するが、中には大がかりな実装であったり操作が複雑なバグ管理システムもあり、使いかたを誤れば逆に作業の効率を低下しかねない。利用ケースに応じたバグ管理システムの選定と有効な運用自体が重要な課題であるといえる。

プロプライエタリソフトウェアにまつわる問題

教育用計算機システムの現場では、さまざまな理由から一部でプロプライエタリソフトウェアを利用している場合がある。主な理由として、代替となるオープンソースソフトウェアが現時点では存在しなかったり、存在していても開発途上であるため機能が不安定であったりすることがあげられる。CMCのクライアント環境では、Webブラウザ用の一部のプラグイン、オフィスツール、日本語入力システム、数式処理ソフト、統計パッケージ等でプロプライエタリソフトウェアが導入さ

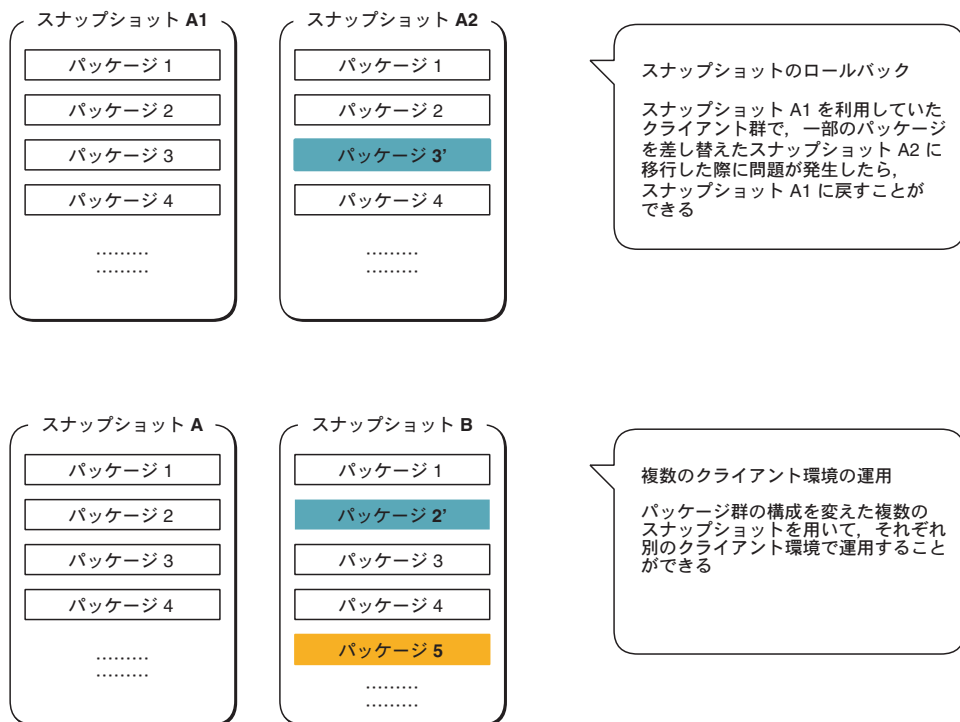


図-2 パッケージ群のスナップショットによる複数環境での運用

れている。

これらのソフトウェアで動作の不具合が明らかになった場合、その原因がベースとなる Linux ディストリビューション側にあるのか、それらのソフトウェア側にあるのか、またどのように対処が可能かを調査する必要がある。残念ながら、プロプライエタリソフトウェア側に問題があると思われるいくつかのケースでは、十分な対応ができない場合があった。

今後の課題

現時点のクライアントシステムの構成では、CMC 固有の要求に基づく実装という側面が多いのは否めない。実際に使用される組織によって運用ポリシーや利用形態が異なるため、固有のカスタマイズや作り込みは必ず必要であるが、そこからさまざまなケースを想定した汎用的な技術的枠組みをより明確に切り出し実装してゆくことは今後の作業でも最も重要な点と考えている。

今回の実装と運用の作業を通じて得ることのできた、教育用大規模クライアント環境におけるさまざまな要件を元に、より汎用的なサイト管理システムのバックエンドの開発を行ってゆきたい。

謝辞 本システムの開発、導入、運用、保守の全過程を通じて、大阪大学サイバーメディアセンター情報メディア教育研究部門の町田貴史助手、小川剛史助手、榊田秀夫助手、中澤篤志講師、清川清助教授、竹村治雄教授から多大なるご協力をいただいたことを深く感謝いたします。

参考文献

- 1) 情報教育用計算機システム－新しい教育用計算機システムの特徴、サイバーメディアフォーラム No.1, <http://www.cmc.osaka-u.ac.jp/publication/for-2000/index.html> (2000).

(平成 16 年 2 月 5 日受付)

