Linux クラスターによる教育・研究両用システムの構築

CAUA 第一回研究教育分科会 (2000 年 10 月 13 日) 予稿

宮澤 三造 (群馬大学工学部共通研究室, miyazawa@smlab.sci.gunma-u.ac.jp)

要旨

工学部学生の情報処理演習用に使用する 120 台の Linux PC を、演習用のデスクトップとしてのみならずクラスター構成により研究用に使用するシステムを構築運用している。クラスターとして使用するためには、 24 時間稼働は前提である。リブートできないようなハード、ソフト両面での設定も必要である。システムの概要をご報告する。

1 はじめに

従来、学生教育用に使用する端末は、主に管理運用の観点からシステムを構築するのが通常である。例えば、UNIX システムの場合には、ワークステーション (WS) のコスト、管理運用のコストの軽減のため、 X- 端末を用いるのが常であった。しかし、近年 WS も急激に安価になり、また PC UNIX を利用する場合には価格的にはむしろ X- 端末より安価になった。 Thin client とのハードの違いは、ディスクがあるかないかである。ホワイトボックスの場合には、性能及び信頼性の観点からパーツを選び組合せることができる利点だけでなく、価格もメーカー品をはるかに下回る。ディスクの追加があっても Thin client より安価である。このような状況では、X- 端末は WS 管理運用のコスト軽減以外に選択する理由を見出せない。 Windows を使用する環境においても管理コストを考えてか、 Windows 端末も販売されている。しかし、管理コストはシステムの運用次第で Thin client 同様のコストに押えられる。大学の計算 / 教育センターにおいては、何よりも管理運用に携われる人的資源に乏しいから、管理運用のコストは機器選定の最も重要なファクターである。私としても、教育用としてしか使用できないのであれば、他を少々犠牲にしても、管理運用の最も容易なシステムを構築したいと思う。一方、当センターにおいては、乏しい予算を遣り繰りして、研究用のシステムもサポートしなければならない。予算の面から考えると、教育用デスクトップの使用率は低く、はなはだもったいない。演習に使用する時間帯は 80-100% の使用率、自習用に開放している時間帯 (8 時 30 分から 22 時) で 20% 程度であろう。 1 年を通して考えれば 10-20% 程度の使用率にしかならない。つまり、少なくとも 80% は無駄に PC を遊ばせていることになる。その意味では、教育用 PC の利用は改善の余地がある。これは教育用 PC だけではなく single user 用の PC のほとんど全てについて言えることであるが、 1-2 年で使用するに耐えなくなる計算機を使用せずほって置くのは無駄である。限りなく研究予算を削られつつある大学の研究者としては、教育用 PC を学生の使用していない時間、遊ばせておく手はないであろう。もちろん Windows を使用する場合は打つ手がないのであるが、幸いにして私の関与する桐生地区における教育システムは、工学部の 2 年次以上の学生を対象としている関係で UNIX 系 OS のみで問題ない。 (ちなみに、群馬大学は 3 キャンパスからなるが、他キャンパスでは Windows のみか、 Windows / linux の dual system を採用している。) 桐生地区における教育システムは計 120 台のデスクトップが必要である。そこですべて UNIX 系 OS で稼働させ、クラスタ構成で研究用にも使用できるようなシステムを構成した。教育研究両用システムとして運用するためには考慮しなければならない問題点がいくつかある。システムの概略をご報告する。

2 ハードウエア構成

まずは、本報告に直接関係していると思われる範囲で、システム構成を簡単に述べておこう。 PC のハードウエア構成は、予算の制約もあり教育用のデスクトップとしての要請のみを考慮して仕様を決めた。それ故、研究用のクラスターとしては CPU が非力ではある。しかし、導入後 1-2 年は台数 (120 台) による効果で十分使いものになろう。表 1 には、参考のためサーバー及び研究用の Dual Alpha 21264A 750MHz の 18 台からなるクラスターについても概略を記した。いずれもホワイトボックスである。

表 1: ハードウエア構成

	PC x 120	PC server x 6	Alpha PC x 18
cpu	Celeron 500MHz	Pentiume II 750 MHz	Alpha 21264A 750MHz x 2
mainboard	A Open AX6BC Type R (Intel 440BX)	TYAN Tiger 100	DP264
memory	ECC 128 MB	ECC 1 GB	ECC 1 GB x 16/2GB x 2
system disk	Seagate (ST36421A) ATA 6 GB	Ultra 160 9 GB	Ultra 2 Wide 9GB
network	3Com 3C905B (10/100Base TX)	3Com 3C905B	3Com 3C905B
video card	Matrox G400	Matrox G200	3Dlabs
sound card	約 20 台程に CREATIVE Vibra 128	-	-

ハードウエアに関して最も重要なことは、24 時間稼働を前提にして信頼できるパーツが得られるように仕様書を記述するということである。私供も随分注意して記述した。例えば、ネットワークカードは chip は同じでも性能にバラツキがあるので、chip メーカーの純正品であること等々を要求した。しかし、残念ながら (予算に比し) 満足のいくパーツが得られたとは言えない。例えば、main board はマニヤ向け製品で正しい選択であったのか? 納入メーカーの事情でこちらの望む製品ではなかった。そのためでもないだろうが、 CPU 電圧が不適当でリブート時にブートしない PC が続出し、原因調査に無駄な時間を使わねばならなかった。 Windows としての使用では、1 度目でブートしなくともおかしいなということですんでしまうが、ブートフェールがおきるようではリモート管理は不可能になる。これは一般的に言えることだが、 PC のハードウエアは初期不良が高頻度なので、十分なチェックを要求することである。チェック項目として必須なのは、 1) メモリーの read/write 2) (特に IDE Disk の場合は) ディスクの read/write テストである。納入後も、3-4 週間で理由もなく停止してしまうような PC の場合は、第一にメモリー、次にメインボードの不良である。台数が多いので原因を突き止める人的余力はない。このような場合は全交換で対処することを納入メーカーに要求した。ついでながら、 PC のメインテナンスは予備機を用意し send-back 方式である。

これらの計算機は、いずれも 100BaseTX で Layer 2 Switch に図 1 のように接続されている。 (ディスクサーバー 2 台は 1G BaseFX である。) (研究用 Alpha クラスターを含め)、現システムは 100BaseTX での接続のため本格的な並列計算用のクラスターとは言えないが、コストを考えると最善であると思う。

3 24 時間稼働のためのセキュリティー対策

さて、 PC cluster として使用するには何より 24 時間稼働する環境づくりが欠かせない。学生によるリブート、シャットダウン、システムの変更等が可能であってはならない。 そのため、 1) 誤って押したり、画面がフリーズした際に押してもシャットダウンしないよう、リセットボタン、電源ボタンを無効にした。 2) linux では画面がフリーズした際にリブートするための CTRL-ALT-DEL が用意されているが、 inittab で無効にする必要がある。 3) BIOS の変更ができないようパスワードでロック、 4) ブート時にシングルユーザーモードなどでブートを試みることができないようブートローダーを設定する必要がある。このシステムでは lilo ではなく汎用ブートローダーである GRUB [1] を用いている。

もちろん、電源ケーブルを抜けばシャットダウンさせることは可能であるが、そこまで防止することはコストを考えれば特策ではなかろう。では盗難防止の面ではどうであろう。 BIOS はロックしてあるが筐体を開ければ、BIOS の初期化も可能である。またメモリーの盗難も考えられる。 そのため、筐体のパネルは特殊ネジを使用し普通のドライバーでは開けることができないように工夫した。私の案ではなかったのだが、特殊ネジ自体は非常に安価であるので盗難防止には名案であろうと思う。

また、Floppy disk と CD-ROM は、悪用されないようなシステム設定をすることは簡単だが、全ての端末に用意するだけの必要性はないため、購入しないことにした。CD-ROM とともかく、Floppy disk はディスクトラブルの際のブート用としてあれば大変助かるのだが、予算的にはともかく、使用を試みる学生により壊れることが予想されるので、これも無しで済ました。外付のブート用機器を接続する SCSI 等は無いため、ディスクトラブルの際は筐体を開けて作業する意外には手はない。ディスク交換の事態にならないよう、必ずブート可能なカーネルを用意するようカーネルのアップグレードの際は注意が肝心である

表 2: デスクトップのセキュリティー対策

ハードウエア	
CD-ROM	無し
Floppy disk	無し
筐体	特殊ネジにより固定
リセットボタン	無効に設定
電源ボタン	無効に設定
ソフトウエア	
BIOS	パスワードによりロック
	キーボード無しでもブート可能に設定
Boot loader	GRUB
	パスワードにより Multi-user mode 以外ではブート不可能にロック
CTRL-ALT-DEL	inittab で無効に設定

4 システム設定

PC へのシステムインストレーションは、一台の PC をシステム設定した後、そのディスクのデッドコピーを納入してもらった。NIS server, DNS, sendmail, ntp 設定はもちろん、automount の設定、tcpwrapper, pam の設定等のシステム管理情報は予め設定しておく。各種サーバーは役割毎に別名 (ns, ntp, pop, smtp, proxy, www, news, ftp...) を設定し、DNS において別名と本名とのマッピング (CNAME) 変更によりサーバーの変更が容易なよう計った。とは言っても、IP address とホストネームだけは各計算機毎に変更せねばならない。single user mode で立ち上げた後、あらかじめ作成しておいた shell script を実行し変更した。面倒なのはプリンター設定で、教室毎でデフォールトのプリンターは異なる。また各教室にも前中後 3 台のプリンターが設置してあるので、princap ファイルも全く同じと言う訳にはいかない。立ち上げた後ファイルを転送し個々に変更した。立ち上げ後システムの管理用ファイルの転送をどのように執り行うか予め定めておき、その為に必要な設定 (例えば、ssh の authorized_keys 等の設定) を忘れないことである。

また、一般に開放している計算機の場合、不必要に login したまま放置するユーザーも多い。また interactive に nice 値も変更せず長時間にわたりプログラムを実行するユーザーもいる。一定の時間入力データのないリモート接続 / デスクトップは、強制終了するプログラム (autolog) を使用している。更に重要なことの一つは、暴走したプログラムをどのように Kill するかである。netscape はその代表だが、CPU だけを消費している状態に陥った emacs もよく見掛ける。これは運用時の PC 管理の一つであるが、この手の house-keeping 管理は、個々の PC でせず管理サーバーの一つにおいて、cron で定時に shell script を起動し、ssh を用いリモートジョブ実行により house-keeping している。netscape は、デスクトップの使用者の終了時 (X sever の終了時) に、killall コマンドにより停止する方法をとっている。また、(デスクトップを使用できない) 深夜に xdm/gdm を毎日再起動することにより暴走プログラムを kill することに努めている。

5　ソフトウエア構成

システムは、パッケージ管理が気に入り, Debian GNU/linux [1] を使用している。第一義的には学生が使用するデスクトップなので、ほとんど全て (?) のソフトウエアをインストレーションした。

デスクトップは GNOME を使用し、Window manager は Enligthenment である。GNOME 及び Enligthenment に関係する種々のデフォールトは、もちろん予め設定しユーザー登録時に ホームディレクトリーの下にコピーするユーザー登録スクリプトを作成している。Gnome 設定では、各種ゲームプログラム、スクリーンセーバーは使用できないよう除去した。ゲームの利用はあながち無意味とは言い切れないが、研究用の PC クラスターでもあるので、CPU を無駄に使用されたくないからである。

Debian には、研究者には必須な LAPACK や FFT 等の数学ライブラリーも各種用意されている。また、PC クラスター上で並列計算するためのライブラリーも利用できる。我々のところでは、PVM, MPI の両方ともインストールしてある。LAPACK の PVM, MPI 版も Debian からである。クラスター上での並列計算は、この PC クラスターではいささか CPU が見劣りする。本格的な並列計算には、(ネットワーク接続は 100BaseTX ではあるが)、Dual alpha 21264A 750MHz の 18 台からなるクラスターを用意してある。この PC クラスターの最も容易な且つ有用な利用方法はバッチ処理である。

表 3: ソフトウエア構成

Linux	debian potato
Desktop	gnome/enlightenment
Browser	netscape
Graphics, DTP	gimp, tgif, tetex + jtex, gs, gv, acroread
Editors	emacs/xemacs, code-crusader, code-medic
Script languages	awk, perl, python, tcl/tk, scheme
Mathematics	R, octave, scilab
Mathematical libraries	BLAS/ATLAS, LAPACK, FFT
Libraries for parallel comp.	PVM, (BLACS, SCALAPACK)
	MPICH MPI (BLACS, SCALAPACK)
	LAM MPI (BLACS, SCALAPACK)
Queueing for cluster	queue, DQS
Temporary file space for better I/O	2 GB in local ATA-IDE disk
Secure ...	telnet-ssl, open ssh

PVM: Parallel Virtual Machine
MPI: Massage Passing Interface
Queue: Load balancing/distributed batch processing and local rsh replacement system
DQS: Distributed Queueing System

バッチ処理プログラムは各種あるが、フリーソフトとしては、DQS, Queue がある。私は、使用方法が簡単なため Queue [2] が気に入っている。各計算機の load average に基づき負荷の小さい計算機上でジョブを実行してくれる。script でジョブを投入する際に注意しないといけないのだが、load average に基づくジョブ実行なので、ジョブを投入後load average に反映されるまで時間を置かないと一台の計算機に多数のジョブが投入されることになるのが欠点である。Queue は、処理能力等を考慮した計算機間でのジョブ投入の優先順位、同時実行するジョブ数、 メモリー使用可能量、CPU 時間の上限等を設定できるので、種々の異なる計算機からなるクラスターでも使用できよう。一方、使用者が学ぶ必要があるのは、1 つのコマンド、数個のオプションの利用法だけである。

Queue を用いて、サーバーよりジョブを投入した例を下に示す。

```
ug% queue -q -d days -n -w -- SMB12L100M05M1MN80.out < /dev/null >& error.log &
ug% ps fxa
26012 ?        S      0:00 queue -q -d days -n -w -h ug -- SMB25L120M05M1MN100.o
26169 ?        S      0:00 queue -q -d days -n -w -- SMB12L100M05M1MN80.out
26191 ?        S      0:00 queue -q -d days -n -w -- SMB10L60M05M1MN50.out
26203 ?        S      0:00 queue -q -d days -n -w -- SMB9L60M05M1MN50.out
26206 ?        S      0:00 queue -q -d days -n -w -- SMB8L60M05M1MN50.out
26208 ?        S      0:00 queue -q -d days -n -w -- SMB7L60M05M1MN50.out
...
...
...
  913 ?        S      0:00 queue -q -d days -n -w -- sol_4_L60.out 3.514
  968 ?        S      0:00 queue -q -d days -n -w -- sol_4_L60.out 3.568
 1006 ?        S      0:00 queue -q -d days -n -w -- sol_4_L60.out 3.570
ug%
ug% kill -9 913
```

このPCクラスターでは、クラスターとしての使用を促進するため、各PCへは外部からloginできないよう制限し、通常のバックグランドでのジョブの実行はデスクトップ利用者以外には不可能なように管理している。では、投入したジョブをkill しなければいけない時はどう対処するのであろうか？ もし、killする必要があるときは、ジョブを投入したサーバーで単にそのジョブに対応するqueue をkillすればよい。

6 ネットワークセキュリテー

近年ネットワークセキュリテ - の重要性が増す中、rsh/rloginを使用せずssh/telnet-sslを用いている。また、学生のパスワード等は秘密保持不可能であるという前提で、PCから学内へのアクセスは、学外の計算機と同じ扱いをしている。つまり、学外からアクセスできないように制限している計算機には、PCからもアクセスできない。一方、PCから学外へはアクセスできるが、学外からはアクセスできないようにルーターで設定している。サーバーへの学外からのログインも学部学生は特に理由がある場合を除き禁止、大学院学生の場合は、国内の大学、政府研究機関のみ許可、教官の場合は国内外の大学、研究機関のみ可能なよう、tcpwrapperおよびPAMで設定してある。それ以外の組織から接続する必要があれば申請してもらうことになっている。

7 まとめ

教育用として使用する120台のPCをクラスター構成により研究システムとして共用するシステムを紹介した。Celeron 500MHzは最新のPentium IIIの半分以下のCPU能力と思われるが、クラスター全体のジョブ処理能力は $120/3 = 40$ 倍である。パラメーターを変えて多数回実行する必要のあるジョブの場合大変有用である。

参考文献

[1] http://www.debian.org/

[2] http://www.gnu.org/software/queue/queue.html

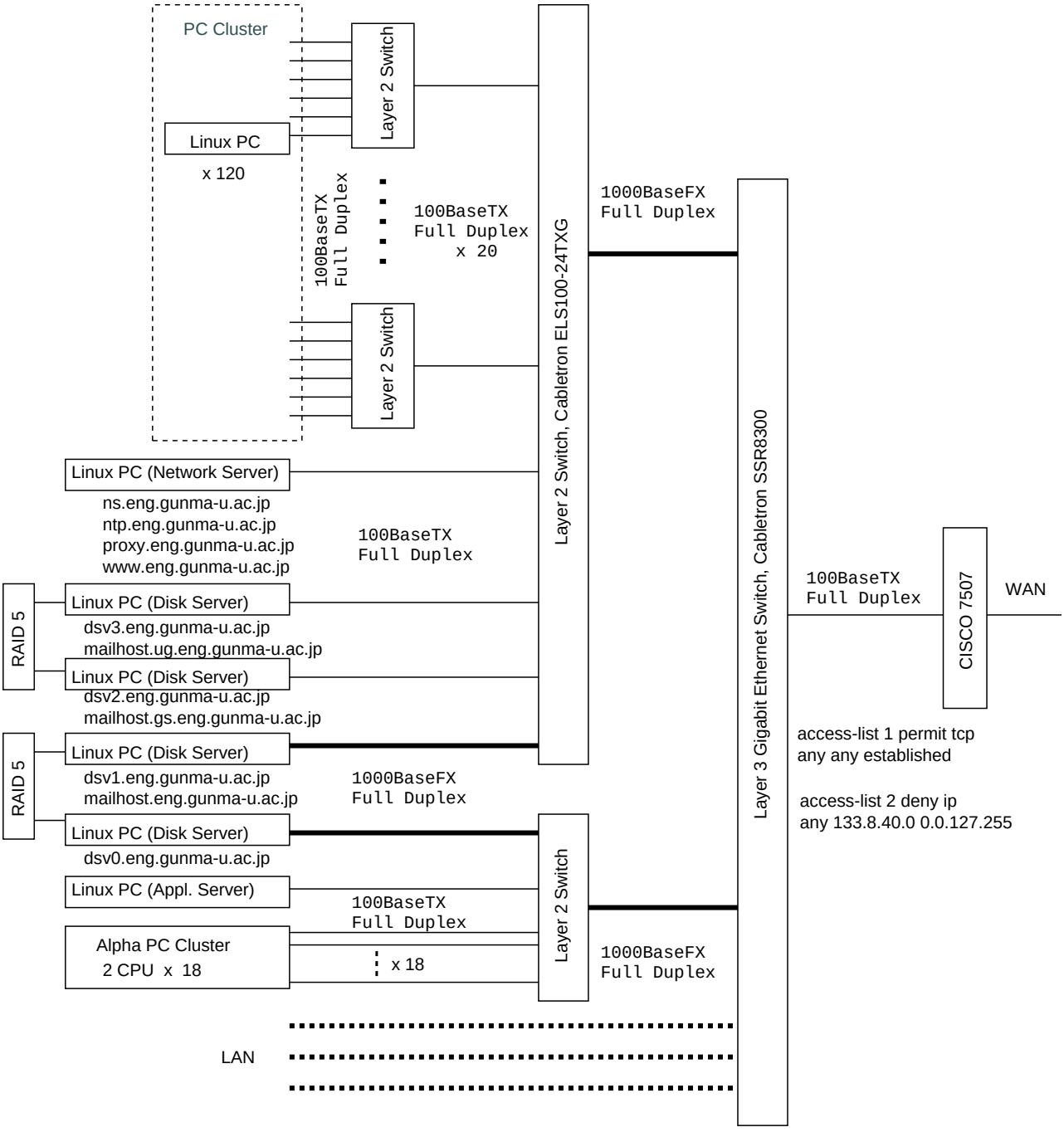

図 1: PC クラスター / サーバーからなるネットワーク構成図