2004－DSM－34　(12)
2004／7／30

教育用PC端末群のPCクラスタ的利用とその運用について

庄　司　文　由†

大学などの教育機関には、学生用の情報インフラとしてPC端末が多数設置されている。その一方で、これらPC端末は、近年のPCの急激な高性能化等を背景に、HPC(High Performance Computing)用の計算資源としての価値が高くなって来ている。本稿では、教育目的のために導入されている多数のPC端末の遊休時間を活用し、PCクラスタ等のHPC用の計算資源として利用するため環境構築について技術的考察を行う。また実証実験を通して得られた知見から、センターサービスとしての運用に向けた課題について検討する。

A use of PC terminals as PC cluster

Fumiyoshi Shoji†

The CPU power of PCs and network devicies have been developed rapidlly. According to current TOP500 list[1], the HPC systems classified PC cluster occupy 58.2% in TOP500 super computers. On the other hands, there are many PC terminals in universities as information infrastructure for students. These PC terminals can be identified as computational resoursies. In this article, we consider the construction of PC cluster environment built by PC terminals set up for educational purpose and investigate the issues for the use of such a PC clusters as HPC computational resourses.

1. はじめに

近年、PC端末の演算能力は数年前のスーパーコンピュータに匹敵するほど高性能化し、またコストも急激に下がって来ている。さらにネットワークについても、高性能化と低価格化が同時に進んでおり、特にここ数年で、ギガビットネットワーク機器の低価格化が顕著になっている。必然的に、PCクラスタ型のコンピュータが高い価格対性能比を示すことになり、最新のTOP500リスト (23rd edition)[1] では、上位500システムのうちの実に291システムを占めるにいたっている。ここ最近は、TOP500リストが改定されるたびにPCクラスタの占有率が増加しており、この傾向は今後ますます顕著になると予想される。

一方、大学などの教育機関には、学生用の情報インフラとしてPC端末が多数設置されている。これらの端末は単体で数ギガFLOPSという高い演算性能を持ち、さらに台数の規模 (広島大学では約600台) を考慮すれば、端末群全体として極めて巨大な計算資源となりうる。

本稿では、大学などの教育機関に設置されているPC端末をHPC用の計算資源としても利用できるような環境構築について議論する。また、小規模な実験環境を構築し、そこから得られた知見をもとに、センターサービスとしての運用に向けた課題について検討を行なう。

† 広島大学 情報メディア教育研究センター
Hiroshima University Information Media Center

2. 広島大学におけるPC端末環境の概要

広島大学では、教養科目の中の情報科目を実施するための教育設備として、また学生の自学自習用設備として、全学に約600台程度のPC端末が設置されており、これらは全て情報メディアセンターが管理している[2]。

PC端末群は、大まかにいって3ヵ所に分散しているものの、基本的に教室単位で集中して設置されていること、同時期にリプレースしたためほぼ同スペックであること、運用ルールが統一されていることなどが特徴である。運用については、600台のうち約200台は授業利用が優先され、授業がない時間は、残りの約400台と同様に、自習用端末として学生は自由に利用することができる。端末室は8:30から22:00まで開室され、22:00以降はシャットダウンする運用になっている。

さらに、今回のPC端末の遊休資源の有効活用の試みを進めるにあたっては、運用側から既存のサービスに影響が出ないよう最大限配慮するように強く求められている。したがって今回、検討を進めて行くにあたっては、以下のような方針で行なう。

- 既存サービスの提供時間 (8:30 〜 22:00) についての利用は検討対象から外す
- 既存の端末環境への影響を避けるため、OS 環境を別途用意する

3. 利用方法についての検討

一般的に、多数の PC 端末群の計算資源を活用する方法としては、Condor[3] 等を利用するデスクトップグリッド的利用と、SCore[4] 等を利用する PC クラスタ的利用、及びこれら 2 つを組み合わせる利用があると考えられる。

Condor はウィスコンシン大学が開発を行なっているジョブスケジューラである。Condor は、Condor プールとして複数の計算機を管理し、投入されたジョブの内容とプール内の各計算機の稼働状況に応じて適切な計算機にジョブ実行を割り当てることができる。さらにジョブが実行されている計算機の稼働状況の変化に応じて、ジョブを他の計算機にマイグレートさせる機能も併せ持っている。特徴としては、利用状況に応じて、時々刻々変動する PC 端末群の遊休資源を有効に活用しながら、多数の逐次型ジョブを効率良く処理することができる。

SCore は旧新情報処理開発機構 (RWCP) で開発された統合型のクラスタリングツールである☆。ノード間通信のための独自軽量プロトコルや、効率的なジョブスケジューリング機能、また、計算ノードの冗長化機能等を実装しており、運用面も含めて高性能で高機能な PC クラスタを構築することができるため、並列型ジョブを高速に実行できることが大きな特徴である。

これらのツールを教育用 PC 端末群に対して使う場合にどのようなメリット・デメリットがあるのか、以下で検討する。

Condor の特徴は、様々なアーキテクチャの計算機が混在し、しかもそれらが状況に応じて変動するような計算資源上での逐次型および並列型ジョブの効率的な実行にあり、今回のような夜間の時間帯を占有して利用するといったような、静的で均一な計算資源上においては、その特徴を充分に生かすことが難しい。また、Condor 上で並列型ジョブを実行する場合には、オーバーヘッド等の問題で、SCore で PC クラスタを構築した場合と比較して、実効性能で劣る可能性が高い。昼間の時間帯に利用するのであれば、Condor を使うメリットの方が大きいが、今回のような、端末環境が静的かつ均一である場合は SCore の特徴がより生かしやすくメリットが大きい判断し、以下では、SCore を中心として環境構築を行なっていくことにする。

ただし、どちらのツールを使っても、昼間と夜間で稼動 OS が切り替わることから来る夜間だけで終らないような長時間ジョブをどう扱うかという問題は、以前として残っている。

☆ 現在は PC クラスタコンソーシアムが開発を引き継いでいる

4. 既存サービスとの共存

現在広島大学で教育用端末として利用している PC の仕様は**表 1** の通りである。

表 1 教育用 PC 端末のスペック

CPU	Pentium4 2.0GHz
メモリ	256MB
ハードディスク	100GB
ネットワーク	100Base-TX
OS	VIne Linux 2.5 ベース

これらの PC 端末群を夜間の時間帯に PC クラスタ化するために、以下のような方法を採用した。まず、OS については、既存の OS 環境に修正を加えることについて、特に SCore は Linux カーネルにパッチを当てる必要があるため、既存の教育用端末としてのサービスに影響が出るおそれがあることを考慮し、PC クラスタとして運用する夜間については、別の Linux で稼動させることとした。ただし、そのためには、時間帯に応じて各端末の起動/停止が遠隔から制御でき、かつ起動時の OS を選択できること、その作業が自動的に行なえることが運用上必要となる。また、同様にクラスタ用の OS(Linux) をハードディスク内に別途インストールすることも、既存サービスへの影響を否定できないため、PC クラスタ用の OS はディスクレスブートで稼動させることとした。ディスクレスにすることで、各端末のハードディスク故障のリスクを回避できるほかに、ソフトウェアアップデートなどの管理業務のほとんどがサーバ上で行えるため、管理運用コストを軽減できるというメリットもある。

また、既存のハードウェアには可能な限り手を入れない方針ではあるが、PC クラスタの実効性能は、各端末間の通信性能が本質的であること、既存ネットワークへの影響を抑えること、等から、各端末にギガビット対応のネットワークインターフェイスを増設し、クラスタ用に別途ネットワークを構築することにした。

5. ディスクレスブートと OS の切り替えについて

前節における検討から、教育用端末として運用する昼間と、PC クラスタとして運用する夜間では稼動 OS が異なり、特に夜間についてはディスクレスブートさせる必要があることがわかった。このような運用を定常的に行なうためには、各端末の起動/停止制御と起動 OS の選択が遠隔から自動的に実行できなければならない。

遠隔からの起動/停止制御は、端末群を管理するサーバ上から、起動時に Wake ON LAN を利用し、停止時

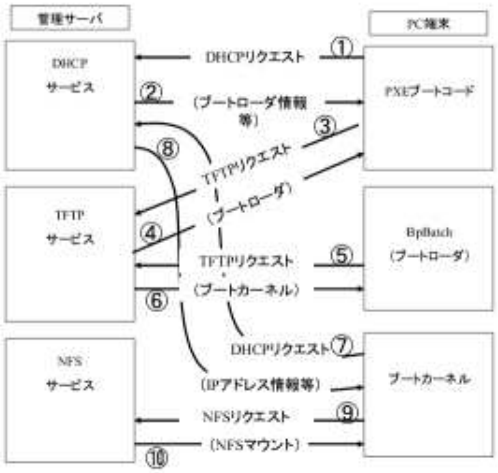

図 1 ディスレスの場合のブートシーケンス

にはリモートシェルを経由したシャットダウンを行うことで容易に実現できる。ただし、各端末のネットワークインターフェイスおよび BIOS が Wake On LAN に対応している必要がある。

次にディスクレスブートと起動 OS の選択については、いくつか方法があるが 、今回は、PXE、DHCP、TFTP、BpBatch[5] を組み合わせる方法を採用した。。PXE（Preboot eXecution Environment）は Intel が提唱するディスクレスブートのための規格である。ハードウェアの要件は、各端末のネットワークインターフェイスおよび BIOS が PXE に準拠していることである。

この方法を用いた場合の各端末のブートシーケンスは**図 1** のようになる。

各端末は、Wake On LAN 等何らかのトリガによりブートを開始する。BIOS の設定で PXE をブートデバイスの最上位にしておけば、ネットワークインターフェイスが PXE ブートコードを実行し、DHCP サーバの探索を開始する。DHCP サーバが見つかれば、ブートローダに関する情報を DHCP 経由で取得する。次に、その情報を元にブートローダを TFTP 経由でダウンロードし実行する。ブートローダは、ブートカーネルを TFTP でダウンロードし実行する。ブートカーネルは、NFS 上にある自身のルートボリュームをマウントし起動する。その後、DHCP で自分に割り振られた IP アドレスを取得し, ネットワークに接続する。ブートカーネルとして、あらかじめ NFS-ROOT と SCore に対応したものを用意しておく必要がある。

原則として、管理サーバ上には各端末のボリュームイメージを別々に保持する必要があるが、端末固有の情報☆は RAM ディスク上あるいは各端末のローカルディスク上に置くことで、単一のボリュームイメージを全ての端末で共有することができる。これにより、管理サーバ上のディスクスペースが大幅に節約できるとともに、ソフトウェアアップデート等の保守作業の効率が格段に向上する。

なお、OS の切り替えは、ブートローダの設定ファイルを変更するか、セカンダリのブートデバイスにハードディスクを設定しておくことで実現できる。後者の方法では、ブート時に DHCP サーバが見付からない場合、自動的にセカンダリブートデバイス (ハードディスク) からのブートに移行することを利用している。すなわち、各端末がブートする前にブートローダの設定ファイルを変更する (あるいは DHCP サーバを稼動/停止させる) ことで、端末で起動する OS を選択することが可能となる。

6. 実運用に向けた課題

前節で述べたように、時間帯によって各端末を別々の OS で起動させ、端末群全体として、まったく異なるサービスに活用することが可能となった。しかし、PC クラスタをセンターサービスとして充分な利便性を確保しながら定常的安定的に運用していくためには、解決しなければならない問題がまだいくつか残っている。

そのひとつは、PC クラスタとしての運用が夜間に限られているため、長時間のジョブが途中で打ち切られてしまう点である。このままではジョブを流す際に、利用者側に稼働時間を意識させる必要があり、特に初心者から見た場合、敷居が高くなってしまうおそれがある。また、クラスタとしての稼動時間が限られているため、利用者はプログラム開発やジョブの投入など全ての作業を夜間に行なわなければならない。

もう一つの問題は、クラスタを構成する各端末に故障が起きた場合の対処である。一般に PC クラスタは多数の端末群から構成されるため、単純に考えて全体の故障率は端末 1 台あたりの故障率の台数倍となる。数千台クラスのクラスタになると、確率的に常に数台に故障が発生してもおかしくない。その一方で、分散並列型のアプリケーションでは、クラスタを構成するすべての端末が稼動していることが前提となっていることが多く、途中一つでも障害が起きると、ジョブ全体が致命的な影響を受けてしまう。しかし、個々の端末の故障率を下げることは極めて困難で, コスト的にも割が合わない。むしろ、数台程度故障してもシステム全体しては影響を受けないような構成を指向すべきである。

上に挙げた 2 つの問題は、SCore のチェックポインティング機能とフェイルオーバー機能を利用することである程度解決できると考えている。SCore では、マルチユーザーモードの場合、ジョブが実行されたとき

☆ /var 等のログ情報、/etc の一部のファイルと SCore のジョブ情報のための /var/scored など

チェックポイント情報を各端末のローカルディスク上に保存させることができる。もし不測の事態が起きてジョブが異常終了したとしても、システムが再稼働する際に保存されているチェックポイント情報を読み出し、その続きからジョブを継続することができる☆。したがって、実際の運用では、夜間の運用を終了する際に、実行中のジョブのチェックポイントを採取しておき、再び PC クラスタが稼動する際に、続きから実行するようにしておけば良い。

プログラム開発やジョブ投入などがクラスタの稼働時間に制約を受ける点については、クラスタ環境のフロントエンドとして、24 時間稼動するサーバを設置することで解決できないか検討している。利用者はフロントエンドサーバでコンパイル等のプログラム開発、テストジョブ等の小規模ジョブの実行、および大規模ジョブのジョブの投入等を行なう。このうち、テストジョブや小規模ジョブはフロントエンドサーバ上で実行し、大規模ジョブについては、バックエンドの PC クラスタの稼働状況に応じてフロントエンドサーバが適宜スケジューリングする。こうすることで、利用者は PC クラスタが稼動していない時間帯でも最低限の作業を行なうことが可能となり、利便性が著しく損なわれる事はない。

PC クラスタを構成する各端末に何らかの障害が起きた場合については、SCore には、障害が起きた端末と、あらかじめ登録しておいたスペア端末を置き換えクラスタを自動的に再構成する機能がある。この機能とチェックポイント機能を組み合わせることで、仮に障害が起きたとしても実質的にジョブの実行には影響が出ないクラスタ環境が実現できる。

7. テスト環境の構築

これまでの検討を元に、小規模な構成で実証実験を行った。用いた端末は、長期休暇期間中のため運用停止している端末群（34 台）である。これらにギガビットネットワークインターフェイスと 512MB のメモリモジュールを増設し、既存ネットワークへの影響を最小限に抑えるため、ギガビットスイッチおよび管理サーバで独立なネットワークを構成した。

ネットワーク機器の仕様を**表 2** に、使用したソフトウェアを**表 3** に、構成図を**図 2** に示す。

管理サーバ上の dhcpd では、各端末の Mac アドレスと対応する IP アドレスを登録し、ディスクレスブート用のブートローダおよびルートボリュームの位置を指定しておく。以下に /etc/dhcpd.conf の一部を示す。

☆ バージョン 5.6 からシングルユーザーモードでもチェックポイント機能が使えるようになった

表 2 ネットワーク関連機器の仕様

	品名	規格
ギガビットインターフェイス	Intel PRO/1000MT	1000Base-T, PXE, Wake On LAN 対応
ギガビットスイッチ	Dell Power Connect 5224	1000Base-T× 24 port × 3
管理サーバ	ショップブランド CPU: Pentium4 2.4GHz x 2 メモリ: 2GBHD: 120GB	

表 3 ソフトウェアの仕様

名称	バージョン
ディストリビューション	Redhat7.3
カーネル	2.4.18 + SCore パッチ
SCore	5.4.0
DHCPD	2.0
ネットワークドライバ	e1000 (4.4.19)

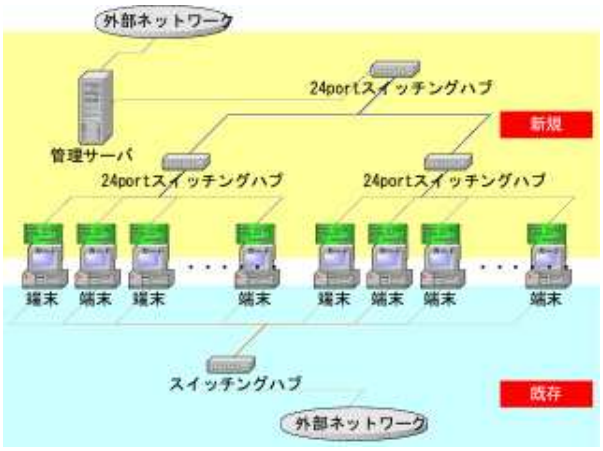

図 2 ネットワーク配線図

/etc/dhcpd.conf

```
........
filename "bpbatch";
option root-path "/work/clust_root";
option dhcp-class-identifier "PXEClient";
option vendor-encapsulated-options 01:04:00:00:00:00
option option-135 "bpbatch";

.......

host clust01.clust.net {
hardware ethernet XX:XX:XX:XX:XX:XX;
fixed-address 192.168.XX.XX;
}

......
```

BpBatch の設定ファイルには、起動用のカーネルイメージとディスクレスブートに関する指定をする。

以下に bpbatch の設定ファイルを示す。

bpbatch.bpb(クラスタ用)

```
set cachenever="ON"
linuxboot "vmlinuz" "root=/dev/nfs ip=dhcp"
```

各端末側では、BIOS のブートデバイスの設定で、PXE を 1st に、ハードディスクを 2nd にしておく。

PC クラスタとして起動させる場合は、管理サーバから各端末に Wake On LAN パケットを送れば、dhcpd と BpBatch の設定に従い、各端末は管理サーバ上のブートカーネルとルートボリュームから起動する。教育用端末として起動させる場合は、管理サーバ上の dhcpd を停止させておくか、BpBatch の設定を以下のようにしておいてから、各端末を起動すれば良い。

bpbatch.bpb(教育用端末用)

```
set cachenever="ON"
linuxboot "vmlinuz(教育用)" "root=/dev/XXX"
```

これまでのところ、

(1) ディスクレスブートによる、PC クラスタ端末としての起動
(2) Wake On LAN とリモートシェルによる遠隔からの起動/停止
(3) ブートローダ設定ファイルの変更および DHCP サーバの稼動/停止トリガとした起動 OS の切り替え
(4) SCore を用いた PC クラスタの構築と並列ジョブの実行

の動作をテスト環境の上で確認することができた。

特に実効性能については、preliminary ながら 32 台のクラスタ構成で、HPL Linpack[6] を用いた計測を行ない、理論ピーク性能の 41% という結果を得ている。

現在、チェックポイント機能を使ったジョブの継続と障害時のフェイルオーバー機能について評価中である。

8. ま と め

学内の教育用 PC 端末の遊休時間を利用して、PC クラスタを構築・運用するための方法を検討した。

遠隔から各端末の起動/停止を制御すること、起動時の OS を遠隔から選択できること、管理運用コストが低いこと、既存サービスへの影響を極力抑えること等々から、ディスクレスブートを利用した構成を採用した。ディスクレスブート化はハードウェアさえ対応していれば比較的容易に実現でき、しかも充分実用に耐えるレベルで動作することが改めて確認できた。また、端末固有の情報は、ネットワークに関する部分は DHCP に集約し、残りの部分については各端末のローカルディスクを利用することで, ディスクレス端末用のボリュームイメージを完全に単一化することができた。これにより、ディスク領域の大幅な節約とアプリケーションアップデート等の管理業務の簡略化を実現できた。

SCore を用いることで、導入コストを抑えながら高い耐障害性と通信性能を持つクラスタ環境を比較的容易に構築することができた。さらに、チェックポインティング機能を利用することで、夜間だけ運用するといったような変則的な運用ルールにも充分対応できることがわかった。

今後は、チェックポインティング機能、フェイルオーバー機能の評価を行ない、運用方針に沿った適切なジョブスケジューリングの方法を確立させたい。

また、今回のようにディスクレスで PC クラスタを構築する場合、端末側の OS を含めたソフトウェアを管理サーバに集約することができる。このことを利用すると、例えば、既存の PC 端末群に、必要なソフトウェアを組み込んだ管理サーバを接続するだけで、PC クラスタ環境を容易に構築することができる。さらにもう一歩進めて、PC クラスタ構築のための管理サーバ用の 1CD Linux ディストリビューション化することも可能と考えている。

今回の実証実験では、SCore に特化して評価を行なったが、逐次型ジョブの実行に限れば、Condor を利用した方がよりよい環境を構築できる可能性が高い。今後は並列型逐次型の両方のジョブが効率的に実行できる環境の構築を視野に入れながら、SCore と Condor の併用について検討を進めて行きたいと考えている。

参 考 文 献

1) http://www.top500.org
2) 庄司文由、長登康、隅谷孝洋、中村純、永井克彦: Linux による一般情報処理教育, 情報処理学会研究報告 99-CE-54, 17-23
3) http://www.cs.wisc.edu/condor
4) http://www.pccluster.org
5) http://www.bpbatch.org
6) http://www.netlib.org/benchmark/hpl