教育用端末の遊休時間におけるHPC資源化について

庄 司 文 由†

今日計算科学は、理論、実験に続くサイエンスの重要な分野としての地位を確立し、さらに今後はコンピュータの高性能化ともあいまって、その重要性はますます高くなると考えられる。しかし、そのために必要となる計算機リソースは、特に地方大学等においては質、量ともに不足しているのが現状で、研究上の競争力を維持していくためにも大規模かつ高性能な計算資源の確保が大きな課題となっている。

一方、多くの大学では、学生用の情報インフラとしてPC端末を多数設置しているが、これらのPC端末は、単体で数ギガフロップスという高い演算性能を有し、さらに端末群全体として見ると、数百ギガフロップスから数テラフロップスに及ぶ潜在的な計算資源とみなすことができる。

本稿では、主に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(High Performance Computing)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、広島大学に導入されたシステムの設計および構築について報告する。

PC terminals as HPC resources

FUMIYOSHI SHOJI†

The computational science has established as an important field following theory and the experiment, and it will be thought that the importance rises more and more in the future. However, the computer resource needed for that is insufficient both qualities and the amounts.

On the other hand, many PC terminals are set up as information infrastructure for the student at universities. These PC terminals can be considered to be a potential computer resources by applying Grid technology.

In this letter, we discuss the use PC terminals for education as computer resources for HPC such as PC clusters etc. And we report on the design and the construction of the system introduced into Hiroshima University.

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、高性能化および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。結果的に、PCクラスタ型のコンピュータが高い価格対性能比を示している。さらに近年はPCクラスタ等の分散並列環境の運用技術が成熟し、高い信頼性が得られるようになってきているため、ミッションクリティカルな研究等の現場でPCクラスタが活用されることが珍しくなくなってきている。

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700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。

教育用端末をPCクラスタ等の数値計算用の資源として利用する試みとして、京都産業大学の事例[1]が報告されている。この事例の場合、11教室に分散されている合計733台の教育用端末の遊休時間を利用し、並列計算を含む数値計算に活用している。ただ、計算用として使われているときに教育用端末としての利用が始まると、それまで実行中だったジョブが打ち切りになってしまう点、導入から6年以上経過しているため、演算性能や通信性能等が相対的に低下している点などが、機能的に不充分と思われる。その他にもいくつかの事例があるが、その多くは、実験的または一時的な試みであったり、特定のアプリケーションや利用者を想定した極めて限定的なものだった。これに対し我々は、定常的なセンターサービスを念頭に置き、ジョブの継続的な実行や障害対策等を考慮に入れたシステム設計の検討を続けている[2][3]。

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

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

環境構築について議論する。はじめに、教育用端末のハードウェア・ソフトウェア構成および運用時間等の考察から、HPC資源として利用する際の利用方法を、効率性等の観点から検討する。次に、教育用端末環境およびHPC環境を共存させながら運用して行くための技術的課題を議論する。特にミドルウェアの検討およびチェックポイント/リスタート機能の活用について触れる。最後に、広島大学で平成17年5月より運用中のPCクラスタ/グリッド☆環境の詳細を紹介し、運用上の課題について議論する。

2. 教育用端末の運用状況の考察 (広島大学の場合)

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700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教育研究センターが管理している。

PC端末群は、大まかにいって3ヵ所に分散しているが、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すべてのPC端末が同スペックであること、端末室ごとにばらつきはあるものの運用時間がほぼ統一されていることなどが特徴である。運用については、約700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500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。各端末室毎の台数および開室時間を表1に示す。

表1 各端末室の端末台数および開室時間

端末室名	端末台数	開室時間
西図書館 3F	235	8:30 – 21:00(土曜 17:00)
西図書館 2F	100	8:30 – 21:00(土曜 17:00)
メディアセンター本館 2F 演習室	93	8:30 – 22:00
メディアセンター本館 2F オープンスペース	41	8:30 – 22:00
中央図書館	20	8:30 – 21:00
霞端末室	149	8:30 – 22:00
東千田端末室	65	8:30 – 21:00(土曜 18:30)

これらの教育用端末群を計算資源として利用するという観点から見ると、ある程度の規模の端末群が集約されていることから、PCクラスタのような分散並列環境を構築した場合に、要素間の結合を密に取ることが可能であることがわかる。次にほとんどの端末の運用時間がおおよそ朝8時から夜10時に集中しているため、HPC環境として利用する場合は、教育用として稼動しない夜間を活用する方法が最も効率的と考えられる。また、運用時間を明確に区分けすることで、教育用、HPC用それぞれの運用が独立することになり、結果としてお互いに影響しあうことを原理的に回避できるため、システムエラーおよびヒューマンエラーのリスク軽減が期待できる。

以上のような教育用端末環境の特性は、台数や運用時間に多少の差はあっても、基本的には多くの機関で共通していると考えられる。

3. HPC資源としての活用方法

前節での教育用端末の特性についての考察をもとに、HPC環境としての活用方法を検討する。HPC環境として利用する際には、MPIを利用した並列型アプリケーション、および要素間通信を行なわない逐次型アプリケーションの2タイプの利用を想定する。

3.1 OS環境について

OSについては、教育用端末環境とは別のOSで運用することが望ましいことは明らかである。HPC環境として利用する場合は、Linuxでの運用が想定されるが、教育用端末の場合はWindows等の運用もありうる。仮に教育用とHPC用とでOS環境を共通化できたとしても、アップデートやアプリケーション追加等のメンテナンスの際にトラブルが起きる可能性が極めて高い。運用上の責任を明確にする意味でも別々のOSで運用することが望ましいと考える。

したがって導入仕様では、教育用の場合は、LinuxとWindowsのデュアルブートとし、電源ボタンの押下およびOS環境の選択を学生自身が行なう運用とした。つまり、学生が使うときだけ端末の電源が入っていることになり、それ以外の時間は電源断の状態となる。HPC用の場合は、後述するSCoreのパッチを当てたカーネルで動作する専用のLinuxで稼動させる。また、HPC用環境の運用開始時に端末全台を一斉起動し、運用終了時に一斉シャットダウンする。つまり平日の場合、OS環境の切り替えが1日に2回行なわれることになる。合計3つのOS環境を端末室毎に異なるスケジュールで切り替えるという複雑な運用を実現するために、ミントウェーブ社のVIDシステムを採用した[4]。このシステムを利用することで、任意の端末を、任意の時間、任意のOSで起動することができ、さらに端末全体を統括するスケジューラを併用することで、端末室毎の端末管理を容易に行なうことが可能となった。また、ミントウェーブ社のVIDシステムはディスクレスであるため、障害対応等の運用コストが大幅に軽減できる利点もある。ただし、今回のシステムでは各端末に40GBのハードディスクを実装し、教育用HPC用それぞれでワーク領域として利用

☆ 本稿では、PCクラスタをMPI等の並列計算のための環境、PCグリッドを逐次計算のための環境という意味で用いる。

することで、ファイル I/O スピードが本質的になるアプリケーションで利用できるようにした。

3.2 ネットワーク構成

前節で触れたように、教育用端末は端末室毎に集約されているため、端末室内の要素間通信については問題が少ないと考えられるが、端末室間の通信については充分な帯域を確保するのは一般には困難である。そのため、並列型アプリケーション用の環境、つまり PC クラスタ的な利用をする際には、基本的に端末室毎もしくはそれ以下の構成とすることが望ましい。要素間のインターコネクトについては、ギガビットイーサネット、Myrinet、InfiniBand 等の選択肢があるが、価格性能比から見て、ギガビットイーサネットが現時点では最適と思われる。教育用端末に多い 32 ビット PCI バスの場合、Myrinet、InfiniBand ではバス幅が不足するため、充分な性能が得られない可能性が高いことと、今回は並列ジョブの規模を最大でも 100 並列程度とするため、ギガビットイーサネットでもある程度の性能が期待できること等が理由である。

今回の場合、端末群全体で合計すると約 700 台程度になるが、この中には、遠隔キャンパスに設置されているものや、管理の手が届きにくい場所に設置されているものもある。今後長期間に渡り安定的な運用を維持していくためには、メンテナンスがしやすいことが求められるため、これらの端末は HPC 用に利用する場合には除外することにした。結果的に、HPC 用として利用する端末は、メディアセンター本館設置分 134 台 (2 教室) とメディアセンター西分室設置分 335 台 (2 教室) となる。

また、各端末のギガビットイーサネットを教育用と HPC 用で共用することとし、24 ポートのギガビット L2 スイッチで結合することにした。また PC クラスタとして利用する部分では、各ノード間の通信帯域の確保および遅延時間の低減を考慮してスイッチ間のカスケード接続は 2 段までとし、またスイッチ間の接続はポートトランキングを用いて可能な限り広い帯域を確保した。結果的に、PC クラスタの構成として見た場合、実効 96 台が 2 セット、実効 64 台が 2 セットとなった。それぞれの PC クラスタでは、後述の SCore の冗長機能を利用するためにそれぞれ 4 台の冗長ノードを含んでいる。PC クラスタに含まれない残りの端末 (PC クラスタの冗長ノードを含めて 149 台) は PC グリッドで利用する。

図 1 は、メディアセンター本館の 2 教室についての配線図の抜粋である。演習室の 93 台のうち、PC クラスタとして利用するのが 68 台 (点線で囲まれた部分) で、この部分については L2 スイッチ A のレベルで、他の部分と物理的に閉じた配線となっている。また、この 68 台については L2 スイッチ B から L2 スイッチ A へのアップリンクを 5 ポート分充てることで、L2 スイッチ A を経由するような要素間通信が発

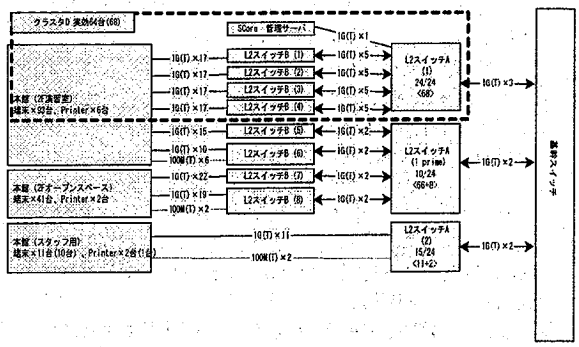

図 1 配線図の抜粋

生してもある程度の通信帯域を確保できるようになっている。

3.3 フロントエンド

前節で、HPC 環境として利用する際は、教育用として稼動しない時間を利用することとしたが、運用時間を区分けすることによるサービスへの影響は少なくない。教育用としては主に昼間の運用となるため、HPC 環境としての運用は必然的に夜間になる。とすると HPC 環境の利用者は夜間まで待たないと使えないということになり、利便性という意味で大きく後退する。理想的には、プログラムの開発や小規模なテストジョブの実行、大規模なジョブのバッチ投入が常時可能であることが望ましい。システム設計の際にはこの点に留意する必要がある。

このため、34 ノードからなる小規模な PC クラスタ (以下では、グリッドサーバと呼ぶ) を HPC 専用の計算資源として運用することにした。利用者は、この上でプログラム開発、小規模ジョブの実行、および大規模ジョブのバッチ投入等フロントエンド的な機能が 24 時間常時利用でき、運用時間の影響は大規模ジョブの実行だけに限られる。グリッドサーバは、並列計算用に 17 ノード (冗長ノード 1)、Gaussian や Nastran 等の ISV アプリケーション用に 17 ノードという形で運用している。

3.4 ミドルウェア

本システムでは、並列ジョブ用の環境として SCore[5] を、逐次ジョブ用の環境として Condor[6] を採用した。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、計算ノードの冗長機能、チェックポイント/リスタート機能等を実装しており、センターサービスのような定常的な PC クラスタの運用にも充分な機能を備えている。

Condor はウィスコンシン大学が開発を行なっている High Throughput Computing を指向したジョブスケジューラである。Condor は、複数の計算機を Condor プールとして管理し、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ため、ユーザーは個々の計算機の状況を知る必要がない。さらにジョブが実行されている計算機の負荷状況の変化に応じて、実行中のジョブを他の計算機にマイグレート (乗り移り) させる機能も併せ持っている。

今回は、SCore と Condor を併用することとし、PC クラスタとして利用する部分 (16 台 1 セット、96 台 2 セット、64 台 2 セット) については、SCore を使用し、Condor では端末群全体 (503 台) をカバーするように設計した。結果的に、逐次ジョブは基本的に PC クラスタ以外の端末で実行されるが、負荷状況によっては PC クラスタに含まれている端末でも実行できることになり、端末群全体の稼働率向上に寄与すると考えている。さらに、各端末室には、SCore 管理サーバおよび Condor のチェックポイントサーバとして、常時稼動のサーバを配置している。PC クラスタの構成を表 2 に示す。PC クラスタに含まれない端末は Condor を通じて逐次ジョブで利用される。

表 2 PC クラスタのグルーピング

クラスタ名	端末室名	ノード数 (冗長ノード)	稼働時間
クラスタ 1	メディアセンター本館 2F 演習室	68(4)	夜間
クラスタ 2	メディアセンター本館サーバ室	17(1)	常時
クラスタ 3	西図書館 2F	100(4)	夜間
クラスタ 4	西図書館 3F	100(4)	夜間
クラスタ 5	西図書館 3F	68(4)	夜間
クラスタ 6(ISV ジョブ用)	メディアセンター本館サーバ室	17	常時

3.5 ジョブの継続

本システムは、教育用端末として稼動しない時間を利用するため、HPC 環境としての運用が平日の場合最長でも 8 時間程度に制約される。そのため、この時間を越えてジョブを実行することができないことになり、利便性の点で大きな問題であると言わざるを得ない。ここでは、ミドルウェアのチェックポイント/リスタート機能を利用して、HPC 環境の稼働時間を越えたジョブの継続実行について検討する。

逐次ジョブの場合は、Condor のチェックポイント/ジョブマイグレート機能を利用することで、部分的に回避できる。例えば、夜間教育用端末の上で実行されているジョブは、HPC 用から教育用への運用切り替えの際に強制終了されるものの、Condor の機能によって、常時稼動の端末 (グリッドサーバ) 上でチェックポイントから継続実行される。ただし、Condor のチェックポイント/ジョブマイグレート機能を利用するには、使用できるコンパイラが限られる等の制約がある。

一方、並列ジョブの場合は SCore のチェックポイント/リスタート機能を利用する。このため、SCore とマルチユーザーモードと監視ツールの sc_watch を組み合わせて利用する。SCore のチェックポイント/リスタート機能は、通常次のように動作する。

(1) sc_watch の監視のもと、SCore の実体である scored が動作している
(2) 端末に障害が起きると scored はハングアップする
(3) sc_watch は scored のハングアップを検知し、あらかじめ登録されている冗長ノードと障害ノードを置き換える
(4) sc_watch が scored を再起動する
(5) 実行中だったジョブは、直近のチェックポイントから再実行される

この手続きを一部変更することで、ジョブの継続実行が可能になる。ここでは sc_watch が scored を再起動する際に、任意のシェルスクリプトを実行できることを利用する。

(1) sc_watch の監視のもと、scored が動作している
(2) 端末の運用切替時に scored はハングアップする
(3) sc_watch は scored のハングアップを検知し、次の運用開始時刻まで sleep する
(4) (運用開始時刻になったら)sc_watch が scored を再起動する
(5) 実行中だったジョブは、直近のチェックポイントから再実行される

SCore のマルチユーザーモードでは、実行中のジョブに対して管理者権限でチェックポイントを取得することができるため、実際の運用では、HPC 用としての運用時間が終了する直前に、管理者側でチェックポイントを取得することで、利用者は特に意識することなくジョブの継続が実現できる。ただし、SCore のチェックポイント/リスタート機能を利用するには、アプリケーション側にいくつかの制約がある。

また、昼間稼働停止している間は、上述のように scored がハングアップするため、scored によるジョブのキューイングおよびスケジューリングができない、そのため、一旦 Generic NQS[7] でジョブを受け付け、クラスタが稼働中の場合はそのままスケジューリングし、停止中の場合は稼働開始時間まで Generic NQS で保留状態にして、稼働後にスケジューリングする方法を採っている。

3.6 障害対策

逐次ジョブについては、既に述べたように Condor のチェックポイント/ジョブマイグレート機能で対応できる。Condor は各端末の動作状況を一定の間隔で把握しているため、障害があれば即座に検知することができる。障害が起きたときは、上述のチェックポイント/ジョブマイグレート機能を利用することで、自動的にジョブは継続される。

並列ジョブについては、前節で述べた SCore の通常の冗長機能を利用すればよい。ただし、ジョブの継

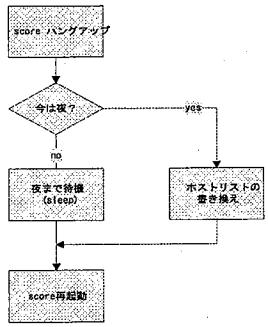

図 2　スクリプトの動作

続対策も併せて行なうため、実際には以下のような手順となる。動作の概念図を図 2 に示す。

(1) sc_watch の監視のもと、scored が動作している
(2) 端末障害時または運用切替時に scored はハングアップする
(3) sc_watch は scored のハングアップを検知し、以下のスクリプトを実行する
  (a) ハングアップが運用切替時刻で起きた場合 (運用切替によるハングアップ):
    → 次の運用開始時刻まで sleep する
  (b) ハングアップが運用時間中に起きた場合 (端末障害によるハングアップ):
    → 障害ノードを冗長ノードに置き換える
(4) sc_watch が scored を再起動する
(5) 実行中だったジョブは、直近のチェックポイントから再実行される

つまり、ハングアップが起きた時刻から、どちらのハングアップか特定し、その後の動作を変えるスクリプトを用意すればよい。ただし、端末障害の場合は、運用切替の場合と違い、管理者側でチェックポイントを取得できないので、利用者自身が実行時にチェックポイント取得のオプションを指定している必要がある。

以上の対策により、計算資源の大部分が夜間に稼動するという特殊な運用であっても、アプリケーションレベルで見た場合の影響を最小限にすることができた。

3.7 ジョブ情報の取得

並列ジョブについては、Generic NQS でキューイングされるため、qstat 等のコマンドでジョブの状況を確認できるが、これだけでは各ジョブが使用しているノード数等の詳しい情報が得られないため、SCore 付属の sctop コマンドを利用している。ただ、sctop は scored がハングアップしている間は使えないため、運用を終了する直前の sctop の情報をファイルにダンプしておき、ユーザーが閲覧できるようにしている。逐次ジョブについては、Condor 付属の condor_q や condor_status 等で確認できる。

4. 運用上の問題点

本システムは、平成 17 年 5 月末から部分的に稼動し、10 月から全面的に稼動する予定である☆。現在までの運用の中で判明した問題点を以下に挙げる。

4.1 並列ジョブの I/O

多くの並列ジョブでは、ファイル I/O の際に、各ノードのローカルディスクを活用することでスループットを確保する方法がしばしば用いられるが、本システムで採用している SCore のマルチユーザーモードの場合、実際にジョブがスケジューリングされないとどのノードが使われるかわからないため、事前に各ノードにファイルをコピーしておくことができない。一方、ファイル I/O に NFS ボリュームを使うと、ジョブで使用するノード数が増えるにつれて NFS サーバへの負荷が高くなり、結果的に I/O の速度が低下してしまう。

解決策としては、PVFS や GFS 等のクラスタファイルシステムを導入し、各ノードのローカルディスクを仮想的にひとつのボリュームとして利用する方法が考えられる。しかし、ファイル I/O の負荷が高くなった場合に、クラスタファイルシステム自体のタスクと本体のジョブの間で資源の競合が起き、演算性能が劣化するおそれがある。また、ハードディスクが故障した場合などの障害対応やデータを保護するための冗長性等にも不安が残る。

4.2 ジョブの継続

逐次ジョブ、並列ジョブともにチェックポイント / リスタート機能が利用できるが、実際にはうまく機能していないケースが多い。利用者に詳しく聞いてみると、あるケースでは、SCore の制限に含まれている永続的なファイルオープンが問題であることがわかってきた。詳細については現在調査中である。

4.3 逐次ジョブと並列ジョブの資源配分

現状は、教育用端末とグリッドサーバ合計 503 台のうち、逐次ジョブ専用が 150 台、並列ジョブ優先が 353 台という資源配分になっている。並列ジョブ優先の PC では、負荷状況に応じて逐次ジョブも実行可能であるが、スケジューラの構成上、並列ジョブが逐次ジョブより常に優先されてしまうため、今後配分の見直しが必要になるかも知れない。逐次ジョブと並列ジョブを同じようにスケジューリングでき、両者の資源配分を管理者がコントロールできるようなスケジューラの開発が待たれるところである。

4.4 未起動端末と実行時の障害発生

今回のシステムでは、昼間の教育用端末の運用から夜間の HPC 用の運用に移行する際に、端末の一斉起

☆ 平成 17 年 9 月末の時点で定常運用しているのは、グリッドサーバ (常時稼動) クラスタ 1、クラスタ 4、クラスタ 5(主に夜間稼動) である。

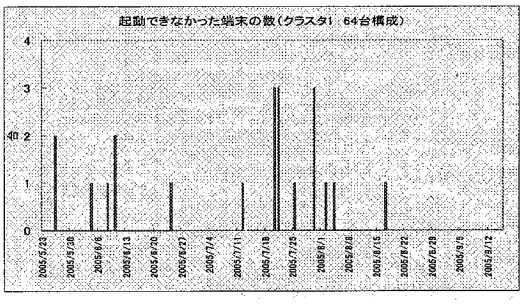

図 3　未起動端末の数と頻度

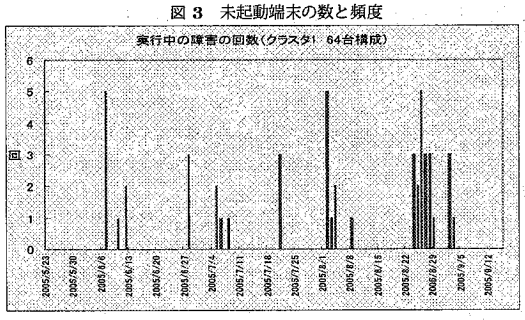

図 4　実行時に発生した障害の頻度

動が発生する。一般的に大量の端末を一斉起動する場合は、さまざまな要因から端末が起動しないケースがある。未起動の端末が冗長ノードの数を越えるとPCクラスタとしての運用ができなくなってしまうため、未起動端末の頻度を把握することは運用上重要である。図3は稼動を開始した平成17年5月末から9月末までの、クラスタ1についての未起動端末の数である。

図からわかるように、端末の未起動が発生する頻度は、安定的なサービスを提供する上では無視できる範囲であると言える。また、未起動が発生した場合でも冗長ノードの数を越えるようなケースはいまのところ起きていない。しかし、ジョブの継続が保証されなくなる未起動端末が2台以上のケースが、4ヵ月で5回程度発生しているので、起動の精度をもっと上げる必要があるかも知れない。

次に、クラスタ1の稼働中の障害発生の頻度を図4に示す。図からは、同じ日に何度も障害が発生する傾向が読み取れるが、よく調べてみると、特定のユーザーあるいは特定のアプリケーションを実行した際に、高い確率で障害が発生していることがわかった。現在は同様の障害が起きないように対策を取っているので、障害の頻度は今後低くなっていくと期待している。

5. ま と め

教育用端末の遊休時間を利用したHPC環境の構築について検討し、システム構築を行なった。特に留意した点は以下の3点である。

- 運用時間を越えたジョブの継続実行
  HPC環境としての運用時間が主に夜間に限られることで、長時間ジョブの実行が原理的に不可能になるが、SCoreおよびCondorのチェックポイント/リスタート機能を活用することで、利用者が特に意識せずとも自動的にジョブが継続される環境を構築した
- 端末障害への対応
  PC端末は設計上、長時間に渡って高い負荷がかかるHPC的な利用を想定していないため、稼動中に障害が起きる割合は相対的に高くなると予想される。したがって、あらかじめある程度の障害が起きることを想定し、仮に障害が起きたとしても可能な限りサービスに影響しないような設計が求められる。今回構築したシステムでは、SCoreおよびCondorのチェックポイント/リスタート機能および冗長機能を活用することで、高い耐障害性をもつシステムを構築した。
- 高い柔軟性
  センター等で提供される計算資源は、できるだけ多くのアプリケーションで利用できることが望ましい。そのために、並列ジョブ用にはSCoreを、逐次用にはCondorを採用し、どちらのジョブも効率よく実行できる環境を構築した。特にCondorのジョブマイグレート機能を活用することで、並列ジョブ用の資源にも負荷状況に応じて逐次ジョブが実行できることになり、端末群全体として負荷が均等化されると期待できる。

このように、いままでほとんど活用されてこなかった遊休時間中の教育用端末を、センターが提供する計算資源として充分に活用できることがわかった。また、遊休資源の集約化と仮想化によって、運用時間が不規則である問題を、ある程度克服できたと考えている。さらに今回のシステム設計では、特別に開発したものは含まれておらず、ほとんどのコンポーネントはフリーソフトであるため、多数のPC端末を抱えている多くの大学等の教育機関、さらには企業等において応用が可能であると考えている。

参 考 文 献

1) http://www.kyoto-su.ac.jp/kurozumi/cgi-bin/index-j.html
2) 庄司文由: 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, 情報処理学会研究報告 2004-DSM-34, 65–69
3) 庄司文由、隅谷孝洋、石井光雄: 教育端末の遊休時間を利用したHPC環境, 学術情報処理研究 No.9 2005, 27–36
4) http://www.mintwave.co.jp/tc/vid.html
5) http://www.pccluster.org
6) http://www.cs.wisc.edu/condor
7) http://www.gnqs.org/