

PC 演習室環境と共生する PC クラスタの構成法とその評価

大柚 智[†], 梶田 秀夫[‡]

s-ohyu07@dsm.cis.kit.ac.jp, h-masuda@kit.ac.jp

[†] 京都工芸繊維大学大学院 工芸科学研究科 情報工学専攻

[‡] 京都工芸繊維大学 情報科学センター

概要 近年、パソコンの普及率が高まり、PC 演習室を設置する教育機関が増えている。通常、PC 演習室には多数のパソコンが備え付けられているが、夜間・休日は警備の問題等で PC 演習室を閉室しており、備え付けのパソコンが遊休状態にある場合が多い。また、平日昼間でも演習室に備え付けられているパソコンの全てが利用されているわけではないため、遊休状態となる時間(遊休時間)が存在する。PC 演習室にある夜間・休日の遊休時間を有効利用するため、パソコンを夜間・休日に PC クラスタとして運用するシステムがあるが、平日昼間の遊休時間を考慮していないため、PC 演習室の稼働率を上げる余地が残されている。そこで、本稿では PC 演習室にあるすべての遊休時間を有効利用することを目指したシステムとして、仮想計算機技術を用いて、PC 演習室のパソコンを常時 PC クラスタとして運用することを提案した。そして、実験的に研究室内の常用 PC の 4 台を用いて PC クラスタを構成してその性能を評価し、仮想計算機技術を使わない通常の PC クラスタに比べて約 85%の性能を持つことがわかった。

キーワード PC クラスタ, 仮想計算機, SCORE

Implementation and Evaluation of shared PC cluster in the PC classroom environment

Satoshi Ohyu¹ and Hideo Masuda².

s-ohyu07@dsm.cis.kit.ac.jp, h-masuda@kit.ac.jp

¹ Graduate School of Information Science, Kyoto Institute of Technology

² Center for Information Science, Kyoto Institute of Technology

Abstract Recently, many universities or colleges have PC classrooms for ICT education. These systems often have many PC but these PCs leave inactive during night or holiday due to physical security issues. Moreover, some PCs still leave inactive in spite of business hour. We want to use all PC's computational power as much as possible.

In this paper, we propose shared PC Cluster in the PC classroom environment. Because we want to operate both educational use and cluster use, our system implements with Virtual Machine technology. We implement cluster node as guest OS, therefore cluster system is difficult to attack the educational use (as host OS). Our system shows a performance of about 85% computational power compared with the occupation type cluster by N-queen benchmark.

Keywords PC Cluster, Virtual Machine, SCORE

1 はじめに

通常、演習室には多数のパソコンが備え付けられているが、夜間・休日は警備の問題等で演習室を閉室している場合が多い。そのため、夜間・休日はシステムの稼働率が低くなり、計算機資源を有効活用できていないと、言い難い。そこでシステムの稼働率を上げるために、演習室に備え付けられているパソコンをクラスタノードとする PC クラスタを構築し、その PC クラスタで High Performance Computing を行うことによる計算機資源の活用が考えられている。

演習室に備え付けられているパソコンをクラスタノードとして運用している教育機関の例としては、広島大学 [1] や大阪工業大学 [2] などが挙げられる。これらの教育機関が採用しているのは、講義や演習が行われない夜間・休日に、パソコンをクラスタノードとして運用するシステムである。しかし、講義や演習が行われている平日昼間でも、全てのパソコンが使用されているわけではないため、平日昼間の遊休時間にもパソコンをクラスタノードとして運用できれば、計算資源がより有効に活用できるようになると考えられる。

そこで本稿では、仮想計算機技術を用い、ホスト OS を演習用 OS として、ゲスト OS をクラスタノード用 OS とするクラスタノードを考え、演習室にある PC を全ての遊休時間にクラスタノードとして利用することを提案する。

2 要件

- クラスタ利用が演習利用の妨げとならないこと。

演習室が開放されている平日昼間にも PC クラスタを稼働させる場合、クラスタノードとしての運用が行われているパソコンに、講義や演習目的の利用者がログインしてくることを想定する必要がある。そして、その場合は演習室としての機能を優先しなければならないため、講義や演習目的の利用者に対して CPU やメモリを優先して割り当てなければならない。また、機能の切り替え作業に時間がかかると、講義や演習目的の利用者に無用のストレスを与えてしまうため、できる限り短い時間で切り替え作業を完了させることが求められる。

- セキュリティやメンテナンス上の問題が無いこと。

機能の切り替えにかかる時間を実質的に無くするには、演習用 OS 及びクラスタノード OS に同一の OS イメージを用いれば良いが、この方法は、メンテナンスやセキュリティの都合上採用するのが難しい。また、メンテナンスやセキュリティ上の問題が無い方法としては、演習用 OS 及びクラスタノード OS に別々の OS イメージを用いたデュアルブートシステムがある。しかしこのシステムは、機能の切り替えの際にシステムの再起動による時間がかかってしまうため採用が難しい。従って、演習用 OS とクラスタノード OS が分離して並行稼働することが求められる。

3 クラスタノードの設計

要件を満たすため、仮想計算機システムを用いて演習用の OS 環境の上でクラスタノード OS を動作させることを考える。

3.1 仮想計算機技術

仮想計算機技術とは、ハードウェアの持つ機能をソフトウェアによって仮想的に実現する技術で、その例としてはフル・パーチャライゼーションとパラ・パーチャライゼーション、カーネルのアプリケーション化等がある [3]。

3.1.1 フル・パーチャライゼーション

フル・パーチャライゼーションとは、汎用の OS 上で計算機ハードウェア自体をエミュレートし、通常の OS を稼働させようとするアプローチである。(VMware[4], VirtualPC[5], QEMU[6] 等)

このアプローチの場合は、ゲスト OS にほとんど手を入れる必要が無いため、大抵の OS をゲスト OS として動かせるが、ネットワークやディスク等のデバイスに対するアクセスの際には、比較的大きなオーバーヘッドがかかってしまう。

3.1.2 パラ・パーチャライゼーション

パラ・パーチャライゼーションとは、仮想計算機を実現する専用のモニタ (VMM: Virtual Machine

Monitor) を持ち、OS のカーネルにもこのモニタに対応した形での実装を行うアプローチである。(Xen[7] 等)

このアプローチの場合は、他の 2 つのアプローチと比べてネットワークやディスク等のデバイスに対するアクセス速度が良好であるが、ホスト OS とゲスト OS のカーネルの両方に手を入れる必要がある。

3.1.3 カーネルのアプリケーション化

カーネルのアプリケーション化とは、ユーザプロセスとして、OS のカーネルを実行させるようにするアプローチである。(coLinux[8], User Mode Linux[9] 等)

このアプローチの場合は、ゲスト OS に手を入れる必要がある。

3.2 仮想計算機の実装

一般に演習室の環境は、できるだけ標準的な構成であることが望ましいため、ホスト OS に手を入れる必要があるパラ・バーチャライゼーションを採用することは困難である。また、VMware や VirtualPC は、プロプライエタリであるため、PC Cluster のノード処理のために細かい制御をしたくなつた場合に手を入れることができない。QEMU はオープンソースであるが、CPU レベルからの仮想化であるため、オーバーヘッドが比較的大きい。よって、カーネルのアプリケーション化で仮想化された OS イメージをクラスタノード OS として用いることを選択する。

3.3 制御モジュール

演習利用を目的とするユーザがログインしてきた場合、演習目的の利用に対して優先的にリソースを割り当てる必要がある。リソース割り当てを制御する手法として、演習用 OS へのログイン状況を監視し、状況に応じてクラスタノード OS へのリソース割り当て制御を行うモジュールを、演習用 OS に組み込むことを考える。制御モジュールに必要とされる機能は以下の 2 点である。

1. 演習目的のユーザの把握。
2. クラスタノード OS のリソース制御。

3.3.1 制御モジュールの設計

一般に演習室環境では遠隔からのログインを禁止しているため、コンソールユーザのみを考えれば良い。また、本稿ではクラスタノード OS としてカーネルのアプリケーション化を用いているため、クラスタノード OS のカーネルがホスト OS 側からはプロセスとして見えるようになっている。従って、クラスタノード OS のリソース割り当て制御には、通常のプロセスに対するリソース制御が適用できる。リソース制御の方法としては、クラスタノード OS をシャットダウンすることや、一時停止すること、クラスタノード OS の優先度を変化させることが考えられる。今回は、演習目的の利用者への影響を抑えるために、リソース制御の方法としてクラスタノード OS のシャットダウンを用いる。

3.3.2 制御モジュールの実装

実装した制御モジュールは、ログイン監視部とクラスタノード制御部からなる。このモジュールにおいて、ログイン監視部は演習用 OS へのログイン状況を一定時間毎に確認し、クラスタノード制御部は、クラスタノード OS に対するリソースの制御を行う (図 1)。

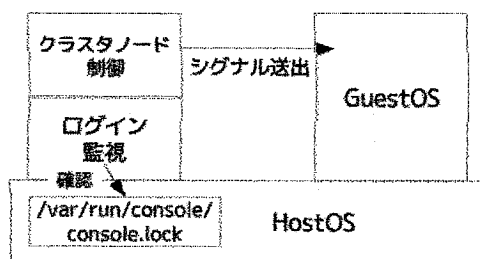


図 1: 制御モジュールの概要

3.4 クラスタの実装

これまでの設計に従い、ミドルウェアとして SCore[10] を用いて PC クラスタを実装した。また、クラスタノード、スケジューラともに OS として CentOS5.2 を用いた。実験を行ったハードウェア環境を表 1 に示す。

またこの PC クラスタでは、演習目的の利用者が現れた際にクラスタノード OS がシャットダウンされる。今回はこれをノード故障であるとみなし、チェックポイント機能を用いてタスクの再割り当てを行う。

4 評価

クラスタノード用 OS に仮想計算機技術を用いた際の PC クラスタ (仮想計算機による PC クラスタ) としての性能について評価を行う。

4.1 評価方法

クラスタノード用 OS に Native な CentOS が用いられている PC クラスタ (通常の PC クラスタ) と仮想計算機による PC クラスタのそれぞれに関して、ノードの数を 1, 2, 3, 4 台と増やしていき、N-Queen 問題の $N = 15$ における計算時間の変化を測定した。N-Queen 問題とは、 $N \times N$ の盤面にチェスのクイーンを N 個並べて、どのクイーンも互いに張り合わないような局面の数を求める問題である。

4.2 評価結果

計算にかかった時間のグラフを図 2、通常の PC クラスタにおいて、ノード 1 台の時の相対性能を 1 としたグラフを図 3 に示す。図 3 のように、双方でノード台数に比例した性能向上が見られたが、仮想計算機による PC クラスタは通常の PC クラスタと比較すると、およそ 85% の性能を持つことが分かった。

4.3 ノード機能停止時の性能

ノード機能が停止した時動作に関する評価を行うため、クラスタノードが 4 台の場合の N-Queen 問題の計算において、計算開始から 5 分の時点でノード 1 台の機能を停止させて同様の測定を行った。この際、チェックポイントを 3 分ごとに生成するように設定した。この測定の結果を図 4 に示す。

表 1: ハードウェア環境

	クラスタノード	スケジューラ
	HP Proliant ML115	
CPU	Athlon 3500+	
HDD	320GB 7200rpm	
メモリ	512MB	1GB

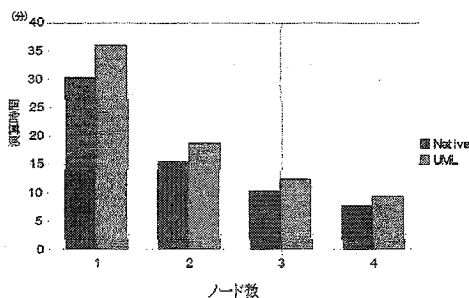


図 2: 計算時間

5 考察

仮想計算機による PC クラスタは通常の PC クラスタの 85% の性能を持つことがわかった。よって、この PC クラスタを 24 時間動作させるのと、通常の PC クラスタを 20.4 時間動作させることは計算量において同等であると考えられる。PC 演習室は通常の場合 8 時間以上開放されているため、仮想計算機による PC クラスタの方が計算量が多くなる。また、夜間は通常の PC クラスタを用いて、昼間にのみ仮想計算機による PC クラスタを用いた場合には、さらに計算量が増えると考えられるため、本方式は有用であると言える。

ノードの台数が 1, 2, 3, 4 台の測定では、台数に比例して計算性能の向上が見られた。このことから、N-Queen 問題のような CPU バウンドなアプリケーションでは、仮想化の有無に関わらずノードの台数の増加に従って PC クラスタの計算性能が向上すると考えられる。

5.1 ノード機能停止時の動作

実環境を想定してノード機能を停止させた測定では、CPU の延べ稼働時間に対するの性能がともに 50% を下回った。これは、チェックポイントの生成が 3 分ごとであったために、機能停止を行ったノードに関して、計算開始 3 分から 5 分の間の 2

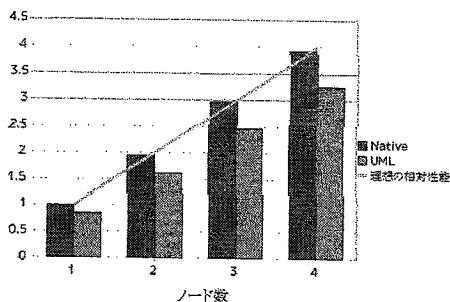


図 3: 相対性能

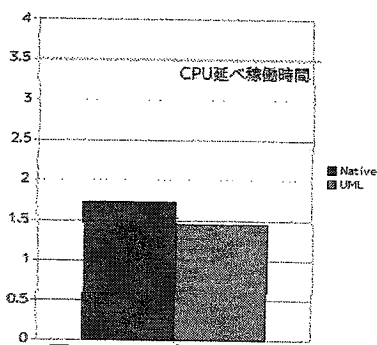


図 4: ノード機能停止時の性能

分間の計算結果が破棄されてしまったことと、スケジューラとして、途中まで計算の終わったタスクが再分割できなかったため、1台のノードにタスクを割付けてしまったことが原因であると考えられる。

これらの解決策としては、静的にチェックポイントの生成を行うだけではなく、ノードが機能を停止する際にもチェックポイントの生成を行うことや、最初にタスクを分割する際、ノードの台数個にタスクを分割するのではなく、ノードの台数×k個にタスクを分割することなどが挙げられる。

6 まとめ

本稿では、PC 演習室と共生するために仮想計算機を用いて、ホスト OS を演習用 OS、ゲスト OS をクラスタノード用 OS として並行稼働させる方式の PC クラスタについて述べた。また実験的に研究室内の常用 PC を用いた性能評価を行い、仮想計算機を用いた PC クラスタの有用性を示した。

今後の課題としては、よりノード数を増やした

場合の挙動や、I/O バウンドなアプリケーションでの性能を評価すること、演習利用とクラスタノード利用の切替えに伴うクラスタ性能の低下を防ぐための、パラメータチューニング方法の検討などが挙げられる。

参考文献

- [1] 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”,
DSM シンポジウム, (2005)
- [2] 大阪工業大学キャンパスグリッド:
<http://www.oit.ac.jp/dim/facilities.html>
- [3] 梶田 秀夫, 齊藤 明紀:
“Xen と unionfs を用いたサーバファーム運用の
可能性の検討”, 情報処理学会研究報告, (2006)
- [4] VMware:<http://www.vmware.com/jp/>
- [5] VirtualPC:
[http://www.microsoft.com/japan/windows/
virtualpc/default.mspx](http://www.microsoft.com/japan/windows/virtualpc/default.mspx)
- [6] QEMU:
<http://fabrice.bellard.free.fr/qemu/>
- [7] Xen:
[http://www.cl.cam.ac.uk/research/
srg/netos/xen/](http://www.cl.cam.ac.uk/research/srg/netos/xen/)
- [8] coLinux:<http://www.colinux.org/>
- [9] User Mode Linux:
<http://user-mode-linux.sourceforge.net/>
- [10] SCORE
<http://www.pccluster.org/>