

PC クラスタを用いたサーバベースドコンピューティングシステムの性能評価

渡辺 英俊 原田 季栄 田中 一男

(株)NTT データ 技術開発本部

e-mail: {watanabehdt, haradats, tanakakza}@nttdata.co.jp

我々は、コモディティPCをサーバとして利用し、ノード構成を自律管理し、可用性とスケーラビリティを兼ね備えたサーバノードを実現するために、自律サーバ技術を開発している。そして我々は、自律サーバ技術を利用したサーバベースドコンピューティング(SBC)システムのプロトタイプである RitaOffice を開発した。本稿では、RitaOffice のサーバノードの増設による拡張性と負荷分散の性能を検証するために、性能評価を行った結果を報告する。その結果、ノード台数を1台から2台に拡張した場合で、負荷分散機能により性能が2倍に向上すること、各ノードの物理メモリ使用量が平準化できることを確認した。

A Performance Evaluation of a Server Based Computing system with PC clusters.

Hidetoshi WATANABE, Toshiharu HARADA, Kazuo TANAKA

Research and Development Headquarters, NTT DATA CORPORATION

The authors have developed a prototype that implements a sever based computing architecture using a PC cluster called RitaOffice. RitaOffice provides increased availability and high scalability, because its server nodes can autonomously manage themselves and balance their own loads. In this paper, the impact in scalability of increasing the number of serve nodes as well as the performance of load balancing between server nodes are evaluated. The results show that increasing the number of server nodes from one to two causes a raise in performance by a factor of two, while still keeping the usage of physical memory balanced among nodes.

1. はじめに

今日、一般企業においても PC の利用が不可欠になっているが、その一方で、分散して存在する多数の端末を管理する負担が増加している。この問題の解決策の一つとして、ユーザが利用するデータや計算機資源をサーバに集め、ユーザのデスクトップをサーバ上で仮想化し、入出力のみユーザの手許の端末で行うサーバベースドコンピューティング(SBC)がある。既存の SBC 製品の代表例として、Sun Ray や Windows Based Terminal 等があるが、サーバの構築費用が高い、ユーザ数や処理内容の増加に対応するための柔軟な規模拡大が難しい、サーバの一部に障害が発生した場合の影響範囲が大きい、等の問題を抱えている。

我々は、SBC に限らず、サーバ側のシステムに対して、オープンソースのソフトウェアを利用して、可用性とスケーラビリティの両方を兼ね備え、かつ個々のノード構成を各ノードがお互いに自律的に管理して運用コストを抑えることのできる技術(自律サーバ技術)を研究開発している。そして、自律サーバ技術の有用性を検証するために PC クラスタを

用いた SBC プロトタイプシステム[1](RitaOffice)を開発した。

本稿では、RitaOffice の特徴のうち、負荷分散機能によるスケーラビリティについて検証した結果を報告する。最初に、RitaOffice 概要を述べる。次に、検証する項目を提示し、測定と考察を行う。考察の中では、サーバ側に必要とされるリソース量の見積もりも含めて行う。

2. RitaOffice 概要

RitaOffice は、自律サーバ技術を利用して開発した SBC のプロトタイプシステムであるが、通常の SBC と比較して、以下のような特徴を持っている。

- システムの初期構築費用が安い
- システムの運用管理の手間を削減可能
- スケールアウトによるサーバ性能向上が可能

RitaOffice のシステム構成図を図 1 に、機能構成図を図 2 に示す。次節からは、これらの図を元に RitaOffice の機能を説明する。

2.1. SBC サーバとしての基本的な機能

RitaOffice は、ユーザのデスクトップ画面を仮想化して仮想デスクトップとしてサーバ側に保持する。

RitaOffice サーバは複数台のサーバノードで構成する。仮想デスクトップとクライアント端末との通信をデスクトップセッション(DTS)と定義する。DTSのサーバ側の制御は仮想デスクトップ制御が行う。また、仮想デスクトップ上とアプリケーションの間の通信をアプリケーションセッション(APS)と定義する。APSの制御はAP制御が行う。

また、RitaOffice サーバは、ユーザセッションの監視を行い、ユーザがログアウトしたり、クライアント端末との接続が切れた場合でも、DTS および APS をサーバ上で永続化させ、次回ログイン時までセッションの状態を保持する。セッション永続化制御がこの処理を行う。

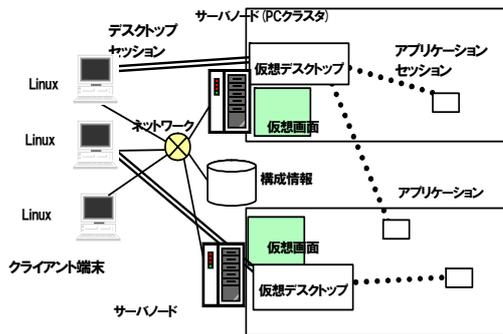


図 1 RitaOffice システム構成

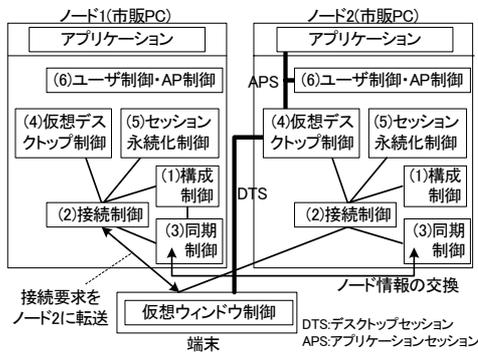


図 2 RitaOffice の機能構成図

2.2. ノード構成の自動認識

サーバノードの構成情報をノード間の共有ファイル上に保持し、定期的に参照、更新することで、各サーバノードはノード構成の自動認識が行える。新規ノードの追加や削除は自動的に処理されるため、システム管理者の管理稼働が削減できる。構成制御と同期制御がこの処理を行う。

2.3. ノードのリソース情報の自動把握

サーバノードの負荷情報やリソース使用量も共

有ファイル上に保持し、定期的に情報を交換する。この機能により、どのサーバノードからも、他のサーバノードの負荷情報やリソース使用量を把握することが可能となっている。構成制御と同期制御がこの処理を行う。

2.4. セッションの動的負荷分散

従来の SBC では、サーバに高価な SMP サーバを使用しなければならないため、システムの初期構築費用が高いことが課題となっている。この課題を解決するために、サーバノードの高性能化による性能向上(スケールアップ)ではなく、サーバノードの増設による性能向上(スケールアウト)を図れるようにする。そのためには、ユーザからの接続要求を各サーバノードに適切に分散し、ノード間でリソースの使用量を平準化する必要がある。RitaOffice ではノード間のリソース使用量平準化の方法として、セッション単位の動的な負荷分散方式を採用した。

DTS については接続数を平準化するため、その時点で最も接続数の少ないサーバノードに割り当てるようにサーバノード間で調停を行う。この処理は接続制御と同期制御が行う。

一方、アプリケーションは CPU リソースの使用量が予測できないため、APS ではサーバノードの空きメモリを平準化できるように、接続時点で最も空き物理メモリの多いサーバノードに割り当てるようにサーバノード間で調停を行う。この処理は構成制御と AP 制御が行う。ここでいう空き物理メモリは、キャッシュおよびバッファ領域を含んだものである。

3. 性能検証の目的

RitaOffice の開発目的は自律サーバ技術の有用性の確認であり、本稿での性能評価も、自律サーバ技術に関連するものを対象とする。RitaOffice はユーザの DTS と APS を自律的に各サーバノードに負荷分散することで、スケールアウトによる性能向上を狙っており、これが RitaOffice の最も特徴的な機能である。そのため、性能検証の目的を以下の3つに設定する。

1. サーバ単体の性能検証

最初に、基本的な対照標準として、RitaOffice のサーバ 1 台にユーザ 1 人だけをログインさせて、アプリケーションの操作を行い、サーバ側で消費したリソースを調べる。次に、サーバ 1 台の構成で複数のユーザをログインさせ、同様にアプリケーションの操作を行う。同時に操作を行うユーザの数を順次増加させることで、1サーバノードあ

たりの限界的な性能(処理可能なユーザ数)と、その時点でのボトルネックを調べる。

2. スケールアウト拡張性の検証

サーバノードを増加させ、1 と同様にユーザを複数ログインさせる。ノード増設によってリソース消費量がどのように変化したかを検証し、スケールアウト拡張性を評価する。

3. サーバの必要リソースの見積

以上の測定の結果から、運用時に想定される同時利用ユーザの数を元に、適切なサーバのスペックを見積もる。

ユーザが毎回同じ速度でアプリケーションを操作するという条件で測定するため、アプリケーション操作をシナリオ化する。このシナリオを各測定で走行させ、おのおの場合においてサーバ側の CPU 使用率、負荷平均(1 分移動平均)、メモリ使用量を測定した。

4. 検証環境

4.1. ハードウェア構成

サーバはラックマウントサーバ(DELL 製 PowerEdgr1650 CPU=PentiumIII-1GHz RAM=1280MB) 3 台を使用した。1 台を構成情報やアプリケーションプログラムを保持するファイルサーバとし、2 台を RitaOffice の DTS、APS の各セッションサーバとした。LAN の速度は 100Mbps である。

ログインするユーザごとに個別のクライアント PC を用意できなかったため、PC 1 台の上で、複数のクライアントデスクトップを立ち上げることにした。これは VNC[2]を利用して、X-Window の画面(ディスプレイ)を仮想的に複数動作させることで実現した。この仮想ディスプレイの上で、RitaOffice の端末ソフトウェアを動作させる。RitaOffice ではディスクレスの Linux マシンをクライアント PC として利用するため、クライアント PC の物理メモリはいずれも 512MB 以上を搭載し、複数の仮想ディスプレイを同時に動作させてもメモリ不足にならないことを確認した。クライアントマシンのスペックを表 1 に示す。

表 1 クライアント端末のハードウェア構成

メーカー	機種	CPU	RAM	台数
DELL	Optiplex GX240	Pentium4 1.8GHz	512MB	2
DELL	Optiplex GX110	PentiumIII 800MHz	512MB	1

4.2. ソフトウェア構成

RitaOffice サーバの OS は RedHatLinux7.3 を利用した。また、SBC システムの主なターゲットは、企業などの社内 LAN 環境であり、そこで利用されるのは主にオフィスアプリケーションと、WEB ブラウザ、メーラである。そこで、評価対象のアプリケーションとして OpenOffice.org1.1.0 (Calc-表計算、Writer-文書作成、Impress-プレゼンテーション作成)、Mozilla0.9.9 (WEB ブラウザ、メーラ)を使用した。

今回の評価では、複数のユーザがログインした時のサーバのリソース使用量を測定するため、ユーザ側のアプリケーション操作を統一した。表 2 に示すシナリオを作り、各ユーザが RitaOffice ログインと同時に同一操作を行えるようにした。アプリケーション操作のシナリオの実行は、VNC の RFB (RemoteFrameBuffer)プロトコルのイベント送信機能を使い、キー入力とマウス操作をバッチ処理することで行った。また、全てのユーザが一斉にログインすることは通常の使用では考えにくいので、60 秒間隔で一人ずつユーザがサーバにログインしていくようにした。シナリオにはキーボードやマウス操作の時間も含んでいるため、シナリオ全体の所要時間は 18 分(1080 秒)となっている。

表 2 操作シナリオ

手順	内容
1	ログインする(DTS が新規に作成される)
2	Calc を起動し、20 秒待機 その後、新規文書の編集作業を行う(250 秒)
3	文書を保存せずに Calc を終了し、20 秒待機
4	Writer を起動し、20 秒待機 その後、新規文書の編集作業を行う(190 秒)
5	文書を保存せずに Writer を終了し、20 秒待機
6	Impress を起動し、20 秒待機 その後、新規文書の編集作業を行う(100 秒)
7	文書を保存せずに Impress を終了し、20 秒待機
8	Mozilla ブラウザを起動し、20 秒待機 Flash を使用したページを閲覧、スクロールする(70 秒)
9	Mozilla メーラを起動する、20 秒待機 メールボックスを確認する
10	ログアウトする(DTS は永続化する)、30 秒待機
11	再度ログインする(永続化中のセッションが復帰する)(20 秒待機) 実行中の Mozilla を閉じる
12	ログアウトする(DTS は永続化する)

5. サーバ単体の性能検証

5.1. 測定方法

この測定の狙いは、ユーザの操作シナリオによってサーバのリソースがどれだけ消費されるかを調べることである。ノード 1 台の状態で行う。初めに、1 ユーザのみログインして、操作シナリオを実行し、その間のリソース使用量を測定した。次に、RitaOffice システムにログインしているユーザ数を 60 秒間隔で一人ずつ増加させながら、各ユーザがそれぞれ操作シナリオを実行し、その間のリソース使用量を測定した。

5.2. 測定結果

1 ユーザでシナリオ操作を行った場合の CPU 使用率、メモリ使用量を図 3、図 4 に示す。これらの図では、操作の折れ線グラフ(右目盛)がシナリオの操作手順を示している。

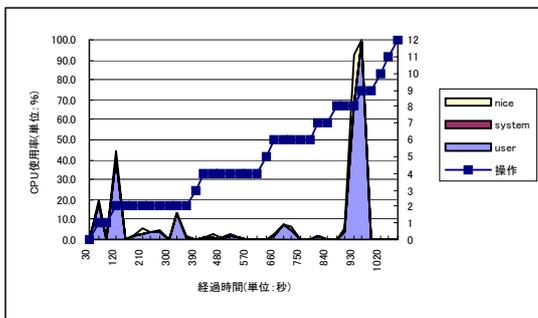


図 3 CPU 使用率(1 ノード、1 ユーザ)

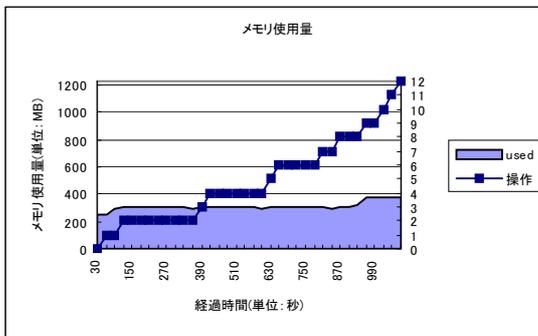


図 4 メモリ使用量(1 ノード、1 ユーザ)

また、ユーザ数を最大 30 人まで増やして測定した場合の CPU 使用率、メモリ使用量の結果を図 5、図 6 に示す。ただし、8 人目のユーザログイン以降では RitaOffice ログインは成功したものの、クライアント端末に仮想デスクトップが表示されなかったり、アプリケーションの起動に著しく時間がかかるようになり、正常に動作しなかった。このため 16 人目のロ

グインまでで新規ユーザのログインを停止した。また、このときの Calc の起動所要時間(プロセスが開始してから、シート入力可能な画面が出てくるまでの所要時間)のグラフを図 7 に示す。

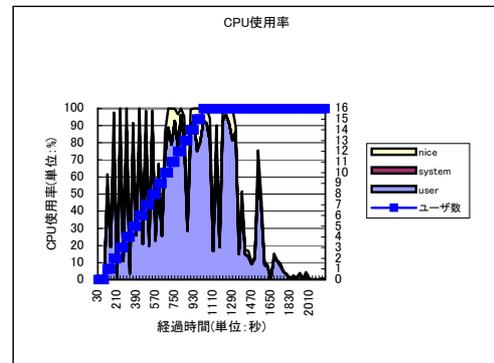


図 5 CPU 使用率(1 ノード、最大 16 ユーザ)

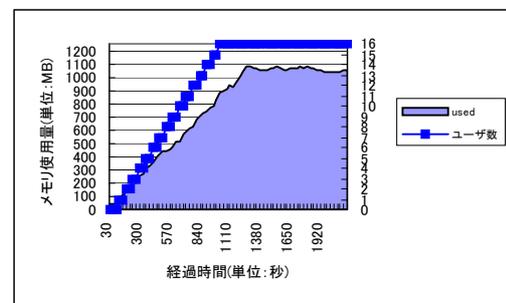


図 6 メモリ使用量(1 ノード、最大 16 ユーザ)

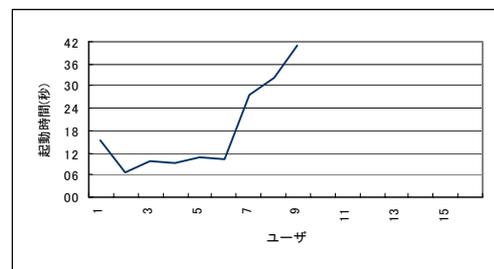


図 7 Calc の起動所要時間(1 ノード)

5.3. 考察

1 ユーザが単独でシナリオを実行した場合(図 3)については、注目すべき点が 2 つある。

1 つ目は、CPU の負荷の変動が激しいことである。この例では、ログイン時とアプリケーション起動時に負荷平均が上昇している。特に、最初のオフィスアプリケーション(Calc)起動時に負荷が高いことである。Writer や Impress の起動時の CPU 使用率がほとんど 0%であるのに対して、Calc 起動時の CPU 使用率は 40%を超えている。Calc, Writer, Impress 各

アプリケーションの起動順を変えて測定してみたが、その場合もアプリケーションの種類に関係なく、最初に起動するアプリケーションで負荷が高い状態になった。これは、OpenOffice.org では Calc、Writer、Impress いずれの場合も共通の実行形式である soffice.bin というプロセスが起動するため、最初のアプリケーションが起動するときだけ CPU に負荷がかかり、次の起動では OS のキャッシュから読み込まれるために CPU 負荷がかからないためと思われる。このことから、複数のユーザがオフィスアプリケーションの新規の起動を同時に行った場合には、CPU に大きな負荷がかかることが予想される。

2 つ目は、Mozilla の負荷が高いことである。Mozilla ブラウザでは、Flash プラグインを使ったページを閲覧しており、このアニメーションによって負荷が高くなったものと思われる。また、Mozilla メールについてはアプリケーションの起動とメールボックスの確認が連続しているため、負荷が高くなったと考えられるが、詳細な解析までは行えなかった。

次に最大 16 ユーザまでログインさせた場合については、同時に使用しているユーザ数が 8 人を超えると、CPU 使用率が常時 100% 近くになっていることが分かる(図 5)。また、Calc の起動所要時間(図 7)を見ると、ユーザ 7 人目で既に 30 秒近くかかっており、ユーザ 6 人までの場合に比べて著しく起動に時間がかかっている。

以上の点から、このサーバードではユーザ 6 人を処理するのが限界であり、その場合のボトルネックは CPU の処理速度であることが分かる。

6. スケールアウト拡張性の検証

6.1. 測定方法

この測定の狙いは、サーバードの台数を増やすことで、CPU 使用率にどのような変化が現れるかを調べることである。ノード 2 台の状態で 5 章と同様に RitaOffice システムにログインしているユーザ数を 60 秒間隔で一人ずつ増加させながら、各ユーザがそれぞれ操作シナリオを実行し、その間のリソース使用量を測定した。なお、ノード 1 台時の測定と同条件で行うために、最大 16 人までのユーザをログインさせて操作シナリオを実行し、その間の CPU 使用率、負荷平均、メモリ使用量を測定した。

6.2. 測定結果

CPU 使用率、メモリ使用量、負荷平均の結果を図 8~図 10 に示す。また、今回の測定では、すべてのアプリケーションが正常に動作した。Calc の起

動所要時間を図 11 に示す。また、DTS は、奇数番目にログインしたユーザについてはノード 1 へ、偶数番目にログインしたユーザについてはノード 2 へそれぞれ分散されていることが分かった。

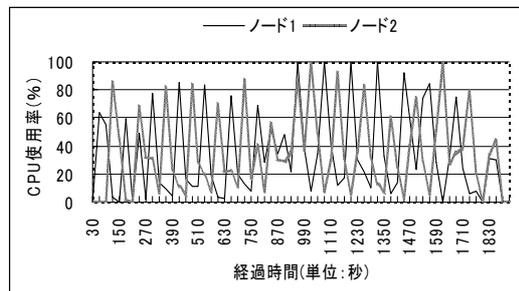


図 8 CPU 使用率(2 ノード、最大 16 ユーザ)

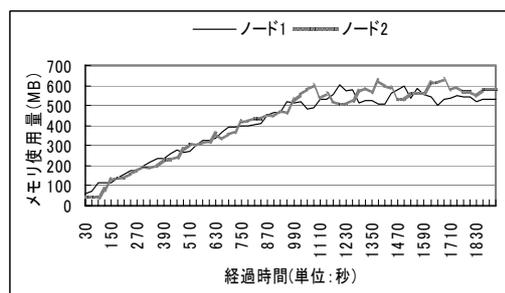


図 9 メモリ使用量(2 ノード、最大 16 ユーザ)

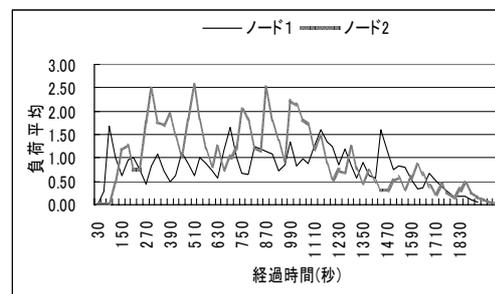


図 10 CPU 負荷平均(2 ノード、最大 16 ユーザ)

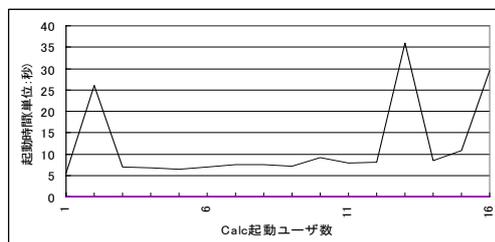


図 11 Calc の起動所要時間(2 ノード)

6.3. 考察

ノード 1 台の場合は、ユーザ数が 10 人を超えると CPU 使用率がほぼ 100% になっていたが、ノード 2

台構成では、ログインするユーザは均等に 2 台のノードに分散されたため、1 ノード当たりのユーザ数は最大で 8 人となり、CPU 使用率が低下した。ノード 1 とノード 2 で、負荷が高くなるタイミングが交互に出現しているのは、ユーザが増加する間隔(60 秒)が CPU 使用率を取得するタイミング(30 秒間隔)のちょうど 2 倍になっているためである。

CPU 使用率と負荷平均の図について、各ノードの負荷の比較を行うと、CPU 使用率としては各ノードで分散されているものの、負荷平均で見ると全体的にノード 2 の方が高いことが分かる。この原因としては以下が考えられる。APS はノードの負荷ではなく空きメモリで分散されるが、メモリ使用量のグラフを見ると、開始から 900 秒辺りまででは、ノード 2 の方がわずかにメモリ使用量が少なく、その後は両者のメモリ使用量が拮抗した状態となっている。このため、測定前半の部分ではノード 2 が APS のサーバになることが多く、ノード 2 の負荷が高くなったと推測できる。

しかし、物理メモリ使用量を見ると、両ノード間でほぼ均等に使用されており、メモリ使用量に関しては負荷分散が有効に機能していることが分かる。また、ノード 1 台の時に比べて、Calc の起動所要時間はおおむね短くなっている。が、13 ユーザ以降では 30 秒以上かかるようになっていく。このことから、1 ノードにあたり 6 ユーザ程度で CPU 処理が限界に達していることが分かる。

7. サーバの必要リソースの見積

メモリ使用量のグラフを見ると、どれも 1 ユーザあたり約 72MB 使用していることが分かる。また、図 4 から、1 ユーザでシナリオを実行した場合にはサーバ側のメモリ使用量も約 70MB であることが分かる。したがって、今回のシナリオでは、1 ユーザにつき約 70MB のメモリを消費し、サーバにログインするユーザ数に比例してサーバ側のメモリ消費量が増えることが分かった。そのため、今回のようなオフィスアプリケーションの実行を想定する場合は、メモリがボトルネックにならないようにするためには、ユーザ 1 名につき 70MB の余裕を持ってメモリを搭載し、1 ノード当たりの想定ユーザ数に比例して物理メモリを搭載すればよい。

CPU 使用率のグラフを見るとノードが 1 台の場合ではユーザ数が 10 人になったあたりで CPU 使用率が飽和している。また、ノードが 2 台の場合とも合わせて見ると、ノードあたりのユーザ数が 7 人になったところで Calc の起動所要時間が急激に増大して

いる。そのため、PentiumIII 1GHz の CPU では、1 ノードあたり 6 人が限度であるが、ノードの台数を増加させれば、それに比例して処理できるユーザ数を増やすことが可能である事がわかった。

8. おわりに

今回の測定で、RitaOffice のサーバの性能として以下のことが分かった。

1. スケールアウトによる性能の向上が確認でき、ノード 1 台の場合に比べて、ノード 2 台では 2 倍のユーザを処理できることが分かった。
2. DTS については適切に負荷分散できることが分かった。
3. APS の分散方法(空きメモリ平準化による分散)により、サーバノード間でのメモリ使用量の平準化が確認できた。しかし、これによって CPU 負荷を平準化することは必ずしもできない。
4. メモリ使用量については、1 ユーザが使用するメモリ消費量を合算しただけのメモリをサーバノードに搭載する必要がある。

今後は以下の課題について取り組む予定である。

1. より高速な CPU を使用しての評価
2. ノード台数をさらに増やした場合のスケールアウトによる拡張性の検証
3. 空きメモリ量以外の項目を考慮した APS の分散方式による評価

9. 参考文献

- [1] 渡辺,伊藤,田中「PC クラスタを用いたサーバベースドコンピューティング方式の提案」情報処理学会第 65 回全国大会,3F-3
- [2] Tristan Richardson, Quentin Stafford-Fraser, Kenneth R. Wood, Andy Hopper, “Virtual Network Computing”, IEEE Internet Computing, 1-2/1998.