

予備系システムのダウンサイジング手法及び評価 A Method of Downsizing for Backup Servers

市原 利浩[†] 原田 篤史[†] 樋口 毅[†]
Toshihiro Ichihara Atsushi Harada Tsuyoshi Higuchi

1. はじめに

東日本大震災を契機に、既存の事業継続計画（BCP：Business Continuity Plan）を見直す機運が高まり、それに伴い、業務システムの災害対策も見直されている[1]。この事業継続計画は、あらゆるリスクを想定し、想定したリスクに対して事前に対策を施しておくことで、リスク発生時に、業務を停止させないようにする、もしくは業務が停止した場合でも早急に業務を再開できるようにするものである。

情報システムの対策としては、主要情報システムのデータセンタでの運用や定期的な遠隔地へのバックアップ、及び遠隔地拠点も利用した情報システムの二拠点化などが挙げられる。これらの対策の中で、特に、二拠点化対策は費用を要することから、対策費用の削減が強く求められる[1]。その為の一手段として、サーバ仮想化技術[2][3]を利用して、遠隔地拠点のサーバを仮想化して集約[4]することで、予備系として用意する物理サーバの台数を抑制する方法がある。このサーバ集約時に利用する技術としては、サーバサイジング技術があるが[5]、従来手法では、平常時の最大リソース使用量をベースに集約計算を行うため、非常時のシステム利用者が減少する場合においては、必要以上のリソースを確保してしまう。非常時のリソース確保を抑制するために、非常時に必要なリソース量を、平常時と非常時の利用人数の比で計算する方法が考えられるが、利用者数とリソース使用量の関係は、通常、非線形の関係となるため[6]、見積もり誤差が大きくなるという課題がある。

本稿では、遠隔地拠点の予備系サーバの集約率を向上させる方法として、利用者とのリソース使用量の相関関係を表す近似式を導出し、この近似式に非常時の利用者数を入力し、非常時に必要な最大リソース使用量を算出することで、人数比計算よりも高精度に見積もる方法を提案する。

本手法の評価では、実際に利用しているサーバに本手法を適用し、従来手法よりも集約率が向上できることと、非常時のリソース使用量を人数比計算よりも高精度に見積もることが出来ることを確認した。

2. 従来技術の課題

2.1 仮想化技術によるサーバ統合の概要

図1に示すように、従来、サイジング技術を利用したサーバ統合では、システム負荷情報の収集、サイジングによる統合先の物理サーバ台数の見積もり、サイジング結果の確認として、試験・評価を行い、評価結果によって、再度、サイジングからやり直す手順となっている。

最初の負荷情報の収集では、統合対象サーバのCPU、メモリ、ディスク容量、ディスクアクセス数、ネットワークアクセス数などのシステム負荷を一定期間計測し、収集する。

次のサイジングでは、統合対象サーバの仕様、収集した負荷情報、統合先の物理サーバの仕様を元に、統合先物理サーバの台数と仮想マシンの配置を計算する。

最後に、シミュレーションや待ち行列解析、プロトタイプピングなどの手法を用いて、サイジング結果の評価を行い、リソースや性能に過不足が生じていないかを確認する。ただし、この評価で、問題があることが判明した場合は、サイジングからやり直すこととなる。このサイジングと評価の手順を繰り返すことで、精度の高いサーバ統合が可能となる。



図1 従来のサーバ統合手順

2.2 仮想化サーバのサイジング

仮想化サーバのサイジングでは、統合前と統合後のシステム動作による仕事量は同じである、という考えに基づき、統合先の物理サーバ台数を計算し、次に、仮想マシンの最適配置の計算を行う。

統合先物理サーバの台数は、統合対象サーバのリソース使用量と統合先物理サーバのリソース量から算出する。CPUリソースの場合の計算式は下記となる。

$$\text{統合先物理サーバ台数} = \frac{\sum (\text{統合対象サーバ CPU使用量})}{\text{統合先サーバ使用可能 CPU量}}$$

この統合対象サーバのCPU使用量の算出式は下記となる。CPU性能は、CPUベンダが公開するSPECintの値を利用する。SPECintは、SPEC (The Standard Performance Evaluation Corporation) [7]が規定した、ベンチマークテストの一つで、整数演算処理性能評価である。

$$\text{統合対象サーバのCPU使用量} = \text{CPU性能} \times \text{最大CPU使用率}$$

統合先物理サーバのCPU使用量は次の式によって算出する。係数Kは、ハイパーバイザの種類と仮想CPU数、物理CPU数の比率で確定する値[8]である。

$$\text{統合先物理サーバのCPU使用量} = K \times \text{CPU性能} \times \text{最大CPU使用率}$$

なお、仮想化サーバへの統合では、仮想化によって生じるオーバーヘッドを考慮することで、見積もり精度を向上させる手法が考案されている[8]。この手法では、ディスク

[†] 三菱電機株式会社 Mitsubishi Electric Corporation

I/O 及びネットワーク I/O のエミュレーションによって生じる CPU オーバヘッドを算出し、上記統合対象サーバの CPU 使用量に加算する。CPU オーバヘッドは下記方法で算出する。

$$\text{ディスク負荷による CPU 使用量} = \frac{\text{CPU 性能} \times \text{ディスク変換率}}{\text{最大ディスク IO 数}}$$

$$\text{ネットワーク負荷による CPU 使用量} = \frac{\text{CPU 性能} \times \text{ネットワーク I/O 変換率}}{\text{最大ネットワーク I/O 数}}$$

ディスク I/O 変換率とネットワーク I/O 変換率は、事前に各 I/O 量と CPU オーバヘッドの関係を調査しておく必要がある[9]。

統合先でのリソース使用量を算出した後は、仮想マシンを配置する物理サーバを決める。この計算では、ビンパッキング問題などを解くヒューリスティックに基づいたアルゴリズムによって決定する方法が最も一般的である。一次元ビンパッキング問題を解くアルゴリズムとして有名な手法としては、下記の手法が挙げられるが[10][11]、本書での説明は割愛する。

- ・ NFD (Next Fit Decreasing) 法
- ・ FFD (First Fit Decreasing) 法
- ・ BFD (Best Fit Decreasing) 法

2.3 課題

以上述べたように、従来のサイジング手法では、平常時の最大リソース使用量を用いて算出する。そのため、非常時のシステム利用者が減少する場合においては、必要以上のリソースを確保してしまうことがある。不要なリソース確保を抑制するために、非常時に必要なリソース量でサイジングを行う必要があるが、図 2 に示すように、利用者数とリソース使用量の関係は、通常、非線形の関係となるため、非常時に必要なリソース量を、平常時と非常時の単純な利用人数の比で計算（線形予測）すると、見積もり誤差が大きくなるという問題があった。

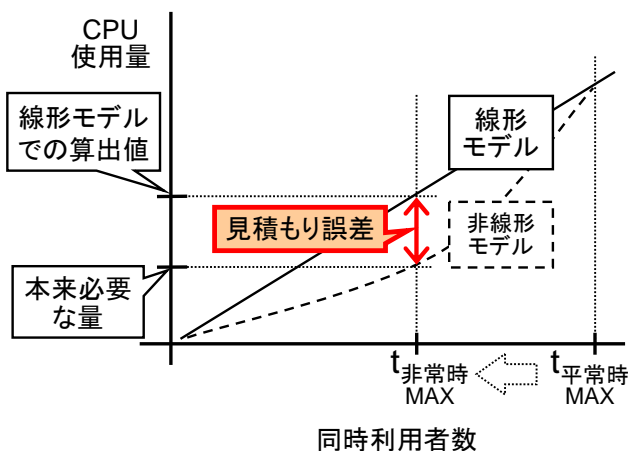


図 2 見積もり誤差

3. 災害対策向けダウンサイジング手法

3.1 概要

2.3 節の課題を解決する方法として、非常時のリソース使用量を、単純な人数比計算よりも精度よく見積もったうえでサイジングを行い、従来よりも対策コストの削減を可能とする方法を提案する。

本手法は、非常時の利用者数が推測できることと、非常時の利用者は対象システムを平常時と同等に利用すること（1人当たりのリソース使用量が平常時と同等であること）を前提とする。その前提下において、非常時の利用者数が平常時よりも減る場合には、予備系として確保するリソース量を従来よりも抑制し、予備系の維持コストを削減を可能とする。逆に、非常時の利用者数が増え、リソース使用量が増加する場合においては、非常時のリソース使用量を精度よく見積もることで、リソース不足に陥る可能性を低減する。

図 3 は、我々が提案するサーバ統合の手順である。従来手法では、統合前と統合後のシステムの仕事量は同じであることを前提としたが、本手法では、サイジングを行う前に、平常時に計測した負荷情報と、非常時に想定する利用者数から、非常時の仕事量（リソース使用量）を見積もり、非常時の仕事量でサイジングを行う。これにより、遠隔地拠点の非常時用サーバとして、余分に確保していたリソースを削減可能とする。

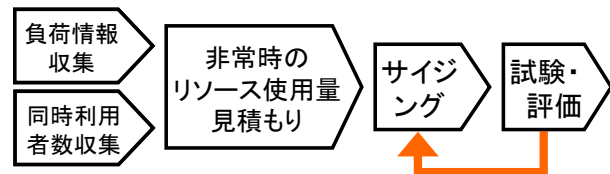


図 3 提案するサーバ統合の手順

提案手法では、従来と同様に平常時の負荷情報の収集を行うが、その際にシステムの同時利用者数をあわせて収集しておく。

次に、非常時のリソース使用量見積もりのステップにおいて、収集した負荷情報と同時利用者数から、システムの同時利用者数とリソース使用量の相関関係を表す近似式を導出する。そして、事業継続計画などによって想定する業務の遂行人数を、非常時のシステムの同時利用者数と考え、この利用者数を近似式に入力することで、非常時のリソース使用量を算出する。

サイジングのステップでは、算出した非常時のリソース使用量をもとに、統合先物理サーバの台数と仮想マシンの配置を計算する。

試験・評価のステップは従来手法と同様に実施し、問題があることが判明した場合には、サイジングからやり直す。システムのリソース使用量と利用者数の相関関係は、業務特性やシステム特性によって異なるため、統合対象サーバ毎に、近似式を導出する。近似式の導出方法も含め、本手法の詳細については、3.2 節にて説明する。

なお、本手法で必要となるシステムの同時利用者数は、平常時の負荷情報の収集とあわせて計測することを前提と

している。利用者数の情報が取得できないシステムにおいては、サーバへ接続する TCP コネクション数を利用者数として用いる。TCP コネクション数のカウントとして、例えば、netstat コマンドを利用する場合は、ユーザアクセス以外のコネクションも含まれてしまうため、ユーザアクセス以外のコネクション数を省く処理が必要となる。この処理は、ループバックとサーバ同士の TCP コネクション数をユーザアクセス以外のコネクション数として計測し、全体のコネクション数から差し引くことで実現可能と考える。

3.2 非常時のリソース使用量の見積もり

本節では、CPU リソースの場合を例として、非常時のリソース使用量の算出方法を述べる。

仮想化によるサーバ統合では、2.2節で述べたように、I/O エミュレーションによって生じる CPU オーバヘッドの負荷を考慮する必要がある。そのため、統合対象サーバを S_i とした場合 (i はシステム ID)、非常時の CPU 使用量を T_{cpu_i} 、非常時のディスク I/O による CPU 使用量を T_{disk_i} 、非常時のネットワーク I/O による CPU 使用量を T_{net_i} 、ハイパーバイザの種類と仮想 CPU 数・物理 CPU 数の比率で確定する値を k とすると、 S_i の非常時の最大 CPU リソース量 W_i は、下記の式で求めることができる。

$$W_i = k \times T_{cpu_i} + T_{disk_i} + T_{net_i}$$

この T_{cpu_i} 、 T_{disk_i} 、 T_{net_i} は、非常時の最大 CPU 使用率を P_i 、非常時の最大ディスク I/O 数を D_i 、非常時の最大ネットワーク I/O 数を E_i とすると、2.2節と同様に、下記の計算式によって求めることができる。

$$T_{cpu_i} = S_i \text{のCPU性能} \times P_i$$

$$T_{disk_i} = S_i \text{のCPU性能} \times \text{ディスク I/O変換率} \times D_i$$

$$T_{net_i} = S_i \text{のCPU性能} \times \text{ネットワーク I/O変換率} \times E_i$$

以上により、非常時の P_i 、 D_i 、 E_i を算出することにより、統合対象サーバ S_i の非常時の最大 CPU リソース量 W_i を求めることができる。ただし、 P_i 、 D_i 、 E_i の値は、各サーバの構成要素やプログラムの実装に依存して異なるため、統合対象サーバ毎に算出する必要がある。

特定条件下における消費リソース量は、性能モデルに基づくシミュレーション手法[12]等により見積もることができる。しかし、これをサーバ統合に利用する場合、統合対象サーバ毎に性能モデルの作成が必要となり、現実的ではない。そこで、非常時に必要なリソース量を、事業継続計画などで想定する非常時の利用者数から容易に見積もれることができる手法を提案する。

本手法では、効率的にリソースの見積もりを行うために、利用者と統合対象サーバ間の構成要素をブラックボックスとして扱うこととし、収集した負荷情報から、同時利用者と P_i 、 D_i 、 E_i の相関関係を表す近似式を、最小二乗法などの回帰分析を用いて導出する。非常時の P_i 、 D_i 、 E_i は、導

出した近似式に、想定する非常時の最大同時利用者数を入力することで算出する。

非常時の最大 CPU 使用率 P_i の場合を例として、近似式の導出手順及び非常時の P_i の算出例を以下に示す。

(1) 統合対象サーバ S_i から収集した平常時の負荷情報を集計処理し、表 1 に示すような、同時利用者数と CPU 使用率の表を作成する。この CPU 使用率とは、ユーザ使用率、システム使用率、I/O 待ち比率の合計値である。

表 1 利用者数毎のリソース使用率例 (集計値)

同時利用者数[人]	1	3	7	13	22	...
平均 CPU 使用率[%]	2.5	3.5	7.2	9.1	19.3	...

(2) 同時利用者数と CPU 使用率の相関関係を表す近似式を、線形最小二乗法を用いて導出する。

線形最小二乗法は、 X を同時利用者数、 Y を CPU 利用率として、計測点列を (x_1, y_1) 、 (x_2, y_2) 、 \dots 、 (x_n, y_n) とした場合、 n 次の近似式を導出することができる。

次数 $n=2$ とすると、下記式を得ることができる。

$$Y = \alpha_{cpu_i} \cdot X^2 + \beta_{cpu_i} \cdot X^1 + \gamma_{cpu_i}$$

係数 α_{cpu_i} 、 β_{cpu_i} 、 γ_{cpu_i} は、統合対象サーバ S_i の CPU 使用率と同時利用者数の相関特性を表す定数値である。

(3) この近似式に、非常時の最大同時利用者数 t_i を代入することで、非常時の最大 CPU 使用率 P_i を得る。この計算式は下記の通りである。

$$\text{非常時の最大 CPU 使用率 } P_i = \alpha_{cpu_i} \cdot t_i^2 + \beta_{cpu_i} \cdot t_i + \gamma_{cpu_i}$$

図 4 は、上記(1)~(3)の手順で導出した近似式との実測値と近似式の関係を図示したものである。

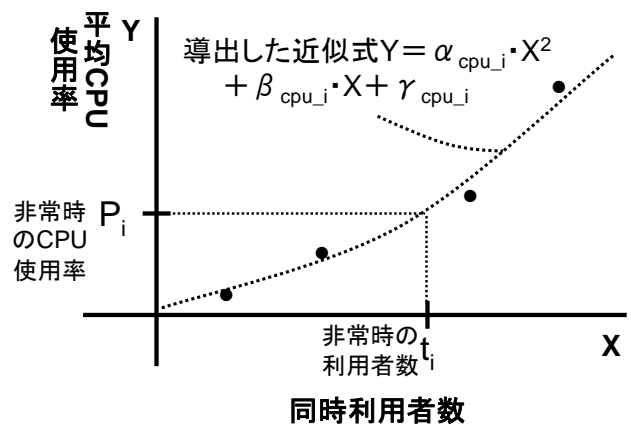


図 4 実測値の近似式の関係

前述と同様の方法で、非常時の最大ディスク I/O 数 D_i と非常時の最大ネットワーク I/O 数 E_i についても算出する。

ディスクに関しては、各時刻の同時利用者数と、ディスク I/O の負荷情報を集計し、表 1 と同じ要領で、統合対象サーバ S_i の利用者数毎の平均ディスク I/O 数の表を作成する。このディスク I/O 数は、ディスク読込 I/O 数とディスク書込 I/O 数を合計値である。

ネットワークについても同様に、統合対象サーバ S_i の利用者数毎の平均ネットワーク I/O 数の表を作成する。このネットワーク I/O 数は、ネットワーク送信 I/O 数とネットワーク受信 I/O 数の合計値である。

同時利用者数とディスク I/O 数の相関関係、及び同時利用者数とネットワーク I/O 数の相関関係を表す近似式をそれぞれ導出し、各近似式に非常時の最大同時利用者数 t_i を代入し、非常時の最大ディスク I/O 数 D_i 及び非常時の最大ネットワーク I/O 数 E_i を得る。これらの計算式は下記のようになる。

非常時の最大
ディスク I/O 数 $D_i = \alpha_{\text{disk}_i} \cdot t_i^2 + \beta_{\text{disk}_i} \cdot t_i + \gamma_{\text{disk}_i}$

非常時の最大
ネットワーク I/O 数 $E_i = \alpha_{\text{net}_i} \cdot t_i^2 + \beta_{\text{net}_i} \cdot t_i + \gamma_{\text{net}_i}$

α_{disk_i} , β_{disk_i} , γ_{disk_i} は、統合対象サーバ S_i の同時利用者数とディスク I/O の相関特性を表す定数値であり、 α_{net_i} , β_{net_i} , γ_{net_i} は、ネットワーク I/O との相関特性を表す定数値である。

以上の方法によって導出した、 P_i , D_i , E_i を算出する近似式が、統合対象サーバ S_i の同時利用者数と各リソースの相関関係を表すモデル式である。

3.3 サイジングによるサーバ台数計算

非常時に必要とするリソース使用量を算出した後の、サーバ台数の算出は、従来のサイジングと同じ方法で算出する。すなわち、統合対象サーバ S_i を統合先で稼働させた場合の仮想マシンを S'_i とすると、仮想マシン S'_i の非常時の最大リソース量 W_i から、2.2 節で述べた BFD などのアルゴリズムを用いて、遠隔地拠点の物理サーバ台数、及び仮想マシン S'_i を配置する物理サーバを決定する。

4. 評価

4.1 概要

本評価では、実際に利用している業務システムのサーバに、今回提案した手法を適用し、2 通りの評価を行った。

1 つは、従来手法と本手法のリソース使用量の比較、すなわち、平常時の同時利用者が最大時のリソース使用量と、非常時に必要な最大リソース使用量の比較である。もう一つは、非常時のリソース使用量の算出を、単純な人数比計算で行った場合との比較である。

また、4.3 節では、本手法を実装したサイジングツールを試作し、本手法がサイジングに適用できることを確認する。なお、本評価では、非常時のサーバの同時利用者数を 5 名と想定し、比較評価を行う。

なお、今回評価対象とした業務システムの負荷情報は、15 秒間隔で 1 週間程度のデータを収集したもので、この期間の最大同時利用者数は、15 名であった。

4.2 評価結果

4.2.1 リソース使用量の比較

3.2 節に示した方法によって、業務サーバの負荷情報（実測値）から導出した、同時利用者数と CPU 使用率及び同時利用者数とメモリの相関関係を表す二次のモデル式（近似式）を以下に示す。

$$\text{CPU 使用率} : Y = 0.0453 \cdot x^2 + 0.131 \cdot x + 2.8495$$

$$\text{メモリ使用量} : Y = -6.1896 \cdot x^2 + 229.51 \cdot x + 1815.6$$

このモデル式をそれぞれ図示すると、図 5、図 6 のようになる。

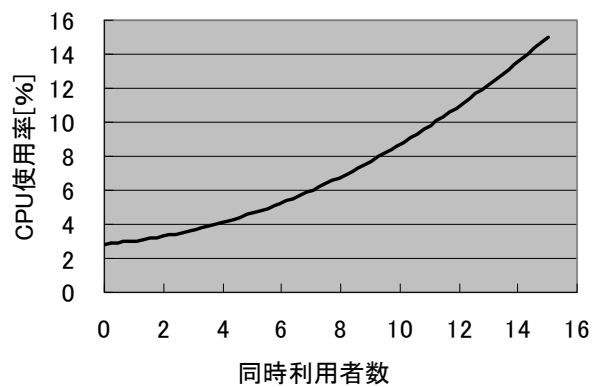


図 5 近似モデルによる予測 (CPU)

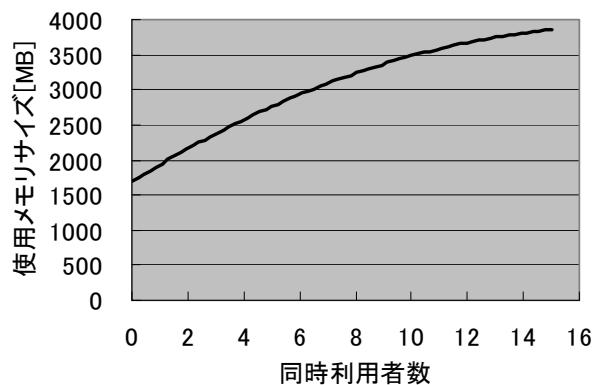


図 6 近似モデルによる予測 (メモリ)

これらのモデル式に、非常時の同時利用者数 5 名を入力し、算出した結果、それぞれの最大リソース使用量は、下記となった。

$$\text{CPU 使用率} : 0.0453 \times 5^2 + 0.131 \times 5 + 2.8495 = 4.6 [\%]$$

$$\begin{aligned} \text{メモリ使用量} : & -6.1896 \times 5^2 + 229.51 \times 5 \\ & + 1815.6 = 2808.7 \text{ [MB]} \end{aligned}$$

CPU リソースは、平常時の最大リソース使用量が 15.7% であったことから、本手法との差分は $15.7\% - 4.6\% = 9.1\%$ となり、今回のサーバ 1 台分だけで、1 割近くのリソースを抑えることができる結果となった。

同様に、メモリリソースにおいては、平常時の最大リソース使用量が 3796(MB)であったことから、 $3796(\text{MB}) - 2808(\text{MB}) = 988(\text{MB})$ となり、1GB 近く少なくて済む結果となった。

これらの結果より、同時利用者によってリソース使用量も変動することと、本手法によって非常時のための予備系サーバのリソース量を抑えることができることを確認した。これによって、集約率が向上すると考える。本結果はサーバ 1 台分の差分量であるため、統合対象サーバの台数が多い程、従来手法との差が大きくなる。事業継続計画の一環で、二拠点化対策を行う企業では、複数のサーバを予備系として構築する機会が多いため、本技術適用によって災害対策コストが削減できると考える。

4.2.2 人数比計算方法との比較

本節では、非常時の最大リソース使用量を、平常時の最大リソース使用量から単純な人数比によって算出する方法と本手法の比較を行う。

4.2.1節で導出したモデル式と人数比計算による式の決定係数 (R^2 値) をそれぞれ算出し、比較する。決定係数は、回帰モデルの適合度の表す指標であり、1 に近い程、標本値によく当てはまっていることを示すものである。なお、決定係数は、Microsoft Excel 2003 の LINEST 関数を用いて算出した。

CPU リソースにおいて、最小二乗法を用いて導出したモデル式 (4.2.1節で算出したモデル式) と人数比計算方法の決定係数は、モデル式が 0.993、人数比が 0.976 となった。メモリリソースにおいては、モデル式が 0.9991、人数比計算方法が 0.8694 となった。人数比計算方法を人数比モデルとし、最小二乗法を用いて導出したモデル式を非線形モデルとした図の比較を、それぞれ図 7、図 8 に示す。いずれの場合においても、決定係数は、最小二乗法で導出したモデル式の方が、実データを推定するモデルとして近いことを確認した。

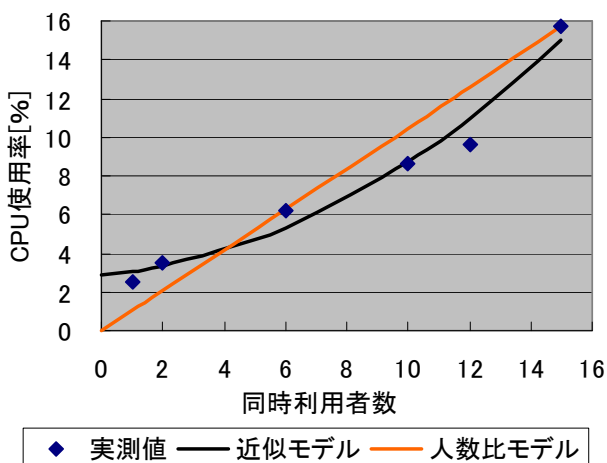


図 7 近似モデルと人数比計算モデル (CPU)

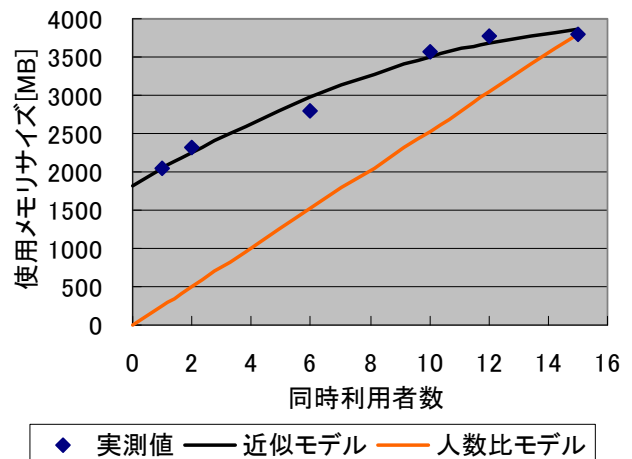


図 8 近似モデルと人数比計算モデル (メモリ)

上述の決定係数の比較及び図 7、図 8 の実測値との比較から、メモリリソースにおいては、乖離が大きい結果となった。これは、人数比計算モデルは、利用者数がゼロの場合、メモリ使用量もゼロとなる線形モデルであるのに対し、今回の評価対象システムは、利用者がゼロの場合でも、システム単独でメモリリソースを消費するためであり、人数比計算モデルがそぐわない場合があることを示している。

次に、人数比計算モデルでの、非常時の最大リソース使用量を算出し、近似モデルとの比較を行う。人数比計算モデルで算出すると、非常時の最大リソース使用量は、それぞれ下記の結果となった。

$$\text{CPU 使用率} : 15.7 [\%] \times \frac{5 [\text{人}]}{16 [\text{人}]} = 5.2 [\%]$$

$$\text{メモリ使用量} : 3796 [\text{MB}] \times \frac{5 [\text{人}]}{16 [\text{人}]} = 1265.4 [\text{MB}]$$

CPU リソースにおいては、 $5.2\% - 4.6\% = 0.6\%$ の見積もり差がある結果となり、単純な人数比計算の方法よりも、今回提案した手法の方が、CPU リソースを抑えることができるという結果となった。

これに対し、メモリ使用量は、 $1265 [\text{MB}] - 2808 [\text{MB}] = -1543 [\text{MB}]$ となった。メモリリソースについては、単純な人数比で算出した方が 1.5GB 程度少なくて済むという結果となったが、実データとの乖離が大きい結果となった。

通常、サイジングでは、見積もり誤差が発生することを考慮し、ある程度の余裕を持たせて集約計算を行うが、集約台数が増えると、見積もり誤差を吸収しきれなくなり、リソース不足を招く可能性が高くなる。人数比計算モデルでは、利用者数が少なくなる程、乖離が大きくなるため、近似モデルを利用する方法によって、予備系のリソース量を抑える方が望ましいと考える。

4.3 サイジングツールへの適用

本節では、3章で提案した手法を実装したサイジングツールを試作し、本手法がサイジングに適用できることを確認する。

本手法を実装したサイジングツールは、統合対象サーバからの負荷情報の収集機能、及び負荷情報の統計処理を含めたサイジング機能を有しており、非常時の利用者数を指定することで、非常時のリソース使用量を考慮したサイジングを行う。

なお、負荷情報収集時のサーバの利用者数は、サーバへのTCPコネクション数から、利用者のサーバアクセス以外のTCPコネクション数を省いてカウントする方式を実装したが、今回、評価対象としたサーバは、利用者数のログ情報が取得できるシステムであったため、このログ情報を、本ツールにインプットして実行した。

今回の試行では、遠隔地拠点に置く予備系サーバは、本番系サーバよりも低スペックのサーバを利用するものとして、平常時の最大利用者数である15名と、非常時の最大利用者数5名の2通りでのサイジングツールを実行した。

図8、図9に示す試行結果の通り、非常時の最大利用者数でサイジングすることで、予備系として必要なリソース量を抑えることができる結果を得た。これにより、今回提案した手法が、サイジングに適用できることが確認出来たと考える。

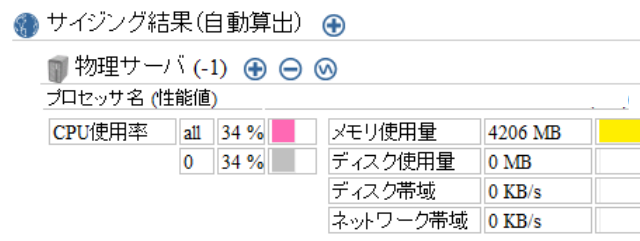


図9 平常時最大利用者数でのサイジング結果

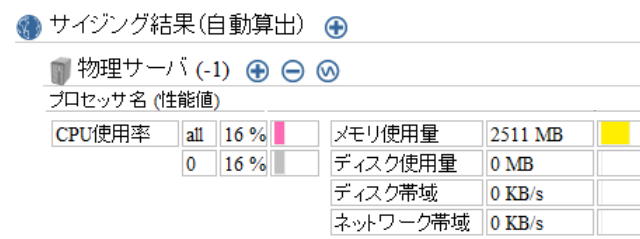


図10 非常時最大利用者数でのサイジング結果

5. 終わりに

情報システムの二拠点化対策のコスト削減を実現する方法として、遠隔地拠点の予備系サーバを仮想化し、サーバを集約する方法が考えられる。この際、平常時と同等のリソース使用量でサイジングを行ってしまうと、不要なリソースを確保してしまい、コスト削減の余地が残ってしまうという課題があった。また、確保するリソース量を抑制するために、平常時の最大リソース量から、単純な人数比計

算で見積もると、利用者数とリソース使用量の関係は、通常、非線形の関係となるため、誤差が大きくなるという課題があった。

そこで、今回、事業継続計画などによって想定する業務遂行人数から、非常時に必要とするリソース量を、モデル式から推定し、サイジングを行う方法を提案した。

モデル式を導出する方法としては、平常時に取得した負荷情報から、線形最小二乗法を用いた回帰分析によって導出する方法を提案した。本方法は、従来のサイジングで収集している負荷情報に、システムの同時利用者数の項目も追加することで、モデル式が導出可能なため、現状のサイジング手法に適していると考える。

今後の課題は、今回実装したTCPコネクション数のカウントによる、システムの同時利用者数情報の収集機能の検証と、近似モデルの精度向上である。

前者については、今後、今回実装したサイジングツールを、実サーバに適用し、本機能の妥当性検証を行う。

後者の近似モデルの精度向上としては、サイジングに設定する情報として、CPUのコア数が得られるため、待ち行列モデルを用いて、CPUの近似モデルを詳細化し、評価を実施する予定である。

参考文献

- [1] 独立行政法人情報処理推進機構, 「情報システムの基盤の復旧に関する対策の調査報告書」, 2012年7月, <http://www.ipa.go.jp/files/000004636.pdf>
- [2] RedHat, Red Hat Enterprise Virtualization, <http://jp.redhat.com/products/cloud-computing/virtualization/>
- [3] VMware, <http://www.vmware.com/jp/virtualization>
- [4] Lauren Whitehouse, 仮想ディザスタ・リカバリ, Storage Magazine 2011年4月号翻訳記事 (JDFS), <http://www.jdsf.gr.jp/backup/stm/201105.html>
- [5] 中島 淳之介, 小原 光弥, 豊原 啓治, VMwareの基本 ~仮想化のための設計・構築・運用のポイントがわかる, 技術評論社, (2014).
- [6] Ian Molyneux, アート・オブ・アプリケーション パフォーマンステスト, 2009,
- [7] Standard Performance Evaluation Corporation, <http://www.spec.org/>
- [8] 金田 典久, 飯塚 剛, 金木 祐介, 仮想マシンのオーバーヘッドおよび負荷評価方法の提案, 情報処理学会研究報告(OS-113), Vol.2010 No.2 (2010).
- [9] 網代 育大, 田中淳裕, 仮想計算機環境における資源管理オーバーヘッドの評価, 情報処理学会研究報告(EVA-17), Vol.2006 No.66 (2006).
- [10] 網代 育大, 田中淳裕, サーバ統合のための組み合わせ最適化アルゴリズムの提案と評価, 人工知能学会全国大会論文集, 21st 巻 pp.3E8-1 (2007).
- [11] 豊田 丈輔, 材料取り合わせのための Revised Minimum Bin Slack法, 経済研究 / 大阪府立大学経済学部 編, Vol.53-4 No.222 (2008).