

## コンピュータネットワークにおける複数資源同時割当て方式の提案

津村 重宏<sup>\*1</sup>, 栗林 伸一<sup>\*2</sup>

Simultaneous resource allocation method for computer communication networks

Shigehiro TSUMURA<sup>\*1</sup>, Shin-ichi KURIBAYASHI<sup>\*2</sup>

**ABSTRACT:** In computer communication networks, many servers, distributed in various locations, offer processing ability for each service request. To utilize the processing ability of the servers, it is necessary to allocate simultaneously both the processing resources and the access bandwidth that connects the server to the backbone network. Resource allocation in computer communication networks can be modeled as multiple resource pairs in the system, each resource pair comprising server (computing resource) and access bandwidth (network resource), and both computing and network resources being allocated simultaneously in one selected resource pair.

This paper proposes a method for the efficient joint resource allocation method of both computing and network resources. It is assumed that there are multiple centers distributed in the network and each center includes both servers and its access bandwidth to the backbone network. The key idea of the proposed method is first to identify the resource type (computing or network) for which the requested amount is the largest proportion of the maximum resource of that type, and then, from the centers that could meet the request, to select the center with the least available amount of the selected resource type. This paper also proposes an extension of the joint resource allocation method for the case where each service request declares the maximum permitted network delay time, and where both the size of resource and the network delay time are considered in the selection of a center. It is demonstrated by simulation evaluations that the proposed methods could reduce the request loss probability and also reduce the total amount of the resources required by up to 20%, compared with the conventional method.

**Keywords:** resource allocation, computer communication network, utility computing, grid computing

(Received September 28, 2006)

### 1. まえがき

コンピュータネットワークでは、地理的に分散して設置されたサーバの計算能力とそのサーバとクライアントを接続するアクセス帯域を要求毎に提供する。近年の動向として、ユーザが自前のサーバやネットワークを構築・運用するのではなく、必要な時に必要な量の計算能力と帯域をネットワーク運用業者などから借り受けるユーティリティサービスやコンバージドサービスを利用する傾向が強くなってきている<sup>(1)</sup>。ユーティリティサービ

ス、グリッドコンピューティング<sup>(2)</sup>、ブロードバンド・ユビキタスネットワーク<sup>(3)</sup>などが今後社会インフラとして導入された場合、この傾向はさらに強まると想定される。

この形態では、同時に発生する複数の要求それぞれに対して、複数のサーバの中から最適なサーバを選択しそのサーバの計算能力を割当てることが経済的なサービス提供上重要な課題となる。また、サーバを利用するためにはそこに接続するためのネットワーク帯域も“同時に”必要になる、ここで、中継バックボーンネットワークは一般に広帯域であり、ネットワーク資源（帯域）の割当てを考える場合には特にサーバと中継バックボーンネットワーク間のアクセス帯域を意識する必要がある。

このため、コンピュータネットワークにおける資源割

<sup>\*1</sup>: 工学研究科物理情報工学専攻博士前期課程学生

<sup>\*2</sup>: 理工学部情報科学科教授 (Professor, Dept. of Computer and Information Science, e-mail: kuribayashi@st.seikei.ac.jp)

当ては、「サーバとアクセス帯域の“資源セット”が複数存在し、要求発生毎にその中から資源セットを1つ選択し、要求された計算能力（コンピューティング資源）とアクセス帯域（ネットワーク資源）の両方を“同時に”割当ててモデル」と考えることができる。一般に、要求される計算能力と帯域の大きさは同じではないため（かつ要求毎に大きさも異なる）、1つの資源種別だけを考慮したアルゴリズムを複数種別資源の割当てに適用しても最適な割当てでは期待できない。

このため、本稿では、コンピューティング資源とネットワーク資源という2つの資源種別の資源を“同時に”割当ててを前提とした最適資源割当て方式を検討する（本稿では、以後“複数資源同時割当て方式”と呼ぶ）。また、本稿では要求棄却率（2つの資源種別で必要な量の資源を割当てられないために要求が拒否される割合）を最小化することを目的に検討を行う。

ところで、類似の研究として、文献(4)はグリッドシステムにおけるデッドラインスケジューリング方式を提案している。これは、サーバ（コンピューティング資源）とネットワークの両方を考慮したサーバ選択方式であり、サーバの処理時間とネットワークの転送時間の総和が最も長いサーバを選択し（但し、許容最大サービス完了時刻（デッドライン）までにサービスを完了できることが条件）、より多くのジョブを処理することを狙ったものである。文献(4)は複数のジョブがサーバならびにネットワークを順番に利用する待ち行列モデルを前提としている。一方、本稿では、クライアントはジョブの処理でなく、専用の計算能力と帯域を要求するモデルを前提としている点異なる。クライアントは、割当てられた資源を用いて独自のジョブやアプリケーションサービスを実施する。従来からCPU貸し、帯域貸しと言われているサービスに近いモデルである。さらに、本稿では、計算能力と帯域の資源セットが複数存在し、要求発生時にその中から資源セットを1つ選択し、そこから計算能力と帯域の両方を“同時に”割当てることを前提としている点も異なる。

2章では、コンピュータネットワークの資源割当てモデルについて述べる。3章では、複数種別の資源を同時に割当てての場合の課題を整理する。4章では、複数資源を同時に割当てることを前提とした最適資源割当て方式を提案し、シミュレーション評価により従来方式に比べた有効性を明らかにする。また、5章では、ネットワーク遅延を考慮した複数資源の同時割当て方式を提案し、4章と同じくシミュレーション評価により従来方式に比べた有効性を明らかにする。さらに、要求棄却率を低下す

ることを目的に、要求を分割する方式ならびに割当て資源を移し変えることで新たな要求を処理する方式を6章で提案する。簡単なシミュレーションにより、その有効性を評価する。7章では、要求棄却率を削減する他の同時割当て方式を提案する。8章はまとめである。

## 2. コンピュータネットワークの資源割当てモデル

コンピュータネットワークの構成を図1に示す。地理的に分散した複数のセンタが存在し、各センタにはサーバ（計算能力）と中継バックボーンネットワークに接続するためのアクセス帯域が存在する。クライアントは、計算能力とそこに接続するための帯域をシステム資源管理装置に要求する。システム全体の資源はシステム資源管理装置により管理され、システム資源管理装置はクライアントからの要求毎に資源を割当て、解放する。ここで、クライアントとサーバ間を接続するためには中継バックボーンネットワーク内の帯域も必要となるが、一般に各センタのアクセス帯域に比べて中継バックボーンネットワークの帯域は非常に大きく、資源割当てのネックとなるのはアクセス帯域と想定される。そこで、本稿ではネットワーク資源として各センタのアクセス帯域のみを扱うものとする。

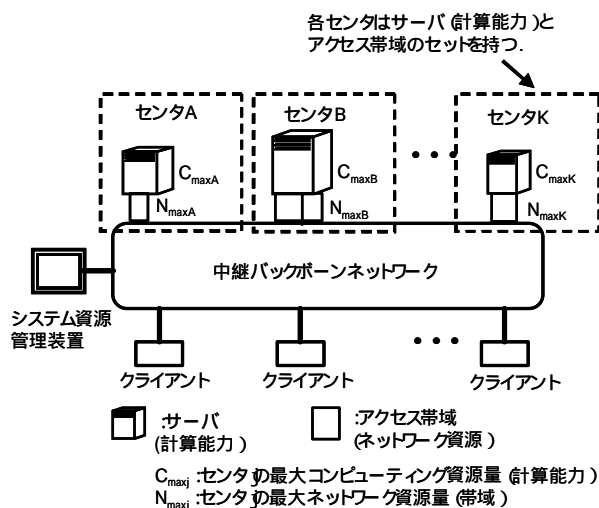
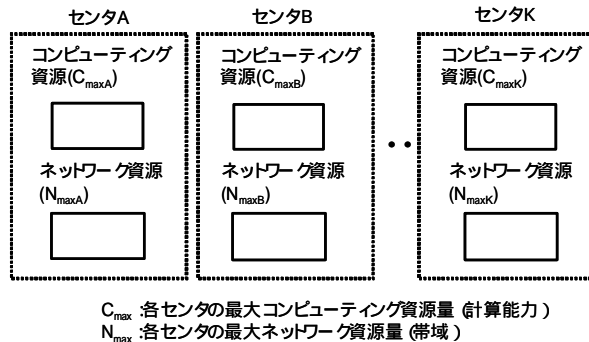


図1．コンピュータネットワークの構成

図2は図1を前提とした資源割当てモデルである。K個の異なるセンタが存在し、各センタは計算能力（コンピューティング資源）とアクセス帯域（ネットワーク資源）の「資源セット」を持つ。要求発生時に、K個の中から最適なセンタを1つ選択し、その選択したセンタからコンピューティング資源とネットワーク資源を“同時に”その要求に割当てて。コンピューティング資源とネ

ネットワーク資源を別々のセンタから割当ててはできない。また、割当てられた資源は各要求専用であり、他要求と共有されない。クライアントは割当てられた専用資源を用いて、ジョブ処理やアプリケーションサービスを実行する。



\* 各要求毎にK個のセンタから1つのセンタを選択し、要求されたコンピューティング資源ならびにネットワーク資源を同時にそのセンタから割当てる。

図2. コンピュータネットワークの資源割当てモデル

### 3. 複数資源同時割当ての課題

一般に、要求される計算能力と帯域の大きさは同じではなく、かつ要求毎に大きさも異なる。このため、従来提案されている1つの資源種別を前提とした資源割当て方式<sup>(5)~(9)</sup>を2章で述べた複数資源同時割当てにそのまま適用しても最適な割当ては期待できない。以下にセンタ数2を前提に例を説明する。

<例1> 図3は2つのセンタの資源使用量をできるだけ均等化するようにセンタ選択を行うケースである。まず、コンピューティング資源だけに注目した資源割当て例を図3(1)に示す。図3(1)において、要求<sub>i</sub>はまずセンタAを選択する(センタA, センタBのどちらでも同じ条件で選択可能であり、たまたまセンタAを選択)。次に、要求<sub>i</sub>はコンピューティング資源の空きが多いセンタBを選択し、要求<sub>j</sub>はコンピューティング資源の空きが多いセンタAを選択する( $C_{maxA} < C_{maxB}$ の大きさ)。

だから、その後、要求<sub>j</sub>はコンピューティング資源の空きが多いセンタBを選択する( $C_{maxA} > C_{maxB}$ の大きさ)。この時点で、コンピューティング資源に着目すると両方のセンタの使用量はほぼ均等になっているが、一方のネットワーク資源に着目すると全く使用量は均等になっていない。これは、コンピューティング資源だけに着目すると要求<sub>i</sub>はセンタAを選択するが、ネットワーク資源に着目すると $N_{maxA} > N_{maxB}$ の大きさのため要求<sub>j</sub>はセンタBを選択すべきであったと言える。これにより、両センタのネットワーク資源使用量の均等化は崩れた。

ところで、要求<sub>i</sub>, 要求<sub>j</sub>, 要求<sub>k</sub>, 要求<sub>l</sub>が使用するコンピューティング資源の大きさに大きな差は無い。そのため、センタAとセンタBを選択する要求を入れ替えてもコンピューティング資源の均等化は維持される。一方、要求<sub>i</sub>と要求<sub>j</sub>が使用するネットワーク資源の大きさは、要求<sub>i</sub>と要求<sub>j</sub>が使用するネットワーク資源の大きさに比べて相当大きい。このため、要求<sub>i</sub>と要求<sub>j</sub>を入れ替えることで、両センタのネットワーク資源使用量は均等に近くなる。ここで提案したセンタ選択を行った例を図3(2)に示す。図3(1)に比べ、コンピューティング資源ならびにネットワーク資源ともに両センタの資源量は均等化されていることがわかる。つまり、1つの資源種別だけでなく、2つの資源種別を同時に考慮することにより両センタの資源使用量を均等化することが可能である。

<例2> 図4は極力一方のセンタに詰め込むことによって、その後発生するより大きな要求にできるだけ対応できるようにセンタ選択を行うケースである。まず、コンピューティング資源だけに注目した資源割当て例を図4(1)に示す。図4(1)において、要求<sub>i</sub>はまずセンタAを選択する(センタA, センタBのどちらでも同じ条件で選択可能であり、たまたまセンタAを選択)。次に、要求<sub>j</sub>はコンピューティング資源の空きが少ないセンタ

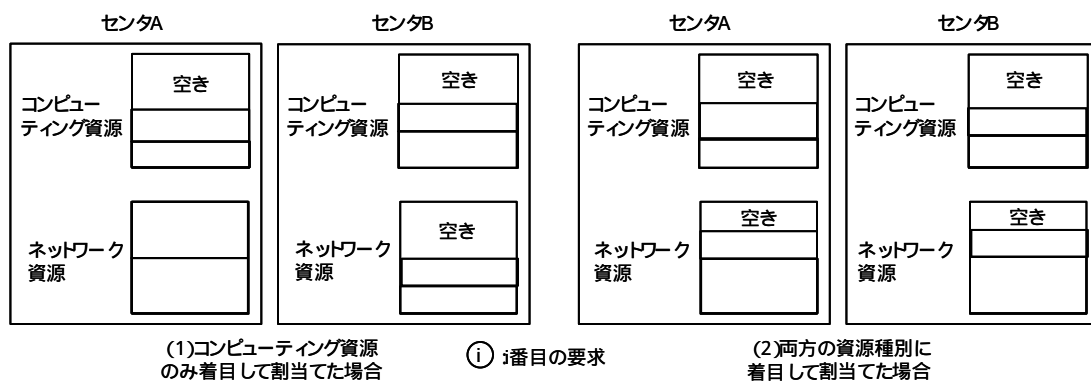


図3 複数資源同時割当ての課題例(両センタの使用量を均等化する場合)

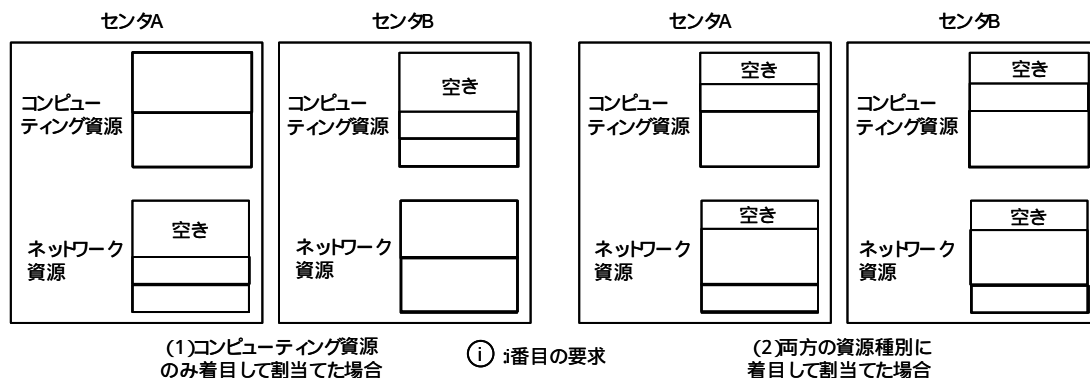


図4 複数資源同時割当ての課題例（極力片方のセンタに詰め込み，一方のセンタを空けておく場合）

A を選択し、要求  $i$  はセンタ A のコンピューティング資源の空きがなくなったため、センタ B を選択する（ $\{ \text{の大きさ} + \text{の大きさ} \} = \text{総資源量}$  だから）。その後、要求  $i$  もセンタ A のコンピューティング資源に空きがないためセンタ B を選択する。この時点で、コンピューティング資源に着目するとセンタ A に詰め込まれているが、ネットワーク資源に着目するとセンタ A ではなくセンタ B に詰め込まれ、かつコンピューティング資源とネットワーク資源それぞれに空きはあるにも関わらず同じセンタで両方の資源を確保できない“虫食い状態”が発生している。これは、要求  $i$  発生時にセンタ A のコンピューティング資源に空きがないためセンタ B を選択したが、ネットワーク資源に着目すると要求  $i$  はセンタ A を選択すべきであった。これにより、両資源ともにセンタ A に詰め込むことができなくなった。

ところで、コンピューティング資源だけみると、要求  $i$  と要求  $j$  でセンタ A のコンピューティング資源は全て使用されるが、センタ A のネットワーク資源には空きがある。一方、ネットワーク資源だけみると、要求  $i$  と要求  $j$  でセンタ B のネットワーク資源は全て使用されるが、センタ B のコンピューティング資源には空きがある。このため、要求  $i$  と要求  $j$  を入れ替えることで“虫食い状態”を回避できる。これを図4(2)に示す。つまり、1つの資源種別だけでなく、2つの資源種別を同時に考慮することにより2つの資源種別共に一方のセンタに詰め込むことが可能となる。

このように、複数種別の資源を同時に割当てる場合には、片方の資源種別だけでなく両方の資源種別を同時に考慮した割当て方式を検討する必要がある。これを4章で提案する。また、複数種別の資源を同時に割当てる方式を評価する場合、資源属性についても考慮する必要がある。例えば、ネットワーク資源属性としては、ネットワーク遅延、実行スループット（ビット誤り率やパケッ

ト廃棄率などに影響される）などがある。例えば、クライアントとサーバ間の応答遅延時間を100m秒以下のクイック応答を必要とする要求に対しては、ネットワーク資源（帯域）に空きがあっても100m秒以上のネットワーク遅延がかかるものは選択できない。このように、仮に資源に空きがあっても、その資源属性から資源割当て条件を満足できないために要求棄却となるケースを極力削減できる方式が必要である。これを5章で提案する。

さらに、要求棄却率を低下させる手段として、1つのセンタで処理しきれない場合には複数のセンタで処理する（要求分割）、または割当て資源を移し変えることで新たな要求を処理する（資源移し変え）、などが考えられ、6章で提案する。これらはコンピューティング資源のみが対象である。

#### 4．複数資源同時割当て

本章では、計算能力（コンピューティング資源）とアクセス帯域（ネットワーク資源）の資源セットを持つ複数のセンタの中からセンタを1つ選択し、その選択したセンタからコンピューティング資源とネットワーク資源を同時に割当ててことを前提とした最適資源割当て方式を提案する<sup>(10),(11)</sup>。

##### 4.1 資源割当てアルゴリズム

まず、比較のため従来方式の1つであるシーケンシャル方式<sup>(8)</sup>の割当てアルゴリズムを以下に定義する。

##### <シーケンシャル方式>（方式）

###### (i) センタ選択

要求発生毎に、K 個のセンタの中から順番に（ラウンドロビン）センタを1つ選択する。もし、選択したセンタに十分な資源の空きがなければ次のセンタを選択する。全てのセンタに十分な空きがない場合にはその要求を棄

却する。なお、要求発生毎に最初に選択するセンタは、前の要求で最終的にどのセンタが選択されたかに関わらず、決まった順番に選ぶ。

## (ii) 資源割当て

(i)で選択されたセンタのコンピュータ資源とネットワーク資源を同時に要求に割当てる。

## (iii) 資源の解放

資源割当て時間が終了した段階で割当て要求を解放する。

次に、新たに提案する複数資源同時割当て方式の割当てアルゴリズムを以下に示す。これは、できるだけ込み合っているセンタを選択することにより、今後発生する大きな資源量を必要とする要求にできるだけ対応できるようにするのが狙いである。

### <方式>

#### (i) キー資源の特定

センタの選択において、2つの資源種別の中で注目する資源種別を「キー資源」と定義する。もし、 $X_C > X_N$  ならコンピューティング資源をキー資源と判断する。そうでない場合にはネットワーク資源をキー資源と判断する。また、コンピューティング資源とネットワーク資源の単位はMIPSとb/s（ビット/秒）で異なる。このため、必要量/総資源量 という相対値を用いて両者の大きさを比較する。

ここで、 $X_C = \{ \text{要求が必要とするコンピューティング}$

資源量の大きさ} /  $X_{C0}$

$X_{C0}$ ：全てのセンタの中で最も総コンピューティング資源量(割当て可能な最大コンピューティング資源量)が小さいセンタの総コンピューティング資源量

$X_N = \{ \text{要求が必要とするネットワーク資源量の大きさ} \} / X_{N0}$  ,

$X_{N0}$ ：全てのセンタの中で最も総ネットワーク資源量(割当て可能な最大ネットワーク資源量)が小さいセンタの総ネットワーク資源量

例えば、総コンピューティング資源量が50と100のセンタが存在した場合、 $X_{C0}$ は50となる。

#### (ii) センタ選択

(i)で決定したキー資源に着目し、空き資源量が最も少ないセンタを選択する。但し、コンピューティング資源ならびにネットワーク資源を必要量確保できることが条件である。また、条件を満足するセンタが複数存在する場合には、その中からランダムにセンタを1つ選択する。なお、条件を満足するセンタが1つも存在しない場合には要求を棄却する。

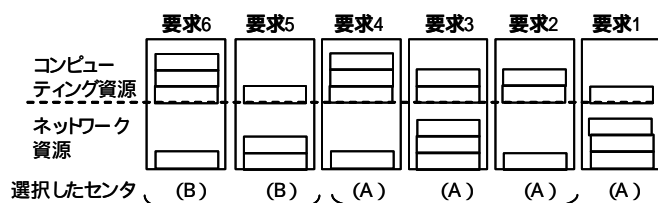
#### (iii) 資源割当て

(ii)で選択されたセンタのコンピュータ資源とネットワーク資源を同時に要求に割当てる。

#### (iv) 資源の解放

資源割当て時間が終了した段階で割当て要求を解放

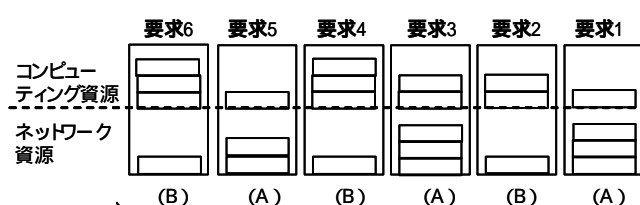
#### 方式で割当てた場合



センタBしか空いていないのでセンタBを選択  
センタAのほうが着目した資源の空きが少ないためセンタAを選択

#### <参考>

#### 方式で同じ要求を割当てた場合



センタを順番に選択

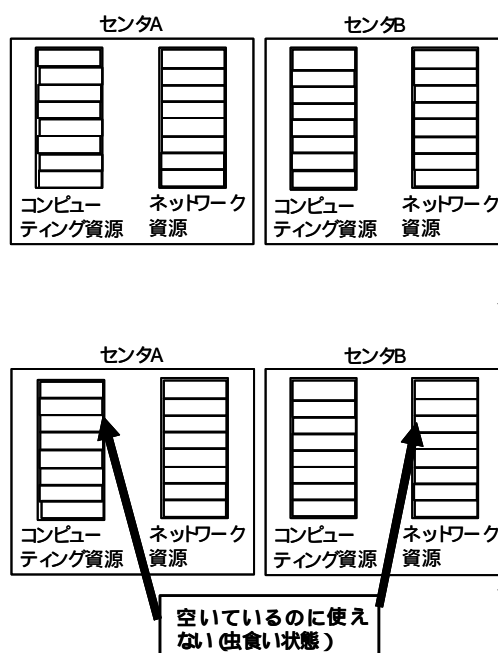


図5. 方式の資源割当てイメージ

する。

方式 を前提としたコンピューティング資源とネットワーク資源の同時割当てイメージを図5に示す。この図からわかるように、方式 では各センタの資源が虫食い状態となるのに比べ、方式 は虫食い状態にならず、より多くの要求を処理できることがわかる。4.2節では、どの範囲でどの程度方式 が有効になるかシミュレーション評価を行う。

## 4.2 シミュレーション評価

### 4.2.1 シミュレーション条件

- 1) 評価モデルは、図2の資源割当てモデルでセンタ数が2の場合を前提とする。つまり、ロケーションが異なる2つのセンタA、Bが存在し、それぞれのセンタにはコンピューティング資源とそれにアクセスするための帯域（ネットワーク資源）が存在する。要求発生時に、資源割当てアルゴリズムに従い、センタAまたはセンタBを選択する。
- 2) C言語を用いたコンピュータシミュレーションにより評価を行う。
- 3) 要求毎に必要なコンピューティング資源量とネットワーク資源量の大きさはガウス分布に従うものとする。また、それぞれの大きさの平均値をC、Nで示す。
- 4) センタA、センタBのコンピューティング資源の総量（最大使用可能な資源量）をそれぞれ $C_{\max A}$ 、 $C_{\max B}$ とする。センタA、センタBのネットワーク資源の総量をそれぞれ $N_{\max A}$ 、 $N_{\max B}$ とする。

5) 要求の発生間隔は平均rの指数分布に従う。資源割当て時間Hは一定とする。割当てられた資源は資源割当て時間経過後に解放される。

6) 要求の発生パターンは「 $\{C = i_1, N = j_1; C = i_2, N = j_2; \dots; C = i_n, N = j_n\}$ の繰り返し」のように記述する。ここで、n個は要求の数、 $i_k (k=1 \sim n)$ はk番目の要求のCの大きさ、 $j_k (k=1 \sim n)$ はk番目の要求のNの大きさ、をそれぞれ表す。また、今回の評価では以下の3つの基本発生パターン（図6）を用いて、提案方式の基本特性を明らかにする。

・発生パターン 1:  $\{C = x, N = x\}$

CとNの大きさが同じパターン

・発生パターン 2:  $\{C = 2x, N = 2; C = 2, N = 2x\}$

CとNの大きさが交互に凸凹するパターン（

は凸凹の度合いを評価するための係数）

・発生パターン 3:  $\{(C = 3, N = 3) \times 5\}$

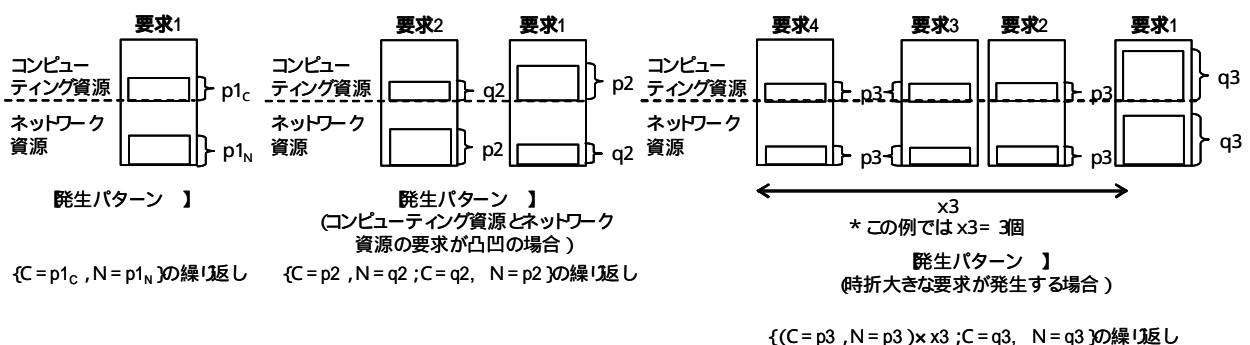
$C = 3x, N = 3x\}$

CとNの大きさが同じで、時折大きい要求が発生するパターン（は大きさの度合いを評価するための係数）

### 4.2.2 シミュレーション評価と分析

図7および図8に示すシミュレーション結果より、以下の諸点が明らかである。

- 1) 発生パターン 1では方式 1と方式 2の差が無いことがわかる（図7＜パターン 1＞）。これは、方式 1の処理できる要求の数と、方式 2の処理できる要求の数が同じだからである。



\* 図は発生間隔に従って発生する要求の資源量を右から順番に並べたものである  
 (要求が連続発生するわけではない)。  
 ・発生パターン 1 ~ 3 は図示されているパターンが繰り返し発生するものとする。

図6 要求発生パターン

2) 発生パターン では方式 の方が方式 よりも要求棄却率が小さくなることわかる(図7 <パターン>)。これは、方式 は図5に示したような虫食い状態が起こりやすいのに対して、方式 は1つのセンタにできるだけ詰め込むので虫食い状態になりにくいと考えられる。また、方式 の有効性は比率 に関係することがわかる。

さらに、図8のセンタ数が2のときに、方式 で方式 と同じ要求棄却率を維持するように総資源量を削減していったところ、方式 は最大で10%程度総資源量を削減できる。

3) 発生パターン では方式 のほうが方式 よりも要求棄却率が小さくなることわかる(図7 <パターン>)。方式 は時折発生する大きい要求を処理できるからである。これは、一つのセンタに詰め込むことによって、できるだけもう一方のセンタを空けておくからである。また、方式 の有効性は比率 が大きいほど有効であることがわかる。

4) 奇数の場合を除いて、センタの数を増加させても上記の2)は成立する(図8)。センタ数が奇数の場合、発生パターンが であると方式 を用いても図5に示したような虫食い状態が発生しやすくなるためである。

以上から、方式 は要求発生パターンが または の場合に従来方式に比べ有利となり、最大で10%の総資源量を削減することが可能である。

## 5 ネットワーク遅延を考慮した複数資源同時割当て方式

本章では、ネットワーク資源の資源属性であるネットワーク遅延を考慮した複数資源同時割当て方式(方式 )を提案する<sup>(11),(12)</sup>。クライアントは、要求時にコンピューティング資源とネットワーク資源の大きさ、資源割当て時間だけでなく、許容最大ネットワーク遅延も指定することを前提とする。例えば、要求によってはクイック応答を必要とするものもあり、その場合には資源に空きがあってもネットワーク遅延の大きなセンタは選択できない。提案する方式の狙いは、できるだけネットワーク遅延の大きなセンタを選択し(但し、許容最大ネットワーク遅延以下が条件)、今後発生する許容最大ネットワーク遅延の小さな要求にできるだけ多く対応できるようにすることである。つまり、方式 では方式 の様に要求資源の大きさに着目するのではなく、ネットワーク遅延に着目するのが基本的な違いである。

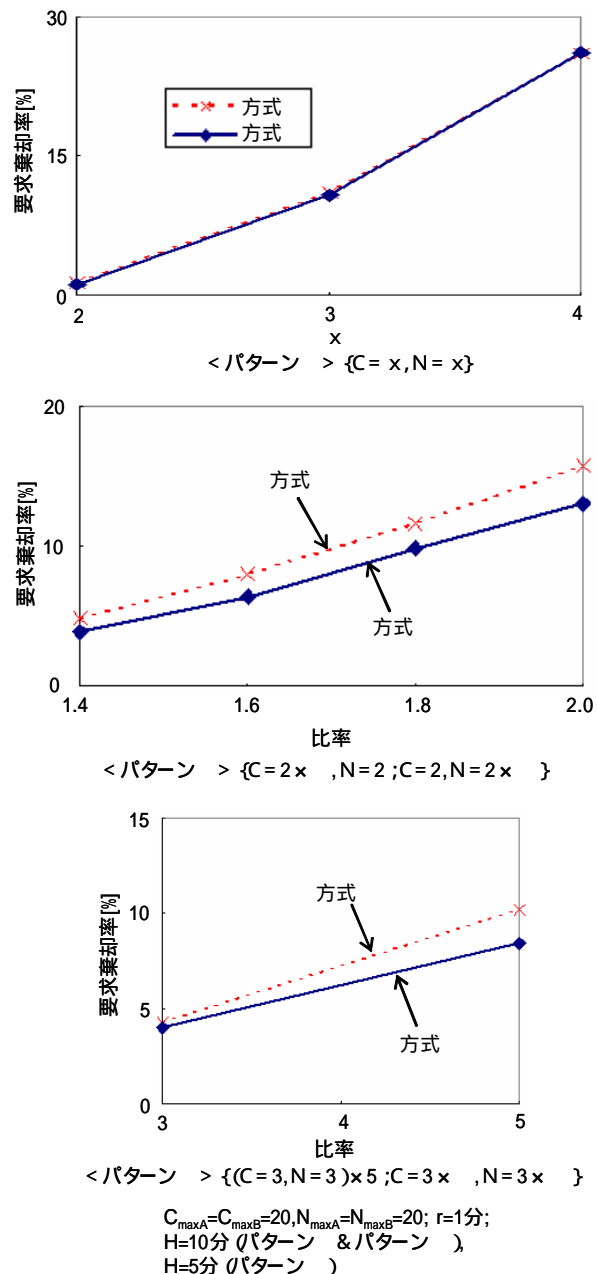
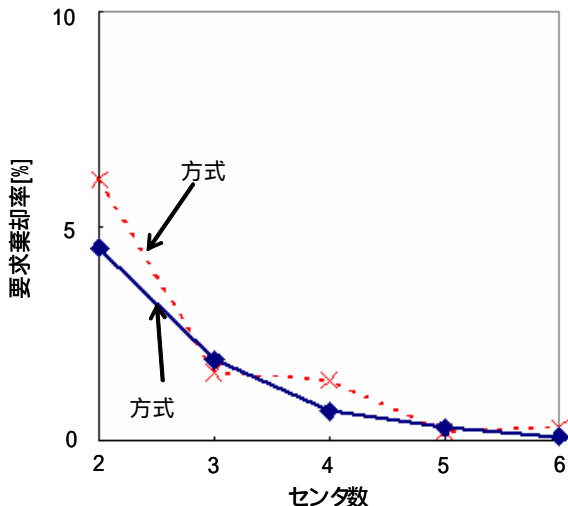


図7 シミュレーション結果(方式 vs. 方式)

### 5.1 方式 の資源割当てアルゴリズム

#### (i) センタ選択

ネットワーク遅延の最も大きなセンタを選択する。但し、ネットワーク遅延は  $L$  以下であり、かつコンピューティング資源ならびにネットワーク資源を必要量確保できることが条件である。また、条件を満足するセンタが複数存在する場合には、方式 と同様に相対的に大きい要求の資源種別に着目して、空きが最も少ないセンタを選択する。なお、条件に合うセンタが存在しない場合には要求を棄却する。



<パターン> {C=4, N=2; C=2, N=4}

$C_{\max}=N_{\max}=20$  (全センタ);  $r=1.4$ 分 (センタ数2),  
 $r=0.93$ 分 (センタ数3),  $r=0.7$ 分 (センタ数4),  
 $r=0.56$ 分 (センタ数5),  $r=0.47$ 分 (センタ数6);  
 $H=10$ 分

図8 シミュレーション結果 (センタ数増加の影響)

## (ii) 資源割当て

(i) で選択されたセンタのコンピュータ資源とネットワーク資源を同時に要求に割当てる。

## (iii) 資源の解放

資源割当て時間が終了した段階で割当て要求を解放する。

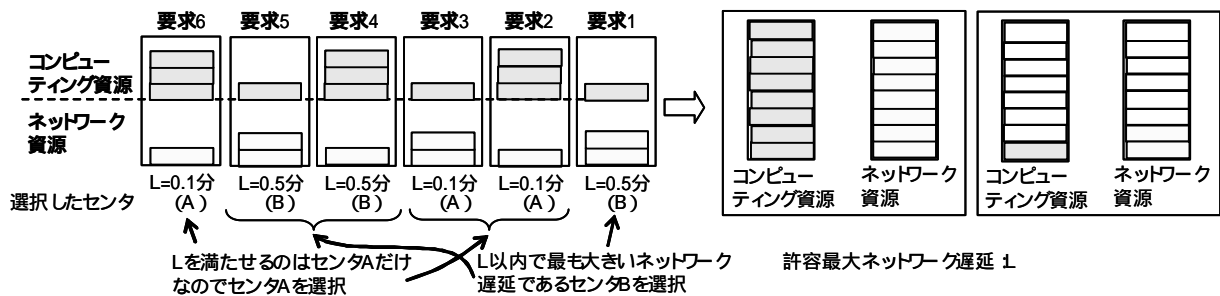
方式の資源割当てイメージを図9に示す。この例では、方式はネットワーク遅延の小さなセンタAに詰め込むように働くため、Lの小さな要求は棄却されやすくなるのに対し、方式はネットワーク遅延の大きなセンタBを極力選択するためLの小さな要求にも対応しやすくなるのがわかる。

## 5.2 シミュレーション評価

### 5.2.1 シミュレーション条件

- 1) 基本的には4.2.1と同じ条件とする。
- 2) 評価モデルは、図2の資源割当てモデルに各センタそれぞれのネットワーク遅延を追加して、センタ数が2の場合を前提とする。センタA、センタBのネットワーク遅延をそれぞれ $t_A=0.1$ 分、 $t_B=0.5$ 分とする。
- 3) 要求の許容最大ネットワーク遅延Lは0.1分もしくは0.5分とし、0.1分の要求が発生する割合をPとする。

方式で割当てた場合



<参考>

方式で同じ要求を割当てた場合

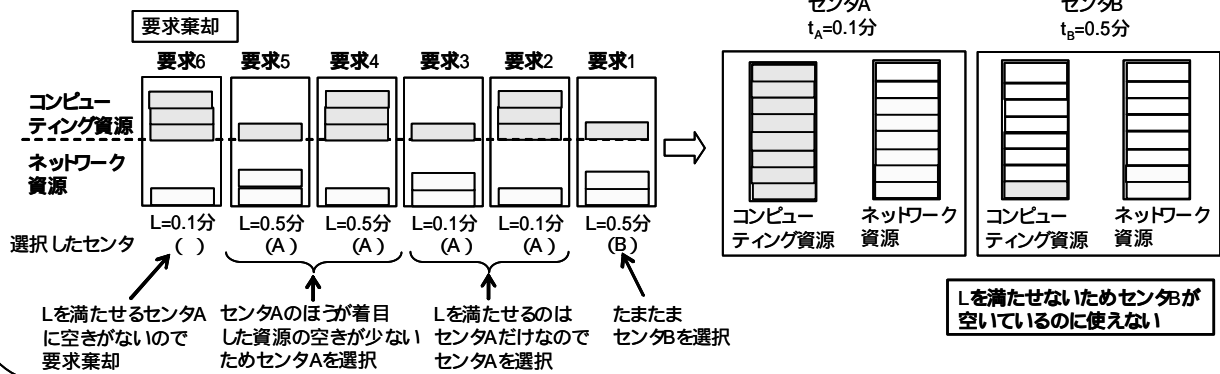


図9. 方式の資源割当てイメージ



### 5.2.2 シミュレーション評価と分析

図 10 に示すシミュレーション結果より、以下の諸点が明らかである。なお、4 章において方式 1 が方式 2 よりも有利であることが明らかになったことから、ここでは方式 1 と方式 2 の比較を行う。

1) 要求発生パターンに関係なく、方式 1 が有利となる。つまり、P が 0 % または 100 % 近辺以外であれば、方式 1 の方が方式 2 よりも要求棄却率が小さく、かつより多くの要求を処理できる。例えば、図 10 < パターン 1 > の場合には、 $P=50\%$  では方式 1 はおよそ  $r=5$  分で要求棄却が発生するが、方式 2 は  $r=2$  分以下にならないと発生しない。

これは、方式 1 では、割当ての途中でセンタ A の方が詰まった状態になると、次の要求を混み合っているセンタ A に詰め込んでしまい、後で発生する許容最大ネットワーク遅延が 0.1 分の要求を処理できなくなってしまう場合があるからと考えられる。一方、方式 2 では、センタ A が混み合っているにもかかわらず許容最大ネットワーク遅延が 0.5 分であればセンタ B を選択する。このため、センタ A の資源を空けておくことができ、後で発生する許容最大ネットワーク遅延が 0.1 分の要求が要求棄却されにくい。

2) 図 10 < パターン 1 > の  $P=67\%$  のときに、方式 1 で方式 2 と同じ要求棄却率を維持するように総資源量を削減していったところ、方式 1 は最大で 15 % 程度総資源量を削減できる。

以上から、方式 1 は要求発生パターンに関係なく方式 2 ならびに従来方式に比べ有利であり、最大で 15 % の総資源量を削減することが可能である。これは、センタ選択において、ネットワーク遅延が一番大きな制約になるからと言える。なお、方式 1 はネットワーク遅延を考慮しない場合には適用できないため、ネットワーク遅延を考慮する必要のない資源割当てにおいては 4 章で提案した方式 2 が最適と言える。

## 6. 要求分割または資源移し変えを考慮した複数資源同時割当て方式

### 6.1 要求分割を考慮した複数資源同時割当て方式 (方式 3) <sup>(10)</sup>

1 つのセンタでは必要なコンピューティング資源を確保できない場合には、要求を分割し複数のセンタの資源を割当てることにより要求棄却を減らす方策が考えられる (図 11)。但し、1 つのセンタで必要な資源を確保し

ないと実行できない要求には適用できない。また、本章では 1 つの要求に対して複数のセンタを使って処理を実行する場合でも、各センタに要求されたネットワーク資源を同じように確保するものとする。

以下に、割当てアルゴリズムを示す。

(1) 必要な資源を確保できるセンタが 1 つ以上存在する場合：4 章で提案した方式 2 を提供する。

(2) 必要な資源を確保できるセンタが存在しない場合：

i) センタ選択 & 資源割当て

全てのセンタの空きコンピューティング資源量の総和が要求されたコンピューティング資源量 ( $z$ ) 以上かどうかチェックし、真なら  $z$  の処理を行う。偽なら要求は棄却する。

コンピューティング資源の空きが最も小さいセンタにまず着目し (新たな要求に必要なネットワーク資源が確保できることが条件)、そのセンタの空き資源量全てまたはまだ不足している分 ( $z_1$ ) だけ、新たな要求に割り当てる。ネットワーク資源については、コンピューティング資源を割当てた全てのセンタに対して、要求が必要とする資源量を割当てる。

$(z - z_1) = 0$  になるまで、 $z$  の処理を繰り返す。 $z$  の処理を繰り返しても条件を満足するセンタが存在しない場合には、要求は棄却する。

ii) 資源の解放

資源割当て時間が終了した段階で割当てた全ての資源を解放する。

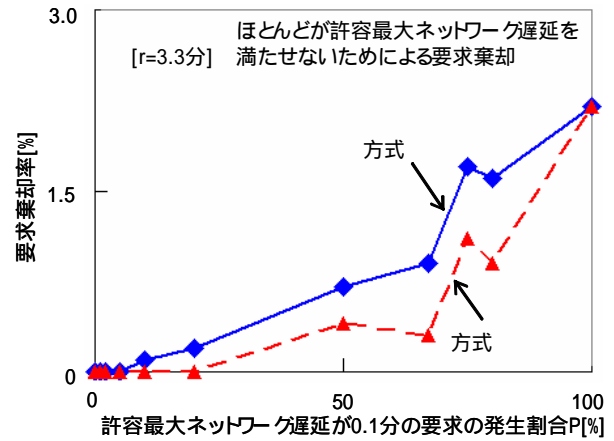
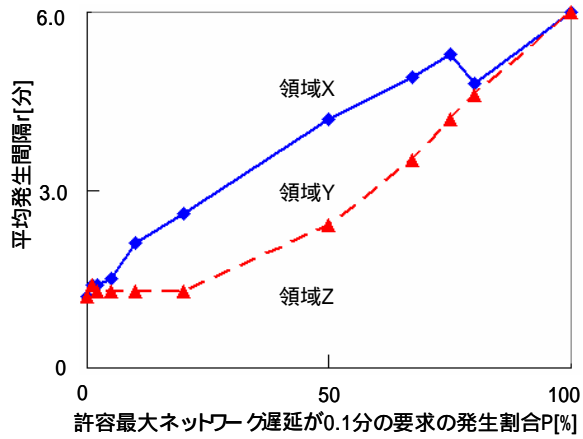
### 6.2 資源移し変えを考慮した複数資源同時割当て方式 (方式 4) <sup>(10)</sup>

特定センタの割当て済のコンピューティング資源の一部を他センタに移し変えることにより、そのセンタの空き資源量を拡大することで大きな資源量を必要とする要求を受け付けられるようにすることが考えられる (図 12)。

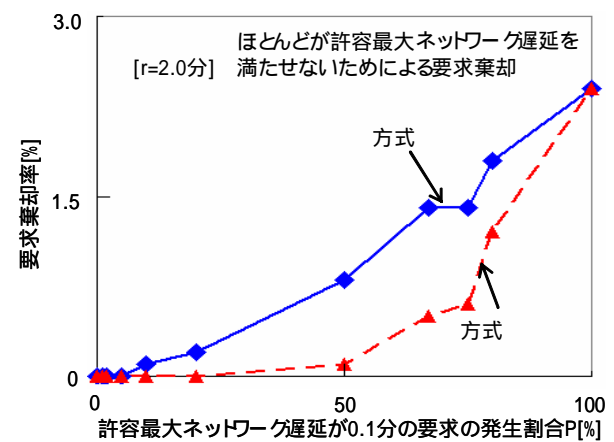
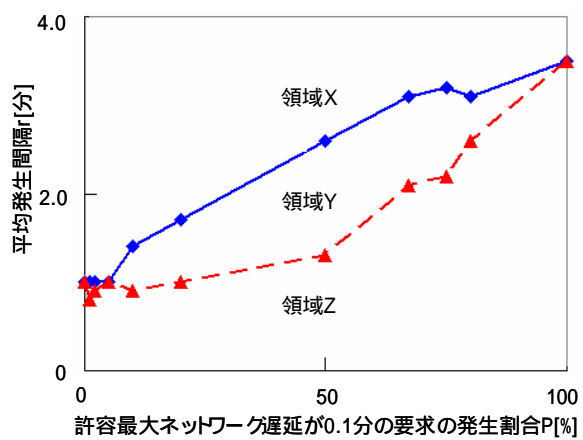
但し、このためには処理中のプロセスを無中断で移動する (プロセス移動) ことが前提となり、それに対応できない要求は対象外である。

以下に、割当てアルゴリズムを示す。

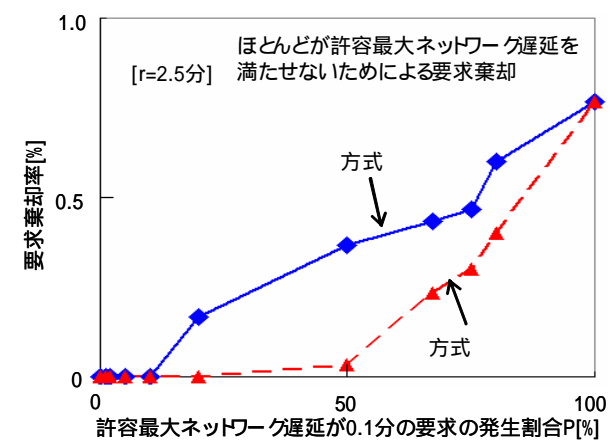
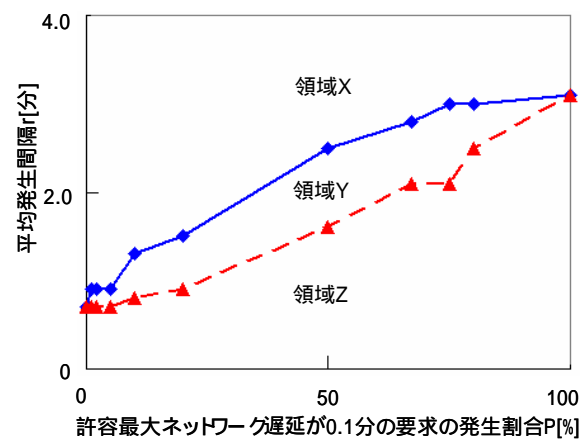
(1) 必要な資源を確保できるセンタが 1 つ以上存在する場合：4 章で提案した方式 2 を提供する。



<パターン> {C=4, N=4}



<パターン> {C=4, N=2; C=2, N=4}



<パターン> {(C=2, N=2) × 3; C=6, N=6}

$C_{\max A} = C_{\max B} = 20, N_{\max A} = N_{\max B} = 20; H = 4$ 分  
 領域X: 方式 と方式 ともに要求棄却が発生する  
 領域Y: 方式 のみ要求棄却が発生する  
 領域Z: 方式 と方式 ともに要求棄却が発生しない

図10 シミュレーション結果 (方式 vs. 方式)

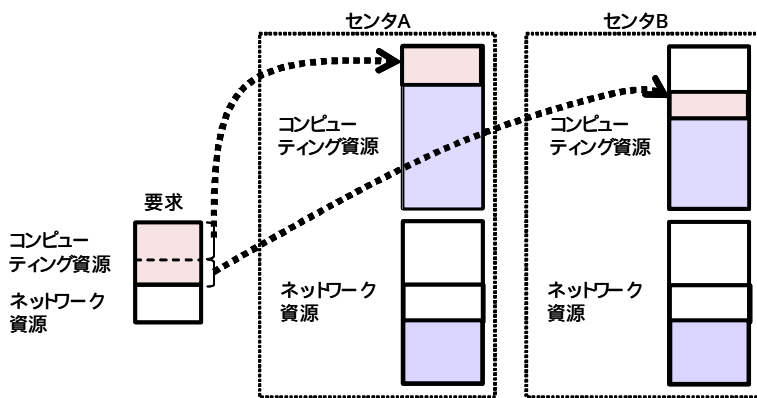


図11．要求分割（方式Ⅰ）の資源割当てイメージ

(2) 必要な資源を確保できるセンタが存在しない場合：

i) センタ選択 & 資源割当て

コンピューティング資源の空きが最も大きいセンタにまず着目し（新たな要求に必要なネットワーク資源が確保できることが条件）、そのセンタで割当て済の資源を新たな要求に必要な分だけ他センタに移動させることができるかどうかチェックする。

移動先センタはコンピューティング資源の空きが次に大きくかつ移動分の空きが存在しているものを選択する。この際、ネットワーク資源も同時に移動する必要があるため、ネットワーク資源にも空きがあることが条件となる。

条件を満たすセンタがあれば、移動させた資源（コンピューティング資源とネットワーク資源の両方）をそのセンタから割当て直す。新たな要求に対しては、割当て済の資源を移動させ資源に空きができたセンタから必要なコンピューティング資源を割当てる。

もし、移動できるセンタがなければ、コンピューティング資源の空きがその次に大きいセンタに着目して同じチェックを行う。

全てのセンタについてチェックしても条件を満たすセンタがない場合には、新たな要求は要求棄却とする。

ii) 資源の解放

資源割当て時間が終了した段階で割当て資源を解放する。

## 6.3 シミュレーション評価

### 6.3.1 シミュレーション条件

方式Ⅳならびに方式Ⅰの大まかな効果を評価するため、方式Ⅰ、方式Ⅱに比べ簡単なシミュレーションを行う。また、方式Ⅰと比較することにより、方式Ⅰならびに方式Ⅴの効果を評価する。

- 1) 基本的には4.2.1に合わせる。
- 2) 要求毎に必要なとなるコンピューティング資源量とネットワーク資源量の大きさは一定とし、それぞれC、Nで示す。
- 3) 要求の発生間隔 $r$ は一定とする。
- 4) 方式Ⅰは要求分割が起こりやすいと想定される以下の発生パターンでのみ評価する。  
{ C=6, N=3 ; C=3, N=6 } ( CとNの大きさが交互に凸凹するパターン )
- 5) 方式Ⅱは資源移し変えの起こりやすいと想定される以下の発生パターンのみで評価する。  
{ C=4, N=2 ; C=3, N=2 ; C=2, N=2 ; C=3, N=2 }

### 6.3.2 シミュレーション評価と分析

表1、表2に示すシミュレーション結果より、要求分

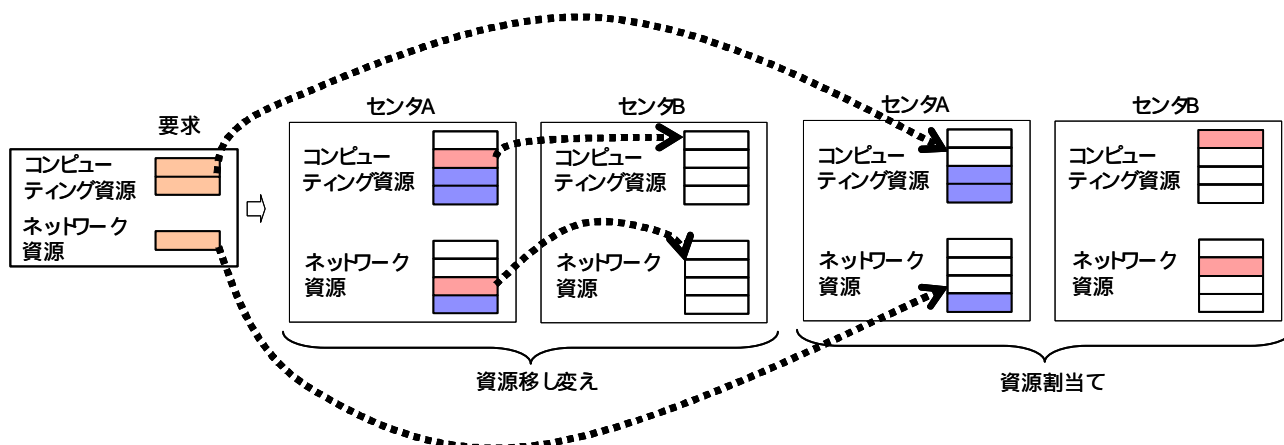


図12．資源移し変え（方式Ⅱ）の資源割当てイメージ

割または資源移し変えを行うことにより要求棄却率は減少するが、その効果はわずかであることがわかる。処理の複雑性を考えると、方式 または方式 は方式 に比べて有利とは必ずしも言えない。

表 1 方式 と方式 の比較結果

	分割を考慮しない (方式 )	分割を考慮する (方式 )
要求棄却率	11%	10%

<条件> 要求パターン {C=6, N=3 ; C=3, N=6 } (C, Nは一定)  
 $C_{\max A}=C_{\max B}=21, N_{\max A}=N_{\max B}=21$ ; r=1分; H=9分

表 2 方式 と方式 の比較結果

	資源移し変えを 考慮しない(方式 )	資源移し変えを 考慮する(方式 )
要求棄却率	6%	4%

<条件> 要求パターン {C=4, N=2 ; C=3, N=2 ;  
C=2, N=2 ; C=3, N=2 } (C, Nは一定)  
 $C_{\max A}=C_{\max B}=20, N_{\max A}=N_{\max B}=20$ ; r=1分; H=13分

## 7 . 要求棄却率を削減する他の割当て方式

4章から6章で提案した方式 , , , 以外に、表3に示す他の割当て方式が考えられる<sup>(13)</sup>。これらの方式については、今後評価していく予定である。以下に、各方式の概要を説明する。

<方式 >

割当て可能な資源がない場合には、資源割当て開始を遅らせることにより要求を受け付ける方式。ただし、許容最大“資源割当て終了時刻”までに資源割当てを終了することが条件。処理イメージを図13に示す。なお、実際の処理は2つの資源種別に対して実施されるが、図13は一方の資源種別のみを抽出した例を示す。また、今まで1つの資源種別に着目して資源割当て開始を遅らせる

方式<sup>(14)</sup>はあったが、2つの資源種別に着目して割当て開始を遅らせる方式ではなかった。

<方式 >

割当て可能な資源がない場合には、要求された資源量よりも小さな資源を割当てること（結果として資源割当て時間が指定された長さよりも長くなる）により、同時に割当てる資源量を少なくし、今後発生するより多くの要求に備える方式。但し、ストリーミング配信など要求資源量を変更できない要求には適用できない。処理イメージを図14に示す。なお、図13と同様に一方の資源種別のみを抽出した例を示す。

<方式 >

方式 とは逆に、（要求時に割当て可能な空き資源がない場合に）要求された資源量よりも大きな資源を割当てて処理を早く完了させ、新たに到着する大きな資源量を必要とする要求に備える方式。

<方式 >

方式 と同様に資源割当て開始時刻を遅らせる機能を方式 に追加した方式。

<方式 >

方式 と同様に資源割当て開始時刻を遅らせる機能を方式 に追加した方式。

## 8 . むすび

コンピュータネットワークにおいて、コンピューティング資源（計算能力）とネットワーク資源（帯域）を同時に割当てることを前提とした最適資源割当て方式（方式 , 方式 ）を提案した。シミュレーション評価の結果、提案方式は従来方式に比べ要求棄却率を低下させ、最大で15%の資源を削減することが可能であることを明らかにした。また、資源割当て条件としてネットワー

表 3 複数資源同時割当て方式の分類

条件	対応する提案方式									
< 1つのセンタだけでは要求資源を確保できない場合の対処 > 要求棄却 要求分割 資源移し変え 資源割当て開始遅れ										
< センタ選択におけるネットワーク遅延の考慮 > 考慮しない 考慮する										
< 要求資源量の変更 > 変更しない 変更する - 1 小さくすることを許容 - 2 大きくすることを許容										

注) 方式 以降は方式 との差分のみ表示。 ☐ : 今回評価した方式

ク遅延を考慮しない場合には方式 ，ネットワーク遅延を考慮する場合には方式 ，が最適な方式であることを示した。さらに，要求棄却率を低下させるための方策を網羅的に整理し，今後評価すべき方式を明らかにした。

今後，本稿で評価していない同時割当て方式の有効性

評価を行う。また，今回提案した方式は，上りと下りの両方向の帯域を同時に割当ててを前提とした ATM 網<sup>(15)</sup>，MPLS 網<sup>(16)</sup>ならびに GMPLS 網<sup>(17)</sup>のパス選択にも応用可能であり，その評価も進めていく予定である。

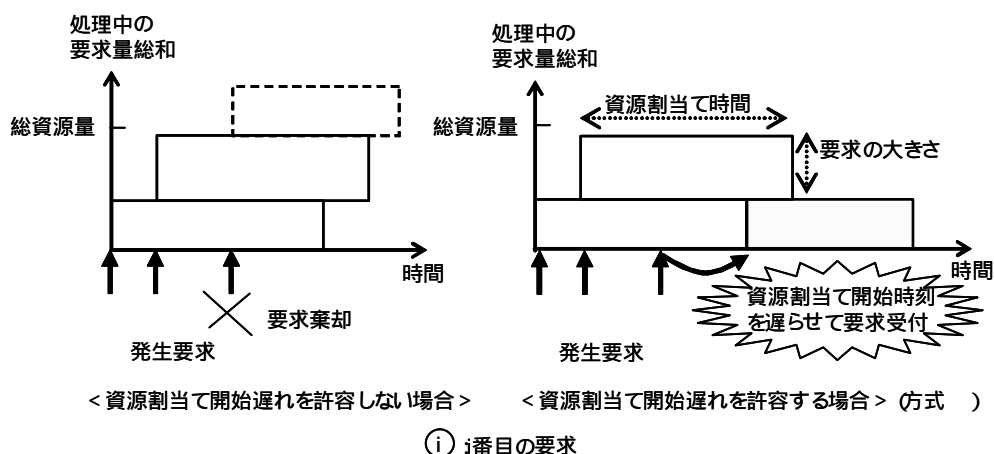


図13 方式 の資源割当てイメージ (一方の資源種別の例のみ示す)

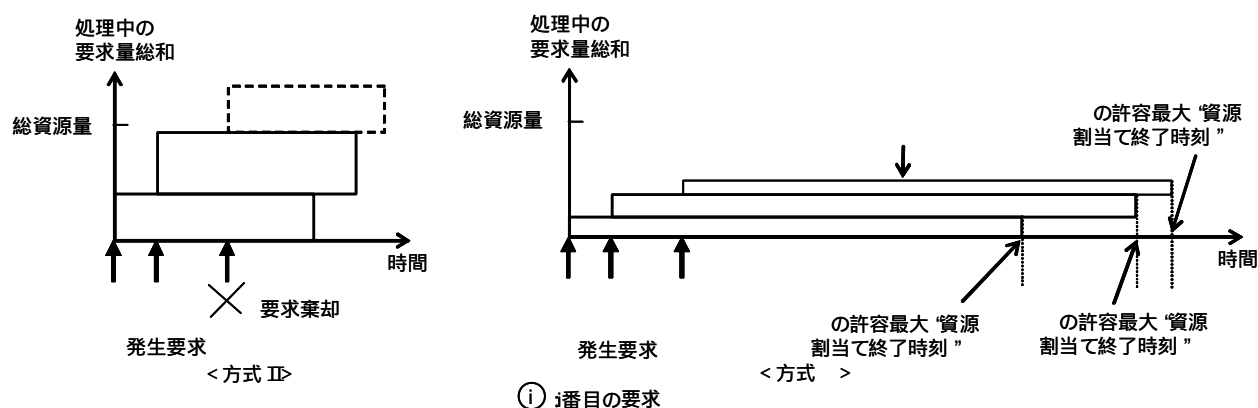


図14 方式 の資源割当てイメージ例 (一方の資源種別の例のみ示す)

## 参考文献

(1) “ULTINA On Demand Platform”

<http://www.japan-telecom.co.jp/english/newsrelease/2006/feb/0209/0209.html>

(2) “The Open Grid Services Architecture, Version 1.5”, Open Grid Forum Document GFD-80.

(3) 平成 18 年度「情報通信白書」総務省

<http://www.johotsusintokei.soumu.go.jp/whitepaper/whitepaper01.html>

(4) A.Takefusa, H.Casanova, S.Matsuoka, and

F.Berman, “A study of deadline scheduling for client-server systems on the computational grid”, Proceedings of the 10<sup>th</sup> IEEE International Symposium on High Performance Distributed Computing.

(5) S.Kuribayashi, “Bandwidth assignment control in ATM system”, Electronics and communications in Japan, Part 1, Vol.74, No.9, pp.30-42, 1991.

(6) T.Takeda, T.Ohba, I.Inoue and S.Urushidani, “Resource Management in Layer 1 Virtual Private Networks”, IEICE Trans., Vol.E88-B, No.8,pp.3343-

3352, 2005.

- (7) S.Shiokawa and S.TAsaka, “Bandwidth allocation considering priorities among multimedia components in mobile networks”, IEICE Trans. Commu., Vol.E84-B, No.5, PP.1344-1355, 2001.
- (8) K.Aida, A.Takefusa, H.Nakada, S.Matsuoka, S.Sekiguchi, and U.Nagashima, “Performance evaluation model for scheduling in global computing systems”, International Journal of High-Performance Computing Application, Vol.14, No.3, pp.268-279, 2000.
- (9) R.Buyya, D.Abramson, and J.Giddy, “Nimrod/G: An architecture for a resource management and scheduling system in a global computational grid”, Proceedings of the HPC ASIA’2000.
- (10) 栗林, 津村, 田邊: “ブロードバンド・ユビキタスネットワークにおける複数種別資源の最適同時割当て方式”, 電子情報通信学会 研究会資料 IN2005-105 (2005.11)
- (11) S.Tsumura and S.Kuribayashi, “Simultaneous allocation of multiple resources for computer communications networks”, APCC2006 (2006.8).
- (12) 津村, 栗林, 田邊: “遅延時間を考慮した複数資源同時割当て方式の有効性評価”, 電子情報通信学会 総合大会 B-7-148 (2006.3)
- (13) 津村, 栗林, 田邊, “サービス開始遅れを許容する複数資源同時割当て方式の提案”, 電子情報通信学会 2006 ソサイエティ大会 B-7-5 (2006.9)
- (14) 池永, 川原, 尾家, “QoS 経路制御における受付遅延の導入とその効果について”, 電子情報通信学会 技術研究報告 IN2001-96, pp45-50 (2001.11)
- (15) T.Ogawa, S.Kuribayashi, A.Senoh, T.Ushiki, S.Shiraishi and T.Sampe, “ATM signaling VC set-up methods and bandwidth selection methods in B-ISDN”, Technical report of IEICE, SSE94-182, 1995.(Japanese)
- (16) IETF RFC3031 “Multiprotocol Label Switching Architecture“
- (17) IETF RFC3945 “Generalized Multi-Protocol Label Switching (GMPLS) Architecture”