分散資源共有での動的複製再配置における通信メッセージ量最適化

布木哲三*・下川俊彦*
楢崎修二**・吉田紀彦**

Communication Optimization in Dynamic Copy Allocation for Distributed Resource Sharing

by
Tetsuzo FUKI*·Toshihiko SHIMOKAWA*·Shuji NARAZAKI**
and Norihiko YOSHIDA**

In open and dynamic environments, where an access pattern to shared resources cannot be predicted statically, or is even changed dynamically, a distributed shared memory system must be adaptive to minimize communication overhead over network. The system must adjust parameters or even switch protocols according to situations which is monitored dynamically.

This paper presents a dynamic adaptation scheme for distributed resource sharing. The system watches the transition of the access pattern (read/write ratio), and accordingly determines the number and allocation of copies (replicas) of the data dynamically and automatically. This paper also presents the system implementation using the meta-level computation framework, so that application programs are never aware of being monitored nor controlled by the system.

1．はじめに

複数のプロセッサを用いて分散環境下で処理を行うには，一般にプロセッサ間で資源の共有が行われる。共有資源の参照・更新を行うときにはプロセッサ間で通信が必要になる。一般に通信コストは計算コストに比べて高いので，共有資源の参照・更新時における通信回数を削減することで効率の良い分散処理を行うことができる。

プロセッサ間で共有する資源の複製を局所的に配置することで参照時における通信回数を削減できる。しかし，更新時には共有資源の一貫性を保持するために通信が必要となる。したがって，参照・更新時における通信回数を最小にするためには複製を最適に配置することが必要である。

これまで通信回数を削減するためにいろいろな資源共有方法が提案されてきた。しかし，共有資源の使用頻度が異なると，もっとも効率の良い方法も異なる。つまり，これまで提案されてきた方法は共有資源の使用頻度の変化に対して対応できない。

各プロセッサの共有資源の使用頻度を基にして共有資源の複製を最適に配置することで，従来の方法より通信回数を減らすことができる。

そこで本研究では，共有資源の使用頻度が動的に変化する場合と，共有資源の使用頻度が事前に判らない・判りにくい場合にも対応できるように，各プロセッサでの共有資源の使用頻度を基に共有資源の複製を最適に配置する方法を提案した[1,2,3]。

また，我々のこれまでの研究ではデータの大きさを

平成11年4月23日受理
*九州大学大学院システム情報科学研究科　福岡市東区箱崎
(Graduate School of Information Science and Electrical Engineering, Kyushu University, Hakozaki, Fukuoka)
**情報システム工学科 (Department of Computer and Information Sciences)

考慮していなかったので，データの大きさを考慮して複製を最適に再配置することで通信回数をより減らすことを考える。

本研究では以下のことを前提条件とする。

- プロセッサ間の通信コストは計算コストに比べて遥かに高い
- 共有資源は強い一貫性の保持が必要
- メッセージの放送機能は使用しないものとする（一対一通信）
- 各プロセッサにおける資源の使用頻度は急激に変化しない（使用頻度の履歴が使える）
- プロセッサや通信路は故障しない
- 共有資源は複製可能なデータである

本論文は，まず2章で共有資源の使用頻度を基にした動的な複製再配置の方法について説明し，3章で，データの大きさも考慮した配置方法について説明する。4章では，本研究で提案した方法で通信回数が減少するかどうかを検証する。最後に，6章でまとめと今後の課題を述べる。

2　複製の動的な再配置

分散環境下での共有資源を管理する方法はいくつか提案されているが，それらの方法は共有資源の使用頻度が変化したときのことを考慮に入れていない。各プロセッサの共有資源の使用頻度は動的に変化する可能性があるので，はじめは複製を所持していた方が通信回数が少なくてすむ場合でも，その後複製を所持しないようにした方が通信回数が少なくてすむ場合もある。

そこで，本研究では各プロセッサの共有資源の使用頻度を基にして複製の配置を動的に決定する方法を提案する。

この章では，この方式の説明を行う。

2.1　動的な再配置方法

2.1.1　基本方針

複製の配置は，共有資源の使用頻度を基にして決定する。共有資源の使用頻度とは，更新と更新の間の参照回数である。ここでは，更新と更新の間を1サイクルと考える。更新頻度が高ければ1サイクルの参照回数は少なくなり，更新頻度が低ければ参照回数は多くなるので更新頻度を考えなくても，1サイクルの参照回数を基にして複製の再配置を決定することができる。

1サイクルの参照回数が多ければ，複製を局所的に所持する方が効率が良い。これは，複製を局所的に所持することで参照に通信が必要なくなるからである。反対に，1サイクルの参照回数が少なければ，複製を局所的に所持しない方が効率が良い。1サイクルの参照回数が少ないということは，更新が頻繁に行われているということなので複製を作らない方が通信回数が減少するからである。

したがって

- 1サイクルの参照回数が多いとき→複製を作る
- 1サイクルの参照回数が少ないとき→複製を作らない

となる。

本研究では，各プロセッサにおける共有資源の1サイクルの参照回数をエージェントと呼ばれる処理主体に動的に検出させて，その1サイクルの参照回数を基に複製の配置を動的に決定する。これらのエージェントを共有資源エージェントと呼ぶことにする（また，単にエージェントと呼ぶこともある）。

また，研究の第一歩として全体を管理する特別なエージェントを置いた方式を考える。この特別なエージェントを管理エージェントと呼ぶ。管理エージェントは，共有資源を常に所持していて，各共有資源エージェントの参照回数を集め複製の再配置を決定する。また，各共有資源エージェントに対して複製を所持させたり，破棄させたりする命令を送信する。

共有資源エージェントは参照を行うとき参照回数を数える。また，更新が起こると1サイクルの参照回数を管理エージェントに集める。

2.1.2　参照方法

共有資源の参照方法を図1に示す。

参照は複製が局所的にある場合は，局所的に参照して，参照回数を数える。このとき，通信は必要としない。

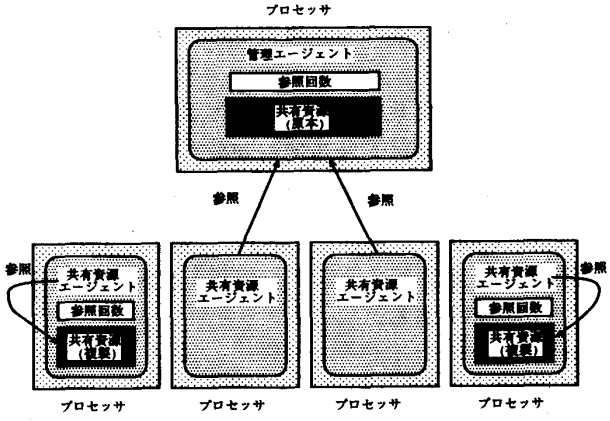

図1．参照方法

複製を局所的に所持していないときは，管理エージェントに対して参照要求を送信する。管理エージェントは参照要求が送信したエージェントに対して共有資源の値を送信し，そのエージェントの参照回数を数える。複製を所持しているエージェントが参照しても通信はしないが,複製を所持していないエージェントは，参照1回につき2回の通信が必要となる。

2.1.3 更新方法

共有資源の更新方法を図2に示す。

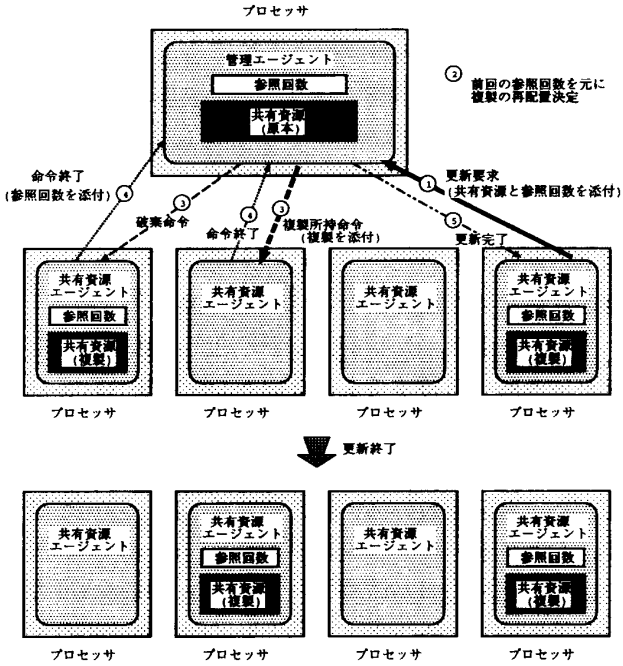

図2．更新方法

更新は管理エージェントに更新要求を送信する。管理エージェントは管理エージェントが所持する共有資源を更新し，各エージェントの参照回数を基に複製の配置を決定する。

複製の配置を決定したら複製を所持させるエージェントに複製を送信し，複製を破棄させるエージェントには破棄命令を送信する。

複製や破棄命令を受信したエージェントは複製の更新や破棄を行い，次回の更新時に必要であるので参照回数を更新終了メッセージに添付して管理エージェントに送信する。

更新を行う前には複製を所持しているエージェントに対してロックをかける必要がある。また，更新が終了するとロックを解除する必要がある。更新時に複製を所持しているエージェントは，ロックで必要な通信だけ余分に通信を行う必要がある。

2.2 複製配置の決定方法

管理エージェントは更新要求を受信した時点では，すべてのエージェントの参照回数を知らない。複製を所持していないエージェントが参照を行うと，管理エージェントに参照要求を送信するので管理エージェントはこれら複製を所持していないエージェントの参照回数は知ることができる。しかし，複製を所持しているエージェントは局所的に参照を行うので管理エージェントとの通信はしない。したがって，複製を所持しているエージェントの参照回数を管理エージェントは更新要求を受信した時点では知らないのである。

管理エージェントがすべてのエージェントの参照回数を知るためには，管理エージェントが更新要求を受信すると，複製を持っているエージェントに対して「参照回数を送れ」という命令を送信すれば良い。こうすることによって，今回の1サイクル間の参照回数を基に複製を再配置することができる。しかし，これでは複製を所持しているエージェント数×2回の通信が余分に必要になり効率が良いとはいえなくなってしまう。

そこで，複製を所持しているエージェントが命令終了を管理エージェントに送信するときに，1サイクルの参照回数を添付して送信することにする。参照回数を添付することによって，更新時に管理エージェントは余分な通信をする必要がなくなる。このようにすると前回の参照回数を基に複製を再配置することになり，各プロセッサにおける参照回数は複製の配置決定に反映されるのが1回遅れることになる。しかし，本研究では，各プロセッサにおける資源の参照頻度は急激に変化しないことを前提にしているので，このことは問題ないと考える。

また，この前提により，本研究では前回の参照回数と前回の更新で使った参照回数との平均を基にして複製の再配置を行うことにする。

2.2.1 システム全体の情報に基づく方法

複製を所持させるエージェントは参照回数が多い順に決めればいいのだが，幾つのエージェントに複製を所持させるべきかを求める方法を説明する。すなわち，最適な複製の数は次のようにして決定する。

参照回数の多い順に各エージェントが並んでいると仮定する。すると，複製の数と参照・更新での通信回数との関係は図3のようになる。縦軸は通信回数であり,横軸は複製を所持させるエージェントの数である。1回の更新に行われる通信回数は複製の数に比例した通信が行われるので，グラフは右上がりの直線になっ

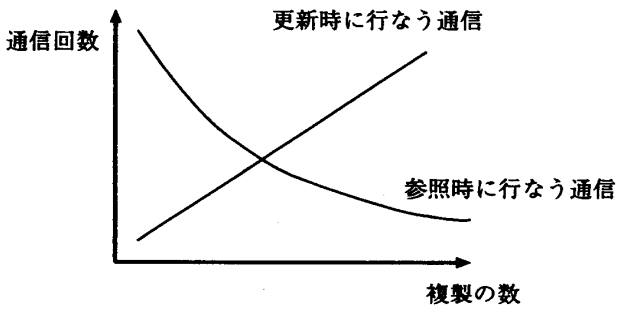

図3．複製所持エージェント数と通信回数

ている。一方，1サイクル間の参照で行われる通信の合計回数は，複製を全く所持しないとき最も多く，すべてのエージェントが複製を所持するとき0になる。また，複製を持たせるエージェントの参照回数に比例して通信回数が減少していくので図3のように曲線で減少していく。(ここでは分かりやすいように曲線で示しているが，通信回数は整数なので階段状に増加したり減少したりする)

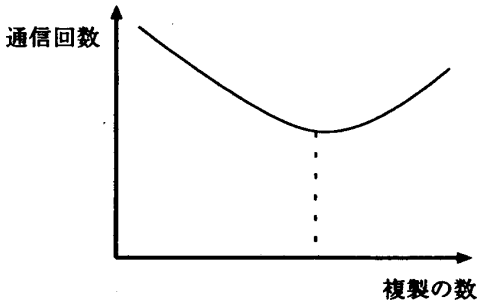

図4．複製所持エージェント数と通信回数

この1回の更新で行われる通信回数と1サイクル間の参照で行われる通信回数の合計を足すと図4のようになる。図4の曲線が最小になるように複製の数を決定することが全体の通信回数を最小にすることになる。すなわち，参照回数の多い順に図4で求めた数のエージェントに複製を所持させると最適な複製配置となる。

2.2.2 個々のエージェントの情報に基づく方法

2.2.1節では，各エージェントを参照回数が多い順に，全体の通信回数が最小になる複製の数のエージェントに複製を所持させる方法を説明した。しかし，個々のエージェントの1サイクルの参照回数だけを基にして複製を所持させるか所持させないかを決定することができる。

全体の通信回数を最小にする複製の数を決めるには，まずすべてのエージェントに複製を所持させないと仮定して，最も参照回数が多いエージェントに複製を所持させる場合と所持させない場合の通信回数を比較する。もし，所持させる場合の方が所持させない場合よりも通信回数が少ないのなら，そのエージェントに複製を所持させることにして，2番目に参照回数が多いエージェントに対して同じように，複製を所持させる場合と所持させない場合の通信回数を比較する。このような計算を繰り返し行う。複製を所持させない場合の方が所持させる場合よりも通信回数が少ないのなら，そのエージェントは複製を所持させない。このように複製を所持させない場合の方が所持させる場合よりも通信回数が少ないなら，複製配置の計算処理を終了する。

この時，複製を所持させると仮定したときの全体の通信回数と複製を所持させないと仮定したときの全体の通信回数を比べているが，この差は1つのエージェントに複製を所持させるときと所持させないときの差である。したがって，そのエージェントに複製を所持させるか所持させないかは他のエージェントの参照頻度の状態には左右されないで決定できる。

つまり，個々のエージェントの通信回数が少くなるように複製を所持するかしないかを決定すれば，全体としての通信回数も最小になるということである。

システム全体の通信回数を計算して最適な複製の数を求め，参照回数の多い順に複製を配置するのではなく，それぞれのエージェントの参照回数からそのエージェントには複製を所持させるべきか，所持させないべきかを決定することで全体としての最適な複製配置を決定できる。このように個々のエージェント毎に複製配置を決定することによって，計算コストが少なくなる。

具体的な基準に関しては，3章で説明する。

3 通信データ量を考慮した最適化

2.2.2節では個々のエージェントの参照回数を基に複製を所持させる・させないを判断し，全体としての複製配置を最適にすることができることを述べた。我々のこれまでの研究では通信データ量を考慮していなかったが，通信データ量も考慮して最適な複製配置を決定することにする。

この章では，個々のエージェントの参照回数から複製を所持させる・させないを判断するときの判断基準となる具体的な参照回数を通信データ量も考慮して求める。

3.1 基本方針

動的な複製再配置方法は，従来の資源共有方法では

考慮していなかった共有資源の使用頻度を基にして複製を再配置し，より通信回数を減らすことができる。しかし，通信はパケット単位で行われる。つまり，送信されるデータはパケット単位で通信されるので，データが大きければ送信されるパケット数が多くなり通信に時間がかかる。

そこで，通信データのサイズも考慮に入れた動的な複製再配置の判断基準について考える。

複製の再配置方法の基本方針は，通信回数をいままで述べてきたようにデータが大きくても1回と見なすのではなく，データの大きさが大きければ，それに見合った通信回数とする。すなわち，送信されるパケットの個数を通信回数とする。

3.2 通信量を考慮した再配置の判断基準

判断基準となる具体的な参照回数を求めるためにパラメータとして n, R, H, H', D を用意する。

n ：分散共有メモリに関係したエージェントから管理エージェントと更新を発生したエージェントを除いたエージェント数

$\boldsymbol{R}=(r_1, r_2, \ldots, r_n)$：$r_i$ はエージェント i における更新と更新の間の参照回数

$\boldsymbol{H}=(h_1, h_2, \ldots, h_n)$：$h_i$ はエージェントiが複製を所持させるなら1，所持させないなら0

$\boldsymbol{H'}=(h'_1, h'_2, \ldots, h'_n)$：$h'_i$ はエージェント i が更新前に複製を所持していたなら1，所持していなかったなら0

D ：データの大きさ／パケットの大きさ，すなわち，パケット数

3.2.1 通信回数

3.2.1.1 更新1回で必要な通信回数

更新で必要な通信は

1．更新を発生したエージェントが更新要求（共有資源添付）を管理エージェントに送信
2．更新前に複製を所持しているエージェントに更新命令（複製添付）を送信，命令終了後エージェントは命令終了を送信
3．更新前に複製を所持しているエージェントに破棄命令を送信，命令終了後エージェントは命令終了を送信
4．更新前に複製を所持していないエージェントに複製所持命令（複製添付）を送信，命令終了後エージェントは命令終了を送信
5．管理エージェントが更新を発生したエージェントに更新完了を送信

の5種類がある。

この5種類の通信で必要な通信回数と行われる回数は，

1．更新要求には更新された共有資源が添付されているので D 回の通信を必要とする。
行われる回数は，更新を発生した共有資源エージェントのみであるので1回
2．更新命令（複製添付）は D 回の通信を必要とする。共有資源エージェントから管理エージェントに命令
終了を送信するには1回の通信を必要とする。
したがって，$D+1$ の通信が必要。
行われる回数は，更新前に複製を所持していて更新後も複製を所持するエージェント数なので $\boldsymbol{H'}\boldsymbol{H}^t$ 回
3．破棄命令には1回，命令終了には1回の通信が必要である。
したがって，2回の通信が必要。
行われる回数は，更新前に複製を所持していて更新後は複製を所持しないエージェント数なので
$(\boldsymbol{H'}-\boldsymbol{H})\boldsymbol{H'}^t$ 回
4．複製所持命令（複製添付）に D 回，命令終了に1回の通信を必要とする。
したがって，$D+1$ 回の通信が必要。
行われる回数は，更新前に複製を所持していなくて更新後は複製を所持するエージェント数なので
$(\boldsymbol{H}-\boldsymbol{H'})\boldsymbol{H}^t$ 回
5．更新完了には1回の通信を必要とする。
行われる回数は，更新を発生した共有資源エージェントのみであるので1回

である。

したがって，更新1回で必要な通信回数は

$$\boldsymbol{H'}\boldsymbol{H}^t(D+1)+(\boldsymbol{H}-\boldsymbol{H'})\boldsymbol{H}^t(D+1)+(\boldsymbol{H'}-\boldsymbol{H})\boldsymbol{H'}^t*2+(D+1) \quad (1)$$

$$\boldsymbol{H}\boldsymbol{H}^t(D+1)+(\boldsymbol{H'}-\boldsymbol{H})\boldsymbol{H'}^t*2+(D+1) \quad (2)$$

となる。

3.2.1.2.1 1サイクルの参照で必要な通信回数

複製を所持していないエージェントでの参照回数の合計が $(\boldsymbol{1}-\boldsymbol{H})\boldsymbol{R}^t$ である。

また，参照時には参照要求で1回の通信と，管理エージェントが参照要求を送信した共有資源エージェントに対して共有資源を送信するとき D 回の通信を必要とする。

したがって，次の更新までの参照で必要な通信回数は，

$$(1-\boldsymbol{H})\boldsymbol{R}^t(1+D) \tag{3}$$

となる。

3.2.1.3　ロックで必要な通信回数

更新を行う前には複製を所持しているエージェントに対してロックをかける必要がある。ロックをかけるには，ロック命令とその返事で2回の通信を必要とする。また，ロックを解除するにはアンロック命令とその返事で2回の通信を必要とする。したがって，ロックで必要な通信回数は

$$\boldsymbol{H}\boldsymbol{H}^t * 4 \tag{4}$$

である。

3.2.1.4　1サイクルの通信回数

更新1回で必要な通信回数と次の更新までの参照で必要な通信回数とロックで必要な通信回数の和を M とおくと，式2と式3と式4より

$$\begin{aligned} M=\boldsymbol{H}\boldsymbol{H}^t(D+1)+(\boldsymbol{H}'-\boldsymbol{H})\boldsymbol{H}'^t * 4+(D+1) \\ +(1-\boldsymbol{H})\boldsymbol{R}^t(1+D)+\boldsymbol{H}\boldsymbol{H}^t * 4 \end{aligned} \tag{5}$$

で表される。

3.2.2　複製所持の判断基準

式を簡単にするために，参照回数の多い順にエージェント1からnまで並んでいると仮定する。

そのときk番目のエージェントに複製を所持させるか·所持させないかを考える。(k－1番目のエージェントまでは複製を所持させるようにする)

・k番目のエージェントが更新前に複製を所持しているとき

－k番目のエージェントに複製を所持させない場合の通信回数→ M_{no}

$$\begin{aligned} M_{no}=&(k-1)(D+1)+2 \\ &+(h'_{k+1}+h'_{k+2}+\ldots+h'_n)*2 \\ &+(D+1) \\ &+(r_k+r_{k+1}+r_{k+2}+\ldots+r_n)(1+D) \\ &+4*(k-1) \end{aligned} \tag{6}$$

－k番目のエージェントに複製を所持させる場合の通信回数→ M_{yes}

$$\begin{aligned} M_{yes}=&(k-1)(D+1)+(D+1) \\ &+(h'_{k+1}+h'_{k+2}+\ldots+h'_n)*2 \\ &+(D+1) \\ &+(r_{k+1}+r_{k+2}+\ldots+r_n)(1+D) \\ &+4*(k-1)+4 \end{aligned} \tag{7}$$

$M_{no}>M_{yes}$ のとき，複製を所持させる。($M_{no}<M_{yes}$ のとき，複製を所持させない)

式6と式7より

$$M_{no}-M_{yes}>0$$

$$\{2+r_k(1+D)\}-\{(D+1)+4\}>0$$

$$r_k>\frac{(D+3)}{(1+D)}$$

よって，

$$r_k>\frac{D+3}{D+1} \tag{8}$$

のとき複製を所持させる。

・k番目のエージェントが更新前に複製を所持していないとき

－k番目のエージェントに複製を所持させない場合の通信回数→ M_{no}

$$\begin{aligned} M_{no}=&(k-1)(D+1) \\ &+(h'_{k+1}+h'_{k+2}+\ldots+h'_n)*2 \\ &+(D+1) \\ &+(r_k+r_{k+1}+r_{k+2}+\ldots+r_n)(1+D) \\ &+4*(k-1) \end{aligned} \tag{9}$$

－k番目のエージェントに複製を所持させる場合の通信回数→ M_{yes}

$$\begin{aligned} M_{yes}=&(k-1)(D+1)+(D+1) \\ &+(h'_{k+1}+h'_{k+2}+\ldots+h'_n)*2 \\ &+(D+1) \\ &+(r_{k+1}+r_{k+2}+\ldots+r_n)(1+D) \\ &+4*(k-1)+4 \end{aligned} \tag{10}$$

$M_{no}>M_{yes}$のとき，複製を所持させる。($M_{no}>M_{yes}$ のとき，複製を所持させない)

式9と式10より

$$M_{no}-M_{yes}>0$$

$$\{r_k(1+D)\}-\{(D+1)+4\}>0$$

$$r_k>\frac{(D+5)}{(1+D)}$$

よって，

$$r_k>\frac{D+5}{D+1} \tag{11}$$

のとき複製を所持させる。

したがって，式8と式11より

$$\begin{cases} \text{更新前に，複製がある時 } r_k > \dfrac{D+3}{D+1} \\ \text{更新前に，複製がない時 } r_k > \dfrac{D+5}{D+1} \end{cases} \quad (12)$$

のとき複製を所持させることによって，通信回数を最小する複製配置が決定することができる。

3.3 更新で差分データを使う場合の判断基準

一般に，更新を行うときは更新するところはデータの一部であることが多い。したがって，更新にデータの差分を使用することで通信量を減らすことができる。この節では，更新に差分を使用する動的複製再配置方法を説明し，複製配置の判断基準を求める。

3.3.1 差分を使う方法

更新時に更新前との差分を通信することによって，データ量を減らせる。差分を使わない動的複製配置方法と差分を使う動的複製配置方法との違いは，更新時に複製を送信する先の共有資源エージェントが更新前の複製を所持しているなら更新前の複製との差分をとり，その差分データを送信することと，差分データを受信した共有資源エージェントは，その差分データと更新前の複製を使って更新後の複製を作ることである。

3.3.2 通信回数

差分データを使う方法と使わない方法の違いは，更新前に複製を所持しているエージェントに対して複製を送信するときの通信回数がDからdになることである。

3.2節で用意したパラメータにdを追加する

d：差分データの大きさ／パケットの大きさ

式1と式3と式4の複製を送信するときの通信回数Dをdにすると更新1回で必要な通信回数と次の更新までの参照で必要な通信回数とロックで必要な通信回数の和Mは，

$$\begin{aligned} M = {} & \boldsymbol{H'H}^t(d+1) + (\boldsymbol{H}-\boldsymbol{H'})\boldsymbol{H}^t(D+1) \\ & + (\boldsymbol{H'}-\boldsymbol{H})\boldsymbol{H'}^t * 2 + (d+1) \\ & + (1-\boldsymbol{H})\boldsymbol{R}^t(1+D) + \boldsymbol{H}\boldsymbol{H}^t * 4 \end{aligned} \quad (13)$$

となる。

3.3.3 複製所持の判断基準

3.2.3と同様に考えると，

- **k番目のエージェントが更新前に複製を所持しているとき**
  - －k番目のエージェントに複製を所持させない場合の通信回数→ M_{no}
  - －k番目のエージェントに複製を所持させる場合の通信回数 →M_{yes}

$M_{no} > M_{yes}$のとき，複製を所持させる。($M_{no} > M_{yes}$のとき，複製を所持させない)

$$M_{no} - M_{yes} > 0$$
$$\{2 + r_k(1+D)\} - \{(d+1)+4\} > 0$$

よって，

$$r_k > \frac{d+3}{1+D} \quad (14)$$

のとき複製を所持させる。

- **k番目のエージェントが更新前に複製を所持していないとき**
  - －k番目のエージェントに複製を所持させない場合の通信回数→M_{no}
  - －k番目のエージェントに複製を所持させる場合の通信回数→M_{yes}

$M_{no} > M_{yes}$のとき，複製を所持させる。($M_{no} > M_{yes}$のとき，複製を所持させない)

$$M_{no} - M_{yes} > 0$$
$$\{r_k(1+D)\} - \{(D+1)+4\} > 0$$

よって，

$$r_k > \frac{D+5}{1+D} \quad (15)$$

のとき複製を所持させる。

したがって，式14と式15より

$$\begin{cases} \text{更新前に，複製がある時 } r_k > \dfrac{d+3}{D+1} \\ \text{更新前に，複製がない時 } r_k > \dfrac{D+5}{D+1} \end{cases} \quad (16)$$

のとき，複製を所持させることによって通信回数を最小にする複製配置を決定することができる。

4 実験と評価

この章では，これまでに提案されてきた分散資源共有方式と本研究で提案した動的複製配置方式の効率を実験で比較する。

4.1 実験結果

この節では，実験を行って本研究で提案した動的複製配置方式の効率を検討する。また，通信データ量を考慮した動的複製配置方法の効率も検討する。

- 読み出し複製方式
- 完全複製方式
- 動的複製配置方式（データ量を考慮しない）
- 動的複製配置方式（データ量を考慮する）

の4つの方式を使って実験する。

この実験では4台のプロセッサを使う。そのうち1台のプロセッサ上に管理エージェントを配置し，残り3つのプロセッサ上に共有資源エージェントを配置して，参照頻度をランダムに変化させて実験を行う。

実験は共有資源の大きさを1000バイト，4000バイト，7000バイトにしたときの送信したパケット数の合計と実行時間を計測する。

「読み出し複製方式」「完全複製方式」「動的複製配置方式（データ量を考慮しない）」「動的複製配置方式（データ量を考慮する）」の4つの方式を使って，共有資源の大きさを1000バイトにしたときの実験結果を表1に，4000バイトにしたときの実験結果を表2に，7000バイトにしたときの実験結果を表3に示す。（数値は3回の平均）

共有資源の大きさが1000バイトのとき（表1）は，共有資源を送信するときに送信されるパケット数は1パケットであるので，動的複製配置方法でデータ量を考慮しない方法と考慮する方法は全く同じ方法になる。したがって，この2つの方式をまとめて動的複製配置方式としている。

4.2 実験の考察

表1・表2・表3をみると，それぞれのデータの大きさで，「動的複製配置方式（データ量を考慮する）」と「動的複製配置方式（データ量を考慮しない）」が「読み出し複製方式」と「完全複製方式」に比べて効率が良いことが分かる。また，「動的複製配置方式(データ量を考慮する)」は「動的複製配置方式（データ量を考慮しない）」に比べて効率が良いことが分かる。

図5に共有資源の大きさが7000バイトのときの各方式のパケット数の比較を示す。これをみると分かるように，完全複製方式はすべてのプロセッサに複製を配置するのでパケット数は比例している。読み出し複製方式がほかの方式に比べて傾きが大きいのは，更新時に複製を所持しているプロセッサに無効化命令を送信しているからである。データ量を考慮しないときは複製を所持させる判断基準は式(1)より参照回数が2回以上のときであるが，データ量を考慮するときは所持させる判断基準は4/3回以上のときである。したがって，データ量を考慮しないと参照回数の履歴が4/3回以上2回以下のときは複製を所持させないことになり，通信回数が増えてしまう。動的複製配置方式（データ量を考慮しない）が更新回数が12回付近から完全複製方式や動的複製配置方式（データ量を考慮する）よりも傾きが大きくなっているのはこのような理由からである。

送信を行うとき，データはパケット単位で送信される。「参照要求」や「更新命令」は1パケットで送信

表1．共済資源の大きさが1000バイトのとき

方　　式	通信回数	パケット数	実行時間(秒)
読み出し複製方式	724	724	18.25
完全複製方式	662	662	17.61
動的複製配置方式	485	485	15.57

表2．共済資源の大きさが4000バイトのとき

方　　式	通信回数	パケット数	実行時間(秒)
読み出し複製方式	724	976	22.05
完全複製方式	662	974	21.70
動的複製配置方式(1)	491	802	20.03
動的複製配置方式(2)	484	758	18.58

(1)データ量を考慮しない(2)データ量を考慮する

表3．共済資源の大きさが7000バイトのとき

方　　式	通信回数	パケット数	実行時間（秒）
読み出し複製方式	724	1228	25.25
完全複製方式	662	1286	25.68
動的複製配置方式(1)	493	1113	23.67
動的複製配置方式(2)	492	1012	21.90

(1)データ量を考慮しない　(2)データ量を考慮する

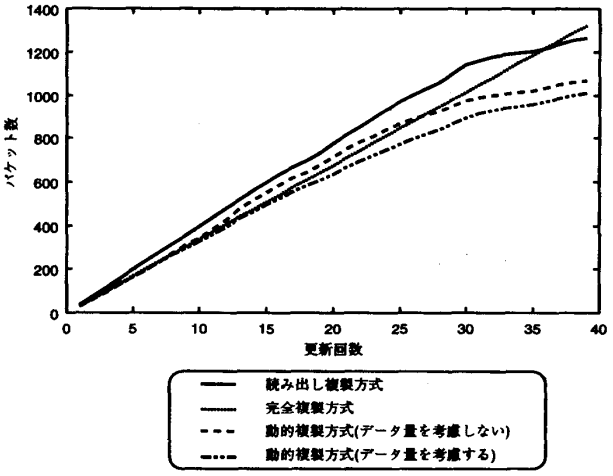

図5．7000バイトのときのパケット数の比較

される。データの大きさが1パケットの大きさより大きければ，データはパケットに入る大きさに分割され送信される。

この実験ではデータを1000・4000・7000バイトにして行った。1パケットの大きさは1,500バイトであるので,データを送信するとき送信されたパケット数は，それぞれ1パケット・3パケット・5パケットである。通信コストは計算コストに比べかなり高いので，データの大きさが大きいと送信時におけるパケット数が増えるので，効率が悪くなる。

この実験で図6に示すように，送信されたパケット数が多くなると実行に時間がかかることが分かる。

図7に示すように，それぞれの方式でも同じパケット数であれば実行時間はほぼ同じになることが分かる。

管理サーバを配置する方式では管理サーバに通信が集中すると管理サーバの負荷が高くなり，処理に多くの時間がかかる。したがって，通信が管理エージェントに集中するとパケット当りの実行時間が増加する。それぞれの方式での同じパケット数に対する実行時間が若干異なっているのはこのような理由からである。

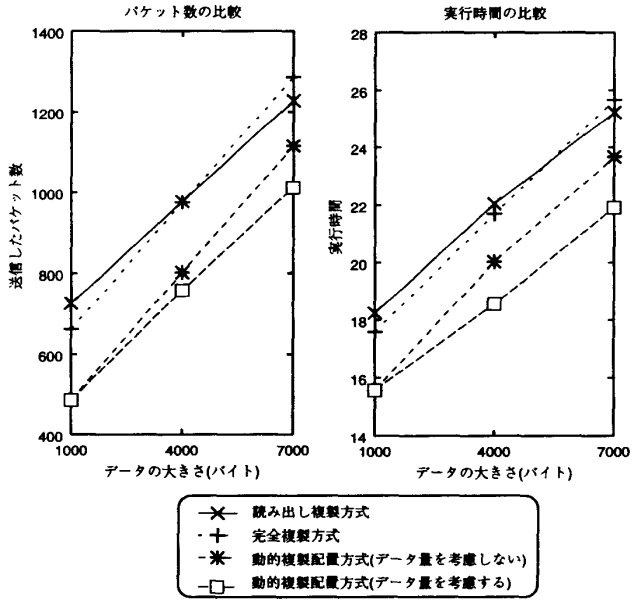

図6．パケット数と実行時間

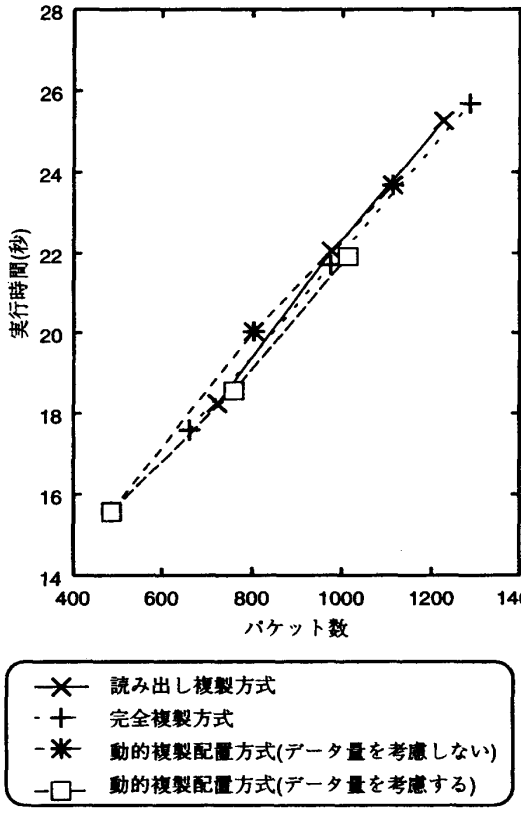

図7．パケット数と実行時間

5　おわりに

本研究では，分散環境下における効率的な資源共有を行うために，共有資源の使用頻度を基に動的に最適な複製配置を行う方法を提案した。また，我々のこれまでの研究では，制御信号を送信するときも共有資源を送信するときも同様に1回の通信と考えて複製配置を決定していた。しかし，実際には送信するデータ量でパケット数が異なる。そこで，通信データ量を考慮した最適な複製配置の判断基準を求めた。

従来の分散資源共有方法と本研究で提案した動的複製配置方法の効率を実験によって比較し，動的複製配置方法の効率が良いことを示した。また，通信データ量を考慮することによってさらに良い効率を得ることができることも示した。

今回は，研究の第一歩として管理エージェントを設置して資源共有を行ったが，今後は，管理エージェントの負荷を分散させる方法について考えていきたい。

Lazy Release Consistency というアルゴリズムが提案されている[7]。更新を行うとすぐにその更新をほかのプロセッサに反映させようとしないで，ほかのプロセッサが共有資源を必要とするときにその更新を共有資源を必要とするプロセッサに反映するというアルゴリズムである。このアルゴリズムを使うことで通信回数を削減できる。今後は，この Lazy Release Consistency を取り入れた方法について考えていきたい。

参考文献

1）濱地弘樹。“分散問題解決による共有資源の実現”。九州大学卒業論文，February 1996

2）濱地弘樹，楢崎修二，吉田紀彦，牛島和夫。“分散問題解決による共有資源の実現”。情報処理学会

「プログラミング」研究報告 97-PRO-14, pp. 31-36

3）濱地弘樹，楢崎修二，吉田紀彦，牛島和夫。“分散問題解決による共有資源の効率的な実現”。情報処理学会第55回全国大会 3 T-2 (1997)

4）Michael Stumm and Songniam Zhou. “*Algorithms Implementing Distributed SharedMemory*”. IEEE Computer, Vol. 23, No. 5, pp. 54-64, May 1990.

5）Gregor Kiczales. “*Tiny CLOS*”. Xerox, ftp: //arisia. xerox. com/pub/openimplementations/tiny/, 1991.

6）“*Guile*”. http: //www. gnu. ai. mit. edu/software/guile/guile. html

7）Pete Keleher, Alan L. Cox, and Willy Zwaenepoel “*Lazy Release Consistency for Software Distributed Shared Memory*”. Proceedings of the 19th Annual International Symposium on Computer Architecture, 1992, pp. 13-21.