Vol. 43 No. 8 情報処理学会論文誌 Aug. 2002

階層型ネットワークにおける集中管理方式と分散管理方式の性能評価

川 村 尚 生† 増 山 博†

階層型ネットワーク内に多数のノードが存在する場合に，データベースをスーパバイザ・ノードが集中して保管している場合と各クライアント・ノードが分散して保管している場合のそれぞれについて，データ検索の効率特性を相対評価した．評価は，1 つのノードが 1 つのデータを求めて得られるまでに要する平均時間によって行った．さらに，分散保管の場合において，各ノードがキューにデータを保持する場合とそうでない場合について，データ・アクセス効率の時刻変化を解析し，キューの適容量の評価方法を示した．また，WWW キャッシュ・システムから抽出されたアクセス記録を用いたシミュレーションによる実験的評価も行った．

Performance Evaluation of Concentrated Management System and Distributed Management System in Hierarchical Networks

Takao Kawamura† and Hiroshi Masuyama†

In this paper, we show a performance evaluation of two types of database in hierarchical networks. In one type, the database is managed by one supervisor node, and in the other type it is distributed among all nodes. In the evaluation, how the time required by an node to obtain a searched data in the latter type (a distributed management system) changes for the length of queue, the difference of network architecture, and the frequency of data access is mainly analyzed. The performance of hierarchical networks is evaluated in a stochastic manner and by simulating a WWW cache system.

1. はじめに

インターネットや企業内イントラネットに代表される最近の情報システムは，コンピュータ間の結合がシームレスになりつつあるために，ますます多様，複雑になってきている．つまり，多数のクライアントと 1 個のサーバとを直結するクライアント/サーバアーキテクチャを初期のシステムとすると，これまでのようなトポロジの固定したクライアントの相互接続による分散アーキテクチャ，さらに将来にはノード相互接続によるマルチノードアーキテクチャというふうにシステムは柔軟な接続形態に変わりつつある．一方，Ad Hoc ネットワークと呼ばれる，必要に応じ一時的に構築されるネットワークもさかんに研究されるようになった[1]．

このため，システム全体はあらかじめ初期に確定したものとしては予想できないトポロジに変わることがある．このような環境の下では，大規模，かつ複雑なデータベースを，1 個のスーパバイザに集中して保管させるのではなく，広範囲に分散して保管する方が，効率が良い場合があると思われる．もちろん，その場合にも広域分散した大規模，かつ複雑なデータベースから必要な情報を効率良く抽出することが必要である．

従来のクライアント/サーバ直結型アーキテクチャやトポロジの固定した分散アーキテクチャ等の性能評価については，以前から各種の立場で多数の報告がなされている[2]．最近では，高速な分散環境のために開発された LAN[3] や分散環境における Kerberos プロトコル[4] の性能評価等もある．

我々は，ネットワーク内に多数のノードが存在するときのデータ検索効率の評価を行っている[5)~8]．

本論文は，ネットワークにおいて，データベースを各クライアント・エージェントが分散して保管している場合（以後，分散管理方式と呼ぶ）とスーパバイザ・エージェントが集中して保管している場合（以後，集中管理方式と呼ぶ）とのデータ検索効率を相対評価したものである．また，ネットワーク上でのデータベースから必要な情報を効率良く抽出するうえで，各エージェントが情報を一時記憶することは重要である．こ

† 鳥取大学工学部
Faculty of Engineering, Tottori University

のため，1度検索したデータを有限長のキューに保持するようなシステムのデータ・アクセス効率の時刻変化についても解析を行い，効率的な一時記憶容量について議論する．さらに，ネットワーク構造やデータの検索頻度の違いが検索効率に及ぼす影響について評価する．評価は，統計的手法に加えて，実働しているWWW キャッシュ・システムから抽出されたWWW アクセス記録を基にしたシミュレーションによって行う．システムトポロジとしては，柔軟性が高く，インターネットや企業内イントラネットにおいて最も一般的なトポロジである，階層型ネットワークを用いる．

本論文は以下のように構成されている．まず2章で対象となるシステムを示す．3章で集中管理方式と分散管理方式におけるデータ検索効率を解析し，定式化する．4章はその評価を行う．次に，5章でWWW アクセス記録によるシミュレーション方法の説明をし，6章でその実験結果を示す．最後に7章で本論文をまとめる．

2. 対象システム

本論文では，ネットワーク内に多数のノードが存在する場合に，1つのノードが1つのデータを求めて得られるまでに要する平均時間の解析を，集中管理方式と分散管理方式において行う．

複数のノードが同時にデータを検索する際の通信路での衝突を解析するためには，いくつかの確率モデルが考えられるが，本論文では，集中，分散の2方式の相対評価を得ることを最大の目標としているので，確率モデルを静的に，かつ単純化して最大競合状態を想定し，次の仮定を設ける☆．

- 各ノード a_i は所望のデータ d_i の検索をいっせいに行う．d_i を保持しているノード $a_j(i \neq j)$ は，d_i 検索メッセージの応答メッセージとしてデータ d_i そのものを a_i に対して送信する．d_i を保持しないノードは d_i 検索メッセージを無視する．また，すべてのノードがデータを取得するまで次のデータ検索は行われない．また，ネットワーク上に存在しないデータの検索は行われない．

集中管理方式においては，唯一のスーパバイザ・ノードがすべてのデータを保持しており，他のノードは，スーパバイザ・ノードに要求することで求めるデータを得ることができる．

分散管理方式においては，データは各ノードに均等に分散されており，ノードがあるデータを必要とするとき，まず自分が保持しているかどうかチェックする．自分が保持していないとき，まず自分の所属するサブネットワーク，次に近傍のサブネットワーク…と，データが見つかるまで順に検索範囲を拡大していくものとする．

また，1度検索して見つけたデータを有限長のキューに保持することで，2度目以降の検索が省略できると考えられるので，キューの有無によって，データを求めて得られるまでに要する時間がどのように変化するかについても，分散管理方式において解析する．

以上のような解析の視点を具体的にするために，上記の仮定に次の仮定を加える．

- 各ノードは均一の処理能力を持つ．したがって，データ検索にかかる時間として，ノード内部でのデータ検索のための処理時間を無視しても解析結果には影響しない．
- データはすべて同一サイズである．
- ノード a_i からのデータ d_i の検索メッセージと，それに応答したデータ d_i を1組のメッセージとし，通信路の占有を別々に考えないものとする．あるノードが要求したデータを保持しているノードは，そのデータを送る．保持していないノードは要求を単に無視する．したがって，ノードからのデータ要求とそれに対する返事を1組のメッセージとしても，解析結果には影響しない．

また，階層型ネットワークに対して以下のような条件を設ける．これらの条件と先の3つの仮定は，統計的解析のために設けるが，いずれも現実的なものであり，集中管理方式と分散管理方式の性能比較を行ううえで，議論の一般性を失わせるものではない．

- 最下層のサブネットワーク（以後，階層1のサブネットと呼ぶ）には w 個のノードが存在する．
- 階層1のサブネットが w 個集まって，上位のネットワーク，つまり w^2 個のノードからなる階層2のサブネットを構成する．この結合が h 回繰り返されることにより，h 個の階層を持つネットワークが構成される．したがって，このとき，ネットワーク上に存在するノードの個数は w^h となる．**図1**は，$w = 3$，$h = 2$ のネットワークである．
- 同時に1メッセージしか通信路上を伝送されない．
- ノード間の通信時間は，衝突がない限り，その距離によらず一定である．

3. 集中管理方式と分散管理方式の比較

同時に複数のメッセージが通信路を流れることはできないので，複数のノードが同時にメッセージを送ろ

☆ この仮定の妥当性については7章で明らかにしている

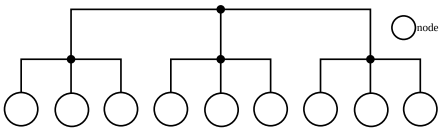

図 1　対象ネットワーク例
Fig. 1　A target network.

うとした場合は，即座にメッセージを送ることのできるノードは 1 つのみで，残りは通信路が空くのを待たなければならない．1 ステップを，ノード a_i がデータ d_i の検索メッセージを送信し，d_i を受信するのに要する時間間隔と定義する．1 ノードが，データを検索しようとしてから得られるまでにかかる平均所要ステップ数（以後，ステップ数と呼ぶ）の期待値を，集中管理方式と分散管理方式で比較を行う．

3.1　集中管理方式

ネットワーク上の任意のノードをスーパバイザ・ノードとする．同時には 1 ノードしかスーパバイザ・ノードにアクセスできないので，w^h 個の各ノードがそれぞれ必要とするステップ数は，0 から w^h-1 のいずれか異なる値をとると考えることができる．したがって，ノード a_i がデータ d_i を獲得するのに必要とするステップ数の期待値を E_c とすると次式で表される．

$$E_c = \frac{\sum_{k=0}^{w^h-1} k}{w^h} = \frac{w^h-1}{2} \tag{1}$$

3.2　分散管理方式

この方式では，ノード a_i は，階層 1，階層 2，⋯，階層 h の順に，その階層に属するノードに向かってデータ d_i の検索メッセージを送り，データを検索する．同一階層内のサブネットの各ノードは，マルチキャストによっていっせいにメッセージを受信できる．つまり，階層 $(l-1)$ のサブネット内の全ノードに d_i 検索メッセージを送信後，1 ステップ経ても d_i が送られてこないとき，次に階層 l のサブネットに向けて d_i 検索メッセージを送信する．以下，同様である．

ノード a_i がほしいデータ d_i をすでに所有している確率を P_0，自分が属する階層 1 のサブネット内でデータが得られる確率を P_1，階層 2 のサブネットのうち，自分が属する階層 1 のサブネットを除いたサブネットで得られる確率を $P_2 \cdots$ とすると，それぞれの場合における，d_i が得られるまでの所要ステップ数の期待値 $E_d(l)$ $(l=0,\ldots,h)$ が求められる．

(1)　$l=0$ の場合

自分がすでに所有するデータを得るのに要するステップ数は 0 なので

$$E_d(0) = P_0 \times 0 = 0 \tag{2}$$

(2)　$l=k(k>0)$ の場合

a_i が k 回目の試みとして，つまり階層 k のサブネットにメッセージを送出しようとしたときに競合する他のノードとは，それぞれが所属する $k-1$ 以下の階層のサブネット内に求めるべきデータが存在しないノードである．したがって，競合するノード数が j になる確率 $Q_{k,j}$ は

$$Q_{k,j} = \binom{w^k-1}{j}\left(1-\sum_{i=0}^{k-1}P_i\right)^j \times \left(\sum_{i=0}^{k-1}P_i\right)^{(w^k-1)-j} \tag{3}$$

となる．また，$j+1$ 個のノードが同時にメッセージを送出しようとした場合には競合が発生するので，$j+1$ 個のノードのそれぞれがかかるステップ数は，1 から $j+1$ のいずれかの互いに異なる値となる．したがって，$E_d(k)$ は次式で表される．

$$E_d(k) = \sum_{m=1}^{k}\sum_{j=0}^{w^m-1} Q_{m,j}\frac{\sum_{i=1}^{j+1} i}{j+1} \tag{4}$$

以上より，分散管理方式のステップ数 E_d は次式で表される．

$$\begin{aligned} E_d &= \sum_{l=0}^{h} P_l E_d(l) \\ &= \sum_{l=1}^{h} P_l \sum_{m=1}^{l}\sum_{j=0}^{w^m-1}\binom{w^m-1}{j} \\ &\quad \times \left(1-\sum_{i=0}^{m-1}P_i\right)^j \left(\sum_{i=0}^{m-1}P_i\right)^{(w^m-1)-j} \\ &\quad \times \frac{\sum_{i=1}^{j+1} i}{j+1} \end{aligned} \tag{5}$$

4. 評　　価

前章において，集中管理方式と分散管理方式のそれぞれの期待値 E_c，E_d が定式化された．各ノードがキューを持つときは，分散管理方式の期待値 E_d の変数 P_l が時間経過によって変化するため，E_d も変化する．本章では，ノードがキューを持たないときと持つときに分け，具体的に P_l を求め，E_d を評価する．

4.1　分散管理方式においてノードがキューを持たない場合

ここでは，各ノードが他のノードから確保したデータを保持するキューを持たないシステムを評価する．

4.1.1 データの分布が均質な場合

データ d_j をどのノードが所有しているかは，すべてのノードについて等確率とする．すなわち，すべてのノードについて，データ d_j を所有する確率は，$1/w^h$ である．任意のノードから見て，階層 l のサブネットのうち，自分が属する階層 $(l-1)$ のサブネットを除いたサブネットに属するノード数は $w^l - w^{l-1}$ であるから，P_l は次式で表せる．

$$P_l = \begin{cases} \frac{1}{w^h} & (l=0) \\ \frac{w^l - w^{l-1}}{w^h} & (l=1,\ldots,h) \end{cases} \quad (6)$$

この P_l を用いて式 (5) が計算できる．

h を固定してノード数を変化させたときの結果を**図 2** に，w を固定してノード数を変化させたときの結果を**図 3** にそれぞれ示す．凡例で concentrated と記されている直線が集中管理方式の場合である．

図 2 では，h が小さければ小さいほどステップ数が小さいという結果になっている．一方．図 3 では，w が 2 の場合より 8 の場合の方がステップ数は小さいが，8 の場合より 22 の場合の方がステップ数が大きくなっている．これは，ノード数を固定したときに，最小の h と最大の w の組合せ，すなわち h が 1 で w が総ノード数の場合がステップ数を最小にするわけではないことを示唆している．

w が大きくなることは 1 度にマルチキャストできるノード数が増えることを意味し，h が大きくなることはシステム全体に並行して送られるメッセージ数が増えることを意味しており，w と h の増減はトレードオフの関係にある．各 P_l が与えられれば，ノード数を固定したときに，最もステップ数を小さくするような w と h の組合せが存在するはずであるが，本論文で議論している比較的小規模なネットワークでは w と h の組合せが限られており，最適構造の議論はできない．

4.1.2 データの分布が非均質な場合

データを複数のノードに分散配置するとき，物理的に近いノードが同じデータを必要とするように配置することが考えられるので，ここではそのようなデータ分布を扱う．すなわち，a_i と階層的に近いサブネットに d_i がより高い確率で存在することを確率的に表現するために，$P_l(l=0,\ldots,h-1)$ が幾何分布に従うものとし，次式を与える．

$$P_l = \begin{cases} p(1-p)^l & (l=0,\ldots,h-1) \\ 1-\sum_{k=1}^{h-1} P_k & (l=h) \end{cases} \quad (7)$$

この P_l を用いて式 (5) が計算できる．

$p=0.1$，$p=0.3$，$p=0.5$ のそれぞれについて，$h=5$

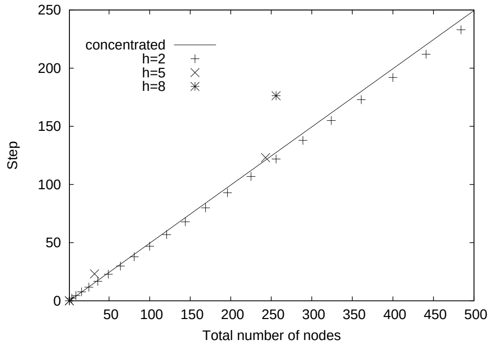

図 2 h を固定したときのステップ数の推移
Fig. 2 Change of the number of steps (fixed h).

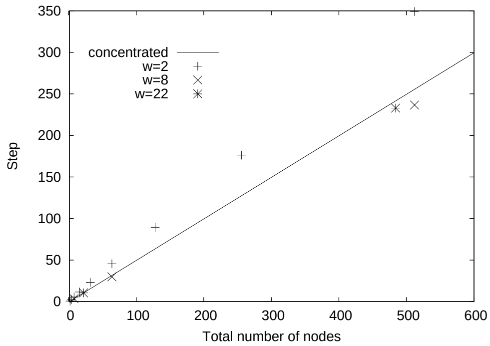

図 3 w を固定したときのステップ数の推移
Fig. 3 Change of the number of steps (fixed w).

として w を変化させた結果を**図 4** に示す．参考のために，4.1.1 項で示した等確率の場合（凡例で uniform と記されている点の集合）と集中管理方式の場合（凡例で concentrated と記されている直線）もあわせて示している．

分散管理方式においては，幾何分布の初期値が大きくなるにつれてステップ数が減少していく．各ノードにとって，求めるデータが近傍で得られる確率が大きくなるので，小さいステップ数でデータが得られるからである．

4.2 分散管理方式においてノードがキューを持つ場合

ここでは，各ノードが他のノードから確保したデータを保持するキューを持つシステムを評価する．2 章での仮定に加え，さらに次の仮定を設けて解析を具体化する．

- n 個のデータが各ノードに同数ずつ分散されている．
- 各ノードは長さ q のキューを持ち，検索したデータをキューの末尾に追加する．このとき，もとも

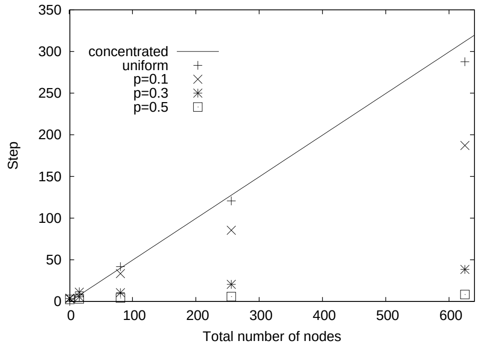

図 4　P_l が幾何分布に従う場合のステップ数の推移
Fig. 4　Change of the number of steps (Each P_l has a geometric distribution).

と自分が持っていたデータについても，それが検索されたならばキューに保持するものとする．キューが一杯になったときにデータを保持する必要が生じた場合，最も古いデータを捨てる．また，同じデータを重複して保持することはなく，古い方を捨てる．

- ノード a_i は任意のデータ d_j について，確率 $p(a_i,d_j)$ で検索する．

4.2.1　準　　備

時刻 t においてノード a_i がデータ d_i を検索しようとしたとき，実は長さ q の自分のキューに d_i を保持している場合があるので，その確率 $p(t,a_i,q,d_i)$ を求め，P_l を定式化する．

a_i が d_i を自分のキューに保持しているためには，時刻 t 以前に d_i を検索しているはずであるから，最後に検索した時刻 $\tau(<t)$ が存在する．このとき，τ の 1 ステップ後の $\tau+1$ から $t-1$ までの間に，d_i 以外の $q-1$ 種類以下のデータしか検索しなければ，d_i がキューに保持されている．

(1)　a_i が時刻 $\tau+1$ から $t-1$ までの間に d_i 以外の 1 種類のデータのみしか検索しない確率 $p(\tau,t,a_i,q,d_i,1)$ は

$$p(\tau,t,a_i,q,d_i,1)=\sum_{d_\beta\in D}\{p(a_i,d_\beta)\}^{t-\tau-1} \tag{8}$$

となる．ただし，

$$D=\{d_\beta|1\le\beta\le n,\beta\ne i\} \tag{9}$$

とする．

(2)　a_i が時刻 $\tau+1$ から $t-1$ までの間に d_i 以外に 2 種類のデータしか検索しない確率 $p(\tau,t,a_i,q,d_i,2)$ は，2 個のデータ d_{β_1}，d_{β_2} の $(t-\tau)$-重複組合せを用いて，以下のようになる．

$$p(\tau,t,a_i,q,d_i,2)=\sum_{d_{\beta_1},d_{\beta_2}\in D}\sum_{\alpha_1,\alpha_2}\frac{(t-\tau-1)!}{\alpha_1!\alpha_2!}\times\{p(a_i,d_{\beta_1})\}^{\alpha_1}\{p(a_i,d_{\beta_2})\}^{\alpha_2} \tag{10}$$

ただし，$\alpha_m(m=1,2)$ は自然数で，式 (11) および式 (12) を満たし，$\beta_m(m=1,2)$ は自然数で，$\beta_1\ne\beta_2$ とする．

$$\alpha_1+\alpha_2=t-\tau-1 \tag{11}$$

$$1\le\alpha_m<t-\tau-1\quad(m=1,2) \tag{12}$$

$$\beta_1\ne\beta_2 \tag{13}$$

(3)　一般に，a_i が時刻 $\tau+1$ から $t-1$ までの間に d_i 以外に k 種類のデータしか検索しない確率 $p(\tau,t,a_i,q,d_i,k)$ は，k 個のデータ $d_{\beta_1},d_{\beta_2},\cdots,d_{\beta_k}$ の $(t-\tau)$-重複組合せを用いて，以下のようになる．

$$p(\tau,t,a_i,q,d_i,k)=\sum_{d_{\beta_1},d_{\beta_2},\cdots,d_{\beta_k}\in D}\times\sum_{\alpha_1,\alpha_2,\cdots,\alpha_k}\frac{(t-\tau-1)!}{\alpha_1!\alpha_2!\cdots\alpha_k!}\times\{p(a_i,d_{\beta_1})\}^{\alpha_1}\{p(a_i,d_{\beta_2})\}^{\alpha_2}\times\cdots\times\{p(a_i,d_{\beta_k})\}^{\alpha_k} \tag{14}$$

ただし，$\alpha_m(1\le m\le k)$ は自然数で，式 (15) および式 (16) を満たし，$\beta_m(1\le m\le k)$ は自然数で，互いに異なるものとする．

$$\sum_{m=1}^{k}\alpha_m=t-\tau-1 \tag{15}$$

$$1\le\alpha_m<t-\tau-1\quad(1\le m\le k) \tag{16}$$

$$\beta_1\cdots\beta_k\quad\text{は互いに異なる．} \tag{17}$$

以上の議論より，時刻 t において，a_i が d_i をキューに保持している確率 $p(t,a_i,q,d_i)$ が次式で表現できる．

$$p(t,a_i,q,d_i)=\begin{cases}0 & (q=0)\\ p(a_i,d_i) & (q=1)\\ \displaystyle\sum_{\tau=1}^{t-1}p(a_i,d_i)\sum_{k=1}^{q-1}\sum_{d_{\beta_1},d_{\beta_2},\cdots,d_{\beta_k}\in D}\times\sum_{\alpha_1,\alpha_2,\cdots,\alpha_k}\frac{(t-\tau-1)!}{\alpha_1!\alpha_2!\cdots\alpha_k!}\times\{p(a_i,d_{\beta_1})\}^{\alpha_1}\{p(a_i,d_{\beta_2})\}^{\alpha_2}\times\cdots\times\{p(a_i,d_{\beta_k})\}^{\alpha_k} & (q>1)\end{cases} \tag{18}$$

これを用いて，P_l が以下のように求められる．

(1)　$l=0$ の場合

ノード a_i が d_i を検索しようとしたとき，すでに d_i を

持っているのは，d_i をもともと所有しているか，他のノードから得た d_i をキューに保持しているかのどちらかである．

a_i がもともと d_i を所有していず，かつ，時刻 t において d_i をキューに保持していない確率 $r(t,a_i,q,d_i)$ は次式のようになる．

$$r(t,a_i,q,d_i) = \left(1-\frac{1}{w^h}\right) \times \left(1-p(t,a_i,q,d_i)\right) \quad (19)$$

$r(t,a_i,q,d_i)$ を用いて，P_0 は次のように表される．

$$P_0 = 1-r(t,a_i,q,d_i) \quad (20)$$

(2) $l = k(0<k<h)$ の場合

階層 $k-1$ までのサブネットで d_i が見つからず，階層 k に属するいずれかのノードが d_i を持つ．したがって，P_k は次のようになる．

$$P_k = \{r(t,a_i,q,d_i)\}^{w^{k-1}} \times \left(1-\{r(t,a_i,q,d_i)\}^{w^k-w^{k-1}}\right) \quad (21)$$

(3) $l = h$ の場合

最上位の h 未満の階層で d_i が見つかっていない場合，最上位の階層 h には必ずデータが存在するので，P_h は次のようになる．

$$P_h = \{r(t,a_i,q,d_i)\}^{w^{h-1}} \quad (22)$$

4.2.2 計 算 値

t を 1 から $+\infty$ に向けて $+1$ ずつ増加させ，t の各ステップごとに $0<\tau<t$ なる τ について式 (18) を計算すれば，各 t に対して得られた P_l を用いて式 (5) の E_d が計算できる．ここでは，データの分布が均質な場合について計算する．本章での仮定によりデータ数 n が与えられているので，式 (18) の $p(a_i,d_i)$ は式 (23) で与えられ，式 (20)〜(22) から求めた P_l を式 (6) の代わりに用いることができる．

$$p(a_i,d_i) = \frac{1}{n} \quad (1 \le i \le w^h, 1 \le j \le n) \quad (23)$$

$w=h=3$，$n=108$ とし，q を 0，2，4，6 と変化させたときの式 (5) の値を**図 5** に示す．

時間が経過するに従い，$p(t,a_i,q,d_i)$ は q/n に収束するので，ステップも一定値に向かって収束していく．また，キューが長いほど，ステップは小さくなっている．ステップを小さくするためにはキューを長くすればよいが，反面，キューが長いということはそれだけ物理的資源を必要とするので，コスト $V(q)$ として次式のような定義をしたとき，

$$V(q) = E_d \times q \quad (24)$$

$V(q-1)/V(q)(=V'(q)$ とおく$)>1$ であれば，q を 1 単位長増加させることによるコストパフォーマンスの改善があると考えることができる．$p(t,a_i,q,d_i)$ を q/n で近似した，$w=h=3$，$n=108$ のシステムでのコストパフォーマンスの推移を**図 6** に示す．$V'(q)$ は，$q=10$ 付近から $q=90$ 付近においてはあまり変化しない．したがって，コストパフォーマンスの見地からは，q の値としては 10 程度が妥当であることが示唆されている．

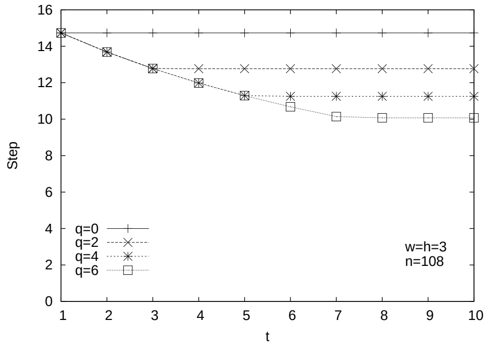

図 5 ノードがキューを持つ場合のステップ数の時間推移
Fig. 5 Change of the number of steps (Each node has a queue).

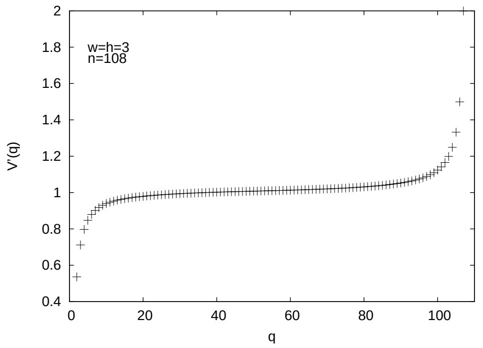

図 6 q と $V'(q)$ の関係
Fig. 6 Relationship between q and $V'(q)$.

5. WWW アクセス記録を用いた実験

ノードがデータを探し始めてから得るまでに要するステップ数はネットワーク全体でのデータ・アクセス・パターンに依存する．統計的評価においては，

- 各ノード a_i は所望のデータ d_i の検索をいっせいに行う．d_i を保持しているノード $a_j(i \ne j)$ は，d_i 検索メッセージの応答メッセージとしてデータ d_i そのものを a_i に対して送信する．d_i を保持しないノードは d_i 検索メッセージを無視する．また，すべてのノードがデータを取得するまで次のデータ検索は行われない．また，ネットワーク上に存在しないデータの検索は行われない．

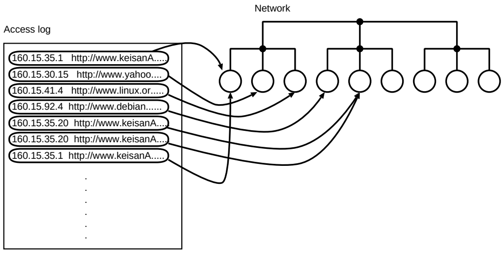

図 7　WWW クライアントのノードへの割り当て
Fig. 7　Assigning WWW clients to nodes.

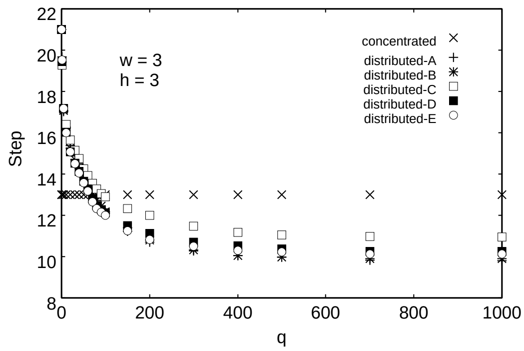

図 8　q とステップ数の関係
Fig. 8　Relationship between q and *steps*.

という仮定を置いたが，この仮定を除き，WWW キャッシュ・システムに残されたアクセス記録から抽出された，現実のデータ・アクセス・パターンを用い，シミュレーション実験を行った．

WWW キャッシュ・システムは，WWW サーバと WWW クライアント間の通信を中継すると同時にデータのキャッシングを行う．次回以降同一データのアクセスがあった際にはそのキャッシュ・データを用いることにより，ネットワークのトラフィックを軽減することを目的としている[9]．

集中管理方式はスーパバイザ・ノードのみがキャッシングを行うシステムに，分散管理方式はすべてのノードがキャッシングを行うシステムに相当する．

以下のような手順で，2 方式のシミュレーションを行う．

(1)　アクセス記録から必要な数の WWW クライアントを選んでノードとする．たとえば，$w = h = 3$ ならば 9 個の WWW クライアントが選ばれる．その他の WWW クライアントに関する記録は無視する．

(2)　**図 7** に示すように，選択した WWW クライアントを階層型ネットワーク内の各ノードに割り当てる．

(3)　アクセス記録に従ってデータ検索をシミュレートし，ステップ数を評価する．ただし，対象としている階層型ネットワークの外へのデータ・アクセスに要する時間は無視する．これは，WWW キャッシュ・システムが WWW サーバにデータを要求して得るまでの時間に相当する．

6. 実 験 結 果

鳥取大学工学部知能情報工学科の WWW キャッシュ・システムの 1997 年 9 月から 1999 年 10 月までのアクセス記録を用いて実験を行った．

6.1　キューの長さとステップ数の関係

w^h 個の WWW クライアントをランダムに選んでノードとし，ノードのキューの長さ q を変化させながらシミュレーションを行った．$w = h = 3$ のときの結果を**図 8** に示す．凡例で concentrated と記されたものが集中管理方式の結果で，distributed-A から distributed-E までは分散管理方式の結果である．選択された WWW クライアントによって多少結果は異なるが，いずれの場合でも，q が小さいときは集中管理方式の方がステップ数が小さく，q が長くなるに従って分散管理方式のステップ数が減少し，$q = 100$ 前後で集中管理方式よりも小さくなっている．

6.2　データ・アクセス頻度とステップ数の関係

次に，データ・アクセスの頻度とステップ数の関係を調べた．**図 9** は 1 週間のデータ・アクセス回数が 15,783 回の週における平均ステップ数で，**図 10** は 1 週間のデータ・アクセス回数が 102,387 回の週における平均ステップ数を q を変化させながら調べた結果である．凡例で concentrated と記されたものが集中管理方式，distributed と記されたものが分散管理方式の結果である．データ・アクセス頻度が小さいときは，集中管理方式も分散管理方式も同様の結果を示しているが，データ・アクセス頻度が大きい場合は，q が大きくなれば分散管理方式のステップ数が集中管理方式のそれを下回る．つまり，データ・アクセス頻度が大きい場合はキューが有効利用されていることが分かる．

データ・アクセス頻度とステップ数の関係を詳しく調べるために，ノードが任意の時間にデータを検索しようとする確率（以後，検索確率と呼ぶ）というパラメータを導入する．

$w = 3$ で $h = 2$ のときの結果を**図 11** に示す．凡例で concentrated と記されたものが集中管理方式，distributed と記されたものが分散管理方式の結果である．この場合，$q = 0$ においては集中管理方式の方が分散管理方式に比べてつねにステップ数が小さいが，検索

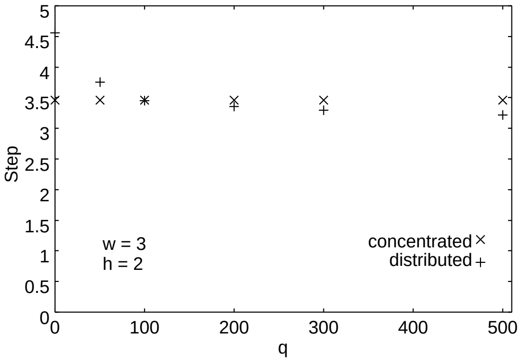

図 9 データ・アクセス頻度が低い場合（1 週間に 15,783 アクセス）の q とステップ数の関係

Fig. 9 Relationship between q and *steps* in the system whose frequency of data access is low (15,783 accesses a week).

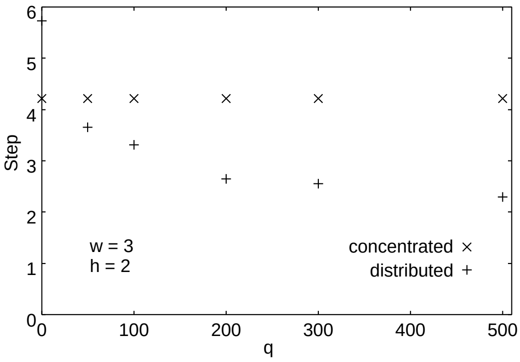

図 10 データ・アクセス頻度が高い場合（1 週間に 102,387 アクセス）の q とステップ数の関係

Fig. 10 Relationship between q and *steps* in the system whose frequency of data access is high (102,387 accesses a week).

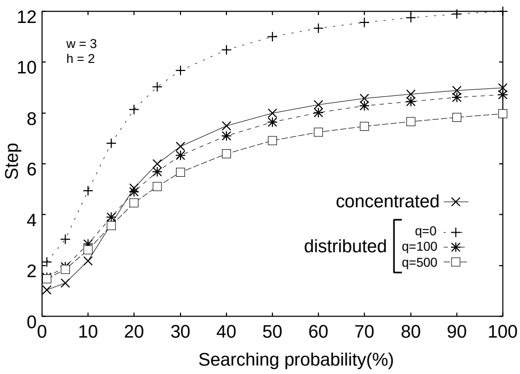

図 11 検索確率とステップ数の関係

Fig. 11 Relationship between searching probability and *steps*.

確率が 15%を超えた場合，$q = 100$ や $q = 500$ においては分散管理方式の方が集中管理方式よりステップ数が小さくなっている．

6.3 ネットワーク構造とステップ数の関係

ネットワークの構造とステップ数の関係を調べるた

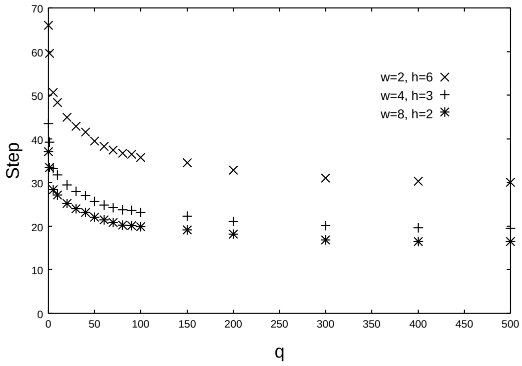

図 12 ネットワーク構造とステップ数の関係

Fig. 12 Relationship between network architectures and *steps*.

めに，64 個の WWW クライアントを選び，w と h を変化させて実験を行った．**図 12** に示すように，w が大きいほどステップ数が小さくなった．4.1.1 項で述べたように，w が大きくなることは 1 度にマルチキャストできるノード数が増えることを意味し，h が大きくなることはシステム全体に並行して送られるメッセージ数が増えることを意味しており，w と h の増減はトレードオフの関係にある．この規模のネットワークにおいては，w の増加による利得が h の増加による利得を上回るものと考えられる．

7. おわりに

階層型ネットワーク上で，分散管理方式と集中管理方式の 2 つのデータ管理方式におけるデータ・アクセスの効率特性を明らかにした．各ノードが必要とするデータを所在する距離に関係なく均等の確率で置く場合に，分散管理方式は集中管理方式と比べて遜色ない効率を持つが，別の場合つまり，各ノードが必要とするデータをそのノードの近くに高い確率で置く場合には，分散管理方式ははるかに高い効率を持つことが分かった．また，分散管理方式において，キューを持つ各ノードが，すべてのデータを等確率で検索する場合におけるデータ・アクセス効率の時刻変化を解析し，その特性を明確にした．たとえば，ノード数 27，データ数 108 のシステムにおいて，長さ 10 のキューが高いコストパフォーマンスを持つように，ノード数，データ数とキューの大きさの最適関係が明示された．さらに，実働している WWW キャッシュ・システムから抽出された WWW アクセス記録を基に，データ検索効率のシミュレーション実験を行った．シミュレーション結果は統計的評価により得られた結果と同じ傾向を示した．このことは，統計的評価を得る際に課した，すべてのノードがいっせいに検索を行い，すべてのノー

ドがデータを取得するまで次のデータ検索を行わないという制約が，結果の有効性を左右しないことを意味している．また，データの検索が頻繁に行われるほど分散管理方式が優位に立つことが確かめられた．

本論文の階層型ネットワークは総ノード数が w^h で与えられる．このようなネットワークは階層型ネットワークとしての一般性を失ってはいないが，総ノード数が比較的小さい場合は，w や h のとりうる値が限られるので，分散管理方式での最適構造の議論はできない．ネットワークの規模が大きくなると，統計的評価における計算や，実験的評価におけるシミュレーションに要する時間が膨大なものになるため，ノード総数が 1000 を超えるような規模についての解析結果が得られていない．今後，解析方法を検討し，分散管理方式での最適構造の議論をすることを目指したい．それにより，WWW キャッシュ・システムの効率的な分散配置方法といった問題の解決に指針が与えられることが期待できる．

参　考　文　献

1) Perkins, C. E. and Bhagwat, P.: Highly Dynamic Destination-Sequenced Distance-Vector Routing(DSDV), *SIGCOMM'94* (1994).

2) 高木, 堀口, 川添, 重井: タスク型マルチプロセッサのシステム性能評価, 電子情報通信学会論文誌, Vol. J72-D-1, No. 5, pp. 357–366 (1989).

3) Mohammed, F., Ayad, N., Esmat, H., Omar, A. and Ghonaimy, M.: Performance Analysis of Access Media Protocols in Local Networks, *Proc. 2nd IEEE Symposium on Computers & Communications*, pp. 249–258 (1997).

4) T.El-Hadidi, M. and H.Hegazi, N.: Performance Analysis of the Kerberos Protocol in a Distributed Environment, *Proc. 2nd IEEE Symposium on Computers & Communications*, pp. 235–248 (1997).

5) 川村尚生, 増山博: 階層型ネットワーク上での分散データベースの性能に関する一考察, 情報処理学会論文誌, Vol. 40, No. 9, pp. 3620–3623 (1999).

6) Kawamura, T. and Masuyama, H.: Performance Analysis of a Distributed Database in Multi-Agent Architecture, *Proc. IASTED International Conference on Parallel and Distributed Computing and Systems*, pp. 161–171 (1999).

7) Kawamura, T., Inoue, M. and Masuyama, H.: Optimal Distribution of Database in Hierarchical Networks, *Proc. IASTED International Conference Parallel and Distributed Computing and Systems*, Vol. 2, pp. 536–543 (2000).

8) 川村尚生, 井上倫夫, 増山博: 階層型ネットワーク上での分散データベースの性能評価, 日本応用数理学会論文誌, Vol. 11, No. 1, pp. 15–26 (2001).

9) Shim, J., Scheuermann, P. and Vingralek, R.: Proxy Cache Design: Algorithms, Implementation and Performance, *IEEE Trans. Knowledge and Data Engineering*, Vol. 11, No. 4, pp. 549–562 (1999).

(平成 13 年 6 月 5 日受付)
(平成 14 年 5 月 15 日採録)

川村　尚生（正会員）

昭和 40 年生．平成 6 年神戸大学大学院自然科学研究科知能科学専攻博士課程退学．博士（工学）．同年鳥取大学工学部助手．マルチエージェントシステムに関する研究に従事．人工知能学会，ソフトウェア科学会，電子情報通信学会各会員．

増山　　博（正会員）

昭和 44 年広島大学大学院修士課程（電気工学専攻）修了後，広島大学助手，宮崎大学助教授・教授，Stanford・Boston 両大学客員教授および大阪大学医学系研究科教授（併任）を経て，現在，鳥取大学知能情報工学科教授．非同期論理回路，計算機アーキテクチャ等の研究を経て，現在，主に並列・分散処理システムやコンピュータ・ネットワークの高信頼化に関する研究に従事．工学博士．電子情報通信学会，IEEE 各会員．