情報処理学会論文誌 Vol. 52 No. 8 2441–2454 (Aug. 2011)

Chordネットワークに配置された配列に対する並列範囲アクセス

福地大輔[†1,†2] 本位田真一[†3,†1]

P2P ネットワークを構造化し，アドレスを指定してデータを管理できるようにしたものが，アドレッサブルネットワークである．そのアドレッサブルネットワークの 1 つである Chord ネットワークにおいて，分散ハッシュテーブルがハッシュテーブルを扱うように，配列を扱うためにつくられたのが，分散配列である．配列では，ハッシュテーブルとは違い，シーケンシャルアクセスのような複数のデータに対する操作が行われる．そのため，分散配列では，複数データにわたる操作に対する優れた手法が求められる．本論文では，通信コストが少なく，所要時間も短い範囲アクセスの手法を提案する．範囲アクセスとは，与えられた範囲に添字が属するすべての要素へのアクセスであり，配列の連続する要素を一括操作する際の基盤操作である．この範囲アクセスを，分散配列の性質に合わせて並列化する．これにより，n をピア数，w をアクセス範囲の幅として，従来の非並列な手法では通信コストと所要時間ともに $\Theta(w+\log n)$ になり，単純な並列化ではそれぞれ $\Omega(w\log n)$ と $\Theta(\log n)$ になるところを，$\Theta(w+\log n)$ の通信コストと $\Theta(\log n)$ の所要時間で実現する．この優位性を近似解析とシミュレーション実験により確認する．

Parallel Range Access in Arrays on Chord Networks

Daisuke Fukuchi[†1,†2] and Shinichi Honiden[†3,†1]

Addressable networks are created by structuring P2P networks and these networks enable address-based data management. In a similar way to distributed hash tables managing hash tables on addressable networks, distributed arrays manage arrays on Chord networks, which are a kind of addressable network. However, unlike hash tables, arrays have inter-element operations such as sequential access. Thus, efficient algorithms for such inter-element operations are desired in distributed arrays. In this paper, we propose a new range access method to reduce communication costs and shorten the required time. Range access is a basic operation for accessing all the elements whose indices are in the given ranges. We parallelize the range access with the distributed arrays. This new method archives $\Theta(w+\log n)$ in communication costs and $\Theta(\log n)$ in required time, where n is the number of peers and w is the width of the access ranges. A previous non-parallel method requires $\Theta(w+\log n)$ in communication costs and time, and a naive parallel method requires $\Omega(w\log n)$ in communication costs and $\Theta(\log n)$ in time. This superiority of the new method was verified by conducting approximation analyses and simulation experiments.

1. はじめに

P2P システムでは，ピアと呼ぶ構成コンピュータを連携させ，最大負荷を減らし，故障点をなくすことにより，スケーラビリティや匿名性を実現する．この特徴から，P2P システムは，大規模なファイル共有などに利用されている．しかし，データの管理などを容易にする集約サーバは，負荷が集まりやすく，故障点になりやすいため，純粋な P2P 環境では使用しない．そのため，データの管理は簡単ではない．実際，無秩序な P2P ネットワークでは，データを一貫して扱うために，フラッディングなどのコストの大きい手段に頼らなければならず，スケーラブルで一貫したデータ管理は実現できない．

純粋な P2P ネットワークにおいて一貫したデータ管理を行うために，**図 1** のようにアドレッサブルネットワークを用いる方法[1)–12)] が提案されている．アドレッサブルネットワークは，メモリのように，指定したアドレスへの一意なアクセスをスケーラブルに実現する．このアドレッサブルネットワークにデータを配置することにより，スケーラブルに一貫してデータを管理できる．

しかし，アドレッサブルネットワークにおけるデータ管理は，**図 2** のように，メモリ上のデータの管理とは趣きが異なる．アドレッサブルネットワークは，ピアにアドレスを割り当て，ピア間を適切につなぐことによりつくられる．そのため，ピア間で負荷を分散するように，データが散らされる．離れたアドレスにアクセスするためには，相応の通信コストが必要になる．したがって，ネットワークの構造やデータ配置に合わないデータアクセスを行うと，通信コストが増大することになる．

一様なハッシュ関数によりデータを配置する分散ハッシュテーブル[1)–4)] では，データ単

†1 東京大学
The University of Tokyo
†2 日本学術振興会特別研究員 DC
Research Fellow of the Japan Society for the Promotion of Science
†3 国立情報学研究所
National Institute of Informatics

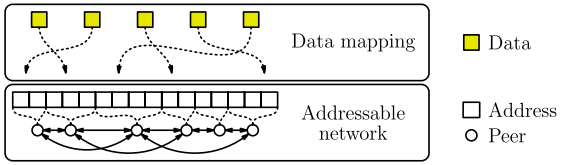


図 1　アドレッサブルネットワークを用いたデータ管理．下層のアドレッサブルネットワークでは，各アドレスにピアが対応付けられる．ピア間の接続をたどることにより，各ピアから任意のアドレスにアクセスできる．上層の配置関数では，データを下層のアドレスに割り当てる

Fig. 1　Data management using addressable networks.

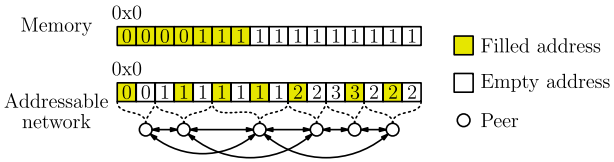


図 2　メモリとアドレッサブルネットワークの違い．アドレス内の数値は，アドレス 0x0 にアクセスした後に，そこにアクセスするために必要なコストを表す．メモリではデータを集め，ごく近くのアドレスへのアクセスは速く，他は一律に遅くなる．アドレッサブルネットワークでは，データを散らし，離れたアドレスへのアクセスコストは通過するピアの数に比例する

Fig. 2　Differences between memories and addressable networks.

体へのアクセスのみを考えることにより，この問題を回避している．

文献 5)–9) では，値の近いデータを近いアドレスに配置することにより，値の近いデータの集合へのアクセスを少ない通信コストで実現している．ただし，値の近いデータの配置箇所は散らないため，負荷分散のために，データが集まる箇所にピアを集める[8] などの対処が別途必要になる．

本論文で取り上げる分散配列[10)–12)] は，アドレッサブルネットワークの 1 つである Chord[4] ネットワーク上で，配列を扱う．配列の要素は，配列名とその何番目かを示す非負整数の添字によって特定できるデータの形式である．分散配列では，Chord ネットワークの構造に合わせ，配列の要素を配置することにより，シーケンシャルアクセス[⋆1] と二分探索のような探索，そして本論文で取り上げる範囲アクセスを少ない通信コストで実現している．また，

⋆1 任意の非負整数 i，j（ただし，$i \leq j$）について，添字が i の要素から j の要素まで添字順に 1 つずつアクセスする．

表 1　各範囲アクセス手法の性能

Table 1　Performance of range access methods.

	通信コスト	所要時間
提案手法	$\Theta(\log n + w)$	$\Theta(\log n)$
単純な並列化	$\Omega(w \log n)$	$\Theta(\log n)$
既存の非並列な手法	$\Theta(\log n + w)$	$\Theta(\log n + w)$

添字が連続する要素は離れたアドレスに配置されるため，負荷も分散できる．

配列は汎用的なデータ形式であるだけでなく，P2P 環境では，負荷分散のためにも利用できる．たとえば，ファイル共有において，共有されたファイルの名前とサイズなどを登録日時順に保存していくとする．このデータを，負荷分散のために，古い順に一定サイズごとに区切るとする．区切られた 1 つ 1 つは配列の要素，全体は配列として扱える．また，データを複製して負荷分散する場合にも，個々の複製を配列の要素として，分散配列を利用できる．

分散配列が提供する操作として，範囲アクセスがある．範囲アクセスでは，任意の非負整数 i，j（ただし，$i \leq j$）について，配列の添字が i から j までの要素に 1 度にアクセスする．範囲アクセスは，添字が連続する要素にアクセスするための基盤操作として，前述の共有ファイル履歴を区切った配列から該当項目を検索する場合や複製間の同期を行う場合などに利用できる．

分散配列において，範囲アクセスの通信コストは，ピア数を n，アクセス範囲の幅を w として，$\Theta(\log n + w)$ にできる[10),11)]．しかし，その手法は対象要素に 1 つずつアクセスする方法であるため，所要時間も $\Theta(\log n + w)$ となる．

範囲アクセスでは，シーケンシャルアクセスのように 1 要素ずつアクセスしていく必要はないため，並列化により，所要時間を小さくすることができる．実際，単純に対象要素ごとに並列にアクセスすることにより，所要時間は $\Theta(\log n)$ になる．しかし，この単純な並列化では，通信コストが $\Omega(w \log n)$ に増大する．

本論文では，分散配列における Chord ネットワークの構造と配列要素の配置規則の特徴に合わせて，範囲アクセスを並列化する．それにより，通信コストは $\Theta(\log n + w)$ のまま，所要時間を $\Theta(\log n)$ にする（**表 1**）．

本論文では，まず，前提となる分散配列について，下層の Chord ネットワーク（2.1 節），上層の配列要素の配置規則（2.2 節），最後に，分散配列での範囲アクセス（2.3 節）の順に説明する．その後，提案手法（3 章）を示し，その優位性を理論（4 章）と実験（5 章）の

表 2　記号，用語
Table 2　Symbols and terms.

b	アドレスおよび添字の上限 2^b-1 を決める非負整数
n	ピア数
w	アクセス対象範囲の幅
$A_{s,t}$	$t2^s$ 以上 $(t+1)2^s$ 未満の整数集合
$c_{x,y}$	ピア x からアドレス y へのアクセスに必要な引き継ぎの回数
$d(x,y)$	アドレス環での x から y への距離を求める関数（式 (3)）
$g(x)$	添字が x の要素の配置アドレスを求める簡易版関数（式 (6)）
$r_i(x)$	x の i-ビット列の逆読み関数（式 (5)）
$u_{i,j}(x)$	x の i-ビット列の上位 j 桁に現れる 1 の数を求める関数（式 (4)）
$\mathrm{pre}(x)$	アドレス環で x の手前のピアを求める関数（式 (1)）
$\mathrm{suc}(x)$	アドレス環で x の先のピアを求める関数（式 (2)）
アドレス環	アドレス空間を円環と見なしたもの
i-ビット列	2 進数表記を i 桁に合わせたもの

両面から裏付ける．

2. 前　　提

本章では，本論文の理解に必要な前提について説明する．まず，分散配列[10)–12)]が使用するアドレッサブルネットワークである Chord[4)] ネットワークについて説明する．次に，分散配列における Chord ネットワークへの配列要素の配置について説明する．最後に，分散配列での範囲アクセスについて説明する．

2.1 Chord ネットワーク

本論文では，分散配列[10)–12)]で用いられるものと同様の Chord[4)] ネットワークを用いる．本節では，その Chord ネットワークの構成とアドレスアクセスの性質をあげる．

定義 1（Chord ネットワークの構成）．

(1)　アドレス空間

- 十分に大きな非負整数 b を用い，整数範囲 $[0,2^b)$ をアドレス空間とする．多くの場合，b として 128 以上の値を用いるため，アドレス空間はきわめて大きいと考えてよい．
- アドレス空間を 2^b-1 の次に 0 が続く円環と見なす．これを**アドレス環**と呼ぶことにする．以降では，$2^b>x>y$ なる任意の非負整数 x，y について，アドレスの範囲 $[x,y)$ は $[x,2^b)\cup[0,y)$ を表すものとする．

(2)　アドレスの担当を決定するルール

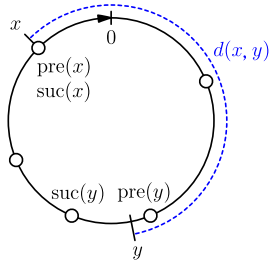


図 3　アドレス環．アドレス x から y への距離 $d(x,y)$ は正方向に測る
Fig. 3　Address ring.

- ピアを重複しないようにアドレス空間に配置する．この配置には，多くの場合，一様なハッシュ関数が用いられる．以降，x をピアが配置されたアドレスとして，ピア自体を x で表す．
- ピアが 1 つ以上存在する場合，任意のアドレス x について，**図 3** のように，アドレス環において，x か x より正方向に手前で最初のピアを $\mathbf{pre}(x)$ とする．x か x より正方向に先で最初のピアを $\mathbf{suc}(x)$ とする．

$$\mathrm{pre}(x):=\begin{cases}x & (\text{ピア } x \text{ が存在})\\ \mathrm{pre}((x-1) \bmod 2^b) & (\text{それ以外})\end{cases} \tag{1}$$

$$\mathrm{suc}(x):=\begin{cases}x & (\text{ピア } x \text{ が存在})\\ \mathrm{suc}((x+1) \bmod 2^b) & (\text{それ以外})\end{cases} \tag{2}$$

- 任意のアドレス x について，$\mathrm{pre}(x)$ を x の担当とする．逆に，任意のピア x について，x に割り当てられるアドレスの範囲は $[x,\mathrm{suc}((x+1) \bmod 2^b))$ となる．ここが，オリジナルの Chord ネットワークと大きく異なる点である[10)–12)]．この違いの詳細および文献 10)–12) で言及されていない影響を 6 章に提示する．ただし，6 章までこの点を意識する必要はない．

(3)　ピア間接続のルール

- 任意のピア x について，x から $\mathrm{suc}((x+1) \bmod 2^b)$ と $\mathrm{pre}((x+2^i) \bmod 2^b)$（ただし，$i=1,2,\ldots,b-1$）へ接続する．ただし，$x$ 自身は除く．ここも，オリジナルの Chord ネットワークとは異なる[10)–12)]が，本論文では意識する必要はない．

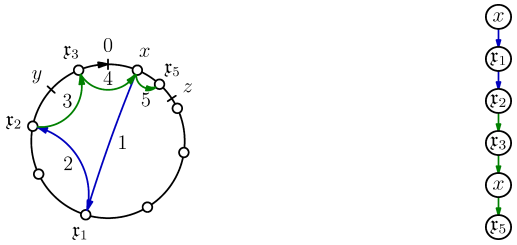


図 4　ピア x からアドレス y へアクセスし，その後，アドレス z へアクセスする操作における処理の引き継ぎ．左図はアドレス環における引き継ぎの経路を示す．数値は引き継ぎの発生した順番を表す．右図は関係するピアと引き継ぎの関係だけを抜き出したものである

Fig. 4　Relays to access addresses y and z one by one from peer x.

(4)　アクセス手続き

- 任意のアドレス x，y について，x から y への距離として，図 3 のように，アドレス環における正方向の距離 $(y-x) \bmod 2^b$ を用いる．これを $\boldsymbol{d}(x,y)$ で表す．
$$d(x,y) := (y-x) \bmod 2^b \tag{3}$$
- 以下を繰り返す．アクセス処理をしているピアを x，目標のアドレスを y とする．x が y の担当であれば，x 内で処理は終了する．x が y の担当でなければ，上記のピア間接続のルールにおいて，x から接続した先のピアの集合を F とする．F の中で，y への距離が最小になるピアへアクセス処理を引き継がせる．

簡単のため，上記の定義 1 はピアの参加・離脱に対応するための手続きを含まない．ただし，一般的な Chord ネットワークと同様にして，ピアの参加・離脱に対応することができる．ピアの参加・離脱とは，時間経過により，アドレス環からピアが消えたり，逆に新たに配置されたりすることである．これにより，関係するアドレスへの pre と suc の適用結果が変わる．そのため，ピア間接続のルールで張った接続が，時間経過に従い，存在しないピアを指したり，指すべきピアとは異なるピアを指したりすることになる．本論文では，5 章と 6 章まで，ピアの参加・離脱に関する部分は省略する．

本論文では，定義 1 のアクセス処理の引き継ぎの際に，そのアクセス手続きを始めた操作の処理も引き継ぐとする．つまり，操作の中でアドレスへのアクセスが行われた後，処理を元のピアに戻さない．そのうえで，通信コストの評価指標として，**引き継ぎの回数**を用いる．所要時間の評価指標として，引き継ぎによってできる経路の深さ（以降，**引き継ぎの深さ**と呼ぶ）を用いる．たとえば，**図 4** のように，アクセス処理を 2 回引き継ぐアクセス手続きと 3 回引き継ぐアクセス手続きが 1 つずつ行われる操作の引き継ぎの回数と深さはともに 5 である．この例では，アクセスを 1 つずつ行うため，引き継ぎの回数と深さが一致するが，2.3 節以降に登場する並列化が行われる場合，引き継ぎの回数と深さは一致しなくなる．

Chord ネットワークにおいて，任意のアドレス x へのアクセスに必要な引き継ぎの回数は，x への距離の二進数表記に現れる 1 の数により見積もることができる．正確には，次の仮定 1 のもとで下記の定理 1 が成り立つ[10)–12)]．

仮定 1. ピア数 n は 2^b 以下の 2 のべき乗である．さらに，ピアはアドレス環に等間隔に配置される．

x をピア，y をアドレスとして，x から y へのアクセスに必要な引き継ぎの回数を $\boldsymbol{c}_{x,y}$ で表す．i を非負整数として，2^i 未満の非負整数の二進数表記を，短ければ上位に 0 を加えて i 桁にし，**i-ビット列**ということにする．つまり，任意の 2^i 未満の非負整数 x について，x の i-ビット列が $x_i x_{i-1} \ldots x_1$ であれば，

$$(\forall k = 1,2,\ldots,i \;\; x_k \in \{0,1\}) \wedge x = \sum_{k=1}^{i} x_k 2^{k-1}$$

である．i を非負整数，j を i 以下の非負整数として，$\boldsymbol{u}_{i,j}$ を i-ビット列の上位 j 桁に現れる 1 の数を求める関数とする．つまり，x を 2^i 未満の非負整数，$x_i x_{i-1} \ldots x_1$ を x の i-ビット列として，

$$u_{i,j}(x) := \sum_{k=i-j+1}^{i} x_k \tag{4}$$

である．

定理 1（Chord ネットワークのアクセス性能[10)–12)]）．任意の 2^b 未満の非負整数 x，y について，

$$c_{x,y} = u_{b,\log n}(d(x,y))$$

である．

実際，仮定 1 により，**図 5** のように，任意のピア x の接続先は x からちょうど 2 のべき乗だけ離れたピア $(x+2^i) \bmod 2^b$（ただし，$i = b - \log n, b - \log n + 1, \ldots, b-1$）になる．そのため，引き継ぎの回数は，**図 6** のように，開始ピアを x，目標アドレスを y とすると，$d(x,y)$ の b-ビット列の上位 $\log n$ 桁に現れる 1 の数となる．

また，定理 1 は，アクセスに必要な引き継ぎの回数が開始ピアと目標アドレスの距離という相対的位置関係のみに依存することを示す．

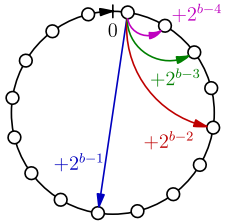


図 5　仮定 1（$n = 2^4$）の上でのピア配置とピア間接続．接続先までの距離がちょうど 2 のべき乗になる
Fig. 5　Peer mapping and connections from peer on Assumption 1 when $n = 2^4$.

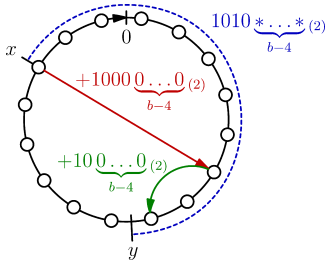


図 6　仮定 1（$n = 2^4$）の上でのアクセス手続き．開始ピア x から目標アドレス y までの距離は $2^{b-1} + 2^{b-3}$ 以上 $2^{b-1} + 2^{b-3} + 2^{b-4}$ 未満である．そのため，長さ 2^{b-1} の接続と長さ 2^{b-3} の接続を順に使用し，x から y にアクセスする
Fig. 6　Access procedure on Assumption 1 when $n = 2^4$.

仮定 1 が成り立たなくても，任意の数のピアを一様なハッシュ関数によりアドレス空間に配置する場合，定理 1 におおよそ従うことが実験により示されている[10)–12)]．

2.2　分散配列における Chord ネットワークへの配列要素の配置

本節では，分散配列[10)–12)] で用いられる配列の配置規則と，それによるデータの分散性および要素間アクセスの性質をあげる．

分散配列では下記の定義 2 に従い，アドレス環へ配列要素を配置する．

a を配列名，x を非負整数として，配列 a の添字 x の要素を $a[x]$ と書く．以降，添字を 2^b 未満に制限するが，2^b はきわめて大きいため，問題はない．i を非負整数として，$\boldsymbol{r}_i$ を i-ビット列の逆読み関数とする．つまり，x を 2^i 未満の非負整数，$x_i x_{i-1} \ldots x_1$ を x の i-ビット列として，

$$r_i(x) := \sum_{k=1}^{i} x_k 2^{i-k} \tag{5}$$

である．

定義 2（配列の配置規則）．a を配列名，x を 2^b 未満の非負整数として，$a[x]$ を次のアドレス $\boldsymbol{f}(a[x])$ に配置する．

$$f(a[x]) := (h(a) + r_b(x)) \bmod 2^b$$

ただし，h は配列名からアドレスへの一様なハッシュ関数である．

定義 2 は，$h(a)$ を除いて考えると，配列要素をその添字の b-ビット列を逆読みしてできるアドレスに配置することを意味する．$h(a)$ は，複数の配列を扱う場合に，配列ごとに使用するアドレスをずらすためにある．以降では，簡単のため，配列名を考えず，添字が x の要素を次の $\boldsymbol{g}(x)$ に配置するとして論じる．

$$g(x) := r_b(x) \tag{6}$$

1 つの配列における要素間の相対的位置関係は f でも g でも変わらないため，以降の g についての理論は f についても成り立つ．

実数列 $g(0)/2^b, g(1)/2^b, \ldots$ は van der Corput 列[13)] と呼ばれ，$[0, 1)$ の範囲に一様に分布する[13),14)]．したがって，定義 2 により，添字が連続する配列の要素はアドレス環に一様に分布する．本論文では，この性質を下記の定理 2 の形で使用する．

B を $[0, 2^b)$ の整数部分集合として，$\{g(x) \mid x \in B\}$ を $g(B)$ で表す．s，t を整数として，整数集合 $[t2^s, (t+1)2^s)$ を $\boldsymbol{A}_{s,t}$ で表す．

定理 2（分散性能[10)–12)]）．b 以下の任意の非負整数 s，2^{b-s} 未満の任意の非負整数 t について，

$$\exists y \in A_{b-s,0} \;\; g(A_{s,t}) = \{x2^{b-s} + y \mid x \in A_{s,0}\}$$

である．

定理 2 は，添字が $A_{s,t}$ の形の範囲に属する要素の集合が等間隔なアドレスに配置されることを示す．**図 7** がその例である．

定理 2 より，b 以下の任意の非負整数 s，2^{b-s} 未満の任意の非負整数 t について，添字が $A_{s,t}$ に属する 2 つの要素が配置されるアドレスを x，y とすると，$d(x, y)$ は 2^{b-s} の倍数となる．したがって，$d(x, y)$ の b-ビット列の下位 $b - s$ 桁は 0 である．よって，定理 1 より，任意の要素にアクセスした後，そこから別の要素にアクセスするために必要な引き継ぎの回数は，次の近似 1 のもとで，下記の定理 3 として評価できる[10)–12)]．

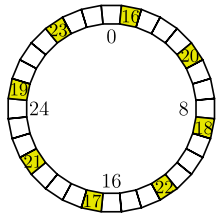


図 7　定義 2 による配置（$b = 5$）．添字が $[16, 24)$ に属す要素が等間隔に配置される
Fig. 7　Mapping by Definition 2 when $b = 5$.

近似 1. 任意のアドレス x について，$\mathrm{pre}(x)$ を x で近似する．

近似 1 は，任意のアドレスを，pre を適用せずに，そのまま担当として扱えることを意味する．たとえば，任意のアドレス x，y について，x にアクセスした後の y へのアクセスは，正確には x の担当 $\mathrm{pre}(x)$ からアドレス y へのアクセスである．このアクセスを，$\mathrm{pre}(x)$ は x であるとして，ピア x からアドレス y へのアクセスと見なせる．

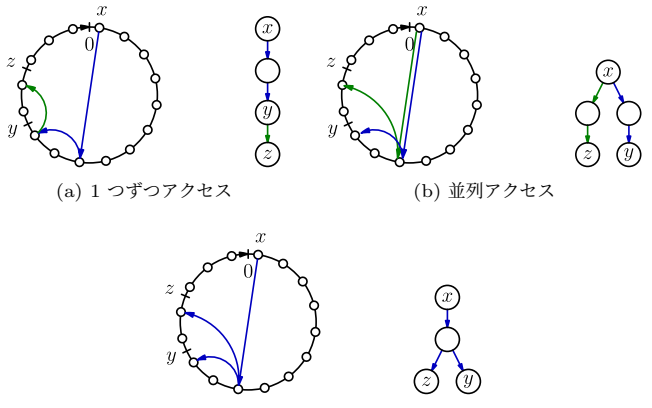
定理 3（要素間アクセスの性能[10)–12)]）．2^b 未満の任意の異なる非負整数 x，y について，x，$y \in A_{s,t}$ なる非負整数 s，t の中で最小の s を s' として，$c_{g(x),g(y)}$ は最大で $\min(s', \log n)$ である．

2.3　分散配列における範囲アクセス

分散配列では，$\Theta(\log n + w)$ 回の引き継ぎで範囲アクセスを実現している[10),11)]．その要点は次のようなものである．定義 2 の配置規則により，定理 2 に示され，図 7 に見られるように，b 以下の任意の非負整数 s，2^{b-s} 未満の任意の非負整数 t について，添字が $A_{s,t}$ に属する要素の集合は 2 のべき乗の間隔に配置される．また，Chord ネットワークを構成するピア間接続は，図 5 に見られるように，2 のべき乗の長さである．したがって，添字が $A_{s,t}$ に属する要素を管理するピアの集合を考えると，その集合内でピアは接続し合っている．そこで，アクセス範囲を $A_{s,t}$ の形を基準に区切り，区切ってできた各範囲の中では配置されるアドレス順に要素にアクセスすることにより，少ない引き継ぎですべての対象要素にアクセスすることができる．

しかし，この範囲アクセスは，**図 8** (a) のように，対象要素に 1 つずつアクセスする方法で実現されているため，引き継ぎの深さは，引き継ぎの回数と等しく，$\Theta(\log n + w)$ である．

範囲アクセスでは，1 要素ずつアクセスしていく必要はないため，図 8 (b) のように，処理を並列に行うことも許される．また，そのとき，図 8 (b) の引き継ぎ経路が図 8 (c) のものになるように，引き継ぎ元と引き継ぎ先が一致する引き継ぎをまとめることができる．本



(a) 1 つずつアクセス　(b) 並列アクセス

(c) マルチキャスト

図 8　ピア x からアドレス y, z へのアクセス．それぞれ，左図はアドレス環での引き継ぎを示す．右図は関係するピアと引き継ぎの関係だけを抜き出したものである．y，z を担当のラベルとして用いている．(a) での引き継ぎの深さは 3 だが，(b) のように並列化することにより，2 になる．さらに，(c) のように一致する引き継ぎをまとめることにより，引き継ぎの回数が 4 から 3 になる
Fig. 8　Access for addresses x and y from peer x.

論文では，このように，1 つ以上の目標アドレスへのアクセスを，並列に，かつ，引き継ぎ元と引き継ぎ先が一致する場合はその引き継ぎをまとめて行う方法をマルチキャストという．マルチキャストは定義 1 のアクセス手続きにならい，次のように定義できる．

- 以下を繰り返す．マルチキャスト処理をしているピアを x，目標のアドレスの集合を B とする．B から x に割り当てられるアドレスを除いた集合を B' とする．

  $$B' := B \setminus [x, \mathrm{suc}((x+1) \bmod 2^b))$$

  B' が空集合であれば，x 内で処理は終了する．B' が空集合でなければ，定義 1 のピア間接続のルールにおいて，x から接続した先のピアの集合を F とする．F に属するすべてのピア y について，B' に属すアドレスの中で，y からそのアドレスへの距離が，F に属するピアからそのアドレスへの距離の中で最小となるアドレスの集合を B_y とする．

  $$B_y := \{z \mid z \in B' \wedge \forall y' \in F(d(y, z) \le d(y', z))\}$$

  B_y が空集合でなければ，y へ B_y へのマルチキャスト処理を引き継がせる．

これは，$|B| = 1$ のとき，定義 1 のアクセス手続きと同じ挙動になる．また，個々の目標アドレスへの引き継ぎの経路は，他の目標アドレスには依存していないため，そのアドレス

への定義 1 のアクセス手続きにおける引き継ぎの経路と一致する．

単純に，範囲アクセスの開始ピアから対象要素が配置されるアドレスにマルチキャストすることにより，引き継ぎの回数は $\Omega(w \log n)$，深さは $\Theta(\log n)$ になる．証明は 4.2 節に示す．この手法は，引き継ぎの深さに関しては最適であるが，引き継ぎの回数は既存手法の $\Theta(\log n + w)$ に劣る．

3. 提案手法

並列化により，範囲アクセスに必要な引き継ぎの深さを減らし，さらに，引き継ぎの回数も少なく抑える手法を提案する．本手法により，引き継ぎの回数は $\Theta(\log n + w)$，深さは $\Theta(\log n)$ になる．証明は 4.1 節に示す．

本手法では，分散配列における既存の非並列な範囲アクセス[10] を，その要点を活かしたまま並列化する．b 以下の任意の非負整数 s，2^{b-s} 未満の任意の非負整数 t について，添字が $A_{s,t}$ に属する要素を管理するピアの集合を考えると，その集合内でピアは接続し合っている．そこで，アクセス範囲を $A_{s,t}$ の形を基準に区切り，区切ってできた各範囲に対し，まず，いずれかの要素にアクセスし，そこからその範囲の残りの要素が配置されるアドレスにマルチキャストする．これにより，引き継ぎの回数を抑えることができる．これは，2.3 節の単純な並列化による**図 9** (a) のような引き継ぎ経路を図 9 (b) のように変えることにより，引き継ぎの回数を減らしていると見ることもできる．

具体的に，提案手法として，次の定義 3 を与える．

定義 3（並列範囲アクセス手法）．開始ピアを x，アクセスする添字の範囲を R とする．

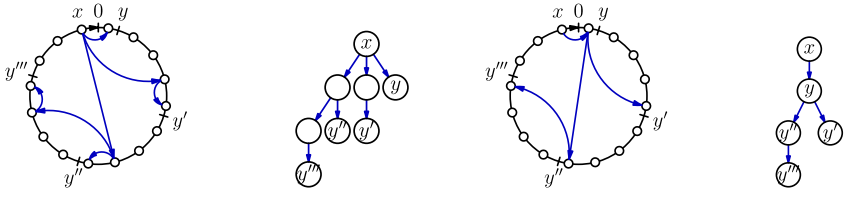
Case 1.　$R = A_{s,t}$ なる非負整数 s，t が存在する場合，$g(R)$ の中で x からの距離が最小のアドレスを y とする．まず，x から y にアクセスし，そこから $g(R) \setminus \{y\}$ にマルチキャストする．

Case 2.　$A_{s-1,2t} \subset R \subset A_{s,t}$ なる正整数 s，非負整数 t が存在する場合，$g(A_{s-1,2t})$ の中で x からの距離が最小のアドレスを y とする．まず，x から y にアクセスし，そこから $g(R) \setminus \{y\}$ にマルチキャストする．

Case 3.　$A_{s-1,2t+1} \subset R \subset A_{s,t}$ なる正整数 s，非負整数 t が存在する場合，Case 2 の動作を，$A_{s-1,2t}$ を $A_{s-1,2t+1}$ で置き換えて行う．

Case 4.　それ以外の場合，$R \subset A_{s,t}$ なる正整数 s，非負整数 t の中で最小の s を s'，そのときの t を t' とする．R を $R \cap A_{s'-1,2t'}$ と $R \cap A_{s'-1,2t'+1}$ に分け，それぞれ R_1，R_2 とする．R_1 は Case 1 か Case 3 に，R_2 は Case 1 か Case 2 にあてはまる．それぞれの場合に従い，最初にアクセスするアドレス y'，y'' を求める．まず，x から $\{y', y''\}$ にマルチキャストし，y' に到達したら，そこから $g(R_1) \setminus \{y'\}$ に，y'' に到達したら，そこから $g(R_2) \setminus \{y''\}$ にマルチキャストする．

図 10 が定義 3 の Case 4 の例である．



(a) x からのマルチキャスト　　(b) pre(y) からのマルチキャスト

図 9　ピア x から等間隔なアドレス y，y'，y''，y''' へのアクセス．それぞれ，左図はアドレス環での引き継ぎを示す．右図は関係するピアと引き継ぎの関係だけを抜き出したものである．y，y'，y''，y''' を担当のラベルとして用いている

Fig. 9　Access for even interval address set $\{y, y', y'', y'''\}$ from peer x.

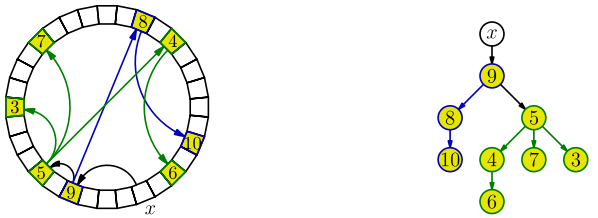


図 10　定義 3 によるピア x から添字 $[3, 10]$ への範囲アクセス（$b = 4$ かつ $n = 2^4$）．左図はアドレス環での引き継ぎを示す．右図は関係するピアと引き継ぎの関係だけを抜き出したグラフである．要素の添字を担当のラベルとして用いている．$[3, 10]$ は $[3, 8)$ と $[8, 11)$ に分けられる．$[3, 8)$ には $[4, 8)$ が含まれ，$[8, 11)$ には $[8, 10)$ が含まれる．まず，$g([4, 8))$ と $g([8, 10))$ のそれぞれにおいて x から最も近い $g(5)$ と $g(9)$ にマルチキャストする．その後，$g(5)$ から $g([3, 8))$ に，$g(9)$ から $g([8, 11))$ にマルチキャストする

Fig. 10　Range access of Definition 3 for range $[3, 10]$ from peer x when $b = 4$ and $n = 2^4$.

4. 理論評価

本章では，仮定 1 と近似 1 のもとで，定義 3 の並列範囲アクセスに必要な引き継ぎの回数が $\Theta(\log n + w)$，深さが $\Theta(\log n)$ であることを示す．また，2.3 節の単純な並列化の場合，引き継ぎの回数が $\Omega(w \log n)$，深さが $\Theta(\log n)$ であることを示す．

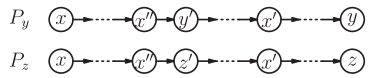


図 11　補題 2 の証明内の仮定
Fig. 11　Assumption in proof of Lemma 2.

以下では，x をピア，B をアドレスの集合として，x から B へのマルチキャストに必要な引き継ぎの回数を $c_{x,B}$，深さを $\boldsymbol{l}_{x,B}$ と表す．マルチキャストでも，個々の目標アドレスへの引き継ぎ経路は，そのアドレスへのアクセスにおける引き継ぎ経路と同じであるため，

$$c_{x,B} \leq \sum_{y \in B} c_{x,y} \tag{7}$$

$$l_{x,B} = \max_{y \in B}(c_{x,y}) \tag{8}$$

である．

4.1　提案手法の場合

まず，定義 3 の各 Case において最初のアドレスへのアクセス後に行われるマルチキャストに必要な引き継ぎの回数が目標アドレスの数を超えないことを，次の補題 1 として示す．

補題 1. b 以下の任意の正整数 s，2^{b-s} 以下の任意の非負整数 t，$A_{s-1,2t} \subseteq R \subset A_{s,t}$ なる任意の範囲 R について，

$$\forall x \in g(A_{s-1,2t})\ \ c_{x,g(R)\setminus\{x\}} \leq |R| - 1$$

である．等号は $\log n \geq s$ のとき成り立つ．$A_{s-1,2t}$ を $A_{s-1,2t+1}$ にした場合も同様である．

補題 1 の証明のために，まず，次の補題 2 を証明する．

補題 2. 任意のピア x，任意のアドレス y，z について，x から y へのアクセスにおける引き継ぎ経路と x から z へのアクセスにおける引き継ぎ経路に同じピアが現れる場合，x からそのピアまでの経路は一致する．

証明．背理法により証明する．x から y へのアクセスの引き継ぎ経路を P_y，x から z へのアクセスの引き継ぎ経路を P_z と呼ぶ．P_y と P_z の両方に現れる任意のピアを x' とする．P_y における x から x' までの経路と P_z における x から x' までの経路に異なる部分があると仮定する．**図 11** のように，P_y における x から x' までの経路と P_z における x から x' までの経路において，最初に異なるピアをそれぞれ y' と z' とする．y'，z' の前の共通するピアを x'' とする．$d(x'', y') > d(x'', z')$ として，一般性を失わない．

P_y において，y' は x'' の 2 のべき乗先のピアの中で y に最も近いピアであるため，$2^m \leq d(x'', y) < 2^{m+1}$ なる非負整数 m を用いて，$d(x'', y') = 2^m$ である．よって，$d(x'', y') \leq d(x'', x')$ より，$2^m \leq d(x'', x')$ となる．

P_z において，同様に，$2^{m'} \leq d(x'', z) < 2^{m'+1}$ なる非負整数 m' を用いて，$d(x'', z') = 2^{m'}$ である．また，$d(x'', x') \leq d(x'', z)$ より，$d(x'', x') < 2^{m'+1}$ となる

$2^m = d(x'', y') > d(x'', z') = 2^{m'}$ より，$m \geq m' + 1$ であるため，前段と前々段は矛盾する．　□

補題 1 の証明．マルチキャストの引き継ぎ経路において，1 つのピアは 1 度しか現れない．なぜなら，マルチキャストの引き継ぎ経路にピア y が 2 回以上現れると仮定すると，補題 2 より，開始ピアから y までの経路は一致する．したがって，y までの経路はマルチキャストにより 1 本にまとめられていなければならない．

また，マルチキャストにおける個々の目標アドレスへの引き継ぎ経路は，そのアドレスへのアクセスにおける引き継ぎ経路と同じであるため，マルチキャストの引き継ぎ経路には，個々の目標アドレスへのアクセスの引き継ぎ経路に現れるピアしか現れない．

よって，開始ピアから個々の目標アドレスへのアクセスの引き継ぎ経路に，開始ピアと目標アドレス（近似 1 によりピアと見なす）以外は現れないならば，開始ピアから目標アドレスへのマルチキャストに必要な引き継ぎの回数が目標アドレスの数を超えることはない．つまり，$g(A_{s-1,2t})$ に属する任意のアドレス x，$g(A_{s,t})$ に属する任意のアドレス y について，x から y への経路上にあるアドレスが $g(A_{s-1,2t})$ に属するか y であれば，補題 1 が成り立つ．以下，これを示す．

$\mathfrak{y}_0 := x$，$\mathfrak{y}_{c_{x,y}} := y$ として，x から y へのアクセスにおける引き継ぎ経路に現れるピアを，順に $\mathfrak{y}_0, \mathfrak{y}_1, \ldots, \mathfrak{y}_{c_{x,y}}$ とする．$i = 0, 1, \ldots, c_{x,y}$ について，

$$\mathfrak{y}_i \in g(A_{s-1,2t}) \cup \{y\} \tag{9}$$

を帰納的に示す．

k を $c_{x,y}$ 未満の非負整数として，i が k のとき，式 (9) が成り立つと仮定する．$k < c_{x,y}$ より，$\mathfrak{y}_k \neq y$ であり，$2^m \leq d(\mathfrak{y}_k, y) < 2^{m+1}$ なる $b - \log n$ 以上 b 未満の非負整数 m が存在する．$\mathfrak{y}_{k+1}$ は $\mathfrak{y}_k$ の 2 のべき乗先のピアの中で y に最も近いピアであるため，$d(\mathfrak{y}_k, \mathfrak{y}_{k+1}) = 2^m$ である．また，$\mathfrak{y}_k$，$y \in g(A_{s,t})$ と定理 2 より，$d(\mathfrak{y}_k, y)$ は 2^{b-s} の倍数である．ただし，$\mathfrak{y}_k \neq y$ より 0 ではない．よって，$d(\mathfrak{y}_k, y) \geq 2^{b-s}$ となる．したがって，$2^{b-s} \leq d(\mathfrak{y}_k, y) < 2^{m+1}$ より $m - b + s \geq 0$ である．

(i) $m - b + s \geq 1$ のとき，$d(\mathfrak{y}_k, \mathfrak{y}_{k+1}) = 2^{m-b+s-1}2^{b-s+1}$ より，$\mathfrak{y}_k$ から $\mathfrak{y}_{k+1}$ への距離

は $2^{b-(s-1)}$ の倍数である．よって，$\mathfrak{y}_k \in g(A_{s-1,2t})$ と定理 2 より，$\mathfrak{y}_{k+1} \in g(A_{s-1,2t})$ となる．

(ii) $m-b+s=0$ のとき，$d(\mathfrak{y}_k, \mathfrak{y}_{k+1}) = 2^{b-s}$ である．よって，$\mathfrak{y}_k \in g(A_{s,t})$ と定理 2 より，$\mathfrak{y}_{k+1} \in g(A_{s,t})$ となる．さらに，$y \in g(A_{s,t})$ と定理 2 より，$d(\mathfrak{y}_{k+1}, y)$ は 2^{b-s} の倍数である．また，$d(\mathfrak{y}_k, y) = d(\mathfrak{y}_k, \mathfrak{y}_{k+1}) + d(\mathfrak{y}_{k+1}, y)$ より，$d(\mathfrak{y}_{k+1}, y) < 2^{m+1} - 2^m = 2^{b-s}$ である．よって，$d(\mathfrak{y}_{k+1}, y) = 0$ となる．つまり，$\mathfrak{y}_{k+1} = y$ である．

以上と $\mathfrak{y}_0 \in g(A_{s-1,2t})$ より，$i = 0, 1, \ldots, c_{x,y}$ に対して式 (9) は成り立つ．

以上の証明は，$A_{s-1,2t}$ を $A_{s-1,2t+1}$ にしても成り立つ．　□

次に，定義 3 の各 Case について，引き継ぎの回数と深さを求める．定義 3 および対応する各 Case の中で定義されている記号 x，R，s，t，y，R_1，R_2 はそのまま使用する．簡単のため，$s < \log n$ と仮定する．

Case 1.　定理 2 より，$g(R)$ はアドレス環に 2^{b-s} 間隔で並ぶアドレスの集合である．そのため，開始ピア x から $g(R)$ の中で x から最も近いアドレス y までの距離 $d(x,y)$ は 2^{b-s} 未満である．つまり，$d(x,y)$ の b-ビット列の上位 s 桁は 0 である．よって，$c_{x,y}$（定理 1 より $u_{b,\log n}(d(x,y))$）は最大で $\log n - s$ である．$s = \log |R|$ より，これは，

$$\log n - \log |R| \tag{10}$$

である．$c_{y,g(R)\setminus\{y\}}$ は，補題 1 より，$|R|-1$ である．よって，引き継ぎの回数は最大で，

$$\log n - \log |R| + |R| - 1 \tag{11}$$

となる．$l_{y,g(R)\setminus\{y\}}$ は，定理 3 と式 (8) より，s である．$s = \log |R|$ より，式 (10) と足し合わせ，引き継ぎの深さは最大で，

$$\log n \tag{12}$$

となる．

Case 2.　Case 1 と同様にして，$c_{x,y}$ は最大で $\log n - s + 1$ である．$\log |R| < s$ より，これは，

$$\log n - \log |R| + 1 \tag{13}$$

以下である．$c_{y,g(R)\setminus\{y\}}$ は，補題 1 より，$|R|-1$ である．よって，引き継ぎの回数は，

$$\log n - \log |R| + |R| \tag{14}$$

以下となる．$l_{y,g(A_{s,t})\setminus\{y\}}$ は，定理 3 と式 (8) より，s である．よって，$l_{y,g(R)\setminus\{y\}}$ は，s 以下である．$s - 1 < \log |R|$ より，式 (13) と足し合わせ，引き継ぎの深さは，

$$\log n + 2 \tag{15}$$

以下となる．

Case 3.　Case 2 と同様である．

Case 4.　式 (7) より，引き継ぎの回数は式 (14) の R を R_1，R_2 に置換した式の和，

$$2 \log n - \log |R_1||R_2| + |R_1| + |R_2|$$

以下である．$|R_1| + |R_2| = |R|$ かつ $1 \le |R_1| \le |R| - 1$ より，これは，

$$2 \log n - \log (|R| - 1) + |R| \tag{16}$$

以下となる．引き継ぎの深さは式 (15) 以下である．

以上より，次の定理 4 が成り立つ．

定理 4. ピア数を n，アクセス範囲の幅を w として，定義 3 の範囲アクセスの引き継ぎの回数は $\Theta(\log n + w)$，深さは $\Theta(\log n)$ である．

証明．引き継ぎの回数は，$|R| = w$ として，式 (11) より $\Omega(\log n + w)$，式 (11) と式 (14) と式 (16) より $O(\log n + w)$ である．引き継ぎの深さは，式 (12) より $\Omega(\log n)$，式 (12) と式 (15) より $O(\log n)$ である．　□

4.2　単純な並列化の場合

まず，引き継ぎの深さについて考える．アクセスする添字の範囲を R，開始ピアを x とする．x から $g(R)$ へのマルチキャストであるため，必要な引き継ぎの深さは $l_{x,g(R)}$ である．式 (8) および定理 1 より，この $l_{x,g(R)}$ は最大で，

$$\log n \tag{17}$$

である．

次に，引き継ぎの回数について考える．

まず，アクセスする添字の範囲 R に対し，$R = A_{s,t}$ なる非負整数 s，t が存在する場合，**図 12** のように，マルチキャストによりまとめられる引き継ぎがない部分を分けることができる．これを，下記の補題 3 として示す．

x をピア，y をアドレスとして，x から y へのアクセスにおける引き継ぎ経路において，y までの距離が最初に 2^i 未満（ただし，i は非負整数）になるピアを $\boldsymbol{q}_{x,y,i}$ と表すことにする（後続との境が分かりにくいときは $q_{\{x,y,i\}}$ とも書く）．図 6 のように，2 のべき乗の長さの接続を用いて貪欲に目標アドレスへ近づくため，$q_{x,y,i}$ から y までの距離は $d(x,y) \bmod 2^i$ である．したがって，$y = (x + d(x,y)) \bmod 2^b$ であることから，

$$q_{x,y,i} := \{x + d(x,y) - d(x,y) \bmod 2^i\} \bmod 2^b$$

とも定義できる．

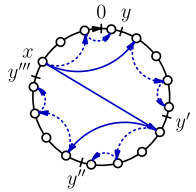


図 12　ピア x から 2 のべき乗間隔のアドレス y，y'，y''，y''' へのマルチキャスト．点線で示した共通する形の部分ではマルチキャストによりまとめられる引き継ぎがない
Fig. 12　Multicast for power of 2 interval address set $\{y, y', y'', y'''\}$ from peer x.

補題 3. b 以下の任意の非負整数 s，2^{b-s} 未満の任意の非負整数 t，2^b 未満の任意の非負整数 x について，

$$c_{x,g(A_{s,t})} = c_{x,\{q_{x,y,b-s}|y\in g(A_{s,t})\}} + \sum_{y\in g(A_{s,t})} c_{q_{\{x,y,b-s\}},y} \tag{18}$$

である．

証明．$g(A_{s,t})$ に属する任意のアドレス y について考える．$\mathfrak{y}_0 := x$，$\mathfrak{y}_{c_{x,y}} := y$ とし，x から y へのアクセスの引き継ぎ経路に現れるピアを，順に $\mathfrak{y}_0, \mathfrak{y}_1, \ldots, \mathfrak{y}_{c_{x,y}}$ とする．また，$g(A_{s,t})$ に属する y とは異なる任意のアドレス z についても，$\mathfrak{z}_0 := x$，$\mathfrak{z}_{c_{x,z}} := z$ とし，x から z へのアクセスの引き継ぎ経路に現れるピアを，順に $\mathfrak{z}_0, \mathfrak{z}_1, \ldots, \mathfrak{z}_{c_{x,z}}$ とする．$\mathfrak{y}_k = q_{x,y,b-s}$ なる $c_{x,y}$ 以下の非負整数 k について，

$$\{\mathfrak{y}_{k+1}, \mathfrak{y}_{k+2}, \ldots, \mathfrak{y}_{c_{x,y}}\} \cap \{\mathfrak{z}_0, \mathfrak{z}_1, \ldots, \mathfrak{z}_{c_{x,z}}\} = \emptyset \tag{19}$$

ならば，x から y へのアクセスの引き継ぎ経路において，$q_{x,y,b-s}$ より先は，x から $g(A_{s,t})$ に属する他のアドレスへの引き継ぎ経路と一致する部分がない．つまり，$q_{x,y,b-s}$ より先では，マルチキャストによりまとめられる引き継ぎはない．よって，式 (19) が成り立てば，x から $g(A_{s,t})$ へのマルチキャストの引き継ぎ経路は，x から $\{q_{x,y,b-s} \mid y \in g(A_{s,t})\}$ へマルチキャストした後に各 $q_{x,y,b-s}$ から y へアクセスした場合の引き継ぎ経路と同じになる．つまり，補題 3 が成り立つ．

以下，背理法により式 (19) が成り立つことを示す．

$\mathfrak{y}_{k+i} = \mathfrak{z}_j$ なる $c_{x,y} - k$ 以下の正整数 i，$c_{x,z}$ 以下の非負整数 j が存在すると仮定する．補題 2 より，$\mathfrak{y}_{k+i}$ と $\mathfrak{z}_j$ 以前の引き継ぎ経路は一致する．よって，$\mathfrak{y}_k = \mathfrak{z}_{j-i}$ である．

(i) $d(x,y) < d(x,z)$ のとき，$d(\mathfrak{z}_{j-i}, z) = d(\mathfrak{y}_k, y) + d(y,z)$ である．$y \neq z$ と定理 2 より，$d(y,z)$ は 2^{b-s} 以上であるため，$d(\mathfrak{z}_{j-i}, z)$ も 2^{b-s} 以上である．また，$\mathfrak{z}_{j-i+1}$ は $\mathfrak{z}_{j-i}$ の 2 のべき乗先のピアの中で最も z に近いピアであるため，$d(\mathfrak{z}_{j-i}, \mathfrak{z}_{j-i+1})$ は 2^{b-s} 以上となる．しかし，$d(\mathfrak{z}_{j-i}, \mathfrak{z}_{j-i+1}) \leq d(\mathfrak{z}_{j-i}, \mathfrak{z}_j) = d(\mathfrak{y}_k, \mathfrak{y}_{k+i}) \leq d(\mathfrak{y}_k, y) = d(q_{\{x,y,b-s\}}, y) < 2^{b-s}$ より，矛盾する．

(ii) $d(x,y) > d(x,z)$ のとき，(i) の場合と同様にして，$d(\mathfrak{y}_k, y)$ は 2^{b-s} 以上である．しかし，$d(\mathfrak{y}_k, y) = d(q_{\{x,y,b-s\}}, y) < 2^{b-s}$ より，矛盾する．

以上より，$\mathfrak{y}_{k+i} = \mathfrak{z}_j$ なる $c_{x,y} - k$ 以下の正整数 i，$c_{x,z}$ 以下の非負整数 j は存在しない．よって，式 (19) が成り立つ．　□

次に，補題 3 の式 (18) を評価する．

まず，式 (18) の右辺第 1 項 $c_{x,\{q_{x,y,b-s}|y\in g(A_{s,t})\}}$ が $|A_{s,t}| - 1$ であることを示す．任意のアドレス y，非負整数 i について，$q_{x,y,i}$ は y まで残り $d(x,y) \bmod 2^i$ のアドレスであるため，x と $q_{x,y,i}$ との相対的位置関係は，$(x + d(x,y) \bmod 2^i) \bmod 2^b$ と y との相対的位置関係と等しい．また，定理 2 より，$g(A_{s,t}) = \{z2^{b-s} + y' \mid z \in A_{s,0}\}$ なる 2^{b-s} 未満の非負整数 y' が存在する．よって，x と $\{q_{x,y,b-s} \mid y \in g(A_{s,t})\}$ との相対的位置関係は，$(x + d(x,y') \bmod 2^{b-s}) \bmod 2^b$ と $g(A_{s,t})$ との相対的位置関係と等しい．つまり，

$$c_{x,\{q_{x,y,b-s}|y\in g(A_{s,t})\}} = c_{(x+d(x,y') \bmod 2^{b-s}) \bmod 2^b, g(A_{s,t})}$$

となる．y' の定義より，$g(A_{s,t})$ は $(x + d(x,y') \bmod 2^{b-s}) \bmod 2^b$ を含むため，補題 1 より，これは最大で $|A_{s,t}| - 1$ である．

次に，式 (18) の右辺第 2 項は，定理 1 と定理 2 より，前段の y' を用いて，

$$\begin{aligned}\sum_{y\in g(A_{s,t})} c_{q_{\{x,y,b-s\}},y} &= \sum_{y\in g(A_{s,t})} u_{b,\log n}(d(x,y) \bmod 2^{b-s}) \\ &= |A_{s,t}| u_{b,\log n}(d(x,y') \bmod 2^{b-s})\end{aligned}$$

となる．これは，$\log n \geq s$ のとき，最大で $|A_{s,t}|(\log n - s)$ である．

以上より，補題 3 の $c_{x,g(A_{s,t})}$ は最大で

$$|A_{s,t}| - 1 + |A_{s,t}|(\log n - s) \tag{20}$$

となる．

定理 5. ピア数を n，アクセス範囲の幅を w として，2.3 節の単純な並列範囲アクセスの引き継ぎの回数は $\Omega(w \log n)$，深さは $\Theta(\log n)$ である．

証明．引き継ぎの深さは，式 (17) より，$\Theta(\log n)$ である．

引き継ぎの回数は，アクセスする添字の範囲を R として，$R = A_{s,t}$ なる非負整数 s，t が存在する場合，補題 3 と式 (20) より，最大で $|R| - 1 + |R|(\log n - \log |R|)$ である．よって，$|R| = w$ より，$\Omega(w \log n)$ である．　□

5. 実験評価

定義 3 の並列範囲アクセスにより，引き継ぎの回数は分散配列の非並列な範囲アクセス[10]なみになり，引き継ぎの深さは 2.3 節の単純な並列範囲アクセスなみになることを，シミュレーション実験により示す．また，実験結果が仮定 1 と近似 1 のもとで求めた 4 章の理論的傾向に従うことを確認する．さらに，ピアの参加・離脱がある環境でも，提案手法の優位性が失われないことを示す．

5.1 設　　定

実験には自作のシミュレータを用いた．Chord ネットワークの構成は定義 1 に従った．b は 64 とした．ピアのアドレスへの配置には，プリフィクスとなる文字列に 10 進数表記のピア番号を続けた文字列から SHA1 のハッシュ値を計算し，その先頭 64 ビットを使用した．プリフィクスを変え，10 通りのピア配置を試した．定義 2 を用いるにあたり，h は，SHA1 のハッシュ値を計算し，その先頭 64 ビットを求める関数とした．配列名は 1 つのみを用いた．範囲アクセスの開始ピアはランダムに選び，アクセス範囲は $[0, 1000)$ に含まれるランダムな整数範囲とした．1 つの配列名，10 通りのピア配置の 1 つ 1 つに対し，最大 10,000 回の範囲アクセスを行い，統計を出した．

ピアの参加・離脱がある環境での性能を評価するため，β を 0 以上 1 未満の実数として，以下のように βn ピアが入れ替わる状況をシミュレートした．まず，$(1+\beta)n$ のピアをランダムに次の 3 つのグループに分ける．

JoinGroup　新たに参加する βn ピア．

ExitGroup　離脱する βn ピア．

StableGroup　それ以外の $(1-\beta)n$ ピア．

定義 1 のピア間接続のルールを適用する際，$\mathrm{suc}((x+1) \bmod 2^b)$ は JoinGroup と StableGroup から選ぶ．$\mathrm{pre}((x+2^i) \bmod 2^b)$（ただし，$i = 1, 2, \ldots, b-1$）は ExitGroup と StableGroup から選ぶ．操作の開始ピアは JoinGroup と StableGroup から選ぶ．アクセス手続きにおいて，引き継ぎ先のピアが ExitGroup の場合，その引き継ぎは失敗する．引き継ぎが失敗したら，次善の接続先への引き継ぎを試みる．これを引き継ぎが成功するまで繰り返す．定義 1 のピア間接続のルールを適用する際，$\mathrm{suc}((x+1) \bmod 2^b)$ は ExitGroup 以外から選ぶため，最悪でも $\mathrm{suc}((x+1) \bmod 2^b)$ への引き継ぎは成功する．

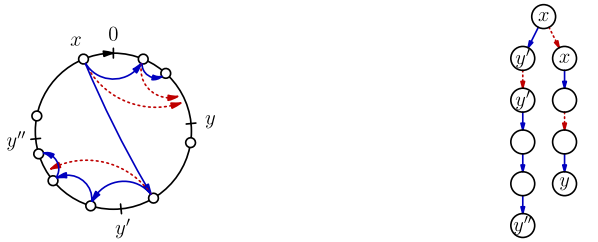
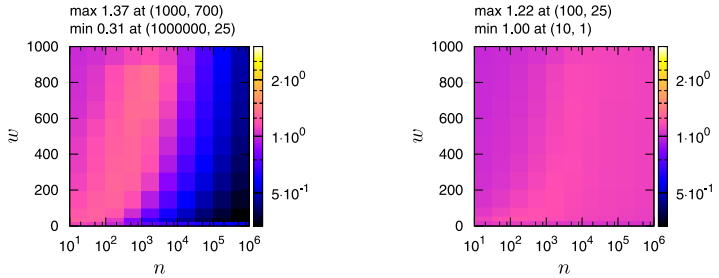


図 13　失敗を含む引き継ぎ．開始ピア x からアドレス y，y'，y'' へのマルチキャストを示す．左図はアドレス環での引き継ぎを示す．点線が失敗した引き継ぎを表す．右図は関係するピアと引き継ぎの関係だけを抜き出したものである．y，y'，y'' を担当のラベルとして用いている．引き継ぎの失敗は，元のピアへ点線でつなぐことにより示している

Fig. 13　Relays including failures.



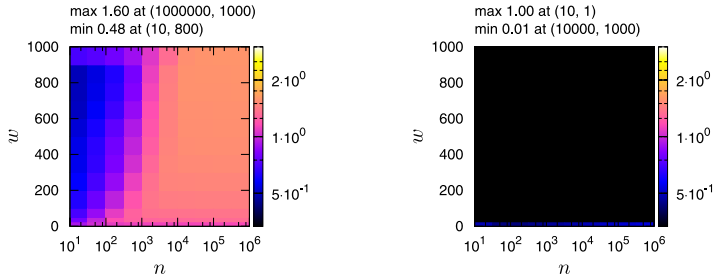
(a) 引き継ぎの回数　　(b) 引き継ぎの深さ

図 14　単純手法の平均値を 1 とした場合の提案手法の平均値．β は 0

Fig. 14　Ratio of average values for proposed method to that for naive method when $\beta = 0$.

引き継ぎ失敗の回数および深さは，成功した引き継ぎの回数および深さとは別に評価する．引き継ぎ失敗の深さは，引き継ぎ経路の中で，最も引き継ぎ失敗が起こった開始ピアから終端までの経路における引き継ぎ失敗の回数とする．たとえば，**図 13** では，引き継ぎ失敗の回数は 3，深さは 2 となる（引き継ぎの回数は 6，深さは 4）．

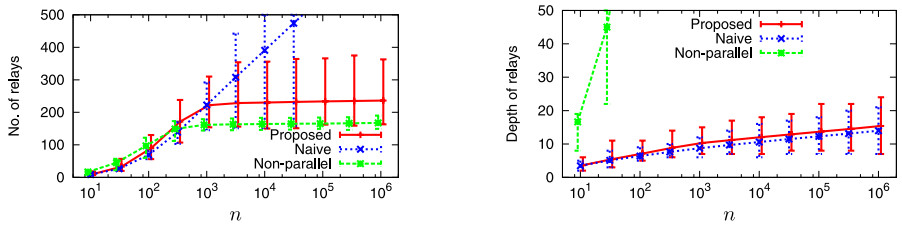
実験変数は，ピア数 n，アクセス範囲の幅 w，前々段の β のいずれかである．n を変化させる場合は，10 から 1,000,000 まで $\sqrt{10}$ 倍ずつ変化させた．w を変化させる場合は，最初



(a) 引き継ぎの回数　(b) 引き継ぎの深さ

図 15　非並列手法の平均値を 1 とした場合の提案手法の平均値．β は 0

Fig. 15　Ratio of average values for proposed method to that for non-parallel method when $\beta = 0$.



(a) 引き継ぎの回数　(b) 引き継ぎの深さ

図 16　$w = 100$，$\beta = 0$ のときの平均，最大，最小

Fig. 16　Average, maximum, and minimum values when $w = 100$ and $\beta = 0$.

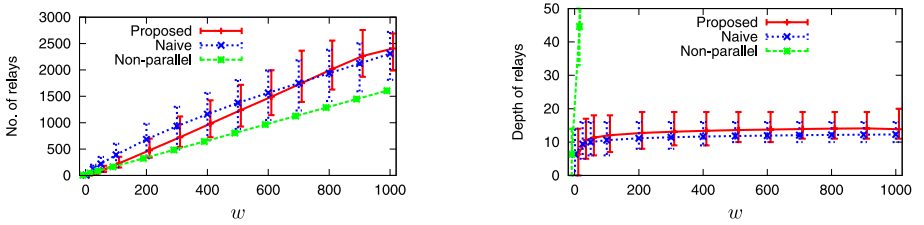
は 1，25，50，100 と変化させ，100 からは 1,000 まで 100 ずつ変化させた．β を変化させる場合は，0 から 0.5 まで 0.05 ずつ変化させた．

5.2 結　　果

本節では，定義 3 の並列範囲アクセスを提案手法，2.3 節の単純な並列範囲アクセスを単純手法，分散配列における非並列な範囲アクセス[10] を非並列手法と呼ぶ．

結果は**図 14**，**図 15**，**図 16**，**図 17**，**図 18** である．

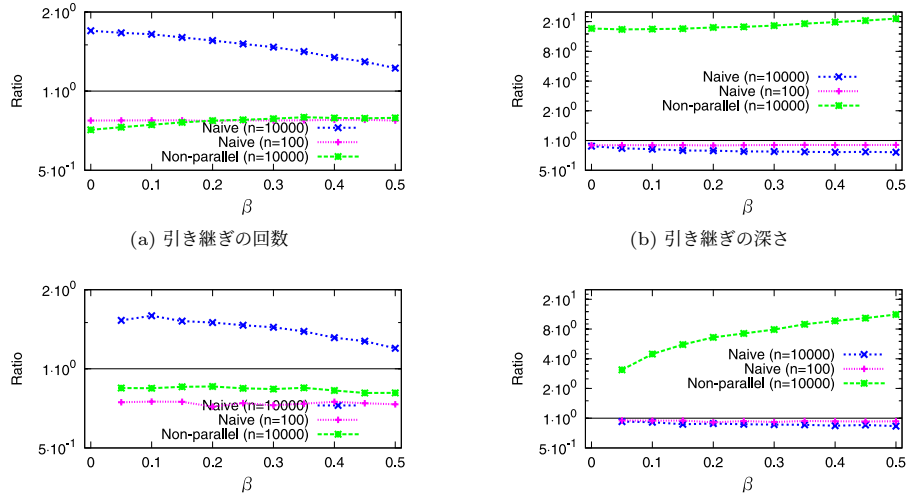
図 14 と図 15 において，提案手法は，つねに，単純手法の 1.2 倍以下の引き継ぎの深さを，非並列手法の 1.6 倍以下の引き継ぎの回数で実現している．また，提案手法の引き継ぎの深さは非並列手法よりずっと小さく，引き継ぎの回数は単純手法の 1/3 以下に抑えられる場合もある．ただし，ピア数がアクセス範囲に比べおよそ 10 倍以下の場合，単純手法の



(a) 引き継ぎの回数　(b) 引き継ぎの深さ

図 17　$n = 10000$，$\beta = 0$ のときの平均，最大，最小

Fig. 17　Average, maximum, and minimum values when $n = 10000$ and $\beta = 0$.



(a) 引き継ぎの回数　(b) 引き継ぎの深さ

(c) 引き継ぎ失敗の回数　(d) 引き継ぎ失敗の深さ

図 18　提案手法の平均値を 1 とした場合の単純手法，非並列手法の平均値．w は 100

Fig. 18　Ratio of average values for naive method and non-parallel method to that for proposed method when $w = 100$.

方が提案手法より引き継ぎの回数の方が少ない．つまり，そのような環境では単純手法の方が優れる．

図 16 (a) において，提案手法の引き継ぎの回数は，n が十分に大きいとき，n に対する依存は小さい．単純手法の引き継ぎの回数は $\log n$ に比例し，その傾きは大きい．図 16 (b) に

おいて，提案手法と単純手法の引き継ぎの深さは $\log n$ におおよそ比例している．図 17 (a) において，提案手法と単純手法の引き継ぎの回数は w におおよそ比例している．図 17 (b) において，提案手法と単純手法の引き継ぎの深さは，w が十分に大きいとき，w に依存していない．

図 18 (a) と図 18 (c) において，非並列手法と提案手法の引き継ぎの回数の比は，ほとんど β に依存していない．単純手法と提案手法の引き継ぎの回数の相対的な差は，n が 10,000 のとき，β が大きくなるにつれ，小さくなっている．ただし，β が 0.5 になっても，順序が入れ替わってはいない．n が 100 のときは，β に依存していない．図 18 (b) と図 18 (d) において，単純手法と提案手法の引き継ぎの深さの比は，ほとんど β に依存していない．非並列手法と提案手法の相対的な差は，β が大きくなるにつれ，大きくなっている．これは，並列な手法では，最深の経路での引き継ぎ失敗だけが深さを増加させるのに対し，非並列手法では，すべての引き継ぎ失敗が深さを増加させるためである．

6. 今後の課題

前章までに，定義 3 の並列範囲アクセスが理論的（表 1），実験的（図 14，図 15，図 16，図 17，図 18）に優れた手法であることを示した．

今後の課題として，手法をさらに改良することがあげられる．たとえば，図 14 (b)，図 16 (b)，図 17 (b) に見られるように，提案手法は引き継ぎの深さに関して単純手法より若干劣り，改良の余地がある．

図 14 (a)，図 16 (a)，図 17 (a) から分かるように，提案手法より単純手法の方が優れる場合もあるため，手法を切り替えることも考えられる．そのためには，切替えに必要な情報を選別し，その情報の取得方法，具体的な切替え手順などを与え，切替えの影響を検証する必要がある．

また，実際の運用のためには，以下に記す，定義 1 の Chord ネットワークとそれよりオリジナル[4] に近い Chord ネットワークとの違いを検証する必要がある．

定義 1 の Chord ネットワークとオリジナルの Chord ネットワークとの最大の違いは，アドレスの担当を決定するルールにある．定義 1 の Chord ネットワークでは，任意のアドレス x について，x の担当は $\mathrm{pre}(x)$ になる．それに対し，オリジナルの Chord ネットワークでは，x の担当は $\mathrm{suc}(x)$ になる．以下では x の担当を $\mathrm{pre}(x)$ にした場合の Chord ネットワークを **PreNet**，担当を $\mathrm{suc}(x)$ にした場合の Chord ネットワークを **SucNet** と呼ぶことにする．

PreNet はアクセスコストに優れ，SucNet はピアの参加・離脱に対応するためのコストに優れる．

まず，SucNet では，アドレスにアクセスするために必要な引き継ぎの回数が，PreNet と比べて 1 だけ増える．それは，Chord ネットワークでのアクセス手続きが，目標のアドレスを x として，基本的に $\mathrm{pre}(x)$ を探す手続きだからである．$\mathrm{pre}(x) \neq x$ の場合，SucNet で x の担当に到達するためには，$\mathrm{pre}(x)$ に到達した後，さらに，$\mathrm{suc}(x)$ へ処理を引き継がなければならない．$\mathrm{pre}(x) = x$ となる確率は $n/2^b$ であるが，$2^b \gg n$ より，0 としてよい．そのため，引き継ぎは 1 回増える．よって，SucNet では，配列の操作に必要な引き継ぎの回数は，PreNet に比べ，アクセスする要素の数 w に比例する分だけが増えることになる．

次に，PreNet では，各ピアが，アドレス環で正方向に自身より手前のいくつかのピアを，SucNet と比べて余分に把握しなければならない．この自身より手前のピアの集合を predecessor リストと呼ぶことにする．PreNet において，任意のピア x がネットワークを離脱した場合，x に割り当てられていたアドレスの新しい担当は $\mathrm{pre}((x-1) \bmod 2^b)$ になる．$p_1 := \mathrm{pre}((x-1) \bmod 2^b)$ とし，この p_1 も x と同時に離脱した場合，x が担当していたアドレスの新しい担当は $\mathrm{pre}((p_1-1) \bmod 2^b)$ になる．以降同様に，

$$p_i := \begin{cases} \mathrm{pre}((x-1) \bmod 2^b) & (i=1) \\ \mathrm{pre}((p_{i-1}-1) \bmod 2^b) & (i>1) \end{cases}$$

とすると，$x, p_1, p_2, \ldots, p_i$ が同時に離脱した場合，x が担当していたアドレスの新しい担当は p_{i+1} になる．そのため，管理するデータの可用性を保つために，x は，i が 1 から適当な値までの p_i を predecessor リストとして把握し，それらに x 自身が管理するデータをバックアップさせなければならない．同様に，SucNet では，successor リストと呼ばれるアドレス環で自身より先のいくつかのピアを把握し，それらに自身が管理するデータをバックアップさせなければならない．ただし，元々，PreNet でも SucNet でも，ピアの参加・離脱がある場合においてアクセス手続きを正しく動かすために，各ピアは successor リストを把握する必要がある．つまり，データのバックアップに際し，SucNet はアクセス手続きの正常動作のために用いる successor リストを流用できるのに対し，PreNet は別途 predecessor リストを必要とする．

参 考 文 献

1) Maymounkov, P. and Mazières, D.: Kademlia: A Peer-to-Peer Information System Based on the XOR Metric, *IPTPS*, Druschel, P., Kaashoek, M.F. and Rowstron, A.I.T. (Eds.), Lecture Notes in Computer Science, Vol.2429, pp.53–65, Springer (2002).
2) Ratnasamy, S., Francis, P., Handley, M., Karp, R.M. and Shenker, S.: A scalable content-addressable network, *SIGCOMM*, pp.161–172 (2001).
3) Rowstron, A.I.T. and Druschel, P.: Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems, *Middleware*, Guerraoui, R. (Ed.), Lecture Notes in Computer Science, Vol.2218, pp.329–350, Springer (2001).
4) Stoica, I., Morris, R., Karger, D.R., Kaashoek, M.F. and Balakrishnan, H.: Chord: A scalable peer-to-peer lookup service for internet applications, *SIGCOMM*, pp.149–160 (2001).
5) Aberer, K., Cudré-Mauroux, P., Datta, A., Despotovic, Z., Hauswirth, M., Punceva, M. and Schmidt, R.: P-Grid: A self-organizing structured P2P system, *SIGMOD Record*, Vol.32, No.3, pp.29–33 (2003).
6) Andrzejak, A. and Xu, Z.: Scalable, Efficient Range Queries for Grid Information Services, *Peer-to-Peer Computing*, Graham, R.L. and Shahmehri, N. (Eds.), pp.33–40, IEEE Computer Society (2002).
7) Aspnes, J. and Shah, G.: Skip graphs, *ACM Trans. Algorithms*, Vol.3, No.4 (2007).
8) Bharambe, A.R., Agrawal, M. and Seshan, S.: Mercury: Supporting scalable multi-attribute range queries, *SIGCOMM*, Yavatkar, R., Zegura, E.W. and Rexford, J. (Eds.), pp.353–366, ACM (2004).
9) Li, D., Lu, X., Wang, B., Su, J., Cao, J., Chan, K.C.C. and Leong, H.V.: Delay-Bounded Range Queries in DHT-based Peer-to-Peer Systems, *ICDCS*, p.64, IEEE Computer Society (2006).
10) Fukuchi, D.: Distributed Arrays: A P2P Data Structure for Efficient Logical Arrays, Master's thesis, The University of Tokyo (2009).
11) 福地大輔，クリスチャン・ソッメル，清　雄一，本位田真一：分散配列：効率的な論理配列を実現する P2P データ構造，情報処理学会論文誌，Vol.50, No.2, pp.721–736 (2009).
12) Fukuchi, D., Sommer, C., Sei, Y. and Honiden, S.: Distributed Arrays: A P2P Data Structure for Efficient Logical Arrays, *INFOCOM*, pp.1458–1466, IEEE (2009).
13) van der Corput, J.G.: Verteilungsfunktionen I, *Akademie van Wetenschappen*, Vol.38, pp.813–821 (1935).
14) Kuipers, L. and Niederreiter, H.: *Uniform Distribution of Sequences*, Wiley (1974).

(平成 22 年 11 月 10 日受付)
(平成 23 年 5 月 14 日採録)

福地　大輔

2009 年東京大学大学院情報理工学系研究科コンピュータ科学専攻修士課程修了．同年同博士課程進学，現在に至る．日本学術振興会特別研究員 DC．国立情報学研究所アーキテクチャ科学研究系リサーチアシスタント．P2P 関連の研究に従事．

本位田真一（フェロー）

1978 年早稲田大学大学院理工学研究科修士課程修了．（株）東芝を経て 2000 年より国立情報学研究所教授，2004 年より同研究所アーキテクチャ科学研究系研究主幹を併任，現在に至る．2008 年より同研究所先端ソフトウェア工学・国際研究センター長を併任，現在に至る．2001 年より東京大学大学院情報理工学系研究科教授を兼任，現在に至る．現在，早稲田大学客員教授，英国 UCL 客員教授を兼任．2005 年度パリ第 6 大学招聘教授．工学博士（早稲田大学）．1986 年度情報処理学会論文賞受賞．日本ソフトウェア科学会理事，情報処理学会理事を歴任．ACM 日本支部会計幹事，情報処理学会フェロー，日本ソフトウェア科学会編集委員長，日本学術会議連携会員．