

Refereed Conference Paper

## P2P オーバレイネットワークにおけるプローブを用いた 共有情報の範囲検索法

中尾 誠治 水戸 将弥 田頭 茂明 藤田 聡

P2P 環境では、様々な属性をもつ情報が多くの計算機（ピア）により共有されている。これらの共有情報の中から、特定の属性に対して、ある範囲内に含まれる情報を発見する範囲検索技術が注目されている。このような検索を P2P 環境で実現するためには、頻繁に発生するピアの参加/離脱を想定した効率的な手法が必要である。本研究では、リング型のオーバレイネットワークをピア間で構築し、そのネットワーク上にショートカット（プローブ）を付加して P2P 環境に適した範囲検索手法を提案する。提案手法では、プローブを用いて効率的な検索を実現し、さらに、楽観的にプローブの付加と管理をおこなうことで、参加/離脱に伴うオーバレイネットワークの変化に柔軟に対応する。我々は提案手法を実験により評価し、その有効性を確認した。その結果から、提案手法は様々な属性値の分布に対して、平均  $O(\sqrt{\frac{n}{q}})$  ステップで目的とする情報を発見できることがわかった。

## A Range Search Method with Randomized Probes in a P2P Overlay Network

Seiji NAKAO Masaya MITO Shigeaki TAGASHIRA Satoshi FUJITA

In many P2P systems, objects with many kinds of attributes are shared with participating hosts. A range search method realizes a mechanism that could return all objects whose attribute values are between two specified values. In this paper, we propose a range search method with randomized probes in a ring-based P2P overlay network. The proposed method can achieve high efficiency for finding a target object by using probes as shortcuts on the overlay network, and enhance the flexibility for a dynamical change of the network topology by employing an optimistic policy for construction and maintenance of these probes. We conducted several experiments to evaluate the performance of the proposed method. The results of experiments implies that the method can find a target object with  $O(\sqrt{\frac{n}{q}})$  steps.

### 1 はじめに

近年のインターネットの普及とブロードバンド化により、ネットワーク上でのコミュニケーションやデータ通信が一般的になってきている。それにとともに、ネットワーク上の情報を効率的に収集することが重要となり、現在では、Google[1]などの検索エンジンが、インターネットを利用するうえで必要不可欠である。現在の検索エンジンは、キーワードによる情報検索が一般的であるが、より高度なクエリによる検索が効率

の良い情報収集を助けると考えられる。そのようなクエリの1つとして、数値範囲による検索（範囲検索）が挙げられる。ネットワーク上の情報には、日付や価格など数値を属性として持つものも多い。そのため、範囲検索をキーワード検索と併せて用いることで、アプリケーションの利便性の向上が期待できる。実際に、サーバ上での範囲検索を提供しているサイトはよく見られ、2004年3月にはGoogleも“Numrange Searches”として範囲検索機能をリリースしている。

その一方で、近年、負荷分散やスケーラビリティに優れる P2P モデルが注目を集めており、P2P モデルに基づいたアプリケーションが普及してきている。このようなアプリケーションにおいても範囲検索は有用であると考えられ、例えば、個人間取引を支援するアプ

広島大学大学院工学研究科情報工学専攻  
〒739-8527 広島市鏡山1-4-1  
Graduate School of Engineering, Hiroshima University  
Kagamiyama 1-4-1, Higashi-Hiroshima, 739-8527 Japan  
{nakao, mito, shigeaki, fujita}@sc.hiroshima-u.ac.jp

リケーション PinPost[2] や人と人とのマッチングを支援するアプリケーションにおいて、取引額や年齢がある範囲に含まれるオブジェクトを検索することは、システムの利便性を向上させるだろう。しかしながら一般に、P2P 環境ではピアは完全分散で流動性が高く、ユーザの参加と離脱が頻繁に起こるため、各ピアが他のピアの状態を正確に保持し続けるコストは高く、そのコストと検索時間がトレードオフの関係にあるといえる。

P2P 環境で情報検索を実現するための手法には、フラッディングを基にした手法と分散ハッシュテーブルを用いた手法がある。フラッディングを基にした手法は、Gnutella[3] や Freenet[4] などで行われているが、ネットワーク上の通信量が大きくなってしまふ。また分散ハッシュテーブルは、検索時間を短縮するために、CAN[5], Chord[6], P-Grid[7], Tapestry[8] などで行われているが、これらの手法はキーワードによる検索を想定して論理ネットワークを構築しており、効率の良い範囲検索を実現することは難しい。

そこで本研究では、範囲検索を低コストでおこなうために、属性値によってソートされたリング型ネットワークを構築し、それを利用した範囲検索法を提案し評価する。提案する検索法では、ピアはメンテナンスのコストが非常に低いランダムなリンク（プローブ）を保持し、プローブを利用して検索すべきピアの絞り込みをおこなうことで短時間での検索を実現する。

提案手法は、キーワード検索等を実現するような他の検索手法との併用を想定しており、ピアをキーワード等により分類し、カテゴリごとにリング型ネットワークを構築するため、1つのリング型ネットワークのサイズは数万程度のもをを対象としている。ピアの分類法としては、例えば、フリーマーケットでの商品の種類や就職支援での勤務地や職種などが考えられる。

本稿の構成は以下のとおりである。まず最初に2節では分散データ構造の提案をおこない、3節ではその構造を利用した分散探針検索法 (DPS: Distributed Probes based Search method) を提案する。4節ではピアの参加と離脱について述べる。5節では提案手法の実現に必要なデータ構造のコストを解析し、性能を評価するために実験をおこなう。最後に6節でまとめる。

## 2 分散データ構造

### 2.1 システムモデル

本研究では以下のようなモデルを考える。P2P ネットワークに参加しているピアの集合を  $P = \{p_1, p_2, \dots, p_n\}$  とする。各ピア  $p_i$  は、1つの情報 (オブジェクト) に対応するレコード  $R_i = (r_i^1, r_i^2, \dots, r_i^k)$  と、他のピアを指す  $d'$  本のリンクの組  $L_i = (l_i^1, l_i^2, \dots, l_i^{d'})$

を保有している。なお、レコード  $R_i$  の  $k$  個の属性値は実数であるとする。ここで、属性値が等しい場合にはピアの ID などにより順序をつけることができるため、属性値は互いに異なると一般性を失うことなく仮定できる。また、ここで対象とする P2P ネットワーク上には、ブートストラップの役割をはたすホストが存在するものとする。

### 2.2 リング型ネットワーク

本研究では範囲検索を実現するために、各属性ごとに値の昇順にソートされた双方向のリング型ネットワークを構築する。このようなリング型ネットワーク上では、ある属性  $j$  に対して、 $\alpha$  以上の最小の属性値を持つピア (基点ピア (BBP: Base Point Peer)) を発見する手続き Find( $\alpha, j$ ) が重要となる。具体的には、基点ピアから大きい側の隣接ピアへのリンクをたどることを繰り返して、属性値がある範囲に含まれる全てのピアを発見することができる。また、基点ピアの発見は、ピアの参加などリング型ネットワークのメンテナンスにも利用することができる。

以降で用いる記法を述べる。ピアは属性ごとに、その属性値に関して小さい側の隣接ピアへのリンク (descending link) と大きい側の隣接ピアへのリンク (ascending link) を保持している ( $d' \geq 2k$ )。これらのリンクを属性  $j$  の隣接リンク (adjacent links) と呼び、“ピア  $p_i$  はそれらが指すピアと属性  $j$  で隣接している”と呼ぶことにする。また、各属性で最大の属性値をもつピアと、最小の属性値をもつピアは隣接している。

### 2.3 プローブ

基点ピアは任意のピアから隣接リンクをたどって走査することで発見できる。しかし隣接リンクを利用するだけの線形探索では、その発見に最悪で線形ステップかかってしまう。そこで本研究では、他の属性での隣接リンクと、ランダムに選択されたリンクをショートカット (プローブ) として用いることにする。1つのピアを持つランダムに選択されたリンクの数は属性の数とピアの数に依存しないため、プローブは低コストでの付加/メンテナンスが可能である。また、他の属性での隣接リンクもプローブとして用いるため、属性が2つ以上あればランダムに選択されたリンクは必ずしも必要でなく、その場合、プローブのための付加/メンテナンスのコストは全くかからない。

プローブとして他の属性での隣接リンクも利用するため、ピア  $p_i$  の持つプローブの集合は属性ごとに異なる。図1を例に説明すると、ピア  $p_1$  は、属性1で  $p_2, p_6$  と隣接しており、属性2で隣接している  $p_5$  へのリンクとランダムに選択された  $p_3$  へのリンクをプローブとし

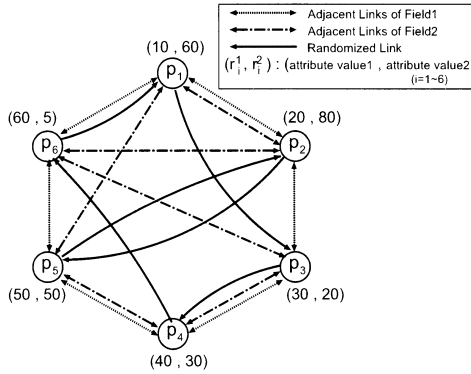


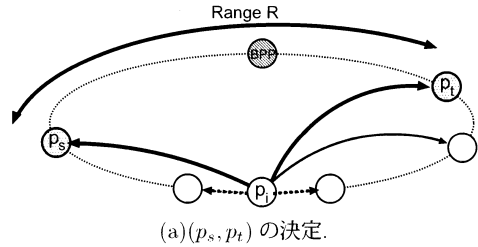
図 1: リング型ネットワークとプローブ.

て用い、属性 2 では  $p_2, p_5$  と隣接しており、属性 1 で隣接している  $p_6$  へのリンクと  $p_3$  へのリンクをプローブとして用いる。

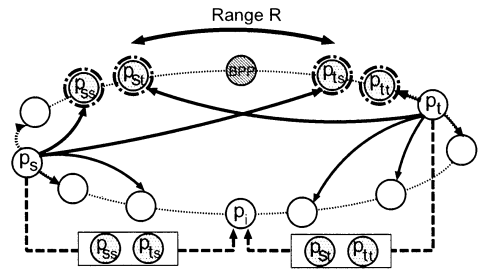
### 3 DPS

本節では、リング型ネットワーク上でプローブを利用して基点ピアを発見する分散探針検索法 (DPS: Distributed Probes based Search method) の概要を述べる。DPS は、プローブと隣接リンクを利用して、基点ピアが存在する範囲  $R$  を徐々に絞り込むことを繰り返し、基点ピアを発見する。DPS の概要を図 2 に示す。いま、 $R$  の両端のピアを  $(p_s, p_t)$  とする。DPS ではまず、図 2(a) のように、検索をおこなうピア  $p_i$  が自身のプローブと隣接リンクから  $(p_s, p_t)$  を特定する。続いて  $p_i$  は、図 2(b) のように、ピア  $p_s, p_t$  のプローブと隣接リンクを利用して、 $R$  の範囲をできる限り狭めるような新たな  $(p_s, p_t)$  を特定する。以降は同様に、図 2(c) のように、新たな  $(p_s, p_t)$  を特定することを繰り返して基点ピアを発見する。

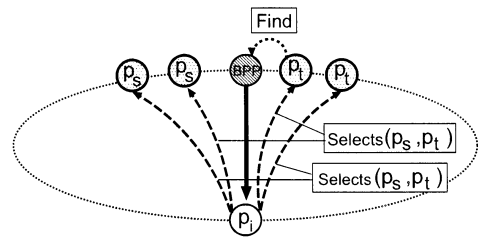
属性  $j$  の基点ピアを求める手続き Find および Refine を図 3 に記す。手続き Find は手続き Refine をさまざまなピアが実行することによって実現される。手続き Refine では、ピアはそのプローブと隣接リンクを利用して基点ピアを発見するか (1.3 in Refine)、 $R$  の範囲を最小とするピア  $(p_s, p_t)$  を特定する (1.4 in Refine)。手続き Find では、まず検索をおこなうピア  $p_i$  が手続き Refine を呼び出し、 $R$  の両端のピア  $(p_s, p_t)$  を特定する (1.2 in Find)。その後  $p_i$  は基点ピアを発見するまで以下の操作を繰り返す (1.3 in Find): 1)  $(p_s, p_t)$  にそれぞれ Refine を実行するようにメッセージを送り、 $(p_{s_s}, p_{t_s}), (p_{s_t}, p_{t_t})$  を得る (1.4-1.5 in Find)。2)  $p_i$  がそれらの中から新たな  $(p_s, p_t)$  を特定する (1.7-1.8 in Find)。



(a)  $(p_s, p_t)$  の決定.



(b) 情報の集約.



(c) 範囲を徐々に狭め基点ピアを発見.

図 2: DPS の概要.

### 4 ピアの参加と離脱

ピア  $p_i$  が P2P ネットワークに参加するときの手続き Join を図 4 に記す。ピア  $p_i$  は最初に全ての属性のリング型ネットワークで自身を挿入し (1.3-1.7 in 図 4)、その後ランダムにリンクをはる (1.8 in 図 4) ことによりネットワークに参加する。各リング型ネットワークで自身を挿入するためにピア  $p_i$  は、まず基点ピア  $p'$  を発見し (1.4 in 図 4)、次にピア  $p'$  に対して、小さい側の隣接リンクを  $p_i$  にはるようにメッセージを送り (1.5 in 図 4)、ピア  $p'$  の小さい側の隣接ピア  $p''$  に対して、大きい側の隣接リンクを  $p_i$  にはるようにメッセージを送る (1.6 in 図 4)。最後に基点ピア  $p'$  とピア  $p''$  に、それぞれ大きい側と小さい側の隣接リンクをはる (1.7 in 図 4)。

つづいてピア  $p_i$  がネットワークから離脱するときの手続き Leave を図 5 に記す。ピア  $p_i$  は、全ての属性のリング型ネットワークで自身を削除する (1.3-1.5 in 図

```

1: procedure Find( $\alpha, j$ ) {For initiator  $p_i$ .}
2:   Calls Refine( $\alpha, j$ );{It returns a BPP or  $(p_s, p_t)$ .}
3:   until gets a BPP;{
4:     Sends  $p_s$  a message to call Refine, and gets a BPP or  $(p_{s_s}, p_{t_s})$ .
5:     Sends  $p_t$  a message to call Refine, and gets a BPP or  $(p_{s_t}, p_{t_t})$ .
6:     If a BPP was not found;{
7:       Chooses between  $p_{s_s}$  and  $p_{s_t}$  as a new  $p_s$ .
8:       Chooses between  $p_{t_s}$  and  $p_{t_t}$  as a new  $p_t$ .} }
9:   Returns the BPP
10: end

1: procedure Refine( $\alpha, j$ ) {For initiator  $p$ }
2: begin
3:   Decisions some of the adjacent links or probes linking to a BPP.
4:   Determines  $(p_s, p_t)$  with its adjacent links and probes.
4:   Returns the BPP or  $(p_s, p_t)$ .
6: end

```

図 3: 基点ピアの発見.

```

1: procedure Join {For initiator  $p_i$ .}
2: begin
3:   for each attribute  $j$ ;{
4:     Calls Find( $r_i^j, j$ ); (It returns a peer  $p'$ . Let  $p''$  be a peer adjacent to  $p'$  with the descending link.)
5:     Sends  $p'$  a message; link  $p_i$  with the descending link.
6:     Sends  $p''$  a message; link  $p_i$  with the ascending link.
7:     Links  $p'$  and  $p''$  with the ascending link and the descending link respectively.}
8:   if ( $d' > 2k$ ); Chooses  $(d' - 2k)$  peers randomly to link them as its probes.
9: end

```

図 4: ピアの参加.

5) ことによりネットワークから離脱する。各リング型ネットワークで自身を削除するために、ピア  $p_i$  は、隣接するピア  $p'$  に対して小さい側の隣接リンクをピア  $p''$  にはるようにメッセージを送り (1.4 in 図 5)、ピア  $p''$  に対して大きい側の隣接リンクをピア  $p'$  にはるようにメッセージを送る (1.5 in 図 5)。ピア  $p_i$  が P2P ネットワークから離脱するとき送らなければならないメッセージの総数は  $2k$  である。

## 5 評価

### 5.1 コスト比較

リング型ネットワークを用いることにより、DPS、線形探索 (LS: Linear Search) のほかに 2 分探索 (BS: Binary Search) が実現できる。具体的に BS は、1 つのリング型ネットワーク上で、各ピアがショートカットを  $\log n - 1$  本用意し、 $2^1, 2^2, \dots, 2^{n-1}$  個隣のピアにショートカットを張ることにより、 $O(\log n)$  ステップ

表 1: アルゴリズムのコスト.

	DPS	LS	BS
リンク数 $d'$	$O(k)$	$O(k)$	$O(k \log n)$
Join	$O(k)$	$O(k)$	$O(kn \log n)$
Leave	$O(k)$	$O(k)$	$O(kn \log n)$
Find	$O(\sqrt{\frac{n}{k}})$	$O(n)$	$O(\log n)$

で基点ピアを発見することができる。しかしながらこのような方法では、隣接リンクに加えて  $O(k \log n)$  のリンクを保持する必要があるだけでなく、リンク先が常に正しい必要があるため、メンテナンスに必要なコストが高くなってしまふ。

DPS、LS、BS を実現するために必要なコストを表 1 で比較する。ここではピアがネットワークに参加/離脱したとき、その分散データ構造を常に正しくメンテナンスするものとしてコストを見積もっている。DPS および LS は属性の数の定数倍という非常に少ないリンク数でアルゴリズムを実現できるのに対し、BS を実

```

1: procedure Leave {For initiator  $p_i$ .}
2: begin
3:   for each attribute  $j$  ;{
      (Let  $p'$  and  $p''$  be a peer adjacent to  $p_i$  with the ascending link and the descending link respectively.)
4:     Sends  $p'$  a message; link  $p''$  with the descending link.
5:     Sends  $p''$  a message; link  $p'$  with the ascending link.}
6: end

```

図 5: ピアの離脱.

現するためには属性ごとに  $\log n + 1$  本のリンクが必要である。また DPS、LS では、ピアが参加/離脱するとき（どこに参加すればよいか判明していれば）各属性の上で隣接したピアにメッセージを送れば十分である。その一方で、BS では参加/離脱の度に全ての属性の、全てのピアの、全てのリンクについて、BS を実現するような構造をメンテナンスする必要がある。

また、基点ピアの発見に必要なステップ数の平均は、LS が  $O(n)$  ステップであるのに対し、BS は  $O(\log n)$  ステップである。DPS はデータ構造の構築/メンテナンスのためのコストが LS とほとんど変わらないにもかかわらず、次節の実験の結果、平均  $O(\sqrt{n})$  ステップで基点ピアを発見できることがわかっている。これらのことから DPS はピアの参加/離脱が頻繁におこる P2P ネットワークでもメンテナンスに必要なコストが低く、ピアの数に対してスケラブルであるだけでなく、他の手法にとってはデメリットでしかない属性の数の増加もプローブの増加をもたらし、基点ピア発見の高速化を助けるという特長を持っているといえる。

## 5.2 実験的評価

### 5.2.1 実験環境

DPS の性能を評価するためにいくつかの実験をおこなう。実験では 1 つのリング型ネットワークを想定し ( $k = 1$ )、基点ピアもしくはある範囲に含まれるピアの発見に必要なステップ数を調べる。ここでは、( $p_s, p_t$ ) を更新する処理を 1 ステップとし (1.4-1.8 in Find, 図 3)、ピアの参加/離脱は起こらないものとする。1 回の実験ではランダムに選択されたプローブを持つリング型ネットワークを構築し、任意のピアから任意の基点ピア、もしくは属性値がある範囲に含まれるピアを発見する。各パラメータは以下のとおりである:

- ピアの数  $n$ : 64~65536
- プローブの数  $d$ : 1~16
- 属性値の範囲:  $0 \sim 2^{18}$
- 各実験の試行回数: 10,000 回

### 5.2.2 基点ピアの発見

本節では DPS の性能を明らかにするために、様々なパラメータを変化させて基点ピアを発見するために必要なステップ数を調べた。まず最初に、さまざまな属性値の分布に対する DPS の性能を調べる実験をおこなった。この時、属性値の分布として一様分布、正規分布、指数分布を用いた。実験はピアの数  $n$  を 65536 に固定し、プローブの数  $d$  を 1 から 16 まで変化させて 10,000 回ずつおこない、基点ピアを発見するまでの平均ステップ数を調べた。その結果を図 6 に示す。この図から DPS は、どの分布に対してもほぼ同様の性能を示していることがわかる。これはプローブをランダムに選択しているためと考えられ、どの分布に対しても性能が変わらないため、以降の実験では一様分布を用いることにする。

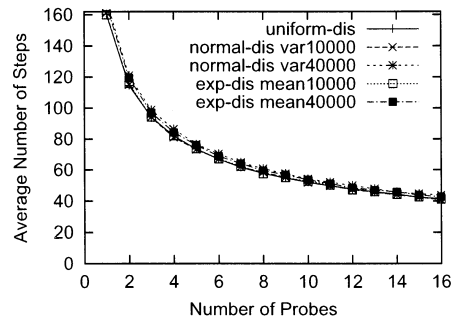


図 6: 様々な分布に対する DPS の性能.

次に、ピアの数とプローブの数が DPS の性能に与える影響を調べるために、ピアの数  $n$  を 64 から 65536 までプローブの数  $d$  を 1 から 16 まで変化させ、それぞれについて 10,000 回ずつ実験をおこなった。その結果を図 7 に示す。この図からプローブの数にかかわらず、ピアの数が 4 倍になると平均ステップ数は 2 倍となっていることがわかる。このことから平均ステップ数は  $\sqrt{n}$  に比例するといえる。また、この図からピアの数にかかわらず、プローブの数が 4 倍になると平均ホッ

ブ数は 1/2 倍になっていることがわかる。このことから平均ステップ数は  $1/\sqrt{d}$  に比例するといえる。以上

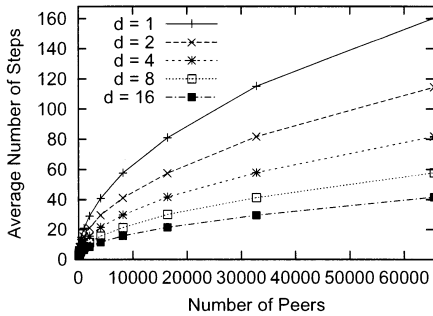


図 7: ピアの数・プローブの数と DPS の性能。

のことから DPS は平均  $O(\sqrt{n})$  ステップで基点ピアを発見できるという結果が得られた。

### 5.2.3 範囲検索

範囲検索は基点ピアの発見によっても実現できるが、範囲に含まれるピアを少なくとも 1 つ発見することでより効率よく実現できる。本節では範囲検索に対する DPS の性能を調べるために、検索範囲に含まれるピアの数、ピアの数、プローブの数を変化させて実験をおこなった。ここでは特に、ピアの数  $n$  が 4096 のときの結果を図 8 に示す。この図より、検索範囲に含まれるピアの数が増えるに従い平均ステップ数は少なくなり、プローブの数を増やしても性能が改善されにくくなる。このような傾向はピアの数  $n$  を変化させたときも現れており、リング型ネットワーク全体のピアの数  $n$  に対する検索範囲に含まれるピアの数の割合が等しければ、同じような性能を示す。このことから検索範囲に含まれるピアの割合が大きければ、属性の数がわずかでも付加的な領域を全く用いず DPS の性能を十分発揮できるといえる。

## 6 おわりに

本研究では、範囲検索を実現するために双方向リング型ネットワークを構築し、メンテナンスに必要なコストが低いプローブを利用した分散探針検索法 (DPS) の提案とその評価をおこなった。その結果、ネットワークの構築/メンテナンスにコストをほとんどかけることなく、平均  $O(\sqrt{n})$  ステップでのピアの発見を実現した。

これらの結果から DPS は、ピアの参加/離脱が頻繁

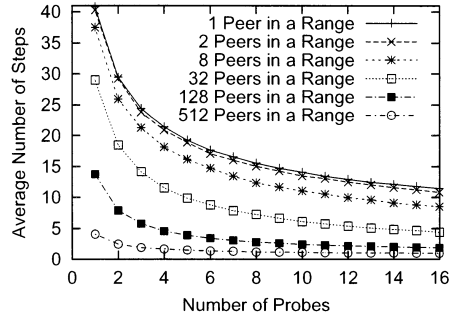


図 8: 検索範囲の割合と平均ステップ数 ( $n = 4096$ ).

に起こるようなピアの数が数千から数万の規模のネットワークで特にその効果を発揮すると考えられる。今後の課題としては、性能の解析的評価、耐故障性の保障、システムの実装があげられる。

## 参考文献

- [1] Google. <http://www.google.co.jp/>.
- [2] PinPost. <http://www.pinpost.com/>.
- [3] Gnutella. <http://www.gnutella.com/>.
- [4] I. Clarke, O. Sandberg, B. Wiley and T. W. Hong. Freenet: A Distributed Anonymous Information Storage and Retrieval System. LNCS, 2009, 2000.
- [5] S. Ratnasamy, P. Francis, M. Handley, R. Karp and S. Shenker. A Scalable Content-Addressable Network. In *Proc. SIGCOMM'01*, pages 161-172, 2001.
- [6] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek and H. Balakrishnan. Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications. In *Proc. SIGCOMM'01*, pages 149-160, 2001.
- [7] K. Aberer, P. Cudré-Mauroux, A. Datta, Z. Despotovic, M. Hauswirth, M. Puceva and R. Schmidt. P-Grid: A Self-organizing Structured P2P System. *SIGMOD Record*, 32(2): 29-33, 2003.
- [8] K. Hildrum, J. D. Kubiatowicz, S. Rao and B. Y. Zhao. Distributed Object Location in a Dynamic Network. In *proc SPAA '02*, pages 41-52, 2002.