P2P を活用した小規模データベースの集約化

藤田昭人[†]　石橋勇人[†]
大西克実[†]　中野秀男[†]

インターネットから学術論文のデジタル・データが容易に入手できる今日，研究者個人が収集した論文データを管理する個人アーカイブの需要が高まっている．個人アーカイブは単に個人が所有する文献データを保管・管理するだけでなく，それらを相互接続することによる研究者間での文献情報の共有を支援する．さらに個人アーカイブによる大規模ネットワークが形成されれば，既存の論文アーカイブ・サービスの収蔵論文数に匹敵する仮想的な論文アーカイブとして機能する可能性がある．

本論文では，広域ネットワーク上に分散した小規模データベースである個人アーカイブの集約化を実現する上での技術的な課題について論じるとともに，その解決策として既存のデータベース管理システム (DBMS) と分散ハッシュテーブル (DHT) を組み合わせたデータ共有モデルを提案する．

A Method for Integration of Small-Scale Databases with P2P Data Sharing

Akito Fujita,[†] Hayato Ishibashi,[†] Katsumi Onishi[†]
and Hideo Nakano[†]

The digital data of the science paper can be easily acquired from the Internet today. The demand for an Personal Archive to manage the paper that the researcher collected has risen. An Personal Archive does not only keeps and manages the document data that reasearcher owns, and they are connected mutually and sharing document information among researchers is supported. Also there is a possibility of functioning as a virtual science paper archive that equals the number of storing paper of existing science paper archive services if the large scale network with an Personal Archive is formed.

In this paper, a technical problem consolidating an Personal Archive that is the small-scale data base distributed on the wide area network is achieved is discussed. It proposes data common model by whom existing Database Management System (DBMS) and Distributed Hash Table (DHT) are combined as the solution.

1. はじめに

様々なコンテンツがインターネットを介して流通する今日，研究論文も著者自身の手により電子データの形でインターネット上に公開されるケースが増えている．また CiteSeer[1)] のように，インターネットに公開される研究論文を対象にしたアーカイブ・検索サービスも登場するようになった．このような有益なサービスを活用することにより，論文の収集作業の効率は格段に改善した．その結果，今日の研究者は参考文献や類似研究に関する大量の論文の電子的コピーを手元に保管している．大量の論文データが容易に入手できる今日の状況，さらに今後この傾向は加速されていくであろう見通しを考えると，これらの研究者個人が収集した電子データを研究者自身の手で作成するアーカイブの需要は拡大していくと思われる．

この研究者自身の手で作成するアーカイブでは，CiteSeer のような論文アーカイブ・サービスと連携して研究者が必要とする既存の論文を効率よく収集・保管できることが期待される．しかし，このような論文アーカイブは非営利で運営されているサービスであることが多く，サービスを持続的に維持していくことに多くの困難があることが指摘されている[2)]．

本研究は複数の研究者が個人的に作成する論文アーカイブ（以降，個人アーカイブ）相互でのデータ共有を図り，CiteSeer のような論文アーカイブ・サービスとの連携を維持しつつ，そのサービス・システムへの負荷を軽減することを企図するものである．特に本論文では研究者の論文アーカイブにおいて中核となるであろう小規模データベースに対して，P2P 技術を活用して分散共有を図る方策について論じる．

† 大阪市立大学大学院 創造都市研究科
Graduate School for Creative Cities,
Osaka City University

2. 個人アーカイブを実現する上での課題

今日，個人が使用する PC を始めとするコンピュータは十分な処理能力とデータ蓄積容量を有するので，個人がデータの収集・管理を行う個人アーカイブを収容することは可能である．個人アーカイブには自らの論文だけでなく参考文献として引用している論文も同時に収容すれば，自らの研究に関わる文献を一括して管理できるので論文執筆などに有効に活用することができる．

このように個人アーカイブでは所有者である研究者が必要に応じて任意に選択した論文の登録・管理を行うため，それ単体では公開論文アーカイブのように論文データを内容に偏りなく網羅することはできないが，ネットワーク接続した複数の個人アーカイブを集約した分散アーカイブを実現できれば，既存の公開論文アーカイブに匹敵する収蔵量のサービスが実現可能だと思われる．個人アーカイブ相互で登録データの共有ができれば，公開論文アーカイブ・サービスへのアクセス量を大きく削減でき，公開論文アーカイブのシステム負荷の軽減に寄与することもできる．

しかしながら，個人アーカイブは所有者の日常的な活動に密着したところで運用されるため，個人アーカイブが稼動するコンピュータやそのネットワーク接続には様々な形態が想定される．このような多様性の高いノードから構成される分散アーカイブを実現するためには次のようなさまざまな課題がある．

2.1 分散アーカイブとしてのスケーラビリティ

個人アーカイブは所有者自身が必要とする論文を選択的に登録するので，それ単体では収蔵内容に特定の傾向を持つ小規模なアーカイブとなる．これらを集約した分散アーカイブが公開論文アーカイブに匹敵する論文データを網羅するためには十分に多くの個人アーカイブを集約する必要がある．

一般的なアーカイブ・システムではデータの格納や管理のため，バックエンド・ストレージとしてリレーショナル・データベースなど汎用データベース・システムを用いる．個人アーカイブを集約した分散アーカイブを実現するためには，分散データベースや大規模データーベースの技術を活用したバックエンド・ストレージが考えられるが，この場合に問題となるのがそのスケーラビリティである．一般的な分散データベースや大規模データーベースの構成ノードに対するスケーラビリティは数百ノード程度が上限であり，インターネット規模での分散が可能な他の分散システムに比べてスケーラビリティの極端な低さが指摘されている[3)]．

個人アーカイブを集約した分散アーカイブでは，集約する個人アーカイブの総数が増大するにしたがって収容データの網羅性や冗長性が向上する．したがって，構成ノードに対するスケーラビリティの高いバックエンド・ストレージのための技術が求められる．

2.2 ネットワーク接続状態の動的な変化

ノート PC などの携帯型情報機器の性能が向上した今日，これらの機器の上で個人アーカイブを稼動させる可能性は大きく，その場合には常時安定したネットワーク接続状態が維持されない．頻繁にネットワークへの接続と離脱を繰り返すこれらのノードにおいて，ネットワーク非接続時にも個人アーカイブとしての最低限の機能が保証され，ネットワーク接続時には自律的に分散アーカイブを構成することが望まれる．

2.3 メタデータのフォーマット

文献のメタデータはアーカイブシステムにおいて文献検索の対象となるデータである．個人アーカイブの集約するためには，原則としてメタデータのフォーマットは全ての個人アーカイブにおいて統一されていなければならない．

文献アーカイブのメタデータの一般的なフォーマットとしては Dublin Core [4)] が知られているが，個人アーカイブに参考文献を登録する場合には文献の多くは複数の公開論文アーカイブから HTTP プロトコルや Dublin Core に基づく収集プロトコルである Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH) [5)] などを使ってメタデータが取得される．公開論文アーカイブが使用するメタデータは Dublin Core に準拠しているものの独自の拡張が加えられている可能性があり，その互換性が保証されているとは限らない[6)]．

また個人アーカイブでは，個人が所有する特性を積極的に活用して独自のメタデータに検索情報を追加するできることが期待される一方，Bib 形式のような所有者にとって利便性の高い他の一般的なメタデータ・フォーマットへ変換できるよう統一性が保たれていることも望まれる．

このように個人アーカイブの利便性を考慮すると，それがサポートメタデータのフォーマットは柔軟性・拡張性と一貫性・統一性という相反する特性を同時に満たすことが求められる．個人アーカイブを集約した分散アーカイブの実現を検討する上で，これは技術的に非常に困難な課題である．

3. P2P 技術による分散アーカイブの集約化

個人アーカイブの集約を行う場合相互接続はインターネットを介することになるが，個人アーカイブが集約された分散アーカイブは広域に分散した自律的なノードから構成されるピア・ツー・ピア (P2P) ネットワークと見なすことができる．P2P ネットワークは構成ノードに対するスケーラビリティが高く，ネットワークへの接続状態へ依存性が少ない自律分散システムのためのアーキテクチュアである．

P2P ネットワークを基盤として分散アーカイブへ活用する方策については既に議論されているが[7]，P2P 分散アーカイブ・システムを実現するための最も重要な技術的な課題は P2P ネットワークに離散するコンテンツ・データの管理方法である．W.S. Ng らは既存の P2P システムと一般的な分散データ管理システムを比較し，次の４つの相違点を挙げている[8]．

(1) **ノードの接続・離脱**
分散データ管理システムではノードは制御された方法でネットワークへの接続・離脱されるが，P2P システムでは任意のタイミングで自律的にネットワークへの接続・離脱を行う．

(2) **クエリ対象の配置**
分散データ管理システムではクエリ対象の配置は明確になっているが，P2P システムでは配置は不明確でノード間でクエリが伝播される．

(3) **クエリへの応答**
分散データ管理システムではクエリは全てのノードからの応答に基づくことが保証されるが，P2P システムでは全てのノードからの応答に基づくとは限らない．

(4) **データの構造定義（スキーマ）**
分散データ管理システムでは共有されるスキーマは予め定義されているが，P2P システムでは多くの場合スキーマは持たない．

前節の個人アーカイブの要求を考慮した場合，上記の P2P システムの特性のうち (1), (2) はその要求に適したメリットとして理解できる．しかし (3) に関しては到達可能なノードからの応答に基づくことを保証する必要があり，また個人アーカイブでは Dublin Core などに基づくスキーマをサポートしなければならないので (4) とは条件が異なる．

3.1 Structured P2P と DHT

(3) のクエリの結果が到達可能なノードからの応答に基づくことを保証する問題を解決する方策として Structured P2P の活用が考えられる．CAN[9]，Chord[10] Pastry [11]，Tapestry [12] などの代表的な研究事例により注目を集める Structured P2P は，分散ハッシュテーブル (DHT) などによる構造化されたオーバーレイ・ネットワークを形成し，ネットワークを構成するノード数によりルックアップに必要なメッセージ数を決定できる特徴を持つ．データのルックアップにはハッシュを使用しなければならない制約はあるものの，Gnutella などに代表される既存の Unstructured P2P に比較してルックアップの予測可能性と網羅性の点で優位であることが知られている．

分散ハッシュテーブル (DHT) は Structured P2P のオーバレイネットワークを形成するための手段の１つとして知られるデータ保存方法で，キーを使ったデータの格納と検索を行う汎用的なインターフェースを提供する．分散ハッシュテーブルでのノードは一般的なハッシュ・テーブルにおけるバケット (Bucket) に類似し，任意のキーを与えるとそれに対応するデータを格納するノードを一意に決定するできる．SHA-1 [13] などのハッシュ関数 (Secure Hash Function) を使って生成されたキーが用いられることが一般的で，その場合にはデータは論理的に 2^{160} の広大なハッシュ空間上に配置される．

一般に一方向性ハッシュ関数である SHA-1 はメッセージ・ダイジェスト (message digest) [14] として活用されるが，分散ハッシュテーブルでもその入力がわずかでも変わると全く違った出力が得られる特性を有効に活用している．分散ハッシュテーブルに格納するデータ自身を入力として得られた SHA-1 出力はコンテント・ハッシュ (content hash) と呼ばれ，キーの値が同じであれば格納されているデータ内容も同一と見なせることから，格納データの比較など分散システムにおいて必須の処理の効率化に役立つ．またコンテント・ハッシュとは反対にデータ内容が変更されても値が変わらないキーをコンシステント・ハッシュ (consistent hash) [15] と呼び，分散ハッシュテーブルに格納されているデータのインデックスなどに利用される．

3.2 個人アーカイブのデータ共有モデル

大規模データベースや分散データベースの研究領域でも Structured P2P を活用した研究事例が存在し，CAN を活用した PIER [16] や Chord を活用した AmbientDB [17] などが報告されている．しかし，これらの既存の P2P データベースの研究はノードのネットワーク常時接続を前提としている事例が多いため，ネットワーク非接続時にもアーカイブとして機能する要求を満たさない．

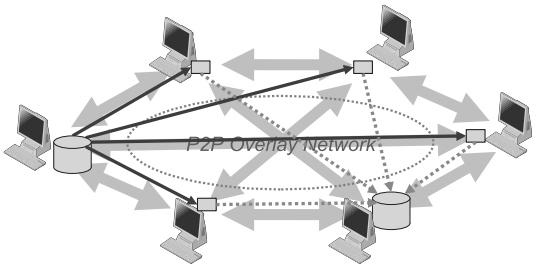

図 1 個人アーカイブのデータ共有モデル

個人アーカイブでは，まず既存の DBMS を利用したスタンドアローンのデータベースをバックエンド・ストレージとする小規模文献アーカイブを構築し，P2P データ共有技術を使ってデータベースに格納されるメタデータを選択的に抽出し，その結果から新たなデータベースを生成するデータ共有モデルを採用する．ユーザーは DBMS を経由してメタデータの登録・検索・参照を行うものとする．

個人アーカイブのデータ共有モデルを図 1 に示す．データベース内部では個々のメタデータをデータ・エンティティとするテーブルが複数存在するものとし，ユーザーはこれらのテーブルを対象に次のような操作を行う．

- **メタデータの格納**
  メタデータのテーブルは一般的な DBMS によって作成される．テーブル名は P2P オーバーレイ・ネットワーク内でユニークとなるような文字列（例えば作成者のメールアドレスやノードの IP アドレスなど）を使用する．
- **メタデータの公開**
  任意のメタデータ・テーブルが公開操作により P2P オーバーレイ・ネットワークに公開される．この際，テーブルのハッシュ化を行う．（ハッシュ化操作の詳細は後述）
- **メタデータの取得**
  ネットワーク接続時には P2P オーバーレイ・ネットワーク内に存在するメタデータ・テーブルから選択条件を指定してメタデータを取得できる．取得したメタデータはローカルの論文メタデータ・テーブルに格納される．

これらの操作は P2P データ共有機能を実現するソフトウェアを介して行う．

3.3 メタデータ・テーブルのハッシュ化

個人アーカイブ間のデータ共有には，DHT の中でも最もシンプルなアルゴリズムである Chord を採用する．Chord などの DHT を使ったデータ共有ではハッシュの利用が必須であるが，ハッシュ関数として使用する SHA-1 の処理コストがシステム・パフォーマンスに大きな影響を与えることが指摘されている[18]．したがって SHA-1 によるエンコーディングの頻度を極力抑えられるようなデータ構造の定義が重要となる．Chord を活用した文書アーカイブ・システムの研究事例である OverCite [19] では，メタデータなどは文書単位に区分してハッシュ化されるデータ構造を採用している．

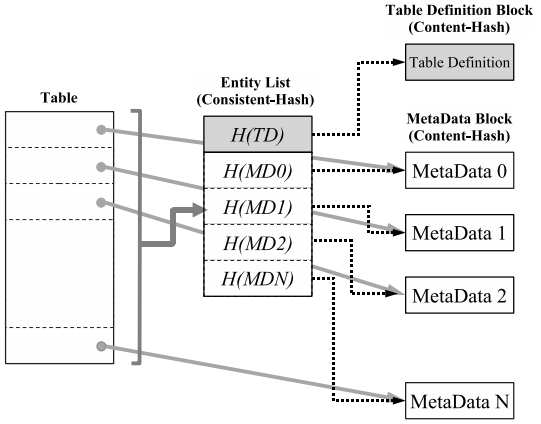

図 2 テーブルのハッシュ化

個人アーカイブのハッシュ化されたテーブルの構造を図 2 に示す．テーブルはメタデータ・ブロック，テーブル定義ブロック，エンティティ・リストに分割してハッシュ化される．メタデータ・ブロックは個々の論文のメタデータをコンテント・ハッシュ化したブロックである．テーブル定義ブロックはテーブル定義情報をコンテント・ハッシュ化したブロックである．エンティティ・リストはメタデータ・ブロックとテーブル定義ブロックのハッシュ値のリストで，テーブル名をハッシュ化したコンシステント・ハッシュ・ブロックとして格納される．

任意のノードでメタデータの公開を行う場合，指定されたテーブルを元にメタデータ・ブロック，テーブル定義ブロック，エンティティ・リストを生成し，各々をハッシュ値に基づくノードへ格納する．さらにエンティティ・リストを受け取ったノードではその後のメタデータの取得に備えて，エンティティ・リストに対応するテーブルをローカル・データベース上に再構築する．

任意のノードでメタデータの取得を行う場合にはテーブル名のハッシュ値と選択条件を指定してリクエストを発行する．エンティティ・リストを格納するノードではリクエストで指定された選択条件で該当す

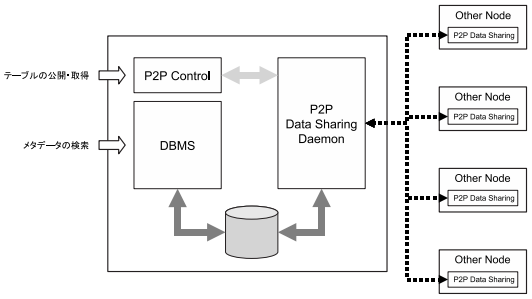

図 3 個人アーカイブのアーキテクチュア

るテーブルの検索を行い，リプライとして該当テーブルのテーブル定義ブロックのハッシュ値と選択条件に合致するメタデータ・ブロックのハッシュ値のリストを返す．メタデータの取得をリクエストしたノードは，各ノードからリプライとして得たメタデータ・ブロックのハッシュ値のリストを元に新たなローカル・テーブルを作成する．

このデータ共有モデルのメリットは，任意の複数のテーブルから選択的抽出を行った論文メタデータが新たなローカル・テーブルとして格納される点である．利用者はネットワーク接続時に明示的に論文メタデータの選択的抽出を行い，ローカル・テーブルを作成する．ひとたびローカル・テーブルが作成されれば以降ネットワーク非接続時でも文献の検索が可能となる．更にこのモデルでは DBMS に外部にてデータ共有機能を実現すればよいので DBMS そのものの分散化を図る必要がない．

一方，このデータ共有モデルのデメリットは，一般の分散データベースとは異なり選択的抽出を行ったローカル・テーブルが継続的な自動更新がされない点にあるが，ネットワーク接続状態が頻繁に変化する個人アーカイブの動作環境での継続的な自動更新を実現するコストは大きく，原則として特定の個人のみの使用を想定する個人アーカイブとしては過剰な機能であり，ローカル・テーブルの更新が必要な場合にはメタデータの取得を再実行を行えば良い．

4. 個人アーカイブのアーキテクチュア

個人アーカイブのアーキテクチュアを図 3 に示す．ソフトウェアは DBMS，P2P Data Sharing Daemon，P2P Control の 3 つのコンポーネントから構成される．

DBMS は一般的なデータベース管理システムで，テーブルの作成，メタデータの格納の機能を提供する．ユーザーがメタデータの検索・参照を行う場合のインターフェースとしても機能する．

P2P Control は P2P Data Sharing Daemon を制御するコンポーネントで，ユーザーはこのコンポーネントを介して P2P Data Sharing Daemon にメタデータの公開・取得を要求する．

P2P Data Sharing Daemon はテーブルによるデータ共有を司るコンポーネントで，P2P Control からの要求に応じてメタデータの公開・取得を実行する．コンポーネント内部では次の機能が実装される．

- DBMS のテーブルに対する各種操作
  （検索，ダンプ，リストア）
- エンティティ・リスト，
  テーブル定義ブロック，
  メタデータ・ブロックのハッシュ化
- ハッシュ化データの転送
- ハッシュ化データのルックアップ

ハッシュ化データのルックアップは，原則として DHT によるルックアップ・プロトコルの実装であればいずれでも動作可能だと考えられる．また，既に様々な DHT ルックアップ・プロトコルの実装が入手可能であるが，現時点でのプロトタイプは Chord プロトコルをベースとした設計となっている．現在，DBMS には SQLite [20]，Chord プロトコル実装には Internet Indirection Infrastructure (i3) [21] の実装の一部を利用したプロトタイプの開発を進めている．

5. おわりに

本論文では，近年デジタル・データとして流通するようになった学術論文の蓄積，管理，分散共有が可能な個人アーカイブを提案し，その実現のための技術的課題として，広域・大規模分散に対応するスケーラビリティの達成，無線ＬＡＮなど接続状態が動的に変化するネットワークへの対応，分散共有に伴う文献のメタデータのフォーマットに関わる問題を指摘した．

更にこれらの課題を解決するための方策として，メタデータをテーブル単位で集約し，DHT を活用して広域・大規模分散を図るデータ共有モデルを提案した．

このモデルはユーザーに複数の文献をテーブルとしてひとまとめにして取り扱う方法を提供し，明示的にテーブルの公開・取得を行うことでネットワークの接続状態に応じたアーカイブの利用が可能となる．また，分散データベースなどにみられる広域・大規模分散データ管理特有の複雑さも回避できるため，主に個人が利用するであろう個人アーカイブのためのデータ共有モデルとして適している．

個人アーカイブの開発は初期段階にあり，実用的なシステムを実現するためには数多くの技術的課題が残されている．ハッシュ化データの削除，複製される

テーブルの管理，メタデータの拡張への対応，公開メタデータの共有範囲の指定などの項目が今後検討すべき課題である．これらの項目は，プロトタイプの作成・評価の後に検討を行う予定である．

参 考 文 献

1) Lawrence, S., Giles, C.L. and Bollacker, K.: Digital Libraries and Autonomous Citation Indexing, *IEEE Computer*, Vol.32, No.6, pp.67–71 (1999).
2) Stribling, J.: OverCite: A Cooperative Digital Research Library, Master's thesis, Massachusetts Institute of Technology (2005).
3) Hellerstein, J. M., Lanham, N., Loo, B. T., Shenker, S. and Stoica, I.: Querying the Internet with PIER., *29th International Conference on Very Large Data Bases (VLDB '03)* (2003).
4) ウィキペディア：*Dublin Core*, `http://ja.wikipedia.org/wiki/Dublin_Core` (2006).
5) ウィキペディア：*Open Archives Initiative Protocol for Metadata Harvesting*, `http://ja.wikipedia.org/wiki/Open_Archives_Initiative_Protocol_for_Metadata_Harvesting` (2006).
6) PSU, I.: *CiteSeer OAI Compliace*, `http://citeseer.ist.psu.edu/oai.html` (2006).
7) Bhatia, K.: *Peer-To-Peer Requirements On The Open Grid Services Architecture Framework*, http://www.ggf.org/documents/GFD.49.pdf (2005).
8) Ng, W.S., Ooi, B.C., Tan, K.-L. and Zhou, A.: PeerDB: A P2P-based System for Distributed Data Sharing, *Intl. Conf. on Data Engineering (ICDE'03)* (2003).
9) Ratnasamy, S., Francis, P., Handley, M. and Karp, R.: A scalable content-addressable network, *Proceedings of SIGCOMM*, ACM (2001).
10) Stoica, I., Morris, R., Karger, D., Kaashoek, M.F. and Balakrishnan, H.: Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications, *Proceedings of the ACM SIGCOMM '01 Conference*, San Diego, California (2001).
11) Rowstron, A. and Druschel, P.: Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems, *IFIP/ACM International Conference on Distributed Systems Platforms (Middleware)*, pp. 329–350 (2001).
12) Zhao, B. Y., Kubiatowicz, J. D. and Joseph, A. D.: Tapestry: An Infrastructure for Fault-tolerant Wide-area Location and Routing, Technical ReportUCB/CSD-01-1141, UC Berkeley (2001).
13) Service, N. T. I.: *FIPS 180-2 - Secure Hash Standard*, U.S. Department of Commerce/NIST, Springfield, VA (2002).
14) アンドリュー・S・タネンバウム：コンピュータネットワーク　第４版，日経ＢＰ (2003).
15) Karger, D., Lehman, E., Leighton, T., Levine, M., Lewin, D. and Panigrahy, R.: Consistent Hashing and Random Trees: Distributed Caching Protocols for Relieving Hot Spots on the World Wide Web, *ACM Symposium on Theory of Computing*, pp.654–663 (1997).
16) Huebsch, R., Chun, B., Hellerstein, J.M., Loo, B.T., Maniatis, P., Roscoe, T., Shenker, S., Stoica, I. and Yumerefendi, A.R.: The Architecture of PIER: An Internet-Scale Query Processor, *The Second Biennial Conference on Innovative Data Systems Research (CIDR)*, Asilomar, CA (2005).
17) Boncz, P. and Treijtel, C.: AmbientDB: Relational Query Processing in a P2P Network, *Proceedings of the International Workshop on Databases, Information Systems and Peer-to-Peer Computing DBISP2P* (2003).
18) Rhea, S., Eaton, P., Geels, D., Weatherspoon, H., Zhao, B. and Kubiatowicz, J.: Pond: The oceanstore prototype, *Proceedings of the Conference on File and Storage Technologies*, USENIX (2003).
19) Stribling, J., Li, J., Councill, I.G., Kaashoek, M.F. and Morris, R.: OverCite: A Distributed, Cooperative CiteSeer, *Proc. of the 3rd NSDI* (2006).
20) Hipp, Wyrick & Company, I.: *SQLite home page*, `http://www.hwaci.com/sw/sqlite/index.html` (2006).
21) Stoica, I., Adkins, D., Zhuang, S., Shenker, S. and Surana, S.: Internet indirection infrastructure (2002).