

## P2P データ共有技術を活用した Web コンテンツの分散管理

藤田 昭人  
大阪市立大学大学院  
Fujita.Akito@gsec.osaka-cu.ac.jp

石橋 勇人  
大阪市立大学大学院  
ishibashi@media.osaka-cu.ac.jp

大西 克実  
大阪市立大学大学院  
onishi@media.osaka-cu.ac.jp

中野 秀男  
大阪市立大学大学院  
nakano@media.osaka-cu.ac.jp

負荷分散が容易で耐障害性に優れる P2P データ共有技術は、今日の多数・大容量の Web コンテンツの管理に関わる問題の解決に効果を発揮する。本論文では P2P データ共有技術、特に structured P2P に関する研究事例を紹介し、それを活用した Web コンテンツの分散管理システムを提案し、その実現方法や解決すべき技術的課題について議論する。

## Distributed Management of Web Contents using P2P Data Sharing

Akito Fujita  
Osaka City University  
Fujita.Akito@gsec.osaka-cu.ac.jp

Hayato Ishibashi  
Osaka City University  
ishibashi@media.osaka-cu.ac.jp

Katsumi Onishi  
Osaka City University  
onishi@media.osaka-cu.ac.jp

Hideo Nakano  
Osaka City University  
nakano@media.osaka-cu.ac.jp

P2P data sharing which provides provable guarantees for robustness and load-blance demonstrates an effect to management of large scale Web contents. This paper introduce research example about P2P data sharing, proposes Web Contents Management System, discuss about the realization method or the technical subject which should be solved.

### 1. はじめに

Web サービスを介したデジタル・コンテンツの流通が日常化した今日、Web サーバーに収容されるコンテンツの巨大化、複雑化は進み、その流通総量も拡大を続けている。このようなトレンドを考慮すると、多数・大容量の Web コンテンツの管理方法は今後の重要な技術課題となると推測される。元より HTML を基本とする Web コンテンツには、コンテンツ管理を複雑化させる次のような問題点があった。

- 一般的な Web ページは HTML、画像、スタイルシートなど複数のファイルから構成されることが多く、ファイル単位での管理が複雑。
- HTML の特徴であるハイパー・リンクは他ページを参照する機構であり、参照されるページの配置や内容の整合性の維持は非常に煩雑である。

- 原則として Web サーバーはコンテンツ参照をサポートするものであり、個々のコンテンツの更新のためには何らかのアクセス手段を別途用意する必要がある。

Web サービスにおいてこのところ注目され始めているコンテンツ・マネージメント・システム (CMS) は、文字通り Web コンテンツの管理を行うことを目的としているが、既存の Wiki や Blog など同様の仕組みを活用している。すなわち、これらの Web サービスは Web ブラウザーのみを使用し、HTML の知識のない利用者でも見栄えのするコンテンツが作成できるのだが、コンテンツ制作者から HTML ファイルを隠蔽する効果もある。一般的な Wiki, Blog, CMS はいずれもサーバーサイドスクリプト言語を利用した Web アプリケーションとして実装されるので、前述のコンテンツ管理の問題はアプリケーションの実装上の問題に帰結される。

しかし、このような Web アプリケーションには、その構造や特性から次のような問題が懸念される。

- 一般的なサーバーサイドスクリプト言語はインタプリタとして実現されており、クライアントからのアクセス時に動的にコンテンツを生成する。したがって、静的コンテンツと比較して Web サーバーによりも大きな負荷がかかる。Wiki, Blog, CMS ではコンテンツ管理は Web サーバー中に閉じているので、Web サーバーが稼動するサーバー・システムに負荷が集中してしまう。
- サーバーが稼動するサーバー・システムに負荷が集中することになる。サーバー・システムがクラッシュしてしまった場合には、Web サーバー中にある既存のコンテンツも失われてしまう。

以上のような問題を解決するため、本論文では負荷分散が容易で耐障害性に優れる P2P データ共有技術を活用した Web コンテンツの分散管理システムを提案する。

## 2. P2P ネットワークによるデータ共有

P2P ネットワークは、構成するマシン(ピア)がサーバーとしてもクライアントとしても機能するアーキテクチャであり、個々のピアは自律的に動作し、他のピアと直接協調する。P2P ネットワークには次のようなメリットがある。

- 拡張性 - 全てのピアが持つリソースをネットワーク全体で活用し、ネットワーク全体のリソースは新たなピアの参加に応じて自然に拡大する。
- 信頼性 - ネットワークを流通するデータは複製され分散して格納されるので、単一障害点 (single point of failure) を持たない。
- 管理の容易性 - ピアは自律的に組織化するので、サーバーを要求に応じて展開する必要はなく、自動的に冗長化や負荷分散が図れる。
- 既存リソースの効率的利用 - ピアとして既に使っているコンピュータをそのまま活用することができる。

このような分散システムとしての P2P の優れたメリットは、ファイル共有、マルチプレーヤー・オンライン・ゲーム、協調型アプリケーション、分散コンピューティング、アドホック・ネットワークなどの実現に活用できる。

近年、分散ハッシュ・テーブル (DHT) [10] を活用した structured P2P と呼ばれる研究事例が、第2世代の P2P 技術として注目される。

### 2.1. Structured P2P

Structured P2P の特徴の1つは分散ハッシュテーブル(DHT)が利用されることである。DHT はハッシュ・テーブルのデータ構造を P2P ネットワークを構成するピア上に分散配置したもので、一般的なハッシュ・テーブルと同様にキーと値のペアの挿入、ルックアップ、削除の操作ができる。この DHT を P2P のデータ探索(ルーティング)に活用した技術は DHT ルーティングと呼ばれている。

既存の flooding と呼ばれるヒューリスティックなルーティングを使用するオーバーレイ・ネットワークはランダム・グラフを形成するのに対し、DHT ルーティング使用するオーバーレイ・ネットワークでは構造化されたグラフを形成するところから、DHT ルーティングを用いる P2P ネットワークは Structured P2P、そうでない P2P ネットワークは Unstructured P2P と呼ばれている。

P2P ネットワークでの DHT によるルーティングのメリットとしてはルックアップの高速性と確実性があげられる[4]。DHT ルーティングをヒューリスティックなルーティングと比較すると、問い合わせが分散ハッシュをキーとしたルックアップに制限されるものの、問い合わせに伴って発生するトラフィック、検索結果に対する保障の点では DHT が優れている。

この DHT ルーティングの実現方式に関わる著名な研究事例として Chord [1], Pastry [2], Tapestry [3] などが上げられるが、以降では本研究に関連の深い、代表的な事例として、Cooperative File System (CFS) と OceanStore について説明する。

### 2.2. Cooperative File System (CFS)

MIT で開発された Cooperative File System (CFS) [5] は DHT ルーティング機構として Chord を活用した分散ファイルシステムである。CFS は原則的にはリードオンリーのファイルシステムであり、登録ファイルは、その所有者のみがファイル単位を更新することができる。複数のピアが集まって単一のファイル・ツリーを形成する、参照型のファイル共有を実現している。

CFS のソフトウェア構成を図 1 に示す。CFS は、ファイルシステムへのインタフェースの FS レイア、ファイル構成するデータ・ブロックの管理を司る DHash レイア、該当するデータ・ブロックが格納されているピアを検索する Chord レイアの大変シンプルな3レイア構造で実現されている。

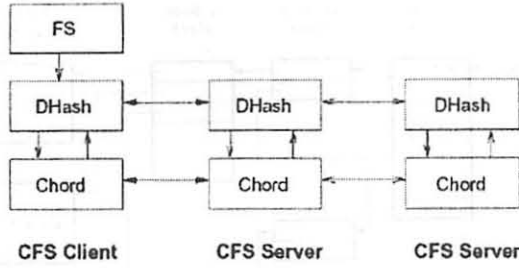


図 1: CFS のソフトウェア構成

CFS のデータ・モデルを図 2 に示す。図からもわかるように、CFS は Unix ファイルシステムと同じブロック構造を持っている。各データ・ブロックはブロック識別子で指定されるが、このブロック識別子にはデータ・ブロックの内容のハッシュ値が使用される。これらのデータ・ブロックを参照するルートとして Root Block が定義されているが、そのブロック識別子には Root Block の所有者の公開鍵を使う。

### 2.3. OceanStore

UCB で開発中の OceanStore は、インターネット・ワイドの永続的なアーカイブ・ストレージの実現を目指した、インフラストラクチャで、電子メールやグループウェアといったコミュニケーション・アプリケーション、巨大なデータを永続的に保存できるアーカイブ・ストレージ、センサーネットワークのデータ収集・配布、ACID セマンティックス [9] に基づくデータベース・アプリケーションなど、多岐にわたる分散型アプリケーションの開発に役立つ共通のインフラストラクチャの実現を目指している[6]。図 3 に OceanStore のソフトウェア構成を示す。図 4 に Pond のデータ・オブジェクトのモデルを示す。

Pond の内部ではファイルはデータ・オブジェクトと表現され、複数のフラグメントから構成される。Pond でも原則的には Chord と同じようにデータは分割して管理されるが、データ・オブジェクトの更新ができることが Chord との大きな違いである。

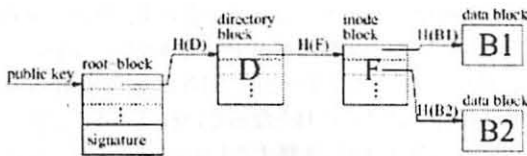


図 2: CFS のデータ・モデル

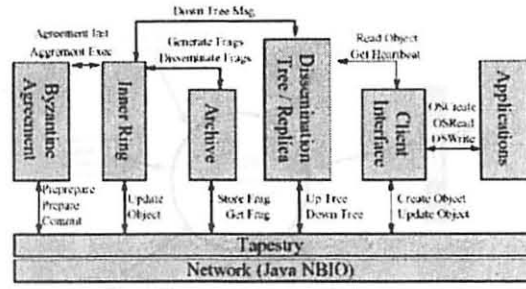


図 3: OceanStore のソフトウェア構成

データ・オブジェクトはアクティブ GUID (AGUID) によって識別されるが、AGUID がポイントするのはバージョン GUID (VGUID) と呼ばれる識別子のリストになっており、個々の VGUID によってデータ・オブジェクトの各バージョンがポイントされる。OceanStore におけるデータ・オブジェクトの更新では、新しいバージョンを登録する内部操作が行われ、実際にデータ変更が行われるのは AGUID がポイントする VGUID のリストに対してのみである。この方法によってデータ・オブジェクトの更新に伴ってデータ変更を行わなければならないフラグメントを最小限に留めることができる。

### 3. P2P データ共有技術を活用した Web コンテンツの分散管理システム

本研究では、ソフトウェア構成の簡単な Chord/CFS の実装をベースに、P2P データ共有技術を活用した Web コンテンツの分散管理システムの試作開発を計画している。

このシステムは、Web サーバー(以降サーバー)と Web ブラウザ(以降ブラウザ)に P2P データ共有機能を追加して、ブラウザからサーバー内のコンテンツの新規登録や更新、削除をできることが基本的なアイデアである。試作システムの概要は次の通りである。

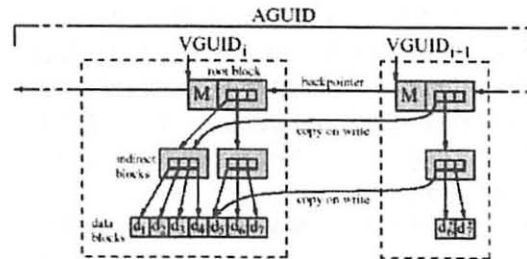


図 4: Pond のデータ・オブジェクトのモデル

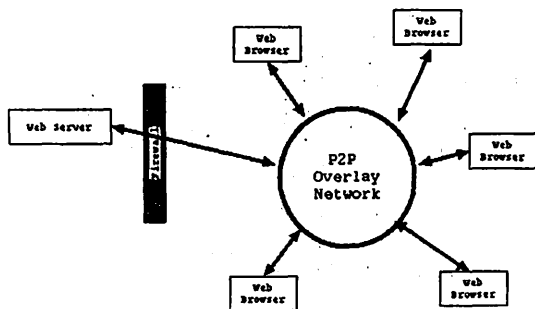


図 5: 想定するオーバーレイ・ネットワーク

- 原則としてイントラネット(IP プライベート・ネットワーク)での稼動を想定する。但し Web サーバーが稼動するホストに関してはインターネット(IP グローバル・ネットワーク)に接続され、ファイアウォール経由で接続することも考慮する。
- オーバーレイ・ネットワークに構成するピアは、Web サーバーの稼動するホスト、あるいは Web ブラウザが稼動するホストとする。
- P2P 機能は Web サーバー・ソフトウェアおよび Web ブラウザ・ソフトウェアに追加される

試作システムが想定するネットワークを図 5 に示す。P2P 機能を持つ Web サーバー・ホストは、イントラネットワーク内、あるいはファイアウォール越しに接続され、初期のオーバーレイ・ネットワークが形成される。P2P 機能を持つ Web ブラウザ・ホストは Web ブラウザ・ソフトウェアが起動されると、オーバーレイ・ネットワークに接続される。利用者は Web ブラウザ・ソフトウェアを操作して、オーバーレイ・ネットワーク内で共有される任意のコンテンツ・データを作成・参照・更新することができる。Web ブラウザ・ソフトウェアが停止すると、Web ブラウザ・ホストはオーバーレイ・ネットワークから離脱する。

Web ブラウザ・ソフトウェアから P2P ネットワーク上のコンテンツを参照する場合には、URL のプロトコル表記部分に P2P ネットワークを示すキーワード chord を使用する。ブラウザから P2P ネットワーク上のコンテンツを新規作成・更新する場合にはツールバー上に専用のボタンを用意し、カレント・ウィンドウのコンテンツの登録操作を行うものとする。

本研究でのソフトウェア開発の前提として、構造がシンプルな Chord/CFS の実装をベースに P2P 機能を実現するものとする。

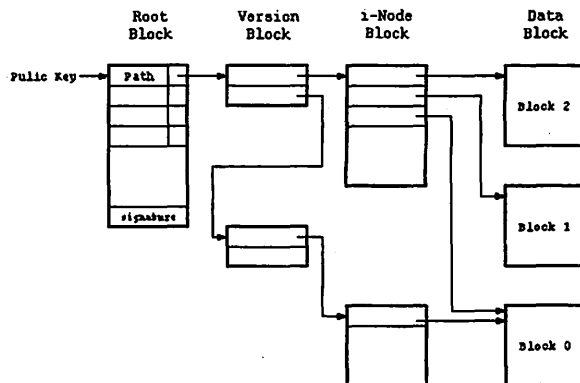


図 6: 試作開発システムのデータ・モデル

#### 4. システム実現のための技術的課題

両者の実装では、Apache あるいは Firefox がサポートする Module あるいは Extension のための固有のインターフェースに関する技術検討も行う必要があるが、ここでは両者に共通する P2P データ共有に関わる技術検討に関する説明に焦点を絞る。

まず、試作開発システムのデータ・モデルを図 6 に示す。全てのピアは Root Block を持つ。各ピアの Root Block は、パス名と該当コンテンツの最新バージョンへの識別子が格納される。Root Block の下位には OceanStore/Pond のように、該当するコンテンツの各バージョンのバイナリ・ツリーを形成する Version Block が位置する。Root Block には最新バージョンの Version Block 識別子が格納され、この Version Block を辿ることにより全てのバージョンにアクセスすることができる。Version Block の下位には CFS と同様の i-Node Block と Data Block が存在し、コンテンツの各バージョンの実データが格納される。Data Block は OceanStore/Pond と同様に、複数のバージョンの間でできるだけ共用する。

##### 4.1. コンテンツ・データの更新

共有データの更新時の一貫性維持は分散システムでの一般的な問題であるが、ピアが任意のタイミングで接続・離脱をする P2P ネットワークでは一貫性維持のための密なコミュニケーションは望めず、内部で保持しているデータへは参照操作しかできない。CFS では原則的にリードオンリーとし、例外的に所有者のみに更新操作を認める仕様となっている。しかし、試作システムでも同じ仕様が踏襲するとコンテンツ・データは作成者しか更新できない。そこで OceanStore のようなバージョン管理の概念を導入する。システム内部で個々のコンテンツに対して所有者の異なる複数のデータセ

ットが別々のバージョンとして存在することを認めることにより、所有者以外の利用者によるコンテンツ・データの更新が可能になる。

#### 4.2. Web ブラウザ・ピアの稼働状態の影響

試作システムでは共有する対象が Web コンテンツに限定されているため、インストール等の手間を軽減するなど実務的な目的でブラウザに P2P 機能を追加することにする。しかしブラウザが P2P のピアとして振舞うため、ブラウザの起動・停止に同期してオーバーレイ・ネットワークへの接続・離脱する頻度が極端に増える可能性が高い。Chord はピアの接続・離脱のタイミングでピア間でのデータの移動が発生するため、ブラウザの起動・停止に非常に時間が掛かってしまう可能性が考えられる。この問題の直接的な解決策としては、利用者に P2P 機能の有効・無効を明示的に操作させる、あるいはブラウザが起動された時に P2P 機能をバックグラウンドで起動し、マシンが停止するまでピアとしての役割を果たし続けるといった対処法が考えられる。

しかし、このピアの接続・離脱する頻度が極端に増えた状態での P2P ネットワークの挙動は、例えば無線 LAN など接続状態の安定しないネットワーク経由でピアがオーバーレイ・ネットワークへ接続している状況などと類似すると思われるので、観察するべき価値があると考えている。Unstructured P2P システムでは、多数のピアの稼働率が低いオーバーレイ・ネットワークでも少数の稼働率の高いピアが存在すれば P2P システム全体の可用性は低下しないことが報告されている。[8] Chord/CFS の DHT ルーティング方法で同様の効果を得るのは難しいと思われるが、このような特性を得るための方策の検討は今後の課題である。

## 5. 結論

本論文では P2P によるデータ共有技術、特に Structured P2P の DHT ルーティングの応用事例とその実装方法について解説し、Web コンテンツの分散管理システムとその実装方法、技術的課題を提示した。試作システムの設計は、主にイントラネット上での運用を想定して検討したものであるが、インターネット上でも無修正で動作すると考える。但し、試作システムをインターネット上で運用する場合には、セキュリティ上の問題が懸念されるためデータの隠匿性を向上させる必要があるが、これは今後の検討課題である。さらにピアの接続・離脱する頻度が極端に増えた状態での P2P ネットワークの挙動も、P2P データ共有技術を実用化する

る上で重要な問題だと考える。まずはどのような影響がでるのか、P2P シミュレータ等を使ってその挙動を確認したいと考えている。

## 参考文献

- [1] Stoica, I. et al. *Chord: A scalable peer-to-peer lookup service for internet applications*, Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication, Aug. 2001.
- [2] Rowstron, A. et al. *Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems*, Proceedings of the IFIP/ACM International Conference on Distributed Systems Platforms (Middleware), Nov. 2001.
- [3] Zhao, B. et al. *Tapestry: An infrastructure for fault-tolerant wide-area location and routing.*, Tech. Rep. UCB/CSD-01-1141, Computer Science Division, U. C. Berkeley, Apr. 2001.
- [4] El-Ansary, S. et al. *Efficient Broadcast in Structured P2P Networks*, 2nd International Workshop on Peer-to-Peer System (IPTPS '03), Berkeley, CA, USA, Feb. 2003.
- [5] Dabek, F. et al. *Wide-area cooperative storage with CFS*, Proceedings of the Symposium on Operating Systems Principles (SOSP), Oct. 2001.
- [6] Kubiatowicz, J. et al. *Oceanstore: An architecture for global-scale persistent storage*, Proceedings of the International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS), Nov. 2000.
- [7] Rhea, S. et al. *Pond: The oceanstore prototype*, Proceedings of the 2003 Conference on File and Storage Technologies (FAST), Mar. 2003.
- [8] Cuenca-Acuna, F. et al. *Autonomous Replication for High Availability in Unstructured P2P Systems*, The 22nd IEEE Symposium on Reliable Distributed Systems (SRDS-22), Oct. 2003.
- [9] ACID - Wikipedia, the free encyclopedia, <http://en.wikipedia.org/wiki/ACID>, 2005/08/26
- [10] [http://en.wikipedia.org/wiki/Distributed\\_hash\\_table](http://en.wikipedia.org/wiki/Distributed_hash_table), 2005/08/26