

# オンラインストレージを利用する JavaScript 用データベースの開発 大容量 Web アプリの開発コストを下げる

## 1. 背景

Web アプリケーションの表現力が広がり、世界中の人が Web でつながるようになった。また、レンタルサーバなどの普及により、Web 上でサービスを開始するにあたっての敷居が低くなった。この結果、Web は、技術者に多くのチャンスを与えるものとなった。特に、Web は、個人をはじめとする小グループのチームに対し、世界に向けたサービスを始めることを可能とした。

一方で、Web 上のサービスを継続的に運用するためには、増大するデータの管理とサーバの負荷に対処する必要がある。Web アプリケーションが大きなデータを扱うようになると、サービス提供者は、データのバックアップの手間や方法に悩まされることになる。また、サービスの利用者数が増えると、サーバの CPU 負荷が増える。通常、個人向けのレンタルサーバでは、CPU の性能は低く設定されていることが多く、サービス開始後に増強することは難しい。

一方、近年のディスクの容量増大・価格下落にともない、オンラインストレージの価格が下がっている。以下の表は、2001 年 4 月から 2007 年 4 月までの、AOL が提供する XDrive (<http://www.xdrive.com/>) というサービスの価格・容量の遷移である。

表 1 XDrive の価格・容量の遷移

	2001年4月	2003年10月	2005年2月	2007年4月
基本料金	4.95\$	4.95\$	9.95\$	9.95\$
基本容量	25MB	200MB	5GB	50GB
100MBあたり料金	198\$	24.75\$	1.99\$	0.2\$

## 2. 目的

本プロジェクトは、技術者が、個人で世界に向けたサービスを始めるにあたってのコストと敷居を下げることを目的とする。本プロジェクトでは、低価格化が進んだオンラインストレージをより容易に利用するためのデータベースシステムを開発する。さらに、データベースシステムに対し、クライアントサイドのプログラムから直接検索アクセスする仕組みを提供し、サービス提供者のサーバの CPU 負荷を下げる機構を実現する。

## 3. 開発の内容

### 3.1. システム構成

本プロジェクトでは、オンラインストレージを利用する JavaScript 用データベースシステム (StorJsDB : online STORAGE based JavaScript DataBase

system) の開発を行った。StorJsDB のシステム構成を図 1 に記す。

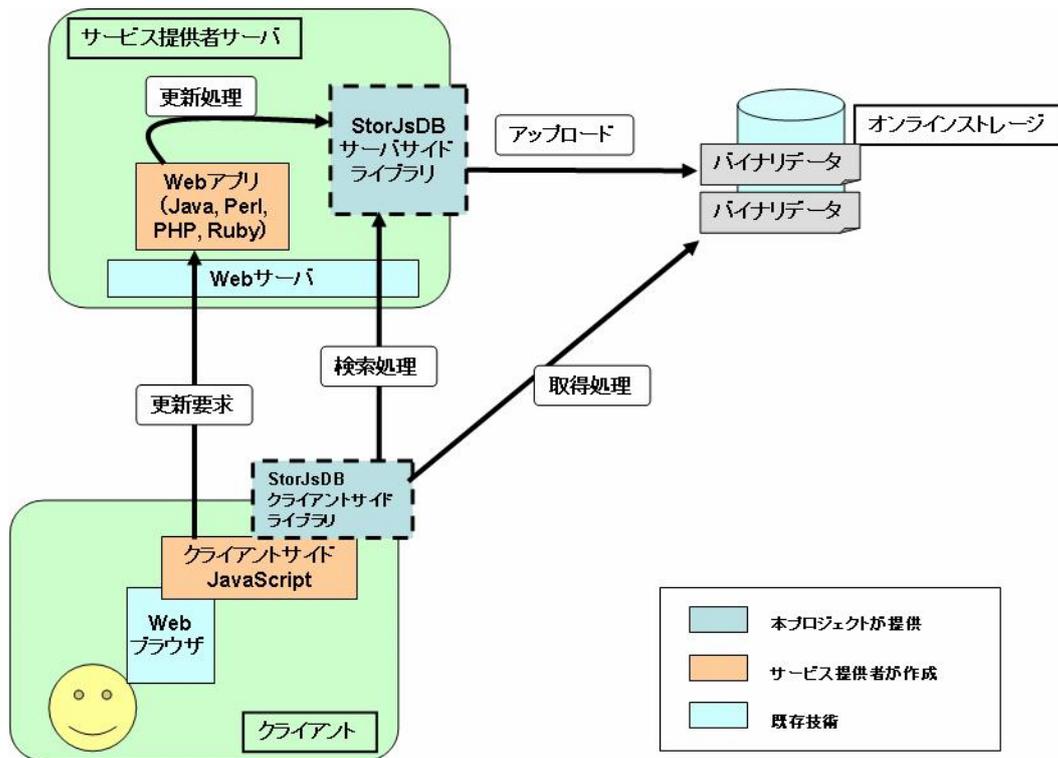


図 1 StorJsDB システム構成図

StorJsDB は、以下の二つのライブラリからなる。

- ・ StorJsDB サーバサイドライブラリ  
サービス提供者の Web アプリケーションに対し、オンラインストレージと連携したデータベース機能を提供する。
- ・ StorJsDB クライアントサイドライブラリ  
サービス利用者の Web ブラウザ上で動作する JavaScript プログラムに対し、データベース検索機能を提供する。

### 3.2. StorJsDB サーバサイドライブラリ

StorJsDB サーバサイドライブラリは、以下のデータベース機能を提供する。

- ・ データ検索
- ・ トランザクション開始、確定、キャンセル
- ・ データ追加、削除、更新

StorJsDB サーバサイドライブラリは、以下の (a) ~ (c) の三種類のライブラリを通して利用することができる。

(a) Java プログラム用の Java API ライブラリ

(例) `jsdb.insert("table1", object1);`

(b) JavaScript プログラム用の JavaScript API ライブラリ

(例) `jsdb.insert("table1", {key1:"value1", key2:"value2"});`

(c) その他言語で作成されたプログラム用の Web API ライブラリ

(例) `http://localhost/storjsdb/webapi?method=insert&name='table1'&object={'key1':'value1', 'key2':'value2'}`

StorJsDB サーバサイドライブラリは、Web アプリケーションからの更新要求を受けると、データベースのデータ、インデックスを作成する。データベースのデータ、インデックスは、JSONP 形式のファイルとして保持される(図 2)。

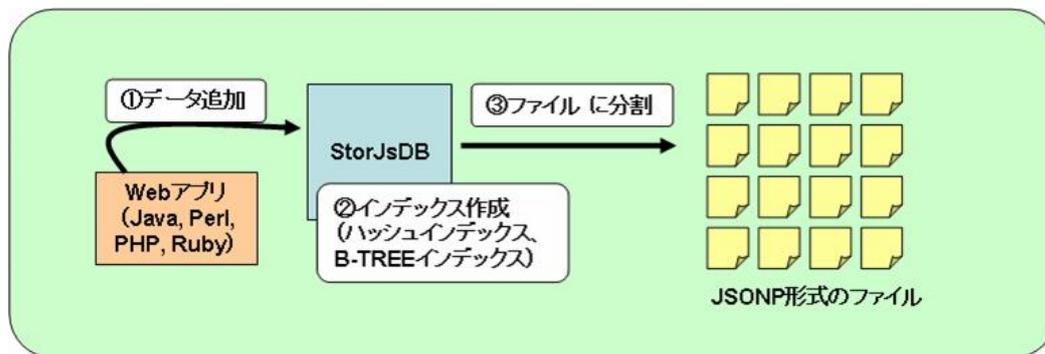


図 2 データ追加時の処理

また、StorJsDB サーバサイドライブラリは、データの追加時、以下の値をオンラインストレージにアップロードする。

- ・ バイト配列データをオンラインストレージにアップロードする
- ・ fp\_で始まるキーに対応する値を、ファイルのパスとして解釈し、ファイルの内容をオンラインストレージにアップロードする

データの値は、オンラインストレージ上のデータにアクセスするための URL に置き換えられた後、データベースに格納される(図 3)。

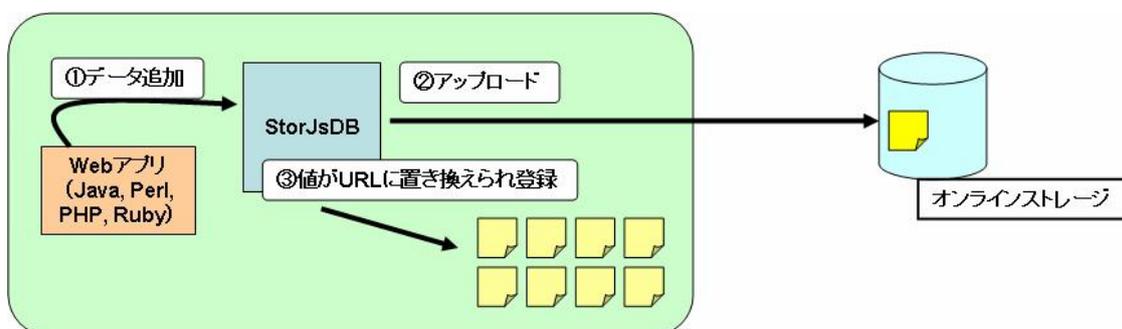


図 3 バイナリデータを含むデータ追加時の処理

### 3.3. StorJsDB クライアントサイドライブラリ

StorJsDB クライアントサイドライブラリは、以下のデータベース検索機能を提供する。

- ・ 指定されたキーワードで、ハッシュ検索を行う  
 (例) グループ"group1"に所属するユーザをすべて得る  
`jsdb.selectByHash("user", "group1");`
- ・ 検索条件をすべて満たすデータの集合を得る  
 (例) グループ"group1"に所属するユーザをすべて得る  
`jsdb.selectByKey("user", "group.name='group1'");`
- ・ ID の集合から、データの集合を得る  
 (例) ID が、id1, id2, id3 のユーザを得る  
`jsdb.selectByID("user", ["id1", "id2", "id3"]);`
- ・ 検索条件をすべて満たすデータの ID の集合を得る  
 (例) 名前が"yhayashi"のユーザの ID をすべて得る  
`jsdb.selectIdByKey("user", "name='yhayashi'");`
- ・ ID の集合から、関連する ID の集合を得る  
 (例) ユーザ ID uid1 に関連するグループの ID をすべて得る  
`jsdb.selectRelation1("rel_user_group", ["uid1"]);`

3.2 で述べたように、データベース内のデータは、JSONP 形式のファイルとして保持されている。StorJsDB クライアントサイドライブラリは、JSONP 形式のファイルのダウンロードと実行を繰り返し、データベースの検索を行う。以下の図は、B-TREE 形式のインデックスを辿り検索を行う動作の様子を表している。

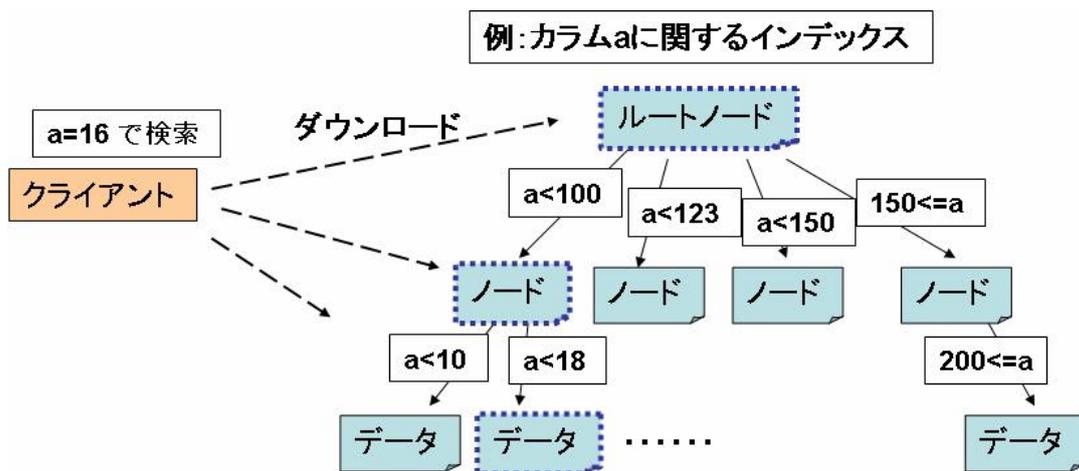


図 4 検索時の動作

#### 4. 従来の技術との相違

3.2 で述べたように、StorJsDB は、データベースのデータを JSONP 形式のファイルとして保持する。一方、従来手法で JavaScript プログラムからデータベース検索を行う場合は、Ajax によりサービス提供者のサーバと通信を行い、Web アプリケーション、データベースと連携して検索処理を行うこととなる。

以下の環境の下で、単純なハッシュ検索に関して、両者のリクエスト処理スループットの比較を行った。

種別：レンタルサーバ（専用サーバ）

CPU：Sempron 2600+

メモリ：512MB

ネットワーク：100Mbps 共用

結果は、以下の通りとなった。

本プロジェクトの手法：1000～リクエスト/秒

従来手法：80～120 リクエスト/秒

以上の結果から、JSONP ファイルへのアクセスによる検索を行うことによって、サーバの負荷を減らし、検索処理スループットを高めることができるという結論が得られる。

## 5. 期待される効果

本プロジェクトでは、オンラインストレージを利用する JavaScript 用データベースシステム StorJsDB の開発を行った。StorJsDB を利用することにより、大容量のデータを扱う Web アプリケーションを、より多くのユーザに対し、低コストで開発・運用することが可能となる。

従来では、個人をはじめとした小規模なグループでは、大容量のデータを扱う Web アプリケーションの開発・運用は困難であった。StorJsDB を利用することにより、この敷居が下がることが期待される。

## 6. 活用の見通し

本プロジェクト開発者が所属する会社で今後リリースする Web アプリケーション・Web サービスの中で、本プロジェクトの成果を積極的に利用していく予定である。

また、今後は、利用者がより簡単に利用できる仕組みを提供する。具体的には、データベース機能の Web サービス化を検討している。

## 7. 開発者名

林 良生（株式会社ナレッジフロー）

<http://knowledge-flow.net>