

次世代クラウドコンピューティング基盤 Raptor の実装と評価

羽鳥 孝広[†] 井原 雄太郎[†] 田胡 和哉^{††}

[†]東京工科大学バイオ・情報メディア研究科コンピュータサイエンス専攻 ^{††}東京工科大学コンピュータサイエンス学部

1 背景

現在クラウドコンピューティングの技術は急速に発展している [1]。クラウドコンピューティングを利用し、構築されたウェブサービスが次々と生まれている。ウェブのトレンドの変化は速度は非常に速く、今後ますますウェブサービスが増えていくこと予想される。そのため、ウェブサービスをより容易に素早く構築できるようにする必要がある。本稿ではクラウドコンピューティングを使用したウェブアプリケーションを容易かつ迅速に構築できるプラットフォームについて述べる。

2 次世代クラウドコンピューティング基盤

現在のウェブサービスでは文字だけでなく、写真や動画像、音楽など様々なコンテンツを扱わなければならない。これらのコンテンツ構成するデータは、ペイロードデータのほかに、動画像のタイトル、制作者、制作日時などプロパティ情報もある。現在多くのアプリケーションでは、コンテンツデータはファイルシステム、プロパティ情報を RDBMS と別々の手法で保存している。そのため、コンテンツを利用するウェブアプリケーションを構築する場合、それらを組み合わせなければならぬ。また、新たな種類のコンテンツが増えるごとに保存の仕方、利用の仕方が異なるためアプリケーションを変更する必要が出てきてしまう。

そこで、コンテンツをオブジェクト化し抽象化を行い管理を行う。コンテンツをオブジェクトとして扱うことによりコンテンツの種類による、保存や利用方法の差を吸収することが可能である。また、オブジェクトにはペイロードとプロパティ、さらには処理を含むことができるため、アクセス手順やロジックを隠すことができる。これにより、新たなコンテンツへ対応する場合リポジトリ部分を対応させることによりアプリケーションに一切手を加えることなく利用することが可能になる。このオブジェクトリポジトリを Raptor[2] と名付ける。

3 Raptor の内部構造

Raptor のアーキテクチャを図1に示す。Raptor ではコンテンツ一つ一つを WrapperObject という Java のオブジェクトとして保存する。すべての WrapperObject はツリー構造で管理されており、URI を用いてアクセスすることができる。

Java プログラムから Raptor を使用する場合、RMI を利用しアクセスを行う。WrapperObject の操作は WrapperObject からさらに通常のファイルに相当するオブジェクトを取得しおこなう。このファイルは CrudObject と呼ばれ、Create, Read, Update, Delete, Search の頭文字から名付けられている。また、CrudObject を通してプロパティの保存や取得することもできる。一つの WrapperObject がコンテンツから必要に応じて複数の CrudObject を生成する仕組みになっている。CrudObject にもすべて名前がついており、URI を用いたアクセスが可能である。

また、HTTP 経由でも利用することができる。Raptor と RMI 通信を行う Servlet を通すことにより可能である。この Servlet が HTTP のリクエストを受け付け、Raptor とやりとりを行い返答を行う。

オブジェクトを構成するデータはそれぞれ、ペイロードデータをファイルシステム、プロパティ情報を RDBMS に保存している。RDBMS、ファイルシステムにアクセスするクラスはプラグイン形式になっており、他のサービスに接続する実装をを作成することにより AmazonS3 や GoogleDrive, XML データベースなどへ接続が可能になる。これにより Raptor は様々なプラットフォーム上で運用することが可能になる。

4 Raptor によるアプリケーション構築

Raptor を使用することにより、ウェブアプリケーションを容易に構築することが可能になる。Raptor のリポジトリはツリー構造をしており、保存されているオブジェクトにすべて URI が付加されている。これにより HTTP との親和性が非常に高く、GET メソッドによるコンテンツの読み込みだけでなく、PUT メソッドによる保存や DELETE メソッドによる削除なども可能である。また、REST を使用することにより、プロパティを保存したり取得したりといったことも可能である。

WrapperObject は複数の CrudObject を保持することができる。たとえば、文書 Wrapper-

Implementation and evaluation of a new cloud computing platform Raptor

[†] Takahiro HATORI(taka@t-lab.cs.teu.ac.jp)

[†] Yutaro IHARA(ihara@t-lab.cs.teu.ac.jp)

^{††} Kazuya TAGO(ktago@cs.teu.ac.jp)

School of Computer Science, Tokyo University of Technology

(†) 1404-1 Katakura, Hachioji, Tokyo 192-0982, Japan

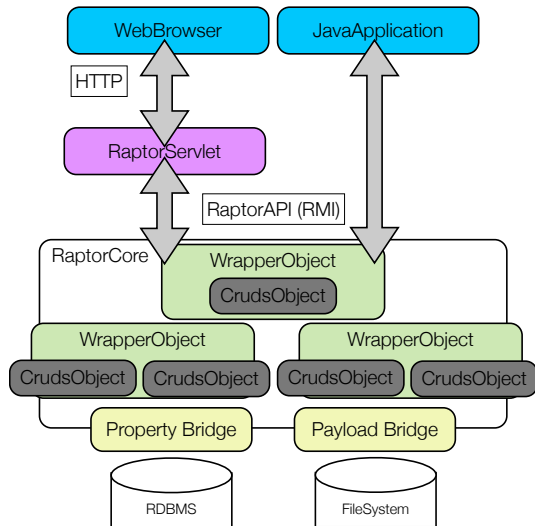


図 1: Raptor のアーキテクチャ

Object が “/sample.doc” に保存されている場合，“/sample.doc/thumb-1.jpg” にアクセスをすると文書の 1 ページ目のサムネイルが取得することができる。実際は “/sample.doc/page1.jpg” にコンテンツは存在するのでなく，“/sample.doc” の WrapperObject がサムネイル画像を含む CrudsObject を動的に生成し返している。この機能を用いることにより、コンテンツに対する処理も URI にマッピングすることが可能になる。よってアプリケーションを実装する場合あたかも通常のファイルシステムにアクセスしているかのように URI を指定し処理を行い結果を受け取ることが可能になる。

URI を用いた処理はすべての WrapperObject に共通な仕様なものもある。たとえば、保存されたコンテンツをウェブブラウザで表示したい場合、URI の後ろに “!app/show.html” をつけることによりブラウザで表示できる形で取得できる。また編集したい場合は “!app/edit.html” をつけることにより編集画面がブラウザ上に表示される。これにより、アプリケーションを構築する場合、コンテンツの種類を気にすることなく、表示や編集機能を付加することができる。

Raptor に保存されているオブジェクトを検索することも可能である。検索はオブジェクトに付加されたプロパティ情報を使用し行うことができる。プロパティはユーザが付加するだけでなく、オブジェクトが動的に付加することも可能である。たとえば、文書コンテンツから、キーワードを抜き出し、プロパティに付加することで自動的に検索の対象にすることも可能である。

Raptor へのアクセスは HTTP 経由だけでなく、Java, Javascript, PHP から利用することができる。

4.1 動画アプリケーションの例

動画を様々な端末やブラウザから再生出来るようにするには、それぞれの端末にあったフォーマットの動画をを用意する必要がある。そのため動画アプリケーションではあらかじめエンコードを行い用意しておく必要がある。Raptor では動画が保存されたとき、WrapperObject が自動的に TaskServer に動画の変換ジョブを渡す。TaskServer は ffmpeg を使用し、複数のフォーマットの動画へ変換を行い、WrapperObject 内に保存する。一つの WrapperObject は複数のコンテンツを複数の CrudsObject として保存することができるため、複数の動画フォーマットも一つの WrapperObject として管理できる。さらに、アクセスごとに振る舞いを変えることができるので、アクセスした端末やブラウザごとに適切なプレイヤーや動画ファイルの CrudsObject を WrapperObject が選択して返すことが可能になる。アクセスする URI は同じなので、どの環境でもアプリケーションの作成者は意識することなく URI を指定して再生させることが可能になる。

5 実装

Raptor の開発言語は Java6 である。OS は CentOS 5.4, Windows 7 Enterprise 64bit である。IDE は NetBeans6.9.1 を使用した。プロジェクト管理には Apache Maven2, ソースコードのバージョン管理には Git を使用した。

ソースコードの規模はステップ数約 30000, ファイル数約 300 となった。ベータ版の実装を 2008 年 6 月から、Version1 の実装を 2009 年 4 月から、そして現在の Version2 は 2010 年 4 月から実装を行い、現在は運用できる形になった [3]。

6 今後の予定

Raptor は今年度中に東京工科大学で全学生に向け大規模に運用される予定である。Raptor を利用する 1 つの方法として、HTML5 を用いたガジェットベースの GUI と組み合わせて統合環境化を行う。これにより Raptor 上のコンテンツをガジェットとして扱えるようにする。これらを利用し講義情報をデータベース化し授業に使用する。

参考文献

- [1] ニコラス・G・カー (2008) 『クラウド化する社会 ビジネスモデル構築の大逆転』 翔泳社
- [2] Raptor Project <http://raptor.oss.teu.ac.jp>
- [3] 授業クラウドプロジェクト <http://www.oss.teu.ac.jp/project/lcloud/>