

# オフラインで実行可能な Ruby on Rails アプリケーションの開発に向けて

須藤 和輝<sup>1</sup> 新城 靖<sup>1</sup>

## 1. はじめに

今日では、オフラインでも利用可能な Web アプリケーションは少なくはない。そのようなアプリケーションの開発を支援するために、HTML5 ではキャッシュマニフェストという機能が定義されている [1]。この機能を利用すると、ユーザがインターネットに接続して Web アプリケーションを利用した際に、オフライン動作に必要なファイルをユーザのローカル環境にダウンロードする。そして、オフラインになった時にはそれを利用して動作する。

オフライン用の Web アプリケーションの作成は容易ではない。例えば、元からあるオンライン用のものに加えて、しばしばオフライン用に新たにデザインされたものを作成しなければならない。

本研究では、オンライン用の Web アプリケーションを元にしてオフラインで実行可能なものを効率的に開発することを目的とする。具体的には、Ruby on Rails に基づき作成された Web アプリケーションをオフラインで実行可能にすることを目指す。この目的を達成するために、本研究では、関係データベースの表を個人化し、オフラインで利用する PC に複製する。オフラインの時に、Web アプリケーションは、PC 上の複製を利用して動作する。オンラインになった時、PC 上で更新されたデータの一部をサーバに書きもどす。本研究では、表の更新方法や同期方法に強い制約を設けることで競合を回避する。

## 2. 提案手法

本研究では図 1 に示すように、Ruby on Rails で記述された Web アプリケーションをオフラインで実行可能にする。この図でサーバには Web アプリケーションを利用している複数のユーザのデータを保存したデータベースがある。そのうち、データベースの同期ツールを使って特定の個人のデータを抽出し、PC に複製する。オフラインの時、PC 上では、サーバと同じ Web アプリケーションのコードを、PC に複製したデータベースを使って実行する。この時、PC に複製されたデータベースの一部は更新される。

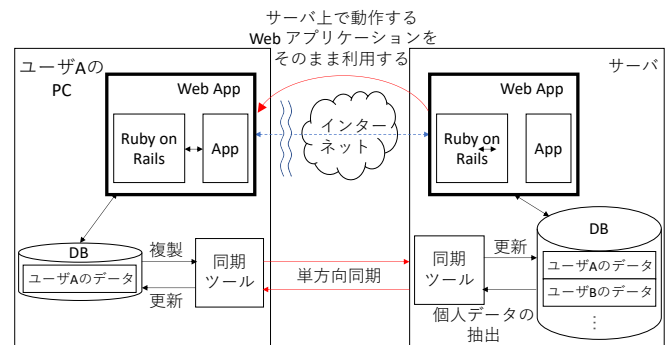


図 1 データベースの単方向同期による Web アプリケーションのオフライン実行

PC が再びオンラインになった時、データベースの同期ツールを使ってサーバにあるデータベースの内容を更新する。

本研究では、データベースの同期ツールとして SymmetricDS [2] を用いることを検討している。SymmetricDS は様々な種類のデータベース管理システムに対応した同期ツールである。SymmetricDS ではテーブルごとに単方向同期・双方向同期を指定することができ、同期時に同期内容をフィルタリングすることも可能である。

任意のデータベースの表を、サーバと PC で双方向で同期することは、競合が発生するので不可能である。本研究では、関係データベースの表を、次の 2 種類のタイプに限定することで対応する。

(1) **読み出し専用・個人化複製:** サーバでのみ更新し PC 側では更新しない。オンラインの時に、特定の個人に関係したデータを抽出し、サーバから PC で単方向で同期する。

(2) **個人状態複製:** ユーザの操作によってのみ更新される。オンラインの時には、サーバと PC で完全に同期される。オフラインの時には、PC 上のもだけが更新され、サーバ上のは更新されない。オフラインからオンラインになった時に、PC からサーバへ単方向で同期される。

本研究では、アプリケーションにどのような性質があれば、この手法で対応可能になるかを明らかにする。また、Ruby on Rails では、設定より規約 (convention over configuration) という考え方があり、本研究でも、この考

<sup>1</sup> 筑波大学  
University of Tsukuba

表 1 記事配信アプリケーションで利用する表.

表名	概要
user	ユーザ情報を扱う (ユーザ名, メールアドレス, パスワードなど)
feed	記事配信の情報を扱う (サイト名, リンクなど)
entry	記事 1 つ 1 つの情報を扱う (記事名, 本文など)
history	閲覧履歴

```
1 insert into SYM_ROUTER (router_id,  
2   source_node_group_id, target_node_group_id,  
3   router_type, router_expression) values  
4   ('server-2-user-id', 'server', 'user',  
5   'column', 'USER_ID=:EXTERNAL_ID');
```

図 2 SymmetricDS の同期設定例.

え方に従い, 関係データベースの表の名前を元にしてデータベース同期ツールの設定を自動生成したいと考えている.

### 3. 記事配信アプリケーション

本研究では Ruby on Rails アプリケーションの例として簡単な記事配信アプリケーションを実装し, 提案手法に基づきオフライン実行可能なアプリケーションが開発可能であるか調査した. このアプリケーションでは, 表 1 に示すような表を用いる. user, feed, および entry は, ユーザの登録や記事の追加などサーバで更新しなければならないので 2 章で述べたタイプ (1) に分類される. オフラインの間は, PC 上のこれらの表が変更できない. 例えば, パスワードの変更や購読するサイトを追加することはできない. history は, 閲覧履歴といった PC 側で変更されるもののため, タイプ (2) に分類される. これにより, ユーザには PC に複製された記事を読覧可能にし, また, 閲覧履歴もきちんと保存しなければならない.

次に, これらの表についてデータベース同期ツールの設定を行う. SymmetricDS では, データベースの同期方法を設定用の表に登録する. 図 2 に表 feed に関する同期方法を示す. この例では, 監視対象の表に変更があった場合のデータ反映の範囲と方向を定義している. feed では各ユーザの PC にユーザ自身のデータのみを反映させたいため, feed の user\_id カラムの値に応じてフィルタリングをする. そのために, router\_type を 'column', router\_expression を 'USER\_ID=:EXTERNAL\_ID' とすることで, ノードを識別する ID である External ID が user\_id と一致したノードにのみデータを反映するように設定している. その他に, SymmetricDS では, 同期のタイミング等の設定を行う.

現在までに, タイプ (1) の表について, SymmetricDS の同期機能を用いて, サーバから PC に個人化されたデータを複製し, この簡単な Ruby on Rails のアプリケーションはオフラインで動作することを確認した. 現在, タイプ (2) の表の同期の部分を実装している.

### 4. 関連研究

Google Gears は, Web ブラウザを拡張することで, リッチな Web アプリケーションを開発可能にするための仕組みである [4]. Google Gears は, Web ブラウザでサーバと通信することなく高度なアプリケーションを動作させるために, 関係データベース (SQLite) を利用する仕組みを提供していた. Google Gears の機能の一部は, HTML5 標準に取り込まれ開発を終了した. Gears on Rails[3] は, Google Gears を拡張し, Ruby on Rails のアプリケーションを Web ブラウザ内で実行することを目指していた.

本研究の目的は, オフライン対応の Web アプリケーションの効率的な開発であり, これは Gears on Rails の目標と一致している. 本研究では, Gears on Rails とは異なり, ブラウザでアプリケーションを実行するのではなく, サーバで動作する Web アプリケーションをそのまま PC で実行することを目指す.

### 5. おわりに

本研究の目的はオフラインで実行可能な Web アプリケーションの効率的な開発を実現することである. そのために, サーバで動作している Ruby on Rails の Web アプリケーションを PC でそのまま実行することを目指す. この時, 関係データベースの表の使い方に強い制約を設け, サーバと PC の間で単方向で同期する. 現在, データベースの同期ツールとして SymmetricDS を使い, サーバ上のデータを個人化し, サーバから PC に複製することで, PC 上でオンライン用の簡単な Ruby on Rails のアプリケーションが動作することを確認した. 今後は, オフライン時に PC で更新された表をサーバに同期する部分を実装する. また, データベースの表の名前から SymmetricDS の設定を自動生成する.

### 参考文献

- [1] The Web Hypertext Application Technology Working Group (WHATWG): HTML Living Standard, <https://html.spec.whatwg.org/>. Accessed: 2021-11-10.
- [2] SymmetricDS, <https://www.symmetricds.org/>. Accessed: 2021-11-10.
- [3] Gears on Rails, <https://code.google.com/archive/p/gearsonrails/>, 2016. Accessed: 2021-11-10.
- [4] Google Gears, <http://gearsblog.blogspot.com/>, 2011-11-3. Accessed: 2021-11-10.