

# クラウドコンピューティングにおけるデータアクセス方法とその評価

志村 惇<sup>†</sup> 中道 上<sup>‡</sup> 青山 幹雄<sup>‡</sup>

南山大学大学院 数理情報研究科<sup>†</sup> 南山大学 情報理工学部 ソフトウェア工学科<sup>‡</sup>

## 1. はじめに

近年、クラウドコンピューティング(以下、クラウド)が新しい情報システムの基盤として注目されている。オンプレミスとクラウド間で連携が求められるが、それぞれのデータストアはデータ構造とインタフェースが異なり統一的に扱うことが困難である[3].

## 2. 研究課題

従来の RDB とクラウドで利用される KVS(Key Value Store)など NoSQL では扱うデータ構造が異なる, RDB ではテーブル全体でスキーマが定義された構造型であるのに対し, KVS では Key 項目に対して Value 項目のみ持つ分散ハッシュに基づく単純なデータ構造である, また Value 項目で扱われるデータはレコード毎に異なる. データ構造の違いから, 一つのアプリケーションで両データベースを利用する場合, それぞれのデータベースのデータ構造に適したデータを容易に利用できる必要がある.

## 3. アプローチ

### 3.1. データアクセス方法の分類

SQL と NoSQL の差異に着目し表 1 に示す 3 つのデータアクセス方法を対象として, データストアへのアクセス方法を分類し, 比較する[1]. それぞれの振る舞いを図 1 に示す.

表 1 データアクセス機能

モデル	SQL	NoSQL	NoSQL
実装	MySQL	HandlerSocket (MySQL)	Cassandra
Connection	getConnection (URL,user,pw)	openIndexSession (URL,port)	Open (URL,port)
Read	Select	Find	Get
Insert	Insert	Insert	Set
Update	Update	Update	Set
Delete	Delete	Delete	Del

### (1) MySQL

RDB は通常, SQL を利用してデータの操作を行う, そのためにサーバ側ではデータを整形し SQL のクエリを生成する必要がある.

### (2) HandlerSocket

MySQL に対して NoSQL と同様のアクセスを提供

Unification Data Model and Architecture on Cloud Computing.  
<sup>†</sup>Atsushi Shimura, Graduate School of Mathematical Sciences and Information Engineering, Nanzan University.

<sup>‡</sup>Noboru Nakamichi, Mikio Aoyama, Department of Software Engineering, Nanzan University.

する. 扱いたいデータを API に引き数として渡すことで MySQL にアクセスできる.

### (3) Cassandra

NoSQL の例として Cassandra がある. Cassandra では SQL をサポートしておらず, データを操作するには特定の API に対して引き数として Key のデータを渡し Value データを操作する[2].

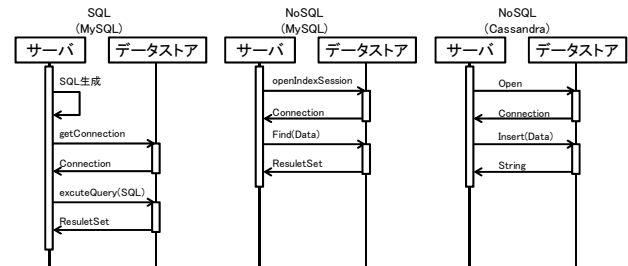


図 1 データアクセスの振る舞い

### 3.2. データストアの隠蔽

データアクセスの実装を隠蔽することに着目し, クライアントからデータをアクセスするインタフェースを一元化し, クライアント側はデータストアへのアクセスロジックを保持せずデータストアが提供するサービスを呼び出す. さらに, データアクセスの機能を Web サービスとして公開し, クライアントは公開されたサービスを利用することでデータアクセスを可能とする.

## 4. データアクセス方法

### 4.1. データアクセスサービスの構成

SQL と NoSQL に対する, データアクセスを Web サービスとして提供するシステムの構成を図 2 に示す.

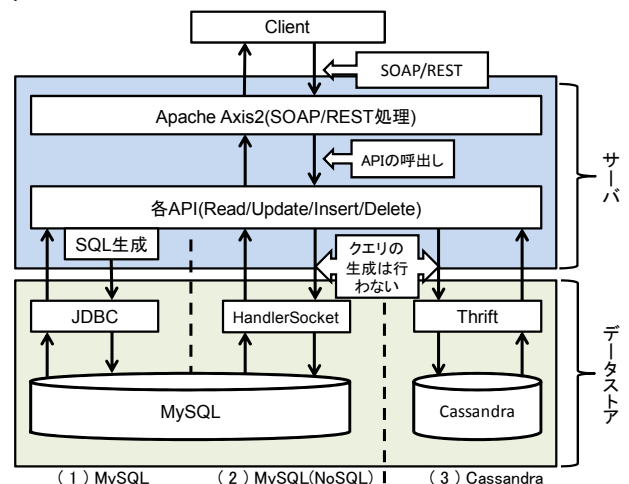


図 2 システムの構成

#### 4.2. クライアントと Web サーバ間

クライアントと Web サーバ間はデータアクセス方法によらない統一的な手法をとる。Web サービス標の準メッセージングプロトコルである SOAP/REST を用いる。クライアントのメッセージは ApacheAxis2 により処理される。また SOAP/REST は実装に依存しないため柔軟なアプリケーションの配置が行える。

#### 4.3. Web サーバとデータストア間

##### (1) SQL(MySQL)

SQL を用いた MySQL へのアクセスはサーバ側がクライアントからのメッセージに基づいて SQL を生成し JDBC を介して MySQL へのアクセスを行なう。データの操作は生成された SQL のシンタックスに依存する。

##### (2) NoSQL(MySQL)

MySQL に対して NoSQL でデータアクセスを行なうには HandlerSocket を用いる。クライアントのメッセージに基づいて API を呼び出し CRUD の実行を行う。サーバ側で明示的にクエリの生成は行わず表 1 に示したデータアクセス機能を利用してデータの操作を行なう。

##### (3) NoSQL (Cassandra)

Cassandra に対するデータアクセスはクライアントのメッセージに基づいて呼び出された API に対して引き数としてクライアントからのデータを渡す。Cassandra は SQL をサポートしないのでサーバは Thrift インタフェースを介して CRUD の実行を行なう。

### 5. プロトタイプ

図 2 に示したデータアクセス方法の構成をプロトタイプとして実装を行い(図 3), 3 つのデータストアに対するアクセスの性能を測定した。

#### 5.1. 実装環境

HandlerSocket のインストールには Linux 環境が必須のため Ubuntu-server を OS として環境の実装を行った。アプリケーションの実装は MySQL と Cassandra のインタフェースとして一般的な JDBC および Thrift と親和性が高く、実装コストが低い Java を利用する。

モデル	SQL	NoSQL	NoSQL	
クライアント	OS	Ubuntu-server(Linux)		
	App	Java		
サーバ	OS	Ubuntu-server(Linux)		
	アクセス	JDBC	HandlerSocket	Thrift
	DB	MySQL		Cassandra

図 3 プロトタイプ実装環境

#### 5.2. 実装方法

Java で構成されたクライアントプログラムから各データストアに対して表 2 に示すデータの登録を 100 回行い、登録に要した時間を測定した。100 件の平均と標準偏差を表 3 に示す。平均の算出では、

MySQL の場合 10 ミリ秒を超える異常なデータ 29 件を除外した。

表 2 登録データ

変数	id	name
データ型	String	String
データサイズ	1byte	15byte

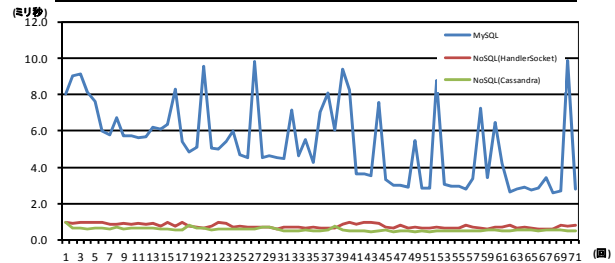


図 4 データ登録実行時間(1)

表 3 データ登録実行時間(2)

モデル	SQL	NoSQL	NoSQL
DB	MySQL	MySQL	Cassandra
平均	5.3	0.8	0.6
標準偏差	2.1	0.1	0.1

単位:ミリ秒

### 6. 評価

#### (1) データアクセス方法の差異

MySQL に対し HandlerSocket を利用してアクセスする場合の実装は SQL の生成を必要とせず、API を呼び出し引き数にデータ渡すというものであるため、Cassandra に対してデータアクセスする場合と大きな差異は無く、RDB と KVS を統一的に扱うために有効であると思われる。

#### (2) データアクセス性能

HandlerSocket を利用して MySQL にアクセスする手法は表 3 に示すように、Cassandra にアクセスする場合と同等の性能が得られた。しかし HandlerSocket を利用する場合、RDB の特徴である SQL を利用した構造的なデータ操作ができず、トランザクション処理もサポートされていない点が課題となる。

### 7. まとめ

本稿ではクラウドコンピューティング環境化におけるデータアクセスの方法と、その性能に関するデータの提示を行った。クラウドコンピューティングでは RDB と KVS の機能を統一的に扱うことが求められるが RDB に対して NoSQL でアクセスすることができれば RDB と KVS へのアクセス手法の差を埋めることができると考えられる。

### 8. 参考文献

[1] 佐々木 達也, NoSQL データベースファーストガイド, 秀和システム, 2011.  
 [2] Apache Software Foundation, Cassandra, <http://cassandra.apache.org/>  
 [3] 清野 克行, クラウド・アーキテクチャの設計と解析, 秀和システム, 2010.