

社長が脱RDBと言い出して困りましたが、 開き直って楽しんでいる話

井上 誠一郎

アリエルネットワークCTO

ワークスアプリケーションズ エグゼクティブフェロー

自己紹介

井上誠一郎

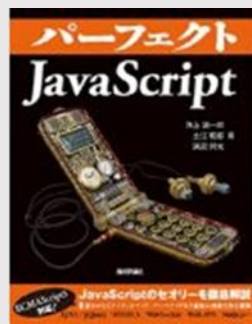
書籍 「パーフェクトJava 第1版/第2版」

「パーフェクトJavaScript」

「実践JS サーバサイドJavaScript入門」

「P2P教科書」

「15時間でわかるJava集中講座」(監修)



自己紹介(続)

1990年代後半 Lotus Notes開発

2000年代前半 P2Pアプリ開発

2000年代後半 Webアプリ開発(RDB脳化が進行)

2010年代前半 脱RDB(今ここ)

今日のトピック

RDBを使わずに
業務アプリを作れそうに思えてきた経験談

RDBなしのほうが良い

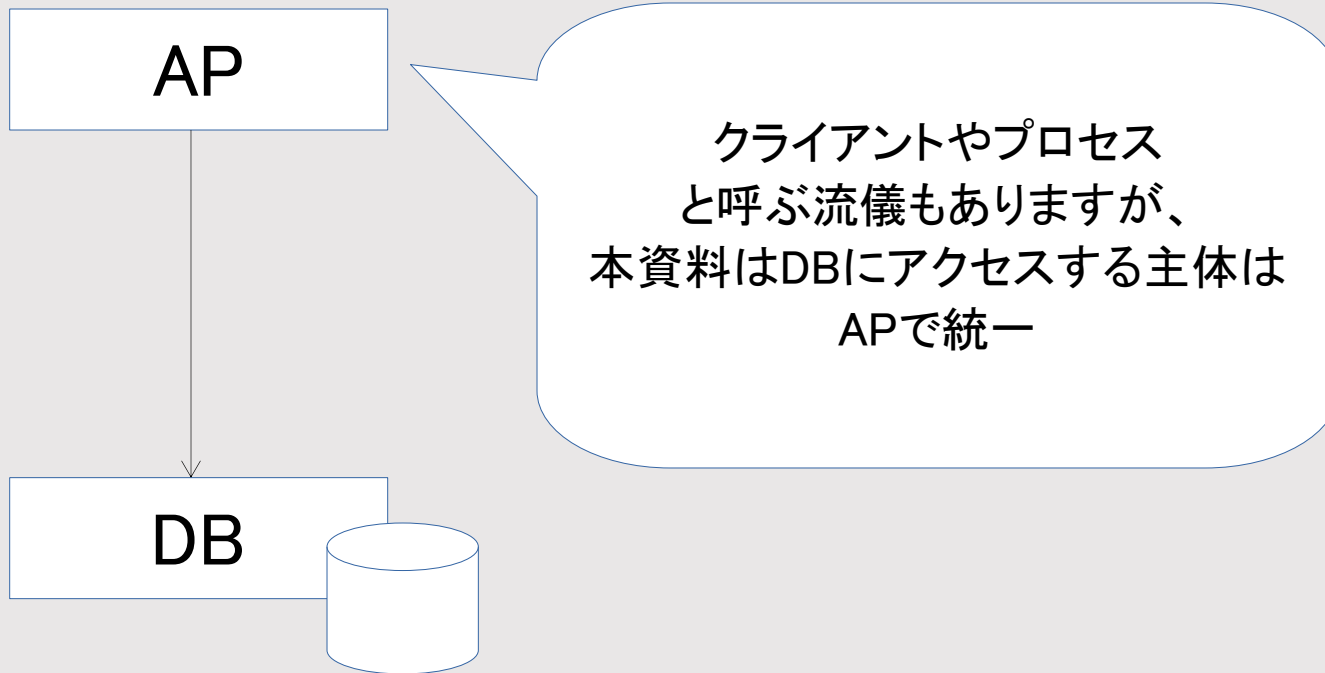
...とまでは言いません。念のため

今日話さないトピック

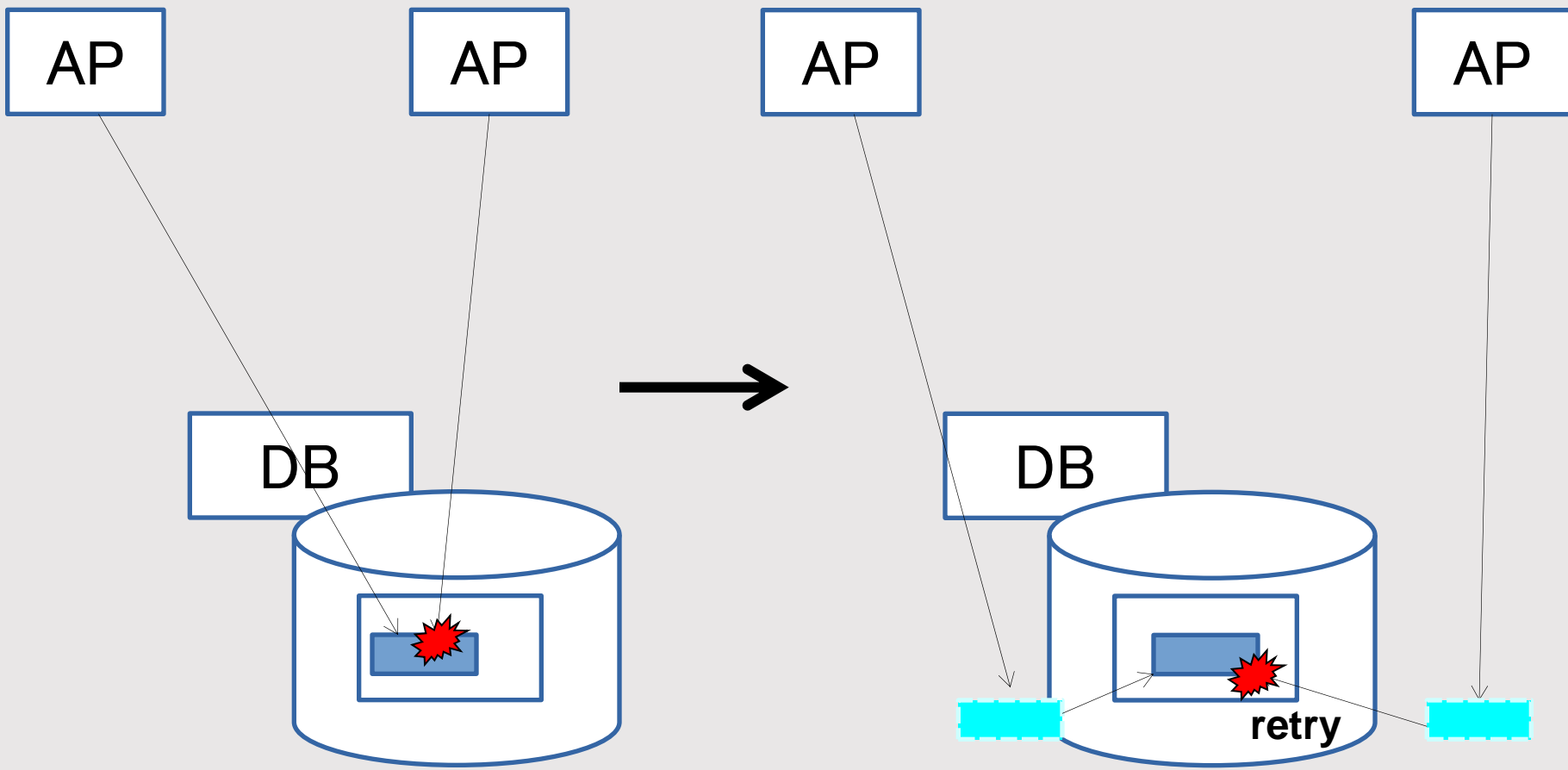
NoSQL実装の個々の比較

CAP定理とかその周辺の話

本プレゼンで使う用語について



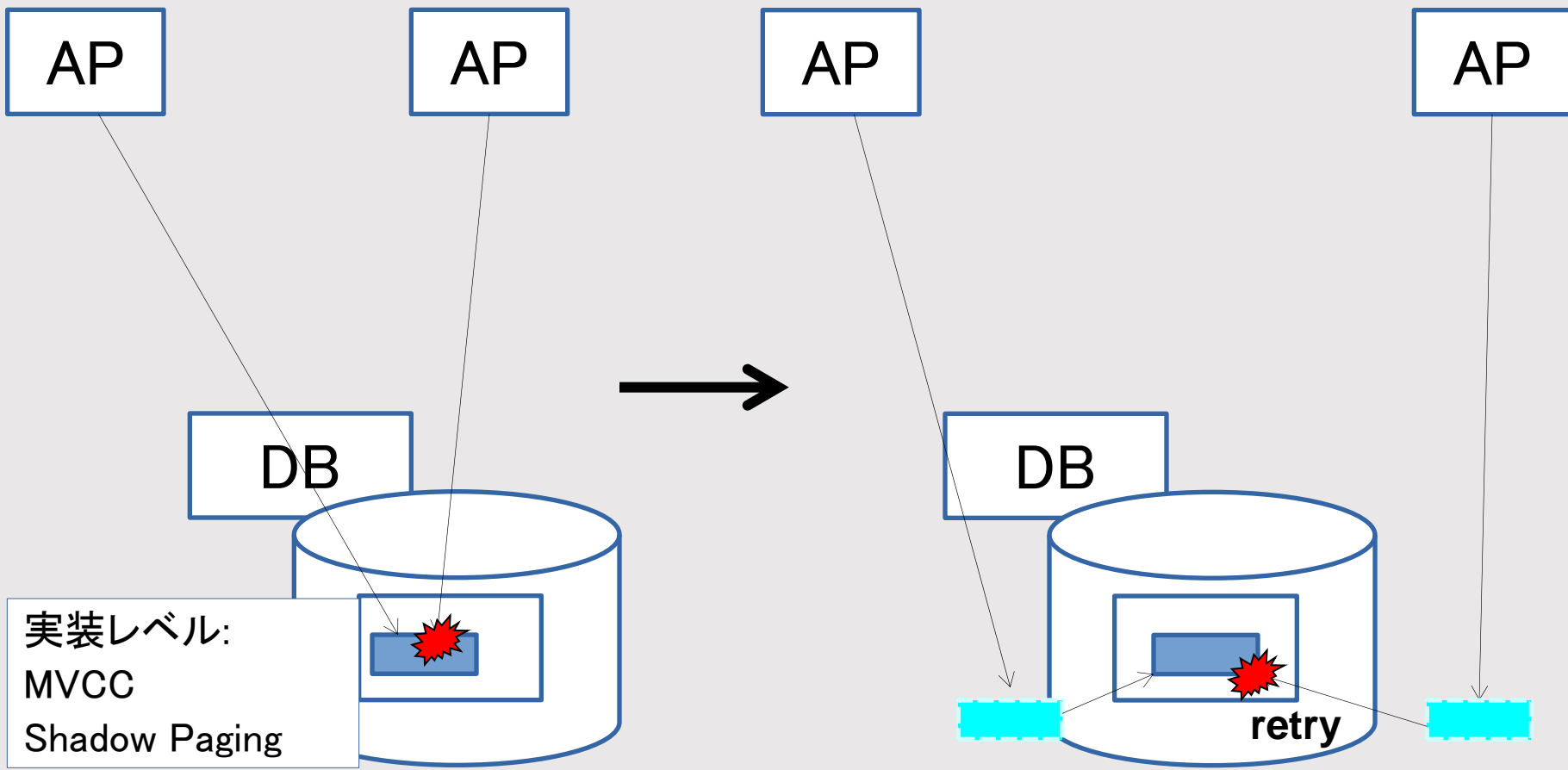
とりあえず結論



ロックと直列化

楽観的CC
(イミュータブルデータとリトライ)

とりあえず結論



実装レベル:
MVCC
Shadow Paging

ロックと直列化

楽観的CC
(イミュータブルデータとリトライ)

目次

1. ちょっとした前置き
2. 何を目指しているのか
3. アーキテクチャ決定の道のり
4. RDBなしの不安
5. まとめ

1. ちょっとした前置き

約 313,000 件 (0.49 秒)

ワークスアプリケーションズ、クラウドネイティブの次世代ERP ...

biz.bcnranking.jp/article/news/1410/141007_137653.html ▾

2014/10/07 - ワークスアプリケーションズ(牧野正幸代表取締役CEO)は、10月7日、ERP(統合基幹業務システム)の新製品「HUE」の製品化に成功したと発表した。「脱RDB(リレーショナルデータベース)」「クラウドネイティブ」の二点が技術的なポイント ...

ワークスアプリケーションズ、SaaS型ERP「HUE」を発表:日経 ...

itpro.nikkeibp.co.jp/atcl/ncd/14/457163/100800179/ ▾

2014/10/07 - 牧野CEOは、「RDBは開発者にとって使いやすいツールだが、グーグルなどの高速な検索システムでは分散型データベースが使われている。我々も脱RDBによって高速な処理を実現した」と話す。牧野CEOによると、RDBを使わないERP製品 ...

まるで、有能な秘書。ワークスアプリケーションズ - PR TIMES

prtimes.jp/main/html/rd/p/000000001.000011485.html ▾

2014/10/07 - ①脱RDBで分散技術の活用ワークスは、RDBから脱却し、分散型データベース・分散処理によって高速化と利便性の向上を実現します。②「オンプレミス」ではなく「クラウドネイティブ」クラウドがもつ柔軟性を最大限活用し、人と人、人と情報 ...

最初の自分の反応

「RDBが遅い」は誤解

仮にNoSQLが速くても、
必要機能を実装したら同程度の速度になるはず。

理論上同程度であれば、
実装の差で、枯れたRDBMSが有利になるはず。

一方で

教科書的なリレーショナルモデルだけで、
元々業務アプリを作っていないのは事実

各種キャッシュ(キャッシュサーバ、ORM層のキャッシュ)

非正規化テーブル(結合済みテーブル、サマリーテーブル)

アプリ側のバリデーション処理(RDBの制約だけでは不十分)

アプリ側のトランザクション管理(楽観的並行制御)

今の自分のスタンス

「脱RDB、これはこれで面白い」

脱RDBしても業務アプリを作れそうな気がしてきた

ただ、脱RDBで圧倒的に速くするには並列処理に適したデータモデルや処理に変えていく必要がある

2. なにを目指しているのか

**そもそも我々は
なにを目指しているのか**

応答速度100ミリ秒の業務アプリ

**応答性能100ミリ秒の業務アプリが
提供できる価値**

トライアンドエラーを待たずにできる

Enterprise Search

カテゴリから探す

- 絞り込み
- 職種
 - 財務職 (1)
 - 総務職 (1)
 - マーケティング職 (5)
 - 商品企画職 (4)
 - 営業企画職 (3)
 - 広告宣伝職 (1)
 - 営業職 (26)
- 役職レベル
 - 主任級 (1)
 - 役職なし (4)
- 事業所
 - 東京本社 (2)
 - 名古屋事業所 (1)
 - メキシコ支店 (1)
 - Mexico Central Oficina (1)
- 社員区分
- 勤務エリア
- 会社
- 在职状態
- 勤続年数

該当者: 5名

ごちらのキーワード
元のキーワード
「ブラジル」に

ブラジル 留学
ブラジル 経歴
ブラジル 大学
ブラジル 国籍



岸田 喜晴
WAP52916 1292
マーケティング職
東京本社

ワークスエンジニアリング株式会社
営業本部 マーケティング部
ディスベンサーマーケティンググループ 第三係

学歴 - [Uオデッサネイロ連邦大学](#) 法学専攻 2010年1月卒
雇用 2010/04/01 - 営業本部, マーケティング部, ディスベンサーマーケティンググループ, 第三係, -, 東京本社, マー...



にしむら さや
西村 紗弥
WAP0800 1003
主任 (主任級)
マーケティング職
東京本社

ワークスフーズR&D株式会社
営業本部

自己申告 2014/04/01 - ...大学時代に、[ブラジル](#)の高度成長... (1年以内に異動希望) (3年以内に異動希望) (5...
昇格 1996/10/01 - 営業本部, 主任, 東京本社, マーケティング職
雇用 1988/04/01 - 人事部, 人事課, 給与計算チーム, -, 虎ノ門事業所, マーケティング職



いしばし たけし
石橋 剛
WAP1640 1071
マーケティング職
名古屋事業所

ワークスフーズR&D株式会社
カスタマーコンサルティング事業部 中部グル...

自己申告 2014/04/01 - [ブラジル](#) サノバワロ支店への異動を希望します。 (1年以内異動希望)
組織改編 2014/02/01 - カスタマーコンサルティング事業部, 中部グループ, 一課, -, 名古屋事業所, マーケティング職
定期異動 2012/04/01 - カスタマーコンサルティング事業部, 中部グループ, 一課, -, 名古屋事業所, 研究開発職

表示基準日: 2015/04/07 全員をリストに追加


Enterprise Search


カテゴリから探す すべて 海外赴任希望 マーケティング職


絞り込み 該当者: 314名 IF キーワード関連順 表示基準日: 2015/04/08 全員をリストに追加

- 役職レベル
 - 副本部長級 (1)
 - 部長級 (5)
 - 副部長級 (12)
- 事業所
 - 東京本社 (112)
 - 北京事業所 (12)
 - 青山事業所 (12)
- 社員区分
- 勤務エリア
- 会社
- 在職状態
- 勤続年数

「海外赴任希望 マーケティング職」に関連するキーワードで検索できます
海外赴任希望 マーケティング職 アメリカ 海外赴任希望 営業企画職 海外赴任希望 商品企画職 海外赴任希望 情報システム職

 ほそだ さやか ワークスフーズ株式会社
細田 清佳
WAP0231 1003 営業推進本部
部長 (部長級) リサーチマーケティンググループ
マーケティング職 東京本社
自己申告 2014/04/01 - 海外赴任を希望する: 前職でシンガポールでの最終消費財のマーケティングを2年間経験し...
昇格 2009/04/01 - 営業推進本部, リサーチマーケティンググループ, 部長, 東京本社, マーケティング職
昇格 2004/10/01 - 営業本部, 副部長, 東京本社, マーケティング職

 まつの たけひさ ワークスヘルスケア株式会社
松野 武久
WAP0267 1003 営業企画本部
部長 (部長級) 通販マーケティング部
マーケティング職 東京本社
自己申告 2014/04/01 - 海外赴任を希望する: 現在マーケティング部門の中で各国の調査のとりまとめを行っている...
昇格 2006/04/01 - 営業企画本部, 通販マーケティング部, 部長, 東京本社, マーケティング職
昇格 2001/10/01 - 営業本部, 副部長, 東京本社, マーケティング職

 なるせりか ワークストリニコ株式会社
成瀬 里歌
WAP25352 4654 企画本部 企画部
企画課 第四係
マーケティング職 東京本社
自己申告 2014/04/01 - 海外赴任を希望する: 前職で海外、特に新興国の市場規模の調査を行っていた経験を活か...
異動 2013/04/01 - 企画本部, 企画部, 企画課, 第四係, -, 東京本社, マーケティング職
異動 2011/04/01 - 天然水事業部, 企画グループ, マーケティング部, マーケティング課, 第三係, -, 東京本社, マーケテ...

 あいはら あきのり 日本レジャンドサイダー株式会社

Insta Report

インスタレポート - 社員プロフィール 詳細帳票

プレビュー 印刷

出力形式 詳細

テンプレートを編集する 🔍 🔍 📄 ☰

テンプレート

おすすめ 評価 新着 🔍

- 社員プロフィール (Basic)**
ランキングNo1 レビュー数No1
★★★★☆ 4.5点 (84件のレビュー)
- 社員プロフィール (Stylish)**
ランキングNo2
★★★★☆ 4.4点 (51件のレビュー)
- 社員プロフィール (Soft)**
ランキングNo3
★★★★☆ 4.3点 (42件のレビュー)
- 社員プロフィール (Simple)**
社内利用No.1
★★★★☆ 4.1点 (34件のレビュー)
- 社員プロフィール (職務履歴)

RESUME

サトウ ミカ
佐藤 美香
Sato Mika



Profile

社員番号
WAP0419

会社
ワークスヘルスケア株式会社

事業所
東京本社

所属
営業企画本部 事業開発部 事業開発課
滞留期間 2ヶ月

役職
係長
滞留期間 1年

職種
経営企画職
滞留期間 2ヶ月

TEL
090-6056-2319

内線
1056

E-mail

異動歴

年月	所属	役職	備考
2014/02	営業企画本部 事業開発部 事業開発課	経営企画職 係長	組織改編
2013/04	営業企画本部 商品企画部 販売促進課 第一係	営業企画職 係長	昇格
2012/04	営業企画本部 商品企画部 販売促進課 第一係	営業企画職 主任	定期異動
2011/04	営業企画本部 商品企画部 マーケティング課	マーケティング職 主任	昇格
2010/04	営業企画本部 商品企画部 マーケティング課	マーケティング職 -	定期異動
2008/04	営業企画本部 通販マーケティング部 通販マーケティング課	マーケティング職 -	定期異動
2006/04	営業企画本部 営業部 一課 第三係	営業職 -	新卒入社

滞留期間グラフ



所属

- 販売物流班
- 戦略精算係
- 経営管理課
- 企画開発課
- 東京分室事業開発課

Enterprise Spreadsheet

新しいリスト 戻る 帳票出力 招待する 送る 保存

表示基準日: 2015/04/08

	社員番号	写真	氏名	入社区分	勤続年数	考課 - 5年平均	直近の評価ランク	婚姻区分	職種	役職レベル	会社	事業所	所属 - 略称	勤務地 - 国
1	WAP0231		ほそだ さやか 細田 清佳	中途	27年	77.8	A	未婚	マーケティング職	部長級	ワークスフーズ株式会社	東京本社	リサーチマーケテ	日本
2	WAP0267		まつの たけひさ 松野 武久	新卒	35年	77.7	A	既婚	マーケティング職	部長級	ワークスヘルスケア株式会社	東京本社	通販マーケティング	日本
3	WAP25352		なるせりか 成瀬 里歌	新卒	6年	76	A	未婚	マーケティング職		ワークスドリンコ株式会社	東京本社	第四係	日本
4	WAP1088		あいほら あきのり 相原 明紀	中途	13年	79.2	A	未婚	マーケティング職	課長級	日本レジェンドサイダー株式会社	青山事業所	ブランド戦略課	日本
5	WAP18125		せきくち たかし 関口 貴史	新卒	11年	73.3	A	未婚	マーケティング職	係長級	ワークスパートナースセールス株式会社	青山事業所	第一係	日本
6	WAP0997		としまかつや 豊島 勝哉	新卒	7年	78.5	A	既婚	マーケティング職	主任級	ワークスフーズ株式会社	東京本社	第二係	日本
7	WAP51872		うちむら たかお 内村 孝男	新卒	6年	69.5	A	既婚	マーケティング職	主任級	ワークスエンジニアリング株式会社	東京本社	第四係	日本
8	WAP48171		ながせ ななみ 永瀬 菜々実	新卒	9年	68.6	A	既婚	マーケティング職	主任級	ワークスリキール株式会社	東京本社	第三係	日本
9	WAP30728		いざわあやか 伊沢 彩加	新卒	2年	86	A	未婚	マーケティング職		ワークスヘルスケア株式会社	東京本社	第三係	日本
10	WAP09176		いたがき けいそう 板垣 恵三	新卒	15年	75.5	A	未婚	マーケティング職		ワークスドリンコ株式会社	東京本社	第一係	日本
11	WAP05268		ウー イェンヨン 吴 言咏	新卒	11年	71.5	A	既婚	マーケティング職		ワークスフーズ株式会社	上海事業所	第四係	中国
12	WAP06722		カレン フィリップス Karen Phillips	新卒	17年	74.8	A	既婚	マーケティング職		ワークスフーズ株式会社	オーストラリア支店	第四係	オーストラリア
13	WAP22891		グロリア ハバード Gloria Hubbard	新卒	6年	77.9	A	未婚	マーケティング職		ワークスフーズ株式会社	メキシコ支店	第二係	メキシコ
14	WAP31819		ケース ウィット Case Witt	新卒	2年	68.3	A	未婚	マーケティング職		ワークスフーズ株式会社	ニュージーランド支店	第四係	ニュージーランド
15	WAP26098		さこた あい 迫田 亜依	新卒	4年	75.8	A	未婚	マーケティング職		ワークスリキール株式会社	東京本社	第二係	日本
16	WAP59940		たにぐち はる 谷口 晴	新卒	2年	73.8	A	未婚	マーケティング職		ワークスエンジニアリング株式会社	東京本社	第四係	日本
17	WAP0419		さとう みか 佐藤 美音	新卒	9年	75.7	A	既婚	経営企画職	係長級	ワークスヘルスケア株式会社	東京本社	事業開発課	日本
18														
19														

3. アーキテクチャ決定の道のり

複雑なクエリをDBに寄せるか アプリに寄せるか？

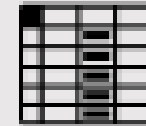
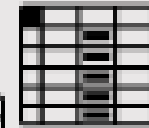
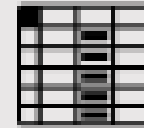
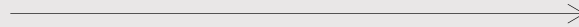
DBを使うアプリ設計の長年のジレンマ

DBに寄せたほうが効率的(データローカルティ)
一方、
アクセス制御や複雑なバリデーション処理は
DBだけではできない

RDBの世界観

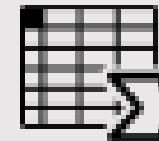
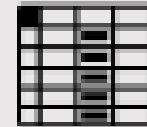
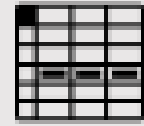
リアルタイムインデクシングとリッチクエリの両立
トランザクション管理(並行制御)の隠蔽

リアルタイムインデクシング



Simple indexing

リレーショナル演算



Rich query

宗教論争

DBMSを高機能化してOLTPとOLAPを統合する世界
(SAP HANA、VoltDB)

リアルタイムインデクシングを諦める世界

自分の中でも揺れてきた。

リアルタイムインデクシングの限界

本当にリッチなクエリ(自然言語処理や機械学習の活用など)は困難

クエリのオプティマイザが優秀になってもなお性能劣化する現実

バックグラウンド処理で割り切る世界

表示に必要なデータを(可能な限り事前に)作っておく

非正規化データを恐れない

データ増大を恐れない

バックグラウンド処理で割り切る世界 ; 更新系

更新系処理は、
遅延可能な処理と不可能な処理がある

遅延できない処理は、
並列処理と投機処理の工夫で高速化

バックグラウンド処理で割り切る世界

並列処理のためのデータ構造

イミュータブルデータ

追記型データ(チェンジセット、イベントデータ)

並列処理の基本戦略

楽観的並行制御(OCC:Optimistic Concurrency Control)

投機実行

ロックフリーアルゴリズム

[参考] プログラミングとの対比

イミュータブルデータ
楽観的CC

余計なメモリ消費
GCが必要
並列性が必要

メニーコア
メモリの余力

NoSQL

余計なディスク消費
GC相当処理が必要
並列性が必要

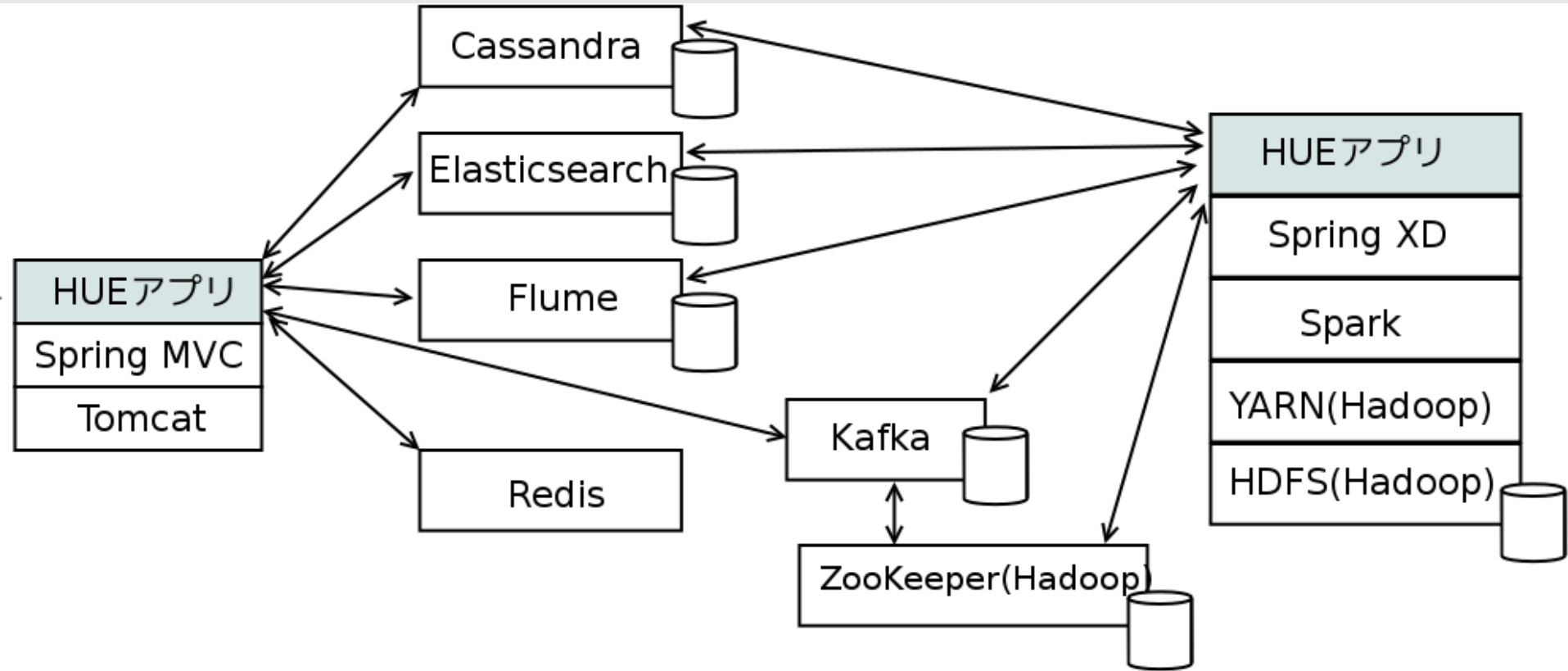
クラウド

バックグラウンド処理で割り切った世界が失うモノ

「one fact in one place」を失う

リアルタイムな一貫性の維持が困難
厳密に言うと、一貫性と更新性能が
トレードオフになる

現アーキテクチャの技術スタック



Cassandraの特徴; データモデル

KVS(Row-Keyとカラム群)

- ワイドローデータモデル($\text{Map}\langle \text{RK}, \text{SortedMap}\langle \text{CK}, \text{CV} \rangle \rangle$ 相当)

<http://www.slideshare.net/patrickmcfadin/the-data-model-is-dead-long-live-the-data-model>

- CQL(クエリ言語)により開発者の概念モデルをRDB風に行している
- スーパーカラム(非推奨。未使用)
- その他
 - 軽量トランザクション(CAS相当のアトミック操作)

Cassandraの特徴; ネットワークモデル

- 分散ハッシュテーブル(1ホップ)
- Row-Keyがパーティションキー(=シャードキー)
- レプリケーション数などはキースペースごとのチューニング項目

Cassandraの特徴; ストレージモデル

- 行指向ストア
- ディスクベース(コミットログ(WAL相当))
- インメモリの追記型領域Memtable
- Log-structuredデータ構造
- データ圧縮とコンパクション処理

選択した戦略; ネットワークモデルの設定

ランダムパーティション

Pros: 高いスケーラビリティ

Cons: レンジ(範囲)クエリができない

レンジクエリはワイドロー(後述)もしくは
Elasticsearchで代替

レプリケーション

レプリケーション数 $N=3$

デフォルトのコンシステンシレベルはquorum

選択した戦略; データモデルの指針

[参考] 先人の知恵:

REA(リソース、イベント、エージェント)

人、場所、モノ、事象

マスター系データとトランザクション系データ

選択した戦略; データモデルの指針

event:

事実の蓄積

クエリの主要な関心ごとは「いつ何があったか」

changeset:

changeの蓄積

changeとは: 変更というセマンティックを持つevent

クエリの主要な関心ごとは「変更の結果」

選択した戦略; データモデルの指針

event型テーブル: event保存

changeset型テーブル: changeset保存

relation型テーブル: changeset間の関係

analyze型テーブル: eventの分析結果

選択した戦略; アプリ開発者に見える世界

アプリ開発者はJavaコードのDTOクラスだけを意識
裏側のクエリを隠蔽
データベースマイグレーションもDTOベースで実施

自動インデクシングの仕組みを独自実装

アプリ開発者はDTOクラスにアノテーションを付与
バックグラウンド処理で別テーブルを自動更新
RDBのCREATE INDEXと同じ感覚で使う
Cassandraのセカンダリインデックスは未使用(遅いので)

トランザクション管理層を独自実装

選択した戦略; 主キーの設計例

```
CREATE TABLE post (  
  volume_number INT,  
  code text,  
  start_time TIMESTAMP,  
  name text,  
  disabled BOOLEAN,  
  PRIMARY KEY (volume, code, start_time)  
);
```

=> volume:パーティションキー、(code, start_time)でソート

=> リレーショナルモデルではなく、スケールのオーダーを
意識した巨大な表を意識して主キーを設計

縦(行数)のオーダーは無限

横(列数)のオーダーは10万程度まで

選択した戦略: 自動インデクシングのコード例

```
@lombok.Data
```

```
@AutoIndex(Employee.class)
```

```
@Join(with = AverageSalary.class, as = "a", where = {"deptId = a.deptId", "sectionId = a.sectionId"})
```

```
class EmployeeWithSalaryAboveAverageIndex {
```

```
    @Key
```

```
    private UUID id;
```

```
    private String name;
```

```
    private String deptId;
```

```
    private String sectionId;
```

```
    @Column(from = "a.averageSalary")
```

```
    private double averageSalary;
```

```
}
```

選択した戦略; 自動インデクシングのコード例

```
@lombok.Data
```

```
@AutoIndex(value = Employee.class, elasticsearchType  
= "{ElasticsearchTypeName}")
```

```
@Join(with = Project.class, as = "p", where = "projectId = p.id")
```

```
class EmployeeWithProjectIndex {
```

```
    @Key
```

```
    private UUID id;
```

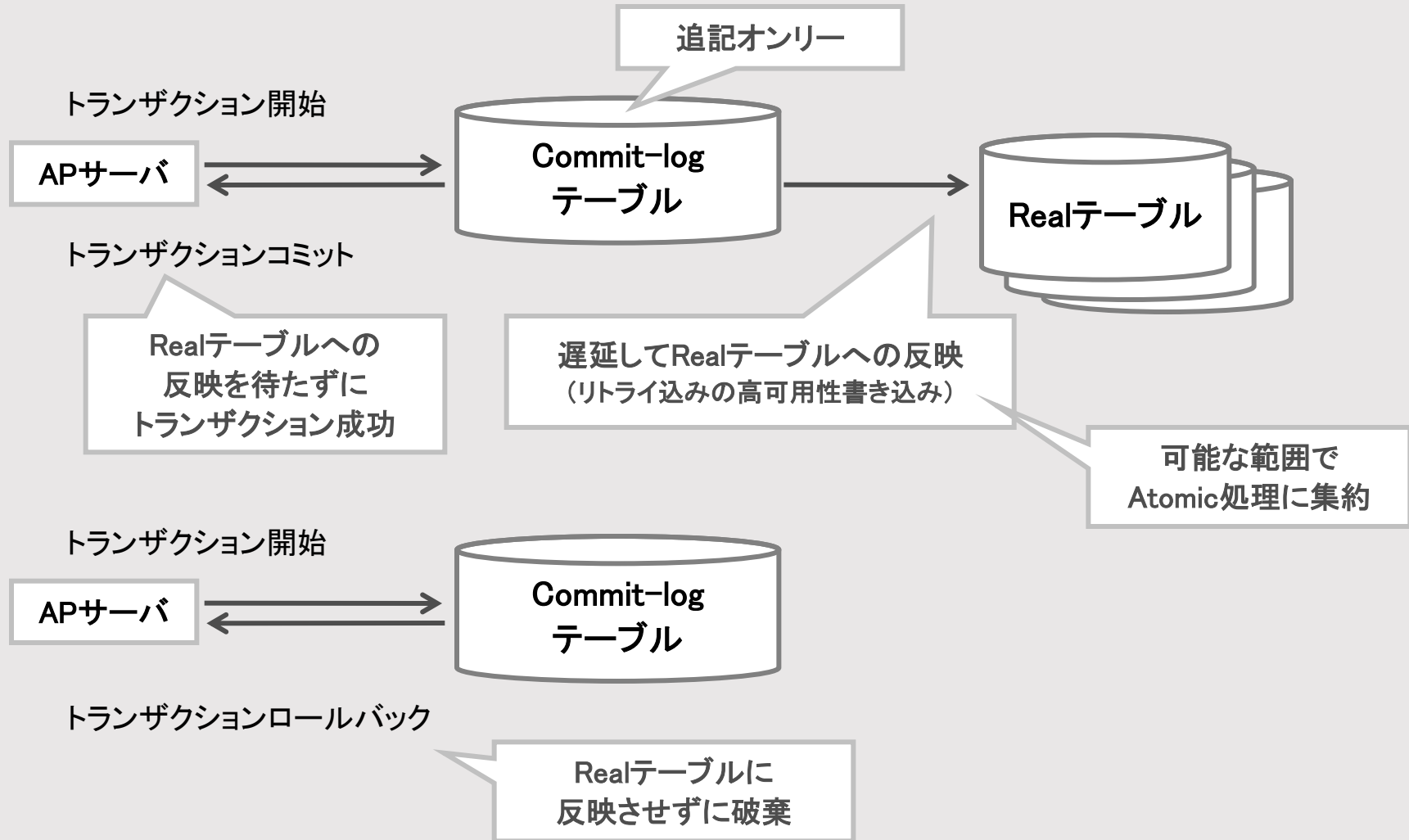
```
    private String name;
```

```
    @Column(from = "p.name")
```

```
    private String projectName;
```

```
}
```

選択した戦略: 独自のトランザクション管理層



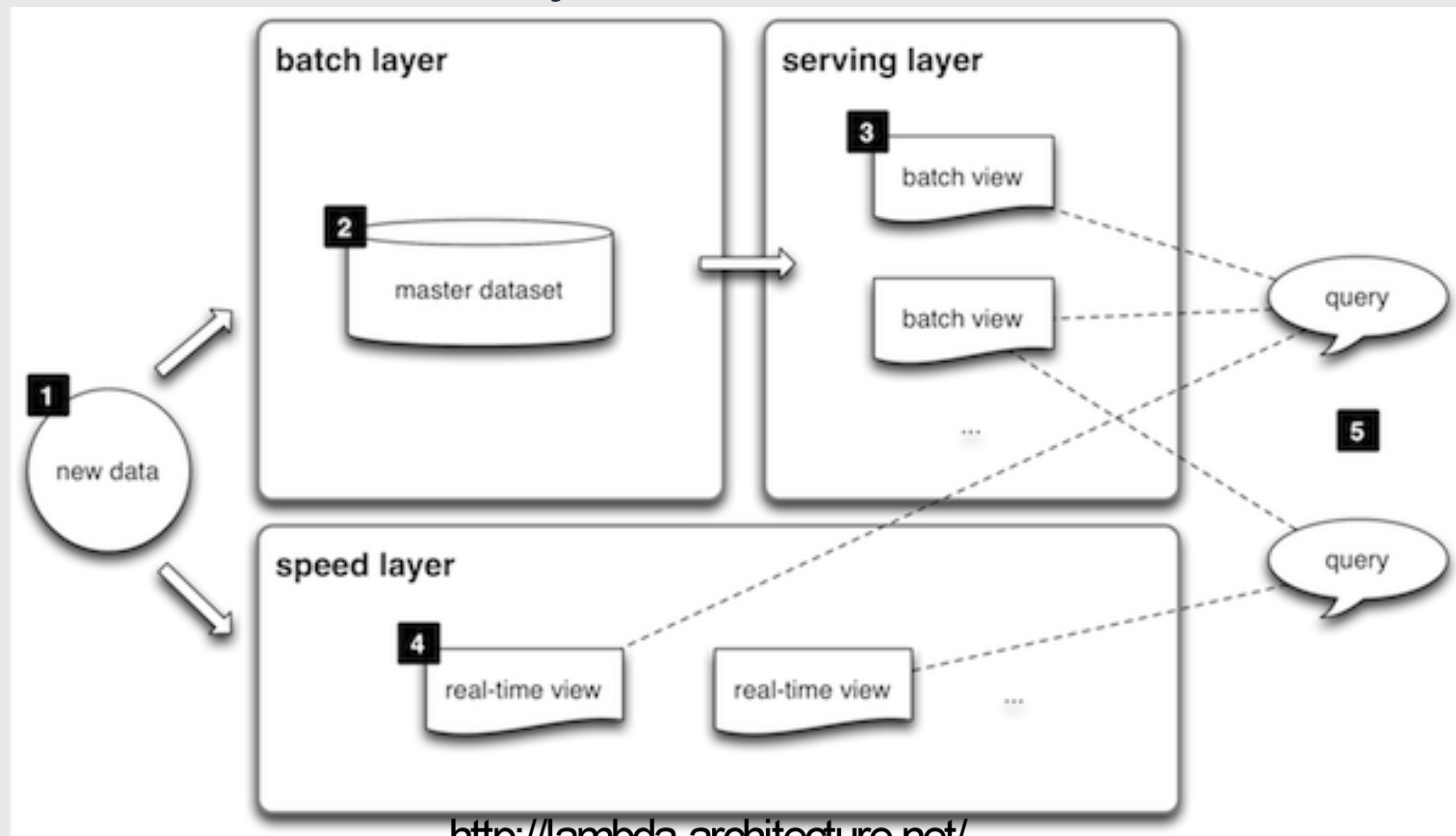
選択した戦略; トランザクション管理の戦略(考える順序)

0. トランザクション管理の分割を考える(MessageQueueで処理分割)
1. アトミック更新(CassandraのライトウェイトTX)で解けるならそれで解く
2. アプリ側の楽観的並行性制御で解けるならそれで解く
3. 結果整合性だけで解ける(収束するマージ操作やアンドゥ操作を定義可能)ならそれで解く
4. イミュータブルデータ(元データの更新の代わりに、新しいデータを追記オンリーで生成)で解けるならそれで解く
5. べき等性のある操作だけで実現可能ならそれで解く(参考 http://songofcloud.gluegent.com/2009/11/blog-post_18.html)
6. commit log tableを使う
7. 分散ロックを使って処理を直列化

3から6は「クライアントコンシステンシ(=更新の反映の遅れ)」の課題がある

選択した戦略: トランザクション管理の戦略 (クライアントコンシステンシ対策)

ラムダアーキテクチャ(read-your-writesコンシステンシ)



<http://lambda-architecture.net/>

最悪、アプリ側でのread repair相当処理(なるべく避けたい...)

現アーキテクチャで得られるモノ

高速な参照性能

裏でインデクシングして単純なクエリにしているから

高速な更新性能

複雑なインデクシングを後回しにしているから

[参考] 次期Cassandra

グローバルインデックス(裏で別テーブルを自動生成)

独自実装の自動インデクシング機能の一部は不要になるかも

RAMPトランザクション(低遅延の分散トランザクション)

独自実装のトランザクション管理層の一部は不要になるかも

4. RDBなしの不安

現アーキテクチャで失うモノ

リレーショナルモデル

ACIDトランザクション

枯れた実装

開発者の慣れ

運用ツールやノウハウ

リレーショナルモデルがなくても大丈夫か?

なければならないでどうにかなる

ワイドローデータモデル + Elasticsearch

発想の転換(リレーショナルモデルは実行状態よりスナップショット的な永続状態に適したデータ構造)

リレーショナルモデルがなくても大丈夫か?

アプリ開発者の典型的不安

任意カラムのインデックス

結合処理

集計処理

ソート処理

リレーショナルモデルがなくても大丈夫か?

アプリ開発者の典型的不安

任意カラムのインデックス

別テーブルで代替

結合処理

(次ページ)

集計処理

別テーブルで代替

ソート処理

ワイドローまたはElasticsearchで代替

結合処理の戦略

アプリ側で結合

IN句相当に制限があるのでクエリ数の $1+N$ 問題が発生

非正規化

別テーブル

ワイドロー

ACIDトランザクションがなくても大丈夫か?

Relaxing ACIDが原則

既に説明したトランザクション管理戦略

必要なら分散ロックで直列化する割り切り

コンシステンシ

文脈によっていくつかある

- レプリケーションの同期、非同期(準同期)の話
- 外部制約のような整合性の話
- one fact in one placeでないために
アプリから見ても整合が取れない話

コンシステンシ

- レプリケーションの同期、非同期(準同期)に関連する話
RDBMSでも発生する話。
CassandraのQuorumはひとつの解
- 外部制約のような整合性の話
Cassandraには、(ある意味)存在しない
- one fact in one placeでないためにアプリから見て整合が取れない話
read-your-writesコンシステンシで妥協

枯れた実装がなくても大丈夫か？

今のところCassandraの安定性に困ってはいない

ノード障害が適度に起きても、
サービス全体が死なない(クラウド脳)

開発者の慣れの問題は大丈夫か？

RDBのアンチパターン(1+N問題、EAVなど)も
アンチパターンとは限らない
などの発想の転換が必要

運用ツールやノウハウは大丈夫か？

社内に5年近い運用経験あり

監視ツール

無停止スケールアップ

ローリングアップデート

Cassandraの課題; 当然良いことばかりではない

テーブル数の上限

ソート機能の制約

レンジクエリの制約

read-repairの同期/非同期の使い分け

削除が弱い

IN句相当クエリの制約

それでも残る懐疑心

並列処理は本質的に難しいのでは？

イミュータブルデータは空間効率が悪くないか？

RDBMSでさばけないほど高負荷の業務アプリ？

シーケンスは実現可能？

分散ロックが必要になるケースはない？

Cassandraはreadが遅い？

脱RDBで本当に速くなる？

業務アプリにNoSQLは向いていないのでは？

それでも残る懐疑心(1)

並列処理は本質的に難しいのでは？

- ⇒ 難しい。脱RDBは脱トランザクション管理ではない。
- ⇒ ACIDとしてRDBMSに丸投げしたほうが圧倒的にラクだが、Relaxing ACIDが不可能とは思っていない

イミュータブルデータは空間効率が悪くないか？

- ⇒ 悪いと思う。特に削除処理に極めて弱い。
- ⇒ GC相当の処理(VacuumやCompaction)を許容できる物理リソースが必須(クラウドとNoSQLに一定の補完関係があるとすればここ)

RDBMSでさばけないほど高負荷の業務アプリ？

- ⇒ 普通に想像する業務アプリとだいぶ違う機能を目指している
- ⇒ マルチテナントで世界規模のサービスを目指している

それでも残る懐疑心(2)

シーケンスは実現可能?

⇒ LW-TXと楽観的並行制御で実現可能(waitがないので獲得側はbusy実装)

⇒ 特別に高速ではない

Cassandraはreadが遅い?

⇒ 今のところ困っていないが、キャッシュに逃げる日があるかも

それでも残る懐疑心(3)

脱RDBで本当に速くなる?

=> 並列制御を考えずに普通に作ると、一定の複雑さを持つアプリではむしろ遅くなると思う

業務アプリにNoSQLは向いていないのでは?

=> 「業務アプリ」とひとくくりにしないほうが良い

=> 会計業務はRDB利用時もINSERT操作中心で、ログを積み上げていく印象
(あるいはそのように見立てる)

5. まとめ

「脱RDBが技術的に正しいか?」と言われると、
技術者として真摯に答えると「わからない」が回答

しかし、RDBを使った普通の業務アプリは普通に作れるので、今さら(個人的に)頑張る意欲がそこまでない

しかし、普通の業務アプリを普通に作って、
世界のマーケットで戦えるイメージもない

「世界のどこで使っても応答性能100ミリ秒で、かつ落ちない業務アプリ」を作れば世界で戦える

この実現のためには、リレーショナルモデルだけでは困難だとは思
う(実装としてのRDBMSの可否は別として)

宣伝

Cassandra Summit 2015

4月
21

Cassandra Summit Tokyo, 2015

Cassandraが開くIoT時代! あのアップルも大規模にCassandraを使っている!

主催: 日本Cassandraコミュニティ



ハッシュタグ: #Cassandra

募集内容	Cassandra Summit Tokyo, 2015 無料	先着順 371/400人
------	---	-----------------

2015
4/21
(Tue)