

データベースシステム

第14回

理工学部情報科学科

松澤 智史

本日の内容

- NoSQLの製品
- 様々なデータベース(続き)
 - データウェアハウス
 - オブジェクトデータベース
 - XMLDB

NoSQLの製品

- Amazon DynamoDB
- Redis
- Cloud Firestore

Amazon DynamoDB

- AWSの1製品でNoSQL型データベース
- スケールに関して気にしなくてよい

Amazon DynamoDBの特徴

- Key-Value型, ドキュメント型を採用
 - データの格納と取得に特化(高度な最適化)されている
- 1日に10兆件以上のリクエストを処理可能
- 毎秒2,000万件を超えるリクエストをサポート
- 広告やゲーム業界などによく利用される

Amazon DynamoDBの特徴

- プライマリキー
 - ハッシュ型プライマリキー
 - このキーだけで一意に決定して検索できるようにする
 - ハッシュとレンジ型プライマリキー
 - HASH のキー & RANGE のキーの2つを使う
 - RANGE で範囲指定して検索できる
- ハッシュキーで別々のサーバに保存する
 - 一つのハッシュキーに処理が集中すると遅くなる

Redis

- キーと数種類の値型の対応関係を格納する高速なNoSQL
- Key-Value型データベース
- 値型
 - Strings – 最大 512MB のテキストまたはバイナリデータ
 - Lists – 追加された順に並べられた文字列の集合
 - Sets – 順序なしの文字列の集合で他の Set 型と交差、和集合、差集合演算を行うことができる
 - Hashes – フィールドと値のリストを保存するデータ構造
 - Bitmaps – ビットレベルの演算を実行できるデータタイプ

Redisの特徴

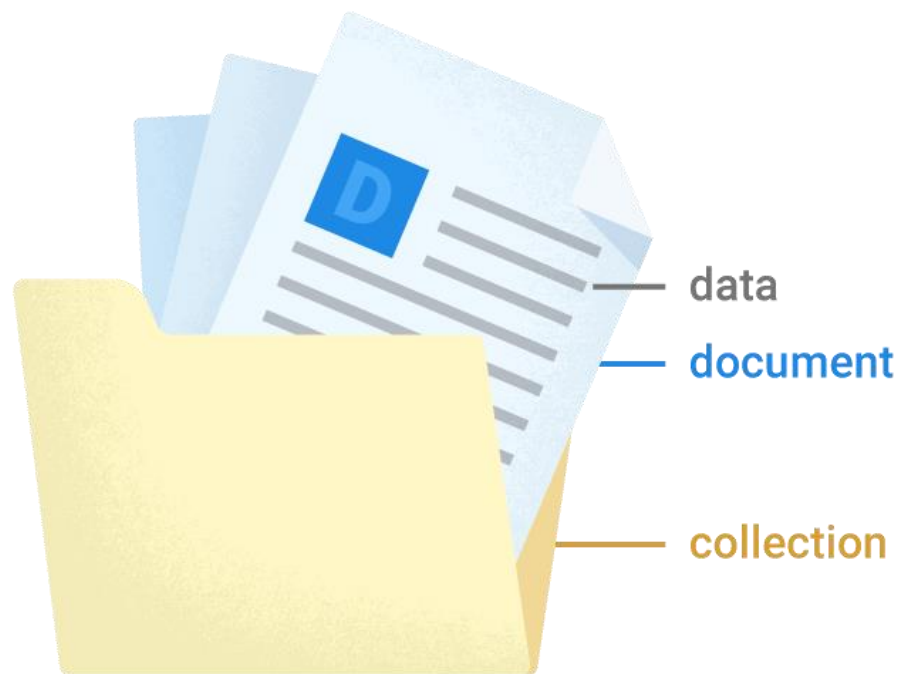
- マスタースレーブ型のレプリケーション
- インメモリデータストア
 - メモリ上のみでの動作が可能のため非常に高速
 - データの永続性が不要ない場合に限る
- 毎秒数百万件の操作をサポート

Cloud Firestore

- Google製
- モバイル・ウェブ・サーバ開発に対応したNoSQLデータベース

Cloud Firestoreの特徴

- PCのファイルシステムのような構造を持つことができる



様々なデータベース

- NoSQL
- データウェアハウス
- オブジェクトデータベース
- XMLDB

データウェアハウス

- とにかく何でも入れるという方針で作成されたデータベース
- 一度記録したら消さない(更新・削除は行わない)
- 正規化は行わない
- どんな形のデータでもそのまま受け入れる
- MySQLではArchiveエンジンが相性が良い
- 分析目的のために使用されることが多い

データウェアハウスの活用例

- 大量の商品が出品され、購入されるオークションサイト
 - 不正行動の監視
 - 出品者や購入者が「どうすればより満足するか」の分析
- 空席で飛行機を飛ばすくらいなら安値でもチケットを売りたいと思っている航空会社
 - 満席になりそうか、空席はいくつになりそうかを予測
 - 空席を埋めるためにダイナミックに値下げ／値上げ

データウェアハウスの比較

通常のRDBMS	データウェアハウス
古くなったデータは捨てる	1度記録したら消さない
最新のデータしかないので 「最新のデータを探す」手間がない	誤ってデータを消すということがない
データを受け入れる前に整形する	どのような形のデータも そのまま受け入れる
前もってデータベースの用途が決まってい るときに向く. データベース構築が簡 単に済む.	データベースを新しい用途に使うという チャレンジが容易に可能

データウェアハウスの利点

- 過去のデータを丸ごと残しておくという方針で作られている
- 未来予測に役立つ場合がある

データウェアハウスの操作

データウェアハウスを使用して分析をするための操作

- スライシング
 - 多次元のデータを2次元の表に切り取る操作
 - 例: 商品別売り上げ表を切り出し, 季節による商品の売り上げを分析
- ダイシング
 - データの分析軸を変更して視点を変える操作
 - 例: 商品ごと分析していたものを店舗ごとにする
- ドリリング
 - 分析の深さを詳細にしたりまとめたりする操作
 - 年月ごとに行っていた分析を粒度をあげて年単位にする(ドリルアップ)
 - 粒度を下げて日単位にする(ドリルダウン)

オブジェクトデータベース

- オブジェクト指向プログラミングで使うオブジェクトの形式で表現されるデータを格納するデータベース
- 「OODB (Object-Oriented Database)」と表記する
- マッピングや問い合わせ言語による指示が不要のためリレーショナルデータベース (RDB) よりも複雑なデータ構造を高速処理できる
- テキスト, 音声, 動画など, データの異なる種類や形式のデータが混在するデータ群の格納に適している

オブジェクト指向

- カプセル化
 - できるだけ他のプログラムから干渉されないように／他のプログラムに干渉しないようにする仕組み
- 継承
 - 再利用性を高める考え方
- ポリモーフィズム
 - 振る舞いを様々に変えられる仕組み

オブジェクト指向言語と関係データベース

オブジェクト指向言語で開発したオブジェクト



関係データベースの形式に変換する

この作業は煩雑で生産性が低いため、1990年代にデータベースにオブジェクト指向の概念を取り入れる動きが誕生

※オブジェクト指向プログラミング言語で記述されたアプリケーションソフトウェアと関係データベース(関係モデル)の間の不整合をインピーダンスミスマッチと呼ぶ

インピーダンスミスマッチを軽減する技術

- オブジェクトデータベース
- オブジェクトリレーショナルマッピング
 - データベースとオブジェクト指向プログラミング言語の間の非互換なデータを変換するプログラミング技法

データベースにオブジェクト指向の概念を導入する手法

- オブジェクトデータベース(ODBMS)
 - 従来のプログラミング言語もしくは新規に開発するプログラミング言語に永続化の機能を追加する
- オブジェクト関係データベース(ORDBMS)
 - 従来の関係データベースにオブジェクト指向の機能(カプセル化や継承など)を追加する
 - 単なるRDBMSとして使われることが多くオブジェクト指向の機能を積極的に活用する事例はあまり多くない
 - PostgreSQLは実はORDBMS
 - Oracleも8からはオブジェクト指向の概念を取り入れている

オブジェクトデータベースの特徴

- データはオブジェクトとしてデータベースに格納される
- オブジェクトに対してはその設計図であるクラスで定義されたメソッドを介してのみ扱うことができる
- オブジェクト問い合わせ言語(OQL:Object Query Language)が策定されているが実装ごとに差異がある
- 検索速度はRDBMSより速くなる可能性がある
 - Joinを行うことはほぼ無く、オブジェクトの参照をたどるのみ

オブジェクトデータベースの今後

- 一部の人々はオブジェクトデータベース技術は失敗であったとの見解をもっている
- オブジェクトデータベース技術の本質的な方向性は現時点においても有効であると考えている人々も多い

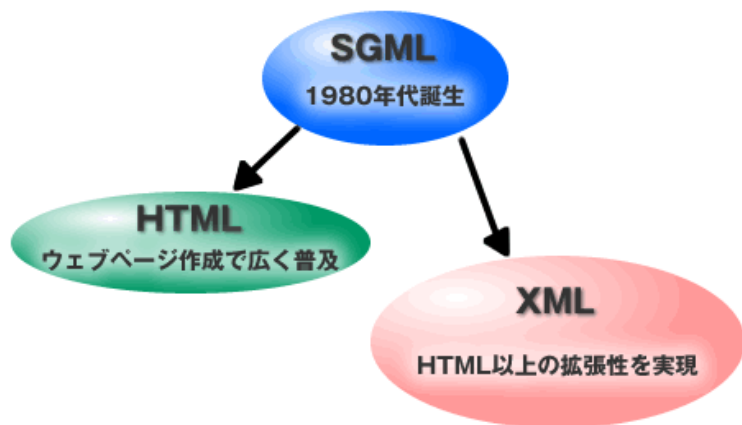
XMLDB

- 狭義
 - XMLのツリー構造をそのままデータ構造として持つデータベース
- 広義
 - 関係データベースにXMLを格納するもの
 - 単にテキストファイルとしてXMLを格納するもの
- 現在ではXPathやXQueryで検索するデータベースをXMLデータベースと呼ぶことが多い

XML

- 基本的な構文規則を共通とすることで、任意の用途向けの言語に拡張することを容易としたことが特徴のマークアップ言語
- Extensible Markup Language
- HTMLとは親言語(SGML)と同じ

SGML、HTML、XMLの関係を示す概念図



```
<?xml version="1.0"?>
<quiz>
  <qanda seq="1">
    <question>
      Who was the forty-second
      president of the U.S.A.?
    </question>
    <answer>
      William Jefferson Clinton
    </answer>
  </qanda>
  <!-- Note: We need to add
  more questions later.-->
</quiz>
```

XPath

- XMLの木構造をファイルパスのように表現する
- 例: \$DB1/sales/month/gets
- for if return 関数などの命令もサポート

```
for  
$i  
in (1 から 10) return 2 * $i
```

XQuery

- SQLのSelect等と同じくFROM, WHERE, ORDER BYなどの機能をXPathに追加したもの
- FLWOR
「For-Let-Where-Order-Return」式

```
for
$inv
in
$DB1/sales/invoice
let
$total := sum (for
$x
in item return
$x/price * $x/quantity)
where
$inv/date >= "2014-01-01"

order by
$total
descending

return
<invtotal>{ $total }</invtotal>
```

XMLDBの今後

- 性能的な問題で普及が遅れているがかなり改善されている
- Oracle DatabaseやMicrosoft SQL Serverでも Xpath, XQueryで検索する機能が実装された

これから流行る・・・かもしれない

まとめ

- NoSQLの製品
- データウェアハウス
- オブジェクト指向データベース
- XMLDB

質問あればどうぞ