
IoT eラーニング

IoTとデータベース①

(データベースの概要、IoTで活用されるデータベース)

国立大学法人 琉球大学

目次

- データベースの概要
 - データベースの仕組み
 - データベースシステムとは
 - データベースモデル
 - SQLとは
- IoT時代のデータベース
 - 新しいデータベースモデル
 - 分散ファイルシステム／分散データベース
 - NoSQLとは
 - NoSQLのデータモデル
 - NoSQLのアーキテクチャ
- IoTで活用されるデータベース
 - クラウド上のデータベース
 - リレーショナルデータベース型
 - NoSQL型
 - データウェアハウス型
 - インメモリ型
- データベースの選定
 - ハイブリッド構成の活用
 - ユーザー事例の参照とプロトタイピング

データベースの概要

● データベースの仕組み

データベースを直訳すると「データ基地」ということになる。
この言葉は第二次世界大戦頃の米軍において使われ始めたと言われている。

当時の米軍は各基地に資料（情報）を分散し保管していた。この方法では、資料を得るためには、まずどの基地にあるのかから調べればならず効率が悪かった。そのため、資料を一か所の基地にあつめ効率化を図った。

「資料を一か所に集めた情報基地」であることから、この基地はData（情報）Base（基地）と呼ばれていた。

このようにして生まれたデータベースは、多様な目的を考慮し整理されたデータの集まりとして現在では情報システムの中心に位置するようになった。

一方、データだけでは情報システムでは機能しないため、データベースを管理する仕組み（Database Management System : DBMS）が付加され、プログラムなどが効率よくデータベースを利用できるようになっている。

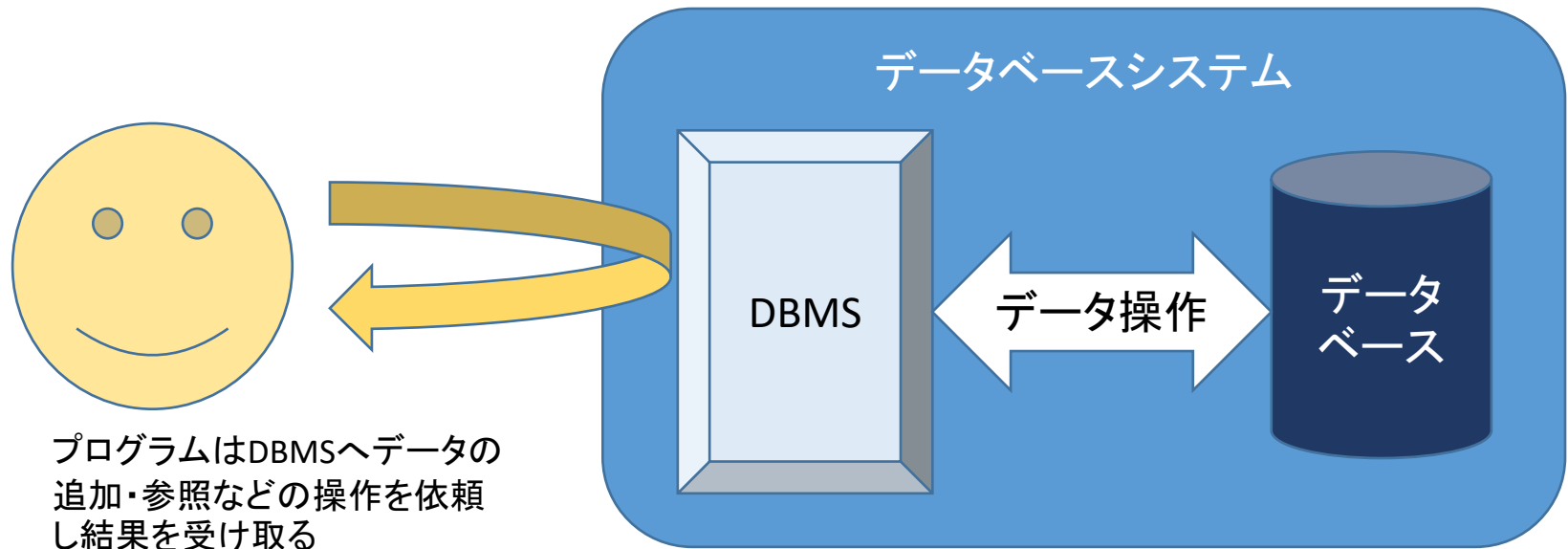
データベースの概要

● データベースシステムとは

データベースシステムとは、データが整理・保管された「データベース」とデータを追加・更新・削除・検索などの管理を行う「データベース管理システム（DBMS）」で構成されている。プログラムなどは、データの参照や更新などの要求をDBMSに依頼し、その結果を受け取ることによりデータを利用できる。

データベースシステムは、プログラムからデータを切り離し独立性を保っており、これによりプログラムの改修などから影響を抑えている。

また、同じデータに対する操作が重ならないよう、排他制御を行いデータの一貫性を保っている。



● データベースモデル

代表的なデータベースモデルに、階層型／ネットワーク型／リレーショナル型の3つがある。

➤ 階層型データベース

データが持つ階層関係でデータを表現したデータベースを階層型と呼ぶ。データは1つの親に1つ以上の子を持つ木構造となっており親子の関係として表現する。データの検索を行う場合は、親子の接続経路に従い処理を進めていく。スタートを1箇所とし接続経路に従い処理する定型的なデータ検索が非常に速いのが特徴である。

しかし、データが追加／削除された場合、木構造を変更しなければならないため柔軟性に欠けるのが欠点となる。メインフレームなどのシステムで利用されていたが、柔軟性の欠点から、今では利用するシステムは見なくなった。

データベースの概要

➤ ネットワーク型データベース

ネットワーク型は、基本的には階層型と同じ構造となる。ただし、データを網目（ネットワーク）上の関係で表現し、親と子の関係は多対多となる。また、検索は複数個所をスタートとすることで階層型に柔軟性を持たせた構造ともいえる。

階層型と比べれば柔軟性を持たせることができたが、非定型処理の対応が不得意な点もあることから、階層型と同じく今では利用するシステムは見なくなった。（利用していたシステムも階層型と同じくメインフレームが主となっていた）

➤ リレーショナル型データベース

データを表形式（行と列）で管理しているのがリレーショナル型となる。表はテーブルと呼ばれ、複数の表を定義することができ、テーブル同士の関係性を表現することができる。また全ての操作（追加・削除・参照等）にかかる処理速度は、ほぼ同じ処理速度となっている。

この型のDBMSをRDBMS（リレーショナルデータベース管理システム）と呼んでいる。代表的な製品として「Oracle」と「SQL Server」があげられる。

● SQLとは

SQLとは、リレーショナルデータベース用管理システム（RDBMS）を操作するための問い合わせ言語である。これによりデータベースに保管されているデータの操作を行う。これはRDBMSのプログラム言語であり、一般的なアプリケーションを構築するプログラムの言語ではない。

リレーショナルデータベースは行と列による表形式でデータを管理している。

データを操作する際には、

- ◆ 何処（行、列）から開始し、何処（行、列）までで終了するデータに対して何（参照、更新等）をする
 - ◆ 1行追加する
 - ◆ 何処（行）から開始し、何処（行）までの行を削除する
- などを効率よく問い合わせることができる構造となっている。

● 新しいデータベースモデル

近年は、取り扱われるデータ量の増加に伴い、リレーショナルデータベース（RDB）の課題の一つとしてパフォーマンス（システムの処理速度）があげられている。

この処理速度に影響与える要因として、データを一元で管理し厳密なトランザクションの管理をする必要があり、ストレージを共有する構成が必要となる。この構成がボトルネックとなり処理速度に影響を与える要因となっている。

もう一つの課題として、データモデルの限界がある。

RDBは表形式のテーブルによりデータを取り扱っており、CSVファイルのデータなど構造が決まっているデータを取り扱うのを非常に得意としている。これらのデータを「構造化データ」と呼んでいる。

これに対して、テーブルでは取り扱うことが難しいデータをまとめて「非構造化データ」と呼んでいる。

非構造化データの例として、XMLファイルやJSONファイルなどのデータがあげられる。

これらは、どのタグはどのような情報を持っているかという規則は定まっているが、どれだけのタグを持っているか、持っている情報がどれだけのサイズでどれだけ持っているかは決めておく必要がない。

このため、予めテーブルの構造を定義しておくテーブルにはこれらのデータを扱うことは非常に難しくなっている。

● 分散ファイルシステム／分散データベース

RDBが抱える課題に対する対応として「データローカリティ」とい考え方が出てきた。

データの処理を、

- ◆ データをプログラムへ持っていく
 - ◆ データのところにプログラムを持っていく
- の2つに分けることができる。

IoTなどによるビッグデータが注目さえるようになった現在において、データとプログラムのサイズは圧倒的にデータが大きくなっている。

そのような中で、

- ◆ 分散ファイルシステム (Hadoop Distributed File System等)
- ◆ 分散処理フレームワーク (Hadoop)
- ◆ 分散データベース (NoSQL等)

が現れた。

これらは、低速なストレージ性能 (CPU等と比べて) を分散で補いつつ、データローカリティによってネットワークの性能による影響を排除することを目的としている。

● NoSQLとは

NoSQLは非RDBMSの総称であり、広い範囲のカテゴリを持つデータベースである。

NoSQLにはいくつかの種類があり、データモデル／アーキテクチャにより分類する方法がある。

データモデルとは「データをどのような形で持っているか」を指す。
アーキテクチャは「データをどのように分散して持っているか」を指す。

● NoSQLのデータモデル

➤ グラフ型

グラフ型とは「ジョージ・ルーカス」-「監督」→「スターウォーズ」のように、グラフ構造と呼ばれる構造を持っている。格納しているデータより相互関係に着目している。Facebookの知り合い機能などで利用されている。

➤ キー・バリュー型

Key : Valueの単位でデータを格納する。シンプルな構造で応答も速いのが特徴である。

➤ カラム志向型

一般的なRDBは行を処理の主と捉える行志向型DBといえる。カラム志向型はRDBと同様に表形式の構造を持ちながら、カラム（列）での処理を主と捉える。行志向では列単位の大量集計、大量更新を苦手としているが、カラム志向型は得意としている。

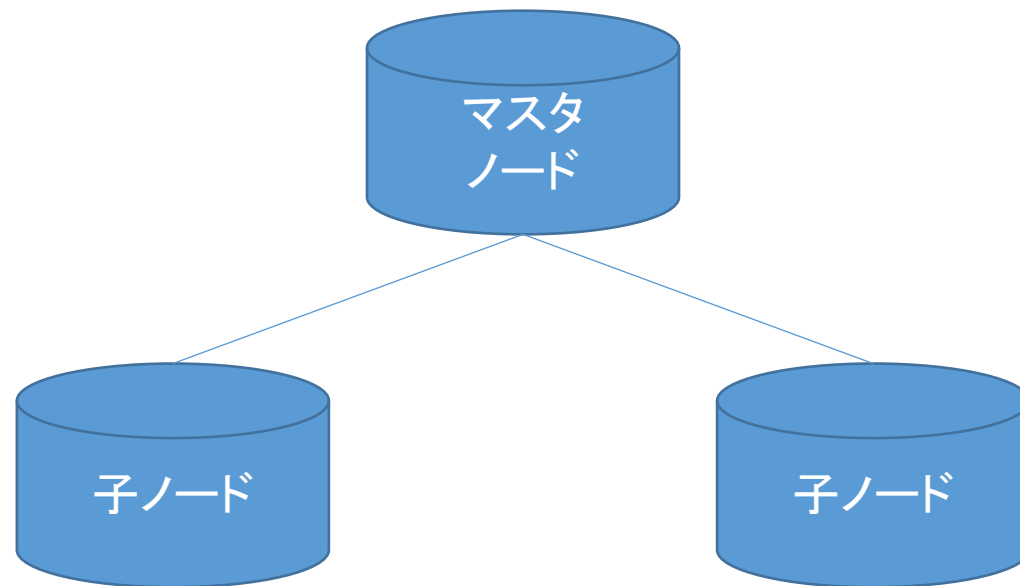
➤ ドキュメント志向型

XMLやJSONなどの構造を持つドキュメントをデータ単位として格納する。RDBのようにテーブルの構造を事前に定義しておく必要がなく、スキーマレスで格納ができる。

● NoSQLのアーキテクチャ

➤ マスタ型

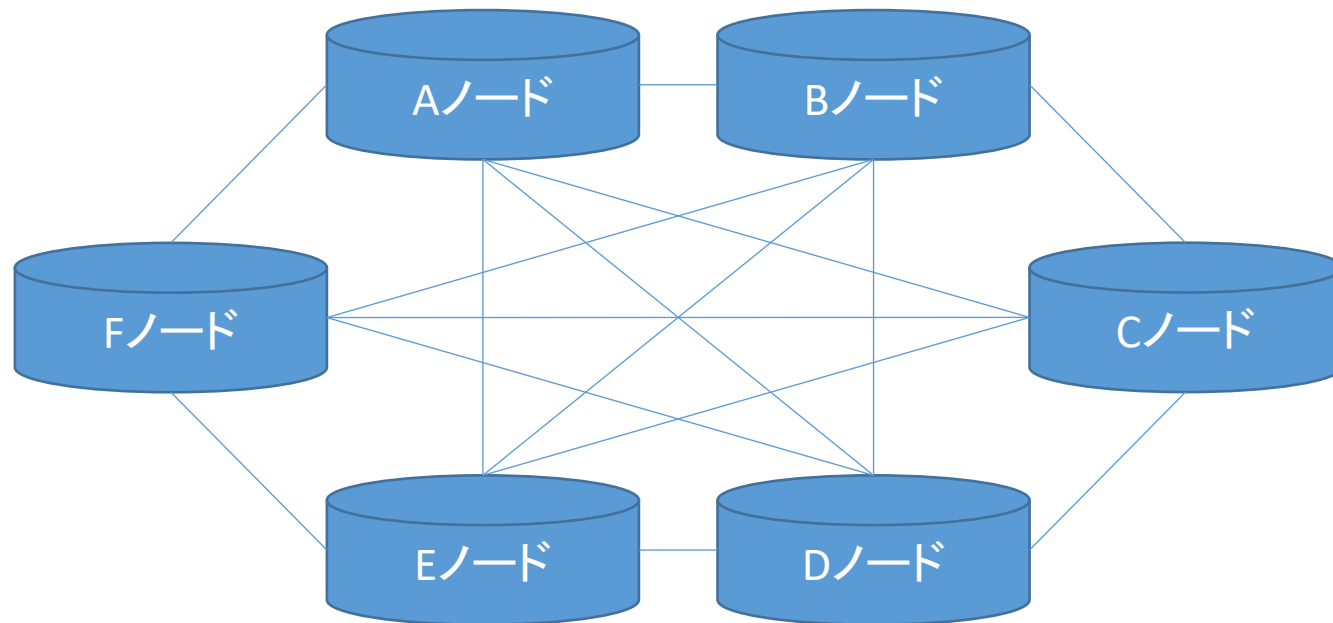
マスタと呼ばれるノードが複数のノードを配下に持ち管理する構造となる。マスタが停止するとシステムも停止してしまう構造と言える（マスタがシステム全体を管理するため）。



IoT時代のデータベース

➤ P2P型

全てのノードが等価の役割を持っている。この方式であると1つのノードが停止してもマスタ型のようにシステムの停止にはつながらない。



● クラウド上のデータベース

IoTの普及が進み、IoT機器などから集めたデータを分析し意思決定を進めて行く上でクラウドが重要な役割を担うようになってきた。

それにより、クラウド上のデータベースの役割も重要となってきたことから、クラウド事業者はDBのマネージドサービスを提供している。また、商用DBベンダーはクラウド戦略を進めており、業務システムなどで使用されていた既存のRDBもクラウド上での利用が進んでいる。

IoTを構築する上で、これらのサービスを利用することも選択肢の1つとして挙げることができる。DBマネージドサービスの種類を分類すると、大きく次の4つになる。

- ◆ リレーショナルデータベース型
- ◆ NoSQL型
- ◆ データウェアハウス型
- ◆ インメモリ型

● リレーショナルデータベース型

いわゆるリレーショナルデータベースをクラウド上で利用できるサービスである。トランザクションを中心にさまざまな利用ができる。

この型でのDBマネージドサービスとしては次のサービスがある。

- Amazon RDS
 - ✓ Amazon Aurora
 - ✓ PostgreSQL
 - ✓ MySQL
 - ✓ MariaDB
 - ✓ Oracle Database
 - ✓ Microsoft SQL Server
- Azure SQL Database
- Oracle Database Cloud Service
- Google Cloud SQL

● NoSQL型

リレーショナルデータベースでは扱いづらいデータを格納できるサービスである。適応性に優れ、あらゆるデータを扱える。

この型でのDBマネージドサービスとしては次のサービスがある。

- Amazon DynamoDB
- Azure DocumentDB
- Google Cloud Datastore

● データウェアハウス型

大規模な集計／分析を高速に処理することを目的に作成されたデータストアのサービスである。

特徴としては、

- ◆ 主題ごとに分類／整理し格納される
 - ◆ 論理的に統合される
 - ◆ 削除／更新はしない
 - ◆ 時系列を持つ
- ことである。

この型でのDBマネージドサービスとしては次のサービスがある。

- Amazon Redshift
- Azure SQL Data Warehouse
- Google BigQuery

● インメモリ型

インメモリ型キー値データ構造ストアにより提供されるサービスである。

データをメモリー上に展開し高速なパフォーマンスを提供している。

また、これを別のデータベースの「フロント」に配置することにより、パフォーマンスの高いインメモリーキャッシュを作成することができ、RDBやNoSQLなどの負荷を軽減することに利用できる。

この型でのDBマネージドサービスとしては次のサービスがある。

- Amazon ElastiCache
- Azure Redis Cache

データベースの選定

● ハイブリット構成によるデータベースの活用

クラウド上でDBを利用する際にどのサービスが最適かは利用状況によって異なる。

ただし、DBの選定においては、自社運用時のシステム構築時の手法と変わりはない。システムの機能要件の他に、性能や保守性などといった非機能要件をもとにサービスを絞り込むこととなる。

DBマネージドサービスでは、さまざまなタイプのサービスを提供しているのでDBを1つに限定する必要がなければ、メイン処理用や集計処理用などハイブリット構成を視野に入れて考えてもよい。

NoSQL型やデータウェアハウス型、インメモリ型とRDBMSを組み合わせることにより、適材適所を考慮しながらシステムの性能向上を図ることもできる。

● ユーザー事例の参照とプロトタイピング

クラウドの国内利用は進んでおり、業種別・用途別の事例が多数公開され充実してきている。検討しているシステムと似た事例を見つけ出せば、システム化を検討して行く上で大きなヒントとなる。

また、クラウドのサービスは、利用する上で特別なシステムを用意する必要がなく、価格やスペック等も公開されているため見積もりを取らずに直ぐに利用できるサービスがほとんどである。

さらに、IoT機器の開発においては、ArduinoやRaspberry Piなどフリーの開発環境が充実しており手軽に機器のプロトタイプを製作できるようになってきている。

これらを活用することで、必要な機能の絞り込みやデータ量の予測などの検討をプロトタイピングで進めることによりコスト削減や失敗しないシステムの作成を進めることができる。