

NoSQL を採用した SensorCorpus[®]

～Why we choose NoSQL for SensorCorpus～

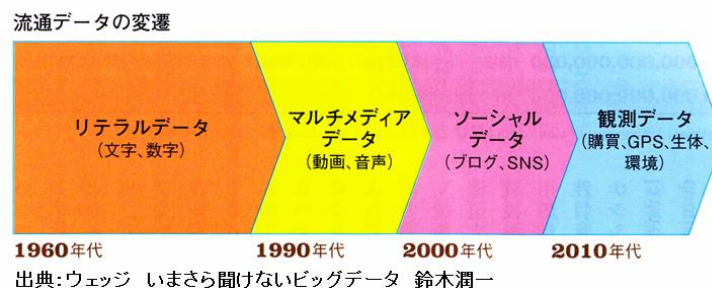
2015年9月10日

Rev1.0

1 IoT データとビッグデータ

IoT で扱われるデータは、今までのデータとは違って、以下のような特徴を持っています。

- データが大量である
- データが多種多様な属性を持っている
- 高速に処理をすることが求められる



アメリカでは年間1兆個のセンサーを使用する社会「Trillion Sensors Universe」が到来すると予測されています。そこでは農業・工業・流通/物流・医療/ヘルスケア・環境・社会インフラ等、ありとあらゆる分野がセンサーで覆われ、これら全てのセンサーがインターネットに接続されます。そしてそこから膨大な量のデータが生み出され、ビッグデータの適用範囲が拡大し、我々の社会や生活を一変させるといわれています。これまでとは桁違いのデータがそこに存在することになるのです。

2 センサーデータの特徴

例えば同じ温度を測るセンサーであっても、メーカーが変われば当然詳細なデータ仕様は異なります。製品情報の他にデータの測定型や単位の属性情報も様々です。また、温度や湿度、気圧などのデータであれば測定型や単位は1つであり、データも1次元で取り扱えますが、例えば加速度センサーの場合、データ型や単位が同じであっても軸数の増加によってデータは2次元～高次元に変化します。このようにセンサーデータが持つ様々な属性情報は多岐にわたります。

センサーデータは時系列データとして格納されます。センサーデータは一度格納すればデータを更新したり削除したりする必要がほとんどなく、データをいかに高速に追加していけるかが重要になります。

説明	測定型	単位	意味
温度計	T	#	単位なし数値
湿度計	H	%	パーセント
超音波距離センサー	US	dB	エネルギー比
圧力センサー	PA	C	温度
光センサー	PD	m	長さ
ジャイロセンサー	AV	g	重さ
放射線カウンター	RC	m/s	速度
3軸加速度センサー	A3	m/s ²	加速度
人感センサー	HD	G	重力加速度
CO ₂ 濃度センサー	MC	hPa	圧力
開閉センサー	FD	deg	度10進法
温度差センサー	TD	deg/s	角度/秒
動作センサー	MD	lx	照度
騒音計	SP	ppm	濃度
照度計	I	Bq	放射エネルギー
GPS測位計	GL	uSv/h	被ばく量
...	その他	その他 ...

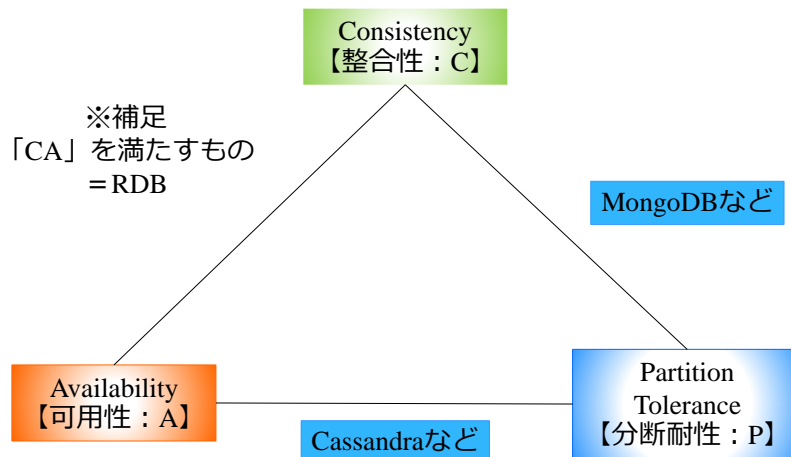
このように多種多様な属性を持ち、増加していくセンサーデータを効率よく管理しなければなりません。特にセンサーデータの属性の追加があるたびにデータベースの定義を変更する作業は煩雑です。これらIoTデータの特徴を満たすデータベースが必要とされます。

3 NoSQL の設計思想

IoT で扱われる、多種多様な構造を持った大量のデータを高速に処理するため、NoSQL(Not only SQL)が開発されました。従来の RDB は「ACID 特性」を基に「データの一貫性(整合性)」と「トランザクション」に重点を置いています。他方 NoSQL は「CAP 定理」を基にした「分断耐性」と「結果整合性」を重視することで RDB では容易には実現できなかったいくつかの技術を取り入れています。

	拡張方法 (Volume)	問い合わせ (Velocity)	データ形式 (Variety)	重要視していること	データの 整合性
RDB	スケールアップ	SQLを利用	データの構造化が前提	<ul style="list-style-type: none"> •Atomicity •Consistency •Isolation •Durability 	トランザクション
NoSQL	スケールアウト	キーでのシンプルな問合せ	ダイナミックスキーマ	<ul style="list-style-type: none"> •Consistency •Availability •Partition-Tolerance 	結果整合性

CAP 定理とは「Consistency : 整合性」と「Availability : 可用性」と「Partition-Tolerance : 分断耐性」のなかでどれに重きを置いてデータベースを設計するかを決める考え方です。NoSQL は特に「分断耐性」に重点を置いて設計されています。これにより NoSQL は「異なるネットワーク上の異なるハードウェアを並列に接続してデータベースを稼働させること」を実現しています。また結果整合性とは「基本的にどんな時でも稼働し、常に整合性を保つ必要はないが、結果として整合性が取れる状態になる」という考え方です。



これにより NoSQL は可用性と性能が格段に向上しています。

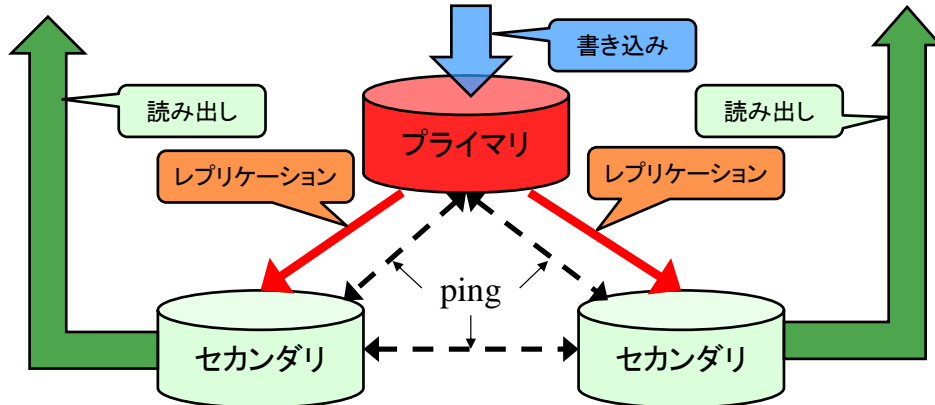
当社では IoT サービスプラットフォームである SensorCorpus を構築するにあたり、当初から NoSQL に着目し、その特徴的な機能を最大限に生かして実装しています。

4 NoSQL 技術の特徴

(1) スケールアウト

NoSQL は「分断耐性」を生かし、当初から「スケールアウト」を前提に設計されています。スケールアウトとは安価な IA サーバや仮想マシンを多く並べて処理能力を上げる拡張方法です。データ量が爆発的に増加することが見込まれるセンサーデータを格納・処理する際、サーバ台数を気にせず並列に追加していくことで、データ容量の追加も CPU やメモリの増設も、より低価格で、システム全体を止めることなく実行できます。

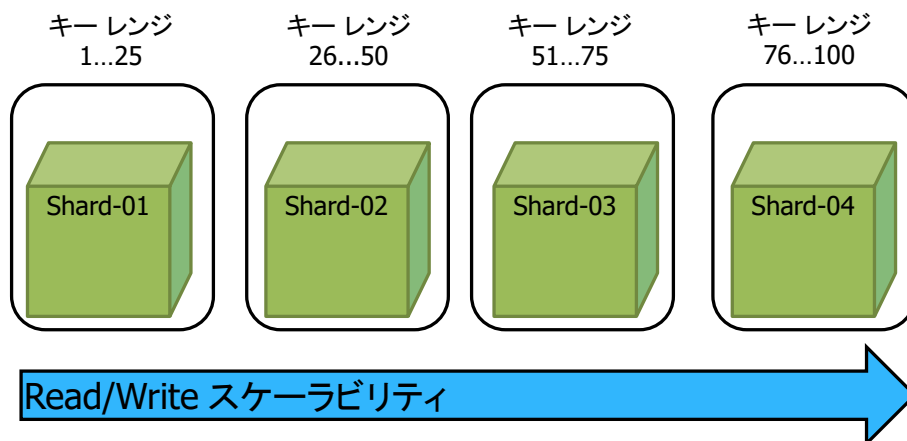
NoSQLはこのスケールアウトの特性を生かし、書き込まれるデータを別のノード(場所)に随時コピーしていく機能「レプリケーション」を備えています。データの複製先は最初にデータが書き込まれたノードとは物理的に異なるノードを自由に指定できます。これにより異なるデータセンター間でのデータコピーも簡単に行えます。



例えば上図の構成の場合、プライマリに書き込まれたデータは2つのセカンダリに随時コピーすることで、データは3重構成で保持されています。もしプライマリが故障してしまった場合は、直ちにセカンダリが自動的にプライマリに切り替わる「フェイルオーバー」が発生して書き込みを続けます。これによりデータベースを停止させずに故障したノードのハードウェア部分を交換することができるので、可用時間をほぼ100%維持しながらメンテナンスの時間とコストの両方を大幅に削減できます。

(2) 分散処理

NoSQLでは大量のデータの高速な書き込みと読み込みが可能で、データを書込む際に複数のサーバにデータを分割して保管することで、書き込み処理の負荷を分散する機能を持っています。また特定のサーバに書き込み負荷の片寄が起らないように自動的に分散の範囲(レンジ)を調整できます。データ量の増加によってレンジが増えればその分サーバを追加すればよいので、この分散処理の機能でNoSQLは圧倒的なスケーラビリティを出すことが可能です。



(3) ダイナミックスキーマ

NoSQLはRDBと比較してデータスキーマが緩やかなので非構造化データを格納することに優れています。またデータを格納した後でも動的にデータ構造を変更することが簡単にできるため、センサーデータのように構造変化が激しいデータであっても柔軟かつリアルタイムに処理が可能です。

5 IoTのために開発された NoSQL

SensorCorpus は、世界で人気の高い NoSQL である MongoDB と Cassandra を採用しています(Cassandra は現在実装中です)。これらは現在、IoT 向けの機能にフォーカスして開発が進められています。以下に IoT に言及している個所を引用します。

(1) MongoDB

MongoDB はドキュメント指向型の NoSQL です。JSON 形式で記述されたデータ=ドキュメントをそのままデータベースに格納できるので、Web アプリケーションとの親和性が高いことが特徴です。今年に入って 1 0 0 0 万ダウンロードを達成するなど、今注目を集めている NoSQL の一つです。

“MongoDB を使えば、かつては不可能だったセンサーデータの処理ができ、作れなかったアプリケーションを実現することができます。しかも、より速く、より少ない投資額で。”

MongoDB.com より引用

<https://www.mongodb.com/use-cases/internet-of-things>

(2) Cassandra

Cassandra はカラム指向型の NoSQL です。最大の特徴は SPOF(単一障害点)がないシステム構成が可能で、大量のデータを高速に書き込むことができます。最近ベンチャーキャピタルから大規模な資金調達が行われました。

“DataStax が扱う Cassandra は IoT 時代のためにデザインされたデータベースです。DataStax 社は、時系列のセンサー情報を高速に処理するための会社として設立されました。可用時間を 100%維持しながら、スケールと容量を簡単に増やすことを可能としました。”

DataStax.com より引用

<http://www.datastax.com/internet-of-things>