

## M2M データ解析手法の検討

安田 晃久<sup>†</sup> 北上 眞二<sup>‡</sup> 小泉 寿男<sup>†</sup>

特定非営利活動法人 M2M 研究会<sup>†</sup> 三菱電機株式会社<sup>‡</sup>

### 1. はじめに

機器同士をネットワークで繋げ、機器が有するデータを交換し合い、または当該データを基に機器の動作を変更することで、人手を介さず自動的にシステム化する技術を M2M (Machine to Machine) [1]と呼ぶ。M2M ではネットワークで繋がる機器全てを対象とするため、システム上には多数の機器が存在し、それに伴い、データサイズやデータ種別も増加する傾向にある。

近年、こうした大量／多種のデータを“ビッグデータ”と呼び、その解析手段として、NoSQL (Not Only SQL) と呼ばれるデータ管理手法が提案されている。M2M データについても、データサイズやデータ種別の増加を伴う場合には、NoSQL を用いた解析を行うのが適切である。

### 2. M2M データ解析における課題

M2M データの解析に NoSQL を適用しようとした場合、データ解析処理実行環境に伴う課題が生じる。一般的に、NoSQL ではデータ解析処理をスケールアウトさせるため、データ解析専用サーバを多数備えた分散処理システムを前提としている。そのため、例えば工場等に設置された M2M データ収集を行う組み込み機器と、データセンタ内に設置されたデータ解析サーバのように、データ収集者とデータ解析者が物理的に離れた環境下でデータ解析処理を行う際、ネットワーク経由で大量のデータをデータ解析サーバへ送信する必要があり、通信負荷増大、延いてはデータ送信からデータ解析終了までの遅延時間の増加を招く。

従って、M2M データ解析の目的がシステム最適化のための機器制御であるならば、遠隔地にあるデータ解析サーバへデータを送信せず、データ収集を行う組み込み機器内でデータ解析処理を行うことが望ましい (図 2-1)。しかし、こ

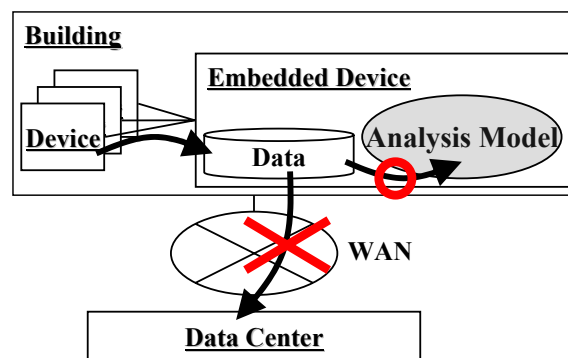


図 2-1 M2M データ解析における課題

のような組み込み機器単体でデータ解析処理を実行するための基盤システムについて、一般化されているものはない。

そこで、本稿では組み込み機器を対象とする、M2M データ解析処理に必要なデータ管理／解析システムを提案する。

### 3. M2M データ解析の提案手法

#### 3.1. 組み込み機器向け NoSQL の適用

組み込み機器に NoSQL を適用する場合、NoSQL の要件が組み込み機器側の要件と合わないケースが多い。提案システムでは、ストレージエンジンとして、オープンソースの KVS (Key Value Store) である LevelDB[2]を適用する事とした。その理由は以下の通りである。

- ファイルベースであるため、メモリの少ない組み込み機器にも適用可能
- サーバ／クライアント方式ではなくライブラリであるため、組み込み機器のように単体で利用するのに適している
- 動作に必要なライブラリが少なく、ポータビリティ性に優れる

#### 3.2. ドキュメント指向型データベース

LevelDB は KVS であるため、解析処理の際、解析モデルの設計の負担が大きくなってしまいうという課題がある。そこで、ストレージエンジ

Consideration of M2M Data Analysis Method  
Akihisa Yasuda<sup>†</sup>, Shinji Kitagami<sup>‡</sup>, Hisao Koizumi<sup>†</sup>  
<sup>†</sup>Specified Nonprofit Corporation Study Group on M2M  
<sup>‡</sup>Mitsubishi Electric Corporation

ンの上に JSON (JavaScript Object Notation) 形式の構造化データを取り扱う機能を搭載し、データの挿入、参照、削除が KVS より容易となる、ドキュメント指向型データベースシステムを提案する事とした (図 3-1) . この提案システムの利点は下記となる.

- レコードの値に JSON 形式の文字列を指定する事が出来るため、スキーマレスにデータを蓄積する事が可能 (キー毎に異なる形式のデータを格納する事が可能)
- アドホックなクエリで、蓄積データの中から任意の値を取得する事が出来る
- 他の解析システムに蓄積データのみを転送しても、値は全て JSON 形式の文字列であるため、容易に解析処理を実行する事が出来る.

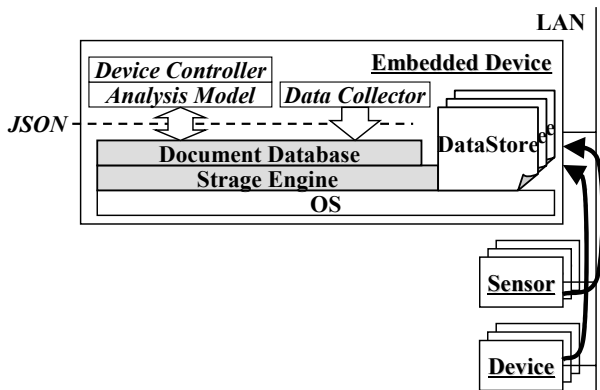


図 3-1 M2M データ解析システム

## 4. 実装評価と考察

### 4.1. 評価

提案システムを表 4-1 の環境に実装し、表 4-2 に示す計測を行った。なお、本稿での計測は全て、単スレッド、単一プロセスで実行した。

表 4-1 実行環境

項目	内容
H/W	Raspberry Pi Model B
CPU	ARM1176JZF-S 700 [MHz]
メモリ	512 [MB]
ディスク	Transcend microSDHC 32 [GB] (転送速度 45 [MB/sec])
OS	Raspbian wheezy (Debian ベースの GNU/Linux)
S/W	ドキュメント指向型データベース (実装プログラム) ----- LevelDB 1.15.0 ----- Snappy 1.1.1 (圧縮ライブラリ)

表 4-2 計測方法

項目	内容
挿入	ドキュメント指向型データベースの API 経由で 100 万件のレコードを連続挿入する。レコードのサイズは、一意なキー : 1~7 [B] と、全て同一の値 (JSON 形式の文字列) : 約 250 [B] のペアとする。
参照	挿入した 100 万件のレコードの中から、ドキュメント指向型データベースの API 経由で、キーとアドホックなクエリを指定し、1 件のレコードの任意部分の値を取得する。

### 4.2. 結果と考察

挿入処理と参照処理をそれぞれ繰り返し実行した際の、処理に要した時間の平均値を表 4-3 に示す。表 4-3 より、100 万件のレコードに対し、挿入/参照処理共に数ミリ秒以内に実行可能である事が分かった。また、蓄積可能なデータサイズについて、キーと値のデータサイズにも依存するが、表 4-1 の環境下で組み込み機器単体に数億件程度は格納可能である事が分かった。

この結果から、提案システムを用いると、遠隔地に設置された解析サーバが存在しない環境においても、組み込み機器単体で M2M データ解析処理を実行することが可能だと考えられる。

表 4-3 計測結果

項目	結果
挿入	0.743 [msec/件] (挿入後ディレクトリサイズ : 27 [MB])
参照	2.8 [msec]

## 5. まとめ

本稿では、遠隔地に設置された解析サーバを必要としない、組み込み機器向けドキュメント指向型データベースを提案し、評価と考察を行った。

今後は、機器障害発生時に蓄積データの信頼性を高めるための手法、及び、データ解析をより迅速に行うための解析モデル構築手法について、継続して研究を行う予定である。

### 参考文献

- [1] D. Boswarthick, O. Iloumi, and O. Hersent: "M2M Communications: A Systems Approach", Wiley, ISBN: 978-1119994757(2012)
- [2] Google LevelDB  
<http://code.google.com/p/leveldb/>