1ZC-04

Node.js と MongoDB による動画データの保存方式

原嵜　彩奈†　　工藤　司†
静岡理工科大学　情報学部　情報デザイン学科†

1. はじめに

今日、IoT(Internet of things)の発展に伴い、多様なセンサから大容量のデータがクラウドサーバのデータベース（DB）に保存され、分析されている。一方で、ネットワーク帯域の制約やセンサのフィードバック制御の遅延があり、エッジコンピューティングが提案されている。これは、センサの近くのエッジサーバで一次処理を行い、結果のみをクラウドサーバに送信するものである。しかし、クラウドサーバの分析でオリジナルデータが必要になった場合には、個別に再送する仕組みが必要になる。

ここで、大容量のデータを蓄積するために NoSQL DB が実用化されている。このうち、MongoDB では大容量データの効率的な保存、検索が可能である[1]。また、Node.js はサーバサイドの JavaScript 言語であり、MongoDB のコマンドを直接実行できる親和性を持つ。従って、エッジサーバで MongoDB を使用してオリジナルデータを保存し、必要に応じてクラウドサーバから検索できる方式は有効と考えられる。しかし、このような提案は見当たらない。

本研究では、この提案方式の有効性を評価するため、実際の在庫管理システムを対象に動画の DB 入出力機能の比較評価を行う。

2. 動画を活用した在庫管理システムの要件

工場では欠品の発生防止のため、適正な在庫を維持することが重要である。しかし、広い敷地内に展開した在庫の棚卸は労力を要する。そこで、在庫状況を撮影して、DB に保存し、在庫の充足度をオフィスで視覚的に判断できるシステムを構築した[1]。しかし、多数の在庫の棚卸棚を 1 枚 1 枚撮影するのは労力と時間を要する。すなわち、効率的な運用の要件として、監視カメラなどから常に動画を入力しておき、必要な部分を自動的に識別してサーバに転送できることが挙げられる。

3. 提案するシステムの構成

この要件を満たす、図 1 の構成の在庫管理システムを提案する。監視カメラあるいは従業員のタブレットから動画を連続的に入力し、動画 DB に保存する。次に、動画 DB から在庫棚の製品ラベルの表示に基づき、在庫画像と製品番号、日付などの管理情報を自動抽出し、生産計画にも続く必要在庫数と共に在庫 DB に保存する。在庫 DB は管理者によりオフィスで参照され、在庫の充足有無が判定される。この時、動画 DB も必要に応じて参照される。さらに、可用性を高めるために、DB はレプリケーションを採用する。

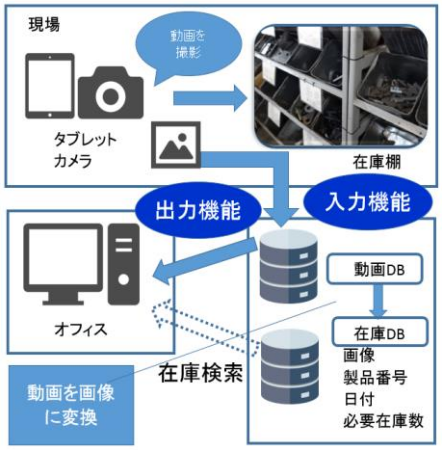

図 1　提案システムの構成

4. 実装

性能評価のため、図 1 の動画 DB アクセス機能を構築した。具体的には、動画データは一旦ファイルシステムに保存されるものとし、ファイルシステムと動画 DB 間の入力機能、出力機能を MongoDB(Ver.3.4.4)、Node.js(Ver.6.11.3)によって実装した。なお、Node.js では MongoDB とファイルシステムのアクセスのために、それぞれ mongodb、fs パッケージを使用した。

さらに、レプリケーションの採用に伴い、Node.js の接続先サーバにはレプリカセット内のプライマリ、セカンダリを指定し、フェイルオーバーできる設定とした。同じく、DB アクセスでは、Node.js の非同期な関数であるコールバック関数を使用し、例えば、入力機能では DB へのデータの書き込みと並行して、次に入力された動画から在庫データを抽出できる構成とした。

5. 実験と評価

性能評価実験は図 2 に示す環境で実施した。クラウドサーバはプライマリ、セカンダリ、アービタの 3 台の PC で MongoDB のレプリカセット

Video Data Storage Method by Utilizing Node.js and MongoDB
†Ayana Harazaki　†Tsukasa Kudo
†Shizuoka Institute of Science and Technology

構成し、1Gbit HUB で接続した。これは、各々、書き込み、読み出し、障害判定の役割を持つ。エッジサーバも PC で構成し、学内 LAN 経由で接続した。使用機器のスペックはエッジサーバが CPU E5-1620 (3.60GHz)、メモリ 8.00GB、クラウドサーバは CPU が E7500 (2.93GHz)、メモリ 4.00GB、OS は共に Windows7(64bit)である。

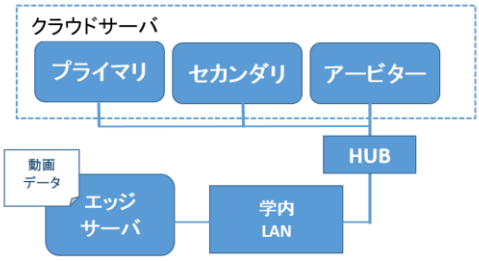

図 2　実装環境

この実験環境で、エッジサーバのファイルシステムと DB の間で、動画データの書き込み、および読み出しを実行した。この時、DB の環境としては、(1)エッジサーバに設置したもの、(2)クラウドサーバでレプリケーション環境のもの、(3)同じく単一サーバ環境のもの、の 3 種類を比較した。ここで、単一サーバ環境は、図 2 でレプリケーション設定を行わずに、プライマリサーバのみを稼動した場合である。

図 3、4 に上記の 3 種類の DB 環境について、各々、書き込み及び読み出しを実行した結果を示す。ここで、横軸は動画データのデータ量、縦軸は実行時間を示す。なお、実行時間は DB アクセスが完了するまでの時間を計測している。図に示されるように、エッジサーバ環境の場合の性能が非常に高く、書き込みでは単一サーバ、レプリケーション環境と比べてそれぞれ 11 倍と 14 倍であり、読み出しでは 21 倍と 20 倍であった。また、図 3 に示すように、書き込みでは単一サーバ環境の方がレプリケーション環境よりも高い性能が得られた。

次に、レプリケーションの可用性を評価するため、プライマリサーバを強制的に停止、再起動することでセカンダリサーバへと降格させ、プライマリサーバとセカンダリサーバを入れ替えた。その結果、サーバが切り替わってもプライマリサーバへの書き込み、セカンダリサーバからの読み出しを継続することができた。

さらに、コールバックによる処理の効率化について評価した結果を表 1 に示す。これはエッジサーバ環境で 181MB の動画を書き込んだ場合であり、約 4 秒の DB アクセス時間のうち、約 8 ミリ秒以降は次の処理が起動可能であった。

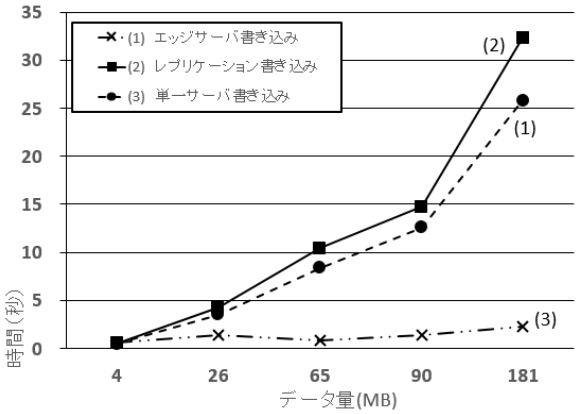

図 3　書き込み効率の比較評価

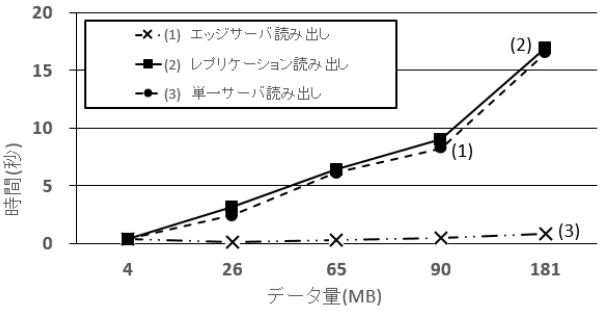

図 4　読み込み効率の比較評価

表 1　コールバックの評価

区分	時間
起動時間	7.914ms 秒
実行時間	4.211 秒

6. 考察とまとめ

IoT における大量データの連続入力のための要件を、動画を活用した在庫管理システムを事例として抽出し、プロトタイプにより DB アクセス性能を評価した。この結果、エッジコンピューティングではクラウドコンピューティングに比較して高い性能が得られ、提案方式は有効であることが分かった。

また、この場合、レプリケーションは性能面よりも、可用性向上の点で有効であると考えられる。逆に、エッジサーバに DB を設定した場合には、可用性の課題が考えられる。この対策としては、予備のタブレットなどを用いたレプリケーション環境の活用が考えられる。

なお、本研究は JSPS 科研費 15K00161 の助成を受けたものです。

参考文献

[1] T. Kudo, et al.: An application of MongoDB to enterprise system manipulating enormous data, Int. Journal of Informatics Society, Vol. 9 (in press).