一般論文

クラウドを活用した大規模ビル向け統合データシステムの開発

神戸事業所　技術第2部　広域第1課
中山 真吾、河野 洋一

1．まえがき

三菱電機㈱のビル管理システムは、国内において300箇所で稼働している。これまでのビル管理システムは、電源設備全般の電力監視及び制御がメインの機能であったが、各ビルのビル管理システムから収集したセンサ情報を集約し、業務システムまたはBEMS（Building Energy Management System）(注1)と連携するシステムが要求されている。

各ビルの情報を集約する場合、オンプレミス(注2)環境では、リソース（CPU・メモリ・ディスク容量）の動的な拡張が困難である。そのため、クラウド上のサーバに各ビルの情報を保存して、クラウド型のBEMSと連携するのが一般的である。

しかし、今回の開発対象は、センサから送られてくるデータ数が多い（最大220,000点）ことに加え、以下の理由から一般的なクラウド型BEMSの適用が困難であった。

- ビル設備サーバ向けエージェントのインストールが必要である
- ベンダ独自のプロトコルで通信を行うために、ビルシステムベンダの保守が困難である
- ベンダ独自のゲートウェイ(注3)（以下、G/W）等のI/F機器が前提で、ビル側にインタフェース（以下、I/F）機器の増加等の制約がある

大規模向けのビルに対応するために、本開発にて、大規模ビルの情報を集約し、他システムと連携するクラウド型のビル統合データシシステム（以下、統合DB）を構築することになった。

本稿では、本開発において発生した技術的な課題に対して実施した対策について紹介する。

2．統合DBの概要

統合DBは、「各I/F機器、情報通信機器の削減」、「一元化されたデータ管理による大容量、高速データ処理」、「機器管理番号(注4)一元化」、「クラウド活用」という目的で開発され、データをシステム間で連携するためのシステムである（図1参照）。

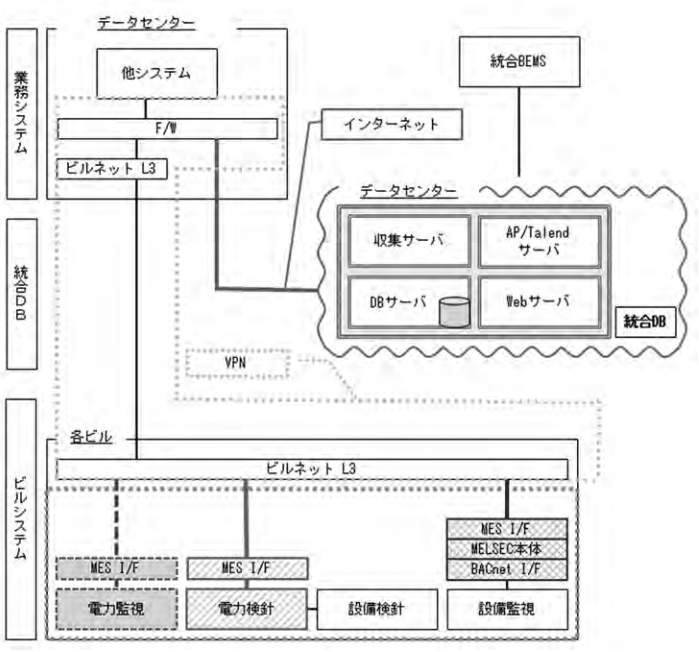

図1．統合DBのシステム構成

各ビルでは、監視対象設備の違いにより、さまざまなシステムが存在する。具体的には、図1の電力監視・電力検針・設備検針・設備監視が該当する。

統合DBは、これらの各ビルシステムからの検針データ、エネルギーデータを収集・蓄積する。

この実現に向けた統合DBの非機能要件は、表1の通りである。

表1．統合DBの非機能要件

項番	非機能要件	内容
1	性能要件（統合DB）	10分以内に10ビルから収集した各種データをDBに格納完了すること
2	性能要件（ビル側）	各ビルから統合DBに対して、1ビル3分以内にデータを送信完了すること
3	性能要件（タブレット）	複数端末（10端末）からの同時実行で、5秒以内に表示すること
4	セキュリティ要件	脆弱性を狙った攻撃からサーバを守るため、以下のセキュリティ対策を実施すること ・クロスサイトスクリプティング対策 ・SQLインジェクション対策 ・セッションハイジャック/リプレイ対策 ・OSコマンドインジェクション対策 ・パストラバーサル対策
5	使用性要件	監視員にとって操作が容易であること

（注1）室内環境とエネルギー性能の最適化を図ることを可能としたビルエネルギー管理システム。
（注2）企業などが情報システムの設備（ハードウェア）を自社で保有し、自社の設備において運用すること。
（注3）コンピュータネットワークをプロトコルの異なるネットワークと接続するためのネットワークノードを指す。
（注4）統合DBで取り扱う、アナログデータ、検針データに複数ビルで一意となる20桁（文字）の番号を指す。

3．課題と対策

表1に記載の非機能要件を満たすうえで解決しなければならない課題は、以下の4点である。

①クラウド環境で一元化された大容量データの高速データアクセスの実現
②WEBアプリケーション及びデータベースセキュリティ対策の実施
③タブレット端末のレスポンシブデザイン(注5)採用とサジェスト機能(注6)の採用によるユーザビリティの向上
④従来のビルシステムごとに設けていた各I/F機器の汎用化

これらの課題と対策について以下に述べる。

3.1 クラウド環境で一元化された大容量データの高速データアクセスの実現

3.1.1 課題

統合DBは、1ビルあたり10分周期で22,000点のアナログデータ及び検針データを取り込む必要がある。統合DBは、最大10ビルまで拡張することができるため、220,000点のアナログデータ及び検針データを取り込まなければならない。

しかし、大量のデータ登録と、更新が実施されるため、ファイルI/Oが性能障害になることが予想された。

今回採用したデータベースは、PostgreSQLである。一般的に本データベースは、フラグメンテーションが発生するため、定期的にバキューム処理(注7)が必要である。

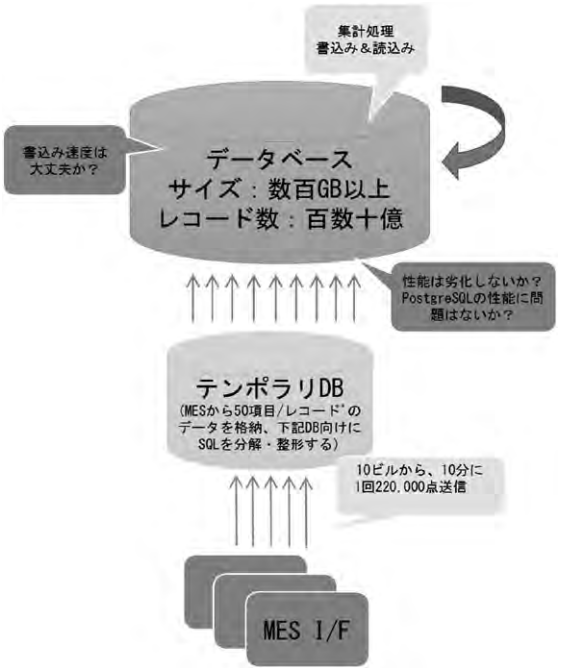

図２．性能検証部分イメージ図

このバキューム処理が動作しても、高速の読み出し及び書き込みを実現する必要があった(図2参照)。

3.1.2 対策

3.1.1の課題を解決するために、事前検証を実施した。

検証の観点は、以下の3点である。

①書き込み速度の検証
②大量データ登録時の応答性能
③バキューム実行時のリソース消費

(1)書き込み速度の検証

連続してカレントテーブルに22,000件の挿入(insert)文を発行し、レコードを登録する経過時間を検証した。また、一度に発行するSQL文の数量、プロセスの数を変更し、書き込みに要する時間の変化を確認した。

結果、3プロセスから100件ずつinsert を実行するよりも、3プロセスから200件ずつinsertを実行するほうが書き込み完了時間が速いことが分かった。しかし、3プロセスから300件ずつinsertを実行すると、200件よりも性能が劣化していた。よって、3プロセスから200件ずつinsertのみを実行する環境であれば、1分以内の22,000件登録は可能と判明した。(図3参照)。

	一度の実行数	終了時間
複数のプロセスを並列で実行	100	4．049秒
		4．263秒
		3．718秒
	200	2．317秒
		2．205秒
		2．374秒
	300	3．442秒
		3．522秒
		3．460秒

200件ごと、3プロセスの実行では、2.5秒以内に22,000×3（66,000）レコードを登録出来ている。

図３．プロセス実行時の書き込み性能

(2)大量データ登録時の応答性能

カレントテーブルに連続して10億件(22,000件×10ビルの1か月間の保存件数)のinsert文を発行し、レコードを登録する。登録は200件ずつinsertを実行する方式とする。1億件ごとに到達時刻を記録し、開始時からの経過時間を記録する。またSQLの応答秒数を記録した。

結果、1億件ごとに到達したタイミングで、200件ごとのinsert文応答時間を記録したところ、平均16ミリ秒程度であった。また、図4のグラフの通り、保存件数が増えても応答性能に影響がないことが分かった。

(注5) レスポンシブデザインは、閲覧者の画面サイズまたはウェブブラウザに応じて表示レイアウトが切り替わるデザインの手法。
(注6) 検索した文字列に関連の深い語句を逐次予測して表示する機能のこと。
(注7) PostgreSQLで、データの更新 (update)、削除 (delete) を行うと変更前のデータは物理的には消されずに残り続ける。その結果、データベースファイルのサイズが肥大化する。この問題を解消する処理を指す。

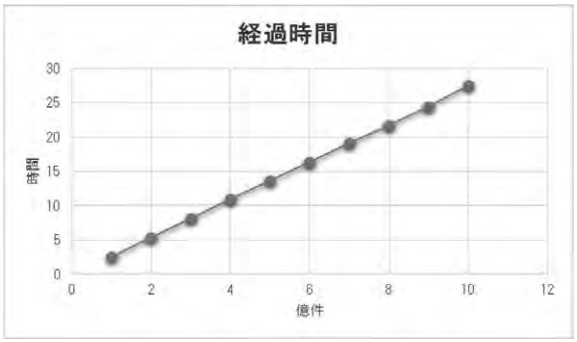

図４．大量データ登録時の応答性能

(3)バキューム実行時のリソース消費

バキューム自体がサーバのリソース(CPU、メモリ、ブロック書き込み数)をどの程度消費するかを確認した結果、バキューム自体がサーバのリソースを大きく消費しないことが判明した(図5、6、7参照)。

図５．バキューム実行時のCPU使用率

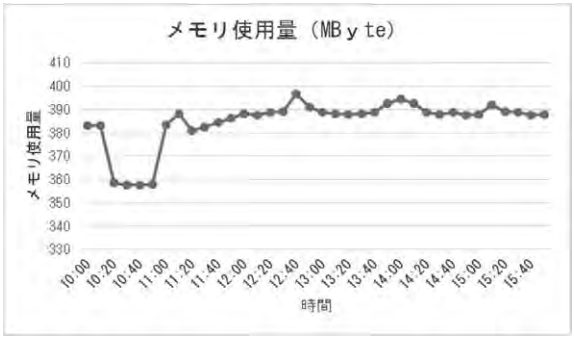

図６．バキューム実行時のメモリ使用量

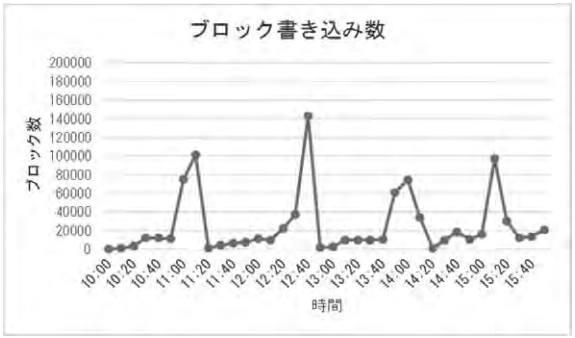

図７．バキューム実行時のブロック書込み数

この3つの観点の検証結果から、以下の対策を実施することでデータベースへの書き込み処理の高速化が実現できた。

- insertのみでデータを保存する仕組みとする。
- updateやdeleteを局所化することで、バキュームの時間を最小化する。

3.2 WEBアプリケーション及びデータベースセキュリティ対策の実施

3.2.1 課題

統合DBでは、クロスサイトスクリプティングやSQLインジェクション対策などさまざまなセキュリティ要件に対する対策を施している。しかし、これまでの施設・ビル分野のシステムでは、オンプレミス構成でのペネトレーションテストであり、セキュリティの試験項目は限定的であった。

しかし、統合DBは、クラウド上に構築するため、幅広くペネトレーションテストを実施してセキュリティ要件が満足していることを確認する必要があった。

3.2.2 対策

統合DBの開発で幅広いペネトレーションテストを実施するためには、脆弱性診断ツールを用いた試験とツールでは検出できない外部有識者の経験による試験が有効である。

ペネトレーションテストのツールは、IPA(注8)が公開している幅広くペネトレーションテストを自動で実行する脆弱性診断ツールである「OWASP_ZAP」を適用することとした(図8参照)。

図８．OWASP_ZAP画面イメージ

(注8) 独立行政法人情報処理推進機構の略称である。日本におけるIT国家戦略を技術面、人材面から支えるために設立された、経済産業省所管の独立行政法人である。

「OWASP_ZAP」で診断したセキュリティ診断項目は、表2のとおりである。

「OWASP_ZAP」にて、7件の警告があったが、この警告の対処及び外部有識者のペネトレーションテストを実施することでセキュリティ要件を満足した。

表2．セキュリティ診断項目

項番	項目	内容
1	パストラバーサル	制限されたディレクトリの外へ抜け出し、システム内の他ファイルやディレクトリへのアクセスができるかを検査する
2	リモートファイルインクルージョン	include/require文でユーザ入力を未検証のまま使用すると発生する脆弱性を検査する
3	Server Side Include	特定パラメータ値がServer Side Includeコマンドを実行する可能性を検査する
4	クロスサイトスクリプティング	ユーザのアクセス時に表示内容が生成される「動的Webページ」の脆弱性を検査する
5	SQLインジェクション	アプリケーションが想定しないSQL文を実行させることにより、データベースシステムを不正に操作する脆弱性を検査する
6	Server Side Code Injection	レスポンスボディに、断片内の文字列（掛け算の結果）が存在するかを検査する
7	OSコマンドインジェクション	Webアプリケーションなどを経由してサーバのOSコマンドを実行する脆弱性を検査する
8	ディレクトリブラウジング	ディレクトリの内容一覧が参照できてしまう脆弱性を検査する
9	外部リダイレクト	外部にリダイレクトしてしまう脆弱性を検査する
10	バッファオーバーフロー	データがあふれても格納されてしまうという性質を悪用して、スタック領域内のプログラムの戻り先を書き換えてしまう脆弱性を検査する
11	書式文字列エラー	printf（）、syslog（）などの関数の引数の特性を悪用して、メモリの内容を不正に参照することや、悪意のあるコードを実行される脆弱性を検査する
12	CRLFインジェクション	Webサーバが返すHTTPレスポンス内に改行コード（CR+LF）が配置できることを悪用して、予定外のHTTPレスポンスヘッダ行や空行を付加する脆弱性を検査する

3.3 タブレット端末のレスポンシブデザイン採用とサジェスト機能の採用によるユーザビリティの向上

3.3.1 課題

統合DBでは、保存したアナログデータ及び検針データをタブレット端末から参照し、メータの指針値確認が可能な機能を提供している。統合DBの開発には、監視員がタブレット端末を利用して該当するメータ情報を最短の操作で参照したいという要望があった。

また、機器管理番号の20桁入力が面倒という改善要望があった。

3.3.2 対策

(1) レスポンシブデザインの採用

タブレット端末のユーザビリティ向上のために、端末による表示の切り替えを実現するレスポンシブデザインを採用した。

タブレットでは、監視員によって端末を縦向き、横向きなど使い方が異なる。縦向きでレイアウト構成していた場合は、横向きにすると画面の表示が切れてしまう。そこで、端末の向きに合わせて画面切り替え時に動的にレイアウトを切り替える仕組みとした(図9、図10参照)。

図9．タブレット画面（横向き）

図10．タブレット画面（縦向き）

(2) サジェスト機能の採用

機器管理番号を入力するごとに候補一覧が表示できるサジェスト機能を採用した(図11枠部分)。これにより、文字入力により候補が絞られるため、該当メータ表示時間が短縮できた。

図11. サジェスト機能画面

3.4 従来のビルシステムごとに設けていた各I/F機器の汎用化

3.4.1 課題

従来は、ビルシステムごと(電力監視、電力検針、設備監視、設備検針)にG/WとしてI/F機器を設けていた。各ビルシステムが、他システムと連携する際は、ビルシステムごとかつ、稼働しているシステムの機種ごとに、プロトコルやデータ構造が異なるため、受注するシステムごとにプロトコルやデータ構造部分の作り込みが必要であった。

そのため、ビルシステムごとかつ稼働しているシステムの機種ごとにI/Fを規定するのではなく、統一したI/F仕様で更新・維持運用コストを削減することが求められた。

3.4.2 対策

統一したI/Fで統合DBに送信するために、シーケンサ(MELSEC)にMESインタフェースユニット(注9)(以下MES I/F)を実装し、統合DBとインタフェースを取る構成とした。統合DBでは、MES I/Fの設定方法及びデータ収集用のプログラムまでを定義し、シーケンサのその他の設定については対象外とした(図12参照)。

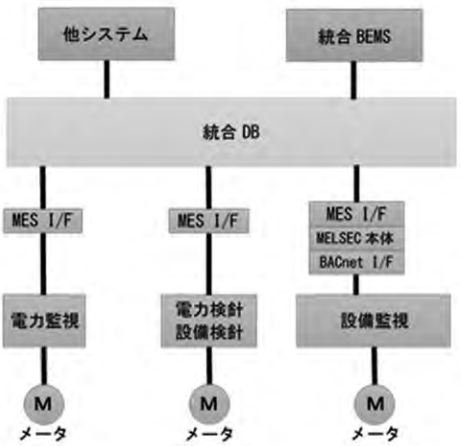

図12. G/W装置（MES I/F）を含めた統合DBの構成

また、シーケンサのデータ収集する保存領域をパターン化することで、ビルシステムごとにプログラムの変更が発生しない仕組みとした。

これにより、ビルシステムが異なっても統合DBとI/F機器との間でインタフェースの統一を図ることができた。

4. 今後の展望

今後は、エネルギー最適化システムとの連携や昇降機遠隔監視システムなどの他システムと連携することで、新たなサービスを顧客に提供していく。

そのためには、アナログデータや検針データだけでなく、画像データや音声データも取り込む必要がある。これらのデータを蓄積するためのアーキテクチャや、分析/解析技術が今後の課題である。

5. むすび

今回、施設・ビル分野のシステムではじめて①クラウドの導入、②大規模データの蓄積・集計、③オンプレミスで扱わなかったセキュリティ技術の適用を実施した。

本開発で、クラウドのメリット・デメリットがノウハウとして蓄積できたので、顧客のニーズにあわせてオンプレミスまたはクラウドどちらで実現するのが最適かを判断できるようになった。

今後も新たな技術要素を取り入れて、顧客にとって最適なシステムを開発していく所存である。

最後に本開発にあたり、技術的な助言をはじめとして、様々な面でサポート頂いた関係者の方々に深く感謝申し上げる。

(注9) シーケンサから能動的に、直接データベースサーバへデータを送信することができるユニットのこと。設定用のソフトウェアで必要なデータを指定するだけで、プログラムレスで通信可能である。

執筆者紹介

中山　真吾　　ナカヤマ　シンゴ
2002年入社。主にビル管理システムのソフトウェア開発に従事。現在、神戸事業所技術第2部広域第1課。

河野　洋一　　カワノ　ヨウイチ
1988年入社。主にビル管理システムのソフトウェア開発に従事。現在、神戸事業所技術第2部広域第1課。