第 15 回報告書 2023 年 6 月 23 日

ElasticSearch サーバーのデータ移行について

祖父江匠真

1 概要

今回は, ElasticSearch サーバー間でのデータ移行と, その際に行われた重複データの削除方法, kibana による可視化結果について報告する.

2 データ移行手順について

co2 のデータ移行を行う上で, タイムスタンプと部屋番号の組み合わせが重複しているデータが一部存在しており, この重複データを取り除いた上でデータ移行を行う必要があったので, 一度, 移行元の ElasticSearch サーバーのデータをローカルマシンにエクスポートして, 重複データを取り除いた上で, 移行先の ElasticSearch サーバーにデータをアップロードした.

2.1 データのエクスポート

移行元の ElasticSearch サーバーのデータのローカルマシンへのエクスポートには, elasticdump [1] ライブラリを使用して, JSON 形式でエクスポートした. その際, co2 という文字列を含むインデックスのデータのみをエクスポートした.

2.2 データの重複削除

重複データの削除は SQLite データベースを用いて行った.

SQLite データベースはリレーショナルデータベースの一種であり, 複合主キーを使って複数のテーブルカラムの組み合わせを一意の識別子として扱うことができる. これにより, 同じ組み合わせのデータを重複して挿入しようとした場合, データベースエンジンがコンフリクトエラーを発生させ, 重複データの挿入を阻止する. そのため, 今回の重複データ削除には適していると判断した.

今回使用した SQLite データベースでは, 部屋番号（number）とタイムスタンプ（JPtime）を一意のキーとして設定した. 以下のリスト 1, リスト 2に示すように, 移行元の ElasticSearch サーバーに保存されている co2 インデックスのドキュメントは, フィールドのメンバーが統一されておらず, 一部センサー情報が存在しない場合がある. そのため, データの挿入時にコンフリクトエラーが発生した場合は, 既

存のレコードと挿入しようとしたレコードを比較し, 既存レコードの値が NULL であるカラムにおいて, 挿入しようとしているレコードの値が非 NULL である場合には, 既存レコードのカラムの値を更新するようにした. これにより, 重複データ削除時に一部センサー情報などが欠けてしまう問題を解決した.

Listing 1: _source フィールドのメンバー数が少ないドキュメント

```
{
  "_index": "co2_e411",
  "_type": "_doc",
  "_id": "nEi2nnoB2-iFXnrMOobM",
  "_score": 1,
  "_source": {
      "utctime": "2020-10-09T05:09:06+00:00",
      "number": "E411",
      "PPM": "481",
      "data": "Thingspeak"
  }
}
```

Listing 2: _source フィールドのメンバー数が多いドキュメント

```
{
  "_index": "co2_e411",
  "_type": "_doc",
  "_id": "YKBqU4QBugDzeydA2gyi",
  "_score": 1,
  "_source": {
      "RH": 26.98,
      "PPM": 423,
      "JPtime": "2022-11-06T22:45:30.080925",
      "ip": "172.23.68.19/16",
      "utctime": "2022-11-06T13:45:30.080895",
      "TEMP": 24.47,
      "index_name": "co2_e411",
      "ms": "",
      "number": "E411"
  }
}
```

2.3 データのインポート

重複データ削除後のデータが保存された SQLite テーブルからすべてのレコードを読み出して, ターゲットの ElasticSearch サーバーに移行した.

その際, python の elasticsearch ライブラリを使用し, タイムスタンプが 2023 年より以前のデータは 2022_co2 という名前のインデックスに保存し, 2023 年のものは 2023_co2 という名前のインデックスに保存した.

3　kibana によるデータの可視化

移行後のデータを kibana を用いて可視化した.

2022_co2 インデックスと, 2023_co2 インデックスについて, 横軸をタイムスタンプとし, 縦軸を PPM, RH, TEMP としてそれぞれプロットしたものを図 1 〜 図 6に示す.

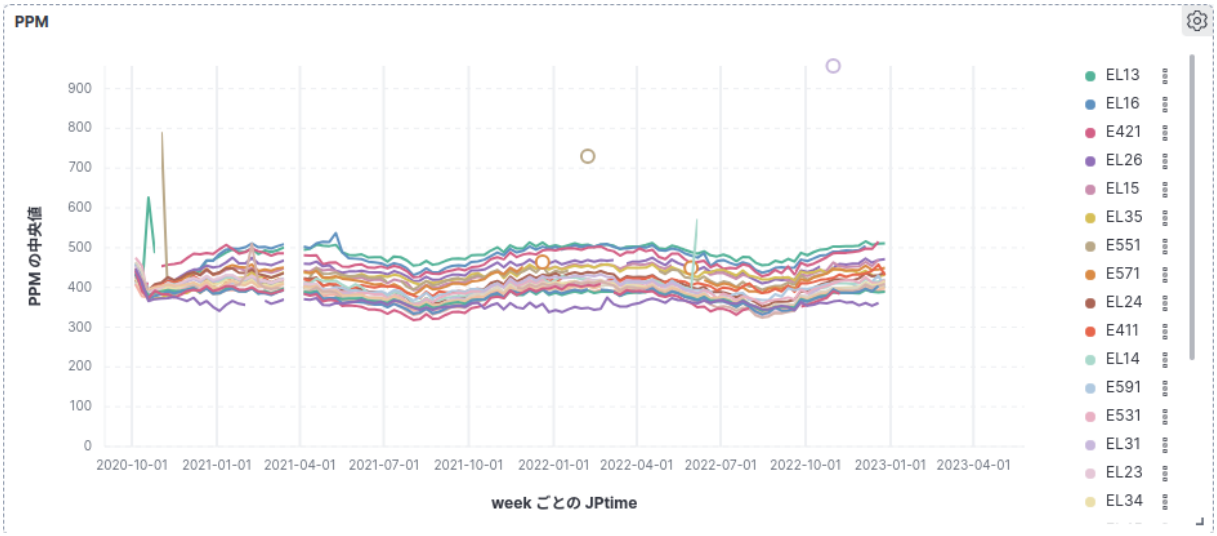

図 1: 2022_co2 の PPM

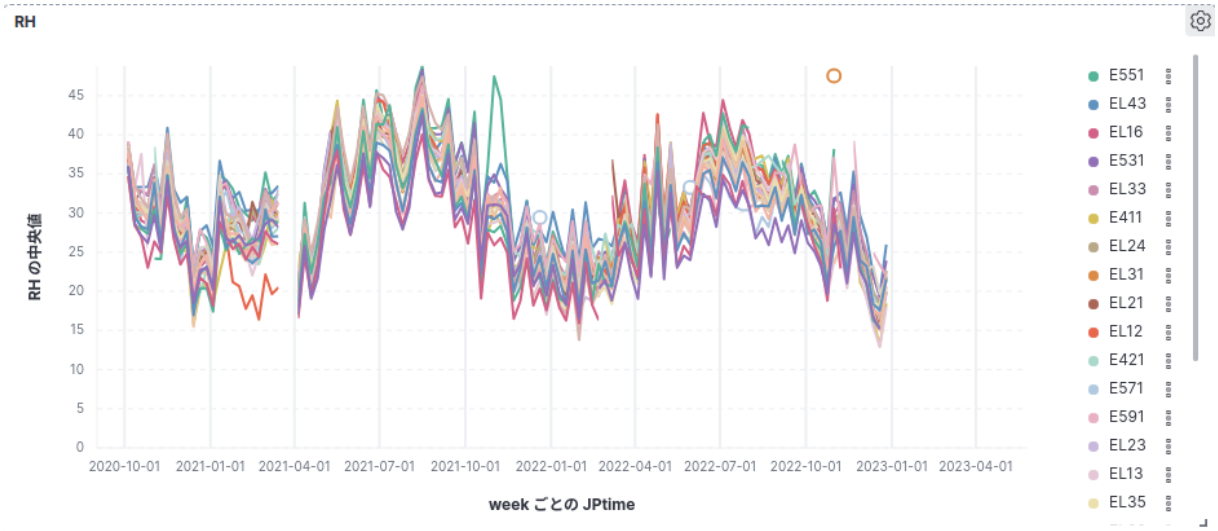

図 2: 2022_co2 の RH

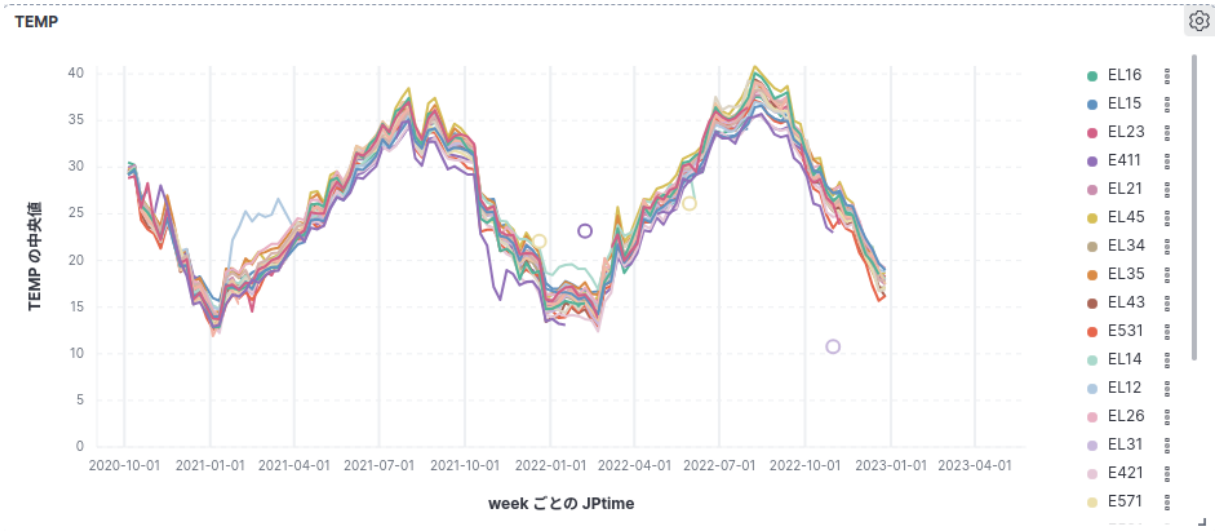

図 3: 2022_co2 の TEMP

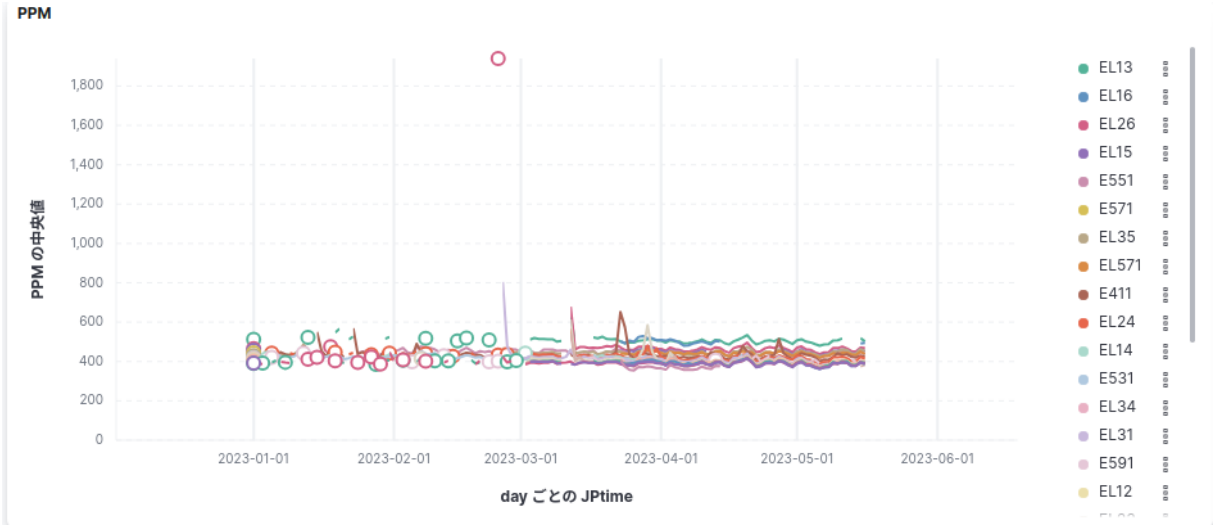

図 4: 2023_co2 の PPM

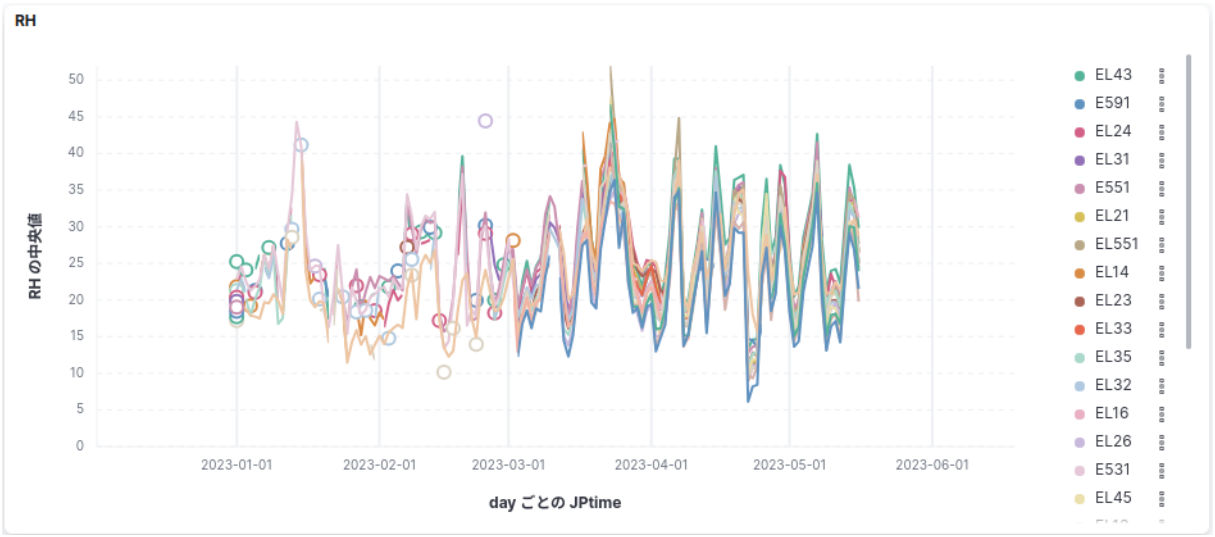

図 5: 2023_co2 の RH

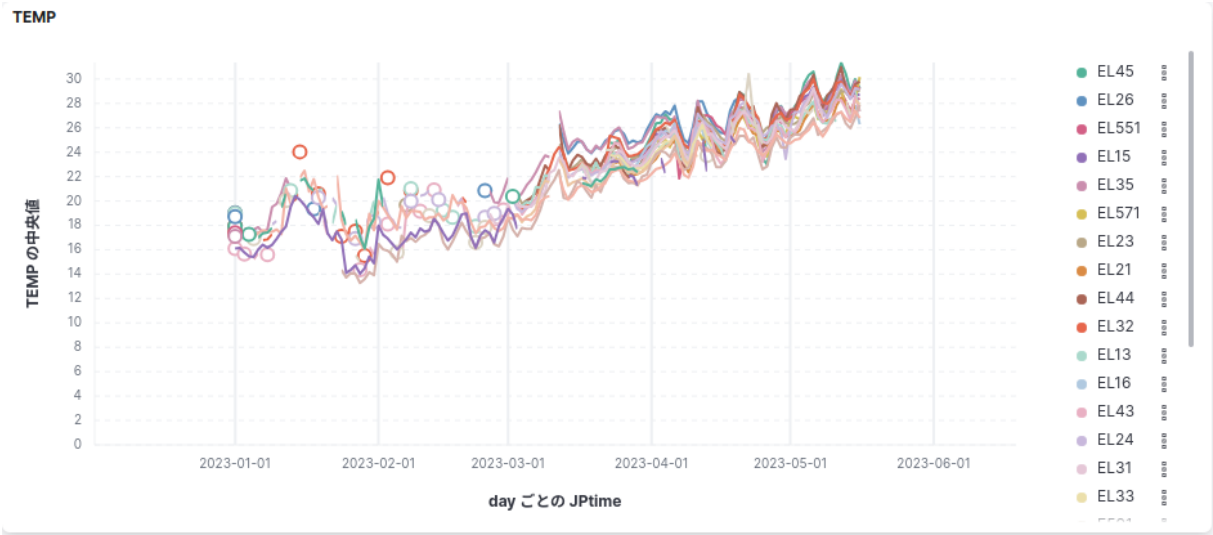

図 6: 2023_co2 の TEMP

4　まとめ

今回は, ElasticSearch サーバー間でのデータ移行と, その際に行われた重複データの削除方法, kibana による可視化結果について報告した.

参考文献

[1] Ferron H, ”ElasticDump ”, https://github.com/elasticsearch-dump/elasticsearch-dump, 参照 June 19,2023.