令和７年度厚生労働行政推進調査事業費補助金
（政策科学総合研究事業（政策科学推進研究事業））
「NDBのユーザビリティ向上を通じて
クラウド上でのデータ二次利用を推進するための研究」
分担研究報告書

利便性の高いクラウド環境の検討・提案

研究分担者

柏木 公一　国立健康危機管理研究機構・国立看護大学校・准教授
杉山 雄大　国立健康危機管理研究機構・国立国際医療研究所糖尿病情報センター

研究協力者

市瀬 雄一　国立健康危機管理研究機構・国際医療協力局グローバルヘルス政策研究センター
古野 孝志　国立健康危機管理研究機構・国際医療協力局グローバルヘルス政策研究センター

研究要旨
NDB(匿名医療保険等関連情報データベース)における HIC(医療・介護データ等解析基盤)では、Redshift・ローカル Postgres 等の解析環境が提供されている。一方、同環境においては、Redshift 永続テーブル作成権限の欠如、ローカル解析サーバのメモリ・ストレージ不足、外部参照データ取込の困難、計算コストの増大といった複合的課題が存在する。
本分担研究では、HIC 内で提供されるクラウドストレージ(Amazon S3)と列指向ファイル形式 Parquet、軽量分析エンジン DuckDB を組み合わせた分析基盤を構築し、その実用性を検証した。
2024 年 4 月から 2025 年 3 月の医科・DPC・調剤の全テーブル(1 年分、計 620GB)を Redshift から S3 上に Parquet として生成し、MD5 ハッシュの加算突合により 684 テーブルすべてについてデータの一致を確認した。加えて、全テーブルの各カラムについて、NULL 数、ユニーク値数、出現頻度上位 50 件、最大・最小・平均・標準偏差からなるデータサマリを生成した。
HIC の乙区分(メモリ 64GB)の環境において、本構成が大規模レセプトデータに対する実用的な処理性能を有することを確認した。ストレージ料金も 1 年分 620GB で月額約 2,100 円と低コストで運用でき、従来の Redshift 依存の分析環境におけるコスト増大・柔軟性の不足といった課題を解消できる可能性を示した。

A. 研究目的

本研究で対象とするNDB(匿名医療保険等関連情報データベース)は、日本の公的医療保険における診療報酬明細書(以下「レセプト」)および特定健診・特定保健指導等の情報を収集・格納した匿名化された大規模医療データベースであり、クラウド基盤上の医療・介護データ等解析基盤(HIC ： Healthcare Information Center)が提供されている。本分担研究は、厚生労働科学研究「NDBのユーザビリティ向上を通じてクラウド上でのデータ二次利用を推進するための研究」の一環として、HIC内の分析基盤の柔軟性およびコスト効率を改善するための技術的検証を担うものである。

既存のRedshift環境には、研究用途における柔軟なデータ処理を妨げる技術的困難が複数存在する。まず、研究者に対して永続的なテーブル作成権限が付与されていないため、段階的に中間成果物を生成しながらデータ処理を進めることができず、複雑な集計・加工処理を単一のクエリとして一括実行する必要がある。例えば、1) ある医療行為が含まれるレセプトの患者IDを取得し、2) この患者IDについて前後月の医療行為を取得しようとする場合、通常は、前段の1のクエリ結果を中間テーブルとして保存し、その後2以降の処理を行う。しかし提供されているRedshift環境では中間テーブルを作成することができず、毎回1の集計を再実行せざるを得ない（一時テーブルであっても、セッションごとに再作成が必要なため、セッションをまたぐ分析では同様の制約が残る)。

Redshiftのクエリ実行は計算資源の利用量に応じたコストが発生するため、大規模な試行錯誤を伴う探索的分析を行う際には費用も無視できない課題であった。さらに、既存のRedshift環境にはCSV等の外部データを直接取り込むためのインポート機能が整備されておらず、傷病名コードや薬剤コードの条件として数千件規模の参照データを利用する場合、条件としてIN句を用いるか、INSERT文やUNION ALLを連結した長大なSQLを作成する必要があった（図1)。このような方法は可読性・保守性に乏しく、分析作業の効率を低下させる要因となっていた。代替手段としてローカル環境にCSV形式でデータをダウンロードし、PostgreSQL等に取り込む方法も提供されている。HICは用途別に甲・乙・丙の3種類の解析サーバ環境を提供しており、利用可能なメモリ容量はそれぞれ128GB, 64GB, 32GB、ストレージ容量は16TB、3TB、1TBである(厚生労働省, 2026)。このため、甲以外の環境ではローカルストレージに大規模データを保持することが困難であり、メモリ制約からも大規模データ分析は現実的ではない。特に、糖尿病、高血圧症、脂質異常症等の生活習慣病のように、数千万人規模の患者を対象とする分析では、ローカル環境での処理は事実上不可能であった。

以上のように、既存の分析環境では、権限制約、参照データ利用の非効率性、ならびにローカル計算資源の不足が複合的に存在し、大規模レセプトデータを対象とした実用的な分析基盤の構築は困難な状況にあった。本研究は、これらの制約を解消する代替アーキテクチャの技術検証を目的とする。

B. 研究方法

本研究では、既存の分析環境における権限制約、計算コスト、およびローカル計算資源の制限といった課題を解決するため、クラウドストレージ、列指向ファイル形式 Parquet、および軽量分析エンジン DuckDB を組み合わせた新たな分析基盤を構築した。

1. Parquet について

Parquet は、Apache Software Foundation が開発するオープンソースの列指向ファイルフォーマットであり、大規模データに対する分析処理を主な用途として設計されている。CSV 等の行指向形式が 1 レコード分の値を行単位で連続的に格納するのに対し、Parquet は同一カラムの値をひとまとまりとして格納する列指向方式を採用している。

この構造には主に 3 つの利点がある。第一に、分析時に必要なカラムのみを選択的に読み込むことができ、不要なカラムに対する I/O が発生しない。第二に、同一カラム内では値の型および分布が類似するため、汎用の圧縮アルゴリズム(本研究では ZSTD を採用)を適用した際に高い圧縮率が得られ、ストレージ容量およびネットワーク転送量を削減できる。第三に、カラム単位の最小値・最大値等の統計値がメタデータとして格納されており、検索条件に該当しないデータブロックを読み飛ばす最適化(predicate pushdown)が可能である。レセプトデータのように、同一スキーマのレコードが多数存在し、かつ分析時には特定カラムのみが参照される用途においては、Parquet 形式の特性が効果的に作用する。

2. DuckDB について

DuckDB は、分析用途(OLAP ： OnLine Analytical Processing)に特化したオープンソースの組込型データベースエンジンである。サーバ構築や事前のデータベース構築を必要とせず、コマンドラインからの利用や、R や Python のスクリプトからも直接利用できる。実行エンジンは列指向かつベクトル化処理を採用しており、集計・結合等の分析クエリを高速に処理する。また、Parquet ファイルを外部テーブルとして SQL から直接クエリする機能をネイティブに備えており、httpfs 拡張を読み込むことで、ローカルディスク上のファイルだけでなく Amazon S3 等のクラウドストレージ上のファイルに対しても同一の構文で操作可能である。さらに、利用可能なメモリを超える大規模データを対象とする場合には、中間結果の一部をディスクに退避させながら処理を継続する機構(spill-to-disk)を備えており、限られたメモリ環境における大容量データの分析に適している。本研究では、DuckDB version 1.4.2 と httpfs エクステンションを解析サーバ上にインストールした。

3. 実装

まず、HIC で一時データ保存領域として利用可能な Amazon S3 に、Redshift 上のレセプトデータを Parquet 形式として保存した。この構成により、大容量データをローカルストレージに保持する必要がなくなり、解析サーバにおけるストレージ容量の制約を実質的に解消することが期待される。

次にデータの検証を行った。Redshift から分割ダウンロードを行った際には、正確にデータが取得できたかを検証する

仕組みが必要である。一般的には取得行数の確認が用いられるが、本研究では100万行単位でダウンロードを行っており、同一ブロックが重複ダウンロードされた場合に行数集計では、重複を検知できない場合がある。そこで、RedshiftとParquetファイルの両者でデータが一致していることを検証する仕組みとして、件数(行数)の突合に加え、ハッシュ値に基づく照合手法を考案し、実行可能性を検証した。

さらに、格納データの分析可能性を検証するため、全テーブルの各カラムに対して、ユニーク値の数、NULLの数、出現頻度が高い上位50件、最大値、最小値、平均値、標準偏差からなるデータサマリを算出した。

C. 研究結果

1. S3上のParquetファイルの生成

実行可能性の検証として、2024年4月から2025年3月の1年間の医科、DPC、調剤の全テーブル情報をRedshiftからS3上にParquetファイルとして格納した。

使用したS3ストレージは、「一時フォルダ格納領域」として割り当てられているworkを用いた。Redshiftから多量データをロードする際のローカルメモリ容量の制約、およびParquetファイルを適切なサイズに留める観点から、1ファイルあたり100万行単位でファイル分割を行った。実装したRスクリプトでは、追加ライブラリとしてRedshiftへのアクセスにPostgres、Parquetファイルの生成にarrowを使用した。
実行時間は1ヶ月分あたり平均5時間34分、12ヶ月合計で66時間53分であった。データ容量は1年分で620GB(内訳:医科399GB、DPC 25.5GB、調剤196GB)であった(Parquetファイルの圧縮アルゴリズムはZSTD、圧縮レベル1を採用した)。

2. RedshiftとParquetのデータ検証

Redshiftから分割ダウンロードしたParquetファイルが一致することを確認するために、検証用カラムとして、いずれのレセプトにも存在し、かつレセプト単位で一意な値を持つseq2_noを対象に、各行のMD5ハッシュ値(128ビット、16進32文字)を算出した。これをそのまま加算すると桁あふれを起こすため、本手法では32文字のうち上位4文字(16ビット相当、10進換算で0〜65,535の整数)を抽出し、テーブル内の全レコードにわたって64ビット整数として合計(SUM)した値を、Redshift側とDuckDB(Parquet読込)側の双方で算出して突合する方式とした。
1つのテーブルを複数の分割ファイルとして保存する場合にも対応できるよう、加算によって全体集計値を得られるハッシュ合成を採用している点が本手法の要点である。上記手法で検証した結果、684テーブルすべてについてデータの一致を確認できた。

3.DuckDBによるS3上のparquetファイルの操作

DuckDBからS3上のファイルを読み込むには、httpfs拡張のインストールとロードを行った後、AWS認証が必要である。HIC環境では、標準的な認証情報探索ルールであるcredential_chainを指定するだけでよい。

```
# 1回のみ
```

```
INSTALL httpfs;
# 以下はセッションごとに必要
LOAD httpfs;
CREATE SECRET (
TYPE s3,
PROVIDER credential_chain
);
```

DuckDB を使用した典型的なクエリは以下のようになる。

```
SELECT * FROM read_parquet(
'S3://[work フォル
ダ]/path/file.parquet)';

SELECT * FROM read_parquet(
'S3://[work フォル
ダ]/fy=2024/**/*.parquet')
```

パスの指定に*（アスタリスク）を使用できるため、年度や月を指定したクエリが指定できる。なお、DuckDB は、標準で複数ファイルを同時に処理するため、ファイルが分割されていることは高速化に寄与しやすい。

パスに/name=value/という形式(HIVE 形式)が含まれる場合 SQL 内で name の値を利用することができる。一般に年月単位でテーブルを JOIN する場合、次のクエリーのように S3 のパスとして ym=202404 が含まれれば、ym カラムとして SQL 内で値を利用できる。

```
SELECT
med_re.id5,
med_si.rcpt_cd,
ym
FROM
read_parquet('S3://[work]/med_si/*
*/ym=202404/*.parquet')  AS med_si
JOIN
read_parquet('S3://[work]/med_re/*
*/ym=202404/*.parquet') AS med_re
ON med_si.req2_no = med_re.req2_no
```

4. データサマリ生成

各カラムのデータサマリを DuckDB を用いて作成した。ID 関連などユニーク値が多いカラムについては、12 ヶ月分をまとめて処理することは解析サーバのメモリ不足によって実行できなかったため、1 ヶ月ごとに各カラムのユニーク値と件数を求めた後に 1 年分を集計した。集計したデータサマリはいったん JSON 形式で保存した後、テーブル別・カラム別でアクセス可能な HTML 形式に加工した。実行時間は以下のとおりである。

1) 月ごとに各カラムのユニーク値と件数を集計: 6 時間 17 分 53 秒

2) 月別集計をまとめて年集計へ: 6 時間 16 分 55 秒

3) 各カラムの固有値の数、最小値、最大値、NULL の数、平均、標準偏差の集計: 31 分 34 秒

4) 各カラムの出現頻度上位 50 位を集計: 31 分 24 秒

5) 全集計データをまとめる: 32 秒

D. 考察

S3 上の Parquet ファイルは、HIC の乙スペック(メモリ 64GB)において問題なく処理できた。これによって、解析サーバのストレージ容量の制約や、中間テーブルを作成できないといった既存環境の問題が実質的に解消される。

Parquet 形式は列指向でデータが格納され、かつ列単位で圧縮が施されるため、クエリ実行時には必要なカラムのみを読み出すことができる。この特性により、ネットワーク転送量を抑制しつつ効率的なデータ処理が可能となる。本研究で使用した HIC 内の EC2 インスタンスにおいて、S3 との通信スループットは単一接続で約 100〜500 Mbps の範囲にあり、複数ファイルへの並列アクセスを組み合わせることで、実効で 1 Gbps を超えるデータ読込性能が得られ、大規模データに対する実用的なクエリ実行性能を確保できた(なお、これらの数値は本研究環境における実測値であり、EC2 のインスタンスタイプやネットワーク構成に依存する)。

Redshift から S3 上への Parquet ファイル生成には、医科・DPC・調剤の 1 年分で約 67 時間を要したが、並列数を 4 まで増やす限りにおいては処理速度が線形に向上することが確認できた。つまり、同時実行数を 4 とした場合は約 17 時間で生成できると考えられる。また、生成した Parquet ファイルは、全利用者で共通に使用可能な性質のものであり、あらかじめ作成したものを読み取り専用として共有する運用も可能と考えられる。この場合は、Redshift は使用せず、解析用 PC を準備し、620GB×年数分の S3 ストレージを共有できればよい。

料金について、Amazon S3 (Standard クラス)の東京リージョンにおけるストレージ料金は GB あたり月額約 0.024 米ドル(本報告時点の為替換算で約 3.5 円/GB/月)と比較的低コストであり、1 年分の医科・DPC・調剤の全データ 620GB を保持しても月額約 14 米ドル(約 2,100 円)である。また、同一リージョン内における S3 と EC2 間のデータ通信は追加料金が発生しないため、大規模データを継続的に利用する研究用途において費用効率の高い構成となる(別途、S3 API 呼出しに対するリクエスト課金が発生するが、GET リクエスト 10 万件あたり約 6 円程度と軽微である)。さらに、S3 は単一アカウントのストレージ総量に事実上の上限がないため、数十億レコード規模のレセプトデータを対象とした長期的な分析基盤としても持続的に利用可能である。

本研究で使用した Redshift データベースと EC2 分析サーバの単位時間あたりの利用料金を比較すると、同等の稼働時間において Redshift のほうがおおむね 10 倍程度高い試算となった(ノードタイプ・インスタンスタイプに依存するため、環境による差異はある)。この点からも、EC2 分析サーバの計算資源のみで分析可能な S3 上の Parquet ファイルを用いる構成は、費用面での優位性が高いと考えられる。

一方、DuckDB と Parquet を用いた分析手法には限界もある。大量データの分析においては解析サーバのメモリが最も強い制約となり、本分析手法では、メモリ不足を回避するために 1 ヶ月ごとに処理した中間結果を後段で統合するといった追加の手間がかかっている。その点、Redshift では潤沢なメモリを前提とした解析が可能であり、たとえば 64 文字で提供される ID を INT32 に置き換えるような処理は、Redshift で実行する方がはるかに手間がかからず高速である。

E. 結論

HIC の実環境において、684 テーブルに対するデータ一致検証、および全テーブル・全カラムのデータサマリ生成という、大規模データを対象とした代表的な集計処理を実施し、本構成が研究用途において実用的な処理性能を有することを確認した。また、本研究で採用した DuckDB はメモリ効率に優れた列指向分析エンジンであり、64GB のメモリ環境においても大規模データに対する実用的なクエリ処理が可能であることが確認された。

以上により、従来の Redshift 依存の分析環境におけるコスト増大や柔軟性の不足といった課題を解消し、低コストかつ再現性の高い大規模レセプトデータ分析基盤を実現できたと考える。

2026 年 3 月現在、DuckDB は HIC 環境の標準ツールとなっておらず、別途申請が必要である。コマンドライン版(CLI)、Python、R のライブラリについては標準ツールとしての採用が望まれる。

F. 健康危険情報

特記すべき事項はない。

G. 研究発表

なし

H. 知的財産権の出願・登録状況

なし

引用文献

厚生労働省 (2026). 匿名医療保険等関連情報データベース(NDB)の第三者提供よくあるご質問(FAQ) 2026 年 3 月版. https://www.mhlw.go.jp/content/12400000/001679224.pdf

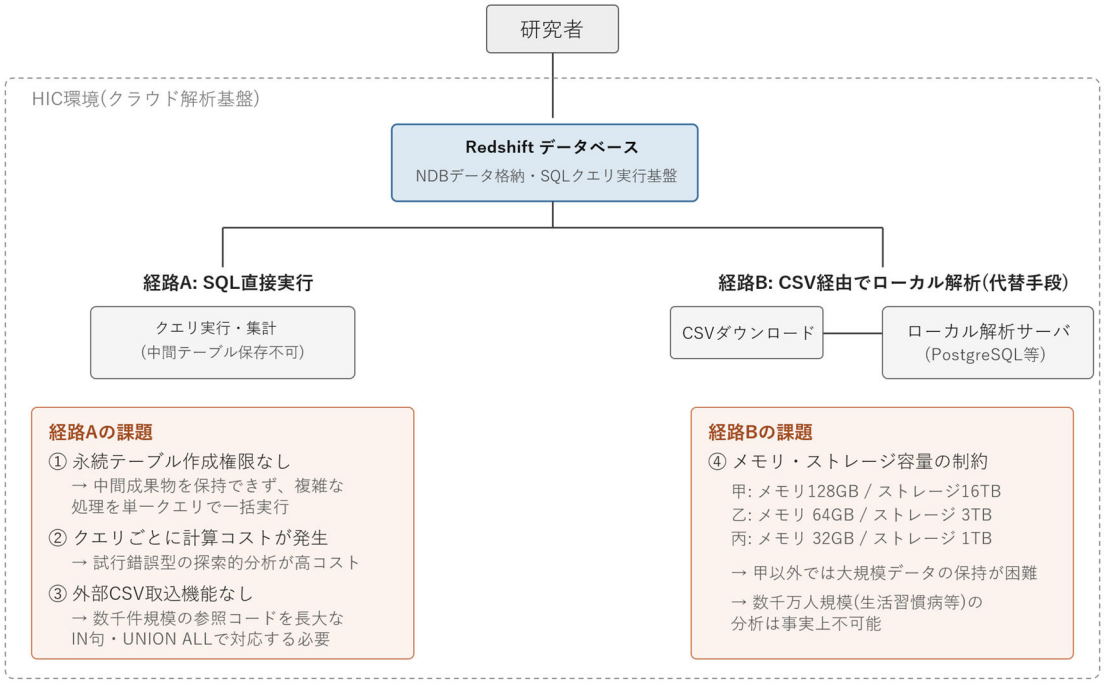

図 1 既存環境におけるレセプトデータ解析の構成と課題

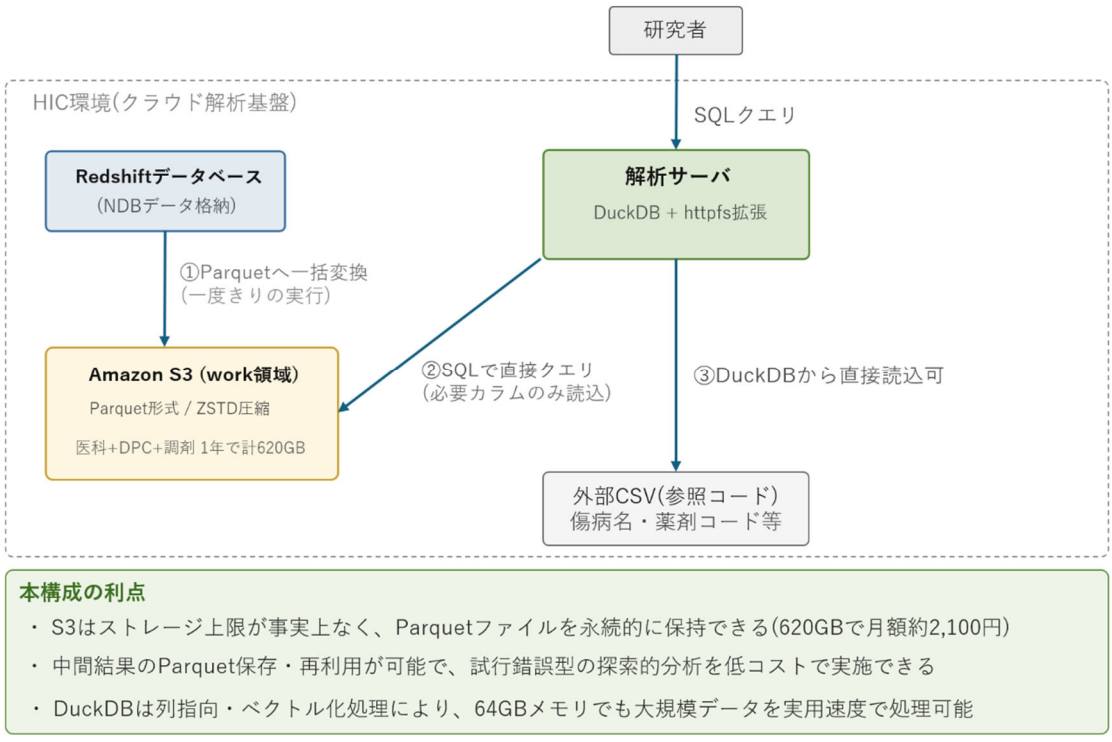

図 2 提案アーキテクチャ(S3 + Parquet + DuckDB)

8