D-034

# 放射線モニタリングデータベース実現に伴う GeoSPARQL の問い合わせ性能評価と改良

## Performance evaluations of GeoSPARQL software for the implementation of the radiation monitoring database

小島　功†　播田　一光†‡　高杉　悠平†‡　田中　良夫†　的野　晃整†　中村　章人†
Isao Kojima　Ikkou Harita　Yuuhei Takasugi　Yoshio Tanaka　Akiyoshi Matono　Akihito Nakamura

### 1. まえがき

産業技術総合研究所では、原子力規制庁の事業における日本原子力研究開発機構(原研)からの委託研究開発として、福島原発事故に関わる環境モニタリング情報の国際標準に基づいたデータベース化を行った。これは GEO Grid[(1)] プロジェクトの応用で、OGC(Open Geospatial Consortium)のセンサ標準に沿った API[(2)]を提供することで他の地理空間データとの連携を容易にするものである。

また、政府統計などのオープンデータは Linked Open Data(LOD [(3)])化されて提供される方針であることから、このデータベースを同時に Linked Open Data 化して他の統計データベースとの連携を可能としている。放射線モニタリングのデータは緯度・経度に基づく地理空間情報であるため、データベースをジオメトリ操作が可能な問い合わせ言語 GeoSPARQL[(4)]のエンドポイントとして提供すると同時に、この有効性を高めるために人口統計と放射線データベースを連携させる応用を構築している[(5)]。

本稿では、このデータベースおよび応用の構築の過程で GeoSPARQL の検索性能の問題に直面したので、データベースの実装を含めた性能評価を行うと同時に簡単な改良を行ったので報告する。

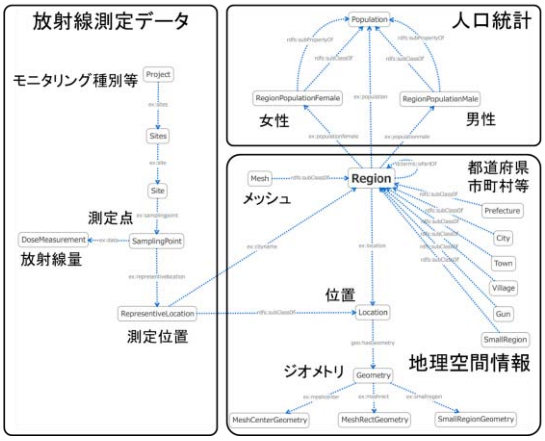
### 2 放射線モニタリングデータベースと応用の構築

#### 2.1 放射線モニタリングデータベースの構築

LOD 化された放射線モニタリングデータベースは、原研の整備提供する CSV データ[(6)]を RDF 化して GeoSPARQL のエンドポイントとして構築したもので、航空機モニタリング(第一次~第七次)のデータを対象とした。航空機モニタリングはある地域を飛行して測定したデータであり、緯度経度に基づいて測定データが提供されている。スキーマの構造は原研が検討中の XML スキーマに基づいて、RDFS(RDF Schema)により構築した。図 1 にその構造を示す。左側の部分が放射線の情報を示し、測定位置 RepresentativeLocation は geo:hasGeometry を有する Location のサブクラス、ジオメトリ以下の情報が GeoSPARQL でジオメトリ演算が可能である。

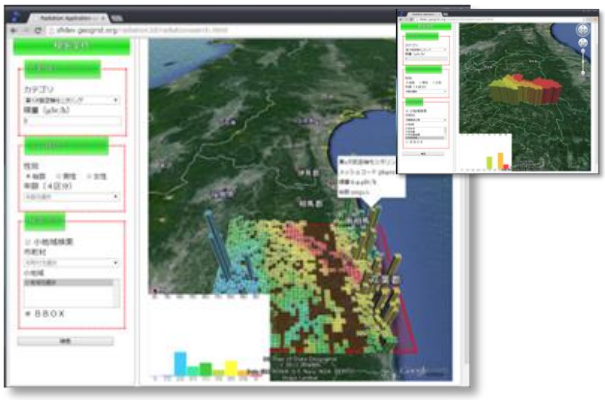
#### 2.2 応用の構築

このデータベースの有効性を検証するために応用を構築した。これは政府統計で提供される人口統計と放射線のデータベースとを連携させて、ある地域全体で集団的に受ける放射線の総量を計算するものである（集団線量）。このために平成 22 年度国勢調査の人口統計を同じく RDFS で設計してデータベース化した（図 1　の人口統計部分）。これは 3 次メッシュ単位の人口と小地域（字や町名など）単位での人口が提供されているので、これらを束ねる Region として地理空間オブジェクトを格納している。放射線と人口の地理空間情報を共に Location として結び付け、両者のデータが連携できるようになっている。




(図 1　全体のスキーマ構造)

構築した応用は、これらの 2 つのデータベースを連携させるもので、A) ある検索範囲の地域（Box で指定）の人口と線量の関係、B) ある小地域(プルダウンで指定)の人口と線量の関係をそれぞれ得るもので、GeoSPARQL の問合せにより実現されている。これを Google Earth に基づく GUI で操作できるようにしたのが図 2 である。



(図 2　Google Earth を用いた GUI とその表示例)

†（独）産業技術総合研究所　情報技術研究部門
‡（株）日本アドバンス・テクノロジー

例えば A)の場合、検索範囲に含まれる 1km メッシュのそれぞれに含まれる放射線測定データの平均と人口を掛け合わせてメッシュごとの数値を計算し、表示すると同時に放射線量の範囲に対応して集計してグラフ化する。

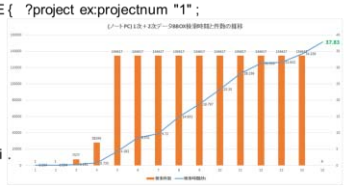
現在、このデータベースおよび応用は産総研から試験公開中である(Radiation-LOD: http://sfidev.geogrid.org/)。データの総量は第一次航空機モニタリング＋人口統計で 6,446,148 トリプルである。地理空間データとしては測定点が 134,738、ポリゴン数が 149,050 で、これは論文[7] で示された実世界(real-world)ベンチマークデータの合計に対し、第一次だけでポイント数が 5 倍、ポリゴン数が 3 倍以上の規模であるが、評価のために作られた合成(synthesized)データセットに比べると少ない。

## 3. GeoSPARQL とその性能

### 3.1 応用の性能

このデータベースに対し、uSeekM/Sesame[8]に基づいた GeoSPARQL に基づいて応用を実装したところ、この検索性能は低いことが判明した。例えば、応用で要求される問合せは以下の図 3 のようなものであり、応答時間は 37 秒を超える。これは索引の設定等を工夫した状態であり実用的に問題がある。

```
1.  SELECT (count(*) as ?count) WHERE {  ?project ex:projectnum "1" ;
2.     ex:sites ?bn1 .
3.     ?bn1 ex:Site ?siteuri.
4.     ?siteuri ex:samplingpoint ?bn2.
5.     ?bn2 ex:SamplingPoint ?spuri.
6.     ?spuri ex:data ?dm .
7.     ?dm ex:radiation ?raddata .
8.     ?spuri ex:representivelocation ?rluri .
9.     ?rluri rdfs:subClassOf ?location .
10.    ?location geo:hasGeometry ?geometry .
11.    ?geometry ex:meshrect ?meshgeo .
12.    ?meshgeo geo:asWKT ?coordinate .
13.    ?siteuri ex:code ?sitecode .
14.    FILTER(  xsd:double(?raddata) >= xsd:double(0)
15.       &&     geof:rcc8ntpp(?coordinate, "POLYGON((140.949087287644
       37.64730011144615,140.95876004713537
       37.64730011144615,140.95876004713537 37.641227789330685,
       140.949087287644 37.641227789330685,140.949087287644 37.64730011144615))")
16.    )
17.  }
```




(図 3　応用で使われている GeoSPARQL の例：
右のグラフは途中行までの実行時間と中間データの数)

これには以下のような理由が考えられた。

1)　スキーマに階層関係がある状況(府県―市町村―町―字、１次～３次メッシュなど)で地理空間情報が階層の最少単位（字や３次メッシュ）に対応（hasGeometry）するため、問い合わせにおいて階層を展開する必要があって変数が増える点と、問い合わせで扱う地理空間オブジェクトの数が増える。

2)　人口と放射線のデータは地理空間情報に基づいて連携されるため、GeoSPARQL のジオメトリ演算性能に依存するが、多くはこのモジュールは元々の SPARQL に対して拡張されたシステムである。

このため、実装の比較も含めて評価を行い、その結果に応じた性能向上対策を取ることとした

### 3.2 GeoSPARQL とその実装。

GeoSPARQL とは、RDF に対する標準問い合わせ言語 SPARQL に対してといったジオメトリ演算を支援する拡張で、OGC で定義された標準である。本稿では、構築したアプリケーションに応じた形の性能評価を、以下の 3 種類の実装について行った。

1) uSeekM[8]：uSeekM はオープンソースで著名な RDF データベースである Sesame の java インターフェイスを用いて実現されており、地理空間処理については PostGIS を併用している。PostGIS で提供される Rtree over GiST と呼ばれる索引構造を用いて地理空間処理を行っている。

2) parliament[9]：地理空間データ型に対して Rtree 索引に基づく内部実装を有している点が 1)と異なっている。

3) strabon[10]:　同じく Sesame を併用した GeoSPARQL 処理系で、地理空間演算処理には PostGIS を併用する実装になっており、索引も同じく Rtree over GiST である。

これらはいずれもオープンソースソフトウェアである。他に商用の高速の SPARQL プロセッサとして virtuoso などが知られているが、ここで必要とする空間演算を十分支援していない、GeoSPARQL の支援する WKT 型を支援していない、といった点が十分でないので評価していない。

### 3.3 問い合わせ評価

問い合わせは、先の応用から以下のような場合をいくつか例として評価した。

A)　矩形に対する空間演算（BBox）：検索領域の矩形のサイズの異なるものをいくつか与える。

B)　小地域の行政領域（ポリゴンが複雑な場合）による検索：行政区域の大きさが異なるものをいくつか与える。

それぞれの問合せ表現は図 4，5 のようなものである。

```
SELECT distinct (count(*) as ?count) WHERE {
  ?project ex:projectnum "1" ;
  ex:sites ?bn1 .
  ?bn1 ex:Site ?siteuri .
  ?siteuri ex:samplingpoint ?bn2 .
  ?bn2 ex:SamplingPoint ?spuri .
  ?spuri ex:data ?dm .
  ?dm ex:radiation ?raddata .
  ?spuri ex:representivelocation ?rluri .
  ?rluri rdfs:subClassOf ?location .
  ?location geo:hasGeometry ?geometry .
  ?geometry ex:meshrect ?meshgeo .
  ?meshgeo geo:asWKT ?tempcoordinate .
  FILTER(
   geof:rcc8ntpp(?tempcoordinate, "POLYGON((140.83947975874557
37.73193100828603,141.03562009458093 37.73193100828603,
141.03562009458093 37.513364968325355,140.83947975874557
37.513364968325355,140.83947975874557 37.73193100828603))"^^sf:wktLiteral)
   &&
   xsd:double(?raddata) >= xsd:double(0)
  )
  }
```

(図 4　BBox 検索の例:
矩形を記述する 4 点のポリゴンで検索)

```
SELECT (count(*) as ?count) WHERE {
  <http://sfidev.geogrid.org/075470290/> ex:location ?loc .
  ?loc geo:hasGeometry ?geometry .  ?geometry ex:smallregion ?srgeo .
  ?srgeo geo:asWKT ?srgeocoordinate .
                    (略)
  ?cityuri ex:regionname ?label .
  ?rluri rdfs:subClassOf ?location .
  ?location geo:hasGeometry ?radgeometry .
  ?radgeometry ex:meshrect ?meshgeo .
  ?meshgeo geo:asWKT ?coordinate .
  FILTER(
   ?cityuri = <http://sfidev.geogrid.org/07547/>      &&
   (geof:rcc8po(?coordinate, ?srgeocoordinate)      ||
     geof:rcc8ntpp(?coordinate, ?srgeocoordinate))
  ) }
```

(図 5　小地域検索の例：市町村コードに対応する
ポリゴンを検索し、この領域で検索する)

### 3.3 性能評価

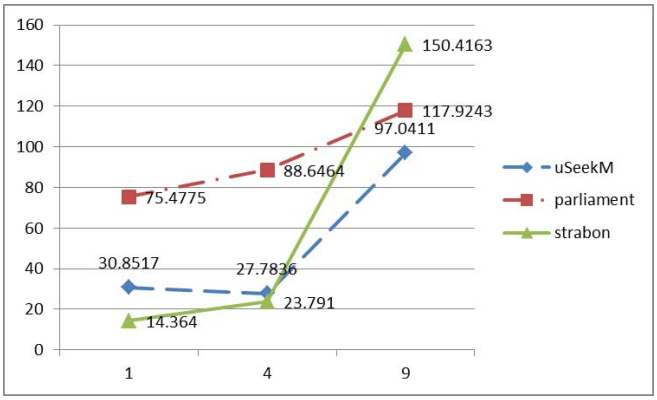
実験は後述する 2 種類の仮想マシン(VM)環境で行い、それぞれの問合せに対して 10 回実行した平均値を取得した。２つの VM 環境は大きくはコア数とメモリの量に相違がある。

| VM | 環境1 | 環境2 |
|---|---|---|
| OS | Scientifi Linux 6.5 64bit | Scientific Linux 6.4 64bit |
| CPU | Core2 4コア Q9100@2.2Ghz | 2.4Ghz x 7コア, 2.4Ghz |
| メモリ | 2GB | 20GB |
| HDD | 40GB | 443GB |

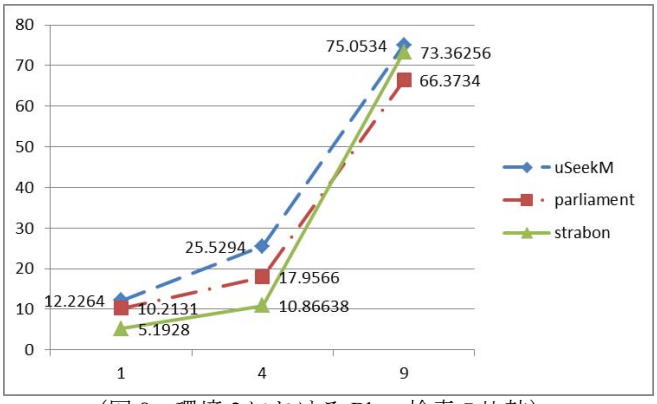
(図 6　評価環境の概要)

#### 3.3.1 Bbox 評価：

Bbox のサイズが異なる 3 種類の問合せをそれぞれ実行した。1 は約 10km 四方、4 は約 20km 四方、9 は約 30km 四方のサイズの Bbox で検索を行った。環境 1,2 での結果をそれぞれ図 7、図 8 に示す。




（図 7　　環境 1 における BboX 検索の比較）




（図 8　環境 2 における Bbox 検索の比較）

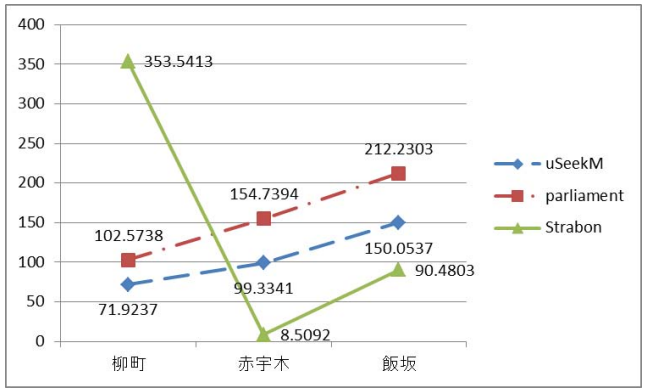
おおむね検索範囲＝扱うデータ数に応じて応答性能が悪くなっているが、図 8 の比較的性能の高い VM での評価を考えると、概して parliament はメモリなどハードの仕様への依存性が高いと推測できる。

#### 3.3.2 小地域評価：

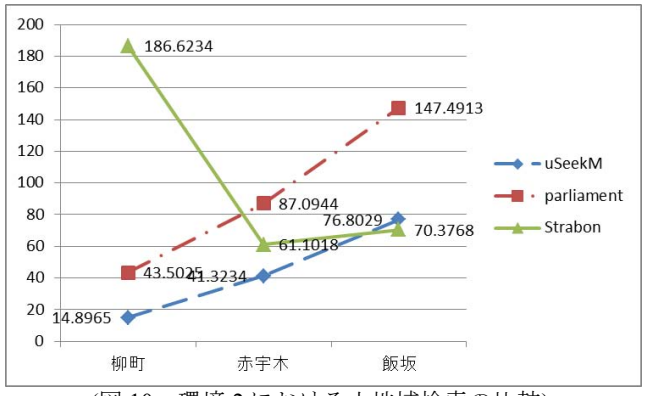
面積および領域の複雑さを考慮して、以下の 3 地域を例として実施した。なお外形はおおむね以下の通りであり、おおむねこの順で面積が増え形状が複雑化している。

1. 福島市柳町（約 100m x 200m）
2. 浪江町大字赤宇木（約 3km x 5km）
3. 福島市飯坂町（約 5km x 5km）

これも同様に図 9、図 10 に結果を示す。柳町における strabon の性能悪化は説明できないが、環境を変えかつ試行を繰り返しても同一であったため掲載した。




(図 9　環境 1 における小地域検索の比較)




(図 10　環境 2 における小地域検索の比較)

### 3.4 考察

空間検索については、単純な Bbox 検索よりもポリゴンを用いた検索の方が遅い。しかも、空間検索索引の実装について，parliament は独自の Rtree 索引、strabon と uSeekM が同じ PostGIS の Rtree over GiST であることは後者 2 つが多少類似の傾向をたどっていることから判別できる。

特に、小地域検索など複雑な検索については PostGIS の索引実装が有利なようで、このこととメモリが大きな場合の Bbox 検索等で十分な性能を見せている parliament が独自の Rtree 実装であることを裏付ける。

なお論文 [7] に strabon の関係者による parliament や uSeekM との比較が存在する。データセットのサイズや検索で用いる演算子の種類などが異なるため直接比較はできないが、Geospatial Selection の項目でほぼ同等の傾向が示されており、本論文の評価についてもおおむね妥当なものと考えられる。

## 4. 性能改善の考案と実装、評価

以上のように、本応用の問合せ処理において実装の差に起因する大幅な改良は期待できないことから、問い合わせ表現も含めて性能の改善を行った。

### 4.1. 部分問合せによる検索範囲の絞りこみ

問合せ処理の中で扱うトリプルの数を減少させるために、高次のメッシュ情報を用いたおおまかな絞り込みを

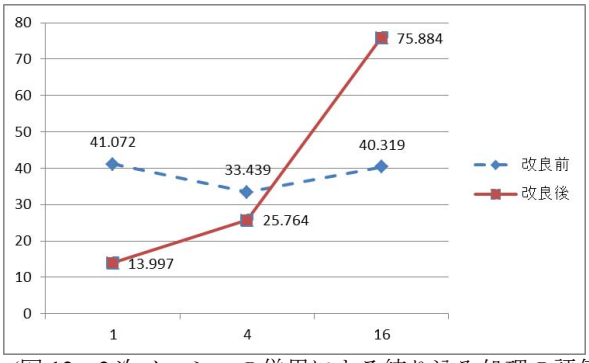
検討した。矩形で与えられる検索範囲に対し、直接 3 次メッシュの段階で検索するのではなく、より高次のメッシュコードを検索してこれを使って事前に絞り込むものである。福島県程度の情報では 1 次メッシュ(約 80km 四方)では大きすぎるので、2 次メッシュ(約 10km 四方)の情報を用い、矩形が入力されたらいったん矩形を含む 2 次メッシュコードを検索して絞り込み、これに対して従来通りの検索を行った。問い合わせ例を図 11 に示す。

```
SELECT distinct (count(*) as ?count) WHERE {
  ?project ex:projectnum "1" ;
  ex:sites ?bn1 .
  ?bn1 ex:Site ?siteuri .
  {
   select ?siteuri where{
    ?s ex:code ?mesh2code;
    dcterms:isPartOf ?meshuri.
    ?meshuri ex:code ?mesh3code.
    ?siteuri ex:code ?mesh3code.
    FILTER(
     ?mesh2code = "564027" || ?mesh2code = "564017" || ?mesh2code = "564120" || ?mesh2code = "564110"
    )
   }
  }
  ?siteuri ex:samplingpoint ?bn2 .
  ?bn2 ex:SamplingPoint ?spuri .
  ?spuri ex:data ?dm .
  ?dm ex:radiation ?raddata .
  ?spuri ex:representivelocation ?rluri .
  ?rluri rdfs:subClassOf ?location .
  ?location geo:hasGeometry ?geometry .
  ?geometry ex:meshrect ?meshgeo .
  ?meshgeo geo:asWKT ?tempcoordinate .
  FILTER(
    geof:rcc8ntpp(?tempcoordinate, "POLYGON((140.917841881177796 37.52424229646267,141.042200056777637.52424229646267,
141.042200056777637.464073210720 12,140.917841881177796 37.464073210720 12,140.917841881177796
37.52424229646267))"^^sf:wktLiteral)
    &&
    xsd:double(?raddata) >= xsd:double(0)
  )
  ?siteuri ex:code ?sitecode ;
  ex:samplingpoint ?bn3 .
  ?bn3 ex:SamplingPoint ?spuri2 .
  ?spuri2 ex:data ?dm2 .
  ?dm2 ex:radiation ?raddata2 .
  ?spuri2 ex:representivelocation ?rluri2 .
  ?rluri2 rdfs:subClassOf ?location2 .
  ?location2 geo:hasGeometry ?geometry2 .
  ?geometry2 ex:meshrect ?meshgeo2 .
  ?meshgeo2 geo:asWKT ?coordinate .
} ORDER BY ?sitecode
```

(図 11　2 次メッシュによる絞り込み問い合わせの例：この例では、検索矩形 BBox が 2 次メッシュ 4 つにまたがっていることが GUI 側で検出されており、そのメッシュコード(mesh2code)でデータを絞り込む部分問合せを付加している。(太字部分上部))

## 4.2　評価及び考察

同様の状況で実験評価を行った。評価環境は VM の環境 2（大）、基盤システムは uSeekM である。検索矩形と 2 次メッシュのオーバーラップについて、2 次メッシュ 1 つ、4 つ、16 とのそれぞれについて、評価を行った。結果を図 12 に示す。




(図 12　2 次メッシュの併用による絞り込み処理の評価　横軸は 2 次メッシュの個数、単位は秒)

図の通り、矩形範囲が 2 次メッシュ 1 個～数個に跨る程度の検索では改善が見られ、特に矩形範囲が 2 次メッシュ 1 個に含まれる場合には 3 倍程度の性能向上を得た。一方、2 次メッシュの数が増えると性能が落ち、広範囲の検索では逆に遅くなることが判明した。2 次メッシュの大きさは一つ 10km 四方で、この程度の領域の検索では確実に高速化されることから実装に取り入れた。

## 5. まとめ

放射線データベースの Linked Open Data 化を目標とした GeoSPARQL の実装について、その検索性能と簡単な改良方法について検討・評価した。一定の性能改善は得たもののなお十分な性能が得られていない点と、現時点においては検討や評価が十分でない点もあり、引き続き検討等を進めて改善を行い、実用性を高める予定である。

## 謝辞

本論文は、昨年度の原子力規制庁の事業で原研からの委託による「平成 25 年度東京電力福島第一原子力発電所事故による環境モニタリング等データベースの構築」での産総研の成果に対して評価及び改良を行ったものであり、同事業の機会を頂いた原子力規制庁および原研各位に深謝します。なお本研究は一部科研費基盤 A[24240015]による。

## 参考文献

(1) GEO Grid: http://www.geogrid.org
(2) OGC Sensor Observation Service: http://www.opengeospatial.org/standards/sos
(3) Linked Open Data: http://linkeddata.org
(4) OGC GeoSPARQL: http://www.opengeospatial.org/standards/geosparql
(5) I.Kojima et al. “Implementation of the Fukushima radiation LOD framework”, Linking Geospatial Data Workshop, 2014.03.
(6) 環境モニタリングデータベース：http://radioactivity.nsr.go.jp/ja/
(7) G.Garbis et al. “ Geographica: A Benchmark for Geospatial RDF Stores”, ISWC 2013. http://www.strabon.di.uoa.gr/files/Geographica.pdf
(8) uSeekM: https://dev.opensahara.com/projects/useekm
(9) parliament: http://parliament.semwebcentral.org/
(10) strabon: http://www.strabon.di.uoa.gr/