Getting practical with GeoSPARQL and Apache Jena

Simon Bin, Claus Stadler, Lorenz Bühmann,
Michael Martin

GeoLD2024 @ ESWC2024

Supported by

Federal Ministry for Economic Affairs and Climate Action
GA No. 01MK21007A

Federal Ministry for Digital and Transport
GA No. 9F2266A

Our work group / institute

InfAI e.V. **Leipzig, Germany**

- Efficient Technology integration,
  head Micha Martin
- ~12 people in our group
- 100% project funded, typically German government, rarely EU
- other groups at InfAI, e.g. DBpedia

Open source Semantic Web framework for Java

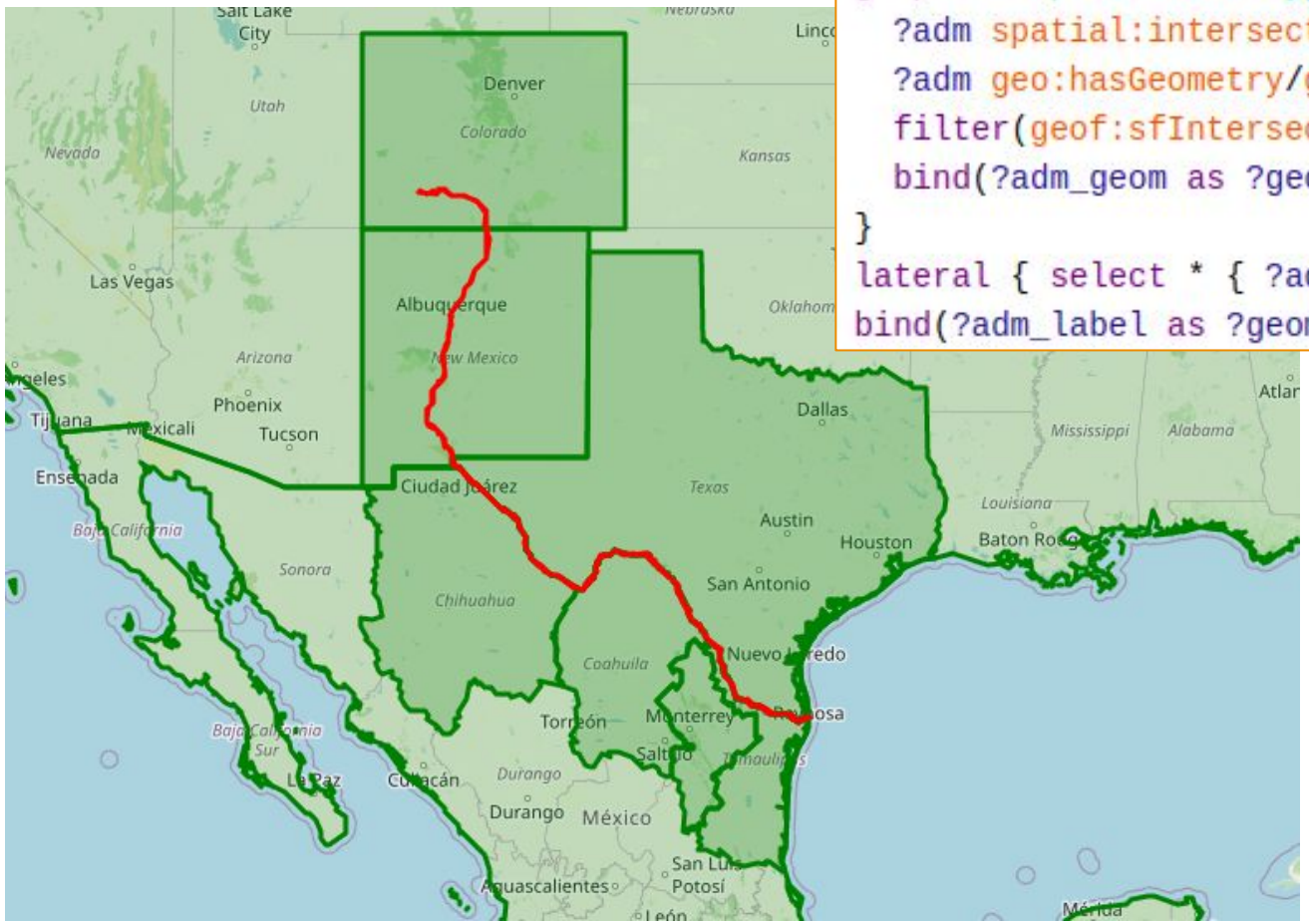
GeoSPARQL

Open Geospatial Consortium

- 2011: Upcoming OGC standard **GeoSPARQL** attempts to unify data access for the geospatial Semantic Web (Semantic Web Journal)
- 2021: GeoSPARQL 1.1 presentation at GeoLD2021 (official publication: 2024)

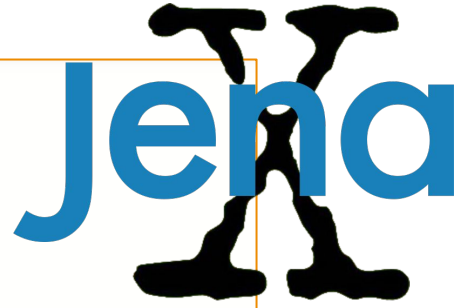
- Ontology, data types, query functions

Combining geodata in the graph




```
graph <https://data.coypu.org/administrative-regions/> {
  ?adm spatial:intersectBoxGeom(?flood_geom) .
  ?adm geo:hasGeometry/geo:asWKT ?adm_geom_lit_ .
  filter(geof:sfIntersects(?flood_geom, ?adm_geom_lit_))
  bind(?adm_geom as ?geom) bind("green" as ?geomColor)
}
lateral { select * { ?adm rdfs:label ?adm_label } limit 1 }
bind(?adm_label as ?geomTooltip) bind(?adm as ?geomLabel)
```

Mapping data using JSONL + GeoSPARQL




```
CONSTRUCT {
  ?country
    geo:hasGeometry ?geometry_node .
  ?geometry_node a geo:Geometry ;
    geo:asWKT ?geometry .
}
WHERE
{
  {
    SELECT
    ?country
    ?geometry_node
    ?geometry
    {
      <env:INPUT> url:textLines ?text .
      BIND(STRDT(?text, xsd:json) AS ?item)

      BIND(json:path(?item, "$.properties") AS ?properties)

      BIND(json:path(?properties, "$.ISO_CODE") AS ?iso3)
      BIND(URI(concat("https://data.coypu.org/country/", ?iso3)) AS ?country)
      BIND(URI(concat(str(?country), "/geometry/boundary")) AS ?geometry_node)

      BIND(json:path(?item, "$.geometry") AS ?geometry_)
      BIND(spatialF:transformDatatype(STRDT(str(?geometry_), geo:geoJSONLiteral), geo:wktLiteral) AS ?geometry)
```

Overview

- Overview of new functions in GeoSPARQL 1.1, and current status in Apache Jena
- Extension modules for Jena ARQ (query engine)
- Extension module for Jena Fuseki (SPARQL server)
- Patches to Jena
- Plan to contribute parts of the extensions directly to Jena

Functions implementation overview

Functions

GeoSPARQL 1.1	J	JX
geof:metricDistance	1	
geof:metricBuffer		
geof:concaveHull		
geof:boundingCircle		
geof:centroid		✓
geof:dimension	✓	
geof:coordinateDimension	✓	
geof:spatialDimension	✓	
geof:geometryType		
geof:is3D		
geof:isEmpty	✓	
geof:isMeasured		
geof:isSimple	✓	

Functions

GeoSPARQL 1.1	J	JX
geof:transform	2	
geof:asWKT geof:asGML	3	
geof:asGeoJSON	✓ ↑	
geof:asKML geof:asDGGS		
geof:$^{(m)}$length		✓
geof:$^{(m)}$perimeter		✓
geof:$^{(m)}$area		✓
geof:geometryN		4
geof:numGeometries		4
geof:max{X,Y,Z}		
geof:min{X,Y,Z}		

Non-standard functions reference:

1 spatialF:distance*
2 spatialF:transformSRS
3 spatialF:transformDatatype
4 spatial:st_dump*

Functions implementation overview

Aggregate functions

GeoSPARQL 1.1	J	JX
geof:aggBoundingBox		5
geof:aggBoundingCircle		
geof:aggCentroid		5
geof:aggConcaveHull		
geof:aggConvexHull		5
geof:aggUnion		✓

Data types

GeoSPARQL 1.1	J	JX
geo:geoJSONLiteral		✓
geo:kmlLiteral		
geo:dggsLiteral		

Non-standard functions reference:

5 geof:collect*

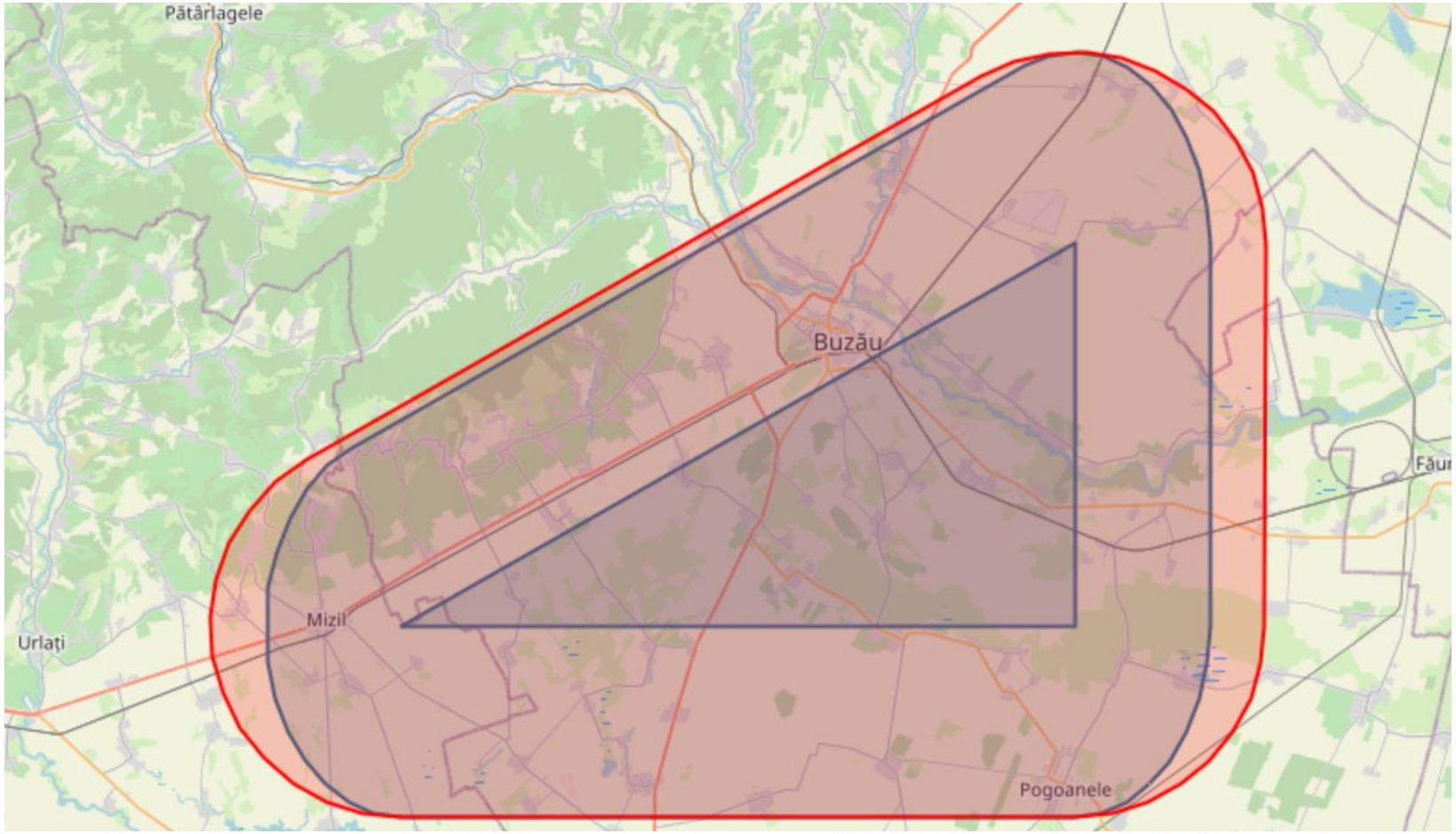
Additional non-standard functions

- create new objects: `spatialF:convertLatLon`[J], `geof:makeLine`[JX], `spatialF:convertLatLonBox`[J]
- simplify polygons: `geof:simplifyDp`[JX], `geof:simplifyVw`[JX]
- create collection: `geof:collect`[JX]
- destructure collection / multipolygon: `spatial:st_dump`[JX]
- lat/lon accessors: `geof:lat`[JX], `geof:lon`[JX]
- manual geo-index lookup: `spatial:intersectBoxGeom`[J], `spatial:withinBoxGeom`[J], `spatial:withinCircleGeom`[J], `spatial:withinBoxMultipolygonGeom`[JX]

H3 grid functions (based on H3 API)

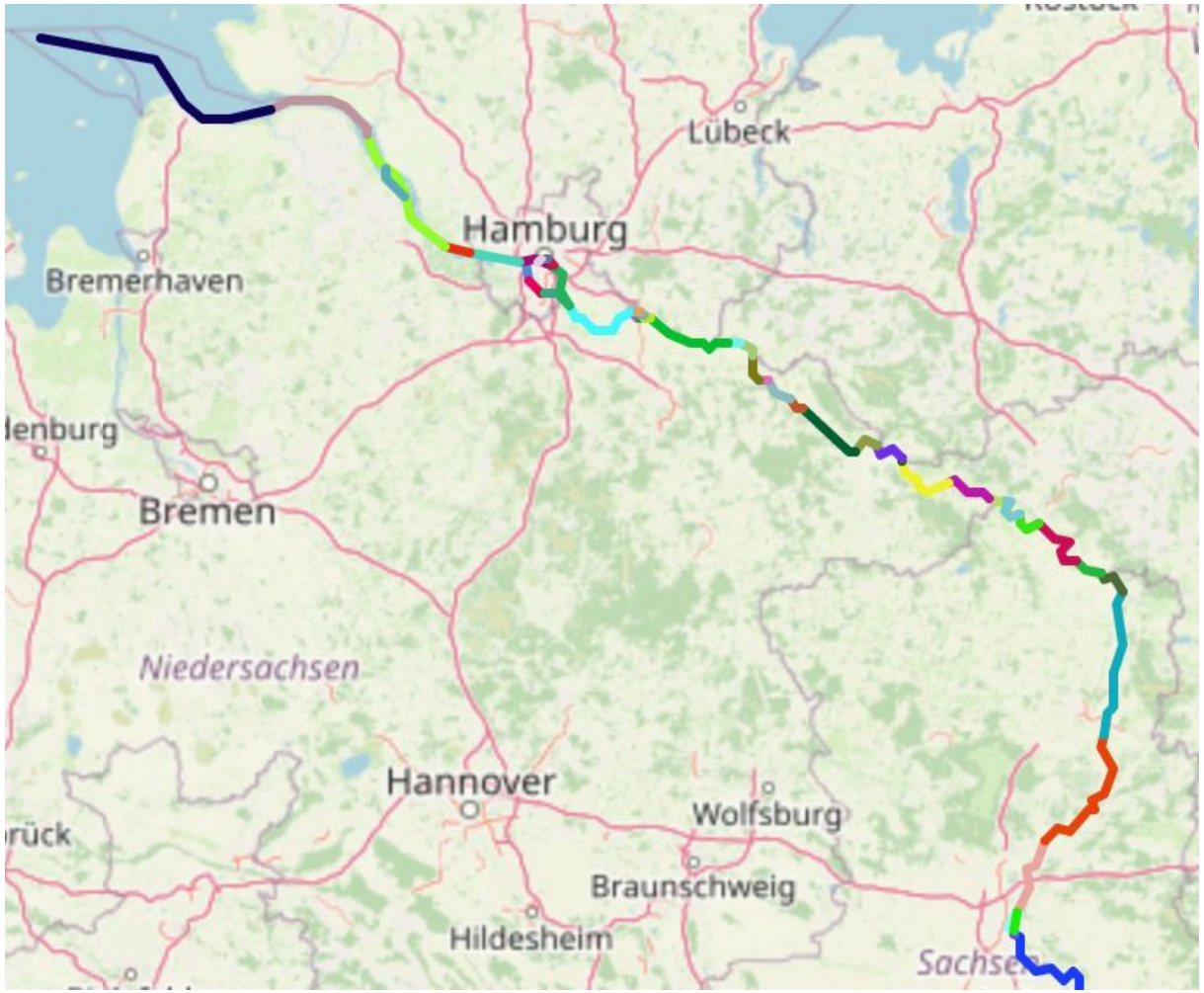
- geof:h3CellIdToGeom (?cellId) -> ?geometry
- geof:h3CellIdToParent (?cellId, ?resolution) -> ?parentId
- geof:h3CellResolution (?cellId) -> ?resolution
- geof:h3GridDistance (?cellId1, ?cellId2) -> ?distanceInCells
- geof:h3IsValidCell (?cellId) -> yes/no
- geof:h3LongLatAsCellId (?long, ?lat, ?resolution) -> ?cellId
- SELECT (geof:h3ToGeom (?cellId) AS ?geom)
- ?childId geo:h3_cellIdToChildren (?cellId ?resolution)
- ?cellId geo:h3_geometryToCellIds (?geometry ?resolution ?fullCover)
- ?cellId geo:h3_gridDisk (?centerCellId ?ringSize)

Issue of non-metric buffering



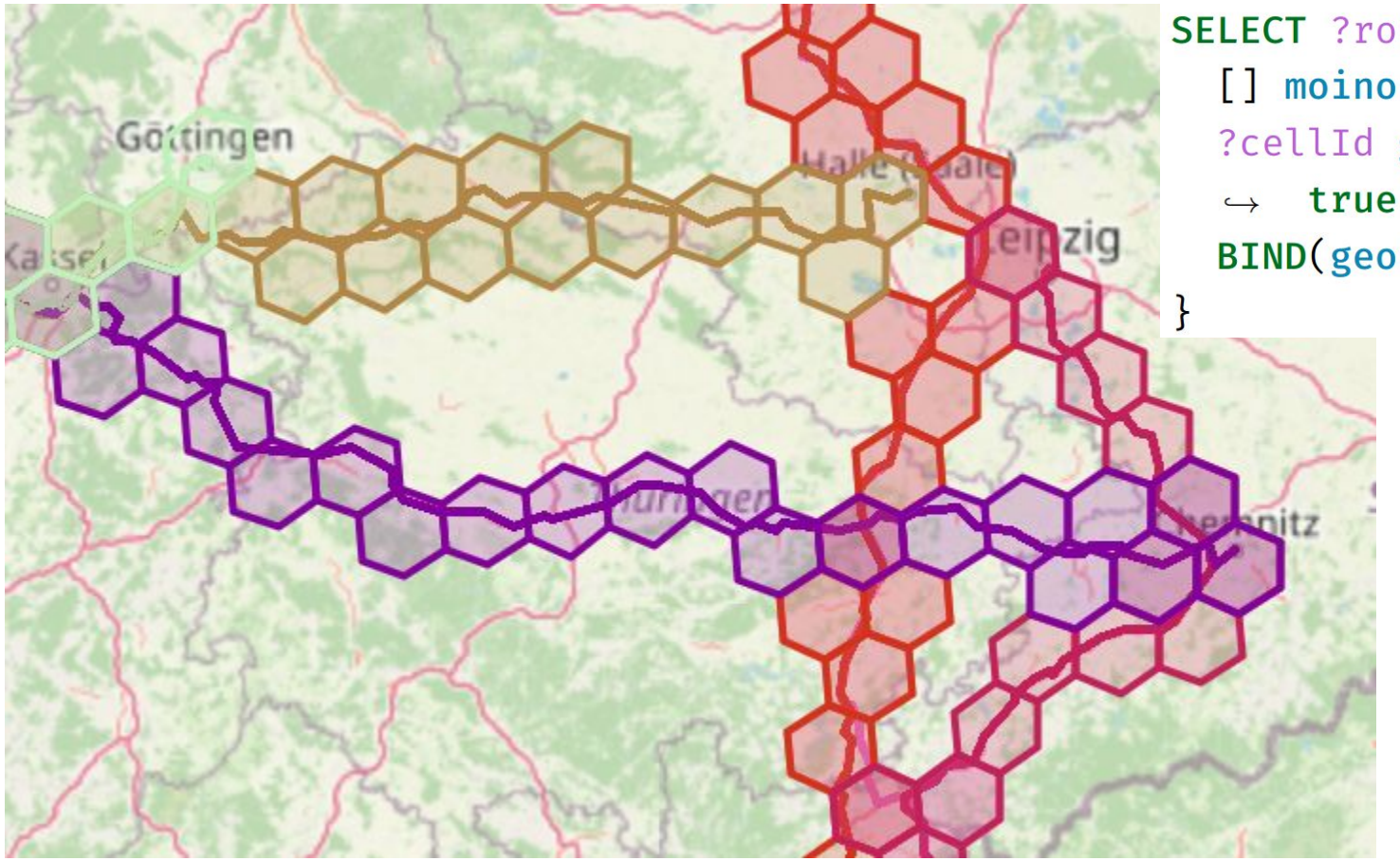

```
BIND(geof:transform(?geo, <http://www.opengis.net/def/crs/EPSG/0/3844>) AS ?metricGeo)
BIND(geof:buffer(?metricGeo, 11113.9, uom:metre) AS ?metricBuffered)
BIND(geof:transform(?metricBuffered, <http://www.opengis.net/def/crs/OGC/1.3/CRS84>) AS
↪  ?geoBufferedCrs84)
```

LineMerge on OSM



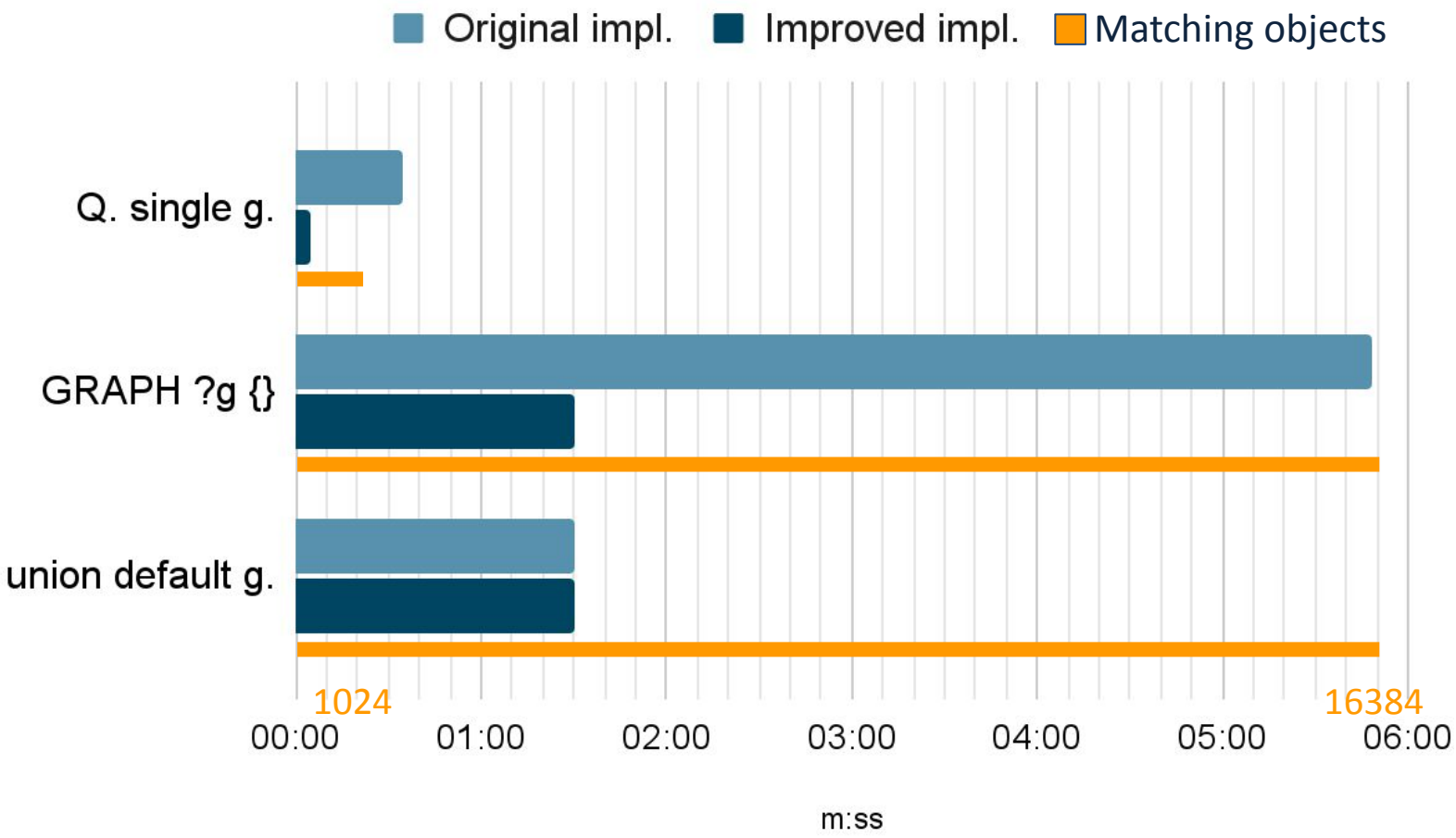

```
SELECT
      ## currently implemented:
      (geof:lineMerge(geof:collect(?wayGeom)) AS ?riverGeom)
      ## possible standards suggestion?
      (geof:aggConcatLines(?wayGeom) AS ?riverGeom)
  WHERE {
    ?s a osm:relation ;
       osmkey:name "Elbe" ;
       osmrel:member/osm:id ?m .
    ?m geo:hasGeometry/geo:asWKT ?wayGeom .
}
```

Transport routes to H3 cells



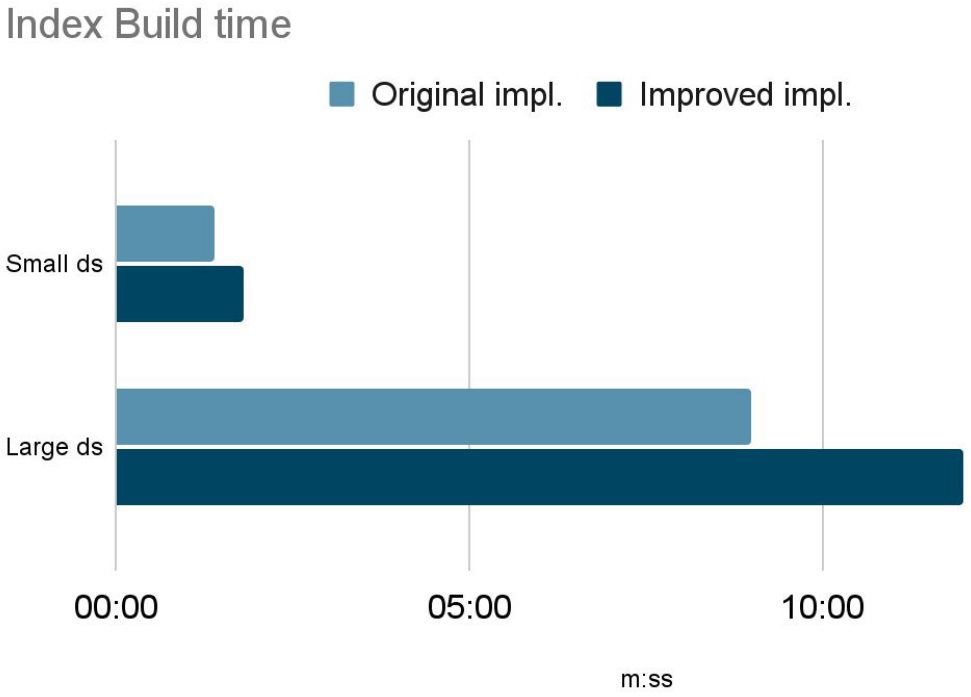

```
SELECT ?routeLineString ?cellGeom {
  [] moino:route ?routeLineString .
  ?cellId geo:h3_geometryToCellIds(?routeLineString 5
  ↪  true)
  BIND(geof:h3CellIdToGeom(?cellId) AS ?cellGeom)
}
```

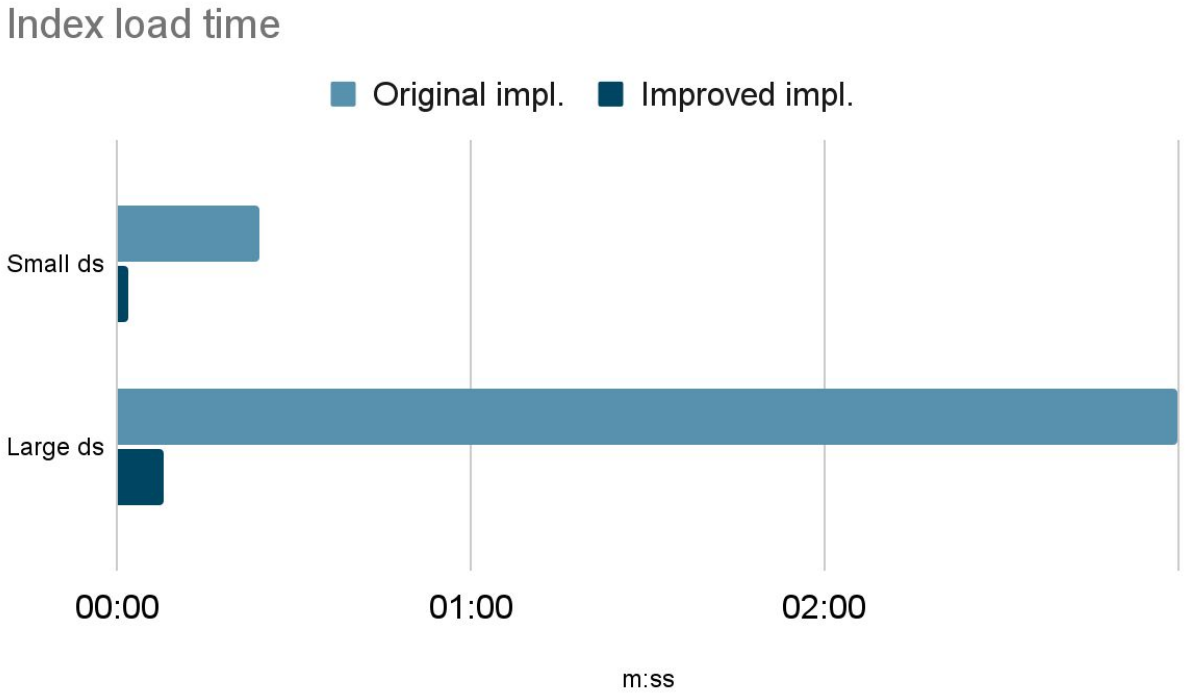
Query of named graphs performance



16 graphs
×
260k objects

Index loading performance





Small ds: 260k simple polygons

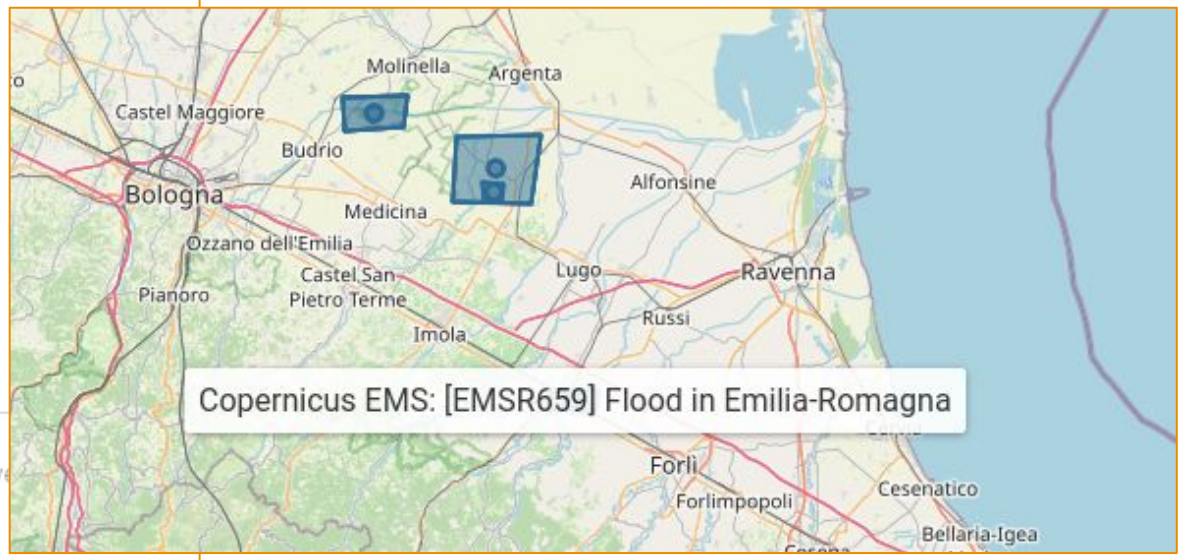
Large ds: 1 mio. simple polygons

Inline Javascript for API transformations

```
27    BIND(xml:path(?event_item, "//georss:polygon/text()") as ?polygon_points
28    BIND(json:je("""
29        let arr = $0.split(' ')
30        let res=[]
31        for (var i = 0 ; i < arr.length - 1; i = i + 2) {
32           res.push(arr[i + 1] + ' ' + arr[i])
33        }
34        return res.join(', ')
35     """, ?polygon_points) AS ?value)
36    BIND("POLYGON((" + ?value + "))" as ?wkt)
37    BIND(strdt(?wkt, geo:wktLiteral) as ?geom)
38  }
39  LIMIT 100
40
```

Table | Response | Gallery | Chart | Geo | Geo-3D | Geo ev

	geom
1	"POLYGON((11.77606211830873 44.48668204095276, 11.80291361887779 44.4849722331955...
2	"POLYGON((11.87157062501555 44.27663992525394, 11.93386214668788 44.2763158019823...
3	"POLYGON((11.7330331630454 44.48405330348036, 11.84236179092796 44.48296196729031,...



Next steps

- Implement/align/evaluate DGGS + H3
- Work on upstreaming changes
- Research local CRS ⇔ differences in calculation of area and length arise (euclidean vs. on ellipsoid)

Links

- Apache Jena: https://jena.apache.org/
  Changes: https://github.com/AKSW/jena/tree/coypu
- Online geo-index module for Apache Jena Fuseki: https://github.com/AKSW/fuseki-mods/tree/adaptions/jena-fmod-geosparql
- JenaX extension module: https://github.com/Scaseco/jenax
- Demo endpoint: https://geosparql.aksw.org/

Thanks

Questions?

Simon Bin
sbin@informatik.uni-leipzig.de

Supported by


Federal Ministry
for Economic Affairs
and Climate Action
GA No. 01MK21007A


Federal Ministry
for Digital
and Transport
GA No. 9F2266A