

# BBoxDB

A Key-Bounding-Box-Value Store

## Demo Paper: Large Scale Spatial Data Processing with User Defined Filters in BBoxDB

Jan Kristof Nidzwetzki / Ralf Hartmut Güting

Fernuniversität in Hagen, Germany

Database Systems for New Applications

{jan.nidzwetzki@studium., rhg@}fernuni-hagen.de

# Outline

## 1 Multi-Dimensional Data in Key-Value Stores

- Key-Value stores
- Partitioning data

## 2 BBoxDB – Architecture and User Defined Filters

- BBoxDB
- Architecture
- User defined filters

## 3 Demonstration and Summary

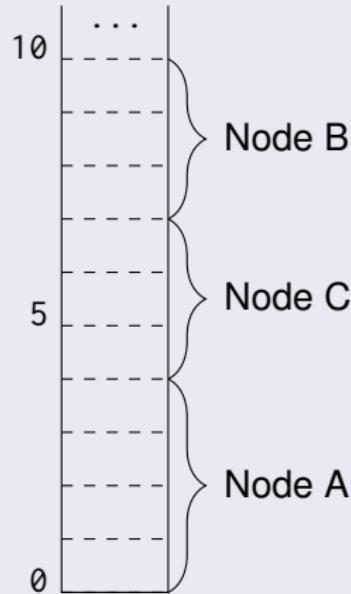
# Basics

## Key-value stores...

- have a simple data model: *key* and *value*.
- support at least the operations `put(table, key, value)` and `get(table, key)`.
- don't support features like *joins* or *transactions*.
- can be implemented as a distributed system.

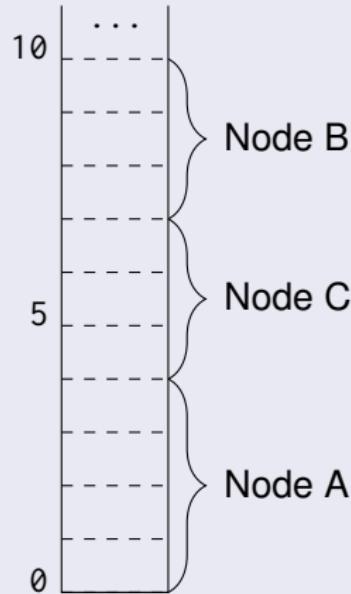
# Partitioning data

## Range partitioning



# Partitioning data

## Range partitioning

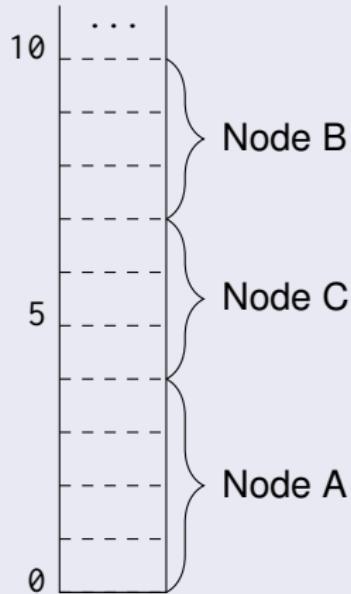


## The key is important...

- to determine which node is responsible for a value.
- to locate the value on the node efficiently (e.g., *binary search* on sorted key-value pairs).

# Partitioning data

## Range partitioning



## The key is important...

- to determine which node is responsible for a value.
- to locate the value on the node efficiently (e.g., *binary search* on sorted key-value pairs).

## Data access path

Keys are the data access path in a key-value store. If a key is not known, an expensive full data scan is needed.

# Multi-dimensional data in key-value stores

One-dimensional data (e.g., a customer record)

customerid=1234,  
firstname=John,  
lastname=Doe

Key

1234

# Multi-dimensional data in key-value stores

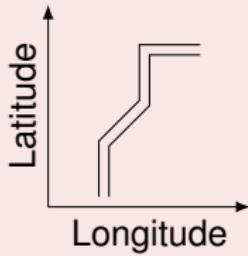
One-dimensional data (e.g., a customer record)

customerid=1234,  
firstname=John,  
lastname=Doe

Key

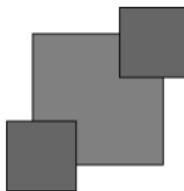
1234

Two-dimensional data (e.g., a road)



Key

?



# BBoxDB

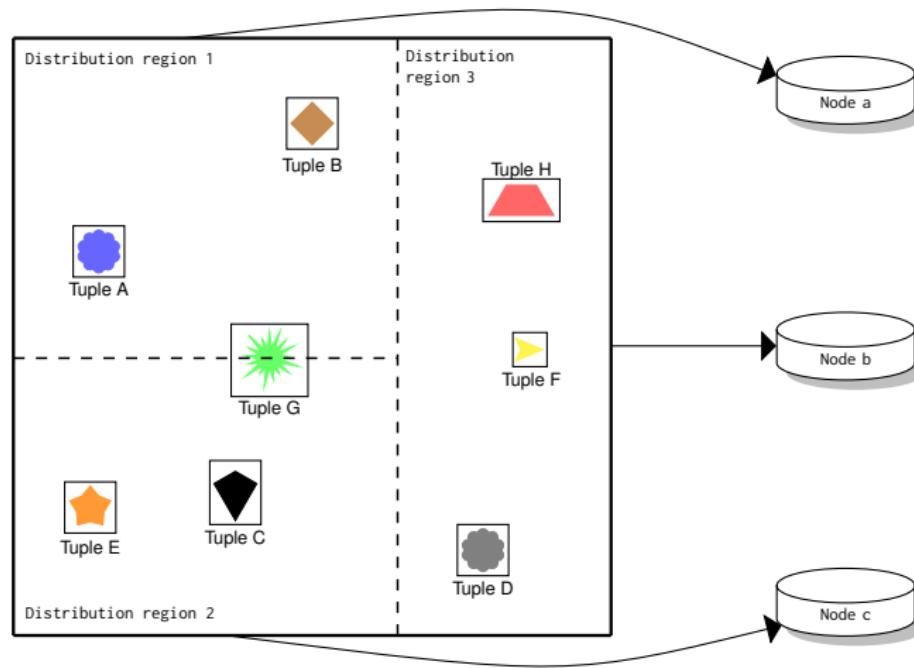
A Key-Bounding-Box-Value Store

# BBoxDB – A distributed key-bounding-box-value store

## BBoxDB...

- is a distributed *key-bounding-box-value store*.
- stores each value together with a *bounding box*. The bounding box determines the location of the value in the  $n$ -dimensional space.
- can handle  $n$ -dimensional *point* and *non-point* big data.
- partitions the space dynamically and redistributes the data.
- stores data co-partitioned for efficient spatial joins.
- is freely available and licensed under the *Apache 2.0* license.

# Partitioning the space



# Supported Operations

## The most important operations

- **Store new data:**

`put(table, key, hyperrectangle, value)`

- **Retrieve data:**

`getByRange(table, hyperrectangle)`

- **Execute a spatial join:**

`join(table1, table2, hyperrectangle)`

# Supported Operations

## The most important operations

- **Store new data:**

`put(table, key, hyperrectangle, value)`

- **Retrieve data:**

`getByRange(table, hyperrectangle)`

- **Execute a spatial join:**

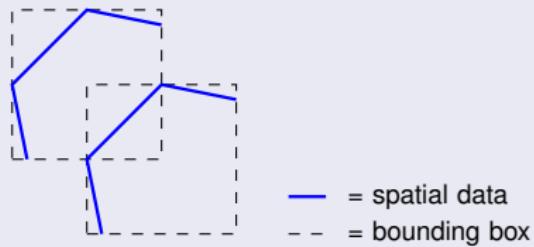
`join(table1, table2, hyperrectangle)`

## Query processing in BBoxDB

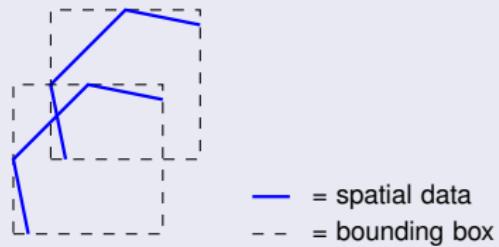
BBoxDB is a generic datastore; values are plain arrays of bytes. The semantics of the stored values are unknown. The software performs operations (e.g., range queries or spatial joins) **only** on the bounding boxes of the data.

# Spatial data with overlapping bounding boxes

## Spatial data **is not** intersecting



## Spatial data **is** intersecting



## Bounding boxes / Spatial join

Overlapping bounding boxes are a necessary but not a sufficient criterion for a spatial join on real geometries.

## User defined filters

# Motivation for user defined filters

## User defined filters . . .

- enhance the query processor, so that the stored values can be decoded (e.g., GeoJSON encoded values).

# Motivation for user defined filters

## User defined filters...

- enhance the query processor, so that the stored values can be decoded (e.g., GeoJSON encoded values).
- **turn the generic data store into a specialized system for a specific data type (e.g., spatial joins on the real geometries of stored values become possible).**

# Implementation details

## User defined filters in BBoxDB

- The UDF acts as a filter and refines the output of the query processor.
- The creation of a new UDF is simple: only the two methods of the interface `UserDefinedFilter` need to be implemented.
- The method `filterTuple` refines range queries; the method `filterJoinCandidate` refines join queries.
- The UDF is compiled into *Java bytecode* and is loaded dynamically at runtime.
- Existing Java libraries can be used (e.g., the *Esri Geometry API for Java*).

# Demonstration

# Summary

## BBoxDB...

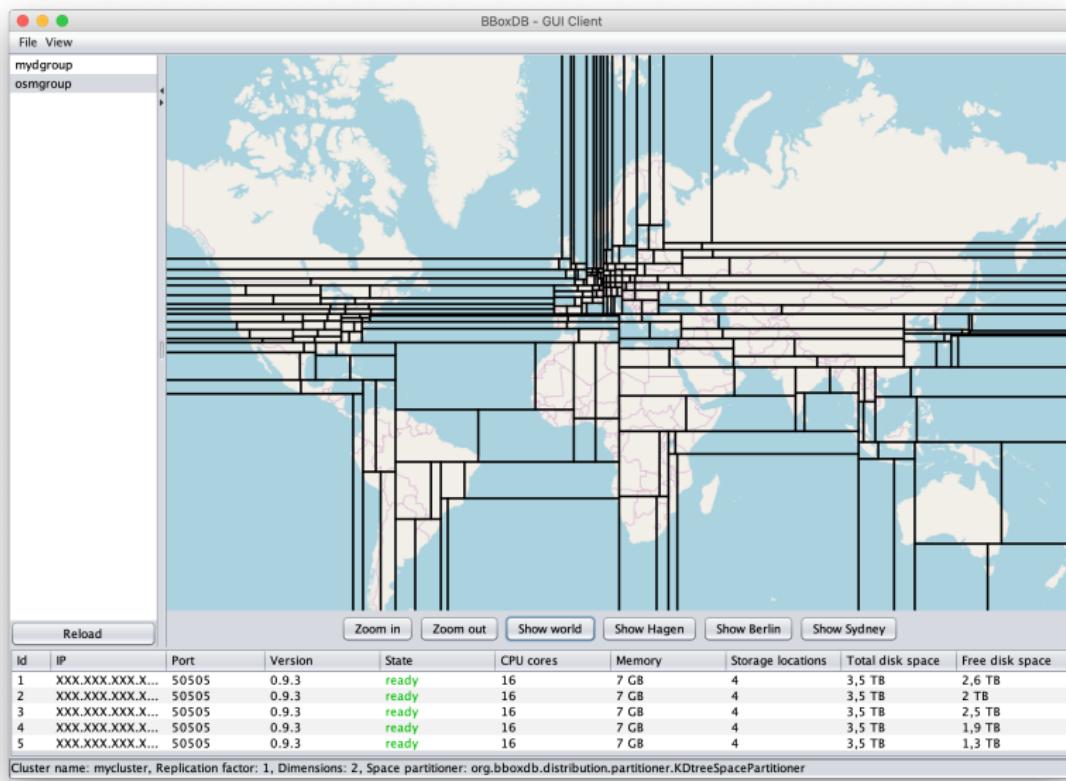
- is a *distributed key-bounding-box-value store*.
- enhances the key-value data model with a bounding box.
- executes operations only on the bounding boxes of the data.
- **can be enhanced by user defined filters. They turn BBoxDB into a system that is specialized on a certain data type.**
- is written in Java and licensed under the *Apache 2.0 license*.

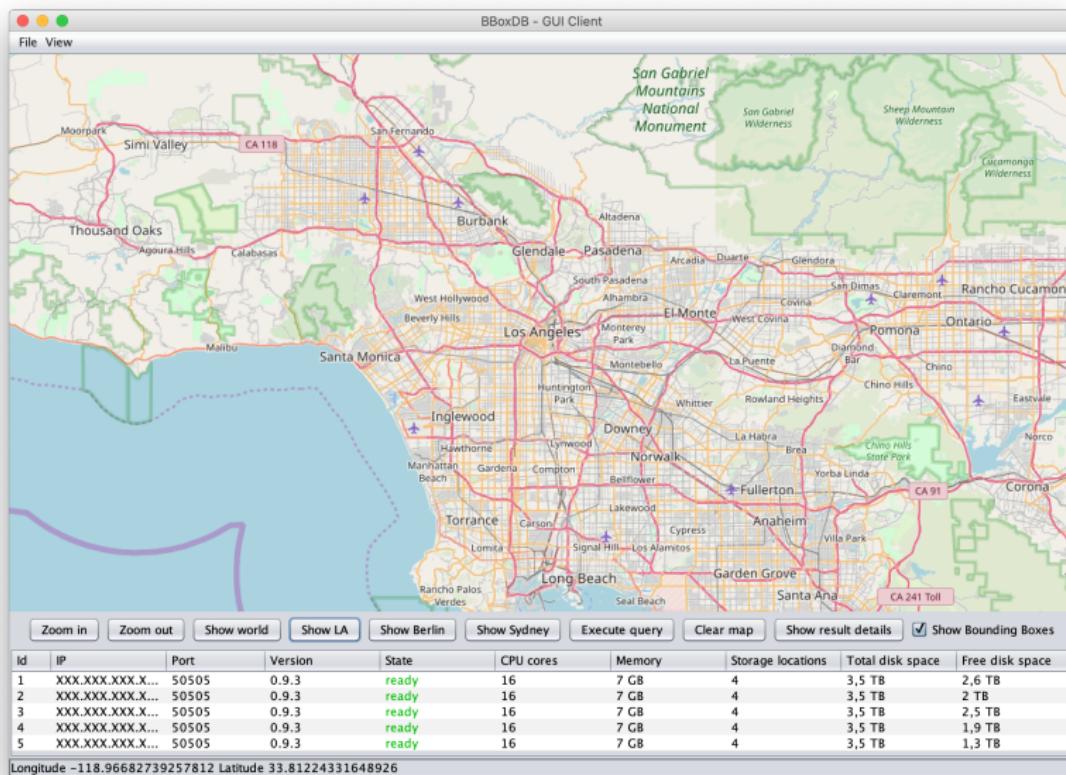
# Questions?

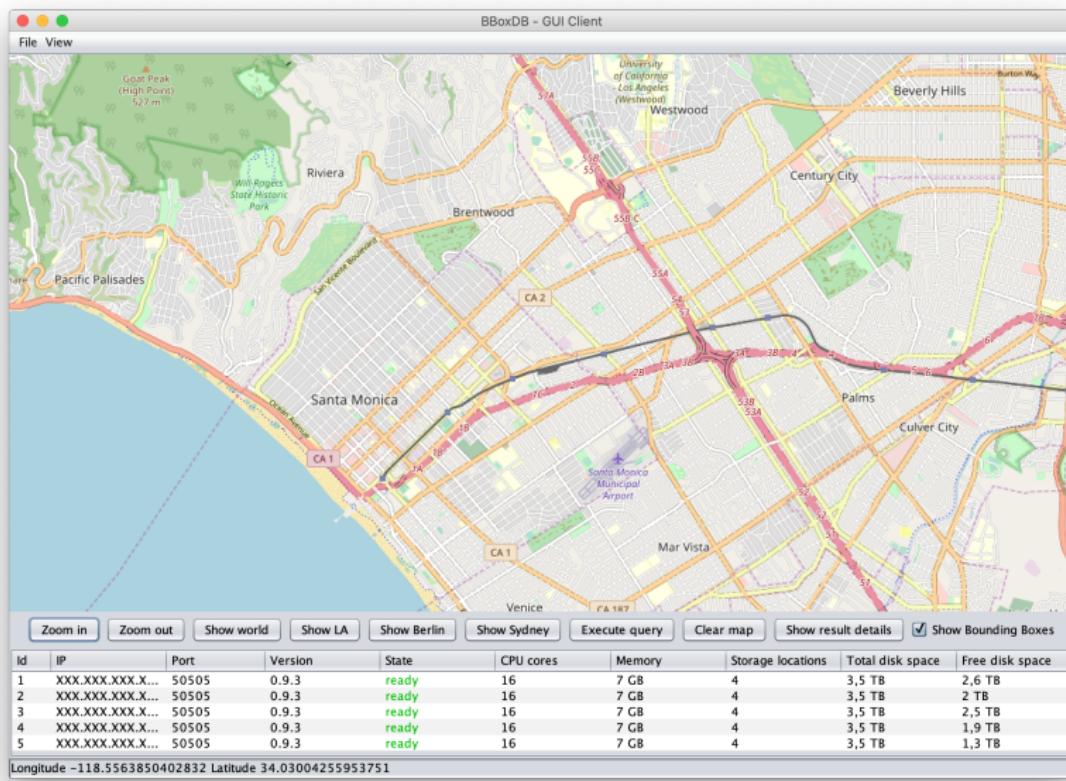
## Further information

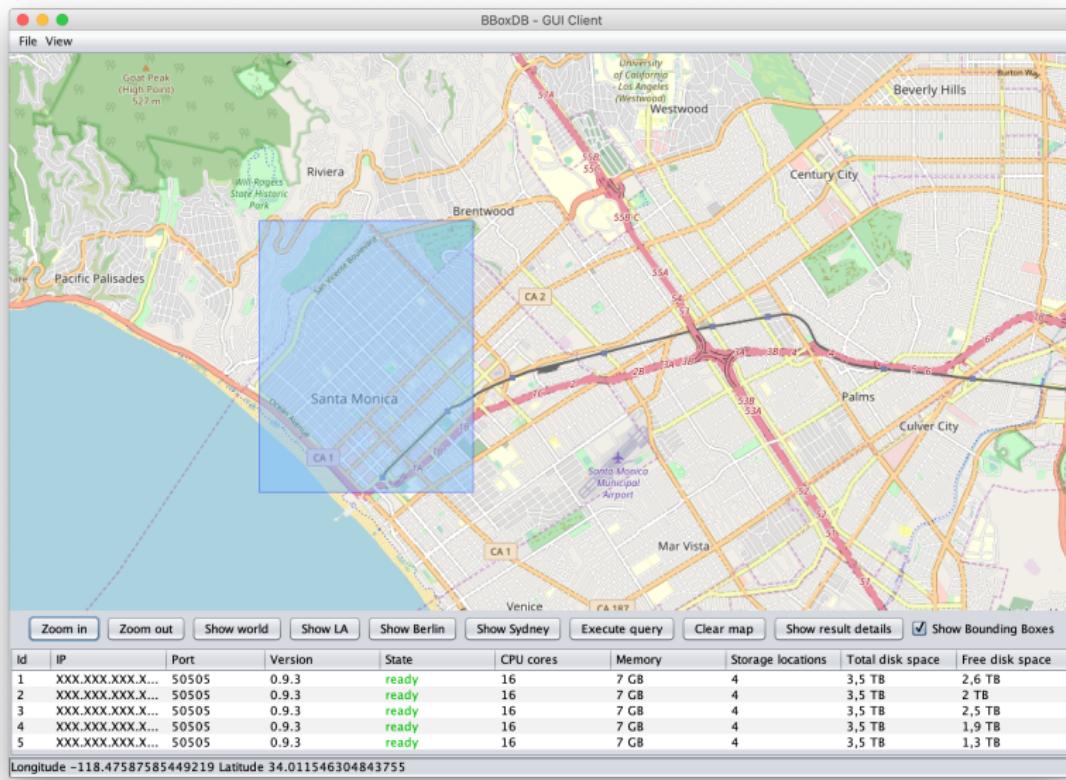
- Web: <https://bboxdb.org>
- Github: <https://github.com/jnidzwetzki/bboxdb>
- Twitter: @bboxdb
- Google Groups: <https://groups.google.com/forum/bboxdb>

# Backup Slides









BBoxDB - GUI Client

File View

BBoxDB - Execute query

**Query**

Type: Range query

Table 1: osmgroup\_road

Color for table 1: Red

Table 2: mydgroup\_table1

Color for table 2: Green

**Parameter**

Longitude begin: -118.51312637329102

Longitude end: -118.47587585449219

Latitude begin: 34.011546304843755

Latitude end: 34.05066824173537

Name:

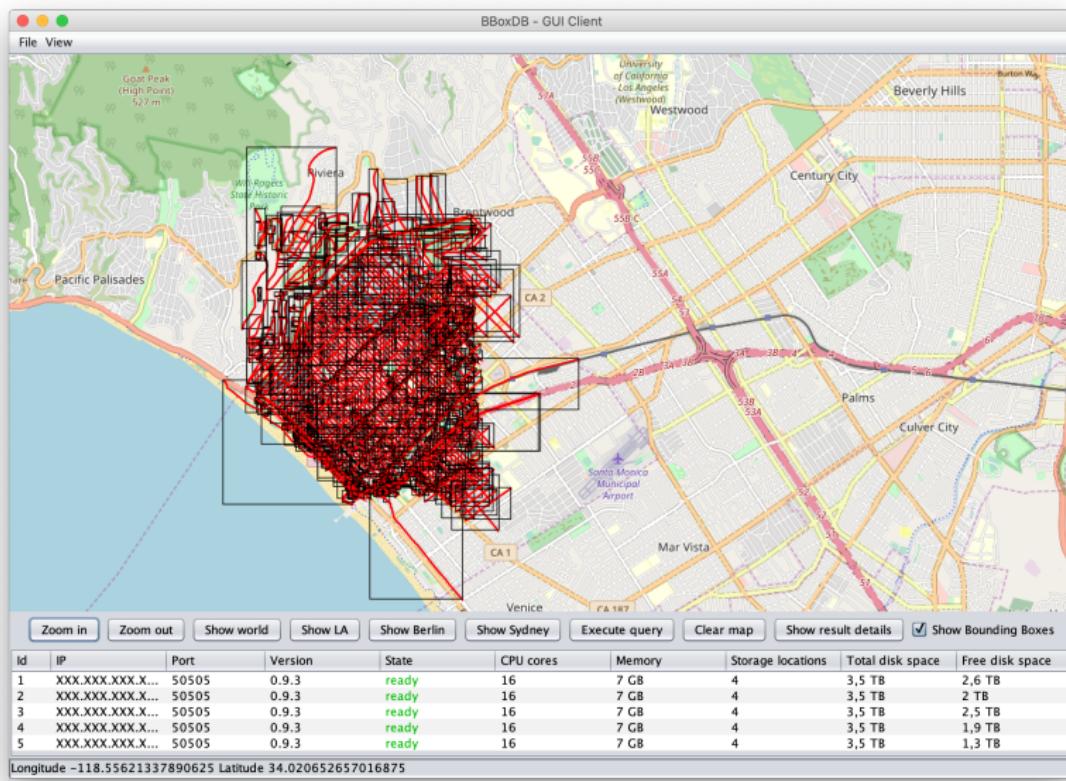
Value:

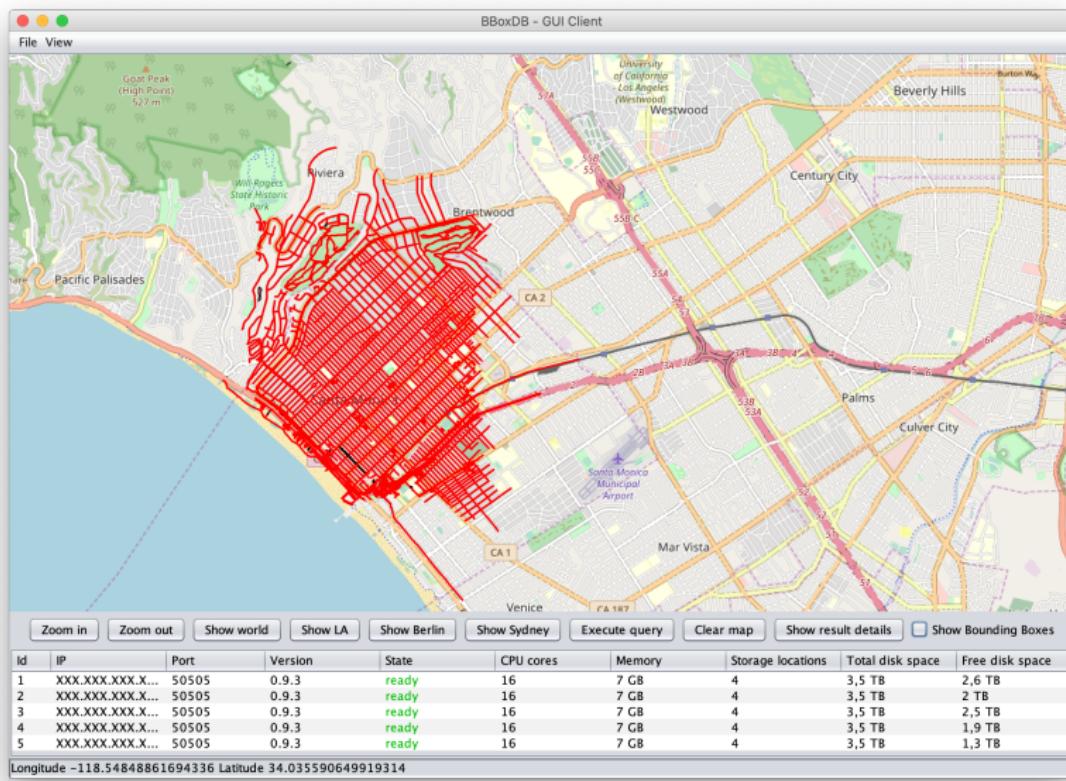
**Close** **Execute**

Zoom in Zoom out Show world Show LA Show Berlin Show Sydney Execute query Clear map Show result details  Show Bounding Boxes

ID	IP	Port	Version	State	CPU cores	Memory	Storage locations	Total disk space	Free disk space
1	XXX.XXX.XXX.X...	50505	0.9.3	ready	16	7 GB	4	3,5 TB	2,6 TB
2	XXX.XXX.XXX.X...	50505	0.9.3	ready	16	7 GB	4	3,5 TB	2 TB
3	XXX.XXX.XXX.X...	50505	0.9.3	ready	16	7 GB	4	3,5 TB	2,5 TB
4	XXX.XXX.XXX.X...	50505	0.9.3	ready	16	7 GB	4	3,5 TB	1,9 TB
5	XXX.XXX.XXX.X...	50505	0.9.3	ready	16	7 GB	4	3,5 TB	1,3 TB

Longitude -118.54351043701172 Latitude 34.028762179464465





BBoxDB – GUI Client

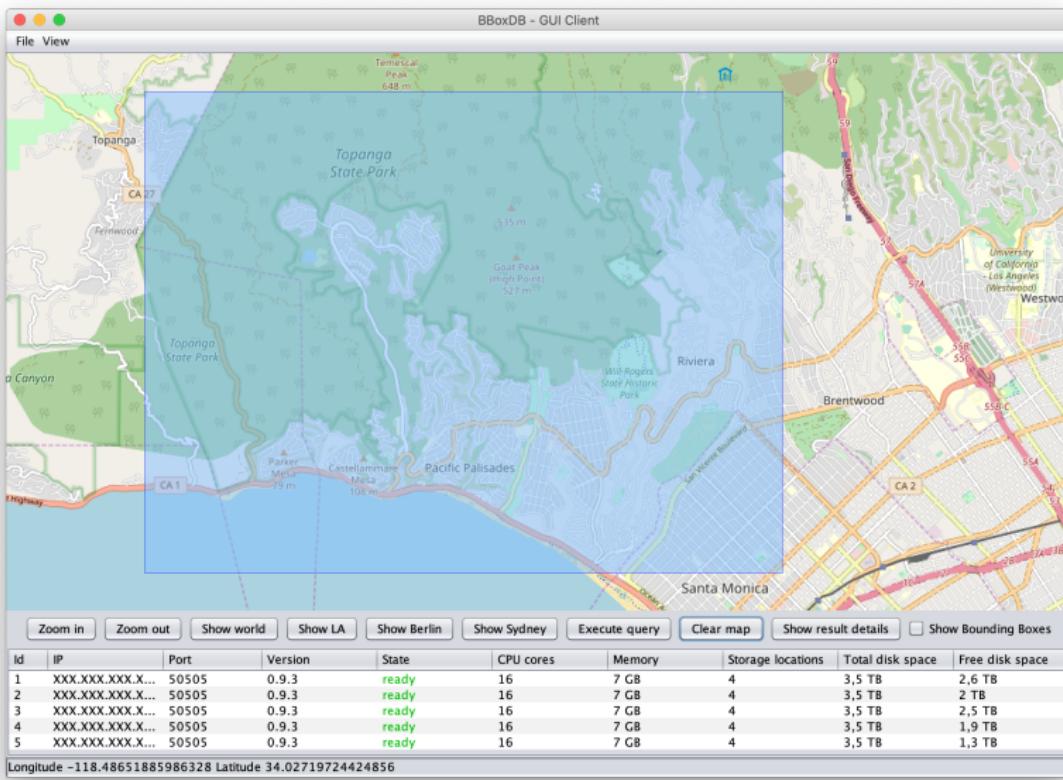
The screenshot shows a map of the Los Angeles area, specifically focusing on the Venice and Santa Monica regions. A large red shaded polygon covers a significant portion of the coastal and inland areas. A callout box is overlaid on the map, containing the following table:

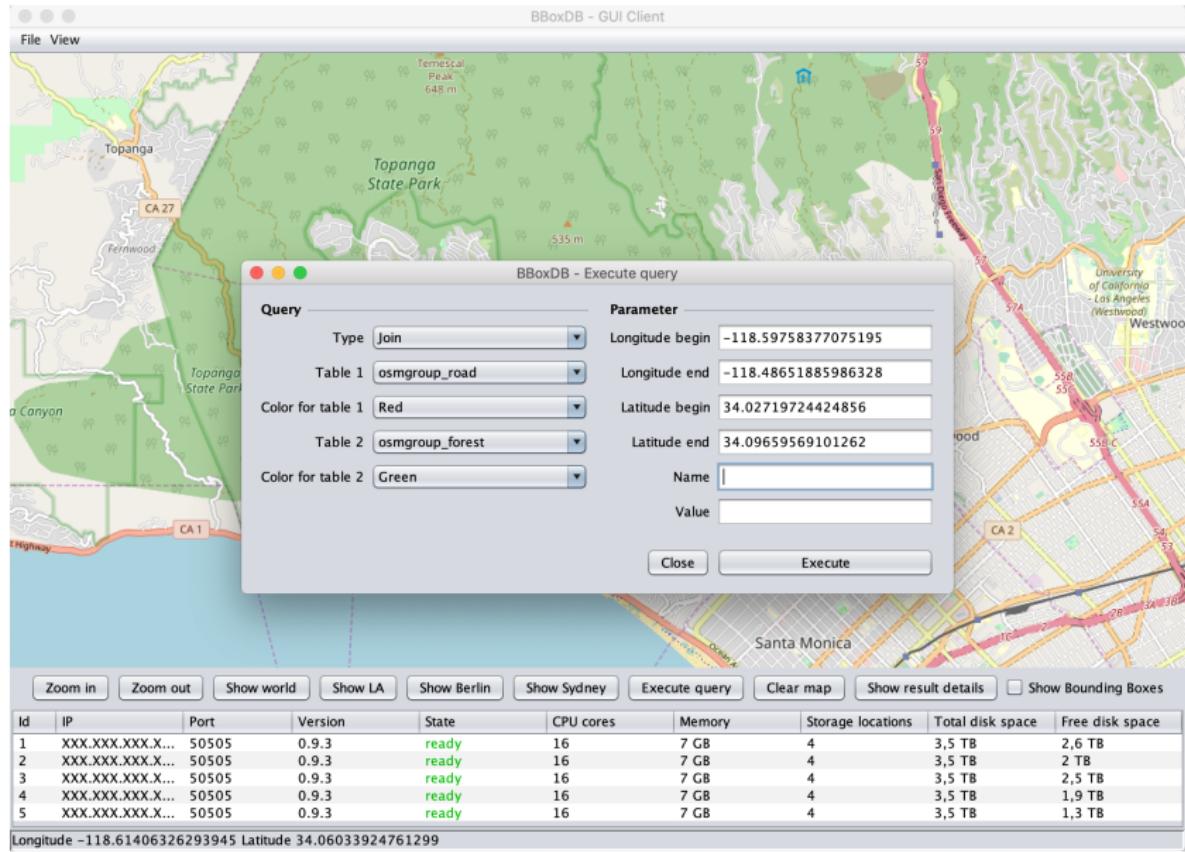
Table: osmgroup_road
id: 13318443
tiger:fcfc A41
name_1: South Anita Avenue
tiger-reviewed: no
name: Anita Avenue
tigername_base: Anita
tigername_type_1: Ave
tigername_type: Ave
tigername_direction_1: S
highway: residential
tigercounty: Los Angeles, CA

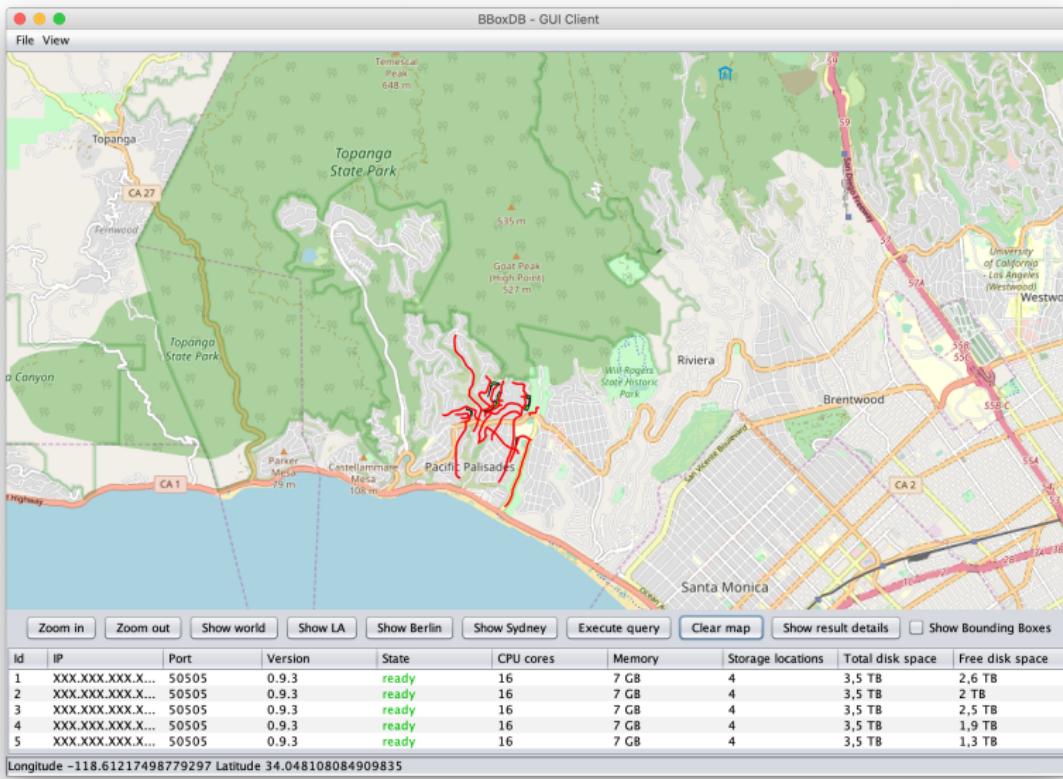
Below the map is a toolbar with buttons: Zoom in, Zoom out, Show world, Show LA, Show Berlin, Show Sydney, Execute query, Clear map, Show result details, and a checkbox for Show Bounding Boxes.

ID	IP	Port	Version	State	CPU cores	Memory	Storage locations	Total disk space	Free disk space
1	XXX.XXX.XXX.X...	50505	0.9.3	ready	16	7 GB	4	3,5 TB	2,6 TB
2	XXX.XXX.XXX.X...	50505	0.9.3	ready	16	7 GB	4	3,5 TB	2 TB
3	XXX.XXX.XXX.X...	50505	0.9.3	ready	16	7 GB	4	3,5 TB	2,5 TB
4	XXX.XXX.XXX.X...	50505	0.9.3	ready	16	7 GB	4	3,5 TB	1,9 TB
5	XXX.XXX.XXX.X...	50505	0.9.3	ready	16	7 GB	4	3,5 TB	1,3 TB

At the bottom left, the coordinates are displayed as Longitude -118.48308563232422 Latitude 34.057210513510306.







BBoxDB – GUI Client

File View

Zoom in Zoom out Show world Show LA Show Berlin Show Sydney Execute query Clear map Show result details  Show Bounding Boxes

ID	IP	Port	Version	State	CPU cores	Memory	Storage locations	Total disk space	Free disk space
1	XXX.XXX.XXX.X...	50505	0.9.3	ready	16	7 GB	4	3,5 TB	2,6 TB
2	XXX.XXX.XXX.X...	50505	0.9.3	ready	16	7 GB	4	3,5 TB	2 TB
3	XXX.XXX.XXX.X...	50505	0.9.3	ready	16	7 GB	4	3,5 TB	2,5 TB
4	XXX.XXX.XXX.X...	50505	0.9.3	ready	16	7 GB	4	3,5 TB	1,9 TB
5	XXX.XXX.XXX.X...	50505	0.9.3	ready	16	7 GB	4	3,5 TB	1,3 TB

Longitude -118.55187892913818 Latitude 34.03957367053948

BBoxDB – GUI Client

**Table: osmgroup\_road**  
 id: 42766630  
 surface: asphalt  
 abutters: residential  
 tigerviewed: yes  
 source\_ref: AM909\_DSCS6390  
 source: usgs\_imagery\_2006:survey:image  
 tigername\_type: Rd  
 tigercountry: Los Angeles, CA  
 tigercfc: A41  
 name: Temescal Canyon Road  
 lanes: 4  
 tigername\_base: Temescal Canyon  
 highway: secondary  
 cycleway: lane

**Table: osmgroup\_forest**  
 id: 63918822  
 natural: wood

Zoom in | Zoom out | Show world | Show LA | Show Berlin | Show Sydney | Execute query | Clear map | Show result details |  Show Bounding Boxes

ID	IP	Port	Version	State	CPU cores	Memory	Storage locations	Total disk space	Free disk space
1	XXX.XXX.XXX.X...	50505	0.9.3	ready	16	7 GB	4	3,5 TB	2,6 TB
2	XXX.XXX.XXX.X...	50505	0.9.3	ready	16	7 GB	4	3,5 TB	2 TB
3	XXX.XXX.XXX.X...	50505	0.9.3	ready	16	7 GB	4	3,5 TB	2,5 TB
4	XXX.XXX.XXX.X...	50505	0.9.3	ready	16	7 GB	4	3,5 TB	1,9 TB
5	XXX.XXX.XXX.X...	50505	0.9.3	ready	16	7 GB	4	3,5 TB	1,3 TB

Longitude -118.5310790252686 Latitude 34.04512113757758

Multi-Dimensional Data in Key-Value Stores  
BBoxDB – Architecture and User Defined Filters  
Demonstration and Summary

Demonstration  
Summary

BBoxDB - GUI Client

File View

Favorites

Recent

Map

Execute query

Zoom in Zoom out Show world Show LA Show Berlin Show Sydney Execute query Clear map Show result details Show Bounding Boxes

Query

Type: Join

Table 1: osmgroup\_road

Color for table 1: Red

Table 2: osmgroup\_forest

Color for table 2: Green

Parameter

Longitude begin: -118.54711532592773

Longitude end: -118.51823329925537

Latitude begin: 34.034381481654364

Latitude end: 34.05262386500679

Name: .UserDefinedGeoJsonSpatialFilter

Value:

Close Execute

Longitude -118.52823257446289 Latitude 34.034025840649385

Storage locations Total disk space Free disk space

ID	IP	Port	Version	State	CPU cores	Memory	Storage locations	Total disk space	Free disk space
1	XXX.XXX.XXX.X...	50505	0.9.3	ready	16	7 GB	4	3,5 TB	2,6 TB
2	XXX.XXX.XXX.X...	50505	0.9.3	ready	16	7 GB	4	3,5 TB	2 TB
3	XXX.XXX.XXX.X...	50505	0.9.3	ready	16	7 GB	4	3,5 TB	2,5 TB
4	XXX.XXX.XXX.X...	50505	0.9.3	ready	16	7 GB	4	3,5 TB	1,9 TB
5	XXX.XXX.XXX.X...	50505	0.9.3	ready	16	7 GB	4	3,5 TB	1,3 TB

BBoxDB – GUI Client

File View

Zoom in Zoom out Show world Show LA Show Berlin Show Sydney Execute query Clear map Show result details  Show Bounding Boxes

ID	IP	Port	Version	State	CPU cores	Memory	Storage locations	Total disk space	Free disk space
1	XXX.XXX.XXX.X...	50505	0.9.3	ready	16	7 GB	4	3,5 TB	2,6 TB
2	XXX.XXX.XXX.X...	50505	0.9.3	ready	16	7 GB	4	3,5 TB	2 TB
3	XXX.XXX.XXX.X...	50505	0.9.3	ready	16	7 GB	4	3,5 TB	2,5 TB
4	XXX.XXX.XXX.X...	50505	0.9.3	ready	16	7 GB	4	3,5 TB	1,9 TB
5	XXX.XXX.XXX.X...	50505	0.9.3	ready	16	7 GB	4	3,5 TB	1,3 TB

Longitude -118.55269432067871 Latitude 34.040889448752345

BBoxDB – GUI Client

**Table: osmgroup\_road**  
 Id: 390152905  
 tigerfcfc: A41  
 surface: asphalt  
 tigerreviewed: no  
 name: Bowdoin Street  
 lanes: 2  
 tigername\_base: Bowdoin  
 tigername\_type: St  
 highway: residential  
 tigercounty: Los Angeles, CA

**Table: osmgroup\_forest**  
 Id: 639188222  
 natural: wood

Zoom in | Zoom out | Show world | Show LA | Show Berlin | Show Sydney | Execute query | Clear map | Show result details |  Show Bounding Boxes

ID	IP	Port	Version	State	CPU cores	Memory	Storage locations	Total disk space	Free disk space
1	XXX.XXX.XXX.X...	50505	0.9.3	ready	16	7 GB	4	3,5 TB	2,6 TB
2	XXX.XXX.XXX.X...	50505	0.9.3	ready	16	7 GB	4	3,5 TB	2 TB
3	XXX.XXX.XXX.X...	50505	0.9.3	ready	16	7 GB	4	3,5 TB	2,5 TB
4	XXX.XXX.XXX.X...	50505	0.9.3	ready	16	7 GB	4	3,5 TB	1,9 TB
5	XXX.XXX.XXX.X...	50505	0.9.3	ready	16	7 GB	4	3,5 TB	1,3 TB

Longitude -118.53076457977295 Latitude 34.046365711752664