A round table for multi-disciplinary research on Geospatial and Climate Data

ARTICLE · SEPTEMBER 2015

READS
15

7 AUTHORS, INCLUDING:

Romulo Goncalves
Netherlands eScience Center
28 PUBLICATIONS **237** CITATIONS
SEE PROFILE

Jason Maassen
Netherlands eScience Center
78 PUBLICATIONS **1,670** CITATIONS
SEE PROFILE

Kostis Kyzirakos
Centrum Wiskunde & Informatica
37 PUBLICATIONS **173** CITATIONS
SEE PROFILE

Oscar Martinez Rubi
Netherlands eScience Center
17 PUBLICATIONS **81** CITATIONS
SEE PROFILE

A round table for multi-disciplinary research on Geospatial and Climate Data

Romulo Goncalves[1], Milena Ivanova[a3], Foteini Alvanaki[2], Jason Maassen[1], Kostis Kyzirakos[2], Oscar Martinez-Rubi[1], and Hannes Mühleisen[2]

[1]NLeSC Amsterdam, The Netherlands {*{r.goncalves,j.maassen,o.rubi}@esciencecenter.nl*}
[2]CWI Amsterdam, The Netherlands {*{kostis.kyzirakos,f.alvanaki,hannes.muehleisen}@cwi.nl*}
[3]NuoDB Cambridge MA, USA {*mivanova@nuodb.com*}

***Abstract*—Earth observation sciences produce large sets of data which are inherently rich in spatial and geo-spatial information. Together with live data collected from monitoring systems and large collections of semantically rich objects they provide new opportunities for advanced eScience research on climatology, urban planning and smart cities to name a few.**

Such combination of heterogeneous data sets forms a new source of knowledge. Efficient knowledge extraction from them is an eScience challenge. It requires efficient bulk data injection from both static and streaming data sources, dynamic adaptation of the physical and logical schema, efficient methods to correlate spatial and temporal data, and flexibility to (re-)formulate the research question at any time.

In this work, we present a data management layer over a column-oriented relational data management system that provides efficient analysis of spatiotemporal data. It provides fast data ingestion through different data loaders, tabular and array-based storage, and a dynamic step-wise exploration.

I. Introduction

Spatial location is among the core aspects of data in climatology and urban planning. Current research in these areas often use a combination of data sources. For example, in [30] a combination of point cloud data (produced by LIDAR scanners), meteorological data (produced by weather stations) and cadastral data is used to study the spatial variability of urban heat islands and thermal comfort within the city boundaries.

In such studies scientists try to turn a collection of measurements into useful information through analysis and interpretation in the context of what they already know. The thoughtful and systematic gathering, analysis, and interpretation of data allows a collection of measurements to be converted into evidence that supports scientific ideas, arguments, and hypotheses.

At the initial stages, huge amounts of raw data are collected to be scanned and filtered to remove noise or irrelevant properties. The data is often stored using domain specific file-based solutions. Although this allows efficient access to the data in its original format, data isolation, data redundancy, and application dependency on data formats are major drawbacks of this approach. Furthermore, complex ad-hoc queries are hard to express, particularly when faced with the challenge to combine numerous data sources. File-based solutions have also poor vertical and horizontal scalability [17].

The filtering stage is followed by a simple aggregation phase to detect if the data is meaningful or not. With a single scan simple conclusions are induced from this type of analysis. However, for more complex analysis external specialized libraries are required for each of the data sets. For statistical analysis R [28] is the preferred tool while for geospatial operations for urban planning research scientists use libraries such as SAGA [1].

The data interchange between external libraries is tedious and the scientist is often forced to move data back and forth between systems and storage formats until the final answer is reached. This process is inefficient and might require several iterations of data conversion. Furthermore, it does not provide enough flexibility to change research direction during the process since the knowledge extracted in each iteration is not kept aside in the same format for easy re-utilization. It forces a scientist to re-design the entire pipeline and repeat all the process we just described.

In this paper we introduce a solution to this problem: a round table architecture that facilitates the integration of different heterogeneous data sets for exploration in four dimensions, 3D space and time.

The solution we propose is based on a single database management system (DBMS) that is extended to provide fast data ingestion of large geo-spatial and meteorological data sets. By offering multiple front-ends to access the data, such as SQL, R [28] and SciQL [22], the scientist is able to use simple queries to combine different data sources and extract new knowledge. This allows dynamic and step-wise exploration of data in a flexible and efficient manner. Although we use for illustration climate research, our solution is applicable to other scientific domains with large spatial data sets such as astronomy, seismology, etc.

The remainder of the paper is as follows. Section II discusses the general architecture. In Section III, through different use case scenarios, flexibility and efficiency on exploring climate and geo-spatial data is shown. Finally the article ends

[a] All contributions were done while working at the Netherlands eScience Center.

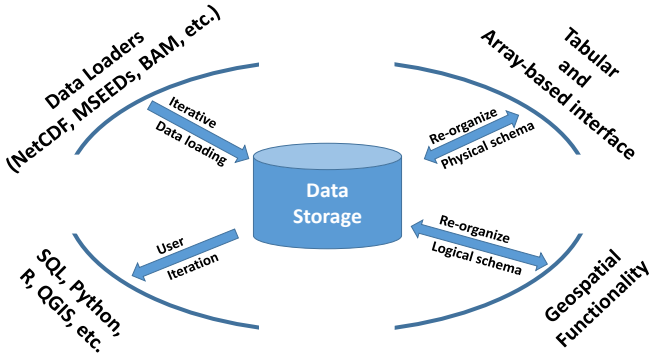


Fig. 1: Round table schema

with a summary in Section IV and future plans in Section V.

II. Round table

In this section we introduce the architecture of our *round table* presented in Figure 1. It has a set of data loaders, one for each scientific data format, it provides several storage and logical schemas, and it supports a set of front-ends for better expressiveness of research questions. All components are connected through a common storage layer and data management kernel, so derived knowledge can be stored and reused for new, or reformulated, queries.

Following the methodology used by urban planning scenarios, the coming sections identify the issues at each stage of the process together with components of our *round table* that solve them.

A. Heterogeneous data integration

The first challenge in combining heterogeneous data sources is to store the data under the same storage without changing the users data structure perception, i.e., the conceptual schema [2] should remain the same. For efficiency reasons the data organization at the physical storage system follows a different schema to optimally exploit the hardware characteristics.

At the core of our *round table* we have a data management system (DBMS). One of the major advantages of a DBMS the clear separation it provides between the physical and the conceptual schema. Such separation creates the opportunity to have different types of applications exploring the same data sets. Furthermore, they are designed to efficiently address scalability issues of managing large volumes of data such as the ones produced in climate observations [13].

B. Scientific data loaders

Scientific data repositories have been a challenge for current relational DBMSs (RDBMS) due to the high cost of converting and loading data into a pre-defined schema. Researchers in [21] have explored a possible solution, *data vaults* for *in-situ* data access by a relational database.

A data vault provides a symbiosis between a DBMS and existing file-based repositories. It keeps data in its original format while scalable processing functionality is offered through the DBMS. Depending on the data format, it provides transparent access to all data kept in the repository through a tabular or array-based interface.

One of the strongest characteristics of data vaults is their ability to exploit the metadata that are present in all data formats. Data loading in data vaults comprises of two phases: the attachment of a file and the import of the file. During the attachment, the file's metadata is loaded into the database. At query time, the metadata is used to decide whether the file has information relevant to the query. In such a case the file is imported into the database. During the import the actual data of the file is loaded.

Inspired by the work in [21], our work creates the same type of access for NetCDF files. NetCDF stands for Network Common Data Form [3]. It is used as input/output format in oceanography, meteorology, and climate research. They support the creation, access and sharing of array-oriented scientific data. NetCDF files are rich in metadata, like creation time, array dimensions, units of measurement, coordinate system, to name a few.

To represent external NetCDF data in our DBMS we defined a mapping between the external data structures and the in-database counterparts. The metadata are represented into catalog tables using a straightforward mapping. Since the NetCDF format is predominantly used for array data, for the current implementation we have chosen an array-based storage. The same storage has been used successfully in use cases similar to ours, e.g., European Earth observation project [4].

The data vaults approach gives the user the opportunity to continue performing data curation activities since the main data archive is the file-based repository, i.e., the data is kept outside of the DBMS. The data imported to the DBMS is easily invalidated in case of updates. For a new data format version the catalog tables are easily updatable and a new data loader is provided.

C. Filtering stage

During filtering, the efficiency in extracting only relevant data for analysis is the major requirement. For that, we have decided to use a column-oriented instead of a row-oriented architecture. For read-intensive analytic workloads, such as the ones encountered in data warehouses, column-oriented architectures offer an order-of-magnitude performance gain compared to traditional row-oriented architectures.

The performance boost obtained by column-stores is achieved by its vertical partitioning and two major optimizations: late materialization and block iteration. With vertical partitioning each column is stored in an independent file which reduces I/O when only a sub-set of the table's columns needs to be read. For late materialization, columns read from disk are joined together into rows as late as possible during the query processing. Together with block iteration, i.e., multiple values from a column are passed as a block from one operator to the next, vectorized query processing is achieved. The block iteration optimization offers about a factor of 1.5 improvement on average, while late materialization offers about a factor of 3

performance improvement in most of read-intensive analytical processing workloads [12].

On top of that, column-stores also provide efficient secondary indices for in-memory filtering. In the filtering step, the majority of the queries are range selections. Such type of filtering is sped up using secondary indexes such as skip lists [27] or column imprints [29]. Column-imprints are exploited by the column-store used in our *round-table*, it resembles bitmaps that index ranges of values in each cache line of each column. This makes them very efficient in range queries since they allow skipping cache lines that do not contain data for a desired range.

To show our concept, we have implemented our approach in MonetDB, a modern in-memory column-store database system, designed in the late 90's with a proven track record in various fields [20].

D. Complex Analysis

Simple aggregations allow scientists to detect whether the data at their disposal is meaningful or not. If meaningful, the scientist extracts the data for complex analysis. During this phase, the data is often extracted from the database to be used as input by external specialized libraries or systems. This process is highly inefficient since it involves transferring huge amounts of data. We propose the opposite, data is kept in the database and functionality of external specialized libraries is brought in.

External functionality that we decided to bring in are SAGA [1] and climate data operators (CDO) [5]. SAGA is the abbreviation for System for Automated Geoscientific Analyses and it is a Geographic Information System (GIS) software. SAGA has been designed for an easy and effective implementation of spatial algorithms. CDO is a collection of command line operators to manipulate and analyze climate data. It is actively used by climate researchers and it is seen as a list of operations required for climate data manipulation. Through SQL, SciQL and R it is possible to have equivalent functionality to most of the climate data operators (CDO). However, further work is necessary to support, for instance, correlation and covariance, regression, empirical orthogonal functions, interpolation, transformation and construction of climate indices.

In addition to SAGA and CDO we also exploit two special extensions of MonetDB, R integration and the Geospatial module which allows the user to run ad-hoc queries for selecting information using both spatial and geographic information. We will describe them in more detail in the following Sections.

1) R integration: The R environment for statistical computing [28] is one of the most popular statistical software packages. One of the core strengths of R is its collection of thousands of contributed packages covering all aspects of statistical analysis. For example, the `gstat` [26] and `geoR` [16] packages provide powerful multivariate or model-based statistical analysis for geospatial data sets [26], [16].

MonetDB supports two different methods of integrating statistical analyses in R: The MonetDB.R client program [25] and the embedded R operators [24]. The MonetDB.R client is the one exploited for our study. It allows R users to connect to a MonetDB server from an R shell or program, run arbitrary queries, and retrieve the results.

2) Geospatial functionality: MonetDB has an SQL interface to the Simple Feature Specification of the Open Geospatial Consortium(OGC) [10] with support for the objects and functions defined in the specification. The spatial query model that is used by MonetDB follows the well established two-step approach of **filtering** and **refinement**.

In the filtering step, the majority of points that do not satisfy the spatial predicate for a given geometry G are identified and disregarded using a fast approximation of the predicate. MonetDB performs the filtering using the column imprints [29].

The refinement step operates on the results of the filtering step that produced a superset of the solution. During this step, the spatial predicate is evaluated against the precise geometry G. This can be very expensive, especially when the geometries are complex. Currently, we are experimenting with GPU technology to speed up the refinement.

The efficiency of the geospatial functionality has been tested in the context of point cloud data using the *Actueel Hoogtebestand Nederland 2* (AHN2) [6] data set which is the Dutch elevation map. The results are compared with a file-based solution, Rapidlasso LAStools and PostGIS [31].

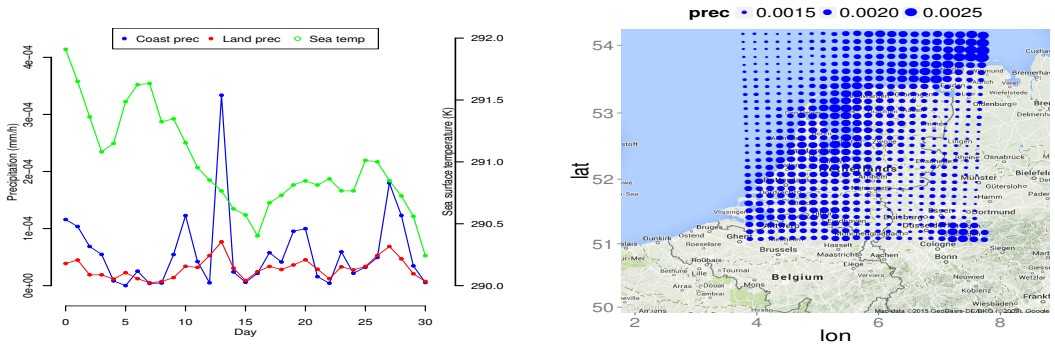
III. Modern Data Exploration

We illustrate the application and efficiency of our system with two use cases, one for meteorology and one for urban planning. With these use cases, this section shows how our solution does fast data ingestion and dynamic step-wise exploration where flexibility and performance have a symbiotic relationship. Flexibility is studied in the context of climate monitoring data. This data is directly accessed in its original format. Statistical questions are answered through the R front-end integrated with MonetDB.

A. Climatology use case

Given various measurements of temperature, precipitation, wind, etc. meteorologists often look for correlations that can be used to create better climate and weather models. An example would be studying the effect of sea surface temperature on precipitation in coastal areas. In combination with geographic data it is possible to analyze the surroundings and weather conditions for different geographic areas. Another example could be analyzing precipitation during a month, or searching for the geographic areas with highest precipitation.

In this scenario, we used data provided by the Royal Netherlands Meteorological Institute (KNMI) [7]. The data is stored in NetCDF format and it represents one month of measurements of precipitation and sea surface temperature



(a) Correlation of coast and land precipitation

(b) Cumulative precipitation

Fig. 2: Coastal and Land climate

over a gridded area covering the Netherlands. It contains one measurement per day for each grid cell.

In addition, there are a land-sea mask and measurements of distance to the coast. The land-sea mask is a value between 0 and 1 that measures the percentage of land in the grid cell, thus a value larger than 0.5 is considered as land. The distance to coast is measured in kilometers and a value of less than 10 kilometers denotes a coastal area.

Our step-wise approach goes as follows. First, using data vaults, we attach the NetCDF data to the database by storing its location and metadata often stored in the file header. The attachment does not imply data loading, therefore, it is possible to explore large NetCDF repositories. Then using the NetCDF catalog different meteorological measurements inside the database are filtered, aggregated and combined. The result of these operations are saved into a table, array or materialized view to allow future usage without the need to re-calculate them. For flexibility, the user might opt for a non-materialized view so the operations are re-computed and their results contain the underlying changes to the NetCDF repositories' data.

Using MonetDB's R front-end we query the NetCDF data, i.e., actual data is loaded at this point. Through the R dplyr package *MonetDB.R* [25] filtering, grouping, and aggregation operations are pushed into the database for execution and only the result is moved to the R environment. In this way, large amounts of data can be inspected and filtered out before transferring it to the R front-end to identify, for instance, correlations. Such a feature was exploited to plot the correlation of coastal and inland precipitation in Figure 2a.

For further advanced analysis or visualization we use the functionality available in the R packages. The example R code below plots the cumulative precipitation points, aggregated inside the database, in the area of the Netherlands, c.f. Figure 2b.

```
# access to Google maps
map<-get_map(location='Netherlands',
    zoom=7)

mapPoints<-ggmap(map)+
    geom_point(
        data=cumprecdf,
        aes(x=lon, y=lat, size=prec),
        colour="blue"
        ) +
    ggtitle("Cumulative precipitation")
```

After loading, the NetCDF data is available to any front-end of MonetDB. For instance, the maximum precipitation measurements (mm/h) over a 10 days period are given by the aggregate query in Figure 3 which is executed through the SQL front-end an uses the same NetCDF data.

B. Urban planning use case

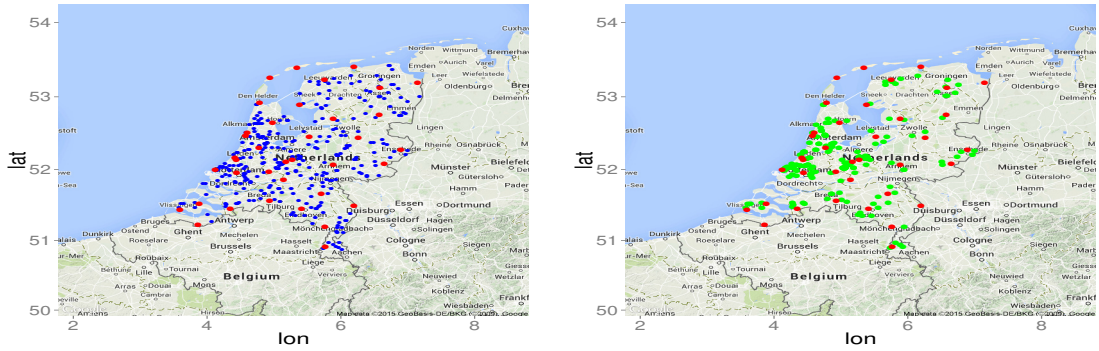
Urban areas are exposed to the same climate as regional areas, however, the urban characteristics can influence thermal comfort of citizens at local scale [30]. Ketterer and Matzarakis [23] showed that air temperature is not enough to quantify the intra-urban spatial variability of climate with respect to human thermal comfort. Human thermal comfort depends on the combined effect of air temperature, air humidity, wind speed, and radiation [19], but wind speed and radiation are affected by the urban geometry such as the height and spacing of buildings. Hence, to improve urban thermal environment it is necessary to understand spatial and temporal variability of local climate [30].

Next to detail climate, climate researchers also need to use spatial parameters like mean building height and the sky view factor (SVF) [15]. These parameters are calculated using a Digital elevation model (DEM). Such an approach was followed by the authors in [30] to study thermal and spatial variability of an urban heat island. They used a network of weather stations located at the city center of Rotterdam and a

```
sql> select time, min(value) as min_prec, max(value) as max_prec
more> from precip1 where time < 10
more> group by time;
```

time	min_prec	max_prec
0	0	0.00078667711932212114
1	-2.8627886727861096e-09	0.0005163105670362711
2	-1.7188804690704274e-07	0.00025988367269746959
3	-4.8412328368385715e-08	0.00030500376544706234
4	-7.5764297946534498e-08	0.00020681128080468625
5	4.4151689249094517e-11	0.00043023109901696444
6	-5.2982029874470982e-10	0.00020797159231733531
7	-7.6117508740480844e-08	0.00017254076374229044
8	-3.5326870317931025e-08	0.00011009947775164619
9	-8.3623298507973232e-08	0.00028709875186905265

Fig. 3: Extreme precipitation



(a) KNMI and WU stations

(b) WU stations 0.1 degree distance from a KNMI station

Fig. 4: Geospatial data combined with Climate data

point cloud data set as DEM. Point Cloud data is collected using airbone laser scanning. Airborne laser scanning is a remote sensing technology which collects large amounts of point data to be the base of digital surface or elevation models.

For the urban planning scenario we combine data from KNMI stations with data from Weather Underground (WU) stations [8] which are the ones used in [30]. We enrich this data with spatial information using the *Actueel Hoogtebestand Nederland 2* (AHN2) [6] point cloud data set which is the Dutch elevation map. AHN2 is composed by 640 billion points stored in 64 000 LAZ files.

The Netherlands has few hundred of WU stations which log information about wind direction, temperature, humidity, precipitation, etc.. Figure 4a shows the location of these stations with KNMI stations colored in red and WU stations colored in blue. The WU measurements are logged every 10 minutes and provided as open data. Using measurements between January 1st 2014 until March 1st 2015 stored as CSV file, data is loaded iteratively, i.e., only the required columns are loaded. For example, only the columns stationID, precipitation, time, location (latitude and longitude coordinates) are loaded. The remaining columns are loaded upon request. Such an approach allows partial vertical loading of wide CSV files, i.e., large number of attributes, into tables with few columns.

Exploiting the fact that KNMI weather stations are highly reliable, their measurements are used to determine the deviation on the WU stations measurements, and thus identify possible heat islands. Using geographic information and functionality from the geospatial module (for polar and Cartesian coordinates conversion and distance calculation) and the R module, Figure 4b plots the WU stations (the ones in green) within 0.1 (polar coordinate) degree distance from each KNMI station (the ones in red).

Our search zooms in into the city center of Rotterdam and checks the temperature difference between WU stations and the nearest KNMI station. Using data from February 2015 the monthly average temperature difference at 4PM is shown in Figure 5a. All stations report on average higher temperatures than the KNMI station. Station *IROTTERD21* has the highest difference.

A study of each station surroundings is necessary for the identification of heat islands. A long with the experiments in [30] we use AHN2 data as DEM. In our approach a flat table is used for storing point cloud data, where a different column is used for storing the X, Y, Z coordinates. As a result, each point is stored as a different tuple in the flat table. Such a storage model facilitates integration of point cloud data with other data sets and exploits the IO efficiency of column-stores.

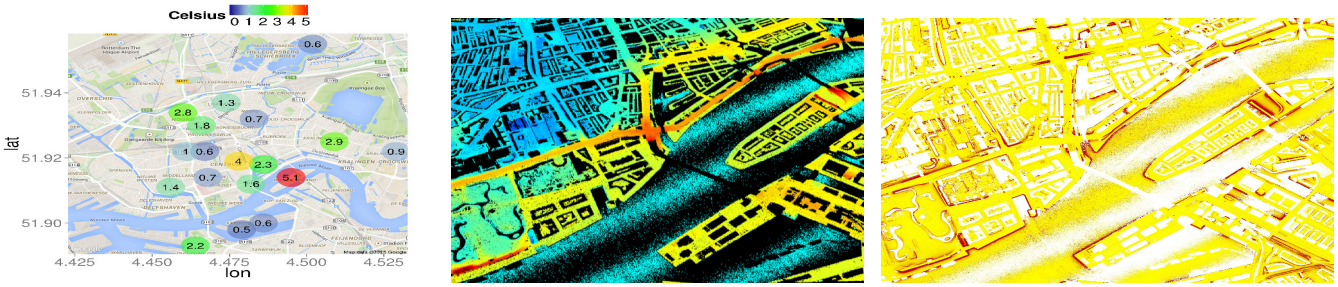
AHN2 has a sample density of 6 to 10 points per square meter. Its density is perfect to create a sky view factor (SVF) of Rotterdam city center. Using the MonetDB geospatial module, we extracted from AHN2 all points comprising Rotterdam city center. The height map of these points is shown in Figure 5b, the points are colored by height, with blue for the lowest points and red for the highest points. The points are loaded into SAGA using the functionality *Import Tables from SQL query*. They are used as input to determine the sky view factor.

Using the SVF function from SAGA with the same parameters as the ones used in [30] we created a plot for Rotterdam city center, shown in Figure 5c. The white areas have high SVF value, yellow medium and red means low SVF. Based on the height map, SVF information, *IROTTERD21's* location (courtyard) and elevation (0 meters), which reports zero average wind speed, it seems we are in the presence of a heat island.

C. A glimpse of the system efficiency

For all these use cases the main focus was flexibility. Nevertheless, efficiency of the solution is already visible in some components, mostly in the geospatial module. The load and indexing of the entire AHN2 data set takes around 18 hours. Rapidlasso LAStools [11] in a similar machine takes around 23 hours to prepare the same data set for efficient querying [31].

With higher cost of data preparation, LAStools is not able to out-perform MonetDB for the Rotterdam city selection out of AHN2. MonetDB takes 5.47 seconds with 1 thread or 3.86 seconds with 16 threads to extract 37345849 points while Rapidlasso LAStools takes 6.08 seconds if it outputs las format, but for text format which is necessary to import the data into SAGA library it takes 73,06 secs. A detailed



(a) Temperature Difference (b) Point Cloud view (c) Sky view factor

Fig. 5: Rotterdam city center

system performance study is out of the scope of this paper and it will be released in a future publication.

IV. Conclusion

In this paper we present our efforts to develop a flexible and efficient data management layer for geo-spatial data analysis. An architecture is presented which provides fast data ingestion through different data loaders, tabular and array-based storage, and a set of frond-ends to explore the data sets in various ways.

The data is loaded from its original format upon request, e.g., for climate use cases stored in NetCDF files only the variables of interest are loaded into the DBMS. Hence, it allows exploration of large NetCDF repositories without a priori data loading. Through a tabular and array-based storage scheme it enables spatial and temporal operations and analyses while providing semantic properties management stored as simple relational tables.

All data is accessible through different front-ends such as R, SciQL and SQL. R is used to express statistical analysis, simple geographic summaries, and for visualization. N-dimensional array based functions are expressed through SciQL. SQL as declarative language is ideal to express complex adhoc queries and data flows. The connection with QGIS [9] for efficient visualization of spatial data is under development [14].

Our open-source solution facilitates data exploration for climatology and urban planning. It fills the gap between the needs of various eScience applications and the available data management technologies and file-based solutions.

V. Future Plans

Currently we are working on the bridge between SAGA and MonetDB. Functionality to convert point cloud data to vector data and to 2D/3D raster data is also under development [18]. In a future publication we will study the advantages and disadvantages of our solution compared to Hadoop-framework solutions and the Hierarchical Data Format (HDF5) software library.

References

[1] http://saga-gis.org/en/index.html.
[2] http://en.wikipedia.org/wiki/Conceptual_schema.
[3] http://www.unidata.ucar.edu/software/netcdf/.
[4] http://www.earthobservatory.eu/SciQL_and_Data_Vault.
[5] https://code.zmaw.de/projects/cdo.
[6] http://www.ahn.nl.
[7] http://www.knmi.nl/index_en.html.
[8] http://www.wunderground.com/.
[9] http://www.qgis.org.
[10] Open Geospatial Consortium. OpenGIS Implementation Specification for Geographic information - Simple feature access - Part 2: SQL option. OpenGIS Implementation Standard, 2010.
[11] rapidlasso GmbH. http://rapidlasso.com/, 2014.
[12] D. J. Abadi, S. Madden, and N. Hachem. Column-stores vs. row-stores: how different are they really? In *Proceedings of the ACM SIGMOD*, 2008.
[13] G. Aloisio and S. Fiore. Towards exascale distributed data management. *Int. J. High Perform. Comput. Appl.*, 2009.
[14] F. Alvanaki, R. Goncalves, M. Ivanova, M. Kersten, and K. Kyzirakos. GIS navigation boosted by column stores. *PVLDB*, 2015.
[15] L. Barring, J. O. Mattsson, and S. Lindqvist. Canyon geometry, street temperatures and urban heat island in malmö, sweden. *Journal of Climatology*, 1985.
[16] P. J. Diggle and P. J. R. Jr. *Model Based Geostatistics*. Springer, New York, 2007.
[17] S. Fiore, A. D'Anca, C. Palazzo, I. Foster, D. Williams, and G. Aloisio. Ophidia: Toward big data analytics for escience. *Procedia Computer Science*, 2013. International Conference on Computational Science.
[18] R. Goncalves, M. Ivanova, M. Kersten, and et al. Big data analytics in the geo-spatial domain, 2014. Proceedings of Target Conference 2014 Big Data Across Disciplines.
[19] P. Hoppe. The physiological equivalent temperature – a universal index for the biometeorological assessment of the thermal environment. *International Journal of Biometeorology*, 1999.
[20] S. Idreos, F. Groffen, N. Nes, S. Manegold, S. Mullender, and M. Kersten. Monetdb: Two decades of research in column-oriented database architectures. *IEEE Data Engineering Bulletin*, pages 40–45, 2012.
[21] M. Ivanova, Y. Kargin, and et al. Data Vaults: A Database Welcome to Scientific File Repositories. SSDBM, 2013.
[22] M. Kersten, Y. Zhang, M. Ivanova, and N. Nes. SciQL, a Query Language for Science Applications. AD, 2011.
[23] C. Ketterer and A. Matzarakis. Human-biometeorological assessment of the urban heat island in a city with complex topography – the case of stuttgart, germany. *Urban Climate*, 2014.
[24] H. Mühleisen. Embedded R in MonetDB, 2014. https://www.monetdb.org/content/embedded-r-monetdb.
[25] H. Mühleisen and T. Lumley. Best of both worlds: Relational databases and statistics. In *Proceedings of the 25th SSDBM*, 2013.
[26] E. J. Pebesma. Multivariable geostatistics in S: the gstat package. *Computers & Geosciences*, 30:683–691, 2004.
[27] W. Pugh. Skip lists: A probabilistic alternative to balanced trees. In *Algorithms and Data Structures*, Lecture Notes in Computer Science, pages 437–449. Springer Berlin Heidelberg, 1989.
[28] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2014.
[29] L. Sidirourgos and M. L. Kersten. Column imprints: a secondary index structure. In *Proceedings of the ACM SIGMOD*, 2013.
[30] L. van Hove, C. Jacobs, B. Heusinkveld, J. Elbers, B. van Driel, and A. Holtslag. Temporal and spatial variability of urban heat island and thermal comfort within the Rotterdam agglomeration. *Building and Environment*, 2015.
[31] P. van Oosterom, O. Martinez-Rubi, and et al. Massive point cloud data management: design, implementation and execution of a point cloud benchmark. *Computer Graphics*, 2015.