Naphtali David Rishe
Director, High Performance Database Research Center
Director, NSF Industry-University Cooperative Research Center
http://HPDRC.FIU.edu http://TerraFly.FIU.edu

FLORIDA INTERNATIONAL UNIVERSITY
Miami's public research university

Indexing Geospatial Data with MapReduce

Naphtali Rishe[+], Vagelis Hristidis[+], Raju Rangaswami[+],
Ouri Wolfson[*], Howard Ho[**], Ariel Cary[+], Zhengguo Sun[+], Lester Melendes[+]

[+]School of Computing and Information Sciences, Florida International University
[*]University of Illinois at Chicago [**] IBM Almaden Research Center

Sponsored by: NSF Cluster Exploratory (CluE)

Content

Introduction

- Spatial databases mainly store:
  - Raster data (satellite/aerial digital images), and
  - Vector data (points, lines, polygons).
- Traditional sequential computing models may take excessive time to process large and complex spatial repositories.
- Emerging parallel computing models, such as MapReduce, provide a potential for scaling data processing in spatial applications.

Introduction (cont.)

- MapReduce is an emerging massively parallel computing model (Google) composed of two functions:
  - Map: takes a key/value pair, executes some computation, and emits a set of intermediate key/value pairs as output.
  - Reduce: merges its intermediate values, executes some computation on them, and emits the final output.
- Here we present our experiences in applying the MapReduce model to:
  - Bulk-construct R-Trees (vector)
  - Compute aerial image quality (raster)
  - Extract United States Governmental Organizational Hierarchies

2. TerraFly

Geospatial Applications Suite

Developed and Maintained by

The High Performance Database Research Center (HPDRC)

Florida International University (FIU)

FIU-HPDRC Expertise

- Database aspects:
  - Data visualization
  - Spatial databases
  - Internet-distributed heterogeneous databases
  - Database design methodologies
  - Information analysis

TerraFly

- Geospatial mapping solution
- Web-Based
- Customized to Industry needs
- Drill down to local information

TerraFly

GIS Solutions Based on New Generation Technology

- Platform
  - GIS-like Internet visualization
  - Open architecture, GIS-oriented API provider
  - 40 TB database of aerial imagery and spatial data
  - NSF and NASA funded technology
- Service
  - Professionally customized to domain requirements
  - Comprehensive and expert service

Mapping Solutions

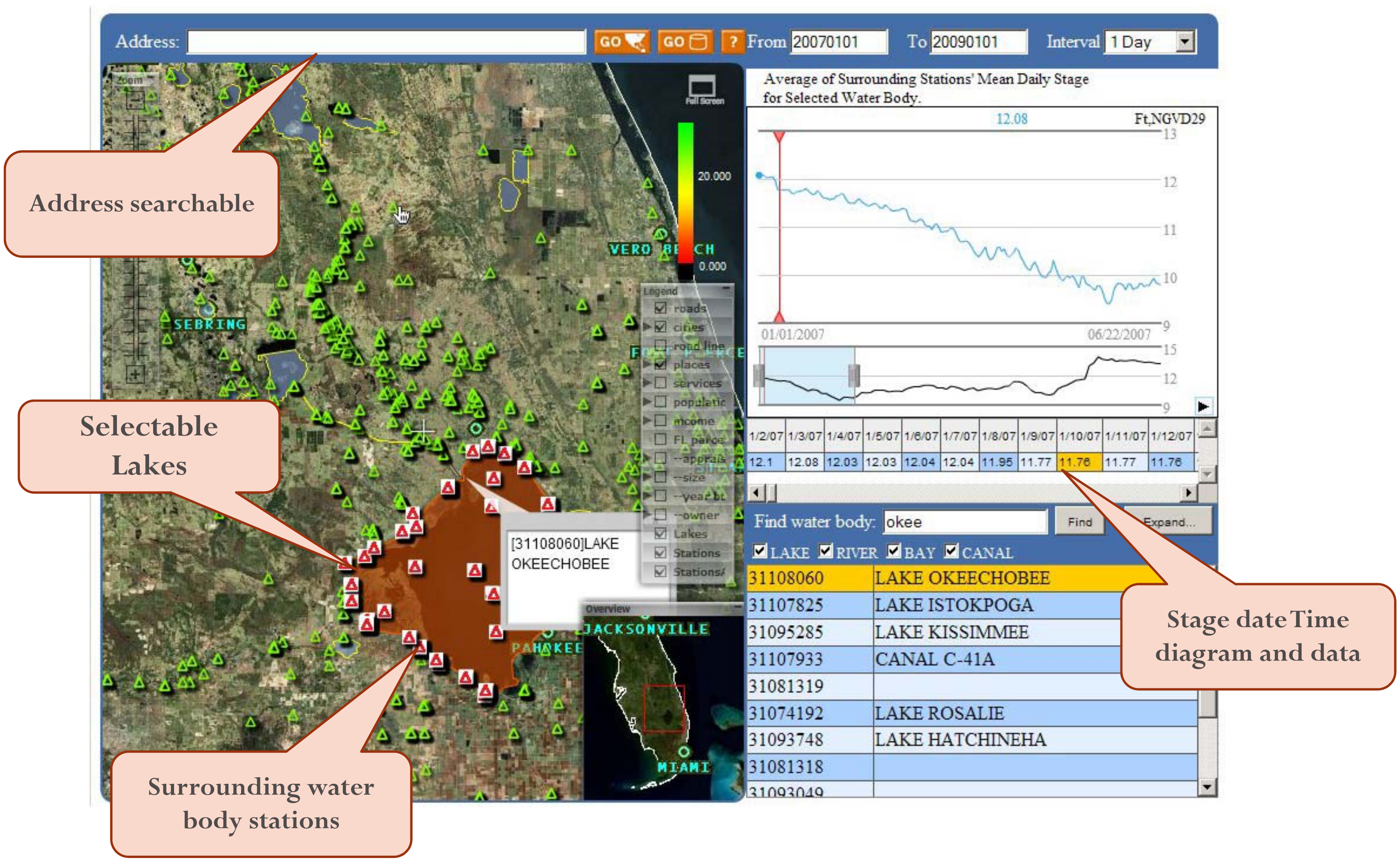
Customized to industry needs. Sample applications:

- The **Hydrology** application shows the mean water level over time over water bodies

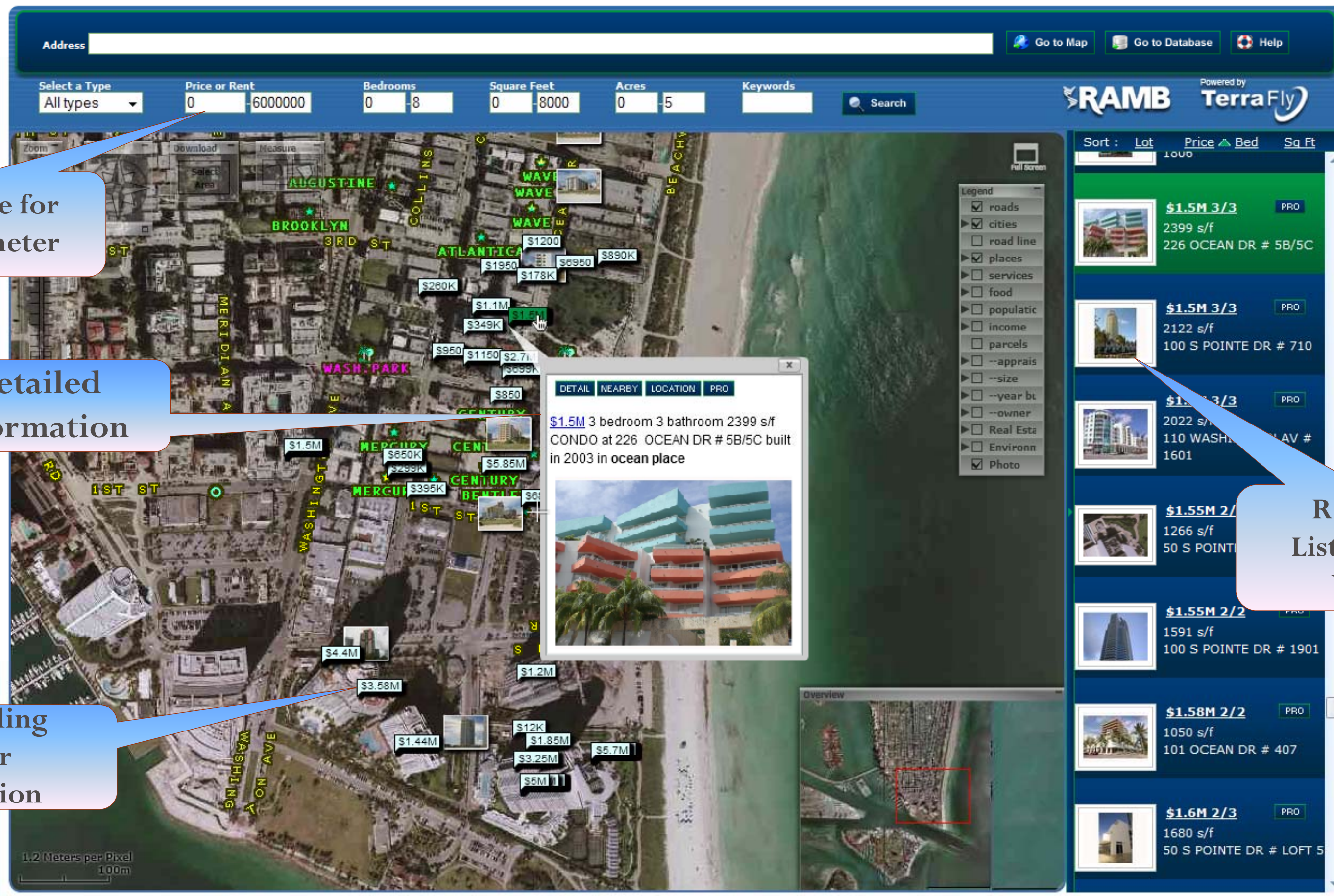
- The **Real Estate** application facilitates visua;ization of listings and allows complex queries via user-friendly interface.

Hydrology Application

Average of Surrounding Stations' Mean Daily Stage for Selected Water Body.



Real Estate



Searchable for any parameter

Detailed information

Surrounding Realtor information

Real Estate Listings in the Vicinity

2. Solving Spatial Problems in MapReduce

- R-Tree Index Construction
- Aerial Image Processing
- Midas Government Domain Application

MapReduce (MR) R-Tree Construction

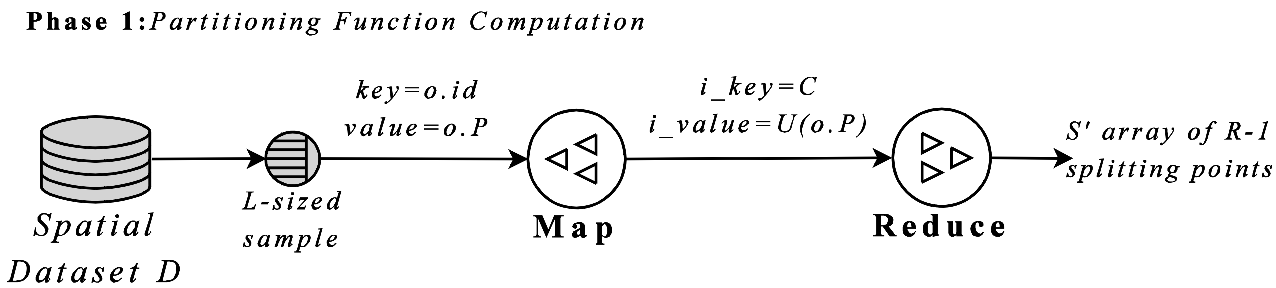
- R-Tree Bulk-Construction
  - Every object *o* in database *D* has two attributes:
    - *o.id* - the object's unique identifier.
    - *o.P* - the object's location in some spatial domain.
  - The goal is to build an R-Tree index on *D*.

- MapReduce Algorithm
  1. Database partitioning function computation (**MR**).
  2. A small R-Tree is created for each partition (**MR**).
  3. The small R-Trees are merged into the final R-Tree.

Phase 1 – Partitioning Function

- Goal: compute *f* to assign objects of *D* into one of *R* possible partitions s.t.:
  - *R* (ideally) equally-sized partitions are generated (minimal variance is acceptable).
  - Objects close in the spatial domain are placed within the same partition.
- Proposed solution:
  - Use *Z-order* space-filling curve to map spatial coordinate *samples* (3%) into an sorted *sequence*.
  - Collect splitting points that partition the *sequence* in *R* ranges.

Phase 1 – Partitioning Function



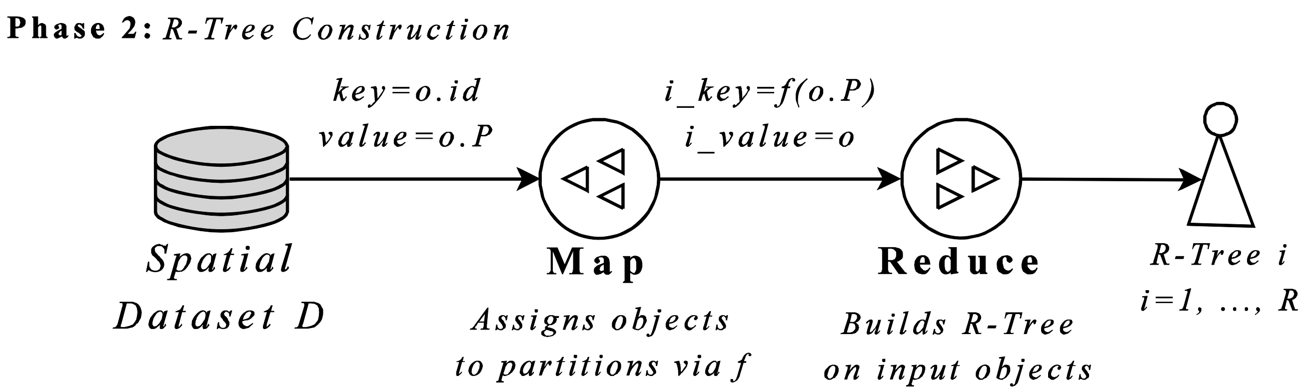
Map and Reduce inputs/outputs in computing partitioning function *f*.

Function	Input: (Key, Value)	Output: (Key, Value)
Map	(*o.id*, *o.P*)	(*C*, *U*(*o.P*))
Reduce	(*C*, *list*(u_i, *i*=1, .., *L*))	*S´*

Where:

- *o* is an spatial object in the database.
- *C* which is a constant that helps in sending Mappers' outputs to a single Reducer.
- *U* is a space-filling curve, e.g. Z-order value.
- *S*' is an array containing R-1 splitting points.

Phase 2 - R-Tree Construction in MR



MapReduce functions in constructing R-Trees.

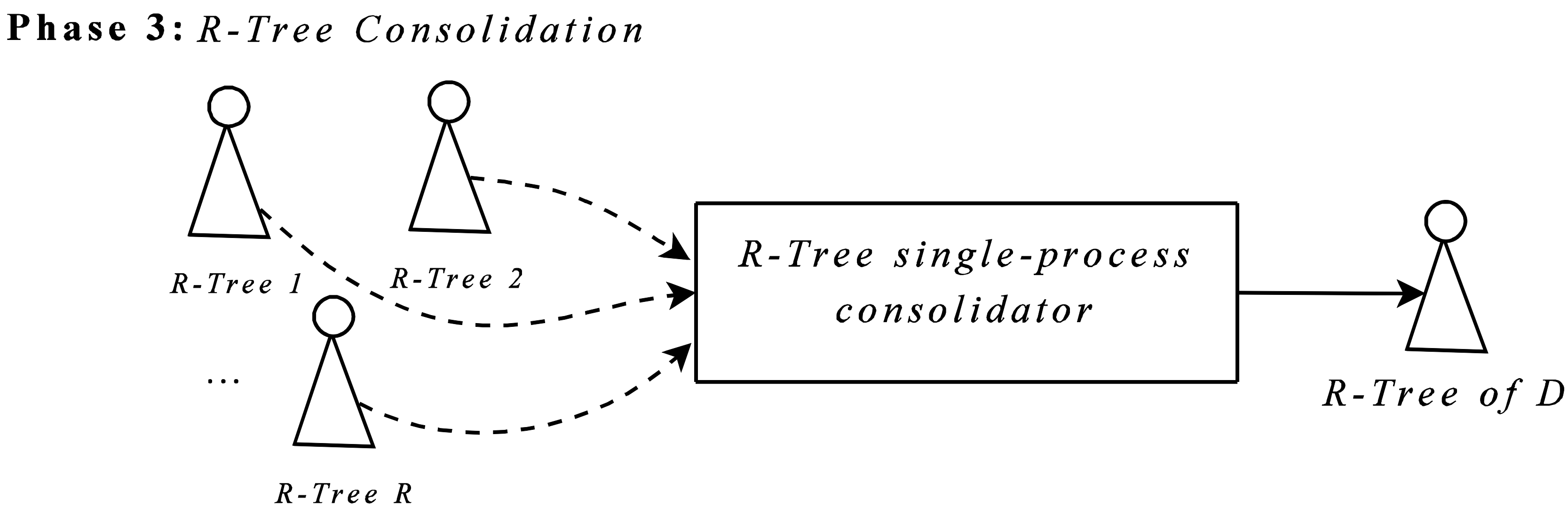
Function	Input: (Key, Value)	Output: (Key, Value)
Map	*(o.id, o.P)*	*(f(o.P), o)*
Reduce	*(f(o.P), list(*$o_{i,\ i=1,..,A}$*))*	*tree.root*

Where:

- *o* is an spatial object in the database.
- *f* is the partitioning function computed in Phase 1.
- *Tree.root* is the R-Tree root node.

Phase 3 - R-Tree Consolidation

- Sequential process



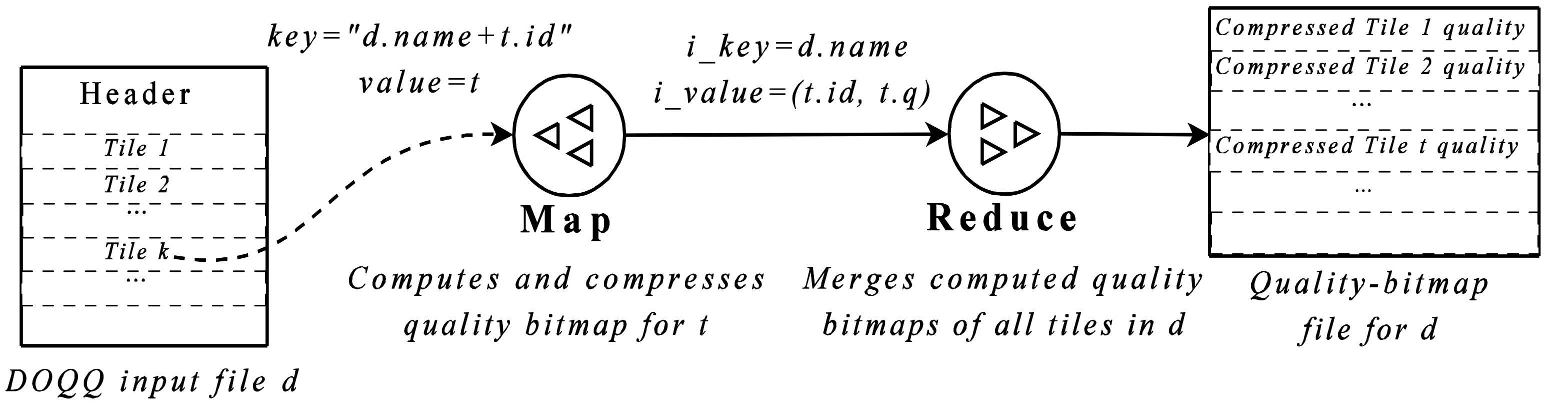
Aerial Image Processing

- R-Tree Index Construction
- **Aerial Image Processing**
- Midas Government Domain Application

Image Processing in MapReduce

- Aerial Image Quality Computation
  - Let *d* be an orthorectified aerial photography (DOQQ) file and *t* be a tile inside *d*, *d.name* is *d*'s file name and *t.q* is the quality information of tile *t*.
  - The goal is to compute a quality bitmap for *d*.
- MapReduce Algorithm
  - A customized *InputFormatter* partitions each DOQQ file *d* into several splits containing multiple tiles.
  - The Mappers compute the quality bitmap for each tile inside a split.
  - The Reducers merge all the bitmaps that belongs to a file *d* and write them to an output file.

Image Processing in MapReduce



Input and output of map and reduce functions

Function	Input: (Key, Value)	Output: (Key, Value)
Map	(*d.name+t.id*, *t*)	(*d.name*, (*t.id*,*t.q*))
Reduce	(*d.name*, *list*(*t.id*,*t.q*))	*Quality-bitmap of d*

Where:

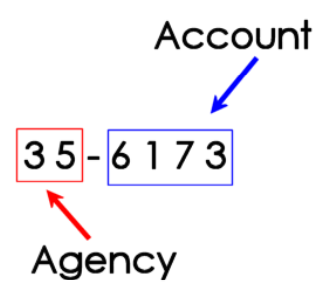
- *d* is a DOQQ file.
- *t is a tile in d*.
- *t.q* is the quality bitmap of *t*.

Midas Government Domain Application

- R-Tree Index Construction
- Aerial Image Processing
- **Midas Government Domain Application**

Linking USASpending.gov with OMB Earmarks

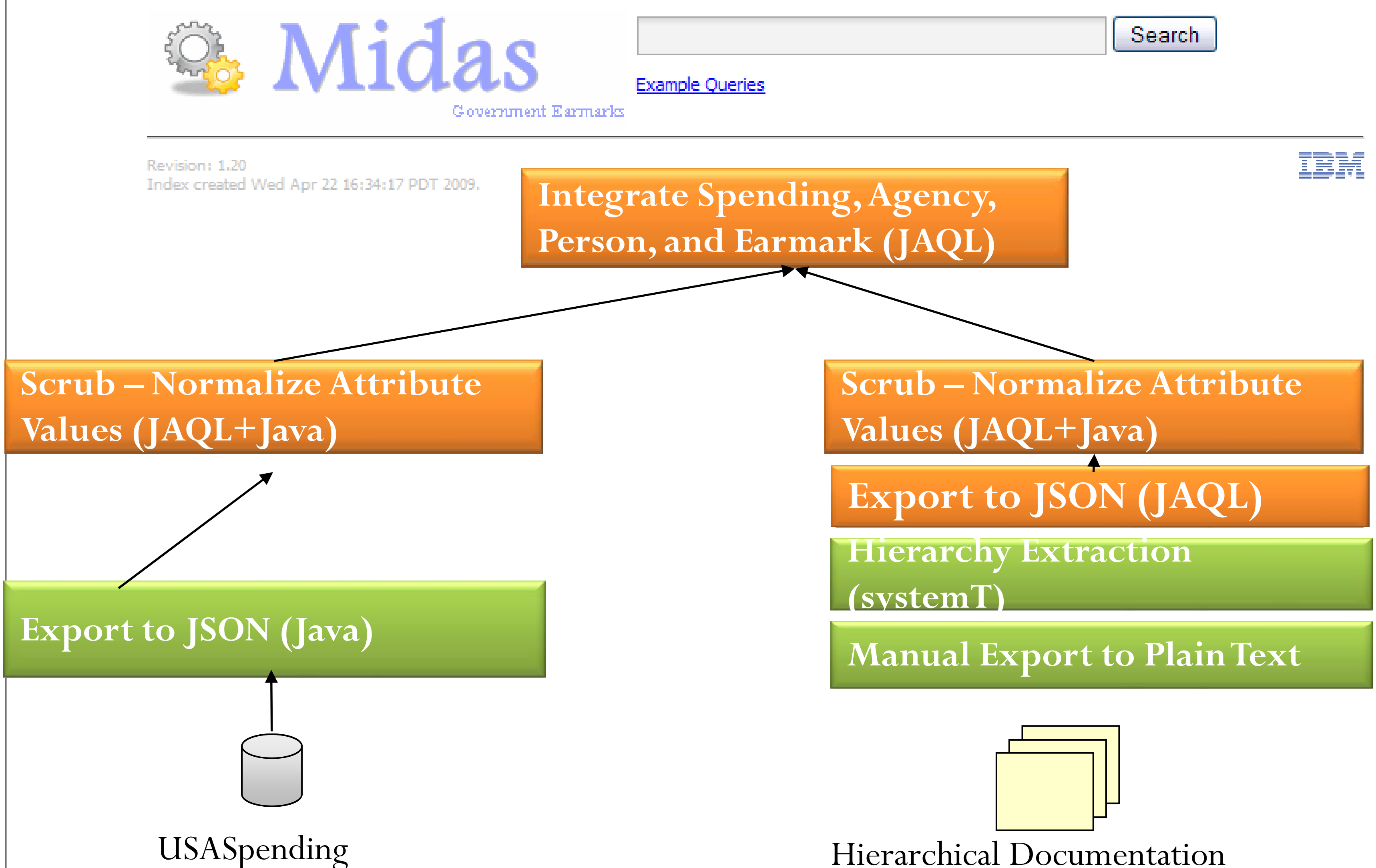
- Organizational/agency hierarchy must be established in order to properly attribute spending and earmark appropriation
- Midas Earmark records do not contain explicit agency information but, web records do.
- Midas Earmark records contain a 6 digit TAS - Treasury Account Symbol
  - First 2 digits = Agency
  - Last 4 digits = Account



Linking USASpending.gov with OMBEarmarks

- Common attributes for linking:
  - Congressional Districts
  - Treasury Account Symbols (TAS)
- We were able to extract semantic agency information from OMB Earmarks we can facilitate linkage.
- This allowed us to obtain finer grained results and use FPDS codes and hierarchical information found in NIST SP 800-87.

Midas for USASpending.gov



Example Queries:

- What percentage of Defense Department spending comes from Earmarks?
- What type of account does the Department of Agriculture spend the least from?
- Are there any vendors who do the majority of their work outside of their own congressional district?
- What percentage of the Department of the Interior's spending goes towards food and food services?
- Is there a congressman who sponsors bills that lead to a particular kind of spending?
  - I.e. Congressman Smith sponsored 100 million in earmarks, of which 90% went to agencies who's primary function is related to national defense.

3. Experimental Results

Experimental Results: Setting

- Data Set

Table 4. Spatial data sets used in experiments*.

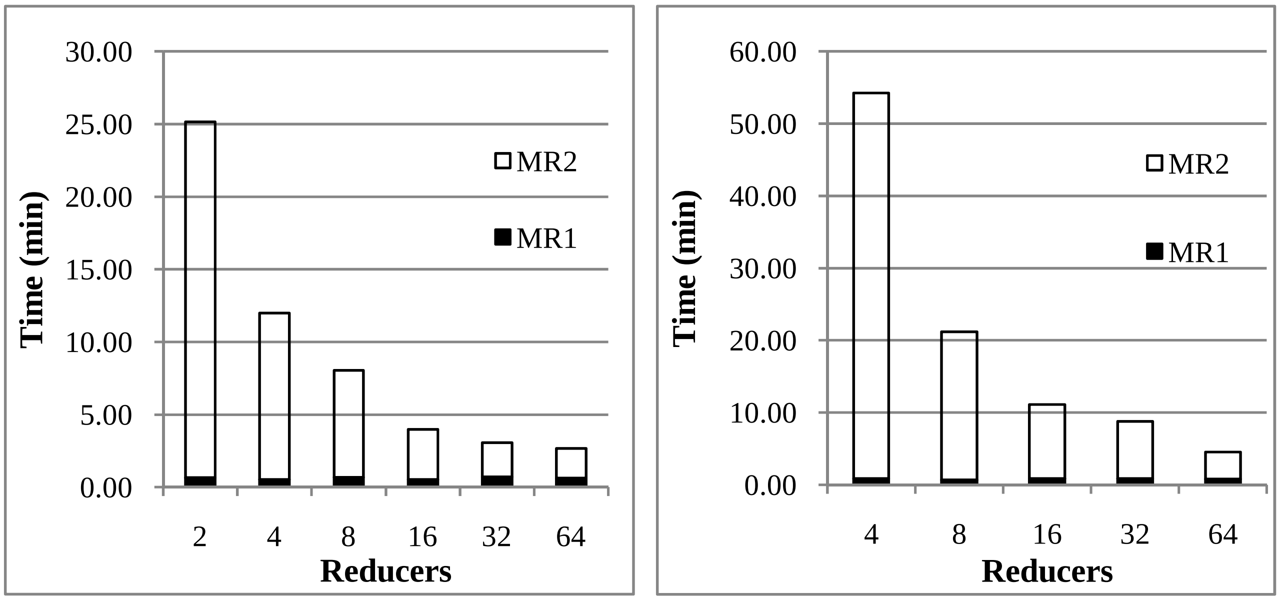
Problem	Data set	Objects	Data size (GB)	Description
R-Tree	***FLD***	*11.4 M*	*5*	Points of properties in the state of Florida.
	YPD	*37 M*	*5.3*	Yellow pages directory of points of businesses mostly in the United States but also in other countries.
Image Quality	***Miami-Dade***	*482 files*	*52*	Aerial imagery of Miami-Dade county, FL (3-inch resolution)

* Data sets supplied by the *High Performance Database Research Center* at Florida International University

- Environment
  - The cluster was provided by the Google and IBM *Academic Cluster Computing Initiative*.
  - The cluster contains around 480 computers running Hadoop - open source MapReduce.

Experimental Results: R-Tree

- R-Tree Construction Performance Metrics



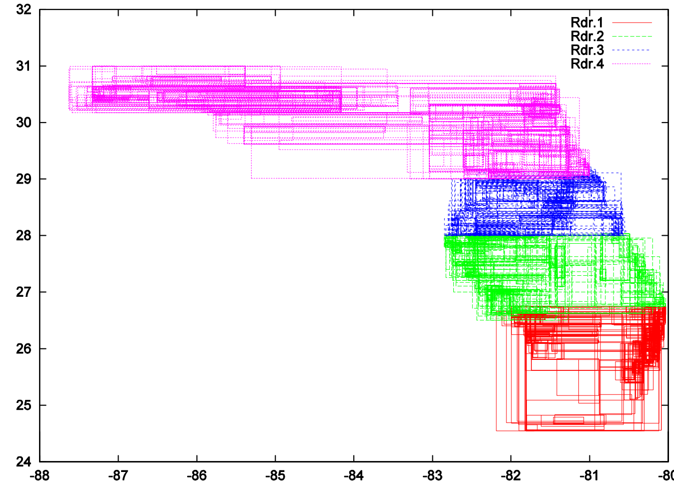
(a) *FLD* data set (b) *YPD* data set

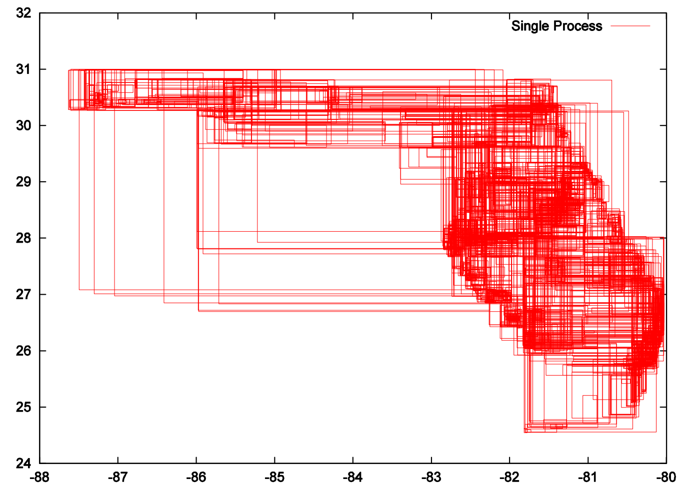
MapReduce job completion times for various number of reducers in phase-2 (MR2).

Experimental Results: R-Tree

- MapReduce R-Trees vs. Single Process (SP)

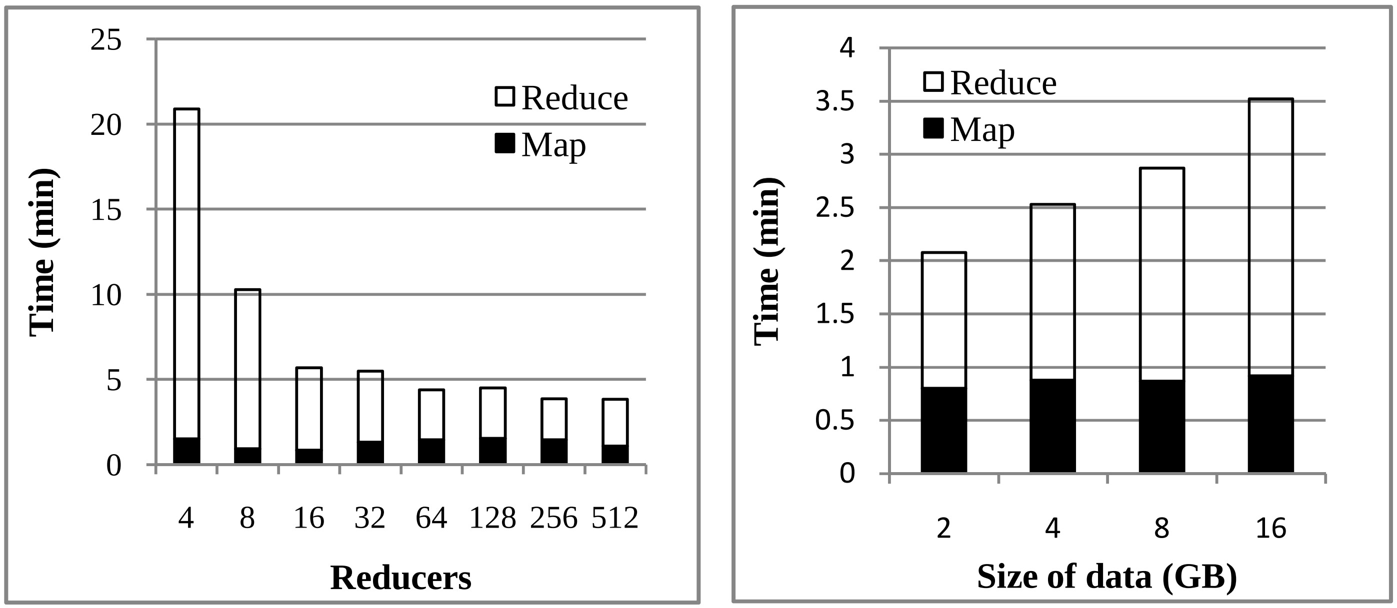
		Objects per Reducer		*Consolidated* R-Tree	
Data set	***R***	**Average**	**Stdev**	**Nodes**	**Height**
FLD	2	5,690,419	12,183	172,776	4
	4	2,845,210	6,347	172,624	4
	8	1,422,605	2,235	173,141	4
	16	711,379	2,533	162,518	4
	32	355,651	2,379	173,273	3
	64	177,826	1,816	173,445	3
	SP	11,382,185	0	172,681	4
YPD	4	9,257,188	22,137	568,854	4
	8	4,628,594	9,413	568,716	4
	16	2,314,297	7,634	568,232	4
	32	1,157,149	6,043	567,550	4
	64	578,574	2,982	566,199	4
	SP	37,034,126	0	587,353	5





Experimental Results: Imagery

- Tile Quality Computation



(a) Fixed data size, variable Reducers (b) Variable data size, fixed Reducers

Fig. 9. MapReduce job completion time for tile quality computation

5. Conclusion

Conclusion

- We employed the MapReduce model to solve two spatial problems on the Google&IBM cluster:
  - (a) Bulk-construction of R-Trees and
  - (b) Aerial image quality computation

  And a document processing for linkage extraction project:
  - (b) MIDAS Government domain application
- MapReduce can dramatically improve task completion times. Our experiments show close to linear scalability.
- Our experience in this work shows MapReduce has the potential to be applicable to more complex spatial problems.