GEOSPATIAL DATA INDEXING ANALYSIS AND VISUALIZATION VIA WEB SERVICES WITH AUTONOMIC RESOURCE MANAGEMENT

Yun Lu

Dr. Naphtali Rishe, Major Professor

School of Computing and Information Sciences

Nov 7th 2013

Outline

- Motivation & Problem Statement
- Main Contributions
- Related Work
- Contributions (Breakdown)
  1. sksOpen
  2. GeoCloud
  3. v-TerraFly
- Conclusions and Limitations
- Future Work
- References

Motivation & Problem Statement

- Almost three-quarters (74%) of smartphone owners get real-time location-based information on their phones as of February 2012, up from 55% in May 2011[Zickuhr12]
- Already more than 1.08 billion smartphone users in the world, 91.4 million are from the United States in 2011
- Google Maps currently has more than 350 million users

Motivation & Problem Statement

- Envision an web-based map services that
  1. Find accurate information by query: find nearest 5 hotel with bay view and swimming pool
  2. Spatial Data analysis and share: how house price related with location
  3. Efficiently host web-based map service: balance resource allocation to different tiers to gain the best QoS
- However, several factors affect functionality
  1. Query may take long time if the data is too big
  2. Lack of analysis model and bad visualization implicate the data analysis
  3. Dynamic web workloads and involve multiple CPU and I/O intensive tiers make it challenging to host web-based map service

FIU

Motivation & Problem Statement

- This dissertation tackles

1. Inefficient indexing and query for Top-k nearest Spatial Boolean queries and poor visualization of query results
2. Complicated and fussy geographic visualization and data analysis
3. Inefficiently host web map service evolves multi-tiers

FIU

Outline

- Motivation & Problem Statement
- Main Contributions
- Related Work
- Contributions (Breakdown)
  1. sksOpen
  2. GeoCloud
  3. v-TerraFly
- Conclusions and Limitations
- Future Work
- References

Main Contributions

1. sksOpen: an open-sourced an Online Indexing and Boolean Querying System for Big Geospatial Data
2. GeoCloud: an extra layer running upon the TerraFly map and can efficiently support many different visualization functions and spatial data analysis models
3. v-TerraFly: techniques to predict the demand of map workloads online and optimize resource allocations considering both response time and data freshness as the QoS target

Outline

- Motivation & Problem Statement
- Main Contributions
- Related Work
- Contributions (Breakdown)
  1. sksOpen
  2. GeoCloud
  3. v-TerraFly
- Conclusions and Limitations
- Future Work
- References

FIU

Related Work

- Geographic information retrieval
  - [Jones04] Spirit Spatial Search Engine analyzing the geographic references in text (single field)
  - [Zhou05] propose a hybrid index structure combined with different partitions of space (grid is not efficient as R-Tree)
  - [Hariharan07] multiple R*-trees (more join operation)
- Spatial data analysis and visualization
  - [Johnston01] [O'Sullivan03] analysis on desktop like Esri
  - [Anselin06] GeoDa analysis tools
- Workload prediction and resource management
  - [Huebscher08] A survey of autonomic computing (no web map)
  - [Rao09] Virtual Machines Auto-configuration (identical between tiers)

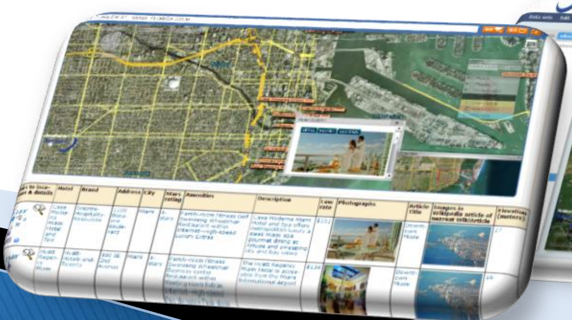
FIU

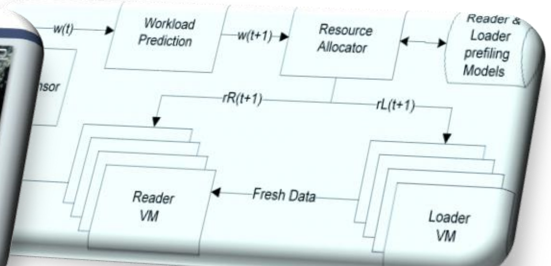
Outline

- Motivation & Problem Statement
- Main Contributions
- Related Work
- Contributions (Breakdown)
  1. sksOpen
  2. GeoCloud
  3. v-TerraFly
- Conclusions and Limitations
- Future Work
- References

sksOpen: Efficient Indexing, Querying and Visualization of Geo-spatial Data

- Integrated with the TerraFly Geospatial database
- Efficient indexing and query engine
- Map Reduce parallel
- Processing Top-k Spatial Boolean Queries
- Provide ergonomic visualization of query results
- Published in [Yun131]





Problem Definition

- Spatial database $D = \{o1, o2, ..., oN\}$;
  - for every $o \in D$ $< p, T >$
- Top-k Spatial Boolean Queries(k-SB) query Q is a triple $< l, k, B >$;
  - *l* is the query location (*spatial constraint*)
  - *k* is the desired output size
  - *B* is the conjunctive Boolean predicate
- *L* is a list of result of the *k-SB* query *Q*

FIU

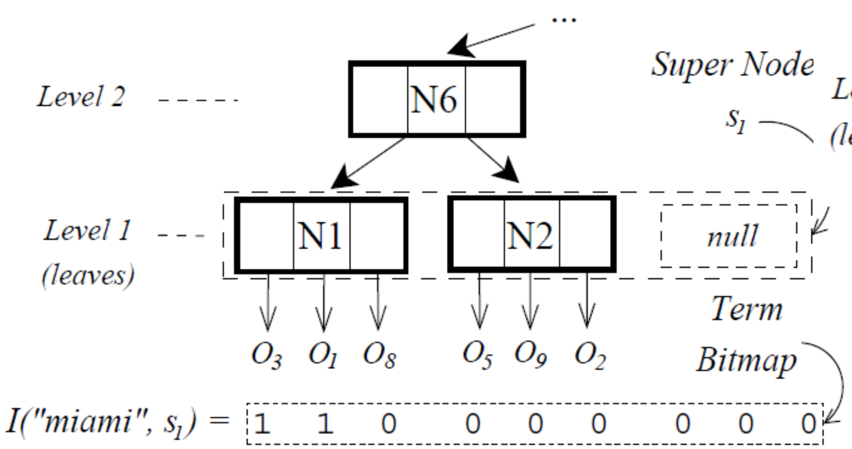
Hybrid Spatial-Keyword Indexing

- Fast retrieval objects even far away
  - R-Tree
- Efficiently filter objects not satisfying keyword constraints
  - Inverted file

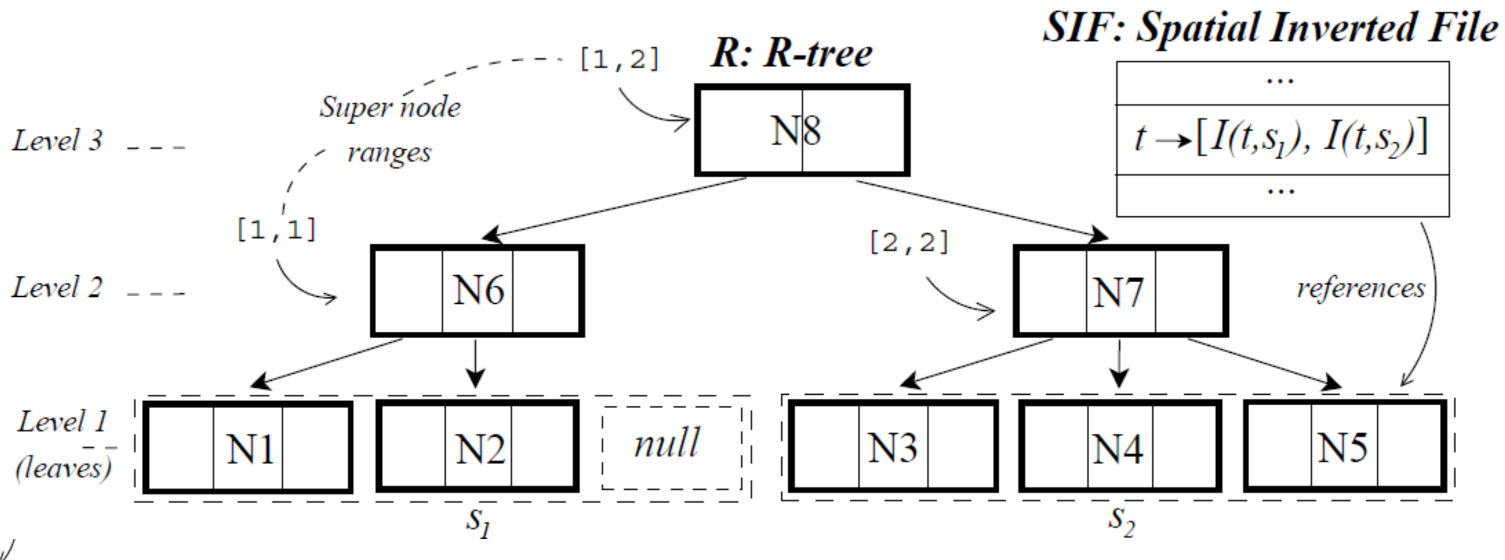
Term	Object List	Term	Object List
backyard (t_1)	$\{o_2, o_3, o_6, o_8\}$	collins (t_4)	$\{o_2, o_6, o_{10}\}$
bathtub (t_2)	$\{o_3, o_5, o_8, o_9\}$	masterbed (t_5)	$\{o_3, o_8, o_{11}\}$
building (t_3)	$\{o_1, o_5, o_7, o_{12}\}$	miami (t_6)	$\{o_1, o_3, o_4, o_{10}\}$

Hybrid Spatial-Keyword Indexing

- Hybrid Spatial Keyword Index [Cary10]



Super node s_1 composed of leaf nodes $[N_1, N_2]$, and term bitmap for *"miami"*.



Term t contains all bitmaps sorted by super node index
Prune if not contain desired B

Bitmap bit operation to speed up query

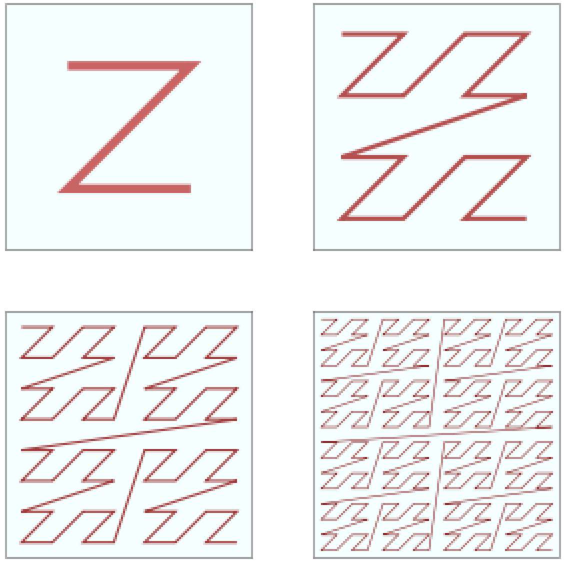
Z-ordering curve

- Z-ordering a function which maps multidimensional data to one dimension while preserving locality of the data points
- Z-ordering can be used to efficiently build a Quad tree for a set of points. The basic idea is to sort the input set according to Z-order
- Z-values for the two dimensional case with integer coordinates $0 \le x \le 7,\ 0 \le y \le 7$

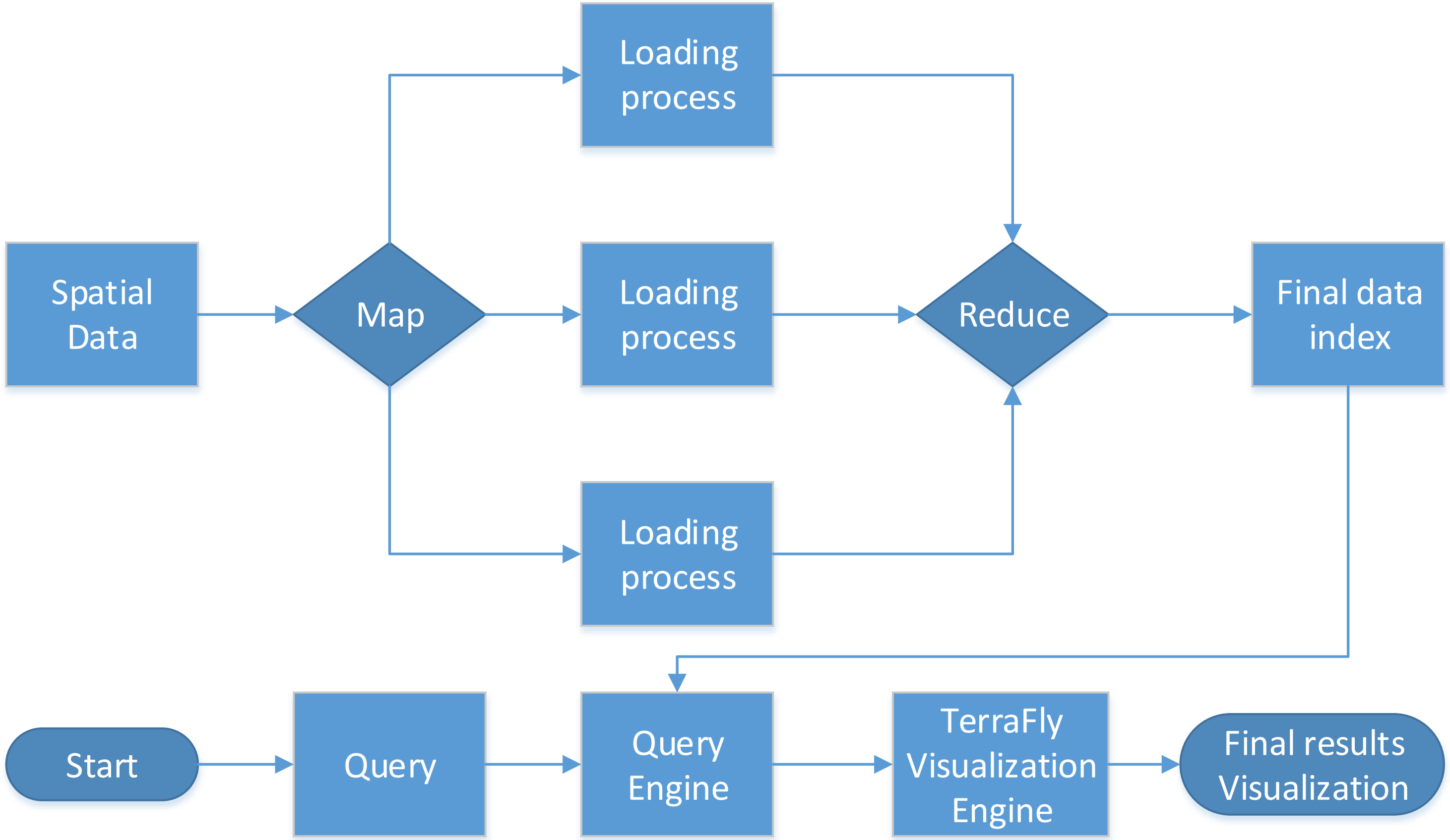
	x: 0 000	1 001	2 010	3 011	4 100	5 101	6 110	7 111
y: 0 000	000000	000001	000100	000101	010000	010001	010100	010101
1 001	000010	000011	000110	000111	010010	010011	010110	010111
2 010	001000	001001	001100	001101	011000	011001	011100	011101
3 011	001010	001011	001110	001111	011010	011011	011110	011111
4 100	100000	100001	100100	100101	110000	110001	110100	110101
5 101	100010	100011	100110	100111	110010	110011	110110	110111
6 110	101000	101001	101100	101101	111000	111001	111100	111101
7 111	101010	101011	101110	101111	111010	111011	111110	111111

interleaving

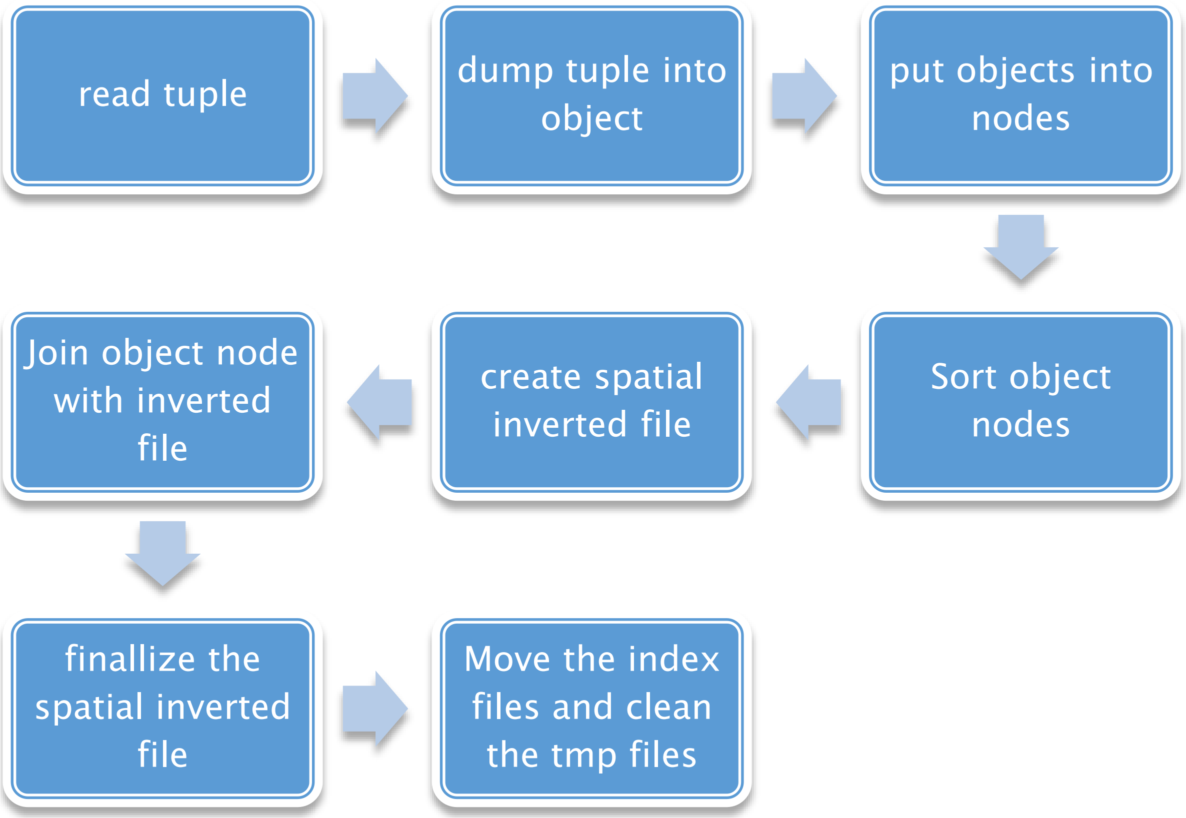
MapReduce parallel design



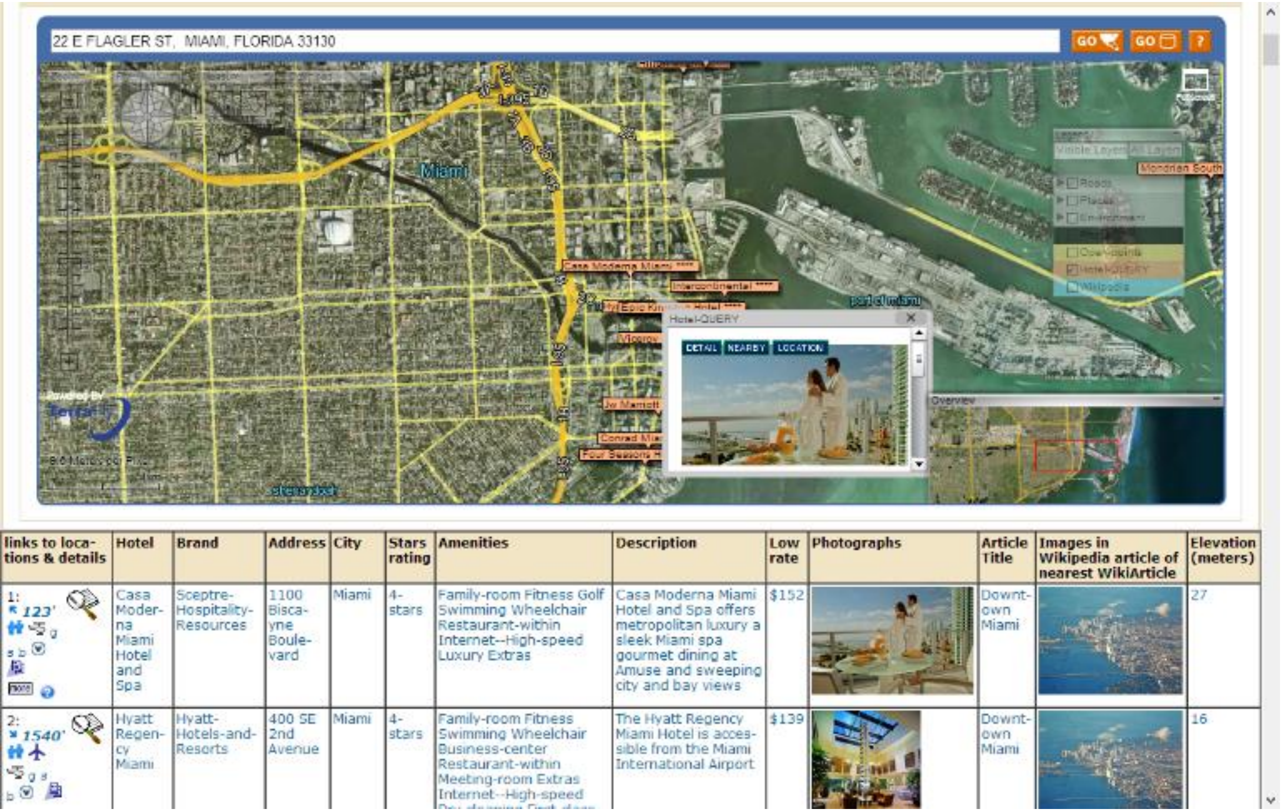
Z-ordering

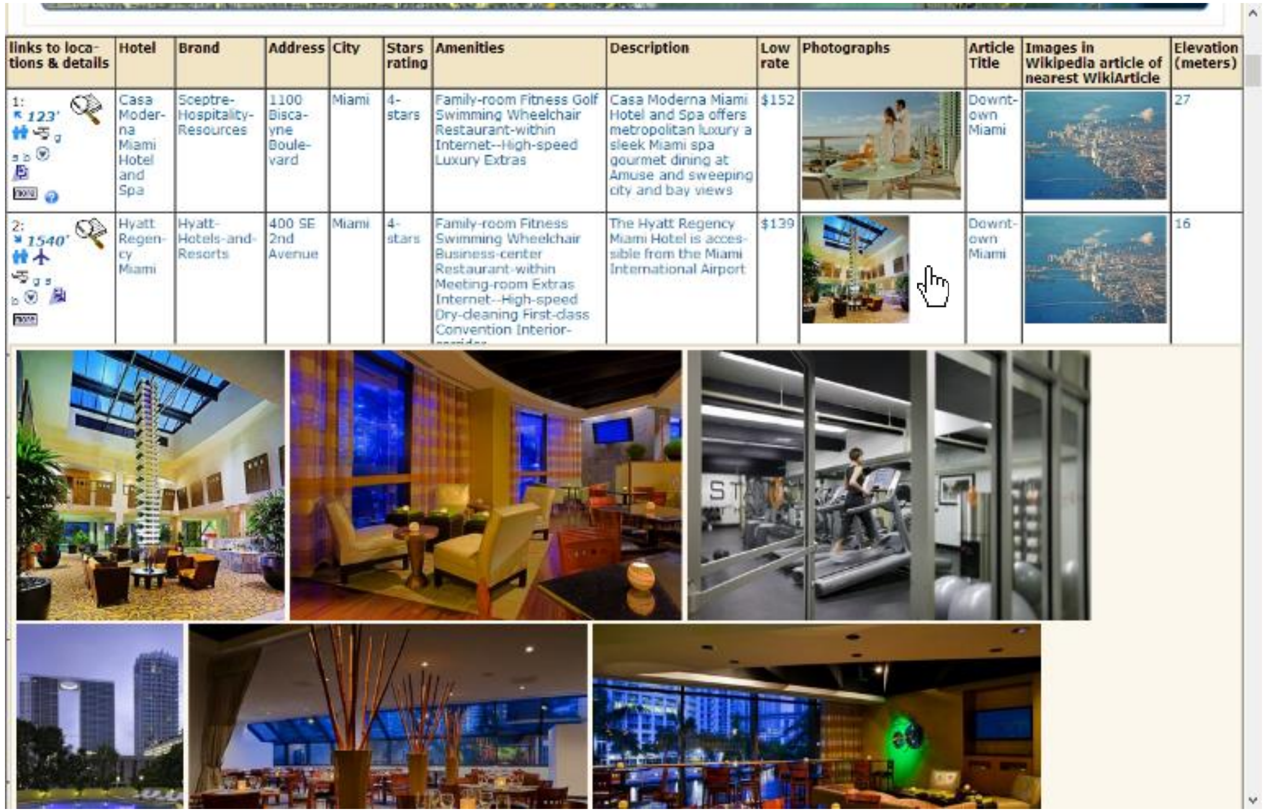


Loading Process



Visualization of sksOpen





Hotel with 4 stars or above and less than $200 per night near downtown Miami

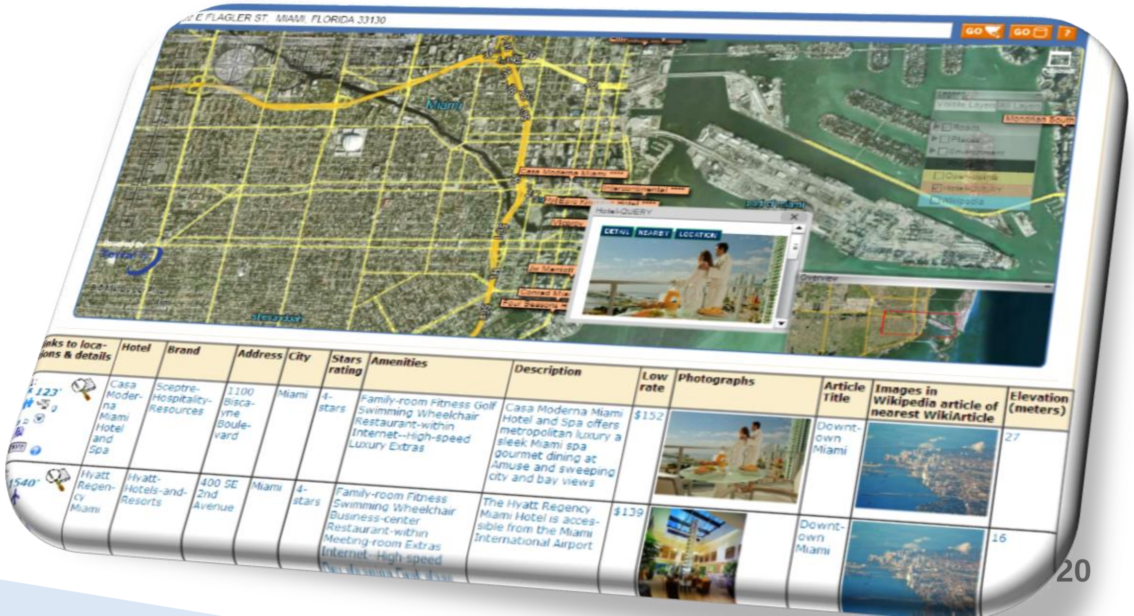
sksOpen performance

- Data file: “us_consumer_2012_full”
  - 68GB   173,483,090 records   136 fields per each record
- KNN query
  - Top 50 records, 38339 characters
  - Query time: **1.211971 seconds**, includes the disk access time for record retrieval.
- KNN query with Boolean restriction CITY=miami&FIRST_NAME=jose
  - Top 50 records, 33308 characters.
  - Query time: **1.707193 seconds**, includes the disk access time for record retrieval.

FIU

sksOpen

- Summary
  - Efficient online indexing, querying, and visualization system for Big Geospatial data.
  - Leveraged MapReduce to Improve a distributed disk-resident hybrid index for efficiently answering k-NN queries with Boolean constraints on textual content.
  - A better interactive user interface.



FIU

Outline

- Motivation & Problem Statement
- Main Contributions
- Related Work
- Contributions (Breakdown)
  1. sksOpen
  2. GeoCloud
  3. v-TerraFly
- Conclusions and Limitations
- Future Work
- References

FIU

GeoCloud: Online Spatial Data Analysis and Visualization

- Extra layer running upon TerraFly map
- Facilitates the end user to visualize and analyze spatial data
- Share the analysis results with URLs
- Supporting many different visualization functions and data analysis models
- MapQL Map creation language
- Published in [Yun132, Yun133, Huibo13]

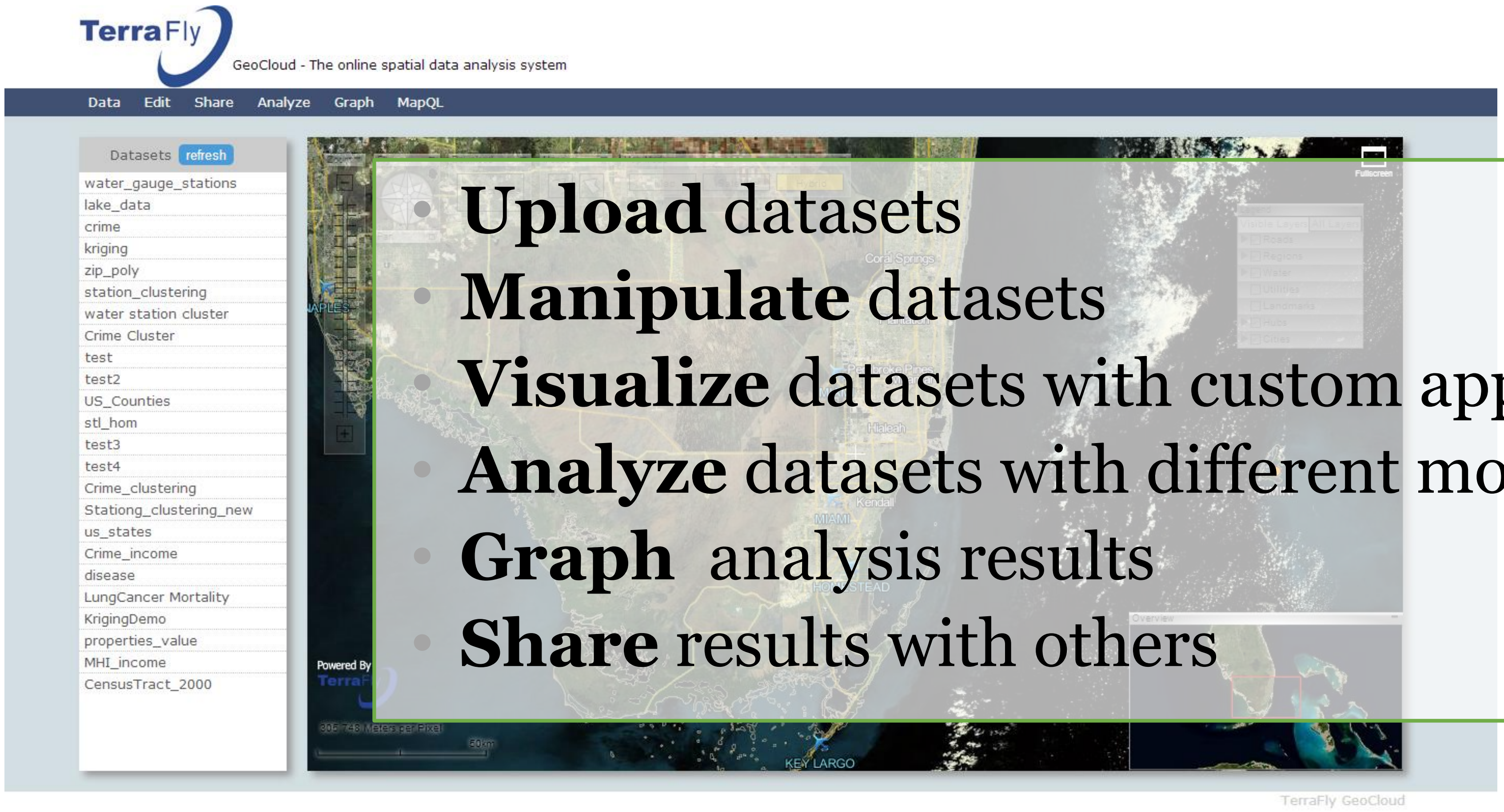
GeoCloud

- Geospatial data analysis is becoming more popular
- Challenges
  - Bad data visualization
  - Complicated and fussy tools to analysis
  - Data analysis is resource consuming
- TerraFly GeoCloud
  - Visualize and manipulate data
  - Online data analysis
  - MapQL feature

GeoCloud

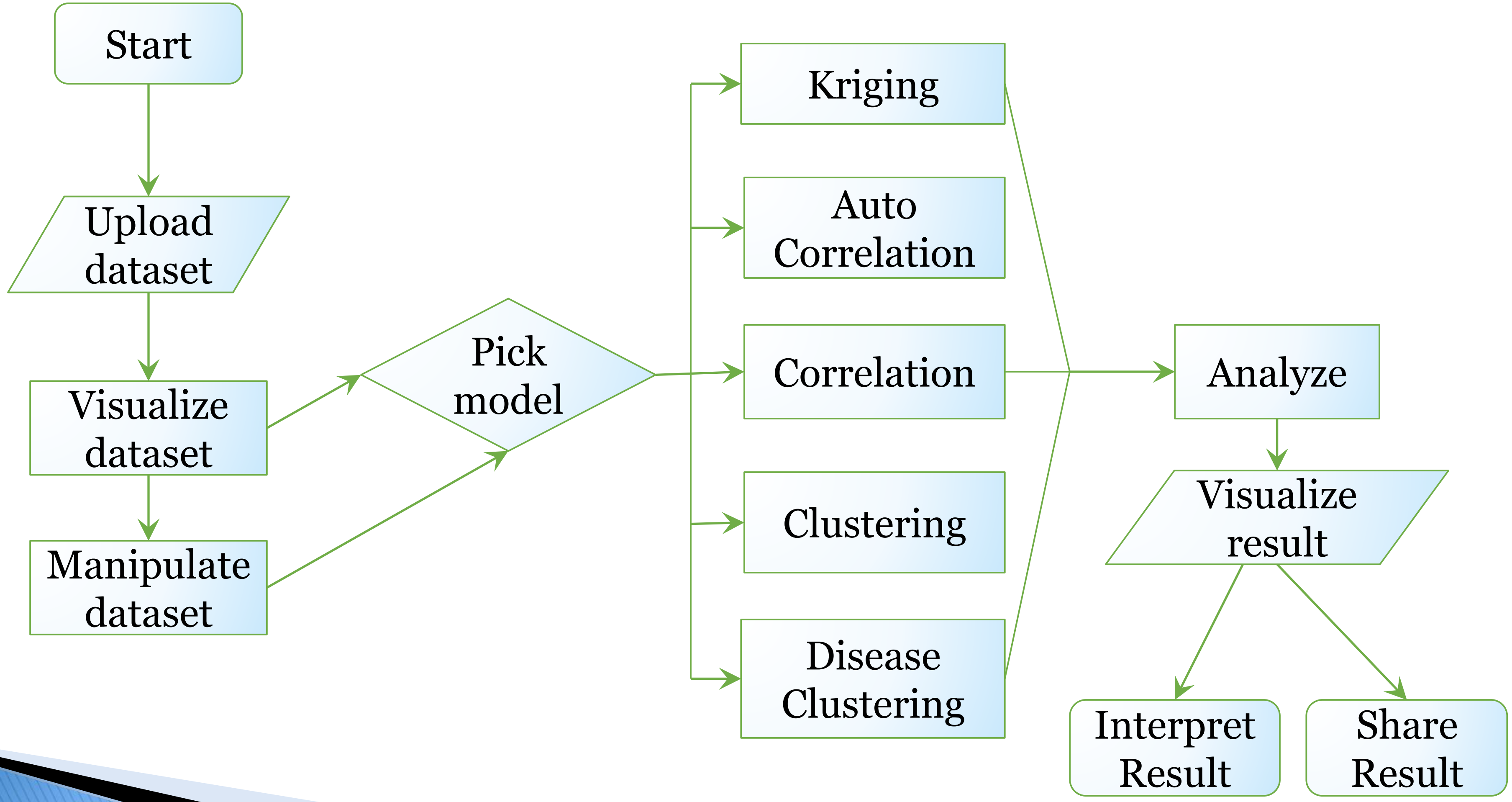
- A prototype spatial data analysis web application
  - Uses TerraFly Maps API
  - JavaScript TerraFly API add-ons
  - JavaScript Web app GUI and charting library
- TerraFly Spatial Analysis – from Module to Cloud:
  - TerraFly to provide online Spatial Analysis Solutions in a high performance cloud Environment.

GeoCloud

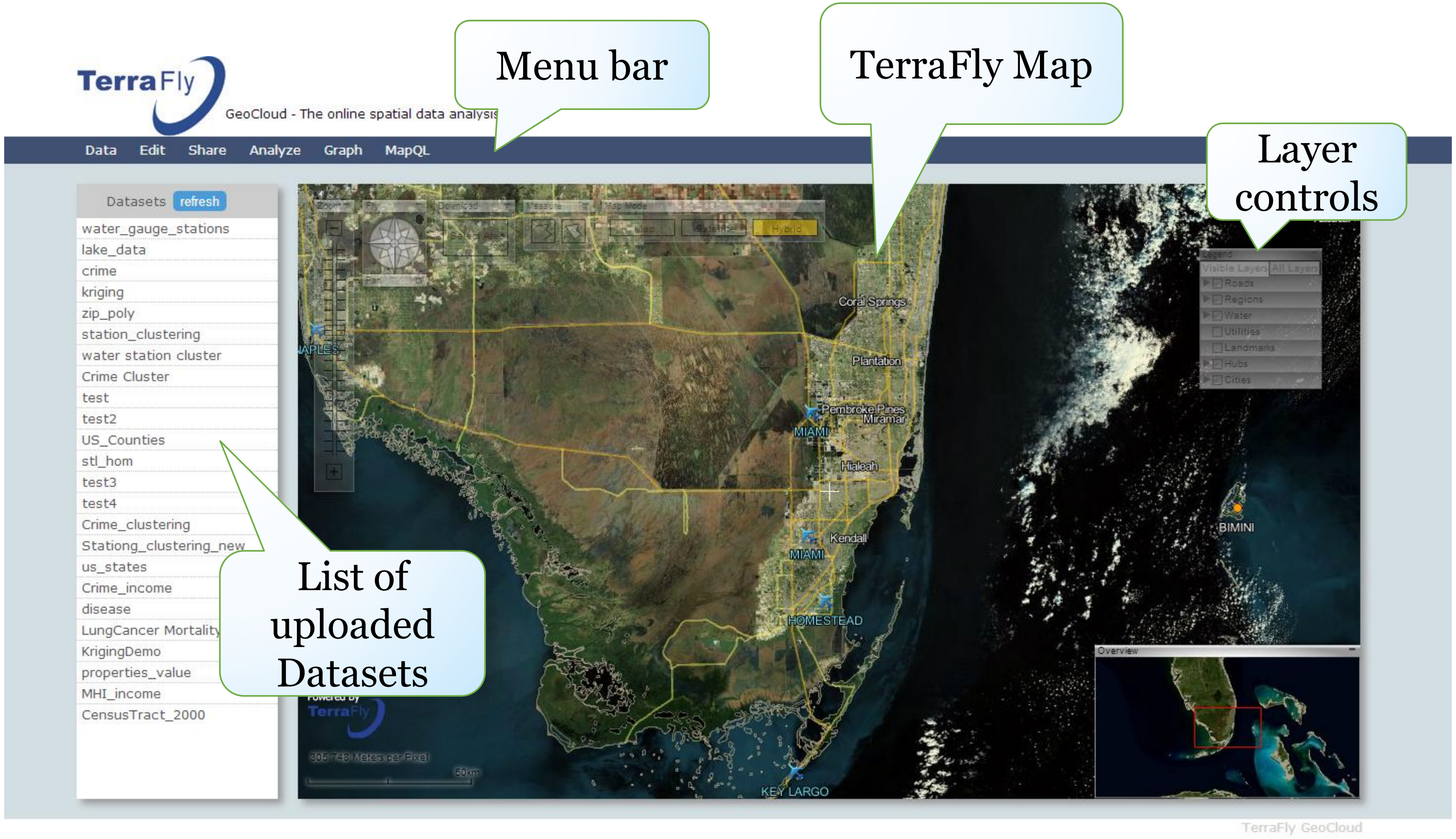


- **Upload** datasets
- **Manipulate** datasets
- **Visualize** datasets with custom appearances
- **Analyze** datasets with different models
- **Graph** analysis results
- **Share** results with others

Workflow

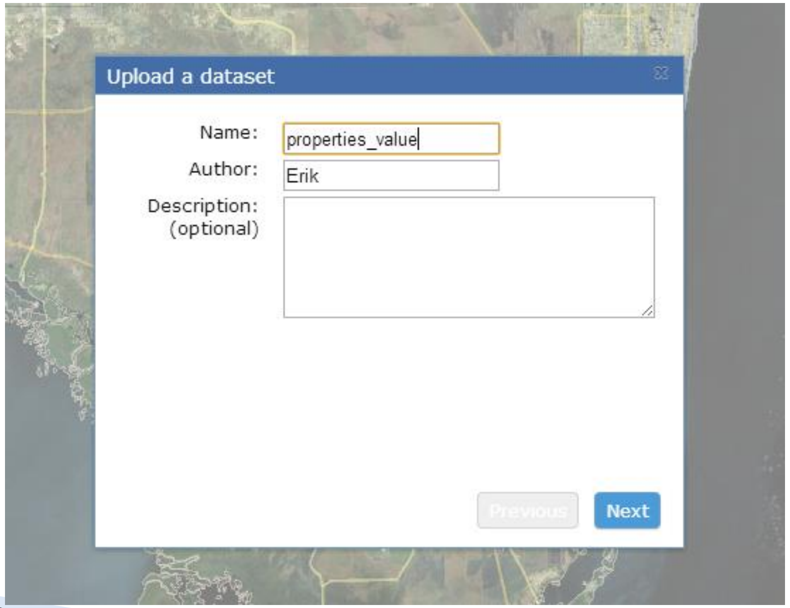

Start
Upload dataset
Visualize dataset
Manipulate dataset
Pick model
Kriging
Auto Correlation
Correlation
Clustering
Disease Clustering
Analyze
Visualize result
Interpret Result
Share Result

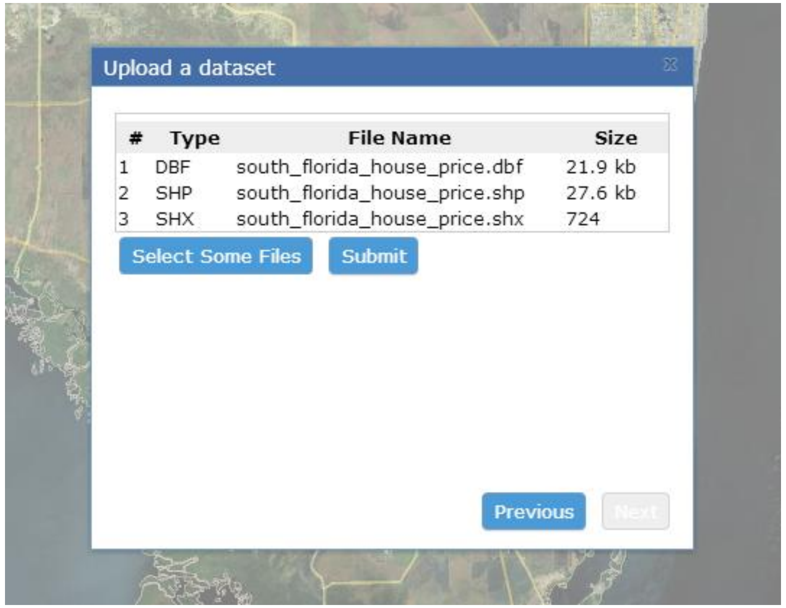
GeoCloud Interface



Uploading a Dataset

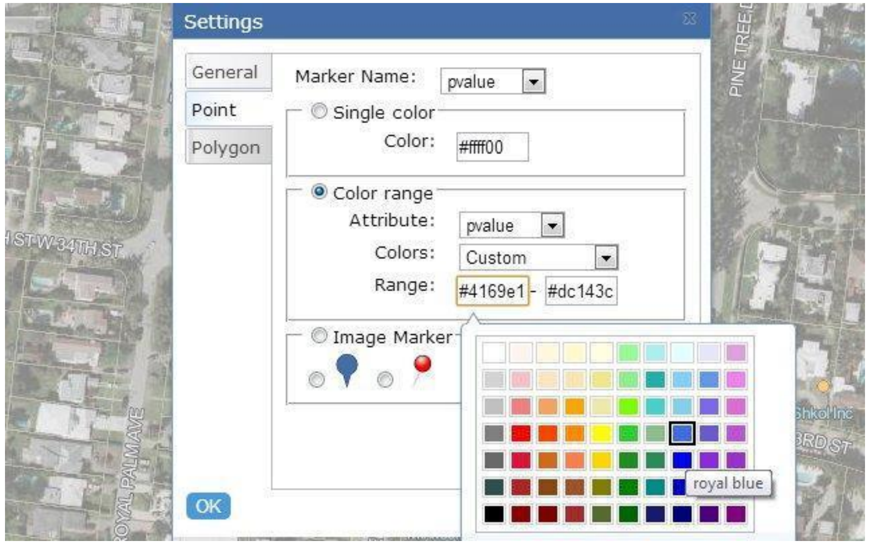
- The user uploads a spatial dataset
  - Supports several file types
  - Sent by HTTP POST to backend

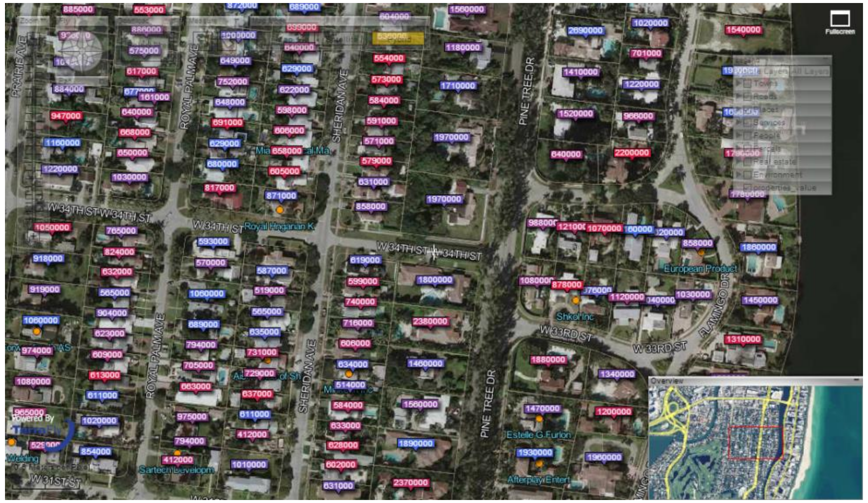




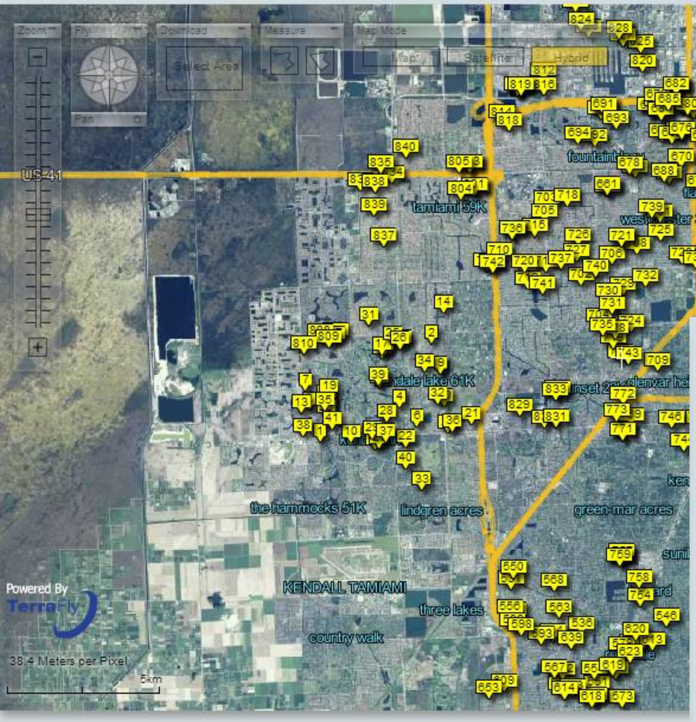
Spatial Data Visualization

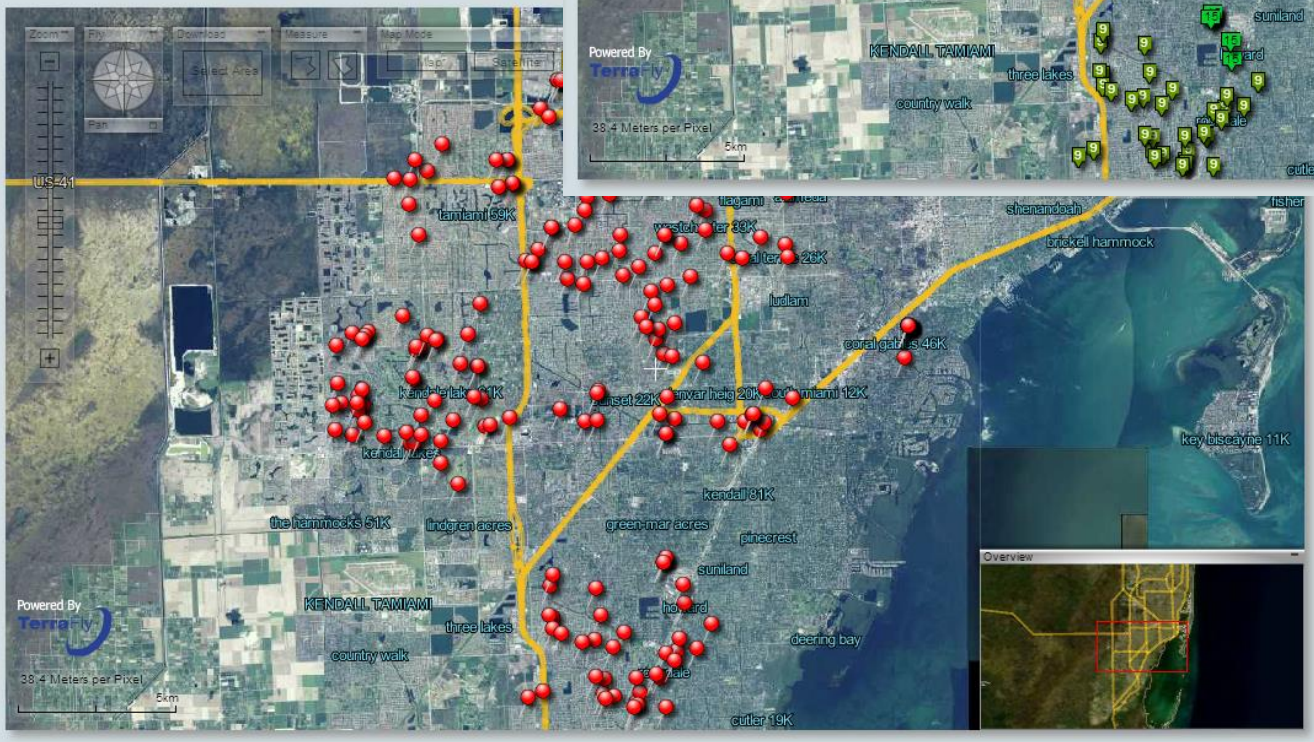
- A user requests data from the backend
  - Adds dataset to map using TerraFly Maps API
  - UI to customize the appearance

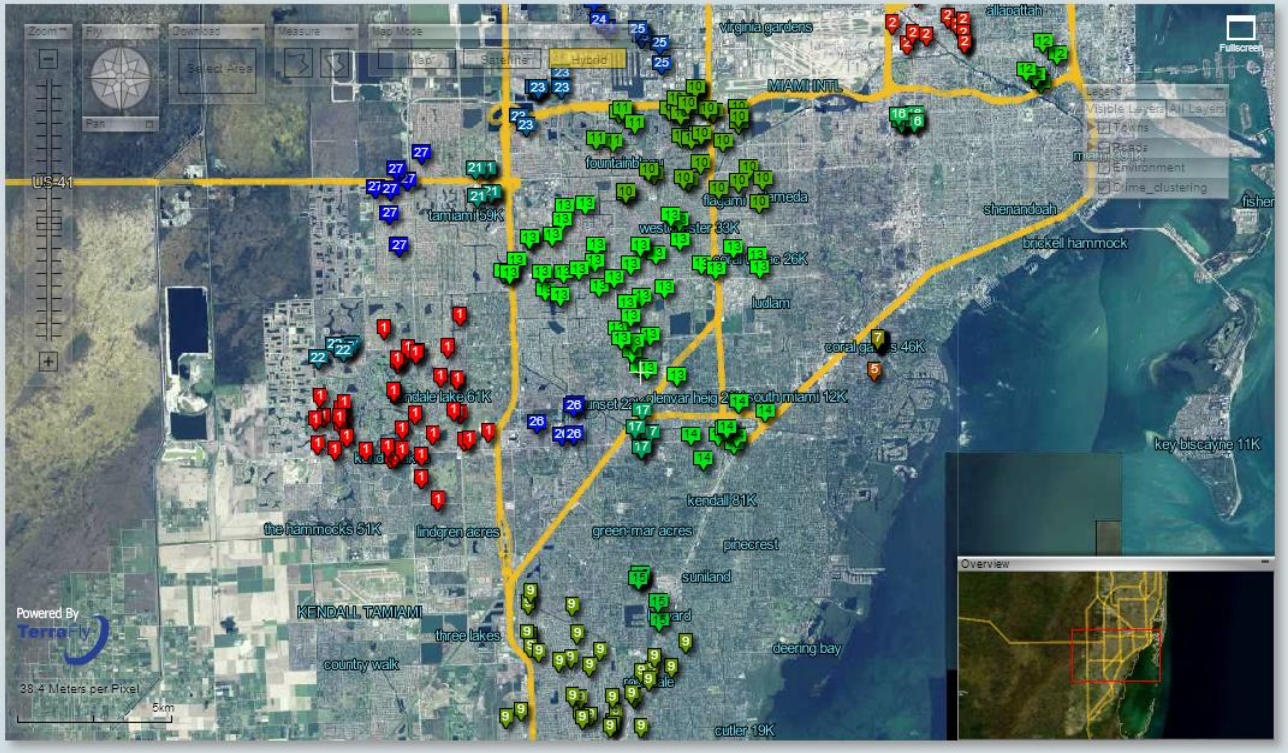




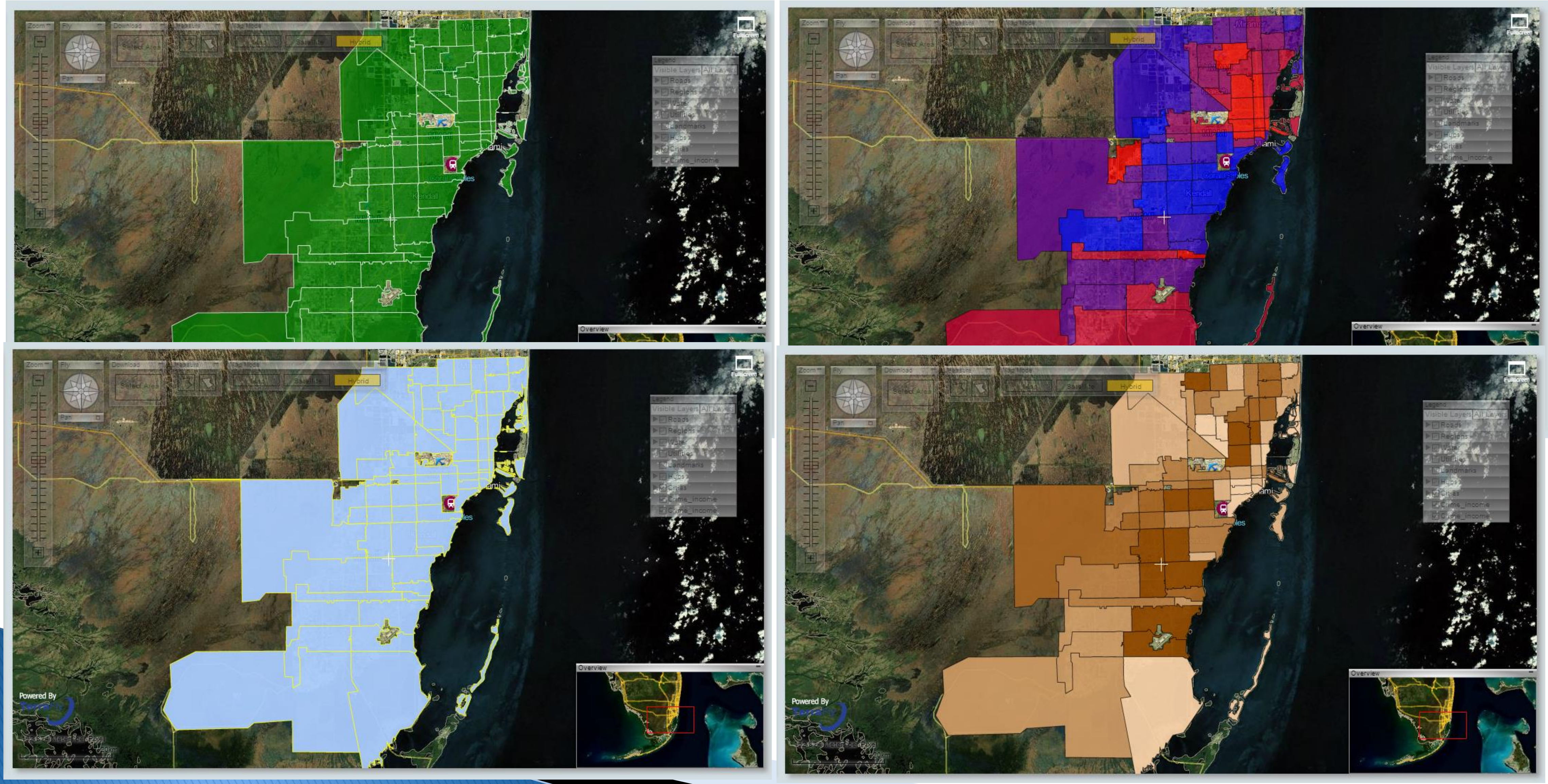
Points


virginia gardens
MIAMI INTL
tamiami 59K
fountainbleau
the hammocks 51K
lindgren acres
green-mar acres
KENDALL TAMIAMI
three lakes
country walk
shenandoah
brickell hammock
Powered By
TerraFly
38.4 Meters per Pixel
5km


tamiami 59K
kendale lake 61K
coral gables 46K
south miami 12K
key biscayne 11K
kendall 81K
the hammocks 51K
lindgren acres
green-mar acres
pinecrest
suniland
KENDALL TAMIAMI
three lakes
country walk
deering bay
cutler 19K
shenandoah
brickell hammock
Overview
Powered By
TerraFly
38.4 Meters per Pixel
5km


virginia gardens
MIAMI INTL
tamiami 59K
ludlam
coral gables 46K
kendall 81K
the hammocks 51K
lindgren acres
green-mar acres
pinecrest
suniland
KENDALL TAMIAMI
three lakes
country walk
deering bay
cutler 19K
shenandoah
brickell hammock
key biscayne 11K
Overview
Powered By
TerraFly
38.4 Meters per Pixel
5km

Polygons



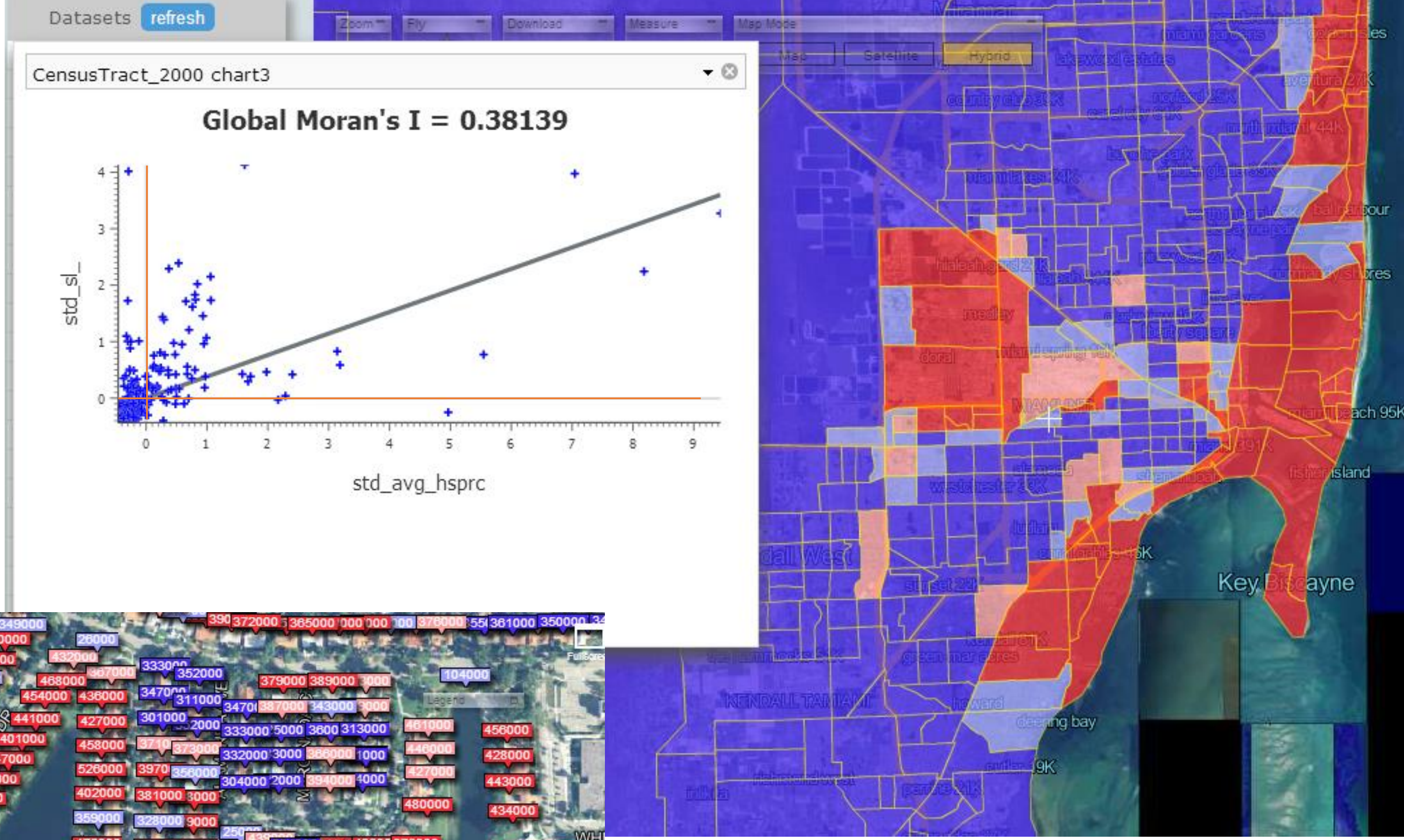
Spatial Data Analysis Models

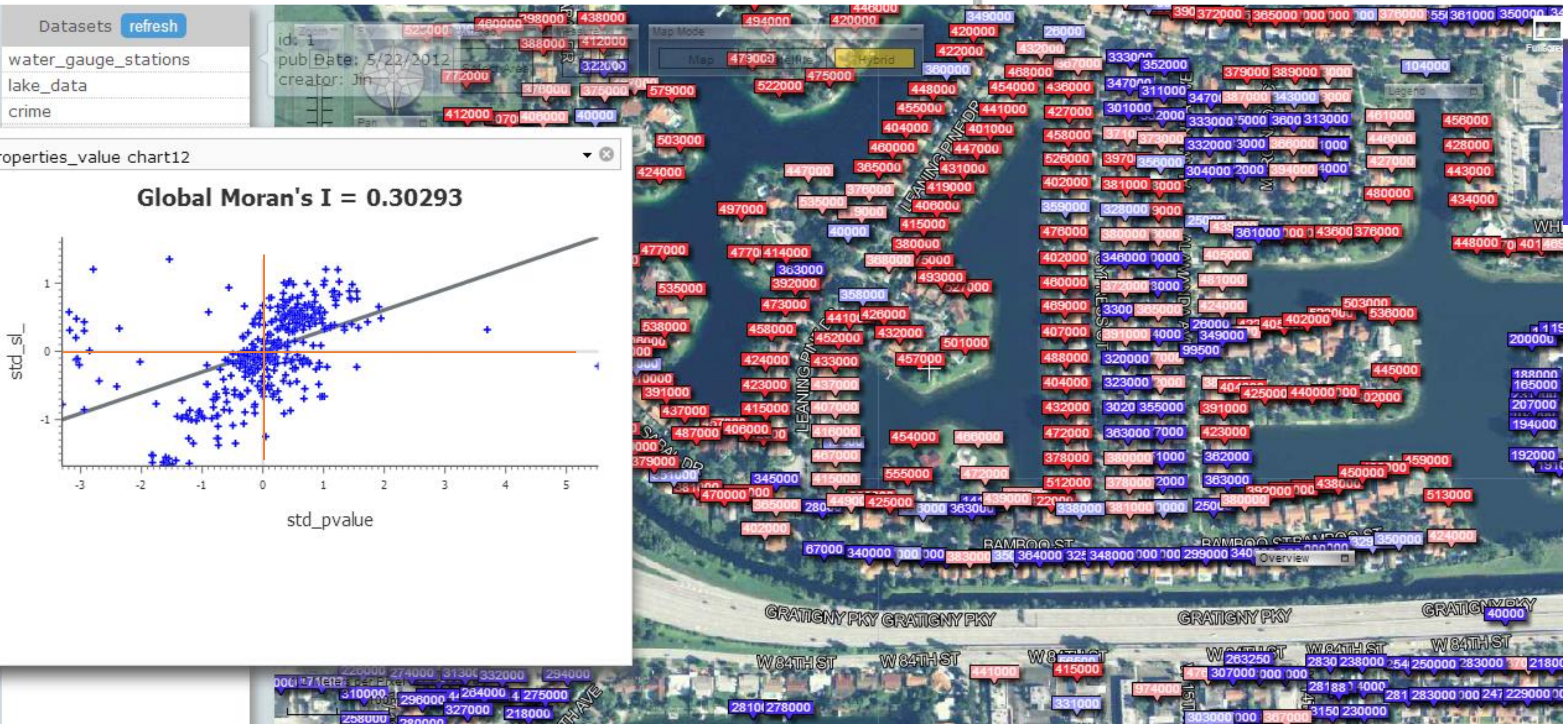
- Spatial Autocorrelation
  - Check for spatial dependency and clusters
- Spatial Correlation
  - Check for Dependency of one variable to another
- Clustering
  - Grouping similar spatial objects
- Kriging
  - Geo statistical estimator for unobserved locations
- Disease Clustering

FIU

Case study

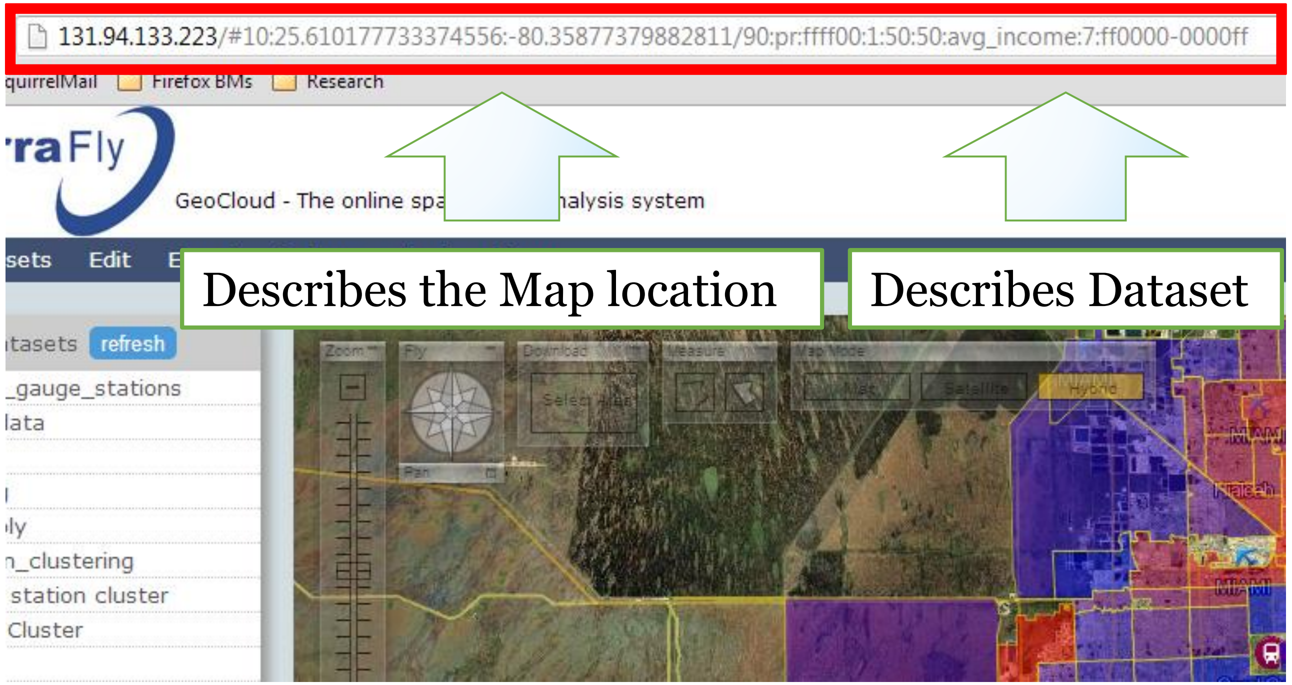
- Spatial Autocorrelation





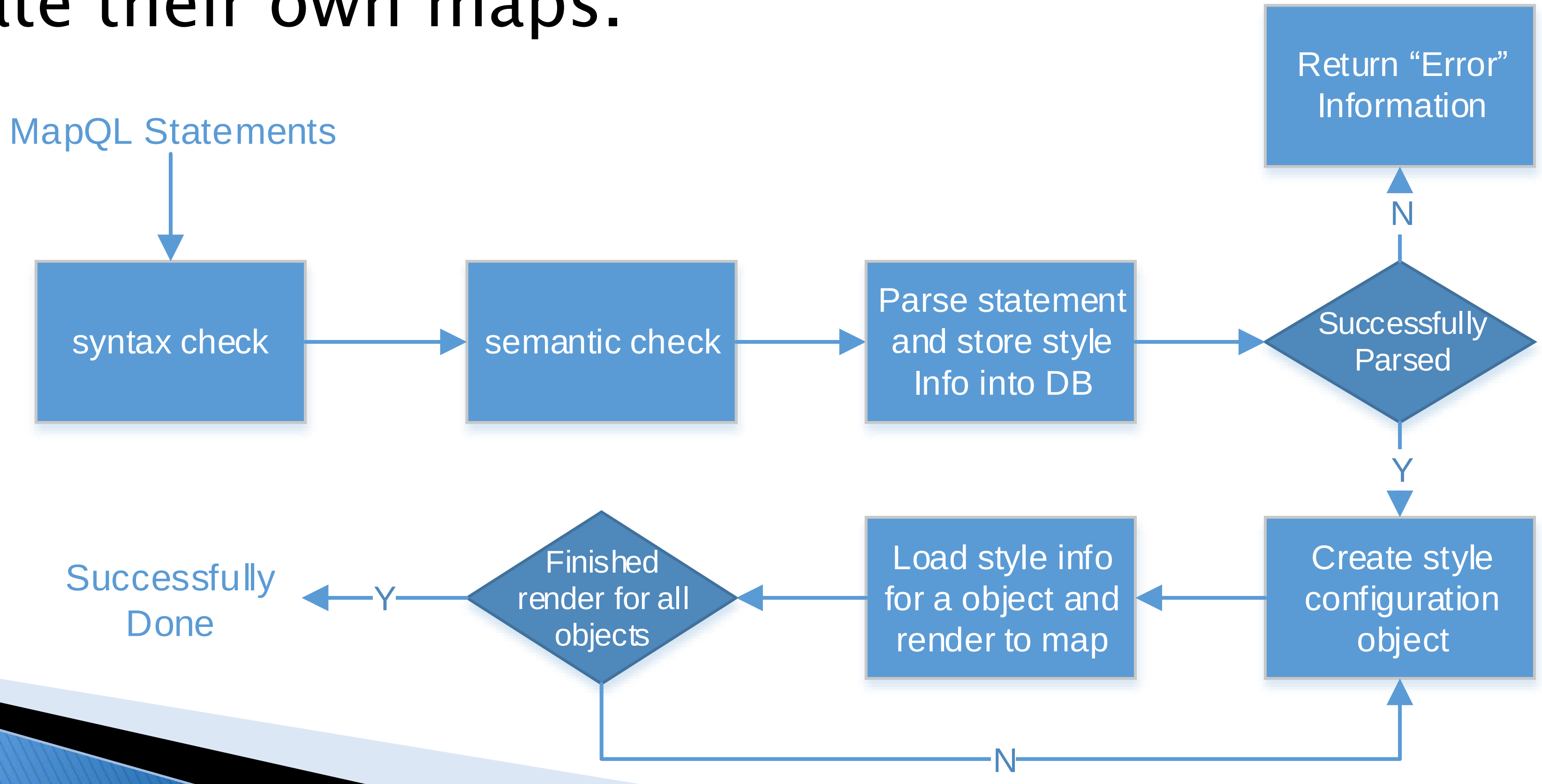
Share

- Share with a URL
  - Share multiple visualized datasets
  - Share analysis results



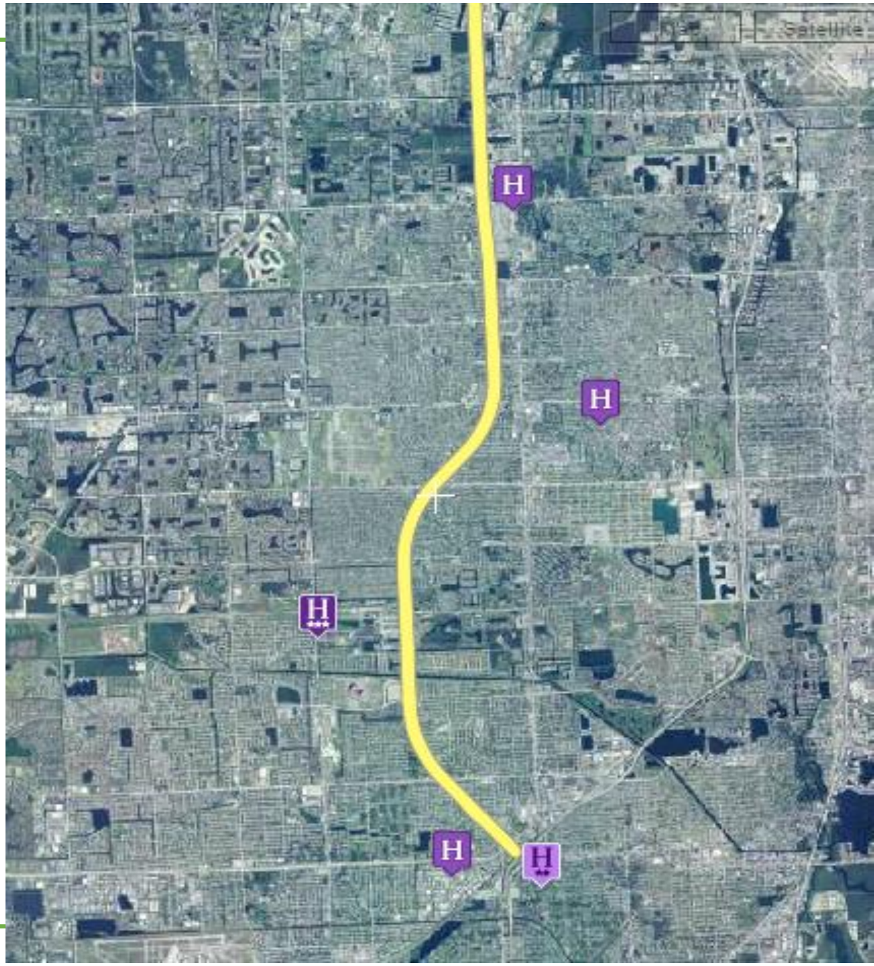
MapQL

- An SQL-like language used to render map layers
- Facilitate developer to use the TerraFly map as their wish
- Easily create their own maps.



MapQL

```
SELECT
    CASE
       WHEN star >= 1 and star < 2 THEN '/var/www/cgi-bin/hotel_1star.png'
       WHEN star >= 2 and star < 3 THEN '/var/www/cgi-bin/hotel_2stars.png'
       WHEN star >= 3 and star < 4 THEN '/var/www/cgi-bin/hotel_3stars.png'
       WHEN star >= 4 and star < 5 THEN '/var/www/cgi-bin/hotel_4stars.png'
       WHEN star >= 5 THEN '/var/www/cgi-bin/hotel_2stars.png'
    ELSE '/var/www/cgi-bin/hotel_0star.png'
    END AS T_ICON_PATH,
   h.Geo AS GEO
FROM
   osm_fl o
   LEFT JOIN
   hotel_all h
   ON
   ST_Distance(o.geo, h.geo) < 0.05
WHERE
   o.name = 'Florida Turnpike';
```



Query the hotels along a certain street within a certain distance

GeoCloud

- Summary
  - Easily analyze and visualize spatial data in browser
  - Satisfy the increasing demand of information sharing
  - Allows users to customize their own spatial data visualization using a SQL-like MapQL language rather than writing codes with Map API

Outline

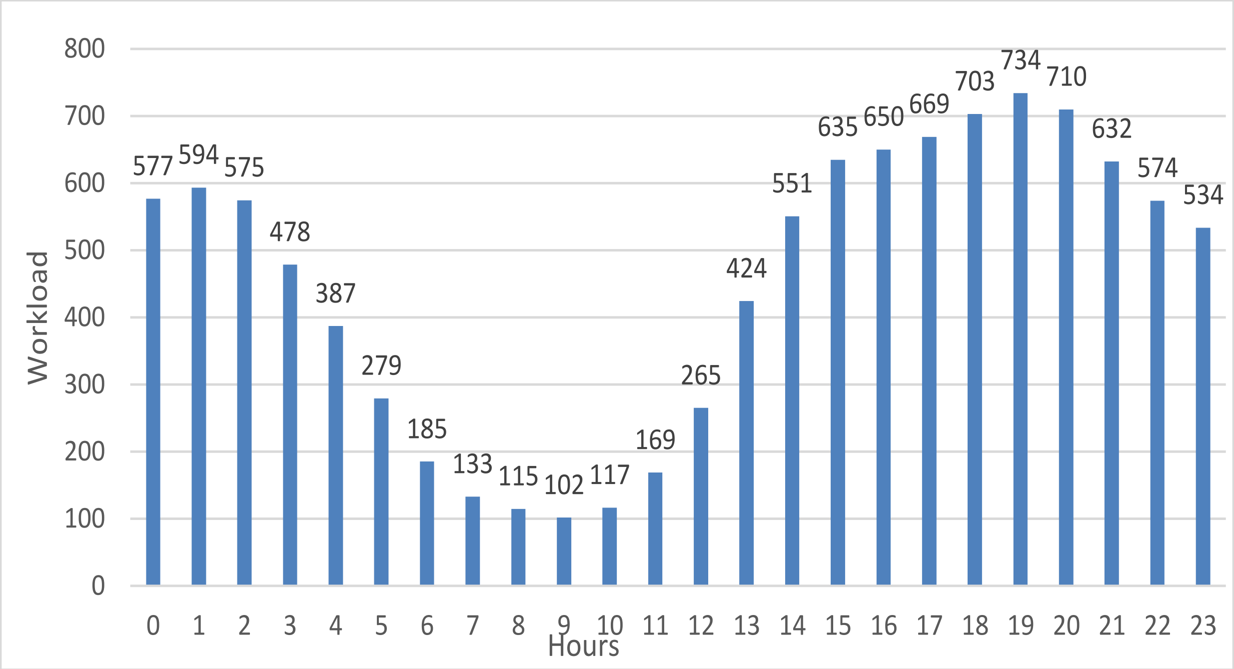
- Motivation & Problem Statement
- Main Contributions
- Related Work
- Contributions (Breakdown)
  1. sksOpen
  2. GeoCloud
  3. v-TerraFly
- Conclusions and Limitations
- Future Work
- References

FIU

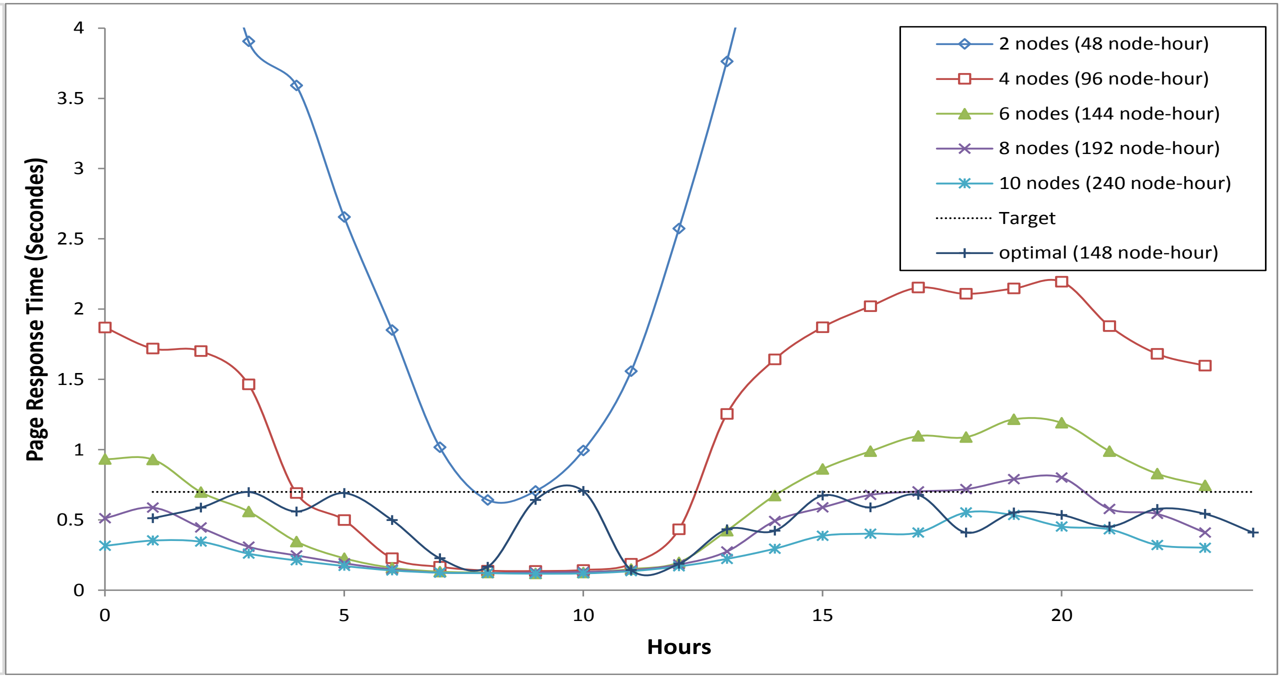
v–TerraFly: Autonomic Resource Management for Virtualized Web Map service

- Predict the demand of map workloads online
- Autonomic resource management
- Optimize resource allocations considering both response time and data freshness as the QoS target
- Involved multiple CPU and I/O intensive tiers
- Published in [Yun134]
- Undergoing journal review [Yun135] [Lixi13]

Motivation



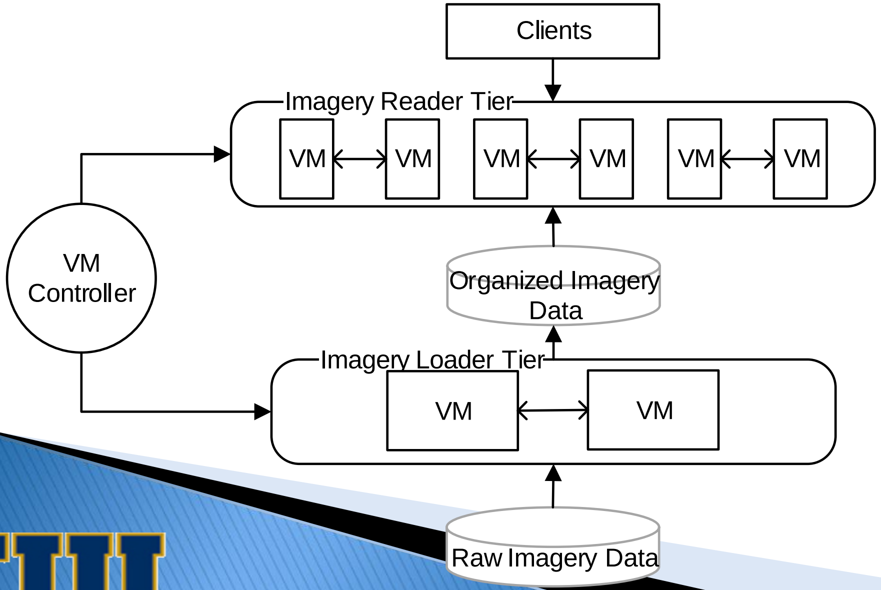
TerraFly reader tier workload pattern



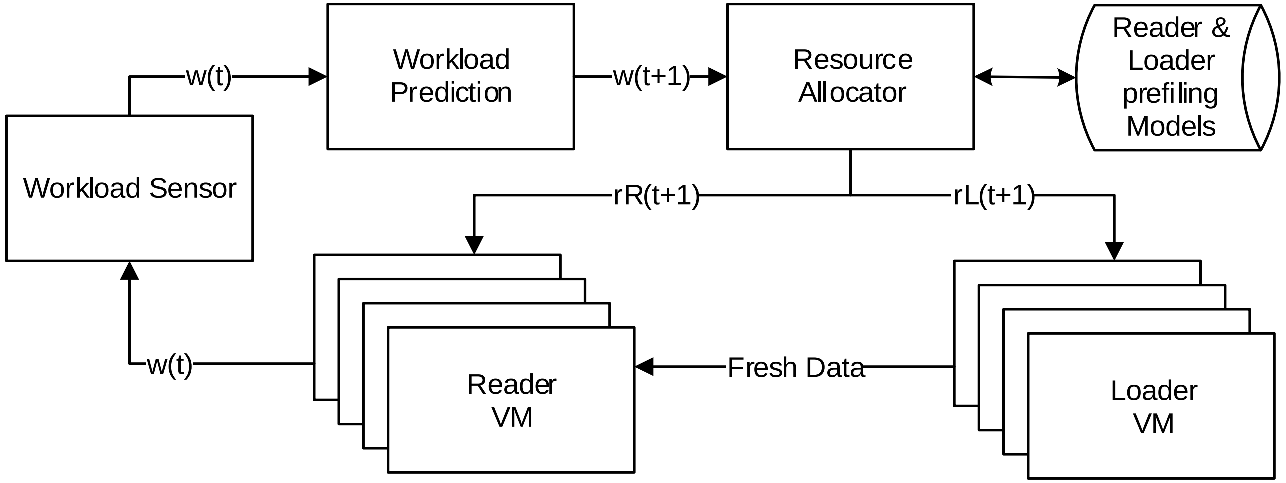
Performance and Resource cost comparison using different deployment schemes

Autonomic resource management

- Multi-Tiers Web map service
- Easy management and maintenance by virtual machine
- Better utilization of computing resource
- Dynamically distributing system



v-TerraFly system



Autonomic resource management system for v-TerraFly

Workload Prediction

- Based on the double exponential smoothing (DES) method
  - Suitable for discrete data sequence with repeated changing patterns

$$Y^{Des}(t+1) = 2S'(t) - S''(t) + \left(\frac{\alpha}{1-\alpha}\right)\left(S'(t) - S''(t)\right)$$

$$S'(t) = \alpha Y(t) + (1-\alpha)S'(t-1) \qquad S''(t) = \alpha S'(t) + (1-\alpha)S''(t-1)$$

- New two-level time series prediction approach

$$\text{Eq. 1: } w'(t+1) = \mu_h w_h^{Des}(t+1) + \mu_d w_d^{Des}(t+1)$$

$$\text{Eq. 2: } w_h^{Des}(t) = 2S'(t-1) - S''(t-1) + \left(\frac{\alpha_h}{1-\alpha_h}\right)\left(S'(t-1) - S''(t-1)\right)$$

$$\text{Eq. 3: } w_d^{Des}(t) = 2S'(t-24) - S''(t-24) + \left(\frac{\alpha_d}{1-\alpha_d}\right)\left(S'(t-24) - S''(t-24)\right)$$

- w_h^{Des} is the *horizontal* double exponential smoothing prediction based on the hourly pattern in the workload
- w_d^{Des} is the *vertical* double exponential smoothing prediction based on the daily pattern of the workload

FIU

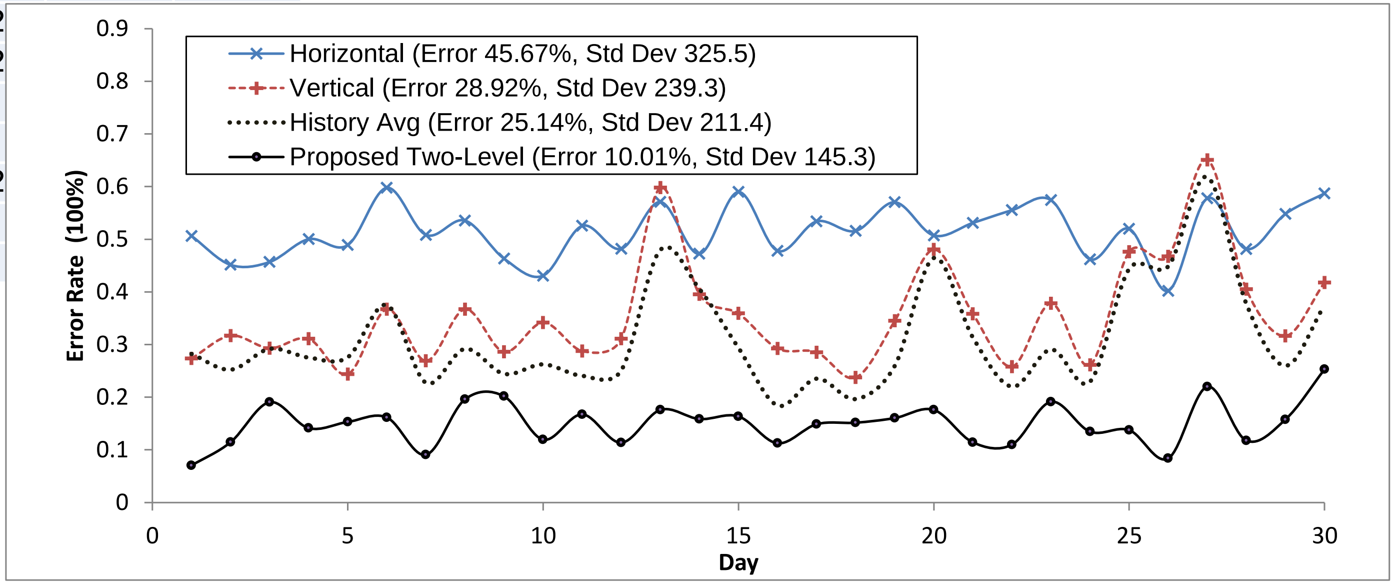
Workload Prediction

- Two-level prediction method delivers significantly better accuracy in predicting the request rate of one month workload

horizontal

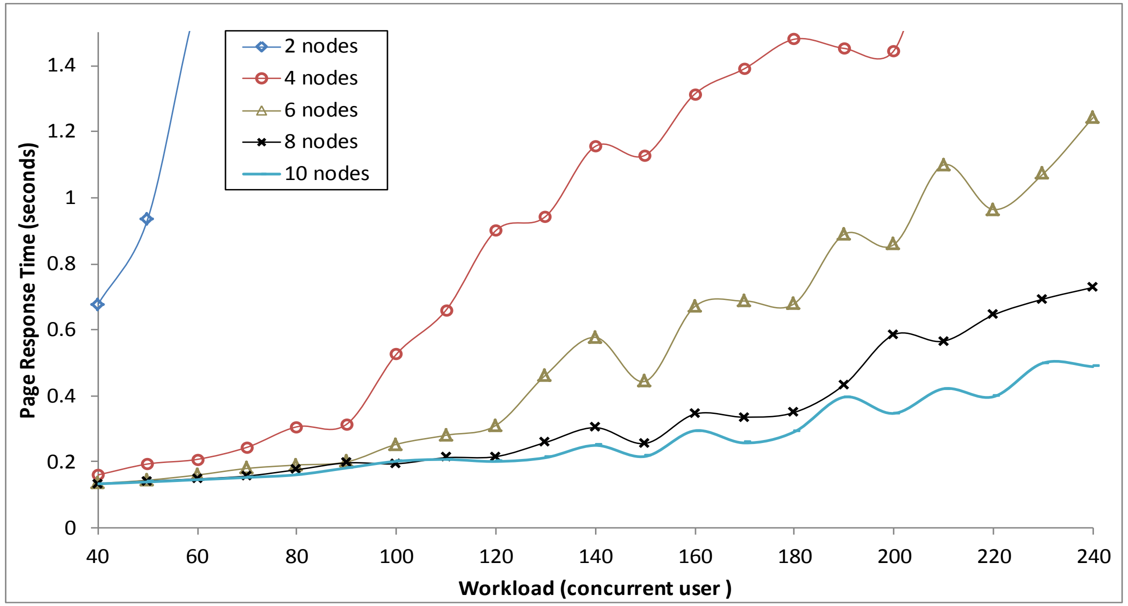
	hour1	hour2	hour3	hour4	hour5	hour6	hour7	hour8	hour9
day1	735	667	973	207	324	345	256	112	…
day2	677	951	689	491	413	246	224	272	…
day3	913	789	707	625	154	248	2		
day4	902	909	944	544	509	430	2		
day5	794	822	777	409	634	344	1		
day6	1016	818	368	483	420	421			
day7	566	657	551	434	305	223	2		
day8	750	677	975	676	293	419	1		
day9	…	…	…	…	…	…	…		

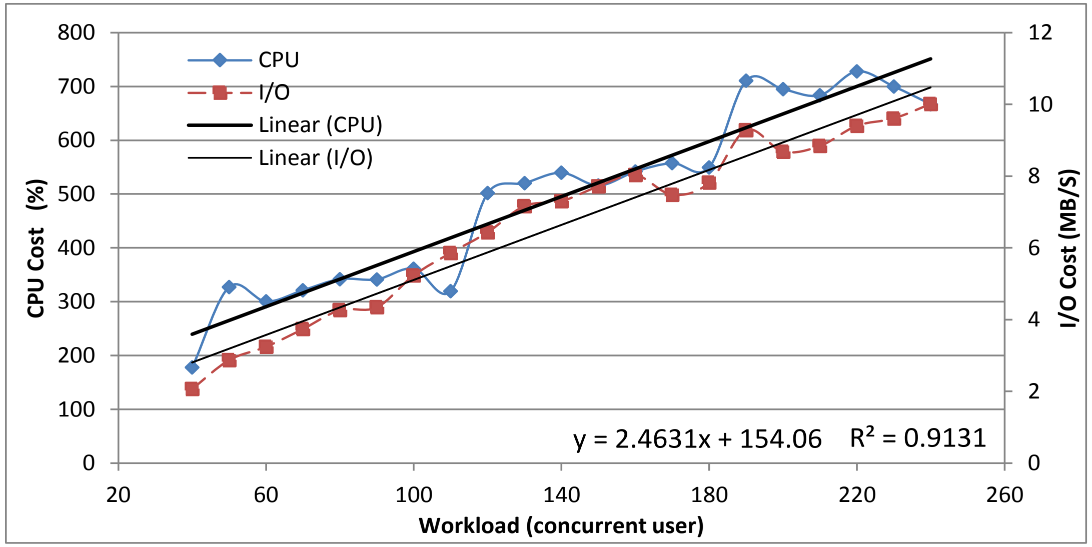
vertical

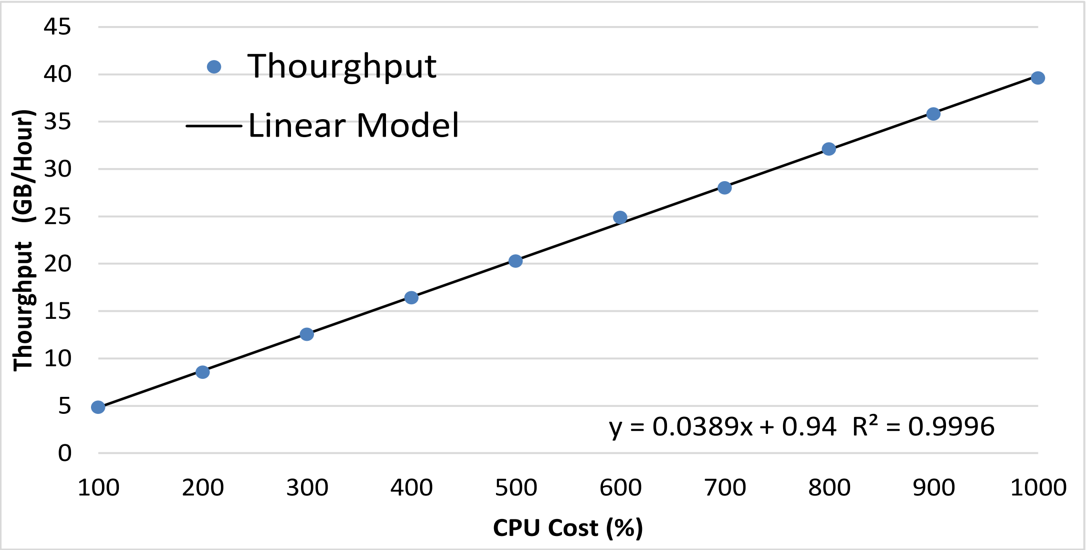


different workload prediction approaches

Reader and loader tiers profiling







QoS Model

- QoS model consider both the responsiveness in serving user mapping requests (reader tier) and the quality of returning geographic information (loader tier)
- The former guarantees acceptable response time and the latter keeps the imagery data up to date
- QoS model is defined to represent the overall system performance

Eq. 5: $QoS(t) = r(t) \times f(t)$

Eq. 6: $r(t) = {}^{RT_{ref}}/_{RT(t)}$

Eq. 7: $f(t) = (1 - \rho) \times f(t-1) + \Delta D(t)/D_{ref}$

r(t) is called the normalized response time

f(*t*) is called the cumulative data freshness

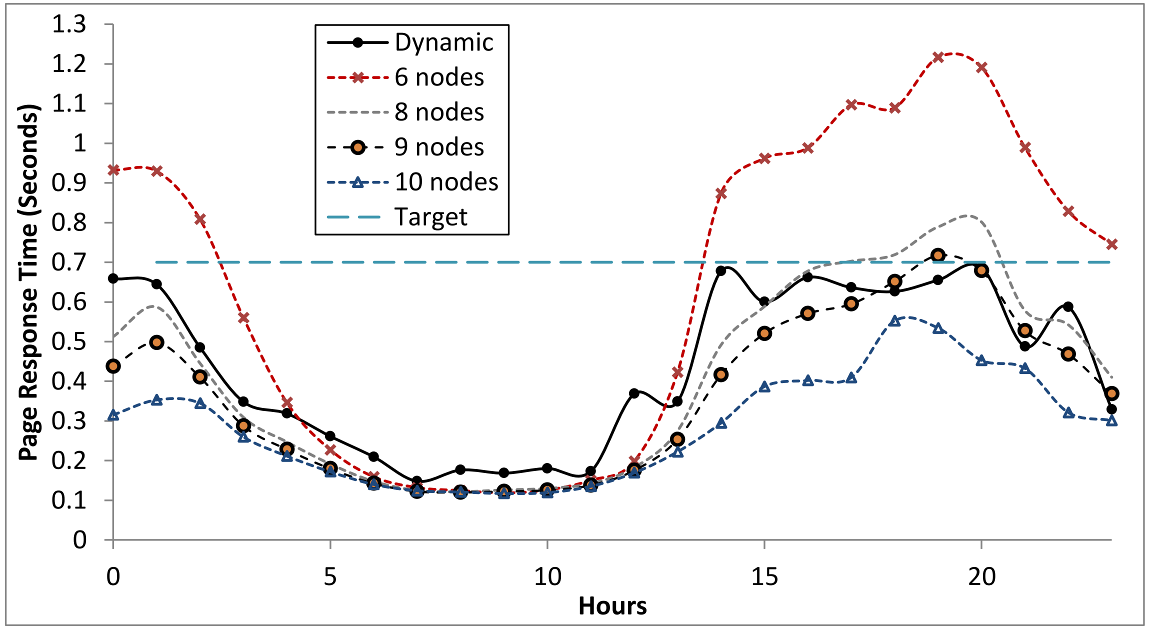
ρ is the decaying factor

FIU

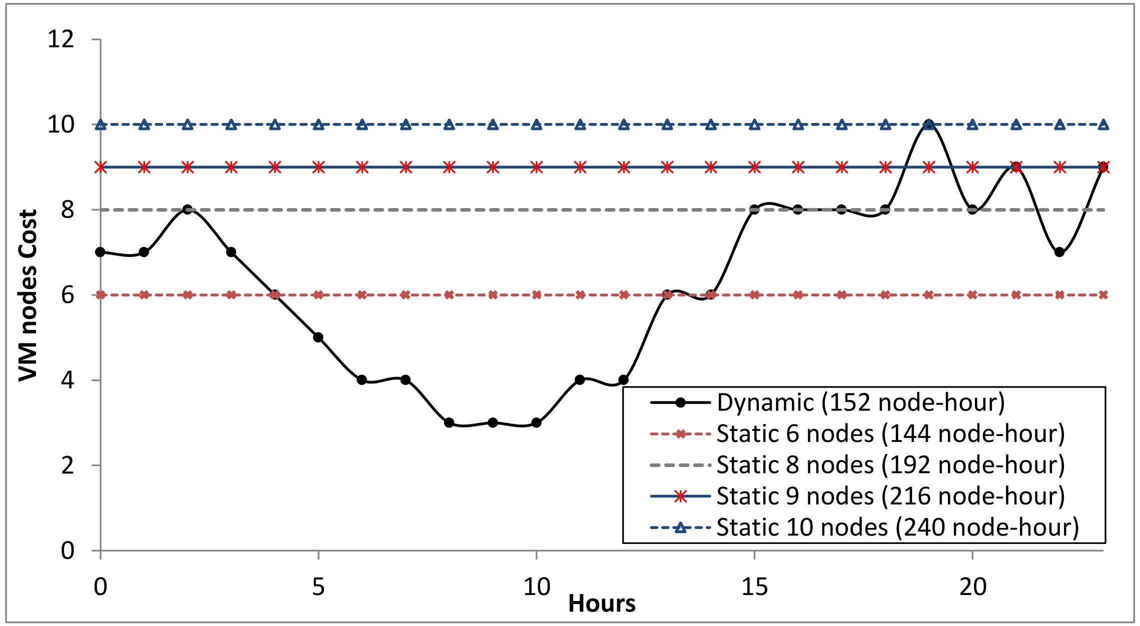
Evaluation

- v-TerraFly prototype
- Real traces collected from the TerraFly production system
- Two Dell PowerEdge 2970 servers
  - Two six-core 2.4GHz AMD Opteron CPUs
  - 32GB of RAM
  - 1TB 7.2 RPM SAS disk
- Windows Server 2008 and Hyper-V
- Each Reader and Loader VM
  - one core CPU
  - 2G memory

Resource Management of Reader Tier



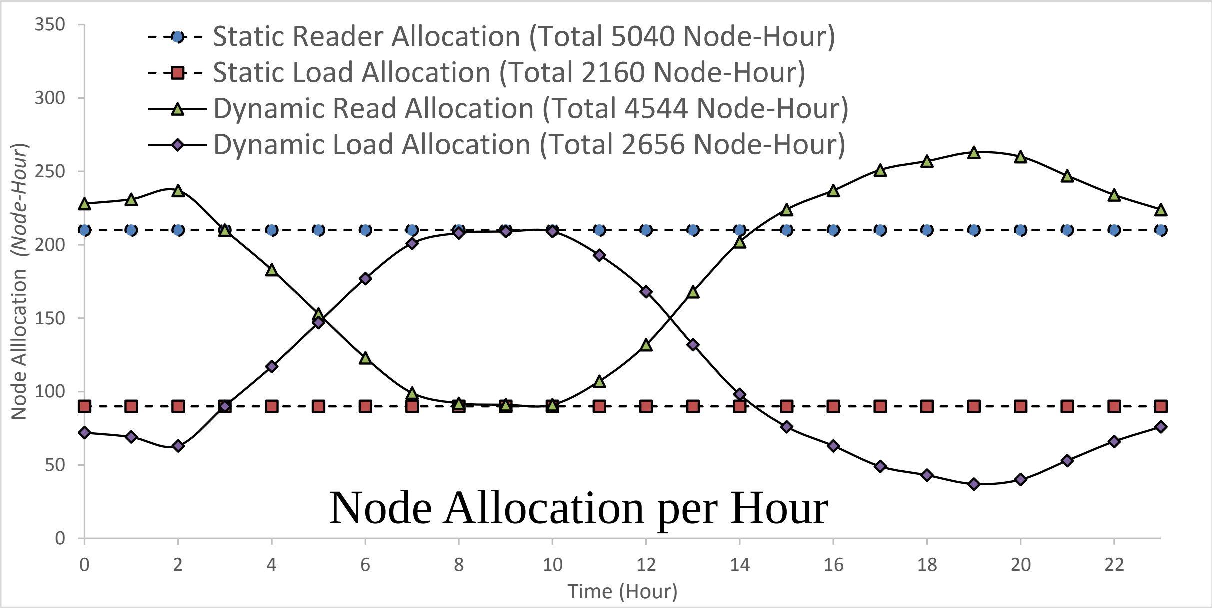
Result: Response time

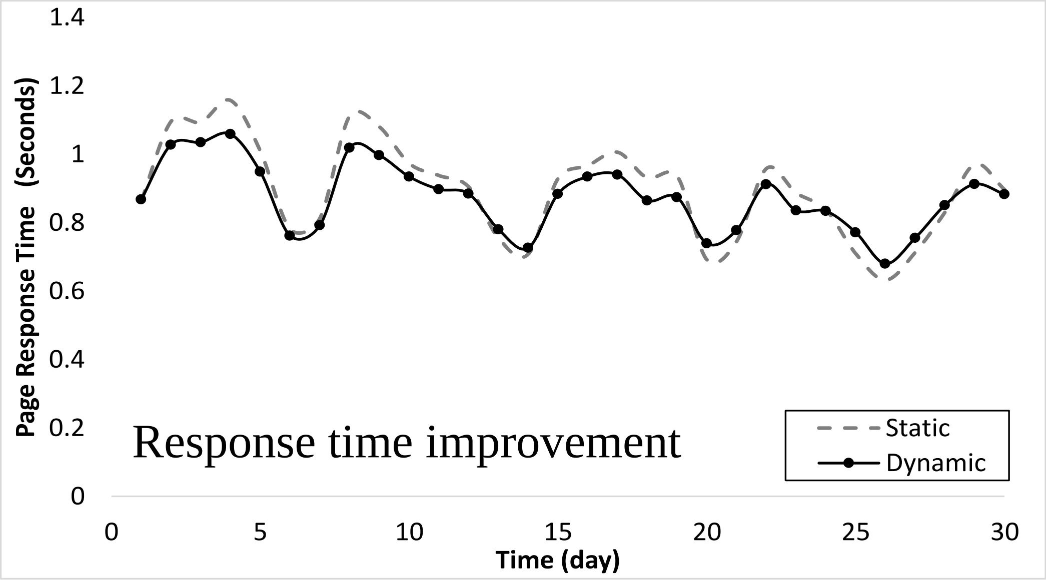


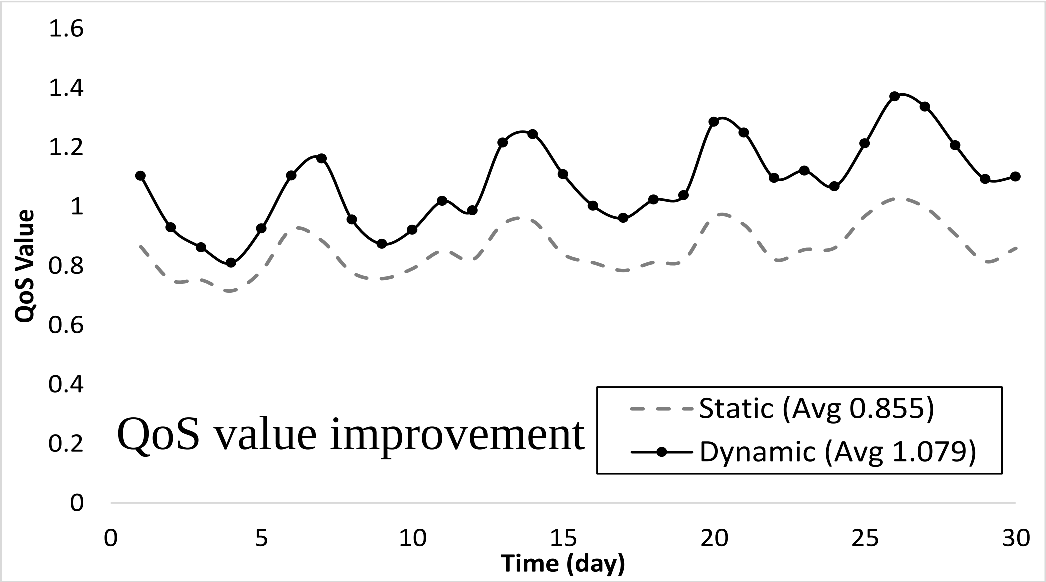
VM nodes cost by hours and total VM nodes Cost

10 nodes plan cost 36.67% more total resources

Resource Management of both Reader and Loader Tiers



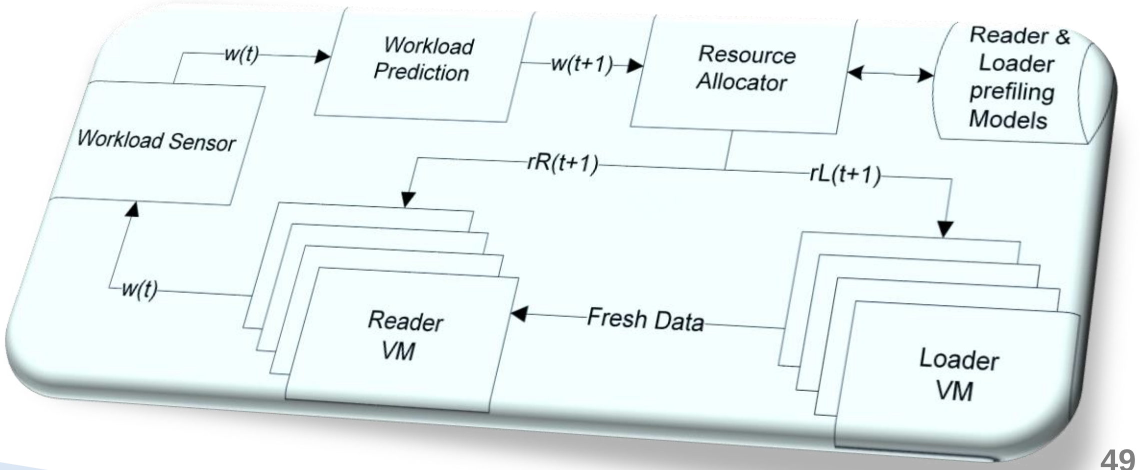




Shows that the proposed dynamic plan achieves much better overall QoS (26.19% improvement)

v-TerraFly

- Summary:
  - Created by virtualizing the multi-tiers of a typical map service system
  - Allowing resources to be dynamically allocated across the tiers
  - Predicting the workload intensity based on historical data
  - Estimating the resource needs of the map service's Reader and Loader Tiers based on their performance models
  - Unique QoS metric is then defined to capture the tradeoff



Outline

- Motivation & Problem Statement
- Main Contributions
- Related Work
- Contributions (Breakdown)
  1. sksOpen
  2. GeoCloud
  3. v-TerraFly
- Conclusions and Limitations
- Future Work
- References

Conclusions & Limitations

- Conclusions
  - sksOpen improves spatial query experience
  - GeoCloud do spatial analysis online and share results with URLs
  - v-TerraFly efficiently manage computing resources for web map services

  - sksOpen provide data input to GeoCloud
  - v-TerraFly provide backend performance support to sksOpen and GeoCloud

Conclusions & Limitations

- Limitations
  - sksOpen
    - Still need to improve the code to be open source
    - Large disk redundancy
  - GeoCloud
    - Need domain expert experience to do analysis
    - Need a MapQL statement generator to open to public use.
  - v-TerraFly
    - No large scale implementation
    - Different input pattern need to be verified

Outline

FIU

Future Work

- sksOpen : Improve the code structure and limit disk cost
- GeoCloud: Better UI for public use
- v-TerraFly: Explore how to apply the principle of v-TerraFly to other applications

FIU

Publications

- CONFERENCES
  - [Yun131] **Yun Lu**, Mingjin Zhang, Shonda Witherspoon, Yelena Yesha, Yaacov Yesha, Naphtali Rishe. (2013). sksOpen: Efficient Indexing, Querying, and Visualization of Geo-spatial Big Data. In International Conference on Machine Learning and Applications BigData Workshop
  - [Yun132] **Yun Lu**, Mingjin Zhang, Tao Li, Yudong Guang and Naphtali Rishe. (2013). *Online Spatial Data Analysis and Visualization System.* In ACM SIGKDD Conference on Knowledge Discovery and Data Mining Workshop on Interactive Data Exploration and Analytics
  - [Yun133] **Yun Lu**, Mingjin Zhang, Tao Li, Chang Liu, Erik Edrosa, Naphtali Rishe. (2013). *TerraFly GeoCloud: Online Spatial Data Analysis System.* In ACM International Conference on Information and Knowledge Management
  - [Yun134] **Yun Lu**, Ming Zhao, Guangqiang Zhao, Lixi Wang, Naphtali Rishe. (2013). *Massive GIS Database System with Autonomic Resource Management.* In International Conference on Machine Learning and Applications BigData Workshop

Publications

- [Huibo13] Huibo Wang, **Yun Lu**, Yudong Guang, Erik Edrosa, Mingjin Zhang, Raul Camarca, Yelena Yesha, Tajana Lucic, Naphtali Rishe. (2013) *Epidemiological Data Analysis in TerraFly Geo-Spatial Cloud*. In International Conference on Machine Learning and Applications BigData Workshop

- JOURNALS
  - [Yun135] **Yun Lu**, Ming Zhao, Lixi Wang, Naphtali Rishe. *v-TerraFly: Large Scale Distributed Spatial Data Visualization with Autonomic Resource Management.* In Journal Of Big Data, SUBMITTED.
  - [Lixi13] Lixi Wang, **Yun Lu**, Jing Xu, Ming Zhao. *Cross-layer Optimization for Virtual Machine Resource Management.* In IEEE Transactions on Parallel and Distributed Systems. SUBMITTED

FIU

References

- [Zickuhr12] K. Zickuhr. Three-quarters of smartphone owners use location-based services. Pew Internet & American Life Project, http://pewinternet.org/Reports/2012/Location-based-services.aspx,2012
- [Jones04]C.B. Jones, A.I. Abdelmoty, D. Finch, G. Fu, and S. Vaid. The Spirit Spatial Search Engine.: Architecture, Ontologies and Spatial Indexing. In Proc. of GIScience, pages 125-139, October 2004
- [Zhou05] Y. Zhou, X. Xie, C. Wang, Y. Gong, and W. Ma. Hybrid Index Structures for Location-Based Web Search. In Proc. Of CIKM, pages 155-162, November 2005
- [Hariharan07] Hariharan, R., Hore, B., Li, C., Mehrotra, S.: Processing spatial-keyword (SK) queries in geographic information retrieval (GIR) systems. In SSDBM, p. 16, 2007.

FIU

References

- [Johnston01] Johnston, K., Ver Hoef, J. M., Krivoruchko, K., & Lucas, N. (2001). Using ArcGIS geostatistical analyst (Vol. 380). Redlands: Esri.
- [O'Sullivan03] O'Sullivan, D., & Unwin, D. J. (2003). *Geographic information analysis*. John Wiley & Sons.
- [Anselin06] Anselin, L., Syabri, I., & Kho, Y. (2006). GeoDa: An introduction to spatial data analysis. Geographical analysis, 38(1), 5-22.
- [Huebscher08] Huebscher, M. C., & McCann, J. A. (2008). A survey of autonomic computing—degrees, models, and applications. ACM Computing Surveys (CSUR), 40(3), 7.

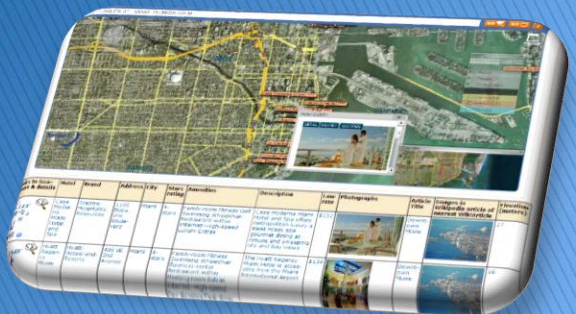
References

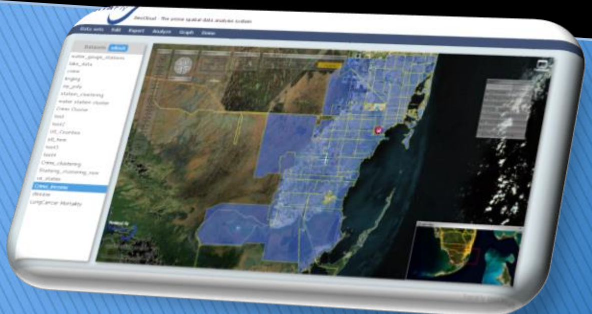
- [Rao09] J. Rao, X. Bu, C. Xu, L. Wang and G. Yin, "VCONF: A Reinforcement Learning Approach to Virtual Machines Auto-configuration", ICAC, 2009.
- [Cary10] Cary, A., Wolfson, O., & Rishe, N. Efficient and scalable method for processing top-k spatial Boolean queries. In Scientific and Statistical Database Management (87-95). Springer Berlin Heidelberg. January, 2010

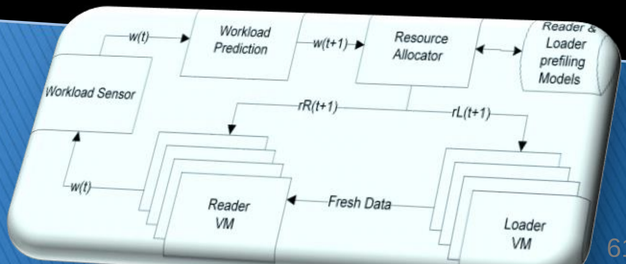
Acknowledgements

- Major Professor, Dr. Naphtali Rishe
- Committee Members
  - Dr. Ming Zhao
  - Dr. Tao Li
  - Dr. Malek Adjouadi
- Members of HPDRC Lab
- My friends

Thank you

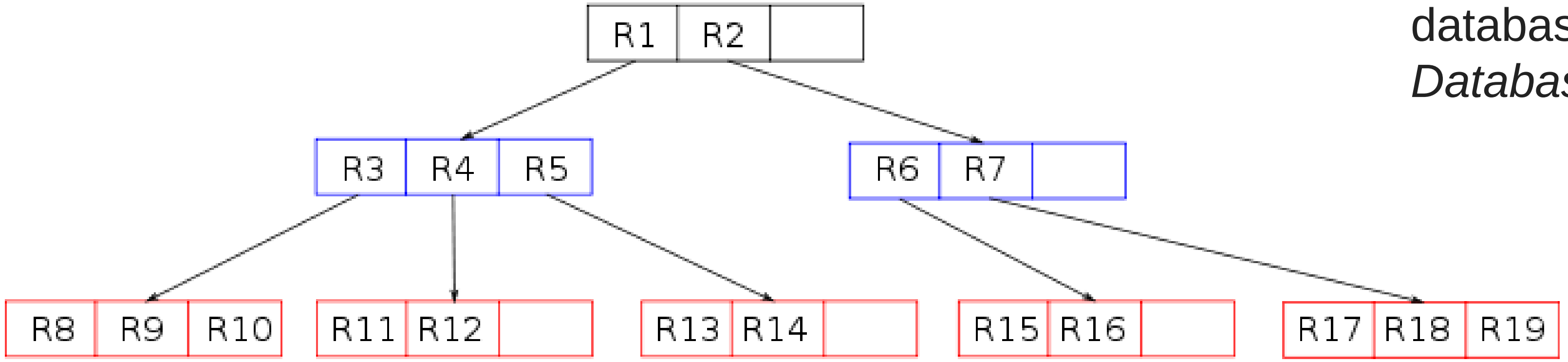





w(t)
Workload Prediction
w(t+1)
Resource Allocator
Reader & Loader prefiling Models
Workload Sensor
rR(t+1)
rL(t+1)
w(t)
Reader VM
Fresh Data
Loader VM

R-Tree KNN Search

- Queue nodes by distance to search point
- Locate the search point
- Keep the top k candidacy in cache
- Back trace and Subtract
- Finish



Hjaltason, G. R., & Samet, H. (1999). Distance browsing in spatial databases.*ACM Transactions on Database Systems*

SKS hybrid SKI search

- Similar to R-Tree KNN search, the best-first traversal algorithm proposed

- Replace the first operation with
  - For each entry e in node n do
    - If (isSubtressCandidate(B,n,[e's position in n])) then
      - Queue.push(e.ptr) with priority dist (e.MBR, l)
      - prune

- isSubtressCandidate evaluates B predicate by merging query term bitmaps on a range of super nodes, one super node at a time, until one candidate is found.

URLs

- http://vn4.cs.fiu.edu/cgi-bin/arquery.cgi?category=hotelsd_wikix2011_elevation&x1=-80.193573&y1=25.773941&vid=&referer=&place_name=Query++4+&extraref=1&arcriteria=1&star_rating%3E=4
- http://sksheavy.cs.fiu.edu:8080/sks/query?category=us_consumer_2012_full&y1=33.68881&x1=-116.18922&vid=&srvc=&&arcriteria=1&&timeout=20&d=99999999&numfind=200&maxeval=2000&printdist=1&header=1&CITY=miami&FIRST_NAME=jose
- homicide8893 http://geocloud.cs.fiu.edu/#7:38.69052889803671:-90.06382140624999/29:ps:ffff00:1:50:50:aaccff
- http://geocloud.cs.fiu.edu/