SPATIAL BIG DATA QUERY PROCESSING

Constantinos Costa

PhD Canditate

Data Management Systems Laboratory

Department of Computer Science

University of Cyprus

http://www.cs.ucy.ac.cy/~costa.c/

EPL670, March 31, 2015,

University of Cyprus, Nicosia, Cyprus

Presentation Outline

- **Introduction**
- Big Data Architectures
- Spatial Big Data Processing
- Spatial Big Data Visualization
- Conclusions
- Future work

Big Data

- *“**Big data** is a broad term for data sets **so large** or complex **that traditional** data processing applications **are inadequate**.” - Wikipedia*
- **Characteristics:**
  - **Size (Volume):** from a few **tens of terabytes** to many **petabytes** in a single database.
  - **Data model (Variety):** *structured* (relational, tabular), *semi-structured* (JSON) or *unstructured data* (Web & log files).
  - **Update Rate (Velocity):** second to minute granularity.
  - **Architectures:** highly *parallel* and *distributed* in order to cope with the inherent I/O and CPU limitations.
  - **Hardware:** mid-scale *private clouds* (datacenters), offering higher privacy, to *large-scale public clouds*.
  - **Functionality:***operational (OLTP)* and *analytic (OLAP)* functionality *stand-alone* or *as-a-Service.*

Spatial Big Data

- **Spatial Big Data** is Big Data with spatial attributes and properties.
  - Geo-tagged photos, text, videos, etc.
- ***Traditional*** *spatial data processing applications* ***are inadequate***
  - ***R-tree – computational cost in distributed enviroments***
- Sources of Spatial Big Data include:
  - Geo-Social Media
    - Twitter, Facebook, Google+ etc.
  - Location of readings of RFID
  - Satellite remote sensing
  - Aerial surveying
  - Sensor networks

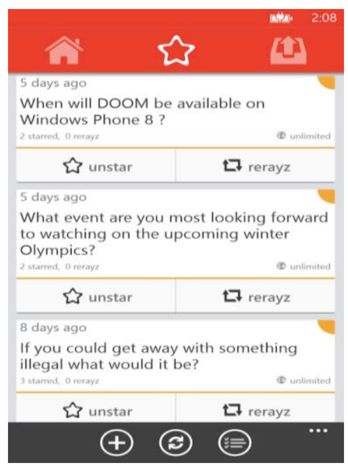
Query Processing

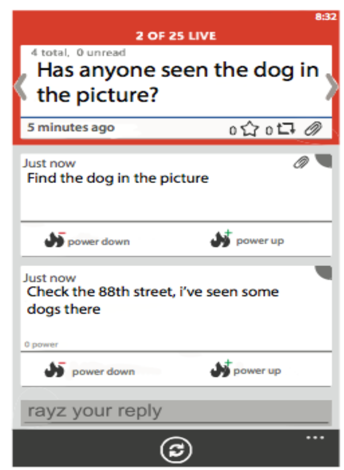
- *"How a data management system* ***evaluates*** *and* ***processes*** *a* ***query****" - Wikipedia*
- **In the RDBMS world:**
  - *(Disk)* ***Indexes****:* fast access to data records.
  - ***Catalogs****:* data statistics, histograms, etc.
  - *Query* ***Optimizers****:* determine most efficient way to execute a query using operators, indexes, catalogs
- **In the Big-Data world:**
  - There is not **enough time** (and **space**) to build to much **meta-data** (indexes, catalogs) on processed data.
  - Some data might never actually be touched upon (e.g., one of hundreds of FLAGS in a telco setting).
- **Objective:** Effective Operators for specific queries.

Motivation

- Rayzit is a **crowd messaging technology** that delivers **questions, inquiries and ideas** to the **kNN users.**
  - Not location-based app (e.g., 5 km) but kNN-based!
- Rayzit was funded by the Appcampus Program (Microsoft, Nokia & Aalto, Finland) in 2013.
  - Ranked among 5 best apps of the program from 3500 apps.
  - Few thousand downloads and active users after their marketing!
- **Research Problem:** Big Data AkNN Query Processing

http://rayzit.com/





Windows Store

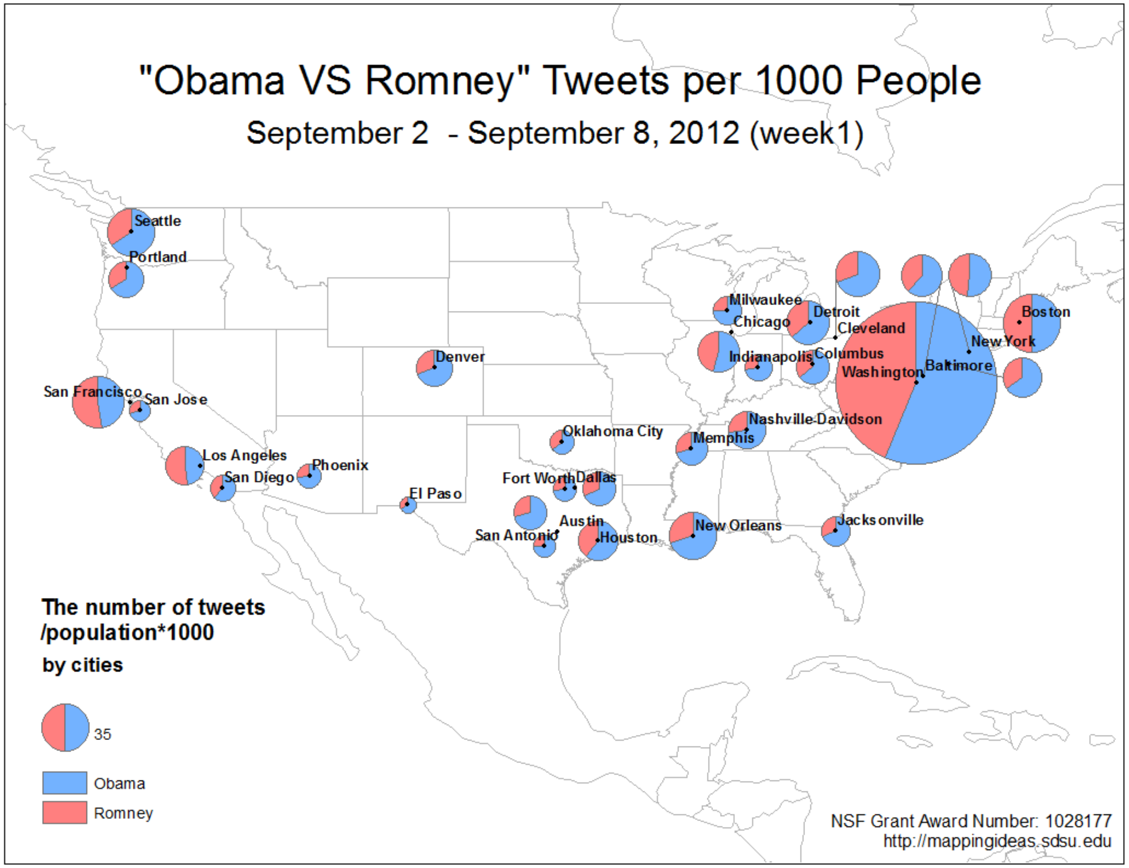
Download on the App Store

GET IT ON Google play

C. Costa, C. Anastasiou, G. Chatzimilioudis, and D. Zeinalipour-Yazti. Rayzit: An anonymous and dynamic crowd messaging architecture. In Mobisocial'15 collocated with *Mobile Data Management (MDM), 2015*, June 2015.

Motivation

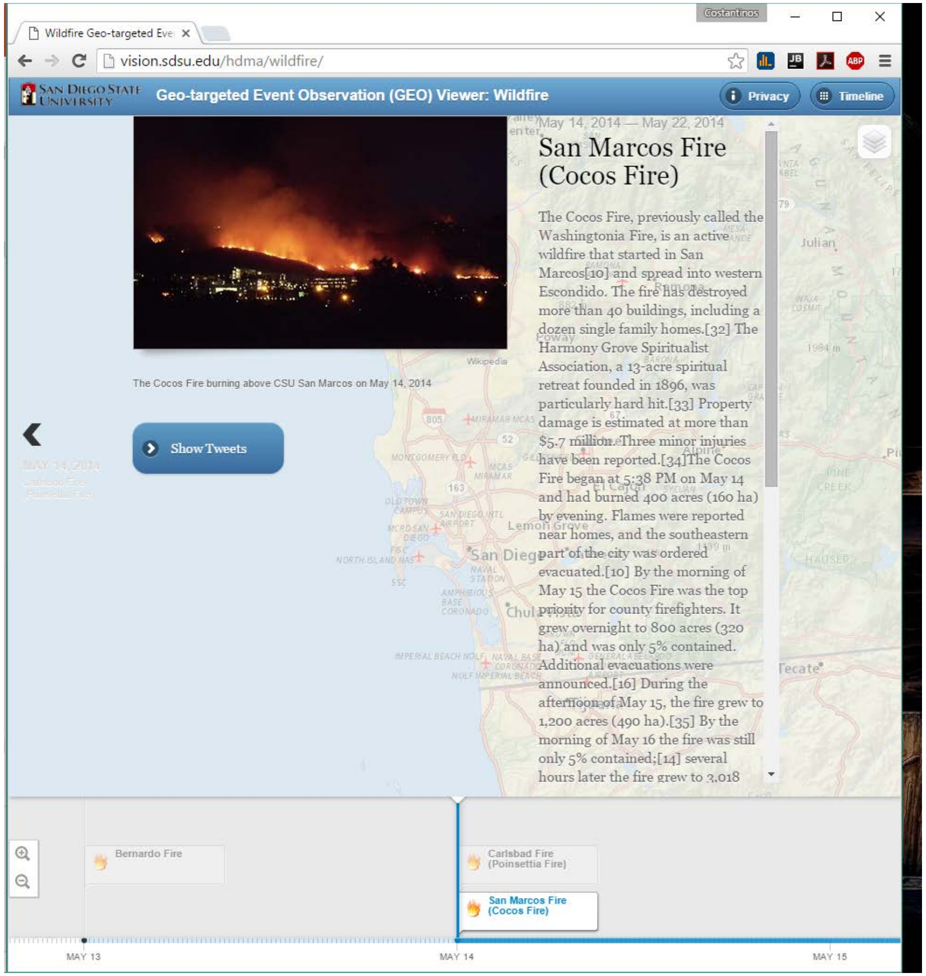
- Spatial Big Data for 2012 Presidential Election



http://mappingideas.sdsu.edu/mapshowcase/election/media/obamaVSromney.html

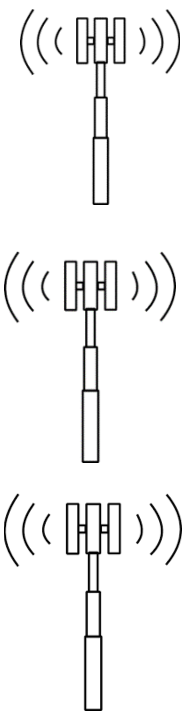
Motivation

- Physical Disasters could be located or avoided.



http://vision.sdsu.edu/hdma/wildfire/

Motivating Scenario (Telco)

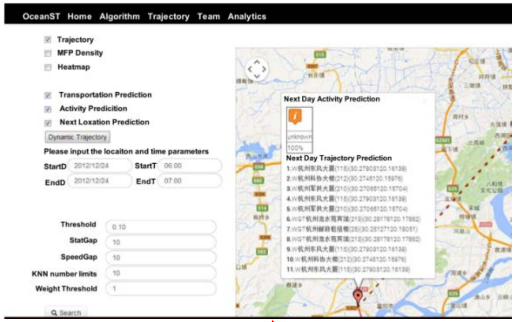


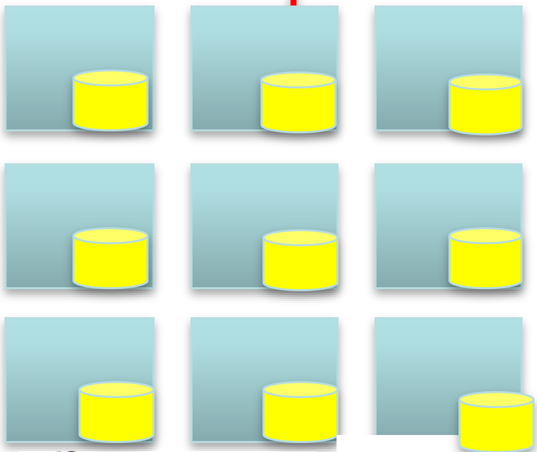
Mobile Broadband Data (MBB)

`(userID, productID, fromNo, deviceID, call drops, bandwidth …)`

5TB MBB / day (Shenzhen, China 10M Huawei customers)

Visual Analytics, Predictions, etc.

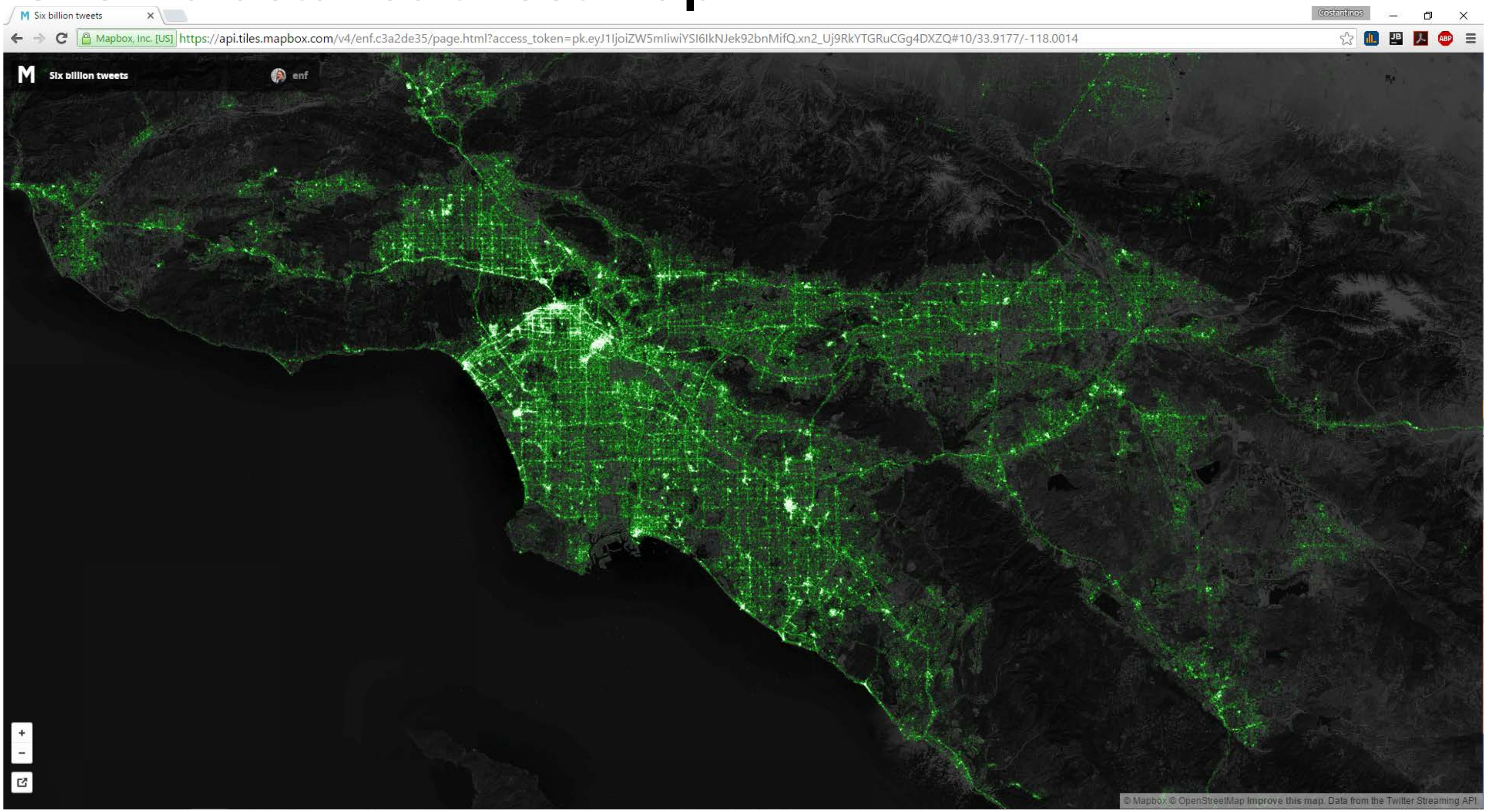




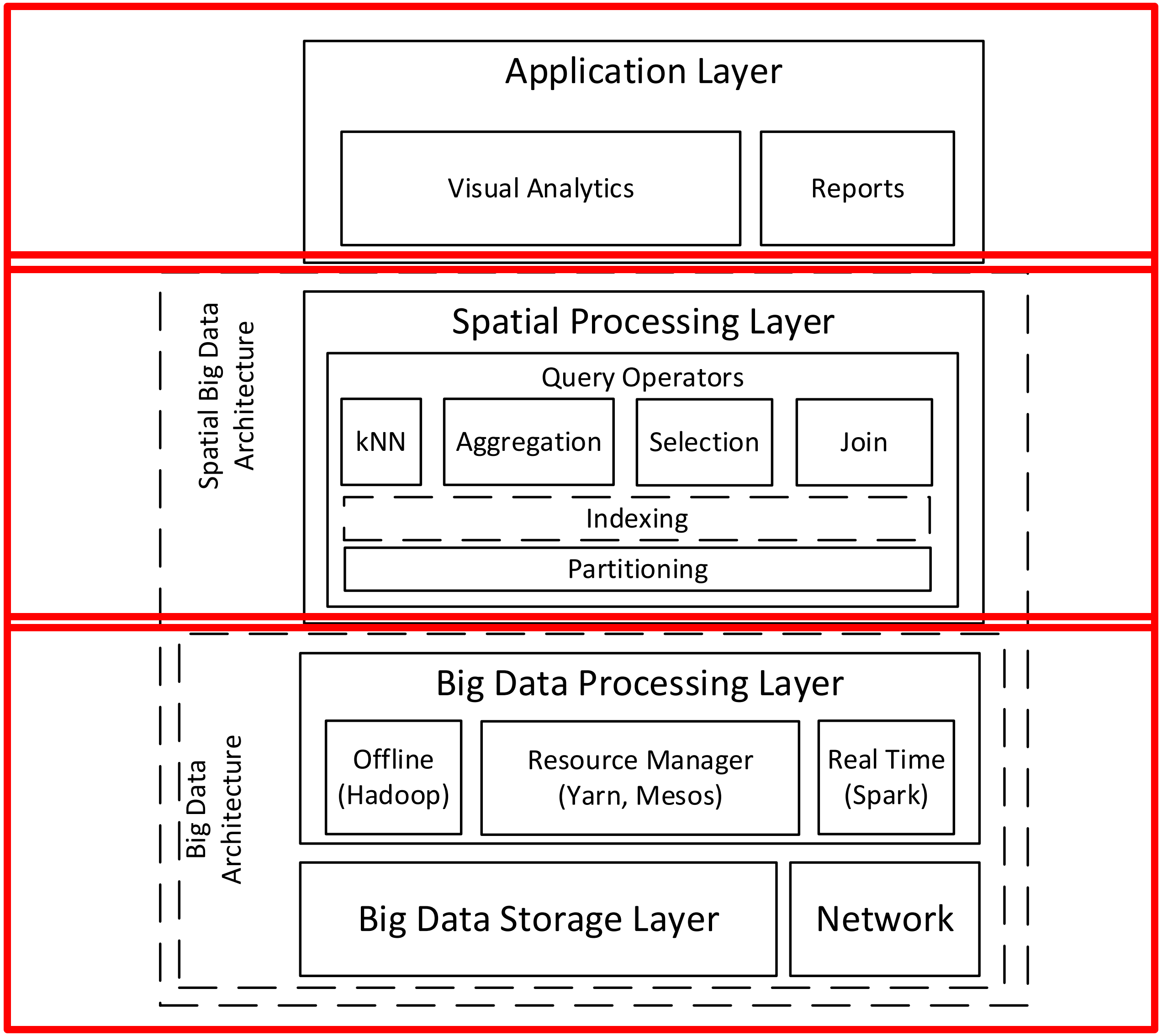
Churn Prediction @ ACM SIGMOD'15.

Motivating Scenario

Show a detailed tweet map

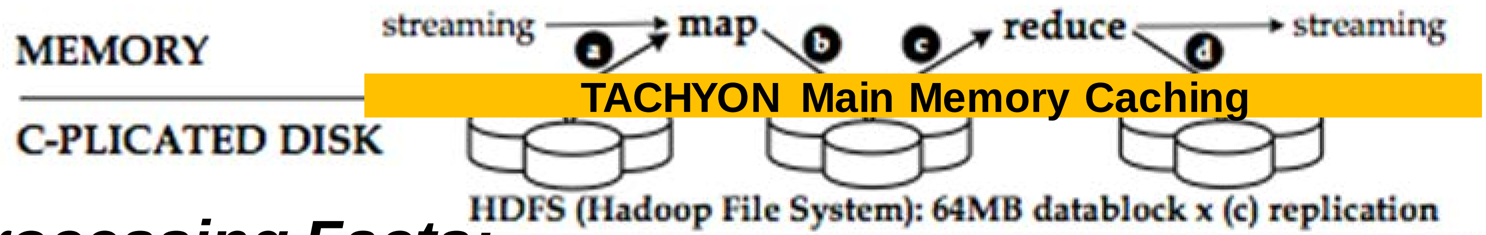


Architecture Outline



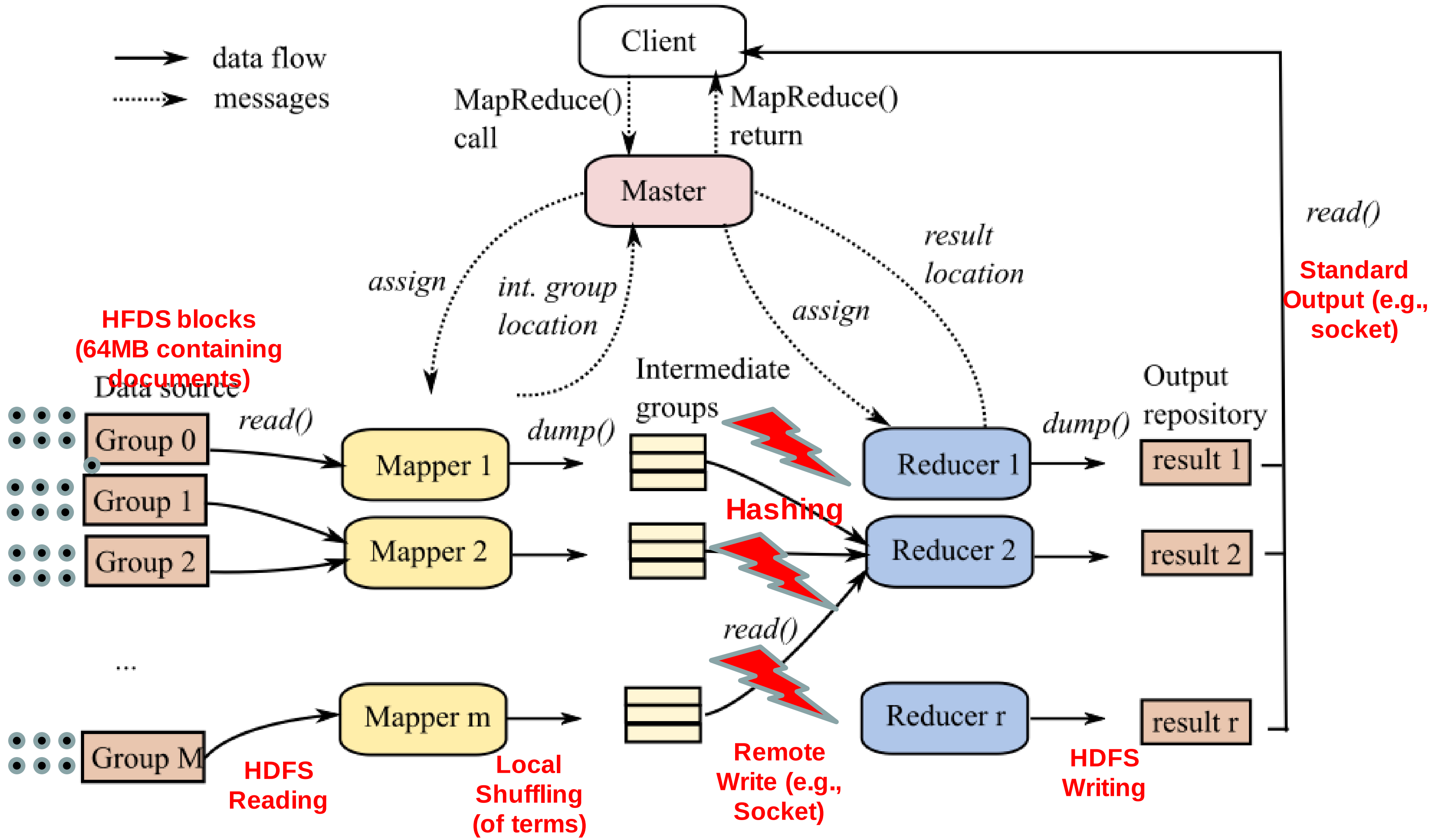
Hadoop Map Reduce

- ***Map-Reduce (MR):*** *a* ***programming model for processing*** *large data sets.*
  - *Google paper @ USENIX OSDI'04*
  - ***Hadoop:*** *Apache's open-source MR software.*
  - **Commodity Hardware:** 2-3 failures/1000 nodes/day
    ➔ Data Sharding (Data blocks replicated to 3 disks)



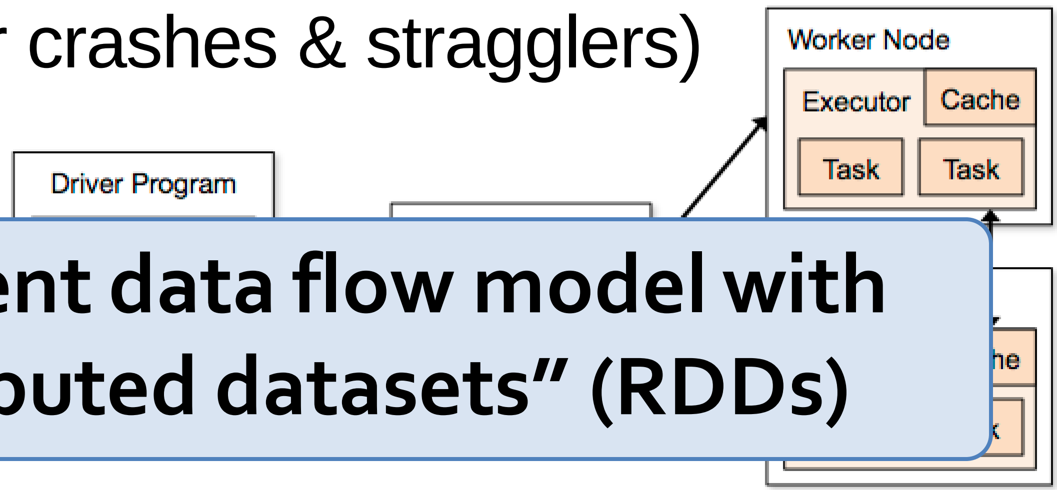
- ***Processing Facts:***
  - 1 TB on 1 PC = 2 hours!
  - 1 TB on 100 PCs = 1 min!
- I/O tasks can be turned into Main -Memory (Tachyon)

Hadoop Map Reduce



Spark

- Provide distributed memory abstractions for clusters to support apps with working sets
- Retain the attractive properties of MapReduce:
  - Fault tolerance (for crashes & stragglers)
  - Data locality



Solution: augment data flow model with "resilient distributed datasets" (RDDs)

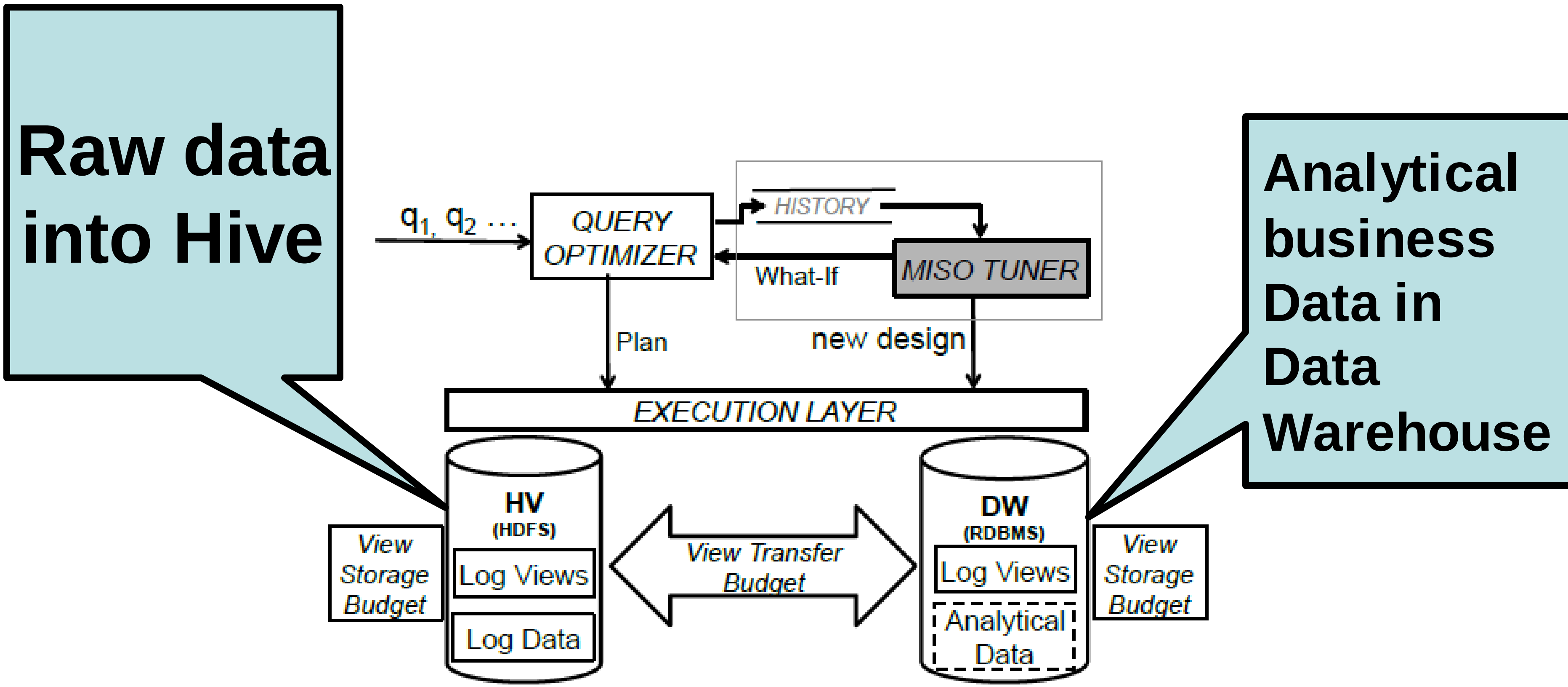
Generality of RDDs

- We conjecture that Spark's combination of data flow with RDDs unifies many proposed cluster programming models
  - General data flow models: MapReduce, Dryad, SQL
  - Specialized models for stateful apps: Pregel (BSP), HaLoop (iterative MR), Continuous Bulk Processing
- Instead of specialized APIs for one type of app, give user first-class control of distrib. datasets

Presentation Outline

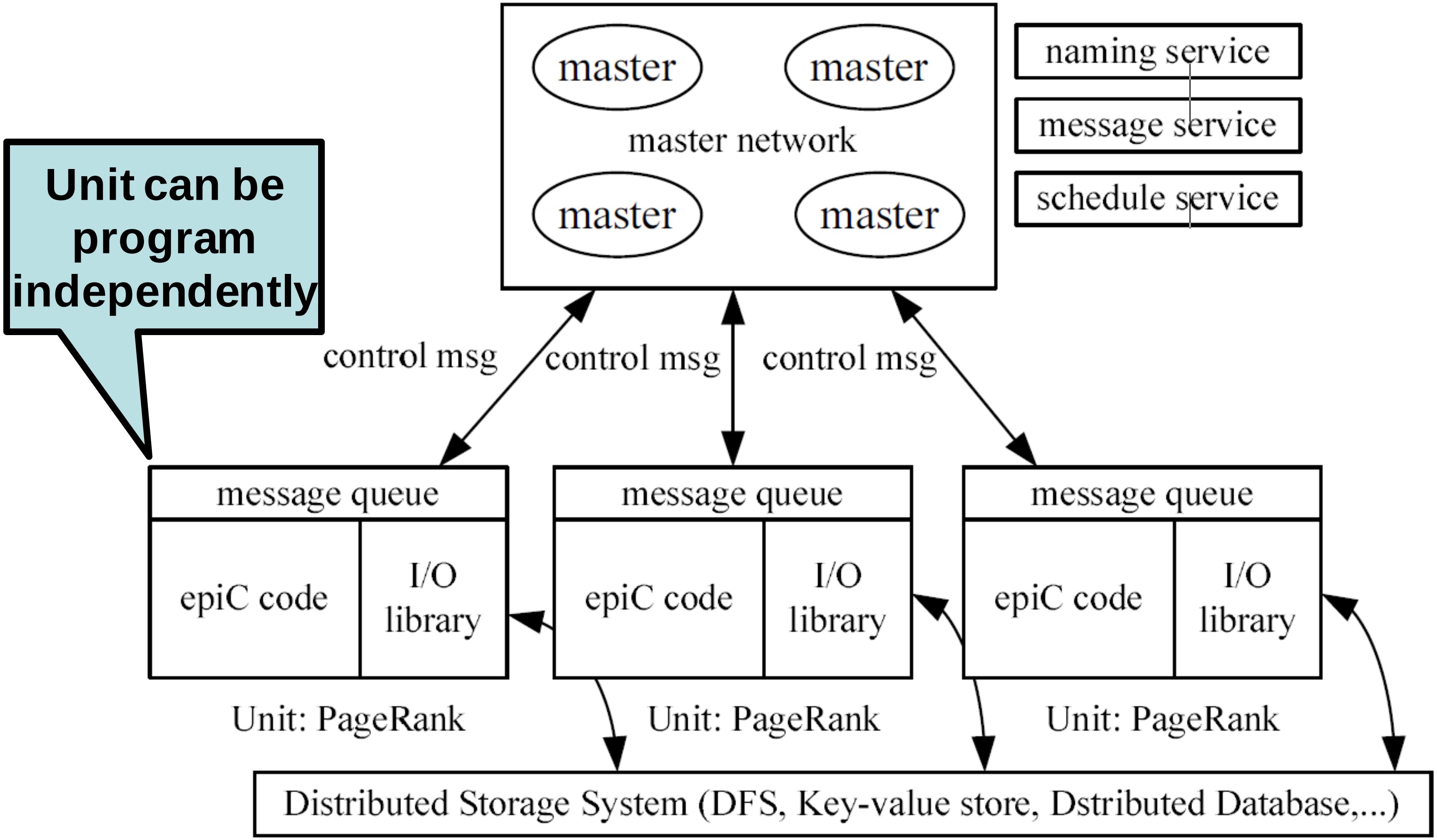
- Introduction
- **Big Data Architectures**
- Spatial Big Data Processing
- Spatial Big Data Visualization
- Conclusions
- Future work

MISO



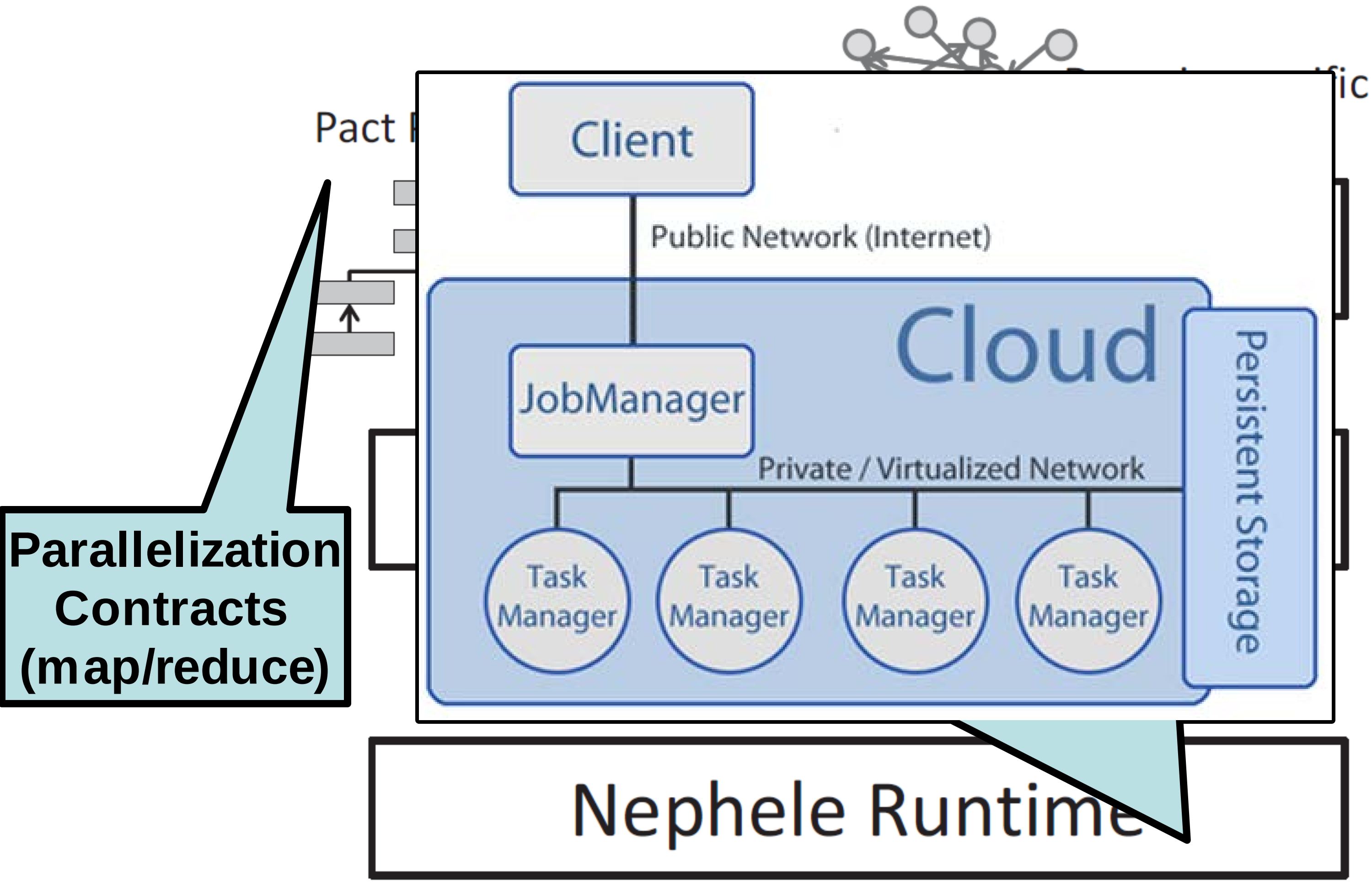
J. LeFevre, J. Sankaranarayanan, H. Hacigumus, J. Tatemura, N. Polyzotis, and Michael J. Carey. Miso: Souping up big data query processing with a multistoresystem. SIGMOD ’14

epiC



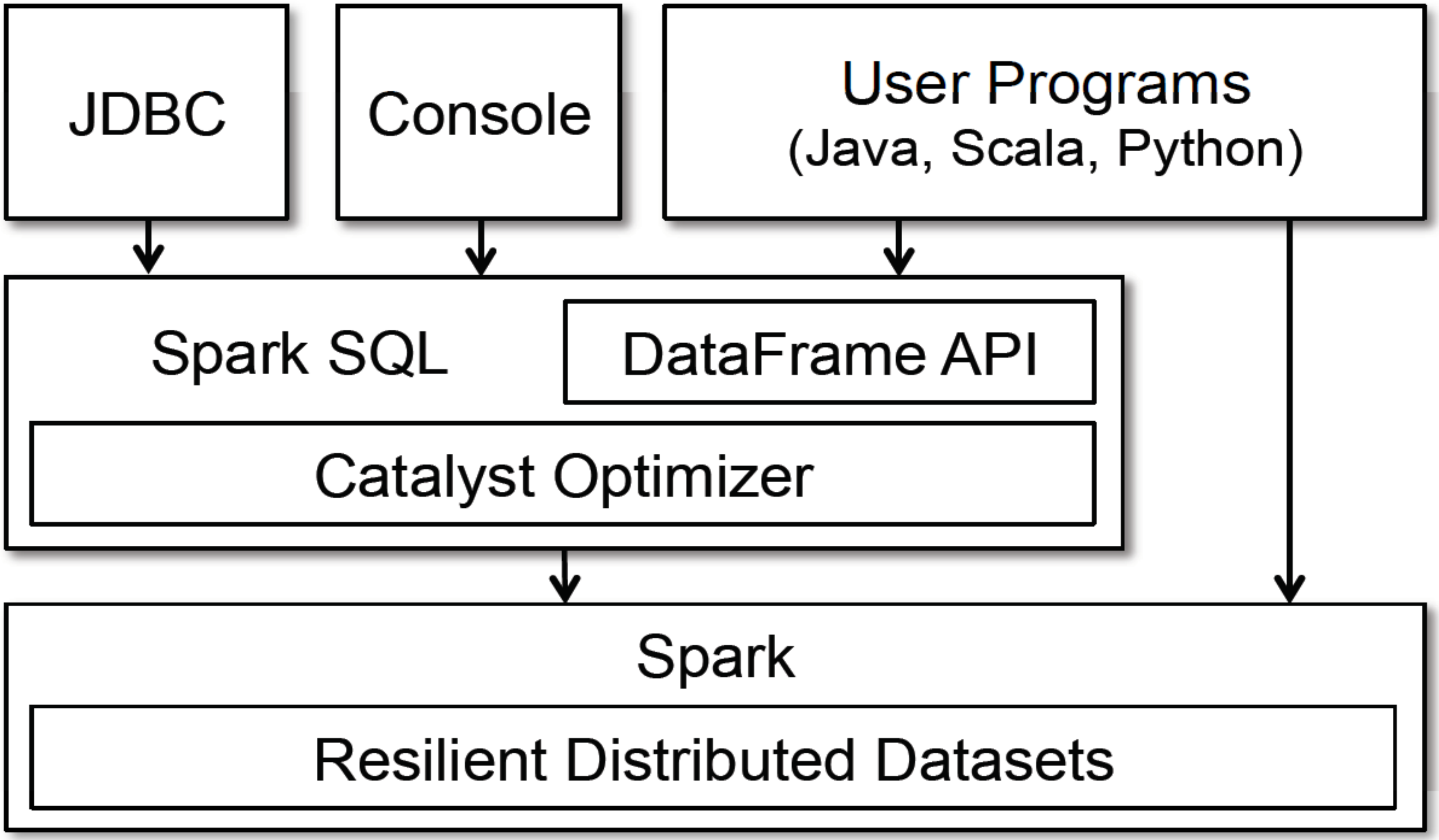
D. Jiang, G. Chen, B. Chin Ooi, K. Tan, and Sai Wu. epic: An extensible and scalable system for processing big data. *VLDB* 2014.

Stratosphere



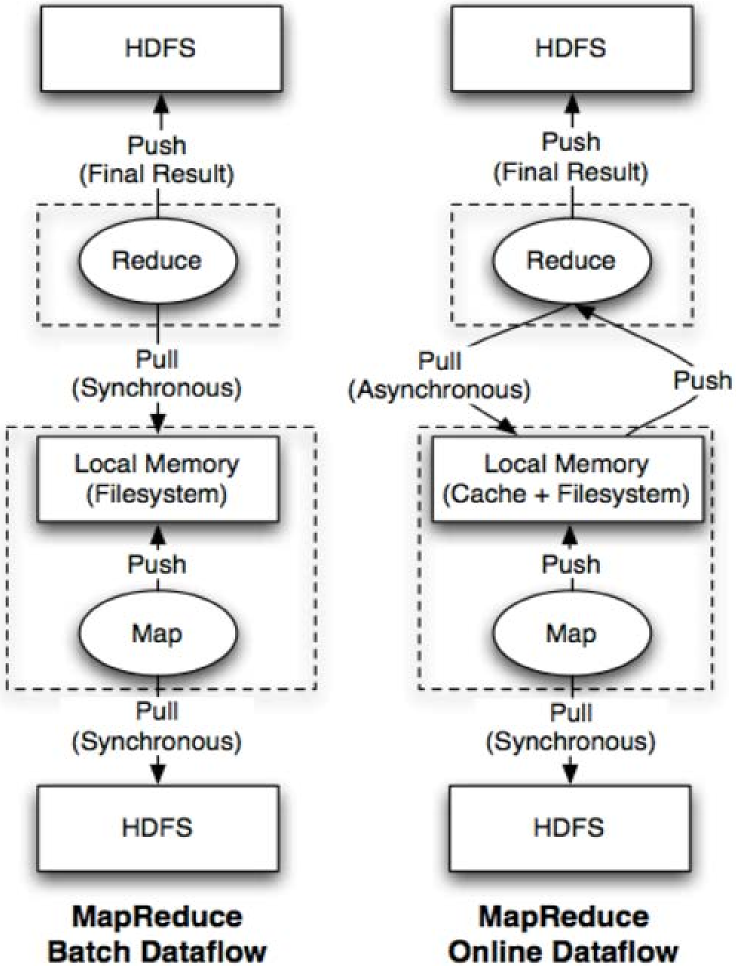
S. Ewen, S. Schelter, K. Tzoumas, D. Warneke, V. Markl. Iterative parallel data processing with stratosphere: An inside look. SIGMOD ’13

Spark SQL



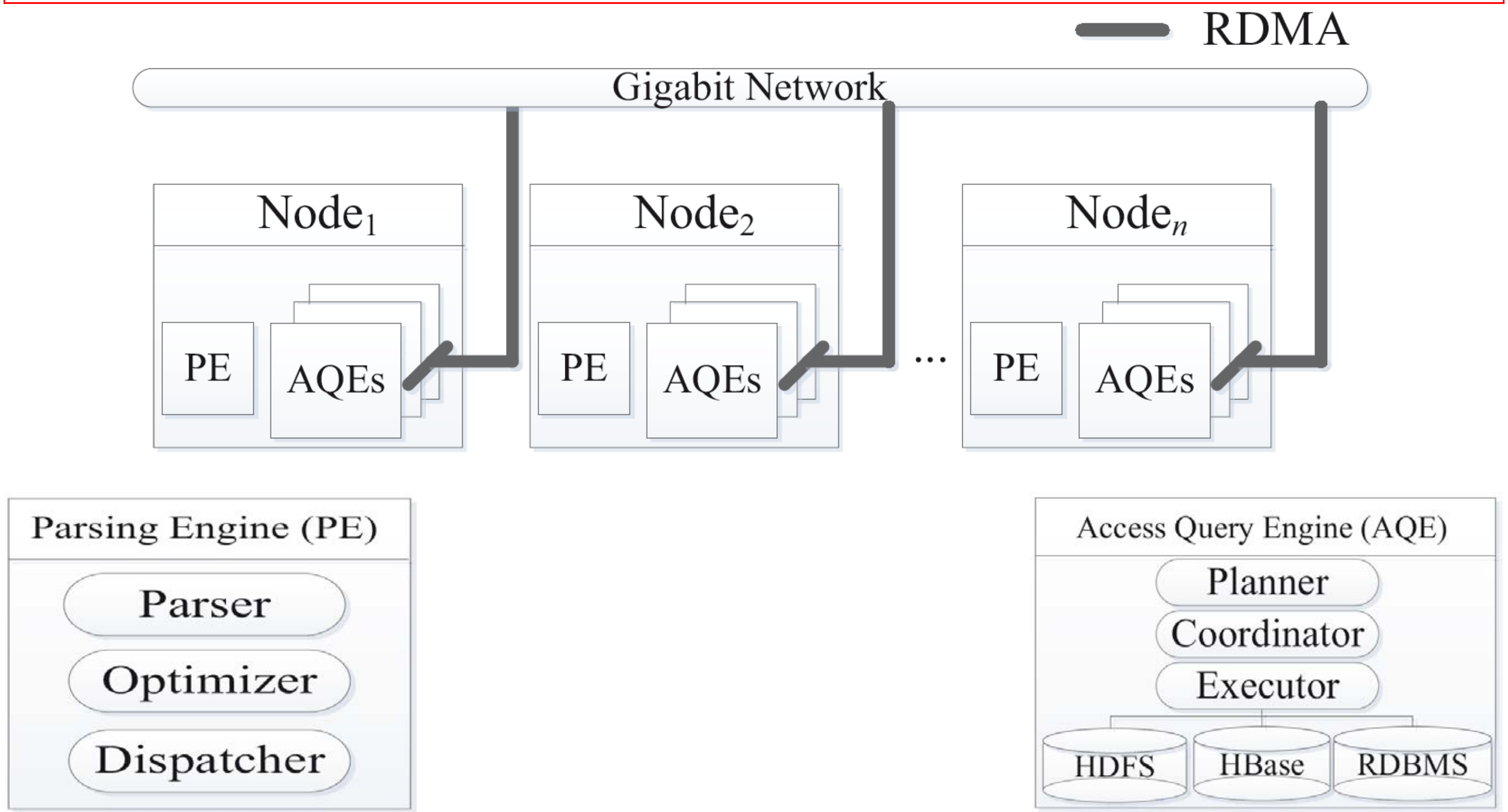
M.Armbrust, R. Xin, C. Lian, Y. Huai, D. Liu, J. K. Bradley, X. Meng, T. Kaftan, M. J. Franklin, A. Ghodsi, and M. Zaharia. Spark sql: Relational data processing in spark. SIGMOD '15

MapReduce Online



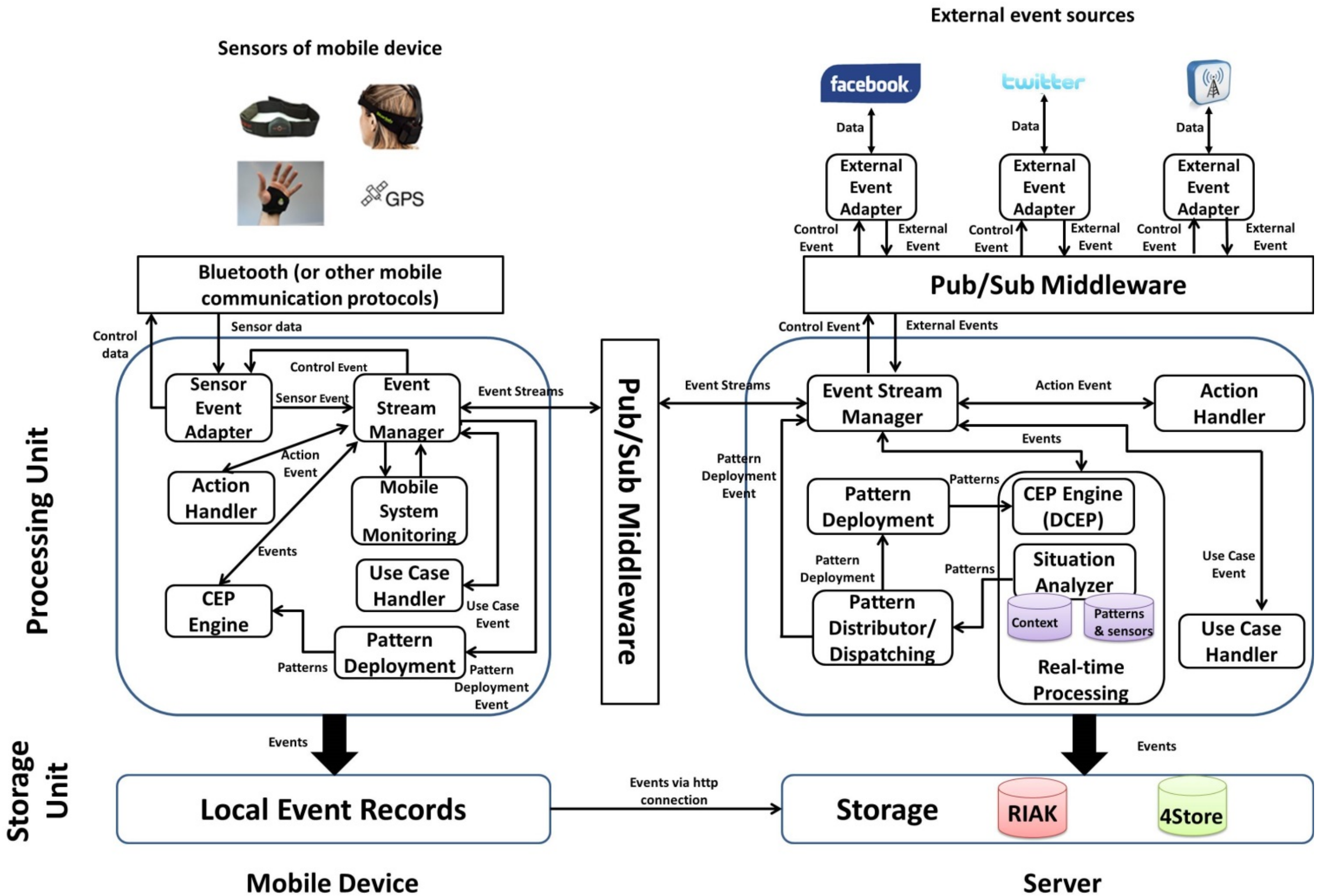
T. Condie, N. Conway, P. Alvaro, J. Hellerstein, K. Elmeleegy, R. Sears. Mapreduce online. NSDI'10

OceanRT



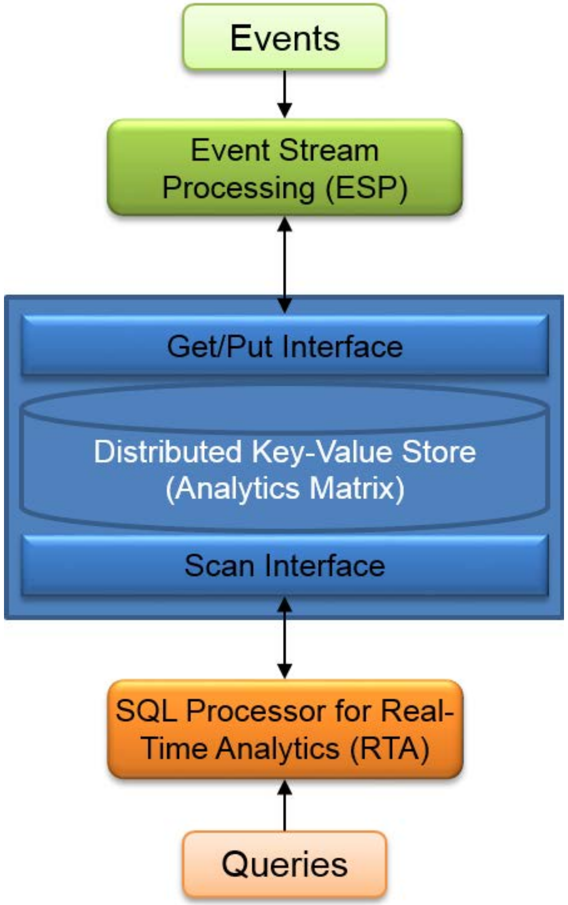
S. Zhang, Y. Yang, W. Fan, L. Lan, and M. Yuan. Oceanrt: Real-time analytics over large temporal data. SIGMOD '14

CEP



N. Stojanovic, L. Stojanovic, Y. Xu, B. Stajic. Mobile cep in realtime big data processing: Challenges and opportunities. DEBS '14

AIM

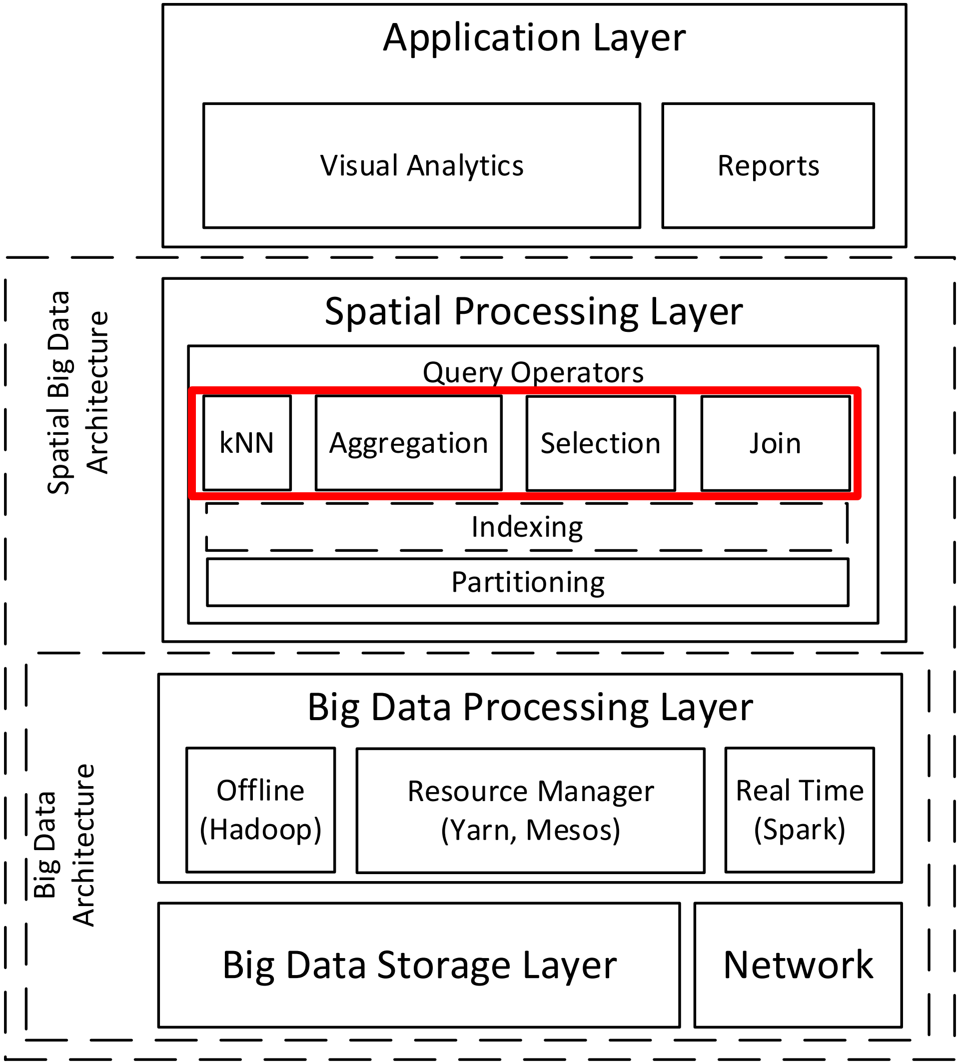


L. Braun, T. Etter, G. Gasparis, M. Kaufmann, D. Kossmann, D. Widmer, A. Avitzur, A. Iliopoulos, E. Levy, N. Liang. Analytics in motion: High performance event-processing and real-time analytics in the same database. SIGMOD '15

Presentation Outline

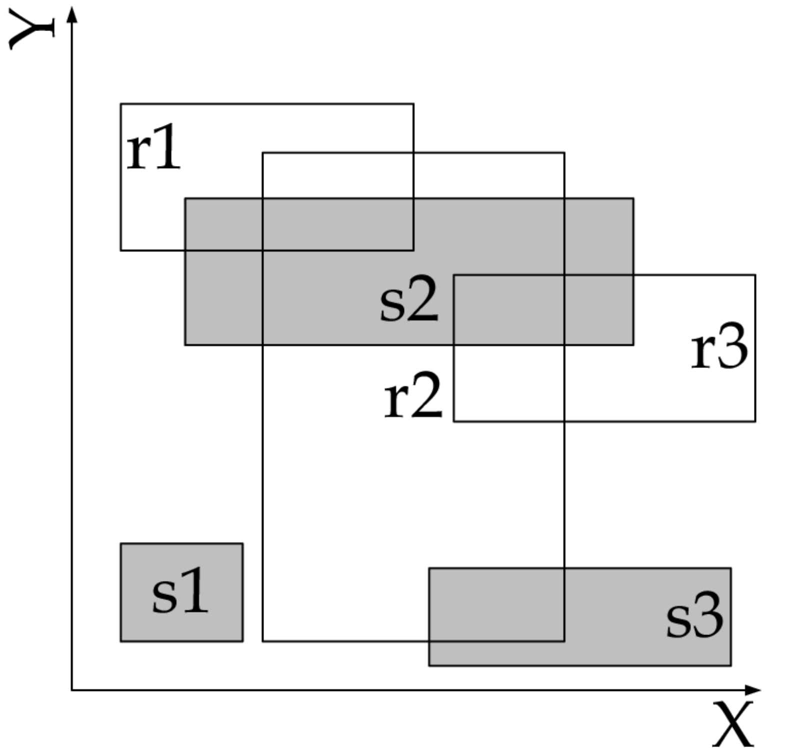
- **Introduction**
- **Big Data Architectures**
- **Spatial Big Data Processing**
- Spatial Big Data Visualization
- Conclusions
- Future work

Query Operators



Query Operators: JOIN

- The spatial join will return various subsets of interest, e.g., $r_1 \bowtie s_2$, $r_2 \bowtie s_2$, $r_2 \bowtie s_3$, and $r_3 \bowtie s_2$
- An example a spatial join query would be **“Find roads that cross rivers”**



Query Operators: Selection

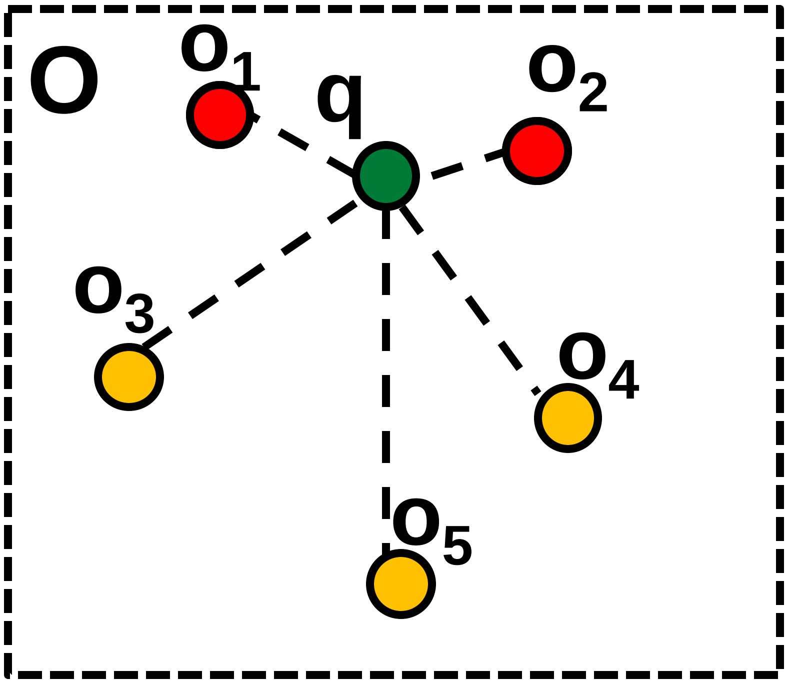
- The spatial selection query finds all the containing points within a query range or a query region
- A range query might include **“Find all countries that are contained by a supplied query region”**

Query Operators: Aggregation

- The purpose of a spatial aggregation query is to combine several geometric characteristics in order to produce the result
- For example a union query may be **“find the average area of the all countries”**

Query Operators: kNN

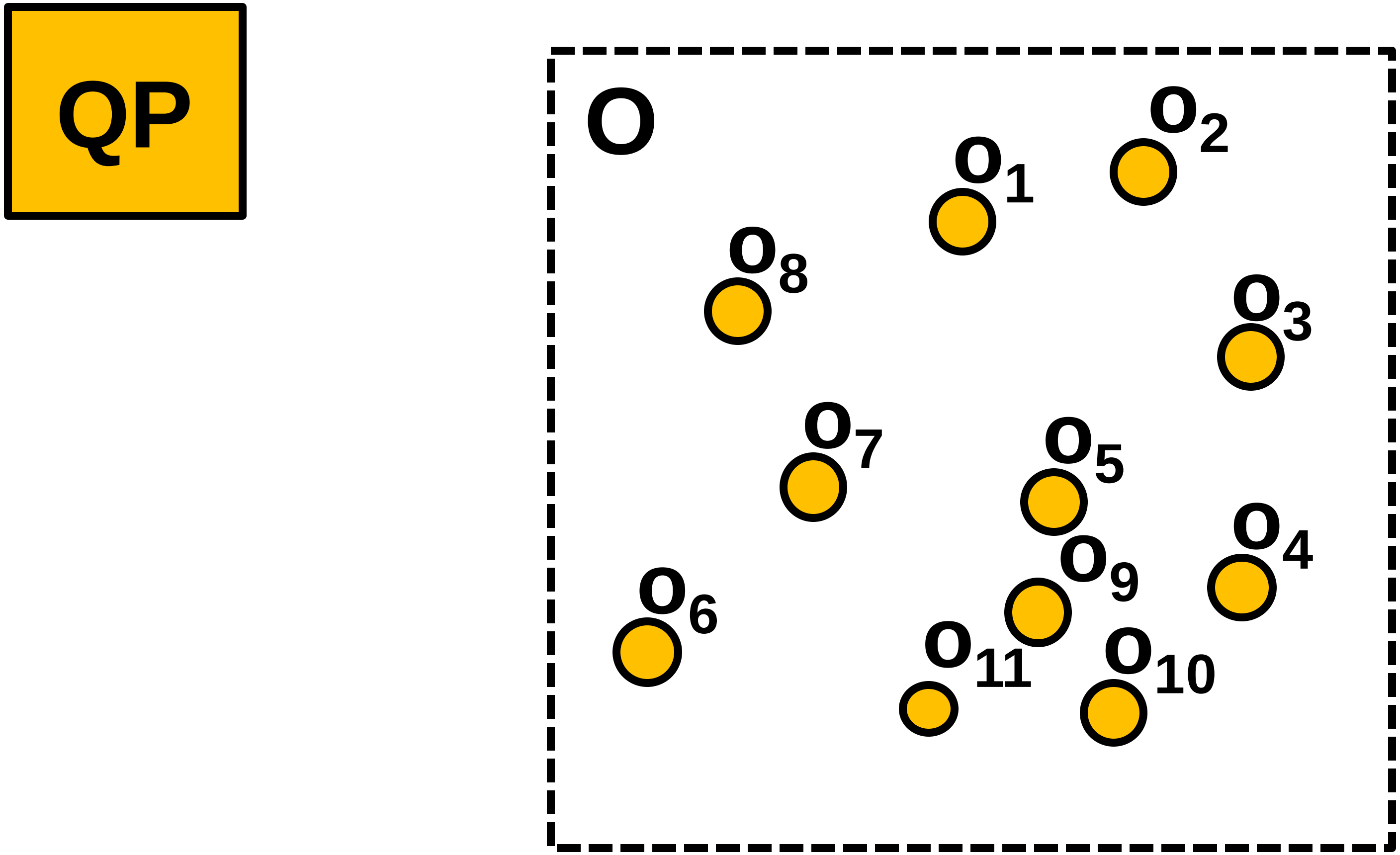
- ***kNN(q, O):*** *Given a multi-dimensional query point **q**, find the **k** objects in **O** that have the smallest distance* to **q**.*
  - **e.g. Manhattan (L_1),* ***Euclidean (L_2),*** *Chebyshev(L_∞)*



2NN(q,O) = {o_1,o_2}

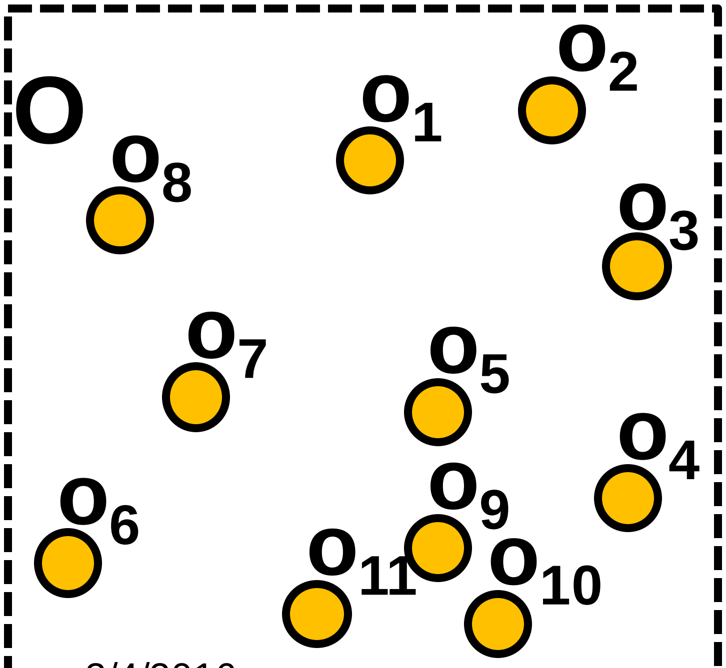
Query Operators: kNN

- ***|O| = n*** *objects report location to* ***QP***
- ***QP*** *computes* ***AkNN*** *query every* ***few seconds****.*



AkNN Query Definition

- ***Find kNN for all objects in O!***
- ***AkNN(O):*** *Given objects* $o_a \neq o_b \neq o_c$, $\forall o_b \in kNN(o_a, O)$ *and* $\forall o_c \in O - kNN(o_a, O)$ *it holds that* $dist(o_a, o_b) \leq dist(o_a, o_c)$



AkNN = kNN Self-Join

kNN Self-Join Definition

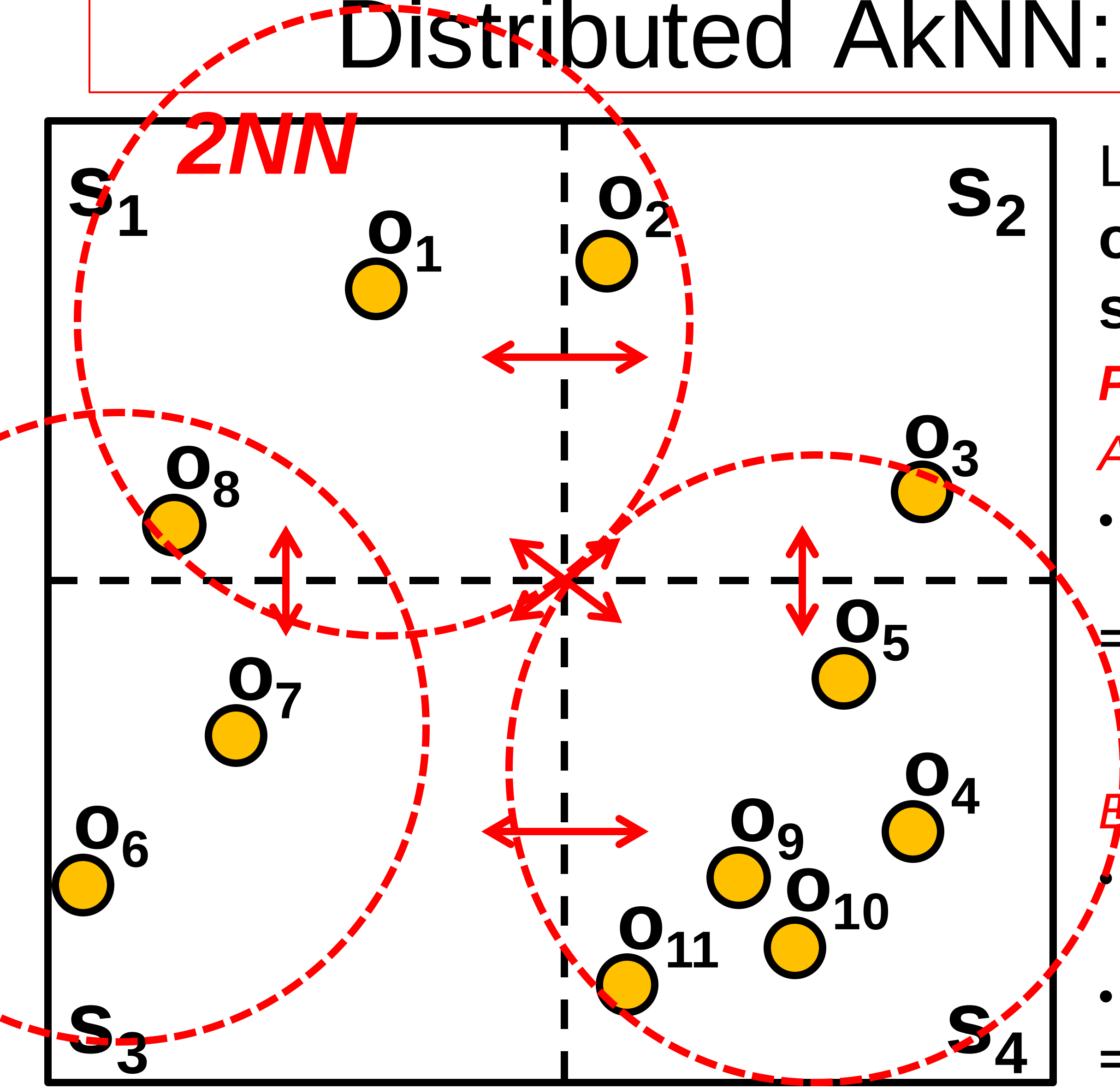
$O \bowtie_{kNN} O =$
$\{(o_a, o_b) \mid o_a, o_b \in O \text{ and } o_b \in kNN(o_a, O)\}$

$O(n^2)$-time search cost

Centralized AkNN

- *Classical problem with many centralized algorithms*
- **Applications:** computational geometry, image processing, data compression, clustering and spatial databases.
- **Basic Types of Algorithms:**
  - Memory-resident structures & algorithms (theory)
  - Disk-resident structures & algorithms (databases)
  - Parallel algorithms (Graphics)
- **Common Problem:**
  - Not designed for distributed Big Data Architectures (shared-nothing cloud infrastructure)

Distributed AkNN: Issues



Let us partition the **11 objects** over **4 servers** $\{s_1,\ldots,s_4\}$

Problems:

A) Communication Overhead

- ***o_1:*** *2NN(o_1) are located on adjacent nodes s1 and s2*

=> We need to minimize the replication (f) among nodes!

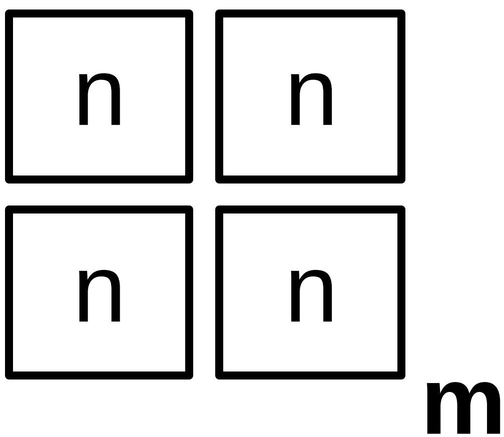
B) Load Balancing

- ***s_4:*** *15 distances among 6 objects (i.e., n(n-1)/2)*
- ***s_3:*** *only 3 distances!*

=> We need to yield a fair space partitioning!

Hadoop Naïve Join (H-NJ)[40]

- **H-NJ Algorithm:**
  - Transfer **O** to all **m** servers
  - On each server, execute $\mathbf{O_i} \bowtie_{\mathbf{kNN}} \mathbf{O}$
    - $\mathbf{O_i}$ contains n/m objects
  - Heavy $\mathbf{O(n^2/m)}$ processing cost
  - Heavy $\mathbf{O(mn)}$ communication cost

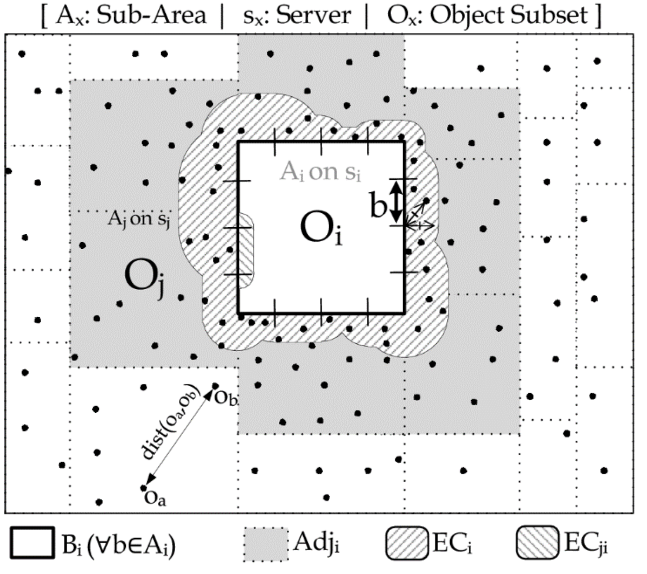


Replication Factor
$f_{H\text{-}NJ}=m$

[40] Lu et. al VLDB'12

The Spitfire Algorithm

- **Spitfire Execution Phases**
  – Partition problem space (each cell at least k)
  – Replicate border cases minimally.
  – Perform local AkNN computation



Distributed Algorithm implemented in MPI

"Distributed In-Memory Processing of All k Nearest Neighbor Queries, Georgios Chatzimilioudis, Costantinos Costa, Demetrios Zeinalipour-Yazti, Wang-Chien Lee, Evaggelia Pitoura, IEEE TKDE'15

Theoretical Comparison

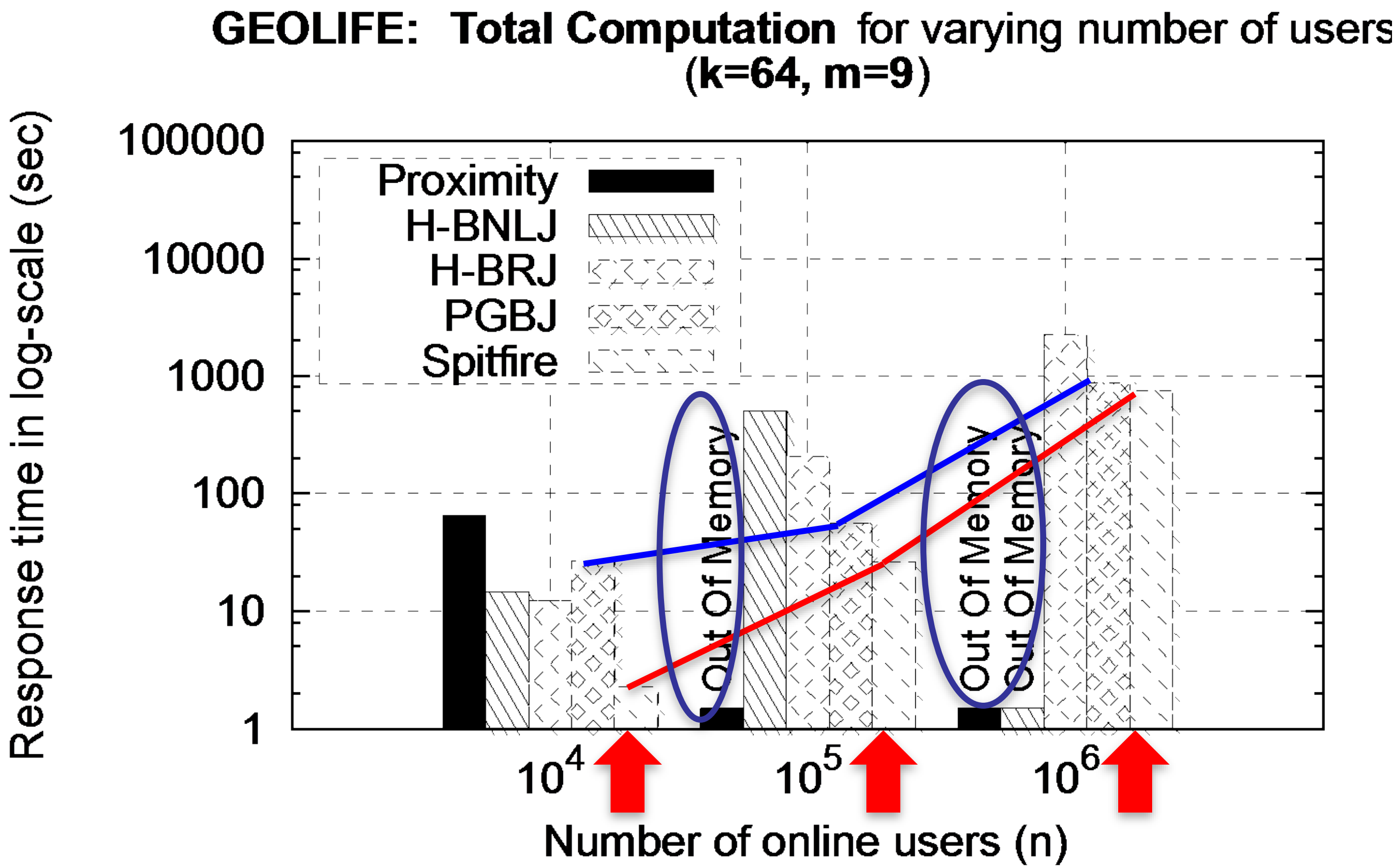
Algorithms for Distributed Main-Memory AkNN Queries

[n: objects | m: servers | f : replication factor | $f << m < n$]

Algorithm	Preproc.	Part. & Replic.	Refinement	Communic.
H-NJ [17]	-	$O(n)$	$O(\frac{n^2}{m})$	$O(mn)$
H-BNLJ [29]	-	$O(n)$	$O(\frac{n^2}{m})$	$O(\sqrt{m}n)$
H-BRJ [29]	-	$O(n)$	$O(n \log \frac{n}{m})$	$O(\sqrt{m}n)$
PGBJ [17]	$O(\sqrt{n})$	$O(n^{\frac{3}{2}}/m)$	$O(f_{\mathrm{PGBJ}} \frac{n^2}{m^2})$	$O(f_{\mathrm{PGBJ}} n)$
Spitfire	-	$O(n)$	$O(f_{Spitfire} \frac{n^2}{m^2})$	$O(f_{Spitfire} n)$

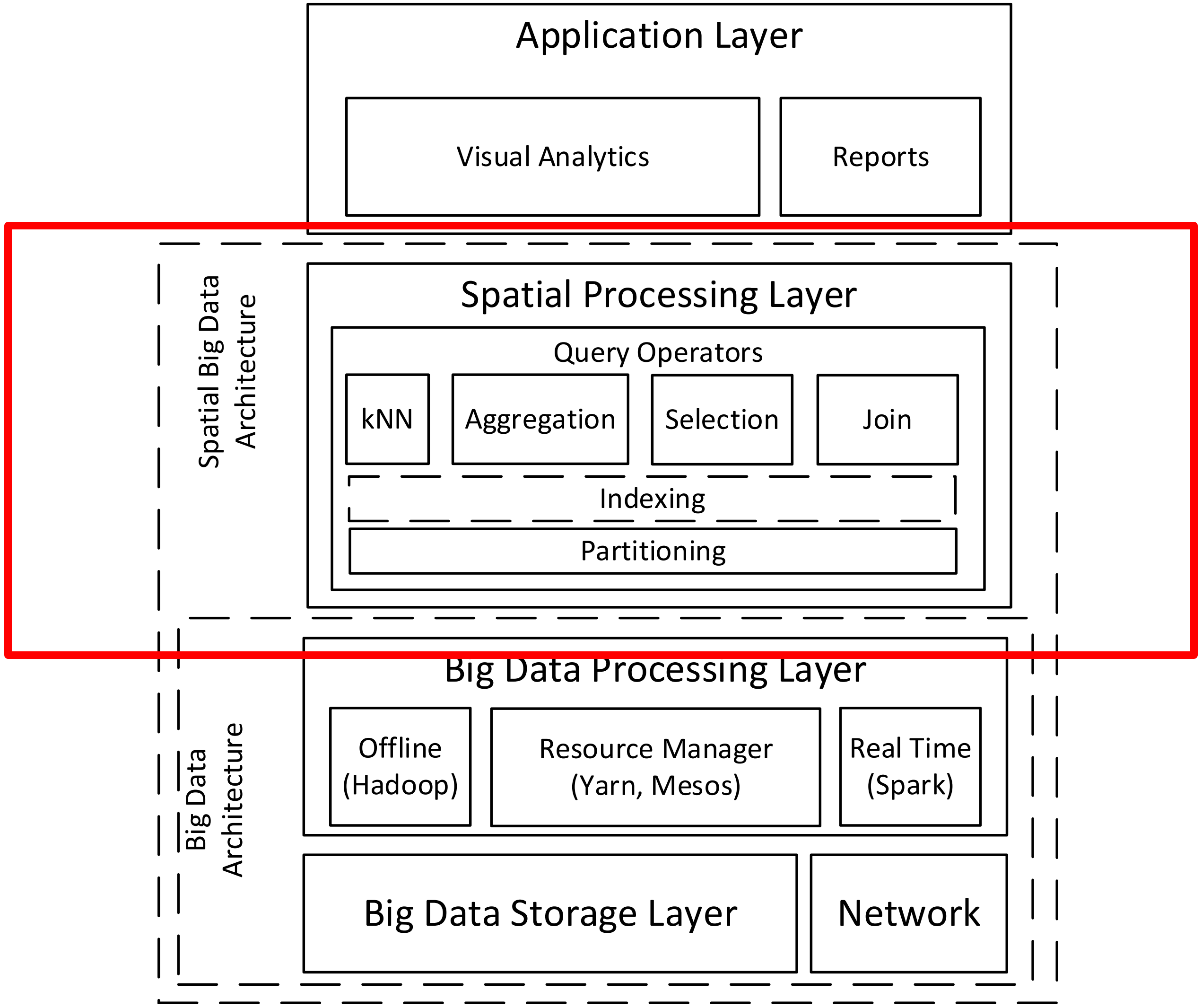
$f_{Spitfire}$ achieved is approx. $2^{0.5}$ times smaller than f_{PGBJ}

Experimental Evaluation

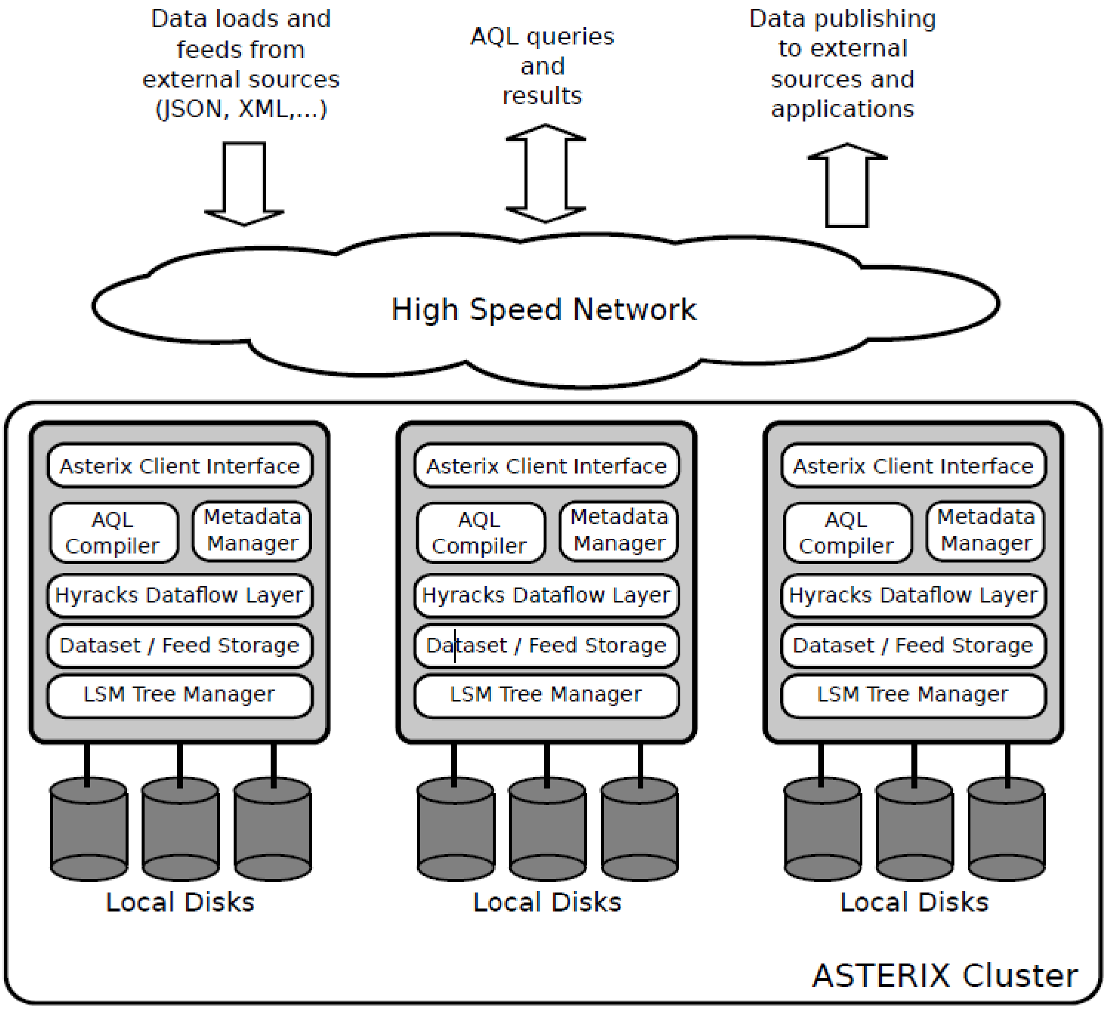


- Plot for most skewed dataset. Others have similar results
- Spitfire outperforms
- Spitfire and PGBJ scale better
  than H-BNLJ and H-BRJ who run out of memory

Architecture Outline

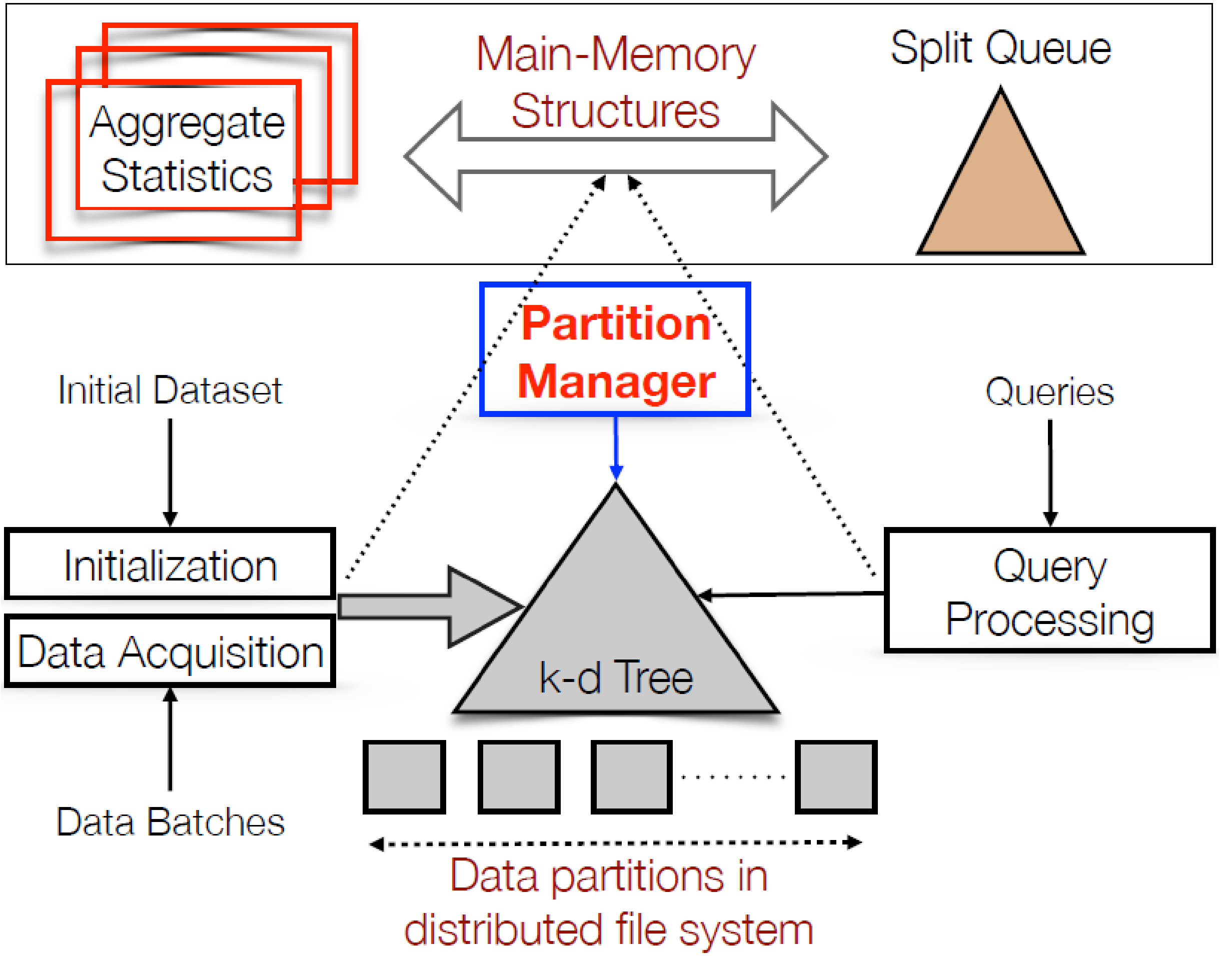


Asterix



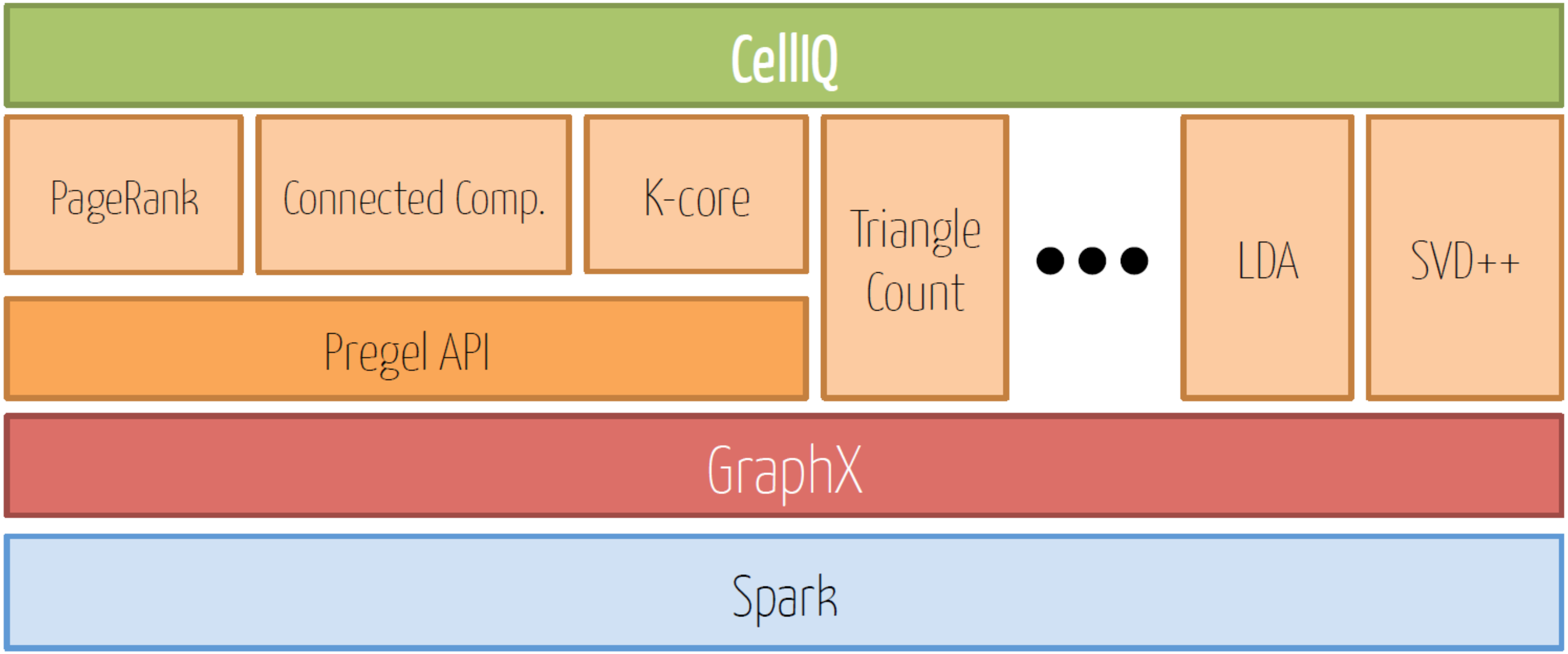
S. Alsubaiee, A. Behm, R. Grover, R. Vernica, V. Borkar, M. J. Carey, C. Li. Asterix: Scalable warehouse-style web data integration. IIWeb '12

Aqwa



A. Aly, A. Mahmood, M. Hassan,W. Aref, M. Ouzzani, H. Elmeleegy, and T. Qadah. Aqwa: Adaptive query workload aware partitioning of big spatial data. *VLDB* 2015.

CellIQ

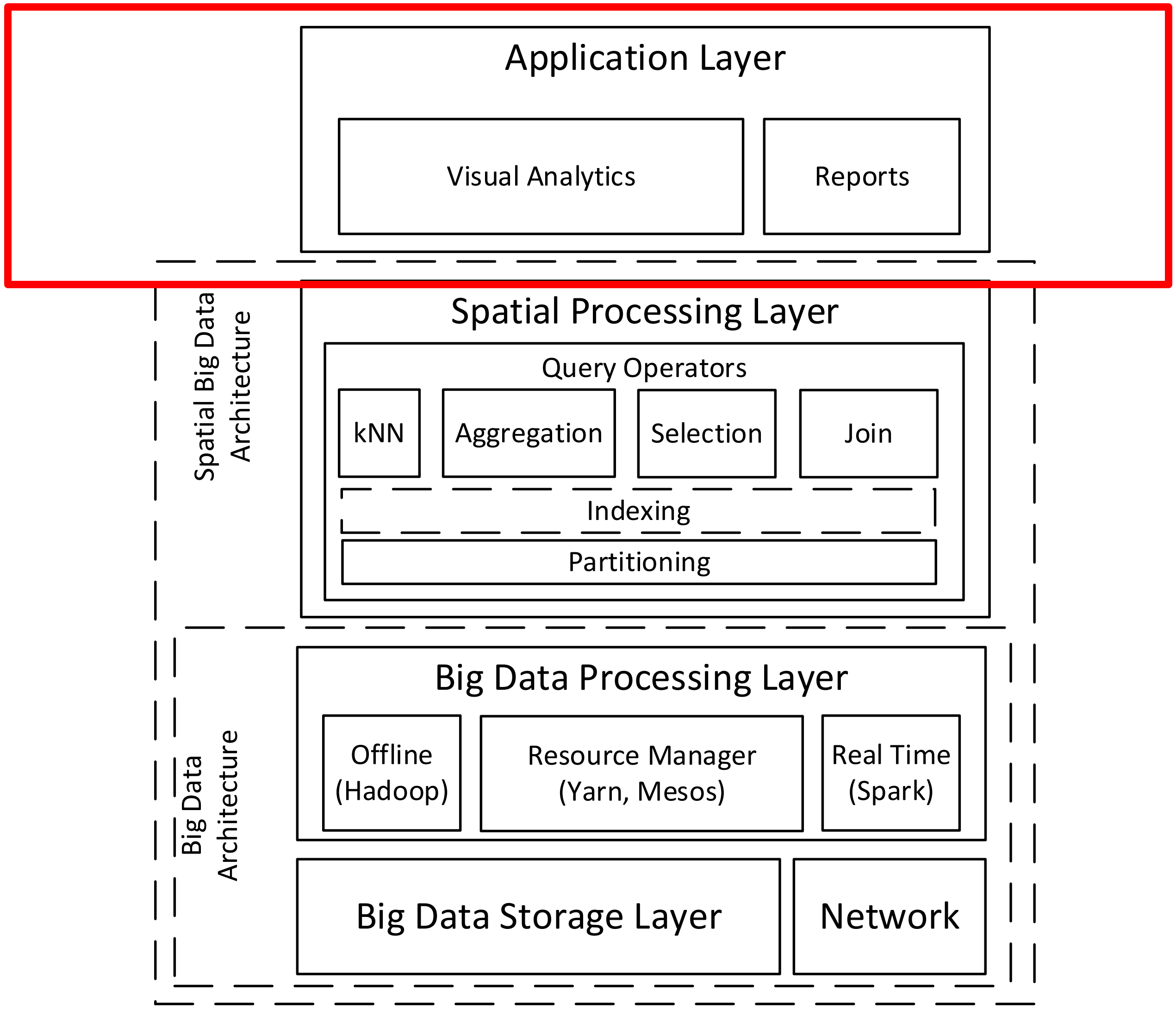


A. Iyer, L. Li, I. Stoica. Celliq: Real-time cellular network analytics at scale. NSDI'15

Presentation Outline

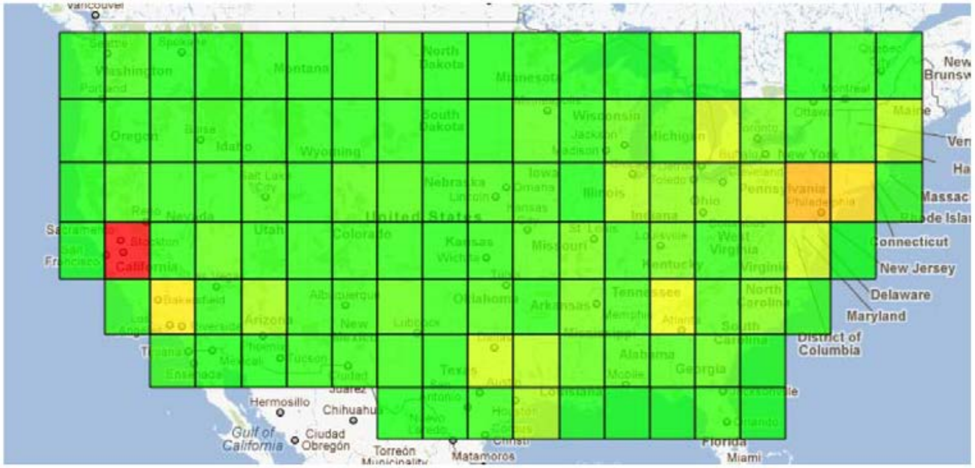
- Introduction
- Big Data Architectures
- Spatial Big Data Processing
- **Spatial Big Data Visualization**
- Conclusions
- Future work

Architecture Outline



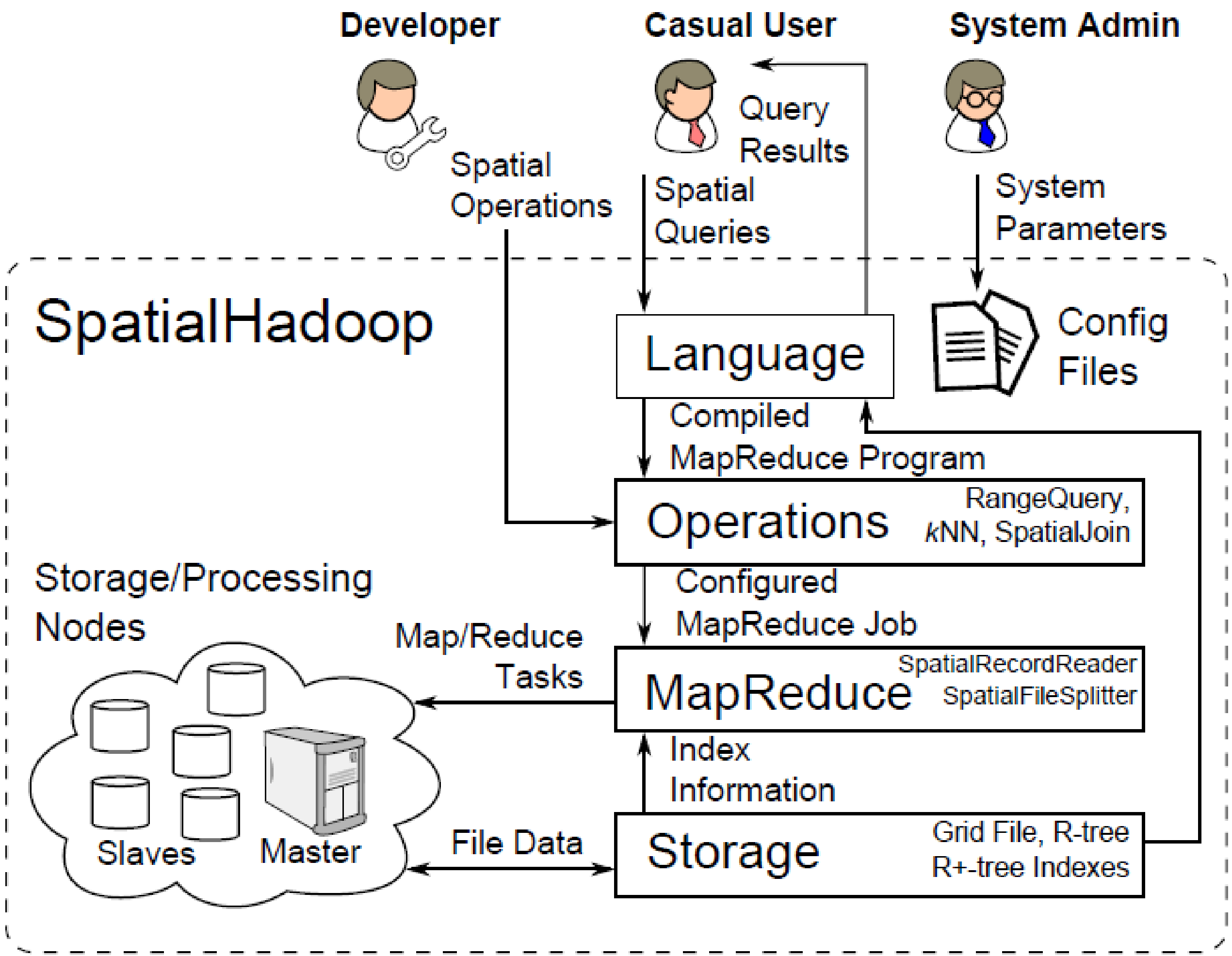
Asterix (previous section)

A visualization of the results of a spatial aggregation query. The color of each cell indicates the tweet count.



Tweets mentioning “Santorum” posted from February 15, 2012 to March 1, 2012, group them on a grid structure, and compute the number of such tweets in each cell in the grid (almost 5,672 tweets)

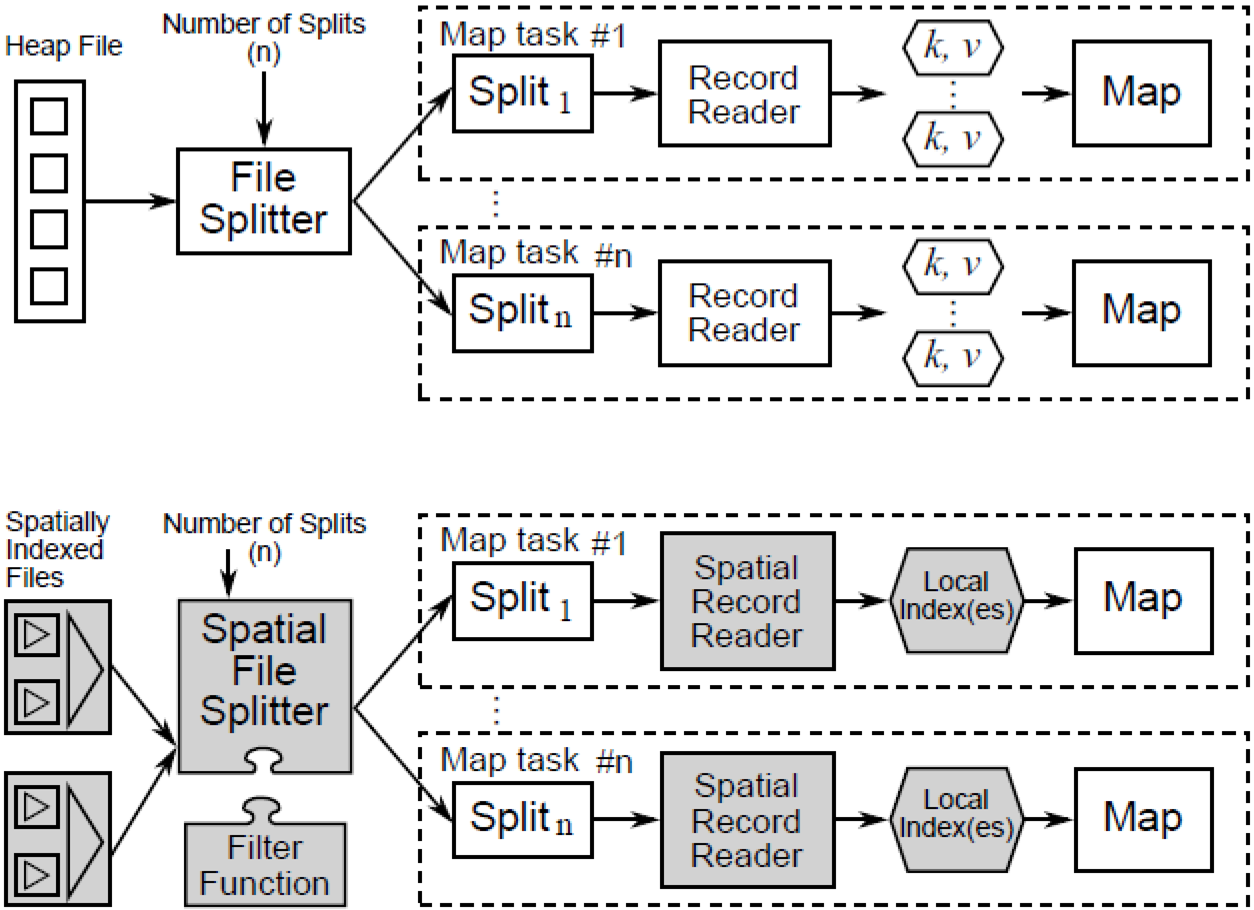
SpatialHadoop



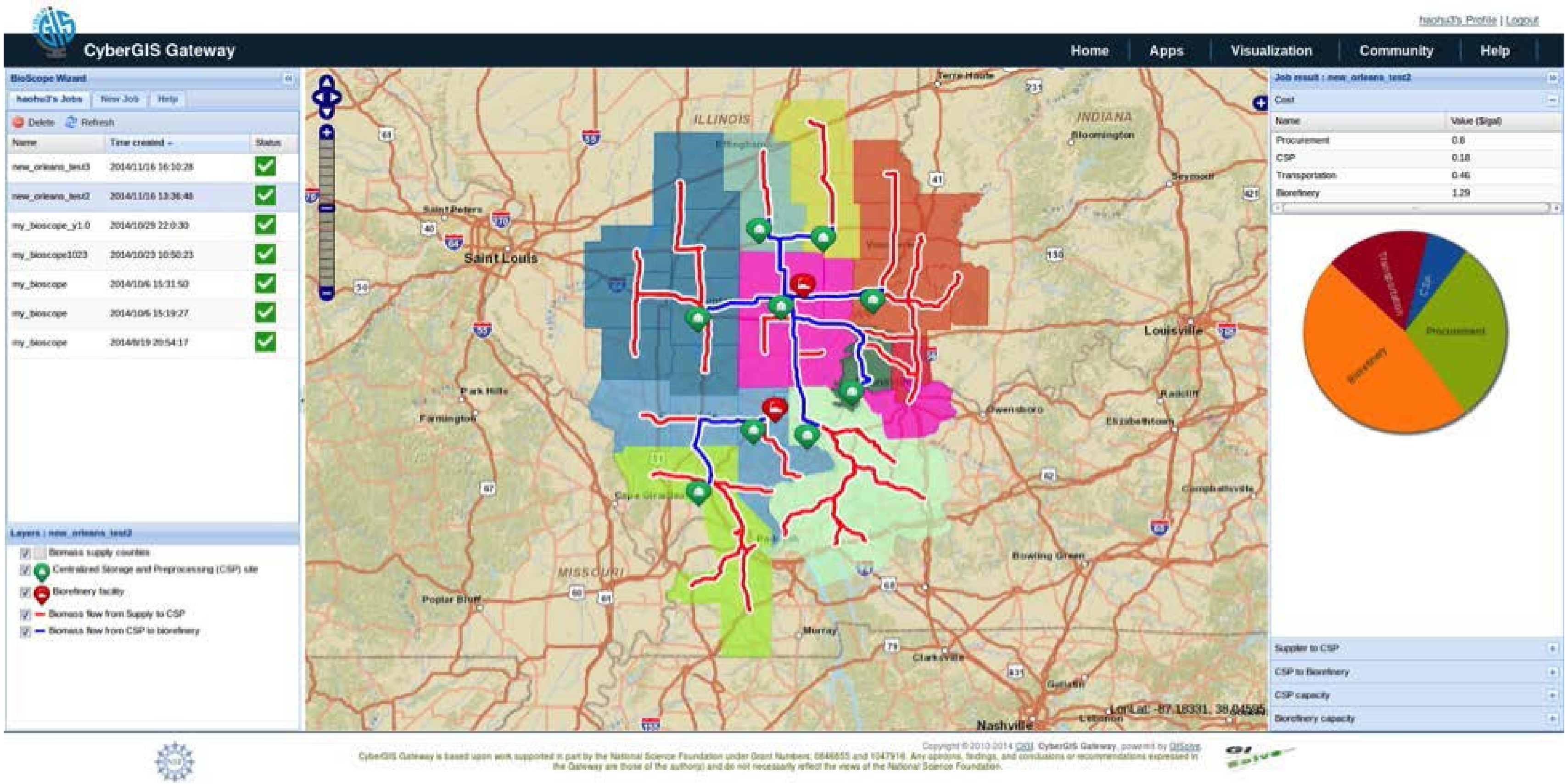
A. Eldawy and M. Mokbel. SpatialHadoop: A MapReduce Framework for Spatial Data. *ICDE 2015*

SpatialHadoop

Map phase in Hadoop (top) and SpatialHadoop (bottom)

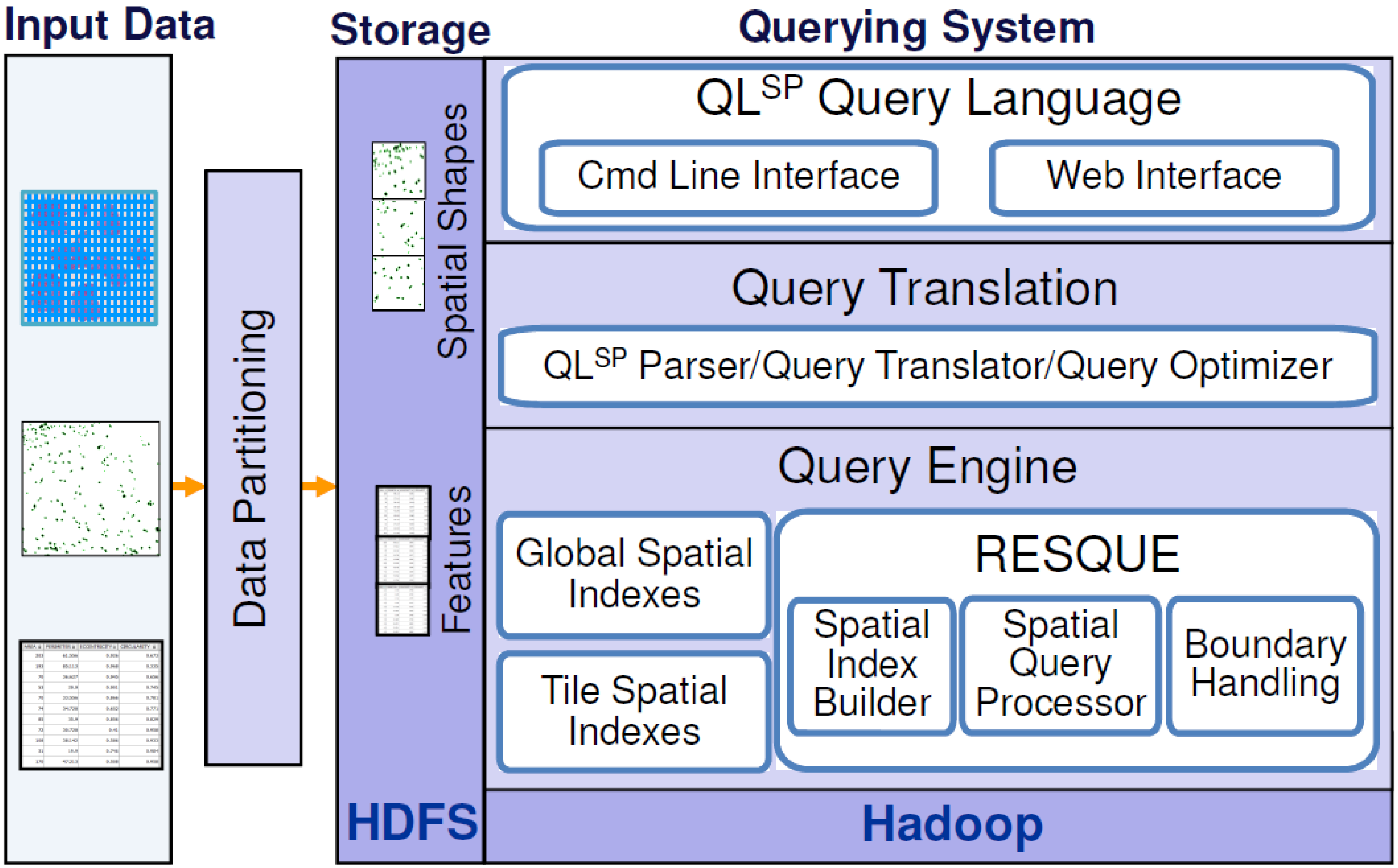


CyberGIS



S. Wang, H. Hu, T. Lin, Y. Liu, A. Padmanabhan, K. Soltani. Cybergis for data-intensive knowledge discovery. *SIGSPATIAL Special*, 2015.

Hadoop-GIS



A. Aji, F. Wang, H. Vo, R. Lee, Q. Liu, X. Zhang, J. Saltz. Hadoop gis: A high performance spatial data warehousing system over mapreduce. *VLDB* 2013

Presentation Outline

- **Introduction**
- **Big Data Architectures**
- **Spatial Big Data Processing**
- **Spatial Big Data Visualization**
- **Conclusions**
- Future work

Conclusions

Systems	Big Data	Spatial	Real-time	Visualization
MISO[37]	YES	NO	NO	NO
epiC[32]	YES	NO	NO	NO
Stratosphere[24]	YES	NO	NO	NO
Spark SQL[8]	YES	NO	YES*	NO
MapReduce Online[16]	YES	NO	YES	NO
OceanRT[67]	YES	NO	YES	NO
CEP[54]	YES	NO	YES	NO
AIM[12]	YES	NO	YES	NO
ASTERIX[4]	YES	YES*	NO	YES
AQWA[5]	YES	YES	NO	NO
CellIQ[30]	YES	YES	YES	NO
SpatialHadoop[23]	YES	YES	NO	YES
CyberGIS[59]	YES	YES	NO	YES
Hadoop-GIS[3]	YES	YES	NO	YES

Conclusions

- We have presented several recent spatial and Big Data systems with innovative techniques.
- In the previous table we compare the systems based on their ability to process:
  - Big Data,
  - answer spatial queries,
  - respond in real -time and
  - provide visualization methods.
- These parameters were chosen in order to show the historical evolution of the systems.

Open Problems

- *Location Prediction*
- *Real-time Spatial Big Data*
- *Routing query*
- *Quality of Service*
- *Privacy*

SPATIAL BIG DATA QUERY PROCESSING

Thanks! Questions?

Constantinos Costa

PhD Canditate

Data Management Systems Laboratory

Department of Computer Science

University of Cyprus

http://www.cs.ucy.ac.cy/~costa.c/

EPL670, March 31, 2015,
University of Cyprus, Nicosia, Cyprus