Efficient Storage of Big-Data for Real-time GPS Applications

Pavan Kumar Akulakrishna

Dr.J.Lakshmi & Prof.S.K.Nandy,
CAD Lab, SERC,
Indian Institute of Scie

November 15, 2014

Outline

- Introduction
- Motivation
- Related Work
- Methodology
- Questions
- Experiment and Results
- Conclusions & Future Work

Introduction

- GPS Applications
  - Finding current location
  - Finding a route from current location to some destination
  - Finding nearest police station or hospital or restaurants etc.,
- GPS Applications need to be
  - More responsive
  - Provide realtime information
- In context of storage these applications require
  - Efficient data layout
  - Efficient management and distribution data

Introduction

- GPS Applications
  - Finding current location
  - Finding a route from current location to some destination
  - Finding nearest police station or hospital or restaurants etc.,
- GPS Applications need to be
  - More responsive
  - Provide realtime information
- In context of storage these applications require
  - Efficient data layout
  - Efficient management and distribution data

Introduction

- GPS Applications
  - Finding current location
  - Finding a route from current location to some destination
  - Finding nearest police station or hospital or restaurants etc.,
- GPS Applications need to be
  - More responsive
  - Provide realtime information
- In context of storage these applications require
  - Efficient data layout
  - Efficient management and distribution data

Introduction

- GPS Applications
  - Finding current location
  - Finding a route from current location to some destination
  - Finding nearest police station or hospital or restaurants etc.,
- GPS Applications need to be
  - More responsive
  - Provide realtime information
- In context of storage these applications require
  - Efficient data layout
  - Efficient management and distribution data

Introduction

- GPS Applications
  - Finding current location
  - Finding a route from current location to some destination
  - Finding nearest police station or hospital or restaurants etc.,
- GPS Applications need to be
  - More responsive
  - Provide realtime information
- In context of storage these applications require
  - Efficient data layout
  - Efficient management and distribution data

Introduction

- GPS Applications
  - Finding current location
  - Finding a route from current location to some destination
  - Finding nearest police station or hospital or restaurants etc.,
- GPS Applications need to be
  - More responsive
  - Provide realtime information
- In context of storage these applications require
  - Efficient data layout
  - Efficient management and distribution data

Introduction

- GPS Applications
  - Finding current location
  - Finding a route from current location to some destination
  - Finding nearest police station or hospital or restaurants etc.,
- GPS Applications need to be
  - More responsive
  - Provide realtime information
- In context of storage these applications require
  - Efficient data layout
  - Efficient management and distribution data

Introduction

- GPS Applications
  - Finding current location
  - Finding a route from current location to some destination
  - Finding nearest police station or hospital or restaurants etc.,
- GPS Applications need to be
  - More responsive
  - Provide realtime information
- In context of storage these applications require
  - Efficient data layout
  - Efficient management and distribution data

Introduction

- GPS Applications
  - Finding current location
  - Finding a route from current location to some destination
  - Finding nearest police station or hospital or restaurants etc.,
- GPS Applications need to be
  - More responsive
  - Provide realtime information
- In context of storage these applications require
  - Efficient data layout
  - Efficient management and distribution data

Introduction

- GPS Applications
  - Finding current location
  - Finding a route from current location to some destination
  - Finding nearest police station or hospital or restaurants etc.,
- GPS Applications need to be
  - More responsive
  - Provide realtime information
- In context of storage these applications require
  - Efficient data layout
  - Efficient management and distribution data

Motivation

- One study estimates we could save more than 600 billion dollars annually by 2020 [8].
- A consumer benefits by saving
  - Time
  - Fuel
- "New ways to Exploit Raw Data may bring surge of innovation", a study says [8].

Motivation

- One study estimates we could save more than 600 billion dollars annually by 2020 [8].
- A consumer benefits by saving
  - Time
  - Fuel
- "New ways to Exploit Raw Data may bring surge of innovation", a study says [8].

Motivation

- One study estimates we could save more than 600 billion dollars annually by 2020 [8].
- A consumer benefits by saving
  - Time
  - Fuel
- "New ways to Exploit Raw Data may bring surge of innovation", a study says [8].

Motivation

- One study estimates we could save more than 600 billion dollars annually by 2020 [8].
- A consumer benefits by saving
  - Time
  - Fuel
- "New ways to Exploit Raw Data may bring surge of innovation", a study says [8].

Motivation

- One study estimates we could save more than 600 billion dollars annually by 2020 [8].
- A consumer benefits by saving
  - Time
  - Fuel
- "New ways to Exploit Raw Data may bring surge of innovation", a study says [8].

Related Work

- Data storage and dissemination is done in either a fully distributed manner or in a centralized manner [2].
- Fully Distributed Manner
  - Geodatabase replication
  - DBMS replication
  - Data copying and loading tools
- Centralized Manner
  - Storage at centralized servers
    -Stored in traditional DBMS.
  - No time dependant data is stored

Related Work

- Data storage and dissemination is done in either a fully distributed manner or in a centralized manner [2].
- Fully Distributed Manner
  - Geodatabase replication
  - DBMS replication
  - Data copying and loading tools
- Centralized Manner
  - Storage at centralized servers
    -Stored in traditional DBMS.
  - No time dependant data is stored

Related Work

- Data storage and dissemination is done in either a fully distributed manner or in a centralized manner [2].
- Fully Distributed Manner
  - Geodatabase replication
  - DBMS replication
  - Data copying and loading tools
- Centralized Manner
  - Storage at centralized servers
    -Stored in traditional DBMS.
  - No time dependant data is stored

Related Work

- Data storage and dissemination is done in either a fully distributed manner or in a centralized manner [2].
- Fully Distributed Manner
  - Geodatabase replication
  - DBMS replication
  - Data copying and loading tools
- Centralized Manner
  - Storage at centralized servers
    -Stored in traditional DBMS.
  - No time dependant data is stored

Related Work

- Data storage and dissemination is done in either a fully distributed manner or in a centralized manner [2].
- Fully Distributed Manner
  - Geodatabase replication
  - DBMS replication
  - Data copying and loading tools
- Centralized Manner
  - Storage at centralized servers
    -Stored in traditional DBMS.
  - No time dependant data is stored

Related Work

- Data storage and dissemination is done in either a fully distributed manner or in a centralized manner [2].
- Fully Distributed Manner
  - Geodatabase replication
  - DBMS replication
  - Data copying and loading tools
- Centralized Manner
  - Storage at centralized servers
    -Stored in traditional DBMS.
  - No time dependant data is stored

Related Work

- Data storage and dissemination is done in either a fully distributed manner or in a centralized manner [2].
- Fully Distributed Manner
  - Geodatabase replication
  - DBMS replication
  - Data copying and loading tools
- Centralized Manner
  - Storage at centralized servers
    -Stored in traditional DBMS.
  - No time dependant data is stored

Related Work

- Data storage and dissemination is done in either a fully distributed manner or in a centralized manner [2].
- Fully Distributed Manner
  - Geodatabase replication
  - DBMS replication
  - Data copying and loading tools
- Centralized Manner
  - Storage at centralized servers
    -Stored in traditional DBMS.
  - No time dependant data is stored

Major issues in road-network related GPS applications

- GPS applications' data is dynamic and large.
- GPS applications need real-time responsiveness.
- User queries are location-sensitive and time-variant.
- Network congestions due to communications involved.
- Use of traditional DBMS don't scale well.

Major issues in road-network related GPS applications

- GPS applications' data is dynamic and large.
- GPS applications need real-time responsiveness.
- User queries are location-sensitive and time-variant.
- Network congestions due to communications involved.
- Use of traditional DBMS don't scale well.

Major issues in road-network related GPS applications

- GPS applications' data is dynamic and large.
- GPS applications need real-time responsiveness.
- User queries are location-sensitive and time-variant.
- Network congestions due to communications involved.
- Use of traditional DBMS don't scale well.

Major issues in road-network related GPS applications

- GPS applications' data is dynamic and large.
- GPS applications need real-time responsiveness.
- User queries are location-sensitive and time-variant.
- Network congestions due to communications involved.
- Use of traditional DBMS don't scale well.

Major issues in road-network related GPS applications

- GPS applications' data is dynamic and large.
- GPS applications need real-time responsiveness.
- User queries are location-sensitive and time-variant.
- Network congestions due to communications involved.
- Use of traditional DBMS don't scale well.

Methodology: Approach

- Computation-closeness to the data is ensured to reduce
  - Communication
  - Latency.
- Data-redundancy is maintained to improve
  - Performance
  - Build fault-tolerant solution.

Methodology: Approach

- Computation-closeness to the data is ensured to reduce
  - Communication
  - Latency.
- Data-redundancy is maintained to improve
  - Performance
  - Build fault-tolerant solution.

Methodology: Approach

- Computation-closeness to the data is ensured to reduce
  - Communication
  - Latency.
- Data-redundancy is maintained to improve
  - Performance
  - Build fault-tolerant solution.

Methodology: Approach

- Computation-closeness to the data is ensured to reduce
  - Communication
  - Latency.
- Data-redundancy is maintained to improve
  - Performance
  - Build fault-tolerant solution.

Methodology: Approach

- Computation-closeness to the data is ensured to reduce
  - Communication
  - Latency.
- Data-redundancy is maintained to improve
  - Performance
  - Build fault-tolerant solution.

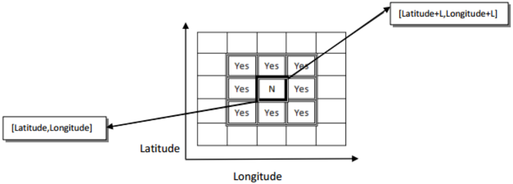
Methodology: Approach

- Computation-closeness to the data is ensured to reduce
  - Communication
  - Latency.
- Data-redundancy is maintained to improve
  - Performance
  - Build fault-tolerant solution.

Methodology: *NineCellGrid*

- In this method, data is distributed not based on the available storage nodes, but based on the region of area on which the computation is intended.

MATHIKERE
IISc Guest House
Indian Institute Of Science
CV Raman Rd
Porcelain Factory
Maharani Lakshmi Ammanni College For Women
HKES College
New BEL Rd
Silver Oak Marg

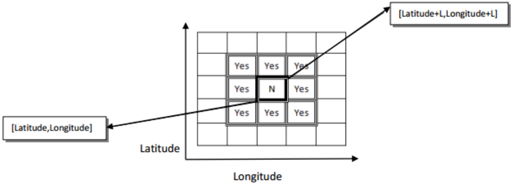


- Entire region is decomposed into L×L cells (L explained later)
- Each cell's data copied to adjacent 8 cells
- A 'NameTable' to store mapping $(Latitude, Longitude) \rightarrow NodeAddress$

Methodology: *NineCellGrid*

- In this method, data is distributed not based on the available storage nodes, but based on the region of area on which the computation is intended.

IISc Guest House
MATHIKERE
Indian Institute Of Science
Silver Oak Marg
New BEL Rd
CV Raman Rd
HKES College
Porcelain Factory
Maharani Lakshmi Ammanni College For Women

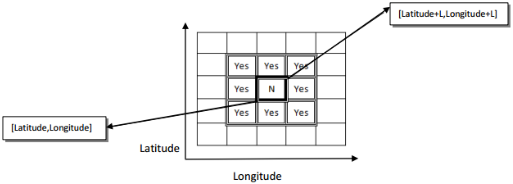


- Entire region is decomposed into L×L cells (L explained later)
- Each cell's data copied to adjacent 8 cells
- A 'NameTable' to store mapping $(Latitude, Longitude) \rightarrow NodeAddress$

Methodology: *NineCellGrid*

- In this method, data is distributed not based on the available storage nodes, but based on the region of area on which the computation is intended.

MATHIKERE
IISc Guest House
Indian Institute Of Science
CV Raman Rd
Porcelain Factory
Maharani Lakshmi Ammanni College For Women
New BEL Rd

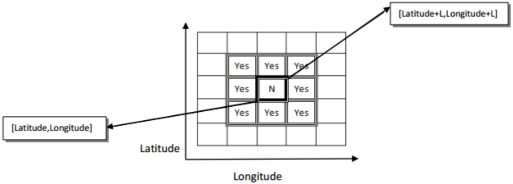


- Entire region is decomposed into L×L cells (L explained later)
- Each cell's data copied to adjacent 8 cells
- A 'NameTable' to store mapping $(Latitude, Longitude) \rightarrow NodeAddress$

Methodology: *NineCellGrid*

- In this method, data is distributed not based on the available storage nodes, but based on the region of area on which the computation is intended.

SUBEDARPALYA
MATHIKERE
IISc Guest House
Indian Institute Of Science
Silver Oak Marg
CV Raman Rd
New BEL Rd
Porcelain Factory
Maharani Lakshmi Ammanni College For Women
HKES College



- Entire region is decomposed into L×L cells (L explained later)
- Each cell's data copied to adjacent 8 cells
- A 'NameTable' to store mapping $(Latitude, Longitude) \rightarrow NodeAddress$

Methodology: *NineCellGrid*

- NameTable

		Longitude				
		-74.49	-74.49 + f(L)	-74.49 + 2*f(L)	...	-73.50
Latitude	40.30	<*NodeAddress*>	...	..	...	
	40.30 + g(L)	<*NodeAddress*>	...	..	...	
	40.30 + 2*g(L)	<*NodeAddress*>	...	..	...	
	...	<*NodeAddress*>	...	..	...	
	41.29	<*NodeAddress*>	...	..	...	

NameTable Here f and g are mappings from miles to Longitude and Latitude units

NineCellGrid : Dimension of cell, L

- A query is of the form 'route A-to-B'
- L is a value in miles, such that the query has A-B Euclidean distance less than or equal to L with high probability
- The probability or percentage is found computationally in steps of 3-5% using the historic query data from NHTS [13].
- Higher the L the more it will lead to centralized pattern
- Lower the L the more is the communication
- Optimum L exists (shown in Results section)

NineCellGrid : Dimension of cell, L

- A query is of the form 'route A-to-B'
- L is a value in miles, such that the query has A-B Euclidean distance less than or equal to L with high probability
- The probability or percentage is found computationally in steps of 3-5% using the historic query data from NHTS [13].
- Higher the L the more it will lead to centralized pattern
- Lower the L the more is the communication
- Optimum L exists (shown in Results section)

NineCellGrid : Dimension of cell, L

- A query is of the form 'route A-to-B'
- L is a value in miles, such that the query has A-B Euclidean distance less than or equal to L with high probability
- The probability or percentage is found computationally in steps of 3-5% using the historic query data from NHTS [13].
- Higher the L the more it will lead to centralized pattern
- Lower the L the more is the communication
- Optimum L exists (shown in Results section)

NineCellGrid : Dimension of cell, L

- A query is of the form 'route A-to-B'
- L is a value in miles, such that the query has A-B Euclidean distance less than or equal to L with high probability
- The probability or percentage is found computationally in steps of 3-5% using the historic query data from NHTS [13].
- Higher the L the more it will lead to centralized pattern
- Lower the L the more is the communication
- Optimum L exists (shown in Results section)

NineCellGrid : Dimension of cell, L

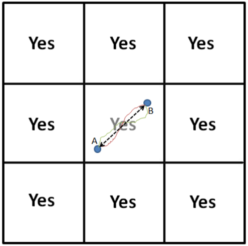
- A query is of the form 'route A-to-B'
- L is a value in miles, such that the query has A-B Euclidean distance less than or equal to L with high probability
- The probability or percentage is found computationally in steps of 3-5% using the historic query data from NHTS [13].
- Higher the L the more it will lead to centralized pattern
- Lower the L the more is the communication
- Optimum L exists (shown in Results section)

NineCellGrid : Dimension of cell, L

- A query is of the form 'route A-to-B'
- L is a value in miles, such that the query has A-B Euclidean distance less than or equal to L with high probability
- The probability or percentage is found computationally in steps of 3-5% using the historic query data from NHTS [13].
- Higher the L the more it will lead to centralized pattern
- Lower the L the more is the communication
- Optimum L exists (shown in Results section)

Case Study

- Case 1: Both A and B lie in the same cell of L×L

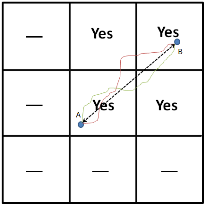


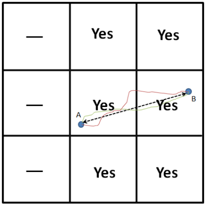
Yes	Yes	Yes
Yes	Yes	Yes
Yes	Yes	Yes

- Any of 9 cells can process the query
- Priority is given based on Euclidean from center to A and B and task load at that node

Case Study

- Case 2: A and B lie at 1 cell distance apart

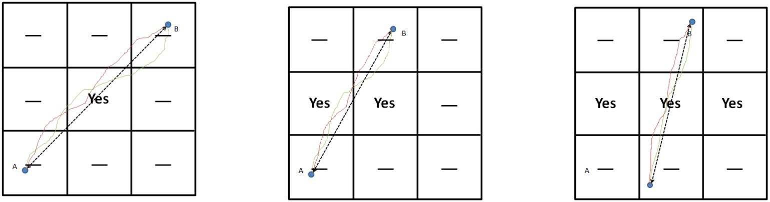




- Number of nodes that can process the query are 4-6 nodes.

Case Study

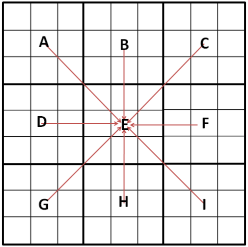
- Case 3: A and B lie at 2 Cell distance apart



- Number of nodes that can process the query are 1-3 nodes.

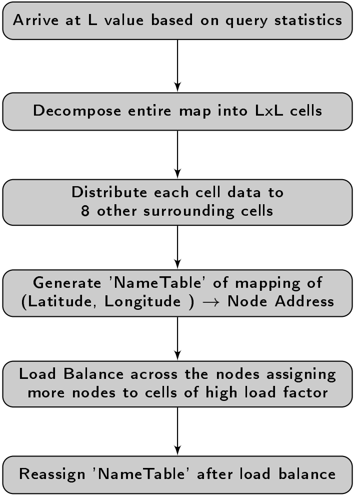
Case Study

- Case 4: A and B lie at more than 2 cell distance apart



- 1/9th of the nodes communicate which is 11.11% of total nodes involving in communication.

NineCellGrid Summary



Arrive at L value based on query statistics

Decompose entire map into LxL cells

Distribute each cell data to 8 other surrounding cells

Generate 'NameTable' of mapping of (Latitude, Longitude) → Node Address

Load Balance across the nodes assigning more nodes to cells of high load factor

Reassign 'NameTable' after load balance

Load Balancing

Algorithm: *Load Balancing*

Inputs: [*NameTable*; Relaxation δ]

Outputs: [Load balanced distribution; Updated *NameTable*]

```
Initialize load based on number of edges, vertices, updates, etc.,
for each node $E$ do:
    if( Load(E) $< M - \delta$ )
        Find node P closest to E with load $\geq M - \delta$.
        Load(P) $\rightarrow$ Load(P) + Load(E);
        Load(E) $\rightarrow$ 0;
        NameTable[Latitude(E)][Longitude(E)] = $Address(P)$.
    endif
endfor
for each node $E$ do:
    if( Load(E) $\neq$ 0 )
        Share the load among N nodes.
        $(Load(E)/N) \in [M - \delta, M + \delta]$ & N $\in$ (1,2,3,...)
        Update NameTable.
    endif
endfor
```

Why nine cell ?

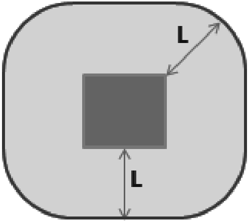
- Consider,$< A \rightarrow B >$ route query with Euclidean distance to be L or less than L (Our claim)
- All possible locations of A in a cell would be

Why nine cell ?

- Consider, $< A \to B >$ route query with Euclidean distance to be L or less than L (Our claim)
- All possible locations of A in a cell would be

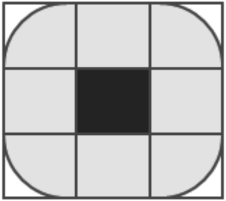
Why nine cell ?

- Locus of B would be



Why nine cell ?

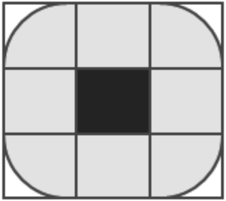
- Best fit is..



- Hence, "NineCellGrid"

Why nine cell ? 💬

- Best fit is..



- Hence, "NineCellGrid"

Is 9 way redundancy not too much ?

- Generally, redundancy is used to maintain fault-tolerant systems and also in cases where the data is highly important.
- State of art Replication Factor(RF) is 3, 5 or may be some cases upto 7.
- Here, in this case we are using redundancy mostly to ensure performance than to ensure fault-tolerant and data consistency.
- And moreover, the data of a cell can be stored distributedly across the 9 nodes in the HDFS(Hadoop Distributed FileSystem) Format[4].
- It is also noted that specific type of data is stored in HDFS format which support reduce operations; else they follow the mechanism of complete data copies i.e., entire file is stored as copies in all the 9 cells.

Is 9 way redundancy not too much ?

- Generally, redundancy is used to maintain fault-tolerant systems and also in cases where the data is highly important.
- State of art Replication Factor(RF) is 3, 5 or may be some cases upto 7.
- Here, in this case we are using redundancy mostly to ensure performance than to ensure fault-tolerant and data consistency.
- And moreover, the data of a cell can be stored distributedly across the 9 nodes in the HDFS(Hadoop Distributed FileSystem) Format[4].
- It is also noted that specific type of data is stored in HDFS format which support reduce operations; else they follow the mechanism of complete data copies i.e., entire file is stored as copies in all the 9 cells.

Is 9 way redundancy not too much ?

- Generally, redundancy is used to maintain fault-tolerant systems and also in cases where the data is highly important.
- State of art Replication Factor(RF) is 3, 5 or may be some cases upto 7.
- Here, in this case we are using redundancy mostly to ensure performance than to ensure fault-tolerant and data consistency.
- And moreover, the data of a cell can be stored distributedly across the 9 nodes in the HDFS(Hadoop Distributed FileSystem) Format[4].
- It is also noted that specific type of data is stored in HDFS format which support reduce operations; else they follow the mechanism of complete data copies i.e., entire file is stored as copies in all the 9 cells.

Is 9 way redundancy not too much ?

- Generally, redundancy is used to maintain fault-tolerant systems and also in cases where the data is highly important.
- State of art Replication Factor(RF) is 3, 5 or may be some cases upto 7.
- Here, in this case we are using redundancy mostly to ensure performance than to ensure fault-tolerant and data consistency.
- And moreover, the data of a cell can be stored distributedly across the 9 nodes in the HDFS(Hadoop Distributed FileSystem) Format[4].
- It is also noted that specific type of data is stored in HDFS format which support reduce operations; else they follow the mechanism of complete data copies i.e., entire file is stored as copies in all the 9 cells.

Is 9 way redundancy not too much ?

- Generally, redundancy is used to maintain fault-tolerant systems and also in cases where the data is highly important.
- State of art Replication Factor(RF) is 3, 5 or may be some cases upto 7.
- Here, in this case we are using redundancy mostly to ensure performance than to ensure fault-tolerant and data consistency.
- And moreover, the data of a cell can be stored distributedly across the 9 nodes in the HDFS(Hadoop Distributed FileSystem) Format[4].
- It is also noted that specific type of data is stored in HDFS format which support reduce operations; else they follow the mechanism of complete data copies i.e., entire file is stored as copies in all the 9 cells.

Experiment and Results

- We used MPI for simulation and Dijkstra for route computation to study the application-level performance of centralized, fully distributed and NineCellGrid approach data dissemination models.
- Notations used

CGWR	:	"Cell Grid Without Replication"
DBMSAR	:	DBMS as Replication factor, RF = 5
Zonal	:	Fully distributed with RF = 5
Centralized	:	Centralized distribution.

- Dataset

Dataset used	:	New York City [(40.3, 41.3),(73.5,74.5)]
Vertices/Junctions	:	264,346
Arcs/Links	:	733,846
Source	:	DIMACS [14]

Experiment and Results

- We used MPI for simulation and Dijkstra for route computation to study the application-level performance of centralized, fully distributed and NineCellGrid approach data dissemination models.
- Notations used

CGWR	:	"Cell Grid Without Replication"
DBMSAR	:	DBMS as Replication factor, RF = 5
Zonal	:	Fully distributed with RF = 5
Centralized	:	Centralized distribution.

- Dataset

Dataset used	:	New York City [(40.3, 41.3),(73.5,74.5)]
Vertices/Junctions	:	264,346
Arcs/Links	:	733,846
Source	:	DIMACS [14]

Experiment and Results

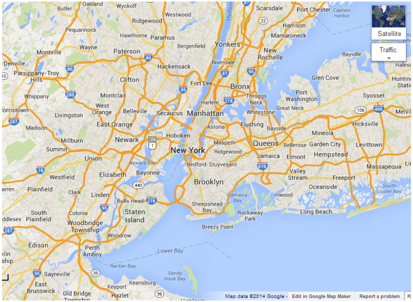
- We used MPI for simulation and Dijkstra for route computation to study the application-level performance of centralized, fully distributed and NineCellGrid approach data dissemination models.
- Notations used

CGWR	:	"Cell Grid Without Replication"
DBMSAR	:	DBMS as Replication factor, RF = 5
Zonal	:	Fully distributed with RF = 5
Centralized	:	Centralized distribution.

- Dataset

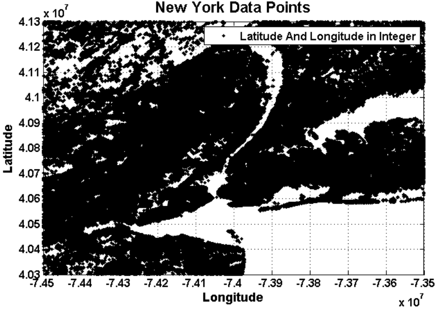
Dataset used	:	New York City [(40.3, 41.3),(73.5,74.5)]
Vertices/Junctions	:	264,346
Arcs/Links	:	733,846
Source	:	DIMACS [14]

Actual Map of the Dataset used



New York [(40.3, 41.3),(73.5,74.5)]
Source: maps.google.com [1]

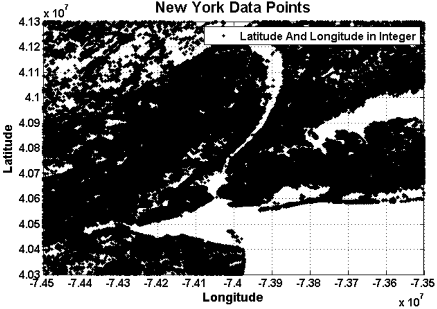
Data-points in actual coordinates



Plot represents junction points
Dataset Source: Dimacs[14]

- L value for this dataset is considered 6.8 miles[13]
- Each data-point contributes to 1 unit of 'Load Factor'

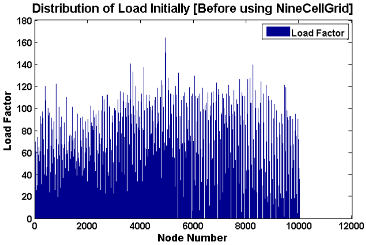
Data-points in actual coordinates

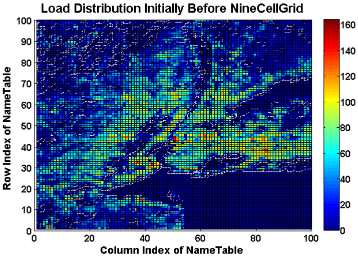


Plot represents junction points
Dataset Source: Dimacs[14]

- L value for this dataset is considered 6.8 miles[13]
- Each data-point contributes to 1 unit of 'Load Factor'

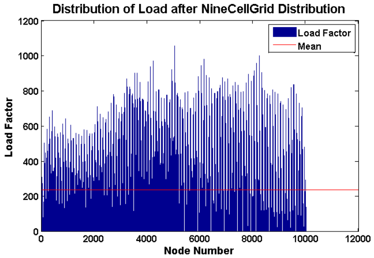
After assigning each cell Load Factor

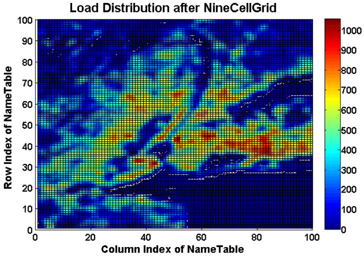




Load Factor is the number of Junctions and Arcs present in that region

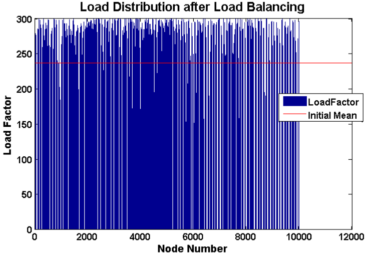
NineCellGrid Distribution is applied

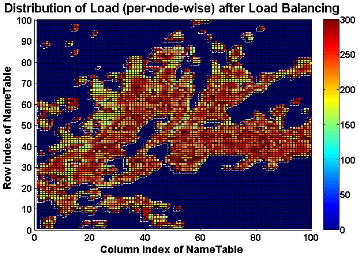




Load Factor of each cell is distributed to all other 8 cells

After Load Balancing

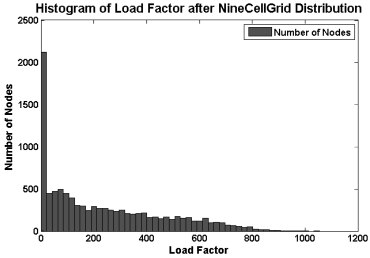


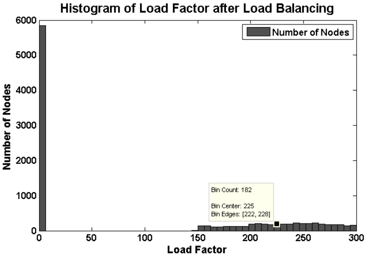


Mean M = 225, Relaxation $\delta = 75$

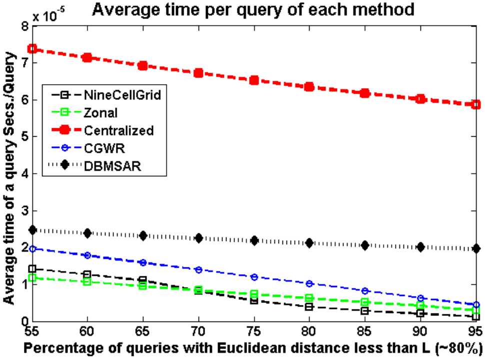
Hence the Min = 150 and Max = 300

Histogram comparison



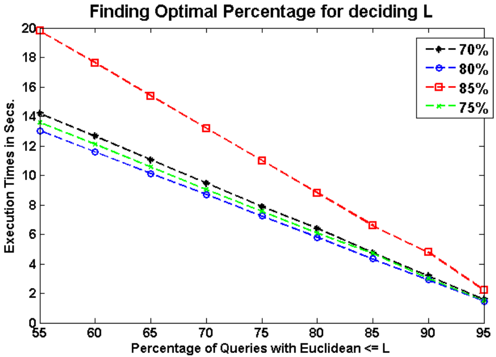


Average Turnaround Times



Number of Queries : 1000

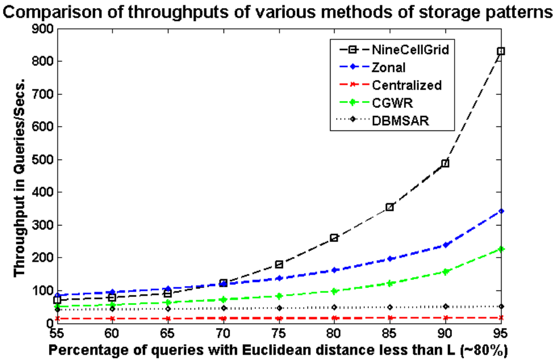
Finding Optimal Percentage for L



Number of Queries : 1000

- Higher the L the more it tends to centralized pattern
- Lower the L the more is the communication

Throughput Comparison



Number of Queries : 1000

Conclusions

- Results show that NineCellGrid storage method achieve better throughput than fully distributed and centralized storage methods.
- Despite redundancy being high and usage of extra space is a burden we are gaining a performance speed up.
- Using this method we gain redundancy in data, fault-tolerance, highly data parallel so increased level of parallelism.
- The overall performance of our method is either higher than or comparable with fully distributed and centralized methods.

Conclusions

- Results show that NineCellGrid storage method achieve better throughput than fully distributed and centralized storage methods.
- Despite redundancy being high and usage of extra space burden we are gaining a performance speed up.
- Using this method we gain redundancy in data, fault-tolerance, highly data parallel so increased level of parallelism.
- The overall performance of our method is either higher than or comparable with fully distributed and centralized methods.

Conclusions

- Results show that NineCellGrid storage method achieve better throughput than fully distributed and centralized storage methods.
- Despite redundancy being high and usage of extra space is a burden we are gaining a performance speed up.
- Using this method we gain redundancy in data, fault-tolerance, highly data parallel so increased level of parallelism.
- The overall performance of our method is either higher than or comparable with fully distributed and centralized methods.

Conclusions

- Results show that NineCellGrid storage method achieve better throughput than fully distributed and centralized storage methods.
- Despite redundancy being high and usage of extra space is a burden we are gaining a performance speed up.
- Using this method we gain redundancy in data, fault-tolerance, highly data parallel so increased level of parallelism.
- The overall performance of our method is either higher than or comparable with fully distributed and centralized methods.

Future Work

- Our future work will focus on balancing the load at each node by manipulating the replication factors.
- Job-scheduling algorithms that learn from previously observed queries can be devised to improve performance even more.

Future Work

- Our future work will focus on balancing the load at each node by manipulating the replication factors.
- Job-scheduling algorithms that learn from previously observed queries can be devised to improve performance even more.

References

- Google Maps http://maps.google.com
- Distributed or Centralized Traffic Advisory Systems - The Applications Take. Otto, J.S. ; Dept. of Electr. Eng. and Comput. Sci., Northwestern Univ., Evanston, IL, USA; Bustamante, F.E.
- B. Hoh, M. Gruteser, R. Herring, J. Ban, D. Work, J.C. Herrera, A. Bayen, M. Annavaram, and Q. Jacobson, "Virtual trip lines for distributed privacy preserving trafïňĄc monitoring," in Proc. of ACM/USENIX MobiSys, Breckenridge, CO, June 2008.
- Research on the Data Storage and Access Model in Distributed Computing Environment. Haiyan Wu Coll. of Comput. and Inf. Eng., Zhejiang GongShang Univ., Hangzhou
- Google White Papers
- H. Zhu, Y. Zhu, M. Li, and L. M. Ni, "HERO online real-time vehicle tracking in Shangai," in Proc. of IEEE INFOCOM, 2008.
- T. Logenthiran, Dipti Srinivasan Department of Electrical and Computer Engineering National University of Singapore, Intelligent Management of Distributed Storage Elements in a Smart Grid, 2011.
- Shashi Shekhar, Viswanath Gunturi, Michael R. Evans,KwangSoo Yang University of Minnesota, Spatial Big-Data Challenges Intersecting Mobility and Cloud Computing, 2012.
- V. Taliwal, D. Jiang, H. Mangold, C. Chen, and R. Sengupta, "Empirical determination of channel characteristics for DSRC vehicle-to-vehicle communication," in Proc. of ACM VANET, 2004.
- Z. Wang and M. Hassan, "How much of dsrc is available for non-safety use?" in Proc. of ACM VANET, September 2008.
- B. Hull, V. Bychkovsky, Y. Zhang, K. Chen, M. Goraczko, A. Miu, E. Shih, H. Balakrishnan, and S. Madden, "CarTel a distributed mobile sensor computing system", in Proc. of ACM SenSys, 2006.
- An ESRI Technical Paper June 2007.
- NHTS: National Household Travel Survey, 2009.
- DIMACS: http://www.dis.uniroma1.it/challenge9/download.shtml
- The Hadoop Distributed File System, Shvachko, K., Yahoo!, Sunnyvale, CA, USA, Hairong Kuang ; Radia, S. ; Chansler, R.
- Dijkstra Shortest Path Computation Algorithm, by Dijkstra.

Thank You!