Grid File index in Big Data Crowdsourcing

1st shahd mosa qari
Dept. of Computer Science
Umm Al-Qura University
Mecca, Saudi Arabia
shahdqari@gmail.com

2nd Manal Abdulrahman Alharthi
Dept. of Computer Science
Umm Al-Qura University
Mecca, Saudi Arabia
mnlharthi@gmail.com

3rd Leena Solayman Houmaidan
Dept. of Computer Science
Umm Al-Qura University
Mecca, Saudi Arabia
leenasolayman10@gmail.com

4thRaghad Mohammed Alqurashi
Dept. of Computer Science
Umm Al-Qura University
Mecca, Saudi Arabia
009.Raghad@gmail.com

5th Ashwaq Ahmed Almalki
Dept. of Computer Science
Umm Al-Qura University
Mecca, Saudi Arabia
ashwaqyou@gmail.com

6th Dr. Louai Alarabi
Dept. of Computer Science
Umm Al-Qura University
Mecca, Saudi Arabia
lmarabi@uqu.edu.sa

***Abstract*—The area of our research in computer science, particularly spatial computing. Spatial computing provides a very important computing technology for any geographically related applications and services. It helps in the construction of geographical websites' profiles and services. Spatial computing can be used in location identification such as google maps and have other uses and services in spatial planning, traffic congestion control, evacuation systems, crowd management, and spatial crowdsourcing. In this research, we have investigated the main component of spatial crowdsourcing. In particular, we introduced a spatial index (Grid File) to a crowdsourcing platform. Next, we constructed the basic spatial operation of crowdsourcing applications. We conducted an experimental study to recommend a suitable big data framework and suitable spatial indexing for crowdsourcing applications.**

we have investigated two main components of spatial crowdsourcing. In particular, we introduced a spatial index (Grid File) to a crowdsourcing platform. Next, we constructed the basic spatial operation of crowdsourcing applications.

***Index Terms*—Spatial, crowdsourcing, indexing, data.**

I. Introduction

With the recent developments on the geographical information system and spatial computing, people can take part in some of the nearby online tasks. This is well known as Spatial crowdsourcing [1]. Two casual users interact with a crowdsourcing system that can be classified into the requester and the workers. The requester is the one who request a specific task within a specified period in a specific location. The worker gains reliability when he/she accepts and completes the task correctly, then the worker can earn a reward from the requester of the task [2].

A detail of each modules of this taxonomy can be described as following:

1) Application: The requester sends a list of tasks that are linked to a specific time and reward and are executed by a crowdsourcing worker to earns the reward or for fun and wasting time. The crowdsourcing applications are grouped into three categories which are (i) Voting System: Voting tasks contain several options for which the crowdsourcing workers determines his answer. The answer that gets the most votes is the correct one. (ii) Information Sharing System: Internet users can share information more easily through websites. Crowdsourcing systems aim to share information among users. The information may also be exploited for other purposes, such as monitoring noise pollution. (iii) Games: There are games that take advantage of people's desire for entertainment to perform tasks and solve specific problems [2].
2) Algorithms: The algorithms in crowdsourcing formalize its design and predict the time a crowdsourcing worker might take to implement a specific task and model the performance in crowdsourcing systems [2].
3) Performance: is a very important aspect of crowdsourcing and is categorized into (i) user participation Which is to study and examine the behavior of crowdsourcing workers. (ii) quality management It is a study of how to get crowdsourcing workers to carry out the required task with good results and the extent to which their behavior affects the task outcomes. (iii) cheating detection It is for some crowdsourcing workers to cheat because their identity is not revealed, and this affects the proper performance of the task [2].
4) Dataset: which is a lot of research dataset in the field of crowdsourcing and it is now available for further research. For example, von Ahn et al. contributed a list of 100,000 images with English labels from their ESP Game11 [2].

The proliferation of mobiles and the speed of their development has extended conventional web-based crowdsourcing to spatial crowdsourcing (SC) where location plays a central role, in fact spatial crowdsourcing is spreading across "the world of speed". Whereas, it enables human workers to solve tasks for other humans in the physical world. So spatial crowdsourcing can be formulated as a goal based on assigning a set of spatial tasks to a group of workers, which means that the workers have to actually travel to those specific locations to perform the tasks assigned to it [3], [4]. For example, Amazon Mechanical Turk is one of the best known and most used of crowdsourcing:

It is a marketplace suitable for doing simple tasks and getting them done quickly while saving time and resources for the company [5].

An advantage of spatial crowdsourcing is that when a requester needs to perform a task in a long distant, they will only send the task to the server and the server will assign a worker to perform the task as required. it is no longer require travel to a long-distance to carry out his task, usually offering these rewards is a lot cheaper than formally hiring people to solve problems. Hiring a larger group of people can speed up the problem-solving process to the highest limit. In This project proposal we are going to investigate two main components of spatial crowdsourcing. In particular, we will introduce spatial indexes to a crowdsourcing platform. Next, we will construct two basic spatial operation of crowdsourcing applications. We will conduct an experimental study to recommend the suitable big data framework and the subtitle spatial indexing for crowdsourcing applications.

The current literature in SC can be classified into distinct categories based on the following criteria: the modes of publishing spatial tasks, and the different constraints being considered. In constraints has four categories is Temporal Constraints, Spatial Constraints, Quality Constraints and Budget Constraints [6], [7].

1) Temporal Constraints: The worker has a certain length of work or a deadline for the task [6].
2) Spatial Constraints: the worker performs task within region and the maximize travel distance [6].
3) Quality Constraints: guarantee that answers provided by workers are correct, or workers may reject the assigned tasks based on his/her lack of domain experts. So, the workers will get more confidence and in turn, guarantee high reliability of each spatial task, and assign at least one worker to do the task and give him a high confidence [7].
4) Budget constraints: the worker gets on reward when performing the task [6].

There are two of publishing spatial tasks on the spatial crowdsourcing server: Server Assigned Tasks (SAT) and Worker Selected Tasks (WST) [8], [9].

1) Server Assigned Tasks (SAT): A SC-server assigns an available worker to a spatial task based on its proximity to the location, availability for task performance, and worker reliability.
2) Worker Selected Tasks (WST): The SC-server publishes a set of tasks and the worker selects one of the available tasks and perform them.

The privacy protection is maintaining the privacy of the workers' site, the SC-server is responsible for preserving the privacy of the location, it is not trusted for its location [6].

Our project focus on supporting spatial indexing techniques on big data crowdsourcing applications [10].

A Spatial Index is a technology for organizing data in spatial data by organizing records into a space memory, so that it allows you to know the places where the data is stored and access it more quickly. To provide efficient query processing for spatial data, we need special data that supports cluster types.

In our project we focused on building a spatial indexing structures that has the capability to support a merit of crowdsourcing operations, such as:

1) Find a set of requests in any arbitrary location. (spatial Range query)

Hence, in our project framework we constructed a spatial library to support spatial indexing that accommodate and digest a variety of spatial crowdsourcing datasets, To clarity, shows Figure 5 the structure.

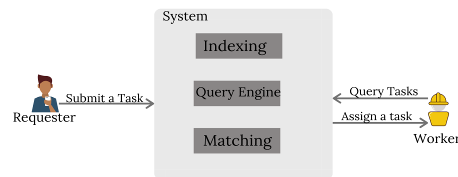


Fig. 1: A taxonomy of crowdsourcing.

We constructed a Grid file spatial indexing technique. By the end of our research we evaluated the performance of index and its scalability for spatial crowdsourcing applications and services.

Although these index and data structures are not new in literature, in our project we augmented these structure with a distributed scalable platform that support the nature of spatial crowdsourcing application. In particular, we focused on streaming and digesting the data into previously mentioned indexe. Next, we built basic spatial crowdsourcing operation of spatial Range, respectively. Finally, we run extensive experiments on our project data structure and operation and discover the best recommendation in a big distributed streaming platform for spatial crowdsourcing applications.

Figure 2 and 3 showing simple example to explain the crowdsourcing functionality with spatial capability.

Fig. 2: requester looking worker nearest

the requester was unable to determine whether the worker was near the neighborhood or not.

Fig. 3: worker looking for a task nearest

the worker was unable to determine whether the work was near the neighborhood or not. When spatial functionality are been used in crowdsourcing, the requester is allowed to locate the worker ,and the worker is allowed to locate the task shown in Figure 4

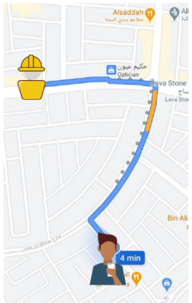


Fig. 4: The requester found the nearest worker and The worker found the nearest task

Secondly, we talked about how to assign tasks in crowdsourcing, and we will discuss the different modes of the task assignment. Thirdly, we will introduce apache ignite and how to store data, then about apache ignite features and capabilities, as well as apache ignite cache, how to store, retrieve and implement data in Query, in addition to apache ignite architecture and types of data processing architectures. Fourthly, we define the spatial index and the cluster types that support the spatial index, in addition to an explanation of the spatial index techniques. Finally, We used gmission, which is synthetic data, to generate the dataset, And we create worker and task locations in a two-dimensional data space; then, we will discuss the different distributions that may affect it.

II. Background

A. CrowdSourcing

With the recent developments on the geographical information system and spatial computing, people can take part in some of the nearby online tasks. This is well known as spatial crowdsourcing [1]. Crowdsourcing is Two casual users interact with a crowdsourcing system that can be classified into the requester and the workers. The requester is the one who request a specific task within a specified period in a specific location. The worker gains reliability when he/she accepts and completes the task correctly, then the worker can earn a reward from the requester of the task [2].

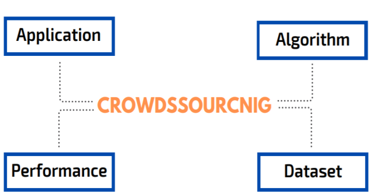


Fig. 5: A taxonomy of crowdsourcing.

Figure 5 depicts the taxonomy of the spatial crowdsourcing services. A detail of each modules of this taxonomy can be described as following:

1) Application: The requester sends a list of tasks that are linked to a specific time and reward and are executed by a crowdsourcing worker to earns the reward or for fun and wasting time. The crowdsourcing applications are grouped into three categories which are (i) Voting System: Voting tasks contain several options for which the crowdsourcing workers determines his answer. The answer that gets the most votes is the correct one. (ii) Information Sharing System: Internet users can share information more easily through websites. Crowdsourcing systems aim to share information among users. The information may also be exploited for other purposes, such as monitoring noise pollution. (iii) Games: There are games that take advantage of people's desire for entertainment to perform tasks and solve specific problems [2].
2) Algorithms: The algorithms in crowdsourcing formalize its design and predict the time a crowdsourcing worker might take to implement a specific task and model the performance in crowdsourcing systems [2].
3) Performance: is a very important aspect of crowdsourcing and is categorized into (i) user participation Which is to study and examine the behavior of crowdsourcing workers. (ii) quality management It is a study of how to get crowdsourcing workers to carry out the required task with good results and the extent to which their behavior affects the task outcomes. (iii) cheating detection It is for some crowdsourcing workers to cheat because their identity is not revealed, and this affects the proper performance of the task [2].
4) Dataset: which is a lot of research dataset in the field of crowdsourcing and it is now available for further research. For example, von Ahn et al. contributed a list of 100,000 images with English labels from their ESP Game11 [2].

The proliferation of mobiles and the speed of their development has extended conventional web-based crowdsourcing to spatial crowdsourcing (SC) where location plays a central role, in fact spatial crowdsourcing is spreading across "the world of speed". Whereas, it enables human workers to solve tasks for other humans in the physical world. So spatial crowdsourcing can be formulated as a goal based on assigning a set of spatial tasks to a group of workers, which means that the workers have to actually travel to those specific locations to perform the tasks assigned to it [3], [4]. For example, Amazon Mechanical Turk is one of the best known and most used of crowdsourcing: It is a marketplace suitable for doing simple tasks and getting them done quickly while saving time and resources for the company [5].

An advantage of spatial crowdsourcing is that when a requester needs to perform a task in a long distant, they will only send the task to the server and the server will assign a worker to perform the task as required. it is no longer require travel to a long-distance to carry out his task, usually offering these rewards is a lot cheaper than formally hiring people to solve problems. Hiring a larger group of people can speed up the problem-solving process to the highest limit. In This project proposal we are going to investigate two main components of spatial crowdsourcing. In particular, we will introduce spatial indexes to a crowdsourcing platform. Next, we will construct two basic spatial operation of crowdsourcing applications. We will conduct an experimental study to recommend the suitable big data framework and the subtitle spatial indexing for crowdsourcing applications.

III. Apache Ignite

1) Apache Ignite Cache:: The Ignite Cache interface provides methods for storing and retrieving data and executing queries,

```
public static void main(String[] args) throws IgniteException {
    // Preparing IgniteConfiguration using Java APIs
    IgniteConfiguration cfg = new IgniteConfiguration();
    // The node will be started as a client node.
    cfg.setClientMode(true);
    // Classes of custom Java logic will be transferred over the wire from this app.
    cfg.setPeerClassLoadingEnabled(true);
    // Setting up an IP Finder to ensure the client can locate the servers.
    TcpDiscoveryMulticastIpFinder ipFinder = new TcpDiscoveryMulticastIpFinder();
    ipFinder.setAddresses(Collections.singletonList("127.0.0.1:47500..47509"));
    cfg.setDiscoverySpi(new TcpDiscoverySpi().setIpFinder(ipFinder));
    // Starting the node
    Ignite ignite = Ignition.start(cfg);
    // Create an IgniteCache and put some values in it.
    IgniteCache<Integer, String> cache = ignite.getOrCreateCache("taskCache");
    cache.put(1, "Task");
    cache.put(2, "Worker");
    System.out.println(">> Created  cache and add the values.");
    // Executing custom Java compute task on server nodes.
    ignite.compute(ignite.cluster().forServers()).broadcast(new RemoteTask());
    System.out.println(">> Compute task is executed, check for output on the server nodes.");
    // Disconnect from the cluster.
    ignite.close();
}
```

Listing 1: Apache Ignite Cache

Ignite provides three different approaches modes of cache operation: PARTITIONED, REPLICATED, and LOCAL.

1) PARTITIONED: approach PARTITIONED Mode, this mode creating one huge distributed store for data By dividing the total dataset evenly into sections, and all sections are split evenly among the participating nodes. The more nodes you have, the more data you can store. The goal of this mode most scalable distributed topology. This mode the Best for the large dataset and frequent updates [11]. e.g.in Figure 6 below, we have key A assigned to a node running in JVM1, which is a partitioned cache with a single backup copy [12].

```
cacheCfg.setCacheMode(CacheMode.PARTITIONED)
cacheCfg.setBackups(1);
    }
```

Listing 2: Partitioned Cache

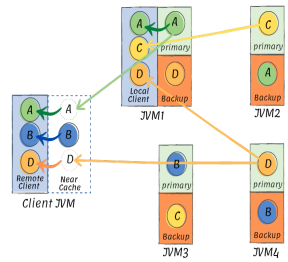


Fig. 6: Partitioned Cache

2) REPLICATED: In approach REPLICATED Mode, all the data is replicated to every node in the cluster, The same data is stored on all nodes of the block. It is available for use without any waiting. And The goal of this approach is to get extreme performance. It is the best for scenarios where the dataset is small, and high read [11]. e.g. in Figure 7 below, the node running in JVM1 is a primary node for key A, but it stores backup copies for all other keys as well (B, C, D) [12].

```
cacheCfg.setCacheMode(CacheMode.REPLICATED)
```

Listing 3: Replicated Cache

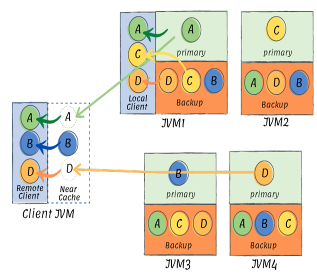


Fig. 7: Replicated Cache

3) LOCAL: approach LOCAL Mode, no data is distributed to other nodes in the cluster data fetching is very inexpensive and fast. It also works very well for read-only [11].

IV. Spatial Indexing

A technology for organizing data in spatial data by organizing records into a space memory, so that it allows you to know the places where the data is stored and access it more quickly. To provide efficient query processing for spatial data, we need special data that supports cluster types.

In spatial data, it is represented in two dimensions so that it is difficult to sort it, but there is a way to convert the 2D data into one-dimensional by using mapping so that it is sort and this is called space-filling curves (SFC) partitioning techniques. There are many ways to represent space-filling curves some of the most popular of which Z-Curve and Hilbert-curve , The Z-curve is Sample point as z-value and partitions the curve into n splits are sorted And The Hilbert-curve is similar to a z-curve, but differs from it in that it uses a Hilbert curve to fill in a curve that has better spatial characteristics but is more complex to generate. There is another technique called space partitioning techniques it contains Grid and Quadtree. Also there is data partitioning techniques such as STR, STR+, and K-d tree [13].

1) Grid File Indexing: Grid File Indexing is a space divided into a rectangular cell, represented in the form of n.m array, so every cell is linked to one disk and every object in a cell is arranged sequentially in the linked page in a cell as shown in Figure 8 [14]. Advantage: The grid file structure seems to meet the requirement in terms of time complexity and space complexity. Disadvantage:

1) Duplicate: Duplicating the object next to the cell will cause an increase in the index And when eliminating this duplication
2) Response time: As the size of the dataset increases, it will become more complex resulting in response time .

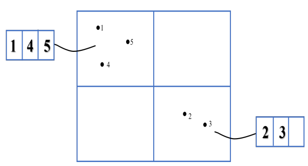


Fig. 8: grid file indexing

Grid File implementation: [15] In the Pseudo code below, the H expresses the average size of the bucket and is specified with a constant value and PNum expresses the number of splits. First we will split the data space so that the bucket size of each cell is balanced And rearrange the records in the grid file so that the records belonging to the same cell are grouped into one bucket and then we store the grid file information in the directory. After that, we check whether the size of the bucket is greater than a certain threshold, because if it is larger than it, the partition will take place again and the information of new sub-grid will be stored in the directory. Then The offset of the parent bucket is redirected to the offset of the sub-grid in the directory and we add a sign to it.

Algorithm 1 how to build grid file

Input: latitude , longitude
Output: grid index

```
latlng x
latlng y
latdiff = (x.latitude-latitude);
londiff = (x.longitude-longitude);
w = ( latdiff / factor );
h = ( londiff / factor );
grid index = (h * maxwidth) + w ;
```

V. Spatial Crowd Sourcing Operation

A Geotag spatial Range Query is determined by both a spatial area A, a temporal interval t and Geotag G. The query finds a set of Tasks $T = \{t_1, t_2, ..t_m\}$ that overlap with the geographic location A and time interval t. For Example, we can find the closest task to the requests in terms of geotags, as each task is associated with latitude, longitude $T_{location} =< lat, long >$, and a specific time t. A typical example of crowd sourcing query a person requests the nearest Nanny close to the neighborhood from 2 pm to 4 pm.

A. Crowd Sourcing Range Query:

A range query is specified by a spatial area A and Geotag G and temporal interval t. In the Range Query, there is a way to find the records inside Query Minimum bounding rectangle (Query MBR) that overlap with Area and temporal interval can be found in two stages. The first stage is filtering, it produces a list of candidates that contains a list of answers that are either true or false. The second stage is Refine stage, at this stage, each candidate would be checked in order to make sure the candidate list reach the final stage of the reported result list [16].

Figure 9 represents an example of a three different spatial index structures of grid file, Quadtree, and R tree. The red rectangle represents Query MBR, which represents the area of interest A.

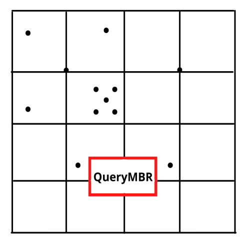


Fig. 9: Grid file Index

Filter Stage: The first stage is filtering, it takes three parameter area A, time t and Geotag G. It produces a list of candidates that contains a list of answers that are either true or false.The filtering stage contain function called overlap represents in **Algorithm 2**, it takes tow parameter QueryMBR and leafsMBR include Condition that using QueryMBR and leafsMBR. The location of the query MBR records will determine then add to list of candidates leafs. The Figure 10 shows the spatial indexes grid file, Quadtree, and R tree after the overlap.

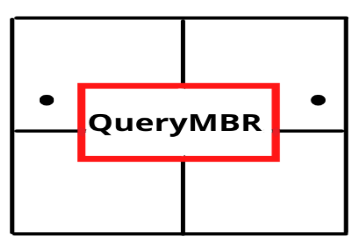


Fig. 10: Grid file Index after applying the overlap

Algorithm 2 Filter Stage.

Input: area A, time t, Geotag G
Output: leafs nodes

overlap(QueryMBR, leafsMBR)
if it's overlap **then**
    add to Candidate leaf node to list.
else
    continue
end if
return List of Candidate leafs

Refine Stage: The second stage is Refine stage and it is shown in **Algorithm 3**, in this stage, it takes area A , time t and Geotag G and receives all of the candidates found during the filtering stage and produces result list . the records inside Query MBR will determined by reading all data records inside each Candidate leaf by using intersection between Records and Query MBR.each candidate would be checked in order to make sure the record is inside the Query MBR.

Algorithm 3 Refine Stage.

Input: area A, time t, Geotag G, List of Candidate leafs
Output: List of records

for all Record ri ∩ QueryMBR **do**
    add ri to the result list.
end for
return List of Result

VI. Spatial CrowdSourcing Dataset

A. gMission SCDataGenerator

gMission is a general spatial crowdsourcing platform for researchers. gMission is open source, It features new technologies including geo-sensing, worker detection, and task recommendation [17].

B. DataSets:

We use synthetic data to generate dataset. The actual dataset includes locations, time stamps for workers, and spatial tasks. let $W = \{w_1, w_2, ..w_n\}$ be a set of n workers. each worker $w_i(1 \leq i \leq 5600000)$ and each worker w is associated with Activeness[a-, a+] for there active time and each worker associated with maximum task maxT(m-, m+). let $T = \{t_1, t_2, ..t_m\}$ be a set of m tasks. each task $t_i(1 \leq i \leq 5600000)$.workers and tasks will be distribution in Lat(minLat, maxLat), Lng(minLng, maxLag) where lat and lng is the boundaries of the location‘ [18].

For synthetic data, we generate worker and task locations in a two-dimensional data space, and there is skewed distributions that may affect it, such as affecting the deadline for performing tasks. and represent different distributions by using QGIS toolbox. The types of distributions is [18]:

1) Skewed Distribution (SD): is similar to the Gaussian distribution. It is that We define the locations of tasks or workers in the Gaussian Cluster uniformly in a two-dimensional space. Figure 11 represents the (SD) distribution, where the red dots represent the tasks and the blue dots represent the workers in a total number 1M for both workers and tasks. The workers dataset amounted to 144 MB and 75.8 MB for the task dataset, as is shown in Table I

Table I represents the properties of both workers and tasks for synthetic data using (SD); the toolbox used is SCDataGenerator [19] toolbox to generate data records for each worker and task. Tables properties contain the amount number of workers(n) and tasks (m), when the latitude of the boundaries range (minLat, maxLat) and the longitude range(minLng, maxLng) represents the working region for the worker, where maxT is the maximum number of tasks range (m-, m+) that a worker can perform, the activeness range [a-, a+] when the worker is available and ready to perform the task, time instance of the task (T-, T+), the task density notes the number of similar tasks at a particular time (d-, d+), and task type region (s-,s+).

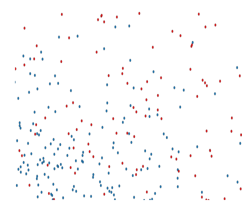


Fig. 11: Synthetic data Distribution using (SD)

Parameters	**Values**	
number of workers, n	1.12M	2.24M
number of tasks, m		
Lat, (minLat, maxLat)	[20.66, 33.87]	[16.91, 36.59]
Lng, (minLng, maxLng)	[37.90, 48.81]	[38.89, 48.95]
maxT, (m-, m+)	[1. 20]	
activeness, [a-, a+]	[0.047, 0.95]	[0.01, 0.96]
time instance, (T-, T+)	[0, 55]	
density, (d-, d+)	[6.14766E-06, 4.99]	[1.42318E-06, 4.99]
task type, (s-, s+)	[0, 64]	
size of dataset, w	144MB	289MB
size of dataset, t	75.8MB	151MB

TABLE I: Properties of Worker and Task using (SD)

Grid file Representation of Skewed Distribution. The numbers of workers was 1.12M. The figure below represent the implementation of Grid file in Skewed Distribution.

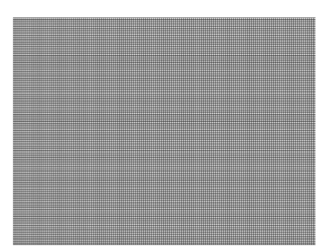


Fig. 12: Grid file in Skewed Distribution

VII. EXPERIMENTS

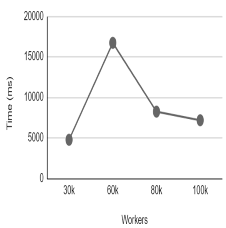
A. Experiment setup

Parameter	**Value**
Distributions	Skewed
Spatial Index	Grid File
Sizes (MB) tasks	(75),37,19 MB
Sizes (MB) worker	147,(72),36 MB
#workers	30,60,(80),100 K
#Tasks	30,60,(80),100 K
Spatial MBR approximity	360x180, 240x100, (180x90)
#KNN	8,16,(32),64
#Workers&Tasks	30,60,(80),100 K

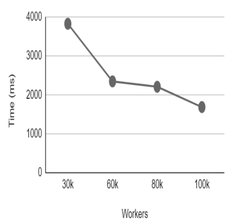
TABLE II: Parameters

All experiments are conducted on a dedicated internal of one server and one client. Each has 64GB memory, 2TB storage, and Intel(R) Xeon(R) CPU 3GHz of 8 core processor. We use apache-ignite-2.8.1 running on Java 8.0. Table II summarizes the configuration parameters used in our experiments.Default parameters (in parentheses) are used unless mentioned.

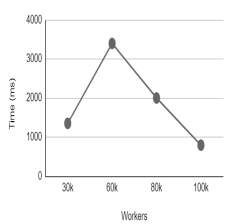
B. Range Query:



(a) Index Time 147MB



(b) Index Time 72MB



(c) Index Time 36MB

Fig. 13: Execution time of Range Query with (Skewed Distribution)

Figure 13 shows the Execution Time of Range Query with different file size. In each experiment, we measure the execution time in ms for Grid file indexing. In Figure 13(a) when the input file size is 147MB in 60k is the worst performance while 30k is the best performance.

In Figure 13(b) when the input file size is 72 MB when number of worker 30k it gives the worst performance while 100k it gives the best performance.

In Figure 13(c) when the input file size is 36 MB the Grid file index had the worst performance when number of worker is 60k and it gives the best performance in 100k .

REFERENCES

[1] Lei Chen and Cyrus Shahabi. Spatial crowdsourcing: Challenges and opportunities. *IEEE Data Eng. Bull.*, 39(4):14–25, 2016.

[2] Man-Ching Yuen, Irwin King, and Kwong-Sak Leung. A survey of crowdsourcing systems. In *2011 IEEE third international conference on privacy, security, risk and trust and 2011 IEEE third international conference on social computing*, pages 766–773. IEEE, 2011.

[3] Hien To, Cyrus Shahabi, and Leyla Kazemi. A server-assigned spatial crowdsourcing framework. *ACM Transactions on Spatial Algorithms and Systems (TSAS)*, 1(1):1–28, 2015.

[4] Leyla Kazemi, Cyrus Shahabi, and Lei Chen. Geotrucrowd: trustworthy query answering with spatial crowdsourcing. In *Proceedings of the 21st acm sigspatial international conference on advances in geographic information systems*, pages 314–323, 2013.

[5] https://www.mturk.com/ amazon. turk.

[6] Srinivasa Raghavendra Bhuvan Gummidi, Xike Xie, and Torben Bach Pedersen. A survey of spatial crowdsourcing. *ACM Trans. Database Syst.*, 44(2):8:1–8:46, 2019.

[7] Peng Cheng, Xiang Lian, Zhao Chen, Rui Fu, Lei Chen, Jinsong Han, and Jizhong Zhao. Reliable diversity-based spatial crowdsourcing by moving workers. *Proc. VLDB Endow.*, 8(10):1022–1033, 2015.

[8] Leyla Kazemi and Cyrus Shahabi. Geocrowd: enabling query answering with spatial crowdsourcing. In Isabel F. Cruz, Craig A. Knoblock, Peer Kröger, Egemen Tanin, and Peter Widmayer, editors, *SIGSPATIAL 2012 International Conference on Advances in Geographic Information Systems (formerly known as GIS), SIGSPATIAL'12, Redondo Beach, CA, USA, November 7-9, 2012*, pages 189–198. ACM, 2012.

[9] Dingxiong Deng, Cyrus Shahabi, and Linhong Zhu. Task matching and scheduling for multiple workers in spatial crowdsourcing. In Jie Bao, Christian Sengstock, Mohammed Eunus Ali, Yan Huang, Michael Gertz, Matthias Renz, and Jagan Sankaranarayanan, editors, *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems, Bellevue, WA, USA, November 3-6, 2015*, pages 21:1–21:10. ACM, 2015.

[10] Yongxin Tong, Lei Chen, and Cyrus Shahabi. Spatial crowdsourcing: Challenges, techniques, and applications. *Proc. VLDB Endow.*, 10(12):1988–1991, 2017.

[11] Michael Zheludkov, Timur Isachenko, et al. *High Performance in-memory computing with Apache Ignite*. Lulu. com, 2017.

[12] https://ignite.apache.org/ apache ignite.

[13] Ahmed Eldawy, Louai Alarabi, and Mohamed F Mokbel. Spatial partitioning techniques in spatialhadoop. *Proceedings of the VLDB Endowment*, 8(12):1602–1605, 2015.

[14] Suhaibah Azri, Uznir Ujang, François Anton, Darka Mioc, and Alias Abdul Rahman. Review of spatial indexing techniques for large urban data management. In *International Symposium & Exhibition on Geoinformation (ISG)*, pages 24–25, 2013.

[15] Ke Yang, Bingsheng He, Rui Fang, Mian Lu, Naga K. Govindaraju, Qiong Luo, Pedro V. Sander, and Jiaoying Shi. In-memory grid files on graphics processors. In Anastassia Ailamaki and Qiong Luo, editors, *Workshop on Data Management on New Hardware, DaMoN 2007, Beijing, China, June 15, 2007*, page 5. ACM, 2007.

[16] Yannis Theodoridis, Emmanuel Stefanakis, and Timos K. Sellis. Efficient cost models for spatial queries using r-trees. *IEEE Trans. Knowl. Data Eng.*, 12(1):19–32, 2000.

[17] Zhao Chen, Rui Fu, Ziyuan Zhao, Zheng Liu, Leihao Xia, Lei Chen, Peng Cheng, Caleb Chen Cao, Yongxin Tong, and Chen Jason Zhang. gmission: A general spatial crowdsourcing platform. *Proceedings of the VLDB Endowment*, 7(13):1629–1632, 2014.

[18] Peng Cheng, Xun Jian, and Lei Chen. An experimental evaluation of task assignment in spatial crowdsourcing. *Proc. VLDB Endow.*, 11(11):1428–1440, 2018.

[19] https://github.com/gmission/scdatagenerator scdatagenerator.