ISSN(Online) : 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 4, April 2016

k Nearest Neighbour Join using Map Reduce for effective Smart Travel Guide

Prashant Raut, Rohan Deokar, Priyanka Kanse, Akshay Bangale, Rahul Thorat

Asst. Professor, Dept. of Computer, SCOE, Savitribai Phule Pune University, Maharashtra, India

B.E. Student, Dept. of Computer, SCOE, Savitribai Phule Pune University, Maharashtra, India

B.E. Student, Dept. of Computer, SCOE, Savitribai Phule Pune University, Maharashtra, India

B.E. Student, Dept. of Computer, SCOE, Savitribai Phule Pune University, Maharashtra, India

B.E. Student, Dept. of Computer, SCOE, Savitribai Phule Pune University, Maharashtra, India

ABSTRACT: An expanding number of Web applications, for example, companion's proposal rely on upon the capacity to join objects at scale. The conventional methodology taken is closest neighbor join (additionally called comparability join), whose objective is to and, in view of a given join work, the nearest set of items or every one of the articles inside of a separation edge to every item in the data. The adaptability of methods using this methodology regularly relies on upon the attributes of the items and the join capacity. On the other hand, some certifiable join capacities are complicatedly built and continually developing, which makes the outline of white-box techniques that depend on comprehension the join capacity unfeasible. Discovering a procedure that can join to a great degree vast number of items with complex join capacities has dependably been an extreme test. In this paper, we propose a commonsense option approach called close neighbor join that, in spite of the fact that does not and the nearest neighbors, ands close neighbors, and can do as such as to a great degree substantial scale when the join capacities are intricate. Specifically, we plan and actualize a super-adaptable framework we name SAJ that is prepared to do best exertion joining of billions of items for complex capacities.

KEYWORDS: Android, Smart travel guide.

I. INTRODUCTION

Now a day mobile phone is a necessary part of the people's life. There is continuously rising in a number of mobile computing applications, concentred on the people's daily life. In such applications, location dependent systems have been detected as an important application. Such application which presents the architecture and implementation of such a location is commonly known as Smart Travel Guide. We propose architecture of mobile tourist guide system for Android Mobile Phones that is able to provide tourism information to the mobile users conveniently. Our system takes advantage of light-weighted mash up technology that can combine more than one data sources to create value-added services, while overcomes the limitations of mobile devices.

The application aims to develop detailed texts and other guidance information are provided, and so people can better understand the tourist attractions and make decision objectively. A problem is shown that tourists are not able to get travel information timely when they are on the move. Therefore, we intend to explore how to build a mobile tourist guide system based on mash up technology to solve this problem.

II. LITERATURE SURVEY

1. Near Neighbor Join
   Year of Publication - 2014
   Authors - Herald Kllapi, BoulosHarb, Cong Yu.

ISSN(Online) : 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 4, April 2016

Description: This paper introduces SAJ, a Scalable Approximate Join system that performs near neighbour join of billions of objects of any type with a broader set of complex join functions. The algorithm used in this paper finds all the close set of neighbours and can do so on an extremely large scale.
Techniques used: Scalable Approximate Join (SAJ)
Algorithms Used: Three Phases
(a) Bottom Up
(b) Top Down
(c) Merge

2. Processing Theta-Joins using Map Reduce
   Year of Publication - 2011
   Authors - AlperOkcan, MirekRiedewald

Description: This paper proposes a join model that simplifies creation of and reasoning about joins in Map Reduce techniques which enable efficient parallel execution of arbitrary theta-joins. Everything is achieved by simply specifying the appropriate (sequential) Map and Reduce functions. It focuses on the problem of how to map arbitrary join conditions to Map and Reduce functions, i.e., a parallel infrastructure that controls data flow based on key-equality only.
Technologies Used:
(a) Map
(b) Reduce
(c) M-bucket I

3. Efficient Processing of k Nearest Neighbor Joins using Map Reduce
   Year of Publication - 2012
   Authors - Wei Lu Yanyan Shen Su Chen Beng Chin Ooi

Description: In this paper, it has been investigated how to perform kNN join using Map Reduce which is a well-accepted framework for data-intensive applications over clusters of computers. An effective mapping mechanism that exploits pruning rules for distance filtering, and hence reduces both the shuffling and computational costs is designed.
Technologies Used:
(a) kNN Join
(b) Map Reduce Framework
(c) Voronoi Diagram-based Partitioning
Algorithms Used:
(a) Bounding KNN.
(b) kNN join.

III. MATHEMATICAL MODEL

WE ARE GIVEN INPUT $I = \{o_i\}_{i=1}^{N}$ WHERE N IS VERY LARGE (E.G. BILLIONS OF OBJECTS). A USER-CONSTRUCTED JOIN FUNCTION F J: I*I->R THAT COMPUTES A DISTANCE BETWEEN ANY TWO OBJECTS IN I, AND THAT WE ASSUME TO KNOWLEDGE EXCEPT THAT FJ SATISFIES THE TRIANGLE INEQUALITY PROPERTY (FOR SAJ TO PROVIDE QUALITY EXPECTATION). A PARAMETER K INDICATING THE DESIRED NUMBER OF NEIGHBOURS PER OBJECT, AND MACHINE RESOURCE CONSTRAINTS. THE TWO KEY RESOURCE CONSTRAINTS ARE

1) THE NUMBER OF OBJECTS EACH MACHINE CAN BE EXPECTED TO PERFORM AN ALL-PAIRS COMPARISON ON.

2) THE MAXIMUM NUMBER OF RECORDS EACH SHUFFLE PHASE IN THE PROGRAM IS ABLE TO HANDLE, WHICH CAN BE DERIVED FROM THE NUMBER OF MACHINES AVAILABLE IN THE CLUSTER.

WE PRODUCE OUTPUT $O = \{(o_i \rightarrow R_i)\}_{i=1}^{N}$ WHERE RI IS THE SET OF K OBJECTS IN I WE DISCOVER TO BE NEAR o_i, ACCORDING TO FJ. FOR EACH o_i, LET R_i^{nearest} BE THE TOP –K NEAREST NEIGHBOURS OF OI.I.E.

ISSN(Online) : 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 4, April 2016

$$\forall (j,k), o_j \notin R_i^{\text{nearest}}, o_k \in R_i^{\text{nearest}}, F_J(o_i,o_j) \geq F_J(o_i,o_k).$$
Let $E_i = \text{AVG}_{o \in R_i}(F_J(o_i,o)) - \text{AVG}_{o \in R_i^{\text{nearest}}}(F_J(o_i,o))$ be

THE AVERAGE DISTANCE ERROR OF RI. OUR SYSTEM ATTEMPTS TO REDUCE AVG(EI) IN A BEST-EFFORT FASHION WHILE RESPECTING THE RESOURCE CONSTRAINTS.

HERE WE SHOW THE NUMBER OF MAP REDUCE ITERATIONS IS $O(\log(N/n)\log((kN + n^2P)/p)/\log(n/m)\log(p/P))$ AND WE BOUND THE WORK DONE BY THE TASKS IN EACH OF THE BU, TU AND MERGE PHASES.

MR ITERATIONS: LET L BE THE NUMBER OF MAP REDUCE ITERATIONS IN THE BU PHASE. SUPPOSE AT EACH ITERATION I WE PARTITION THE INPUT FROM THE PREVIOUS ITERATION INTO LI PARTITIONS. EACH PARTITION R HAS SIZE : $\lfloor mL_{i-1}/L_i \rfloor$ SINCE M REPRESENTATIVES FROM EACH OF THE LI-1 PARTITIONS ARE SENT TO THE NEXT LEVEL. THE NUMBER LI IS CHOSEN SUCH THAT N/2<=|R|<+N ENSURING THAT THE PARTITIONS IN THAT LEVEL ARE THE LARGE BUT FIT IN ONE TASK (I.E. DO NOT EXCEED N).THIS IMPLIES THAT AT EACH ITERATION WE REDUCE THE SIZE OF THE INPUT DATA BY A FACTOR $c = L_{i-1}/L_i \geq n/(2m)$. .THE BU PHASE STOPS AT THE FIRST ITERATION WHOSE INPUT SIZE IS <=N (AND W.L.O.G. ASSUME IT IS AT LEAST N/2). HENCE, THE NUMBER OF ITERATIONS IS AT MOST:

$$L = \log_c(N) - \log_c(n/2) = \log_c(2N/n) \leq \frac{\log(2N/n)}{\log(n/(2m))}.$$

For example, if $N = 10^9$, $n = 10^3$, and $m = 10$, then $L = 4$.

IV. ALGORITHM

For our project "Smart Travel Guide" we have used 2 different algorithms with 2 approaches
Algorithms:

1. Near neighbor Join:
   a. Top Down Approach
   b. Bottom Up approach
2. Map Reduce Functionality

We used location based dataset for implementation of these techniques. We fetch user's positions by GPS technology in mobile devices.

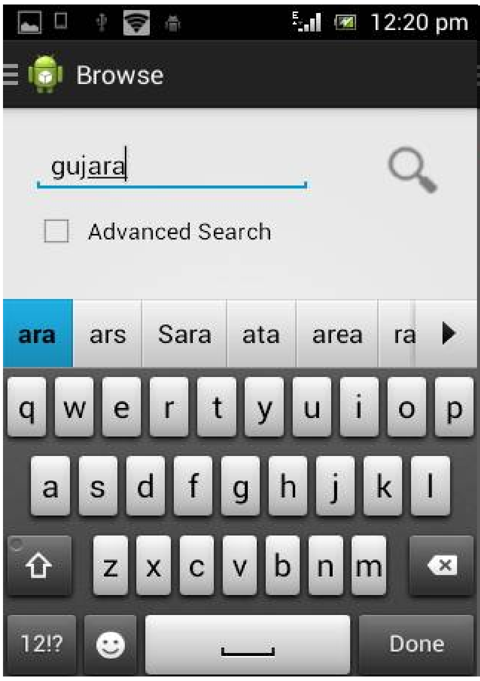
Explanation for GPS: GPS, the Global Positioning System run by the United States Military, is free for civilian use, though the reality is that we're paying for it with tax dollars. However, GPS on cell phones is a bit murky. In general, it won't cost you anything to turn on the GPS in your cell phone, but when you get a location it usually involves the cell phone company in order to get it quickly with little signal, as well as get a location when the satellites aren't visible (since the government requires a fix even if the satellites aren't visible for emergency 911 purposes). It uses up some cellular bandwidth. This also means that for phones without a regular GPS receiver, you cannot use the GPS at all if you don't have cell phone service. For this reason most cell phone companies have the GPS in the phone turned off except for emergency calls and for services they sell you (such as directions). This particular kind of GPS is called assisted GPS (AGPS), and there are several levels of assistance used.

V. RESULTS

1. **Post new:** Here the user once logged in to the app can post his journey details via filling the form as shown in fig.1. The user can post from where he is going to travel source to destination, date and time details, travelling via, vehicle details etc in the post new form. Also the user can select the visibility for his/her post.

Fig.1 Post new

2. **Browse:** In the browse section the user can search for the places where he/she wants to travel as shown in fig.2. Advance search is also provided in which the user can select many choices such as gender, vehicle type, Ac/Non Ac etc. as shown in fig.3.



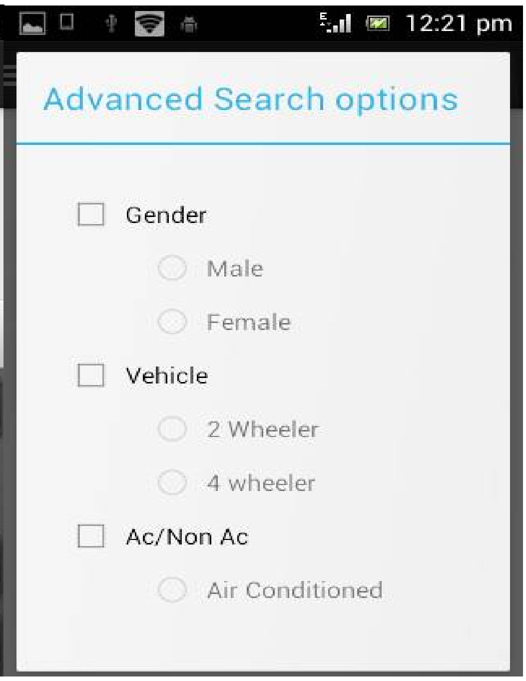


Fig.2 Browse Fig.3 Advance search option

After selecting the desired location the list of peoples travelling to that location are traced by using near neighbour join algorithm at the database and the details are forwarded to user. The user can select any person randomly who is

ISSN(Online) : 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 4, April 2016

travelling via same route as convenient to the user. He can also view the details such as travelling from, travelling to, travel date and time, vehicle details, availability of seats etc. As shown below in fig.4.

We can also trace the journey details by using GPS as shown in fig.5. The GPS will completely provide the route and distance to be travelled by navigating the user.

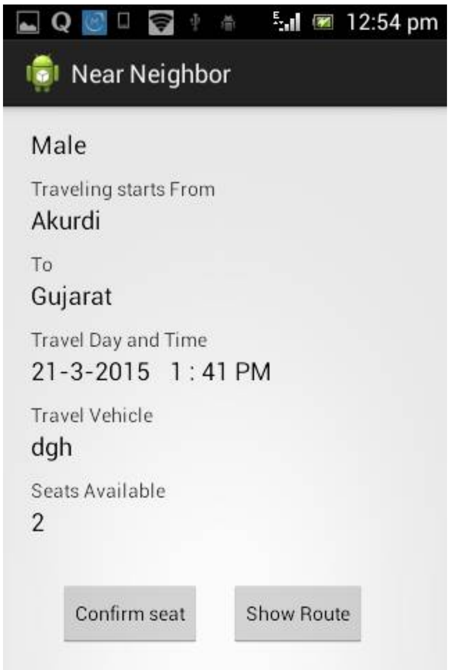


Fig.4 Availability of seats

Fig.5 Navigation

VI. CONCLUSION AND FUTURE WORK

Smart travel guide is very effective means to reduce pollution and the congestion of vehicles in cities. It also provides an eco-friendly way to travel. It also provides an opportunity to meet new people. As today most people prefer private vehicle to travel due to delay caused in public transport system and luxuries provided by private vehicles. Pre-registration ensures that only identified people get into the vehicle so that trust can be established. The people registered are allotted specific days on which they should take their private vehicle, so that no inconvenience is caused to its registered passengers for daily commute. Thus the proposed Smart travel guide will be effective in reducing environment pollution.

REFERENCES

1. Kllapi,H., Harb, B. ; Cong Yu, Dept. of Inf. &Telecommun., Univ. of Athens, Athens, Greece, "*Near neighbor join*,"
2. J. Dean and S. Ghemawat, "*MapReduce: simplified data processing on large clusters*," Commun. ACM, vol. 51, pp. 107-113, January 2008.
3. A. Okcan and M. Riedewald, "*Processing theta-joins using MapReduce*," in SIGMOD, 2011.
4. Hongrae Lee, Raymond T. Ng, Kyuseok Shim"*Similarity join size estimation using locality sensitive hashing*"