Customized Travel Planner using MapReduce and Approximation Algorithm

Pallavi S Ghogare
Department of Computer Engineering
MMCOE,Karvenagar, Pune,
Savitribai Phule Pune University,
Maharashtra

Harmeet K. Khanuja
Department of Computer Engineering
MMCOE, Karvenagar, Pune,
Savitribai Phule Pune University,
Maharashtra

ABSTRACT
It is fun to travel but painful to arrange the trip. When travelers start off planning they need flights, accommodation and attractions. Which scattered across multiple websites on the internet? Traveler spends time scouting each of them for the best deals, and gets the attraction reviews from established planners in the market. It will be always good if traveler gives specified designations and time he wants to spend for the trip and some platform will automatically did everything for traveler with added bonus of optimal and customized itineraries. This system is designed for such travelers to design customized itineraries which will be optimal and consist of Point-Of-Interest (POI) selected by traveler, rather than go and visit the traditional and static trip plans by many travel agencies. This system is two-stage processing system for cost effective and optimal results. First is preprocessing stage works offline uses parallel processing engine as MapReduce to precompute Single-Day Itineraries. In further stage which is online the precomputed itineraries are combined to give multiday itineraries. These itineraries produced are optimal as per travelers selected Point-Of-Interest (POI).Here Greedy Approximation Algorithm is used to combine the single day itineraries. In this way Team-Orienteering-Problem (TOP) is transferred to Set-Packing Problem another NP-complete problem.

Keywords
MapReduce, team-orienteering-Problem, Itinerary, Point-Of-Interest, location-based service.

1. INTRODUCTION
Travel agencies are unable to satisfy individual requirements of customized tour plans though they seem efficient for experienced travelers. Some interested POIs are not at all part of traditional plans which is inconvenient for bag-pack traveler. Therefore, they have to plan their trips in detail, such as selecting the hotels, picking POIs for visiting, and contacting the transportation service.

To assist traveler to preplan or plan their travel activities during their trip it is needed to provide traveler with real time information. Travelers, whether leisure -tourists- or business travelers will want on top of the attractions that a given destination offers, a fast, flexible, and convenient transport mean(s) to reach it, as well as local weather information, accommodation options and availability, recreational and cultural activities and other value added information services.[5]

However, information is becoming the essential component of the travel service, and therefore it is necessary to be provided at the right time, place and format. Moreover, real time information gains extremely valuable importance due to the stochastic, dynamic changes of the travel schedules of the various transportation modes.

In other words, what travelers in general and tourists in particular expect are services that:

i. are customized to their individual needs and preferences

ii. Are available to them in a timely and accurate manner. However, it is impossible to list all possible itineraries for users. A practical solution is to provide an automatic itinerary planning service. The user lists a set of interested POIs and specifies the time and money budget. The itinerary planning service returns top-K trip plans satisfying the requirements. [1]

Actually computing optimal itineraries is a Team-Orienteering-Problem with no polynomial approximation. TOP is the problem where a limited number of vehicles are available to visit POI from a potential set, the travel time of each vehicle being limited by a time quota, POI having different corresponding profits, where a POI being visited at most once. The aim of TOP is to organize an itinerary of visits so as to maximize the total profit. [8]

As TOP problem can be solved using some approximations which lead to incorrect results which are not optimal. So here we are converting TOP into another weighted Set-Packing NP-Complete problem.[5]

Suppose we have a finite set S and a list of subsets of s. Then, the set packing problem asks if some k subsets in the list are pairwise disjointing i.e. no two or more set contains same POI. In this way we can get set of single-day itineraries which are disjoint and combined further for computing multiday itinerary.

1.1 Motivation for using Hadoop
Although input data is small in size the results of possible itineraries produced is very large and these computation are so complicated cannot be done with single machine. MapReduce is solution to partition the task between different machines. Also parallel computing effectively reduces the running time of preprossesing. Scalability is increased by adding more nodes to a cluster. The workload is shared by the all nodes. [6]

2. LITERATURE SURVEY
In [2] an interactive and user friendly travel planning system is proposed. It basically works on Point-Of-Interest (POI) feedback model constructed through feedback of the users who completed their itinerary. The system will recommend

best itinerary plan on the basis of feedback of the POI. In this paper algorithms used are able to generate single-day itineraries only. And needed to collect the comments on each new POIs from the users, which is very time consuming. In [3] photo-streams are used to generate meta-data about the any Point-Of-Interest(POI).All photo-streams by individual user is col-lected and processed. Then all Photo-streams by all users are combined to form a network graph of POIs. From these graphs automated travel itineraries are generated. In this approach first data mining algorithms are used to retrieve the information and again further processed for itinerary planning. In [4] sub graph analysis is done using Hadoop. Here MapReduce framework is used to reduce cost of processing NP-complete problem. But in this approach Team-Orienteering problem is used to solve directly which will not ensure about the optimal results. In this system TOP is converted into weighted Set-Pack problem which further uses initialization-adjustment model to get opti-mal resulted itineraries. In [5,8,9] Team-orienteering-Problem formulations and different approximations based algorithms are proposed. In [12] a Location based information delivery system is designed. The influence of such a richer context model on the user interaction for both the capturing of context and context-aware user/device interaction is discussed.

3. MATHEMATICAL MODEL

Consider S = P, Sp, K, G, L, Lk, H, map(), reduce(),initialization(),adjustment() be the System where,

P = set of POI in the system.

(Sp,K) = user input to the system where, Sp = set of user selected POI list.

K = no. of days dedicated for trip.

G = V, E is POI graph generated from user input. L = V0, V1., hj Single-day itinerary

H = h1,., hj set of Hotels as a POI list.

Lk = L1, L2, Lk K day Itinerary set of single-day itineraries. map() = Mapper function to compute intermediate possible single-day itineraries.

reduce()= Reducer function which removes duplicate itineraries and shuffles itineraries with highest score(profit) initialization() = function used to generate K-day itinerary as seed.

adjustment() = function used to generate improved itineraries with their independent set.

4. PROPOSED SYSTEM

To reduce the processing cost here two-stage planning system is used. In its preprocessing stage single-day itineraries are precomputed via MapReduce jobs. In its online stage an approximate search algorithm is used to combine single-day itineraries. Fig.1. shows system architecture.

4.1 Preprocessing

In the preprocessing, POIs are set into an undirected graph, G. The distance of two POIs is evaluated by Google Maps APIs [1].In the preprocessing stage we iterate all candidate single-day itineraries using a parallel processing framework, MapReduce. The results are maintained in the distributed file system (DFS) and an inverted index is built for efficient itinerary retrieval. To construct a multiday itinerary, we need to selectively combine the single itineraries. The preprocessing stage, in fact, transforms the TOP into a set-packing problem [5], which has well-known approximated algorithms.

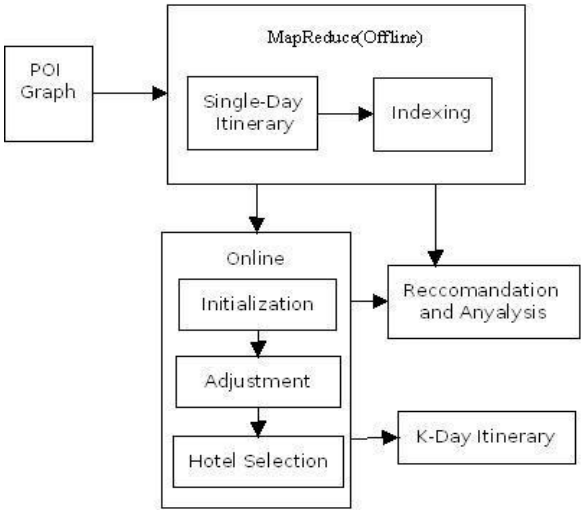

Fig.1. System Architecture

4.1.1 Single-Day Itinerary

Using MapReduce iterations Single-day itineraries are computed. The mappers load the partial paths from the DFS, which are generated in the previous MapReduce jobs. Then try to append new POI to the existing itineraries. For each new path, it test whether it can be completed within one day. If not, it will discard the new path.

4.1.2 Itinerary Index

To efficiently locate the single-day itineraries, an inverted index is built. The key is the POI and the values are all itineraries involving the POI. By scanning the index, we can retrieve all the itineraries. We create an index file for all POI's in the DFS. The file includes all single itineraries involving the POI, which are sorted based on their weights. For example, if"1.idx" contains all itineraries for the first POI. The itinerary"1j5j20j12j40"is the most important itinerary in the index file with weight 320. The inverted index is constructed via a MapReduce job. The mappers load the single-day itinerary and generate key-value pairs for all POI's involved in input. The reducers collect all itineraries for a specific POI and sort them based on the weights before creating the index file. In our system, the size of the index file may vary a lot. Some POI may have an extremely large index file, due to its popularity and short visit time. In reducers, those POIs may result in the exception of memory overflow in the sorting process. To address this problem, in the map phase, instead of using the POI as the key, we generate the composite key by combining the POI and the itinerary weight [1].

4.2 Greedy-Based Approximation Algorithm

After the itinerary indexes are constructed, the user request can be processed by selecting k best itineraries from the indexes. Namely, the problem of generating optimal k-day itinerary is transformed into a weighted set-packing problem.

[1] There are three phases in this algorithm.

4.2.1 Initialization

The initialization phase applies the greedy-based heuristic approach to generate a k-itinerary as the seed, which is further improved in the adjustment phase by replacing the itineraries

with their independent sets. [1] First sort the selected POIs by their weights. Then in each iteration; we try to form a group, which contains a subset of POIs that can be accessed within one day. Then greedily select the POI with shortest distance and add it into a group. There are maximally k groups generated. All groups are used as our seeds for searching the index. We will use the First itinerary that contains all the POIs in the group as our candidate itinerary. Although after the weight adjustment, itineraries in the index file are no longer sorted by the weights.To improve the weights of the obtained itineraries in the greedy algorithm, we adopt the adjustment phase.

4.2.2 Adjustment

In the adjustment phase, new solutions are searched and used to replace the greedy itineraries. The process repeats until no improvement can be obtained.The adjustment phase greatly increases the processing cost.In the adjust-ment phase, the query engine loads the itinerary index from the DFS, which incurs high I/O cost. One way to reduce the cost is to increase the index buffer size. After an indexed itinerary is loaded from the DFS, we cache it in the buffer. If the buffer is full, we apply the LRU strategy to remove the less used entries. [1]

Hotel Selection In fact, hotels can be considered as a special type of POIs. It must appear as the last POI in the itinerary. We need to calculate the traveling time from other POIs to the hotel POIs. Hotel POIs do not incur access cost and their weights are set as users rankings for the hotels

4.2.3 Recommendation and Analysis

In this module itineraries generated and used by tourist are analyzed for the performance of the system. And also by checking ranking of POI through feedback new itineraries can be defined which will differ from traditional tour packages.

If a tourist comes with similar set of POI choices which is already planned with another tourist then our system will suggest same itinerary to that tourist rather computing another itinerary.

5. EXPERIMENTAL RESULT

5.1 Dataset Description

To evaluate performance of the system here traveling in-formation is crawled from Yahoo Travel. Yahoo classifies POI into hotels, things to do and cities. We use hotels and things to do for our experiment. Here minimum 400 POI's are used congaing both Hotels and visiting places as POI. As far as this is the largest dataset used for automatic itinerary formaton. Initial weight is also crawled from Yahoo travel. We accumulate each POI score by user as its weight. The average visiting time can be calculated from Yahoo as well. Edge cost between two POI is computed using Google Map API.also assumed each user will spend 8 hours for traveling per day.

5.2 Experimental Settings

Following Table shows experimental settings of the system. Different parameters are as follows

Table 1 Experimental Setting Parameters

Parameter	Value
K(No.of Days)	3(1-5)
A	2
Sp(User POI List)	10(5-20)
No.of MapReduce per node	2
Data Chunk Size of HDFS	64 M
No. of Max. Mapreduce Jobs	10

5.3 Statistical Result

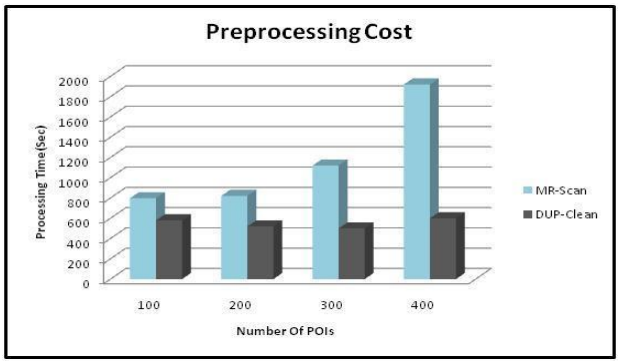

Fig 2. Preprocessing Cost

Fig.2. shows the total cost for preprocessing i.e. MapReduce jobs and clean job which is offline process.

As number of POIs increases from 100 to 400 MR-Scan cost increases. But cost of Dup-Clean is not correlated with number of POI as its result is neutralized by parallel processing.

Fig. 3. In our dataset most of itineraries consist of

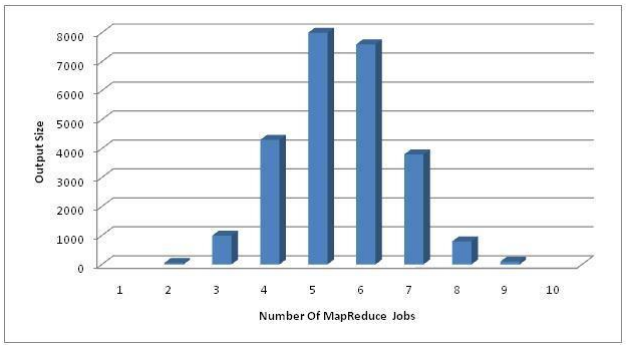

Fig 3. Size of Single-Day itinerary

4-5 POI. By setting m to 10 we process most of itineraries. Fig. 4. shows effect of number selected POI where cost of MR-Set increases as number of POI selected increase as in adjustment phase it has to look up the indexes for better replacement. However it is effective than TOP whose cost is more than MR-Set.

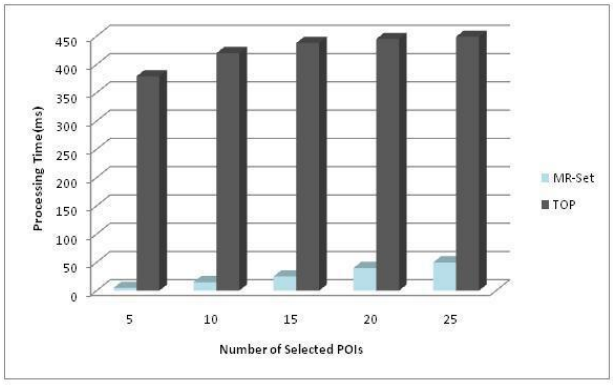

Fig.4. Effect Of selected POI's (processing Time)

Here Initialization-Adjustment Model is used. Also we have feedback from travelers using which we have the analysis and system. In future scope dataset which used is limited to some geo-locations it can be broader, may be global.

6. CONCLUSION AND FUTURE WORK

An automated itinerary is generated as per the travelers selected list of Point-Of-Interests (POI).Which gives traveler a customized multiday travel plans. This problem of generating optimal itineraries is NP-complete problem, which has no polynomial time approximate algorithm. For efficient travel itineraries here two-stage processing is used. In first stage MapReduce framework is used to generate indexed single-day itineraries. Parallel processing engine allows to iterate through whole dataset and index as many as itineraries as possible. After Preprocessing stage Team Orienteering Problem is converted into weighted Set-Pack Problem. In this stage by using Greedy-based Approximation algorithm single-day itineraries are combined to produce multiday itinerary.

7. REFERENCES

[1] Gang Chen, Sai Wu, Jingbo Zhou, and Anthony K.H. Tung, ”Au-tomatic Itinerary Planning for Traveling Services” ,TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 26, NO. 3, MARCH 2014.

[2] S.B. Roy, G. Das, S. Amer-Yahia, and C. Yu, ”Interactive Itinerary Planning”, Proc. IEEE 27th Intl Conf. Data Eng. (ICDE), pp. 15-26, 2011.

[3] M.D. Choudhury, M. Feldman, S. Amer-Yahia, N. Golbandi, R.Lempel, and C. Yu,” Automatic Construction of Travel Itineraries Using Social Breadcrumbs”,Proc. 21st ACM Conf. Hypertext and Hypermedia (HT), pp. 35-44, 2010.

[4] Z. Zhao, G. Wang, A.R. Butt, M. Khan, V.A. Kumar, and M.V.Marathe, ”SAHAD: Subgraph Analysis in Massive Networks Using Hadoop”,IEEE Intl Parallel and Distributed Processing Symp.(IPDPS), 2012

[5] V.S.P. de Aragao, H. Viana, and E. Uchoa, ”The Team Orienteering Problem: Formulations and sfor Transportation Modeling Optimization and Systems (ATMOS)”, vol. 14, pp. 142-155, 2011.

[6] J. Dean and S. Ghemawat, ”MapReduce: A Flexible Data Processing Tool”,Comm. ACM, vol.53, pp. 72-77, Jan. 2010.

[7] http://hadoop.apache.org/2014.

[8] W. Souriau, P. Vansteenwegen, G.V. Berghe, and D.V. Oudheusden,”A Path Relinking Approach for the Team Orienteering Problem”, Comput-ers and Operations Research, vol. 37,pp. 1853-1859, 2010.

[9] F. Chierichetti, R. Kumar, and A. Tomkins, ”Max-Cover in Map-Reduce”, Proc. 19th Intl Conf.World Wide Web (WWW), pp. 231-240, 2010.

[10] P. Vansteenwegen, W. Souriau, and D.V. Oudheusden,”The Orienteering Problem: A Survey”,European J. Operational Research, vol. 209, pp. 1-10, Feb. 2010.

[11] A.Z. Idris, and N.A.Yahaya, ”Design and Implementation of an Aggregation-based Tourism Web Information System”, IJCSNS Inter-national Journal of Computer Science and Network Security, vol. 9(12), pp.143-148, 2009.

[12] Ando, and Y. Mimura,” A Study to Develop aan Information Providing System on Travel Time”, Int. J. ITS Res., vol. 8, pp. 77-84, 2010.