An Asynchronous Distributed Data Collection Approach for Mobile Group Consumption

Weiran Chen[†], Zuoyou Li[‡], Yue Liang[‡] and Weiping Zhu[‡§*]
[†]Economics and Management School, Wuhan University, P. R. China
[‡]International School of Software, Wuhan University, P. R. China
[§]State Key Lab. for Novel Software Technology, Nanjing University, P. R. China

***Abstract*—Mobile group consumption is an important kind of consumption where a group of people such as couples, families, colleagues, friends, etc. use mobile devices to query or check the information of commodities and make the decision of consumption together. To support the business and marketing activities targeting such kind of consumption, a proper data collection system is needed. Existing data collection systems that can be used for mobile group consumption are based on centralized server, which suffers from performance bottleneck, one-point failure, overhead of powerful server, etc. Moreover, existing works are based on synchronous clock, which is sometimes impossible due to hardware constraint or privacy concerns, or at least need large overhead of synchronization from time to time. In this paper, we propose an asynchronous distributed approach to collect the data of mobile consumption for the first time. Our approach does not need a central server or synchronous clock. We first formally build the system model of data collection of mobile consumption in a distributed way and without synchronized clock. The we propose the solution framework which covers the procession in a local sub-region, in consecutive sub-regions, and in several or even all the sub-regions. After that, a detailed approach is proposed to demonstrate how to analyze the casuality of two events happened in the system based on pure message exchange. Extensive simulation results show that the proposed approach is effective for data collection of mobile group consumption and is complementary to the classic method based on synchronized clock.**

***Index Terms*—Asynchronous, Distributed, Data Collection, Mobile Group Consumption.**

I. Introduction

Group consumption is an important kind of consumption that a group of people such as couples, families, colleagues, friends, etc. perform consumption together. This kind of consumption is different from other consumptions since the interaction and negotiation among the members of a group is complex and leads to the final decision of the consumption. Group consumption is not uncommon in our daily life. In fact, people spend 70 percent of their time in public places with other persons [1] and group consumption probably happens when people are gathered.

Group consumption is previously investigated by the researches on business management, marketing, psychology, and so on. In recent years, group consumption attracts a growing attentions from computer science due to the rapid progress of mobile internet and mobile e-commerce. The people use mobile devices such as smartphones, pads, laptops, etc. to scan, search and receive the information from retails and order various services and goods. We call the group consumption based on mobile devices as *mobile group consumption*. An example of it can be seen in Fig.1.

According to a survey, about 46% people had made purchases using their mobile device and 80% people plan to conduct mobile consumption in the next 12 months [2]. In USA, the mobile consumption reaches 86 billion US dollars, which is about 24 percent of overall consumption [3]. In China, according to the latest report of China Internet Network Information Center, the mobile consuming is half of the electronic consuming [4]. Many companies begin to put increasing marking effort to mobile consumption. In 2014, the global advertisement for mobile consuming reaches 32.7 billion US dollars, which is about one forth of the total network advertisement [5].

To analyze mobile group consumption, it is highly demanded that an effective data collection is built. The data need to be collected include the trajectories and actions of members in a group in a lasting time duration. Various technologies can be used for such purpose. For example, WIFI signals from access points to mobile phones of consumers can be collected. An RFID tag attached to a consumer and several RFID readers scattered in the environment also can be utilized for analysis.

Existing data collection system [6], [7], [8] that can be used for mobile group consumption usually based on centralized procession. A central server is needed to gather the information from the devices in the system. Although it works in small-scale systems, it encounter the following challenges for mobile group consumption: 1) In many scenarios, it is difficult to find a proper powerful server. 2) Even there is a server available, the centralized server suffers from serious one-point failure and computation bottleneck, especially in the scenarios including a large number of mobile consumers and shops. Moreover, existing work assume the existence of a synchronized clock, which incurs high overhead when the consumers is in large number and constantly moving. Sometimes the synchronization is even impossible due to hardware constrains of devices or privacy concerns from people. A new approach is needed to overcome the aforementioned problems.

In this paper, we design an asynchronous distributed data collection approach for mobile group consuming. Our approach is not relied on central server or synchronized clock but on distributed solution and asynchronous message com-

*Corresponding Author: Weiping Zhu, Email: wpzhu@whu.edu.cn

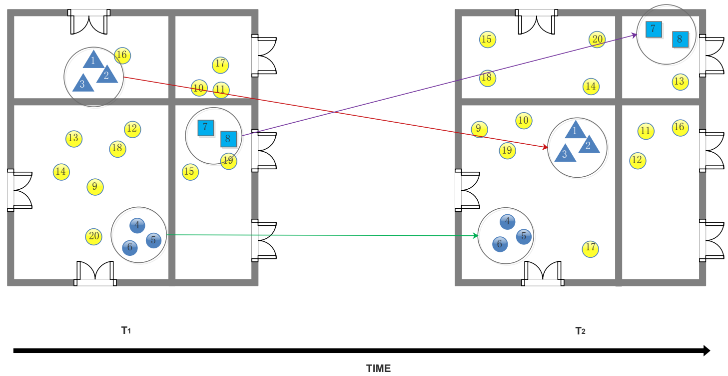

Fig. 1. Illustration of mobile group consumption

munications. We first build the asynchronous distributed data collection system model for this problem. Then we propose a three-layer solution for distributed procession, including that for local sub-regions, consecutive sub-regions, and multiple sub-regions. After that, we propose a novel approach to detect the casuality of two specified marketing events, a typical way to analyze the effectiveness of marketing strategies. The temporal order of different events happened in the system is not based on synchronized clock but the sending and receiving of messages. Extensive simulations are carried out to evaluate our system. The results show that the proposed approach can effective support the data collection of mobile group consumption. In summary, this paper makes the following contributions:

- We built asynchronous distributed data collection system model for mobile group consumption. The data collection is not based on centralized server or synchronized clock.
- We proposed an approach to collect data for mobile group consumption in a asynchronous distributed way. We designed a three-layer mechanism that the data collection is firstly handled locally, and then coordinated in convective regions or more regions if the data collection spans a wide area. The temporal order of different events happened in the system is based on exchange of messages and the induced vector clocks.
- We conduct extensive simulations to validate the proposed approach. The results show that the proposed algorithm is quite effective.

The rest of the paper is organized as follows: Section II builds the system model for mobile group consumption. The detailed solution is proposed in Section III which include the three-layer solution framework and a detailed approach to detect the casuality in based on asynchronous communications. The simulation results are reported in Section IV. Section V reviews the related works and finally Section VI concludes the paper.

II. System Model

In this paper, we assume that mobile group consumption happens in a place such as a shopping mall, a supermarket, etc. where several retailers conduct promotion activities. Several groups of people wandering around the place for shopping and can be attracted by the promotion activities. Each group includes one or more members. The members reach the consensus of consumption decision via their discussion and negotiation. Each promotion activity is assumed lasting for a certain period of time. If a number of N of person is gathered in a shop during its promotion time, we call this a *gathering event*. Each gathering event exists in a time interval which is delimited by the start time of the gathering event (the number of gathered people is more than N for the first time), and the end time of the gathering event (the number of gathered people is less than N for the first time).

There is a sensing system to collect and analyze the behaviors of customers in the group consumption. There is no central server in the system. The whole region is spitted into several small sub-regions, each of which has a coordinator. Such coordinator can be a fixed powerful device if possible, or a dynamically changed one when each device takes a totally equal role.

The temporal relations among events in this system can be based on *physical clock* or *logic clock*. For the physical clock, all the clocks in different devices are synchronized through proper way and guarantee no obvious time drift. The temporal sequence of events can be determined by measuring the timestamps of the synchronized clocks. For the logic time, there is no such synchronized clock in the system. The sequence of events can only be determined by the exchange of messages among objects [9]. Fig. 2 show the difference between physical time and logical time. Based on physical time, the sequence of e^1 and e^4 can be determined by their timestamps directly. However, based on logic time, their sequence can only be determined by the message passing between P_1 and P_2. Since there is a message from e^1 of P_1 to e^4 of P_2, it can be concluded that e^1 happens before e^4. Similarly, e^2 happens

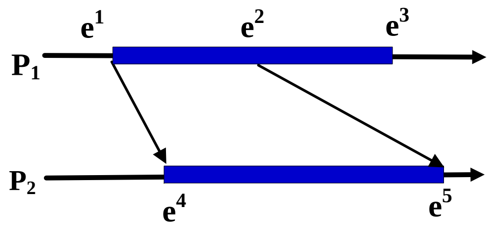

Fig. 2. Temporal relations

before e^5.

Let us define the temporal relations under asynchronous communications more formally. Suppose that there is a number of n devices P_i $(i = 1...n)$ and each device P_i records its alternate local states and events $s_i^0, e_i^0, s_i^1, e_i^1, ..., e_i^j, s_i^j$ [10], where s_i^k denotes the kth local state and e_i^k denotes the kth event, at the process i.

We call that state s_a *happen before* state s_b, denoted by $s_a \rightarrow s_b$, if

1) s_a is a state before s_b in the same device.

2) the event just after state s_a sends a message and the event just before state s_b receives that message.

3) there is a state s_c such that $s_a \rightarrow s_c$ and $s_c \rightarrow s_b$ [11].

If s_a does not happen before s_b and s_b does not happen before s_a, we call that s_a is *concurrent with* s_b, denoted by $s_a \mid\mid s_b$.

III. Data Collection Algorithms

In this section, we illustrate the algorithms to collect data for mobile group consumption in an asynchronous distributed way.

A. Distributed Data Collection Solution

We propose a three-layer solution for distributed data collection of mobile consumption. As shown in Fig. 3, the first layer is the procession of a local sub-region, the second layer is the procession of consecutive sub-regions, and the third layer is the procession of multiple sub-regions or even all the regions. In order to achieve the distributed procession, we split the whole target region into several sub-regions. In each region, the data collection requirements are handled by a local coordinator or fully distributed communications among devices. Most of data collection requirements can be handled locally in the first layer. Due to the boundary constraint, data procession may not be optimal or even correct in some cases. For example, the users may need to collect the data in a $10m^2$ area centered in a special location A. It is fine when A is a inner location of a sub-region since the area considered is included in the sub-region, while it is difficult when A near the boundaries of a sub-region since the considered area may span over two or more consecutive sub-regions. A simple coordination among consecutive regions in the second layer can be used for this procession. For other data collection requirements that need multiple sub-regions, we need the third layer to coordinate.

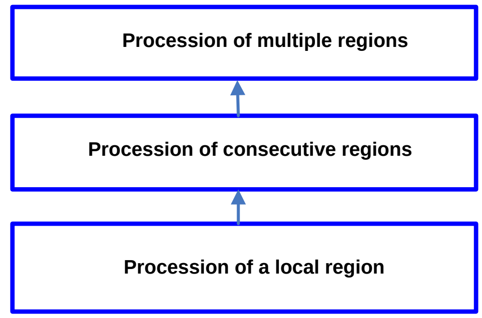

Fig. 3. A three-layer solution framework

B. Asynchronous Distributed Data Collection Approach

We further propose the Asynchronous Distributed Data Collection Approach (ADDC) which can determine the order of different events based on asynchronous communications. We do not assume a central server or synchronized clock in the system. All the orders of the different events are determined by messages exchanged among devices. More specifically, the orders are based on vector clocks [12] maintained in the system.

We illustrate the approach using a typical example in the analysis of mobile consumption, the casuality analysis. It is briefly described as follows: A shop's promotion for group consumption is considered effective if a certain number of N people are attracted to enter the shop. Such gathering can be regarded as an gathering event with a certain time interval. Two gathering events are considered to have causality relation if they have temporal sequence, e.g. a gathering event happen before another gathering event. In this section, we aim to find all the pairs of events

In a classical system with synchronized clocks, such relation can be detected based on the value of clocks directly, while in asynchronous distributed system, such a clock is not available. We illustrate our approach for this in Algorithm 1 and Algorithm 2.

The approach specifies the actions of two kinds of entities in the system, the consumers and the shops. The consumers's actions are performed by its mobile devices and the actions of shops are performed by its computer.

Algorithm 1 specifies three kinds of actions that are performed by consumers. The first action triggers the detection of the whole event pattern. When a consumer is in the affecting area of the shop and the shop is just broadcasting the promotion information, the consumer can decide, by itself or by the group, whether to enter into the shop for checking the detailed information (line 1). In this case, if the consumer decides to accept the promotion and enters into the shop (line 2), its smart mobile will update his/her location to be the shop (line 3) and send a message to the entered shop for notifying such action (line 4). Similarity, when the consumer walks out of the scope of the shop (line 6), corresponding smart mobile will update the location and notify the shop (line 7-8). The enter message and exit message can trigger the shop to detect

Algorithm 1: Asynchronous Distributed Data Collection Approach (Consumer)

1 **When** PROMOTION(*shopID*) is received **do**
2 **if** *accpeted* **then**
3 *location* = *shopID*
4 send MSG_ENTER(*ID*) to *shopID*
5 **end**
6 **When** walk out of the scope of *shopID* **do**
7 *location* = ∅
8 send MSG_OUT(*ID*) to *shopID*
9 **When** START_TRANSITION(*initiator*, *sender*, *depth*, *recClock*) or END_TRANSITION(*initiator*, *sender*, *depth*, *recClock*) is received **do**
10 *clock*[*ID*]=*clock*[*ID*]+1
11 *clock*[*1..n*]=max(*clock*[*1..n*], *recClock*[*1..n*])
12 send corresponding START_TRANSITION(*initiator*, *sender*, *depth*, clock) or END_TRANSITION to the nearest shop

the start time and end time of gathering events. The third action performed by the consumers is the message relay, which is to forward the start message or end message of gathering event from the initiator to others to build the logic "happen before" relations. The orders of different events are based on vector clocks used in asynchronous system. So the vector clock in the devices are firstly updated (line 10-11) and then corresponding messages are sent to the nearest shops (line 12).

Algorithm 2 specifies the action performed by the shops. They include the detection of the start time and the end time of a gathering event, and building a routing tree whose root can finally store the detection result of a event pattern. When a enter message of consumers is received (line 1), the recorded number of consumers in the shop and the detailed ID of consumers are updated firstly (line 2-3). If the gathered number of consumers is greater than the threshold *gatherThrehold*, a new gathering event is detected (line 4), and a relay message is sent to the persons around the shop (line 6). The persons received this message will further walk to other places and/or send the information to their neighbors, hence the "happen before" relation is built. Similarly, when an exit message is received (line 13), the recorded number of consumers and detailed ID of consumers are updated (line 14-15). If the number of person gathered is less than *gatherThrehold*, the end time of existing gathering event is detected (line 16), and a message is sent to persons around the shop (line 18). The vector clock is updated locally in a proper way (line 5, line 17). Since we aim to detect a pari of events that may have casuality relation, when an end time of a gathering event happens before the start time of another gathering event, a successful detection is achieved and the result will be transmitted the root of routing tree (line 7-11). Since the consumers are moving all the time, we build a routing tree based on fixed shops for message exchange. When a relay message is received, the shop set its parent in the routing tree as the sending node (line 22-23). The initiator of the detection of the event pattern and the ID of the event pattern are also recorded (line 23-24). The routing tree is adaptively changed with the progress of event detection. In order to achieve a routing tree of small depth, when the depth of routing tree is more than a threshold *depthThreshold*, current shop is automatical changed into the root node and a

Algorithm 2: Asynchronous Distributed Data Collection Approach (Shop)

1 **When** MSG_ENTER(*personID*) is received **do**
2 *cardinality* = *cardinality* + 1
3 *persons* = *persons* $\bigcup$ {*personID*}
4 **if** *cardinality* $>$ *gatherThrehold* **then**
5 *clock*[*ID*]=*clock*[*ID*]+1
6 broadcast START_TRANSITION(*ID*, *ID*,1,*clock*) to neighboring persons
7 **foreach** *eventID* $\in$ *eventList* **do**
8 **if** *ori*[*eventID*] $\neq$ ∅ **then**
9 send MSG_SUCCESS(*ID*, *clock*) to *parent*[*eventID*]
10 **end**
11 **endfch**
12 **end**
13 **When** MSG_OUT(*personID*) is received **do**
14 *cardinality* = *cardinality* - 1
15 *persons* = *persons* $-$ {*personID*}
16 **if** *cardinality* $<$ gatherThrehold **then**
17 *clock*[*ID*]=*clock*[*ID*]+1
18 broadcast END_TRANSITION(*ID*, *ID*,1,*clock*) to neighboring persons
19 **end**
20 **When** START_TRANSITION or END_TRANSITION(*initiator*, *sender*, *eventID*, *depth*) is received **do**
21 **if** *parent*[*eventID*]=∅ **then**
22 *parent*[*eventID*] = *sender*
23 *ori*[*eventID*] = *initiator*
24 $RS = RS \bigcup$ {*eventID*}
25 *depth*[*eventID*] = *depth+1*
26 **if** *depth*[*eventID*] $>$ *depthThreshold* **then**
27 send REVERSE(*ID*, *sender*, *eventID*) to *sender*
28 **end**
29 **end**
30 **When** REVERSE(*oriID*, *sender*, *eventID*) is received **do**
31 *priviousParent* = *parent*[*eventID*]
32 *parent*[*eventID*] = *sender*
33 send REVERSE(*oriID*, *ID*, *eventID*) to *priviousParent*

reverse information is sent to its previous nodes (line 26-28). When the reverse information is received by a shop, it updates its parent to the sender of the message and further forward the message to its antecessors to also change the parent (line 30-33).

C. Discussion

The casuality investigated in the previous sub-sections are based on happen before relation. This is a little different from the cases based on physical time that can specify a time duration between two gathering events. Such quantity measurement is difficult in logic time, because a synchronized clock is not available. If the exchange of messages among devices are sufficiently frequent, we can use the number of messages from one event to another event to estimate the time duration between them.

Another concern is the complexity of temporal relations. There are 29 kinds of relations between two time intervals (dense time mode) [9]. Also, since the temporal relations often cannot be determined certainly. Occurrence probability [13] can be used for the specification of this scenario. For the two cases above, we can easily modify the algorithms to distinguish these conditions to meet the users' requirements.

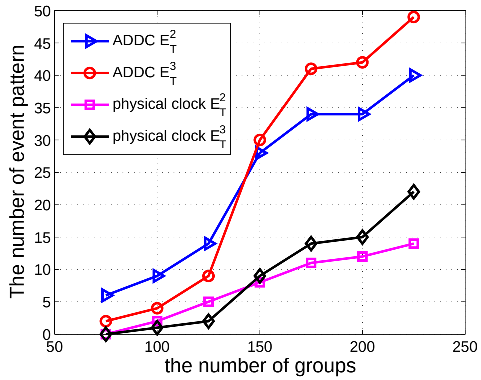

Fig. 4. The number of event pattern vs. the number of groups

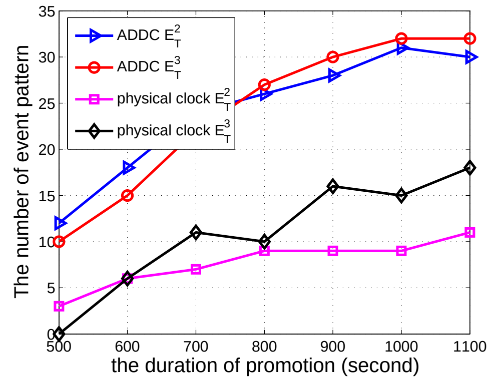

Fig. 5. The number of event pattern vs. the duration of promotion

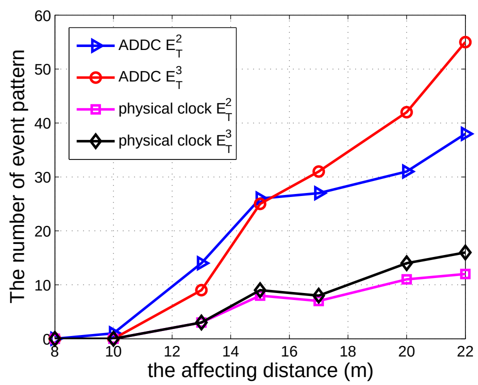

Fig. 6. The number of event pattern vs. affecting distance

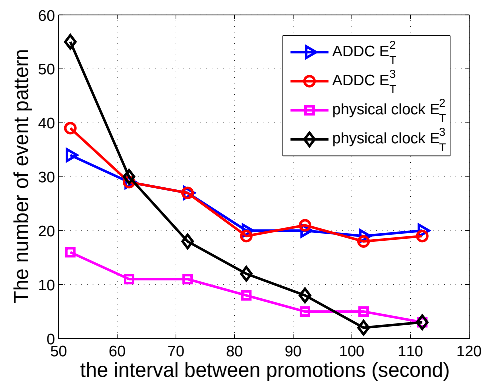

Fig. 7. The number of event pattern vs. the interval between promotions

IV. Performance Evaluation

We conduct simulations to validate the effectiveness of our proposed approach. We compare the results of our approach ADDC with those under physical clock. The comparing example we used are the detection of the casuality of two events and three events. The number of event patterns satisfying the causality of two events and three events are denoted by E_T^2 and E_T^3, respectively. For the detection under physical clock, we specify the duration between two events T to be 500 seconds, while for the detection under logic clock, we need two events to have happen before relation. E_T^2's time duration is measured by the start time of an event to the start time of the second event, and E_T^3's time duration is measured by the end time of an event to the start time of the second event.

A. Simulation Setup

In the following several sub-sections, we simulate a shopping mall of 10m $\times$ 10m where there are 12 shops in it. Several groups of people wandering in it. In each second, each shop begins promotion in a probability of $p_o = 0.00307$ to attract the people and it lasts a duration of s. The promotion can attract the people in the distance of less than $d < 15$ m. When there is a promotion, the group can be attracted in a specified probability of $p_m = 0.49$. We vary different parameters in the simulations to check the performance of our approach. 30 runs simulations are repeated to get each data point of Fig. 4 - **??**.

B. The Number of Event Patterns Detected

We first vary the number of groups in shopping mall to check the detected event pattern of E_T^2 and E_T^3. The result is shown in Fig. 4. It shows that generally that the number of E_T^2 and E_T^3 increase when the number of group increase. When the number of groups is 75, both the number of E_T^2 and E_T^3 detected by ADDC or physical clock are small, less than 7. The number of E_T^2 detected by ADDC is increased to around 40, when the number of groups reaches 225. The number of E_T^3 detected by ADDC is also increased when the number of groups increase, but with a larger increasing speed. Its number is smaller than that of E_T^2 when the number of group is less than 150. This is because that the constraint of E_T^3 is more strict than that of E_T^2. And its number becomes larger than that of E_T^2 when the number of group is more than 150. This is because the combination of E_T^3 is larger than E_T^2 when the number of event is larger. The trend of results of ADDC is similar with those under physical clock. However, the results of ADDC is larger than the those under physical clock, since we only specify the happen before constraint while the detection under physical clock need a gap of 500

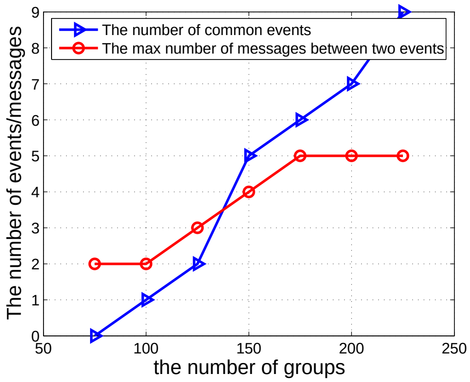

Fig. 8. The number of common event pattern / the max number of messages between two events vs. the number of groups

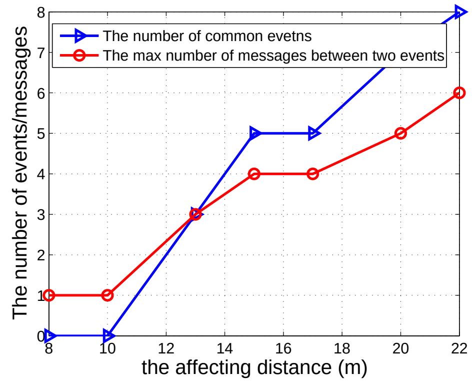

Fig. 9. The number of common event pattern / the max number of messages between two events vs. the affecting distance

second. When the gap is decreased, the difference between the two approaches become a continuous smaller.

We then change the duration of promotion activity from 500 to 1100 seconds and the result is shown in Fig. 5. The number of E_T^2 and E_T^3 detected by ADDC is quite similar in this case since we only use the start time of each event to determine "happen before" relation. Such relation is not sensitive to the duration of promotion activities. Differently, the results under physical clocks are more related to the duration of promotion activity with regrading to E_T^3, which is determined by the end time of the first event to the start time of the following event. The results of ADDC is still larger than those under physical clocks, since its constraint is less strict.

After that, we change the affecting distance of marking activity from 8 to 22 and the result is shown in Fig. 6. It is show that the increase distance leads to the increase of the number of E_T^2 and E_T^3. The number of E_T^3 is increased more fast than that of E_T^2. The result of ADDC is like that under physical clock but with larger values. The result is consistent with those under different number of groups, and different durations of activities.

Finally, as shown in Fig. 7, we change the interval of two promotions to check the number of event patterns. It can be seen that when such interval increases, both the number of event pattern detected by ADDC and physical clock decrease. This is because the increased interval leads to the decreased number of gathering events, hence the number of event pattern decrease. The number of event pattern detected by ADDC is still larger than that detected by physical clock, which is consistent with previous results.

C. Common Events and Passing Messages

We further investigate the number of common event patterns detected by ADDC and the approach under physical clock. The result is shown in Fig. 8 and Fig. 9. As shown in Fig. 8, the common events of the two approaches is growing when the number of group increases. The difference of those is due to the several reasons. First, the gap between two events under physical time cannot be described accurately using asynchronous communications, hence some event patterns can be detected by ADDC but not under physical clock. Second, the delay of messaging passing lead to that some event patterns can be detected under physical clock but not by ADDC.

Fig. 8 also show the maximum number of messages when detect E_T^2. It can seen that the number of it is increased constantly when the number of groups is less than 175 and quite stable after 175. This shows that the number of groups is sufficient large and make little affct to the detection.

Fig. 9 show the common events and the maximum number of messages between two events when adjusting the affecting distance. The results are similar with the results in Fig. 8. When the affecting distance increases, the common events increase, and the the maximum number of messages between two events also increase. This is consistent with the uncertainty of asynchronous communications.

V. Related Works

In the literature, there are some works related to the data collection of mobile group consumption.

In work [6], Namiot et al. utilize WIFI to collect the locations, access duration and the number of access of users. Yang et al. collect the browsed websites of the users and its location to recommend new retailers to the users [8]. Kanda et al. utilize laser rangefinder and particle filter to analyze the realtime trajectory of users, and obtain the time and venue that the consumption is the most likely to happen [14]. In [15], Andrews et al. investigate the crowdedness of environment and how it affect the mobile comsumption. In [16], Guo et al. investigate how to use RFID technology to promote mobile-commerce. In [7], Lu et al. collect the mobile users' movements and purchase transactions information for prediction. The system "me-Commerce" [17] collect the context-information for providing dedicated, targeted and personalised information access to users. Gu et al. studied more complex activities problem using sensors [18].

All these works are based on centralized server and synchronized clock, which suffers from performance bottleneck, one-point failure, overhead of powerful server and synchronization, etc. and sometime even infeasible due to hardware constraint or privacy concerns.

In the field of distributed computing, there are works about asynchronous distributed data collection. Due to the uncertainty of asynchronous distributed computing, *definitely* and *possibly* modalities are introduced in [19] for detecting the events. The lattice is invented as a tool for detecting generic events, called *relational predicates* [19], [20]. As a special class of predicates, *conjunctive predicates*, specified by a conjunctive expression of local states, are also investigated [10].In [13], Zhu et al. use a occurrence probability to refine the *possibly* modality to provide more detailed information and can support the detection of multiple occurrence of events. In this paper, we follow the basic idea of asynchronous distributed computing and adjust it for mobile group consumption.

VI. Conclusion

In this paper, we proposed an asynchronous distributed approach to collect the data of mobile consumption. Considering the large number of consumers and constant moving of them, our approach is scalable and efficient since it does not need a central server or synchronous clock. We first build the system model of data collection of mobile consumption in a distributed way and without synchronized clock. The we propose a three-layer solution framework. Three kinds of processing including the procession in a local sub-region, in consecutive sub-regions, and in several sub-regions are considered. After that, we demonstrate how to analyze the casuality of two events happened in the system based on message exchange. We conduct extensive simulation to validate the proposed approach. The results show that our approach is effective for data collection of mobile group consumption.

In the future, we plan to further investigate the data collection for more complex casuality analysis, including the casuality among three or more gathering events and the complex temporal relations among several time interval of gathering events. We also plan to studies the detailed algorithms to estimate the duration in the physical time using the number of messages exchanged.

Acknowledgment

This research is supported in part by the Fundamental Research Funds for the Central Universities of China No. 216-410500026, Nature Science Foundation of Hubei, China No. 2015CFB340, and Outstanding Young Academic Talents Start-up Funds of Wuhan University No. 216-410100004.

References

[1] M. Moussaid, N. Perozo, S. Garnier, D. Helbing, and G. Theraulaz, "The walking behaviour of pedestrian social groups and its impact on crowd dynamics," *PLoS ONE*, vol. 5, no. 4, doi:10.1371/journal.pone .0010047, pp. 521–544, 2010.

[2] InMobi Insights Team, "Global mobile media consumption," 2013. [Online]. Available: http://info.inmobi.com/rs/inmobi/images/Global_Mobile_Media_Consumption_Wave_2_Whitepaper.pdf

[3] eMarketer, "Tablets, smartphones drive mobile commerce to record heights," 2013. [Online]. Available: http://www.emarketer.com

[4] China Internet Network Information Center (CNNIC), "The thirty-fifth statistical report on the development of China's internet in 2015," 2015. [Online]. Available: http://www.cnnic.net.cn

[5] eMarketer, "Mobile ad spend in China hits $7 billion this year," 2014. [Online]. Available: http://www.emarketer.com

[6] D. Namiot and M. Schneps-Schneppe, "About location-aware mobile messages: Expert system based on wifi spots," in *IEEE INternational Conference on Next Generation Mobile Applications, Services and Technologies*, 2011, pp. 48–53.

[7] E.-C. Lu, W.-C. Lee, and V. Tseng, "A framework for personal mobile commerce pattern mining and prediction," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 5, pp. 769–782, May 2012.

[8] W. S. Yang, H. C. Cheng, and J. B. Dia, "A location-aware recommender system for mobile shopping environments," *Expert Systems with Applications*, vol. 34, no. 1, pp. 437–455, 2008.

[9] A. D. Kshemkalyani, "A fine-grained modality classification for global predicates," *IEEE Trans. on Parallel and Distributed Systems*, vol. 14, no. 8, pp. 807–816, 2003.

[10] V. K. Garg and B. Waldecker, "Detection of weak unstable predicates in distributed programs," *IEEE Trans. on Parallel and Distributed Systems*, vol. 5, no. 3, pp. 299–307, 1994.

[11] L. Lamport, "Time, clocks, and the ordering of events in a distributed system," *Communications of the ACM*, vol. 21, no. 7, pp. 558–565, 1978.

[12] C. Fidge, "Logical time in distributed computing systems," *IEEE Computer*, vol. 24, no. 8, pp. 28–33, 1991.

[13] W. Zhu, J. Cao, and M. Raynal, "Predicate detection in asynchronous distributed systems: A probabilistic approach," *IEEE Transactions on Computers*, vol. PP, no. 99, pp. 1–1, 2015.

[14] T. Kanda, D. F. Glas, M. Shiomi, H. Ishiguro, and N. Hagita, "Who will be the customer?: a social robot that anticipates people's behavior from their trajectories." *in Proc. of UbiComp*, pp. 380–389, 2008.

[15] M. Andrews, X. Luo, Z. Fang, and A. Ghose, "Mobile ad effectiveness: Hyper-contextual targeting with crowdedness," *Marketing Science*, 2014.

[16] W. Guo, "Research on mobile commerce based on rfid," in *IEEE International Conference on Service Operations and Logistics, and Informatics(SOLI)*, vol. 1, Oct 2008, pp. 853–857.

[17] A. Komninos, P. Barrie, J. Newman, and S. Landsburgh, "me-commerce: An infrastructure for personal predictive mobile commerce," in *Proc. of International Conference on Mobile Business (ICMB)*, June 2006, pp. 39–39.

[18] T. Gu, L. Wang, Z. Wu, X. Tao, and J. Lu, "A pattern mining approach to sensor-based human activity recognition," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 9, pp. 1359–1372, Sept 2011.

[19] R. Cooper and K. Marzullo, "Consistent detection of global predicates," in *Proc. of the ACM/OCR Workshop on Parallel and Distributed Debugging*, 1991, pp. 163–173.

[20] K. Marzullo and G. Neiger, "Detection of global state predicates," in *Proc. of the 5th Workshop on Distributed Algorithms*, 1991, pp. 254–272.