MDPI

Article

An Asynchronous Distributed Data Collection Approach for Mobile Group Consumption

Weiping Zhu [1,2,*], Weiran Chen [3], Zejie Hu [1], Zuoyou Li [1], Yue Liang [1] and Jiaojiao Chen [1]

[1] International School of Software, Wuhan University, Wuhan, 430079, P. R. China
[2] State Key Lab. for Novel Software Technology, Nanjing University, Nanjing, 210046, P. R. China
[3] Economics and Management School, Wuhan University, Wuhan, 430079, P. R. China
* Correspondence: wpzhu@whu.edu.cn; Tel.: +86-27-68778712

Academic Editor: name
Version January 23, 2016 submitted to Entropy; Typeset by LaTeX using class file mdpi.cls

Abstract: Mobile group consumption refers to the consumption performed by a group of people such as couples, colleagues, friends, etc. based on mobile communications. It is different from the consumption involved only individuals due to complex relations of group members. Existing data collection systems for mobile group consumption are centralized, which suffers from performance bottleneck, one-point failure, and the overhead of server. Moreover, they are based on synchronized clock, which is often impossible due to hardware constraint, privacy concerns, or synchronization cost. In this paper, we propose an asynchronous distributed approach to collect data of mobile group consumption for the first time. We formally build the system model of it based on asynchronous distributed communication. Then we design a simulation system for that and propose a three-layer solution framework. After that, we describe how to detect the causality relation of two/three gathering events happened in the system based on the collected data. Various definitions of causality relation based on asynchronous distributed communication are supported. Extensive simulation results show that the proposed approach is effective for data collection of mobile group consumption.

Keywords: Asynchronous; Distributed; Data Collection; Mobile Group Consumption.

1. Introduction

Group consumption refers to the consumption activities performed by a group of people such as couples, families, colleagues, friends, etc. This kind of consumption is different from the consumption involved only individuals, because the decision of a group is made by the interaction and negotiation of group members. Group consumption is not uncommon in our daily life. In fact, when people are in public places, 70 percent of their time are spent with other persons [1]. In many cases, they perform group consumption.

In recent years, a growing number of people utilize mobile phones to perform group consumption. They use mobile phones to scan, search and receive information from retails, and order various services and goods. According to a survey of InMobi Insights Team, about 46% people had made purchases using their mobile devices and 80% people plan to do it in the next 12 months [2]. We call the group consumption based on mobile devices *mobile group consumption*. An example of it can be seen in Figure 1. Having observed this trend, many companies put increasing marketing effort to mobile group consumption. In 2014, the global advertisement for that reaches 32.7 billion US dollars, which is about one forth of the total network advertisement [3].

To analyze mobile group consumption, an effective data collection system should be built first. Typical data needed to be collected include the trajectories and actions of group members in a time

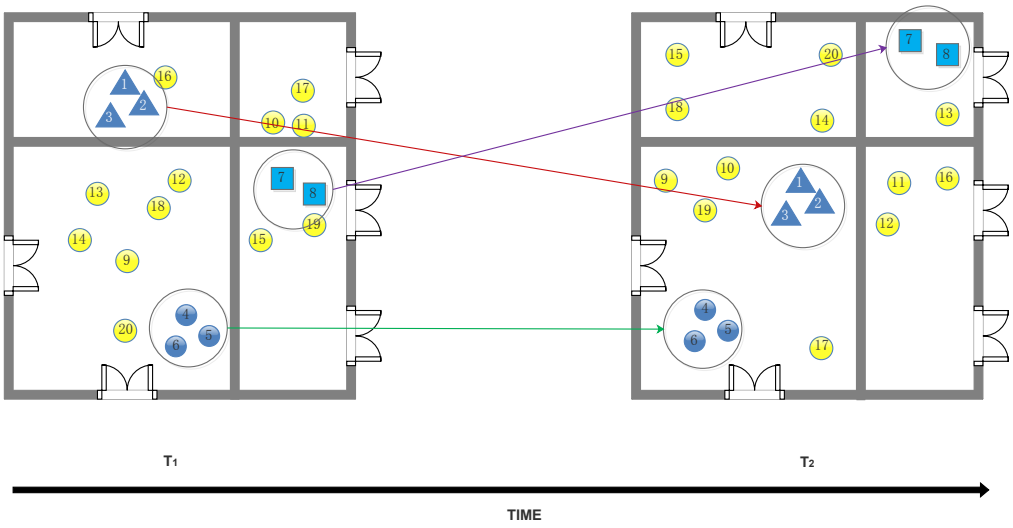

Figure 1. A illustration of mobile group consumption

duration. Various technologies can be used for such purpose. For example, WIFI signals from access points to mobile phones of consumers can be collected. The RFID tags attached to mobile phone also can be read for analysis.

Existing data collection systems [4–6] that can be used for mobile group consumption are based on centralized procession. A central server is needed to gather information from consumers. These systems encounter challenges as follows: 1) In many scenarios, it is difficult to find a proper powerful server. 2) Even there is a server available, it suffers from serious one-point failure and computation bottleneck, especially in the scenarios including a large number of shops and constantly moving consumers. Moreover, existing work assume the existence of a synchronized clock, which incurs high overhead when the consumers is in a 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 highly demanded to solve 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 and asynchronous message communications. We first build the system model of asynchronous distributed data collection. Based on it, we design a simulation system. We then propose a three-layer solution for generic distributed procession, including the procession for local sub-regions, consecutive sub-regions, and multiple sub-regions. After that, we propose an approach to detect the causality of two specified marketing activities in group consumption. Extensive simulations are carried out to evaluate our approach. The results show that the approach can effectively support the data collection of mobile group consumption. In summary, this paper makes the following contributions:

- We built the system model of mobile group consumption based on asynchronous message passing. It is not based on centralized server or synchronized clock. A simulation system based on it is also designed and developed.
- We proposed a three-layer mechanism to collect data for mobile group consumption in an asynchronous distributed way. The data collection is firstly handled locally, and then coordinated in convective or more regions if the data collection spans a wide area.
- We conduct extensive simulations to validate the proposed approach. The results show that the proposed algorithm is quite effective.

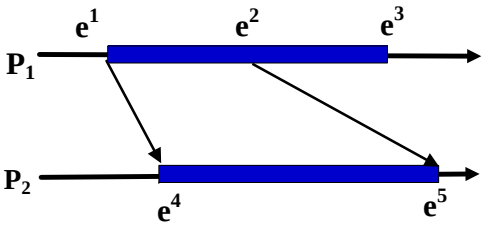

Figure 2. Temporal relations under different clocks

The rest of the paper is organized as follows: Section 2 builds the system model of asynchronous distributed data collection for mobile group consumption. Section 3 illustrate the simulation system based on it. The detailed data collection solution is described in Section 4. The simulation results are reported in Section 5. Section 6 reviews the related works and finally Section 7 concludes the paper.

This paper is based on our previous conference paper [7] published in IIKI2015. In this version, we add the design of our simulation system, extend the approach to support more complex temporal relations for casuality detection, support the message passing with arbitrary speed, and also provide more analysis details and simulation results.

2. System Model

We assume that mobile group consumption happens in a place (e.g. a shopping mall) where shops broadcast various promotion information from time to time. Groups of people wander around the place for shopping and may be attracted by one or more promotions. Each group includes one or more members. Each promotion is assumed lasting for a certain period of time. People can only receive the promotion information from a shop if they are in a specified area of the shop, called *affecting area*. If N persons are 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 mobile group consumption. There is no central server in the system.

Temporal relations 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 ways. The temporal sequence of events can be determined by measuring timestamps of the synchronized clocks. For the logic time, there is no synchronized clock in the system. The sequence of events can only be determined by the exchange of messages among devices [8]. Figure 2 shows the difference of temporal relations based on physical time and logical time where P_1 and P_2 are devices and $e^1, e^2...$ are events happened in corresponding devices. According to physical time, the sequence of e^1 and e^4 can be determined by their timestamps directly. However, according to 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 before e^5. If the message passing among devices are with finite and arbitrary time delay, we call it *asynchronous communication*.

Let us define the temporal relations under asynchronous communications more formally. Suppose that there are n devices P_i ($i = 1...n$) and each device P_i records its alternate states and events $s_i^0, e_i^0, s_i^1, e_i^1, ..., e_i^j, s_i^j$ [9], where s_i^k denotes the kth state and e_i^k denotes the kth event that change the local state from s_i^{k-1} to s_i^k, at the device P_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

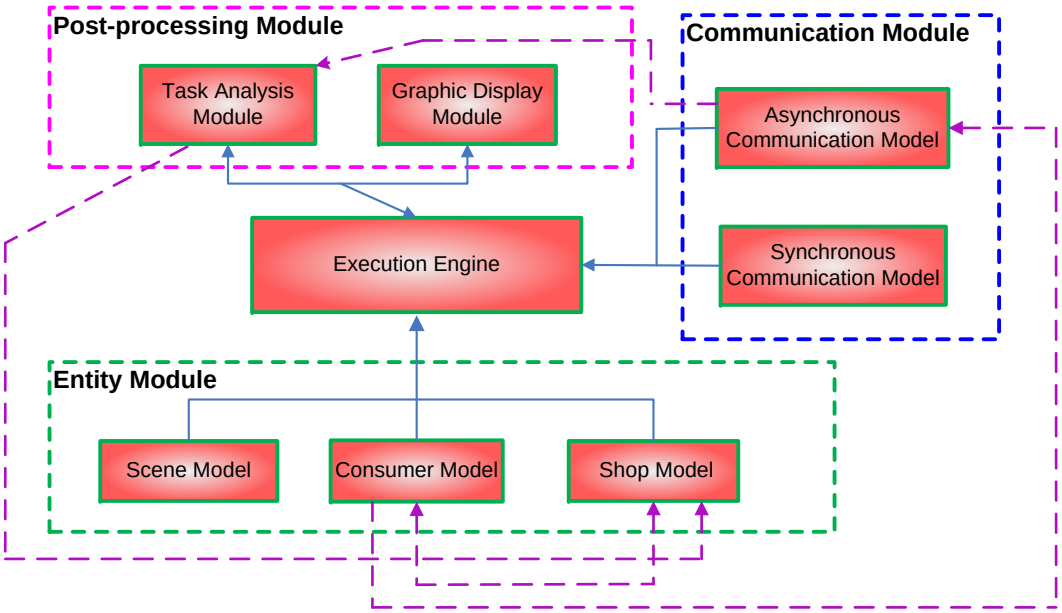

Figure 3. The architecture of simulation system for mobile group consumption

message.

3) there is a state s_c such that $s_a \rightarrow s_c$ and $s_c \rightarrow s_b$ [10].

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 $s_a \mid\mid s_b$.

3. Simulation System

In order to investigate the problem efficiently, we develop a simulation system called Group Consumption Simulation system (GCS). The top-level architecture of it can be seen in Figure 3. This system includes 4 modules, execution engine, entity module, communication module, and post-processing module.

As the core of the system, execution engine controls the whole simulation process and triggers the execution of other modules. Execution engine configures the simulation environment based on entity module, performs the simulation based on communication module, and finally conducts analysis and graphic display based on post-processing module.

Entity module includes three major models, scene model, consumer model, and shop model. Scene model defines the area of shopping mall, the layout of shops in it, the composition of each group, and the initial location and moving direction of each person. Consumer model defines the role of each member in a group (e.g. group leader, normal member), the actions reacting to a shop's promotion, the motion pattern in the shopping mall in terms of groups and individuals. It is noted that the motion pattern is different in the spare space and after attracting into a shop. Finally, shop model defines the promotion time, promotion duration, and promotion frequency of each shop.

Communication module includes asynchronous communication model and synchronous communication model, which are based on message passing and global clock, respectively. The asynchronous communication model follows our specification in Section 2, which is the major concern of this paper. The synchronous communication model assumes that each entity has a synchronized clock and can be used as the ground truth for the comparison with message passing.

Post-processing module includes task analysis module and graphic display module. Task analysis module allows the users to define their interested tasks. For example, the causality of different promotions, the impact of a promotion in an area, etc. Graphic display module is used to show the results of analysis in a graphic way.

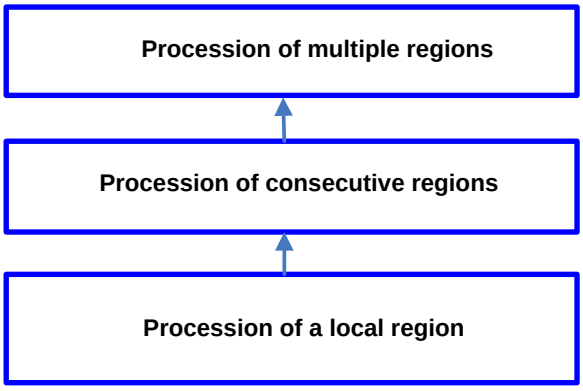

Figure 4. A three-layer distributed data collection framework

It is noted that some models may affect each other. We illustrate them using dashed lines in Figure 3. The consumer's motion may affect the shop's promotion strategy, and vice versa. Therefore there are bidirectional dashed lines between consumer model and shop model. Shop model may also be affected by the results of task analysis module. For asynchronous communication model, the parameters of message passing may be affected by the consumer's motion on one hand, and affect the results of task analysis module on the other hand. The impact of different models are left for future work and are not implemented in current system.

4. Data Collection Approach

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

4.1. Distributed Data Collection

We propose a three-layer solution for distributed data collection of mobile group consumption. The whole target region is splitted into several sub-regions. As shown in Figure 4, the first layer handles the procession of a sub-region, the second layer handles the procession of consecutive sub-regions, and the third layer handles the procession of multiple sub-regions or even all the region. Most of data collection requirements can be handled locally in the first layer. However, local procession sometimes may not be optimal or correct. For example, the users may need to collect the data in a $10m^2$ area centered in A. Local procession is fine when A is in the inner of a sub-region since the considered area is included in the sub-region, while it may be wrong when A nears the boundaries of a sub-region since the considered area may span over two or more consecutive sub-regions. Coordination among consecutive regions can be used in the second layer. For other data collection requirements that involve multiple sub-regions, we need the third layer to coordinate them.

4.2. Asynchronous Distributed Data Collection

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 events are determined by messages exchanged among devices. More specifically, the orders are based on vector clocks [11] maintained in the system.

We explain the approach using a typical example in mobile group consumption, the analysis of causality relation. It is briefly described as follows: A shop's promotion is regarded to be effective if N people are attracted to enter the shop. Such gathering is called an gathering event. Two gathering

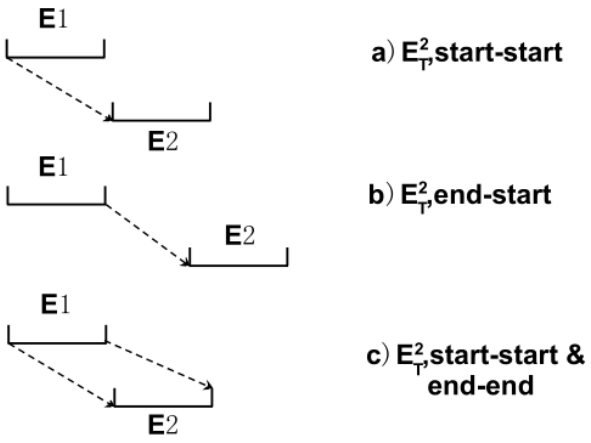

Figure 5. Casuality detection based on different definitions

Algorithm 1: Asynchronous Distributed Data Collection Approach (Consumer)

When PROMOTION(*shopID*) is received **do**
if *accepted* **then**
    location = *shopID*
    send MSG_ENTER(*ID*) to *shopID*
end
When walk out of the scope of *shopID* **do**
location = *null*
send MSG_OUT(*ID*) to *shopID*
When START(*initiator*, *sender*, *eventID*, *depth*, *recClock*) or END(*initiator*, *sender*, *eventID*, *depth*, *recClock*) is received **do**
clock[*ID*]=*clock*[*ID*]+1
clock[*1..n*]=max(*clock*[*1..n*], *recClock*[*1..n*])
send corresponding START(*initiator*, *sender*, *eventID*, *depth*, *clock*) or END(*initiator*, *sender*, *eventID*,*depth*, *clock*) to neighboring shops or people

events are considered to have causality relation if they have temporal sequences, e.g. a gathering event happens before another one. In this section, we aim to find all such pairs of events.

In a classical system with synchronized clocks, causality relation can be detected based on the value of clocks directly, while in an asynchronous distributed system, such clocks are not available. We need to utilize message passing to detect causality relation. The detection of casuality relation can be based on different definitions. We describe the most common ones in Figure 5. In the first case, the casuality is based on the start time of the first gathering event and the start time of the second gathering event. We call it $E^2(start - start)$ where 2 refers to the two events considered and *start-start* refers to the delimitation time of two gathering events. In the second case, the casuality is based on the end time of the first gathering event and the start time of the second gathering event. We call it $E^2(end - start)$. In the third case, the situation is more complex. The casuality is based on that the start time of the first gathering event happens before the start time of the second gathering event, and the end time of the first gathering event happens before the end time of the second gathering event. We call it $E^2(start - start \ \& \ end - end)$. The dashed lines in Figure 5 denote the message passing between the two events hence their happen before relations. Our algorithm can support all these three definitions.

Table 1. Notations in the algorithms

Variable Name	Description
shopID	the ID of a shop
PROMOTION(*shopID*)	promotion information broadcast by *shopID*
accepted	the variable denoting whether a consumer accepts a promotion
location	current location of a consumer
initiator	the first shop sending the message
sender	last shop sending the message
eventID	the ID of an event
depth	the depth of *sender* in the routing tree
recClock	the vector clock attached to a message
clock	the vector clock maintained by a consumer or a shop
cardinality	the number of attracted consumers in a shop
persons	a collection recording the detailed consumer IDs in a shop
personID	the ID of a consumer
increaseID	a function to generate the ID of a new event
gatherThreshold	the threshold of gathered consumers to denote the occurrence of a gathering event
detFlag	the type of causality relation to be detected
eventList	a list of events received by current shop
oriStart[*eventID*]	the initiator of *eventID* after it starts as current shop knows
oriEnd[*eventID*]	the initiator of *eventID* after it ends as current shop knows
SS[*preID*]	a flag recording the detection result of the first part of SSEE
parent	the parent of current shop in the routing tree
children	the children of current shop in the routing tree
depth[*eventID*]	the parent of current shop in the routing tree regarding *eventID*
MSG_NTER(*ID*)	a message denoting the consumer *ID* enters current shop
MSG_OUT(*ID*)	a message denoting the consumer *ID* exits current shop
START	a messgage denoting the start of a gathering event
END	a message denoting the end of a gathering event
SUCCESS	a message denoting a successful detection of a causality relation
REVERSE	a message to notify the reverse of the routing tree
E	the specification of an event needed to be detected
coordinator[E]	denote whether current node is the coordinator of E

We illustrate our approach in Algorithm 1, Algorithm 2, Algorithm 3 and Algorithm 4. They specify the actions of two kinds of entities in the system, consumers and shops. The consumers's actions are performed by its mobile devices, and the actions of shops are performed by their computers. There exist interactions between consumers and consumers and between shops and consumers via message passing.

Algorithm 1 specifies the actions that are performed by consumers. The first action triggers the detection of events related to casuality relation. When a consumer is in the affecting area of a shop and the shop is just broadcasting promotion information, the consumer can decide, by itself or by the group, whether to enter into the shop for checking more details (line 1). If the consumer decides to accept the promotion and enters into the shop (line 2), its smart mobile updates his/her location to the shop (line 3) and sends 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 updates the location and notifies the shop (line 7-8). The enter message and exit message can trigger the shop to detect the start time and the end time of gathering events. The other action performed by consumers is message relay, which is to forward the start message or end message of gathering events from the initiator to others to build logic "happen before" relations. The orders of different events are based on vector clocks used in asynchronous system. So the vector clocks in the devices are firstly updated (line 10-11) and then corresponding messages are sent to the neighboring shops or people (line 12).

Algorithm 2: Asynchronous Distributed Data Collection Approach (Shop)

```
1  When MSG_ENTER(personID) is received do
2  cardinality = cardinality + 1
3  persons = persons ⋃ {personID}
4  if cardinality > gatherThreshold then
5  |  clock[ID]=clock[ID]+1
6  |  eventID=increaseID(eventID)
7  |  broadcast START(ID,ID,eventID,1,clock)
8  |  foreach preID ∈ eventList do
9  |  |  if detFlag="SS" and oriStart[preID] ≠ null then
10 |  |  |  send SUCCESS(preID, eventID, clock, "SS") to parent
11 |  |  else if detFlag="ES" and oriEnd[preID] ≠ null then
12 |  |  |  send SUCCESS(preID, eventID, clock, "ES") to parent
13 |  |  else if detFlag="SSEE" and oriStart[preID] ≠ null then
14 |  |  |  SS[preID]=true
15 |  |  end
16 |  endfch
17 end

18 When MSG_OUT(personID) is received do
19 cardinality = cardinality - 1
20 persons = persons − {personID}
21 if cardinality < gatherThrehold then
22 |  clock[ID]=clock[ID]+1
23 |  broadcast END(ID,ID,eventID,1,clock)
24 |  foreach preID ∈ eventList do
25 |  |  if detFlag="SSEE" and SS[preID] and oriEnd[preID] ≠ null then
26 |  |  |  send SUCCESS(preID, eventID, clock, "SSEE") to parent
27 |  |  end
28 |  endfch
29 end
```

Algorithm 2 and Algorithm 3 specifies the actions performed by shops. They include the detection of start time and end time of gathering events, the detection of casuality relation based on different definitions, and the building of routing tree whose root can finally store detection results.

We illustrate the details of it as follows. When an enter message of a consumer is received (line 1), the recorded number of consumers in the shop and the detailed IDs of the consumers are updated firstly (line 2-3). If the recorded number of consumers is greater than the threshold *gatherThrehold*, a new gathering event is detected (line 4). The local clock is increased by one (line 5) and the function *increaseID* is invoked to generate the event ID (line 6). The new event ID is a string including ID of the shop and also a local event sequence ID (e.g. shop1_event6). It is used to distinguish different events generated by shops. After that a relay message is sent to the persons around the shop (line 7). The persons will walk to other places and/or send the information to his/her neighbors, hence the "happen before" relation is built.

When a new gathering event is found, the algorithm further detects the casuality relation. We use SS to denote $E^2(start - start)$, ES to denote $E^2(end - start)$ and SSEE to denote the $E^2(start - start \ \& \ end - end)$. If current detection is based on SS, the start time of a previous gathering event (recorded in *oriStart*[*preEventID*]) lead to a successful detection (line 9-10). Similarly, the detection based on ES can be seen in line 11-12. If the detection is based on SSEE, only the first part can be complete, the start time of the first gathering event happens before the start time of the second gathering event. The result is recorded for further process (line 13-14).

Similarly, when an exit message of a consumer is received (line 18), the recorded number of consumers in the shop and the detailed IDs of the consumers are updated (line 19-20). If the recorded

Algorithm 3: Asynchronous Distributed Data Collection Approach (Shop)(cont)

```
30 When START(initiator, sender, eventID, depth,clock) is received do
31 if oriStart[eventID]=null then
32 |  oriStart[eventID] = initiator
33 |  eventList = eventList ⋃{eventID}
34 |  if parent=null then
35 |  |  parent = sender
36 |  |  depth[eventID] = depth+1
37 |  |  if depth[eventID] > depthThreshold then
38 |  |  |  send REVERSE(ID) to sender
39 |  |  end
40 |  end
41 end

42 When END(initiator, sender, eventID, depth, clock) is received do
43 if oriEnd[eventID]=null then
44 |  oriEnd[eventID] = initiator
45 |  eventList = eventList ⋃{eventID}
46 |  if parent=null then
47 |  |  parent = sender
48 |  |  depth[eventID] = depth+1
49 |  |  if depth[eventID] > depthThreshold then
50 |  |  |  send REVERSE(ID) to sender
51 |  |  end
52 |  end
53 end

54 When REVERSE(sender) is received do
55 priviousParent = parent
56 parent = sender
57 send REVERSE(ID) to priviousParent
```

number of consumers is less than *gatherThrehold*, the end time of existing gathering event is detected (line 21). The vector clock is updated locally (line 22) and a message is sent to persons around the shop (line 23). The causality detection based on SSEE can be complete here by checking the second part, the end time of the first gathering event happens before the end time of the second gathering event. If it is hold, the whole casuality relation is detected and a successful message is sent out (line 24-27).

When the message START and END are received via the forwarding of consumers. The shop records the initiator of the message and the event ID (line 32-33,44-45). Since the consumers are moving all the time, we build a routing tree based on fixed shops for message exchange. The detection results of casuality relation are transmitted along the routing tree. This routing tree is build as follows: When a relay message is received for the first time, the shop sets its parent in the routing tree as the sending node (line 34-35, 46-47). The depth of the node in the tree is also updated (line 36, 48). In order to achieve a routing tree of small depth, we design a depth-adaptive routing tree. When the depth of the routing tree is more than a threshold *depthThreshold*, current shop is automatically changed into the root and a reverse information is sent to its previous ancestors (line 37-39, 49-51). When the reverse information is received by a shop, it updates its parent to the sender of the message and further forwards the message to its antecessors (line 54-57). The operations to clock when receiving the message START and END are similar with line 10-11 in Algorithm 1 and are omitted here.

Finally, Algorithm 4 illustrates how to coordinate the processes of sub-regions to handle a user's detection requirement. Usually, a user's detection requirement involves several sub-regions. We first calculate such sub-regions and put them into a set S (line 2). The event to be detected and

Algorithm 4: Multiple sub-region coordination algorithm for casuality analysis

```
1  When event specification E is received from the user do
2  calculate the related sub-regions and put them into S
3  send (E,S) to children

4  When (E,S) is received from other nodes do
5  if node ∈ S and parent.coordinator[E] = null then
6  |  coordinator[E] = true
7  end
8  if children ≠ null then
9  |  send (E,S) to children
10 |  wait for the results from children
11 else if node ∈ S then
12 |  collect events for E and put them into events
13 |  area=ID
14 |  send (events,area) to parent
15 end

16 When (events,area) are received from all the children do
17 if coordinator[E] = false then
18 |  if node ∈ S then
19 |  |  collect events for E and put them into levents
20 |  |  area=area ∪ {ID}
21 |  |  events=events ∪ levents
22 |  end
23 |  send (events,area) to parent
24 else if area < S then
25 |  forward the result to neighboring sub-regions
26 end
```

the calculated sub-regions are sent to its children nodes in the routing tree (line 3). The message is forwarded in the tree and each forwarding node waits for the results of its children (line 8-10). When a node belonging to a needed sub-region is detected for the first time, the node changes its flag *coordinator* (with regards to current event) to be true, which marks it as the coordinator of its sub-region. If leaf nodes in the routing tree are detected to belong to one of S, related events and node ID are recorded (line 12-13) and sent to its parent (line 14). When the results of all children are received, the node forwards the results to its parent (line 23), after collecting local events and node ID if it is also belonging to one of S (line 18-21). If the node is a coordinator (*coordinator*[E]=*true*), it compares the accumulated area with S. When more detection is needed, the coordinator forwards the result to its neighboring sub-regions for generating final results (line 22-24).

4.3. Discussion

In this section, we discuss several issues related to asynchronous distributed data collection.

First, Algorithm 2 works under a moderate speed of message passing. We use an example to explain it. Suppose that the gathering events A of shop X happens before the gathering event B of shop Y in the real world, and we want to detect the casuality relation of them based on asynchronous communications. The START message of A needs to reach Y before B happens, if the detection is based on SS, then the casuality relation can be detected. Since the message passing among shops and consumers are uncertain under asynchronous communications, this assumption can not be guaranteed in some cases especially when the messages are forwarded quite slowly. It means that Y may receive the START message of A (say M) after the START message of B (say N), although M happens before N. To handle this problem, we need to revise the algorithm to invoke the detection

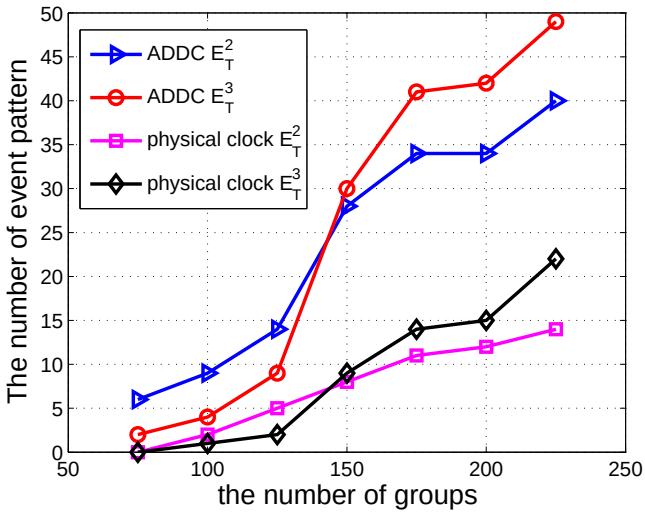

Figure 6. The number of event pattern vs. the number of groups

logic (line 9-15, 25-27) once each message is received. The check of event order also needs to consider the vector clocks of preID and current event, besides $oriStart[preID] \neq null$.

The second concern is more complex temporal relations based on asynchronous communications. ADDC detects causality relations according to the temporal relations based on time points. More complex temporal relations can be based on time intervals. There are 29 and 40 kinds of relations between two time intervals under dense time mode and and 40 kinds of relations between two time intervals under non-dense time mode [8,12]. Moreover, since the temporal relations under asynchronous communications often cannot be determined certainly. Occurrence probability [13] can be used to specify the probability that a relation holds. For the concern discussed above, we can modify ADDC to meet the requirements.

Finally, it is noted that the causality analysis in this paper are based on logic temporal relations under asynchronous communications. This is a little different from the temporal relations based on physical time that can specify a time duration between two gathering events. Such quantity measurement is difficult to measure 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. However, The effectiveness of such solution varies in different situations of message passing.

5. Performance Evaluation

We conduct simulations to validate the effectiveness of our proposed approach ADDC. We compare the results of ADDC with those under physical clock. The detection of causality relation among two/three gathering events are used as the comparison example. The number of event patterns satisfying the causality relation of two gathering events and three gathering events are denoted by E_T^2 and E_T^3, respectively. For the detection under physical clock, we specify the duration between two gathering events T to be 500 seconds. For the detection using ADDC, we need two gathering events to have happen before relation (T is no use in this detection). E_T^2's time duration is measured by the start time of the first gathering event to the start time of the second gathering event, and E_T^3's time duration is measured by the end time of the first gathering event to the start time of the second gathering event.

5.1. 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

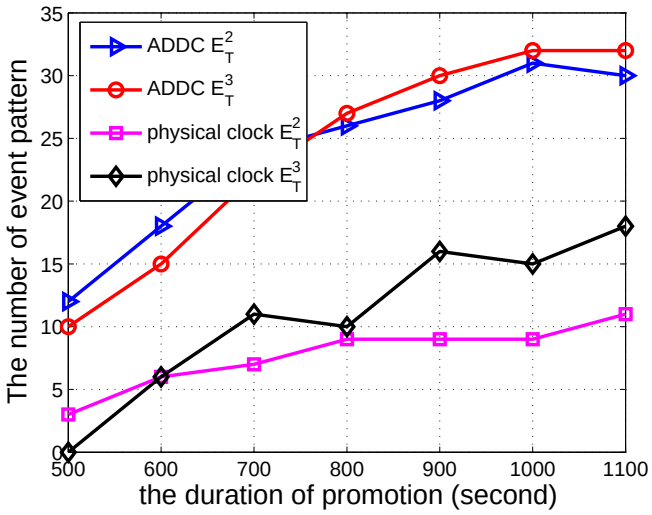

Figure 7. The number of event pattern vs. the duration of promotion

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 Figure 6 - 14.

5.2. 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 Figure 6. It shows that generally speaking the number of E_T^2 and E_T^3 increase when the number of group increases. 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 increases, but with a larger increase speed. Its number is smaller than that of E_T^2 when the number of group is less than 150. This is because 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 gathering events is large. The trend of results of ADDC is similar with those under physical clock. However, the results of ADDC is larger than those under physical clock, since we only specify the happen before constraint while the detection under physical clock need a gap of 500 second. When the gap is decreased, the difference between the two approaches become continuous smaller.

We then change the duration of a promotion from 500 to 1100 seconds and the result is shown in Figure 7. 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 Figure 8. 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.

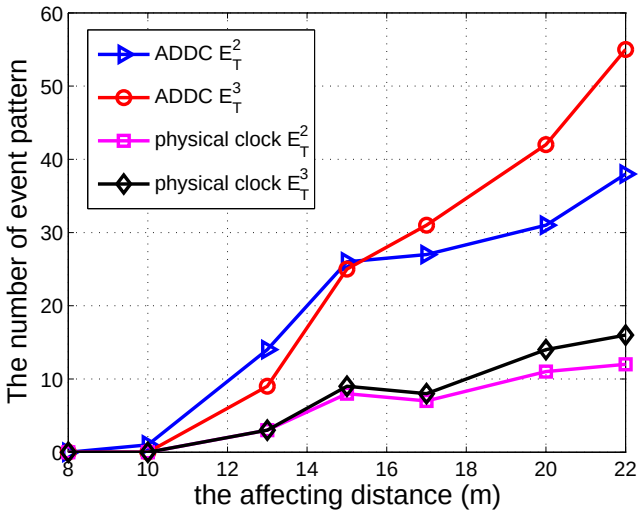

Figure 8. The number of event pattern vs. affecting distance

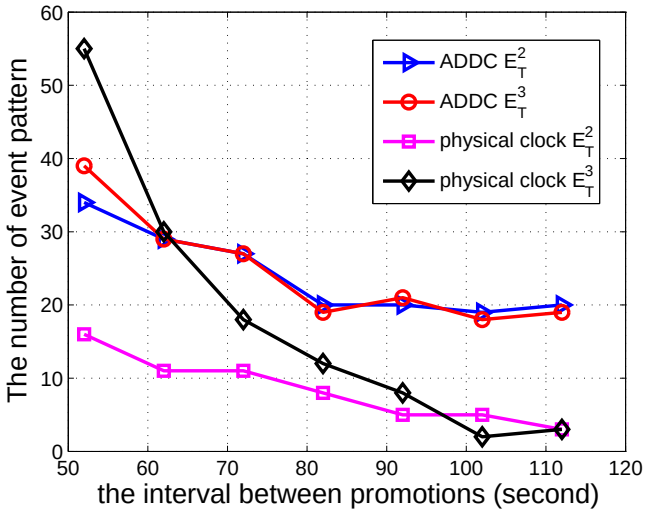

Figure 9. The number of event pattern vs. the interval between promotions

Finally, as shown in Figure 9, 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.

5.3. The Comparison under Different Casuality Definitions

We then compare the detected casuality relation under different definitions. We check the results under synchronized clocks and asynchronous message passing. In each clock environment, we consider multiple definitions including $E_T^2(start - start)$, $E_T^2(end - start)$ and $E_T^2(start - start \ \& \ end - end)$. By default, T is set 500 for synchronized clocks and no use for asynchronous message passing. The detailed description of these definitions can be seen in Section 4.2 and Figure 5. The results can be seen in Figure 10.

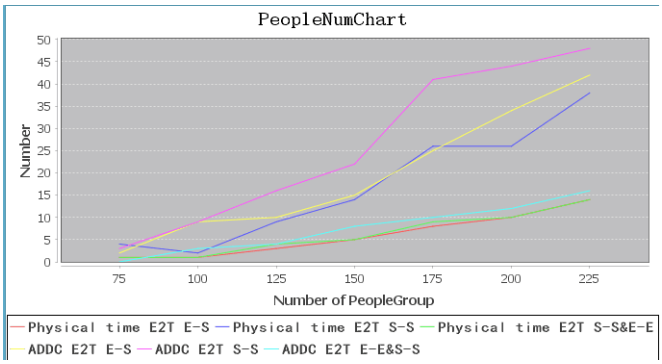

Figure 10. The number of event pattern vs. the number of groups

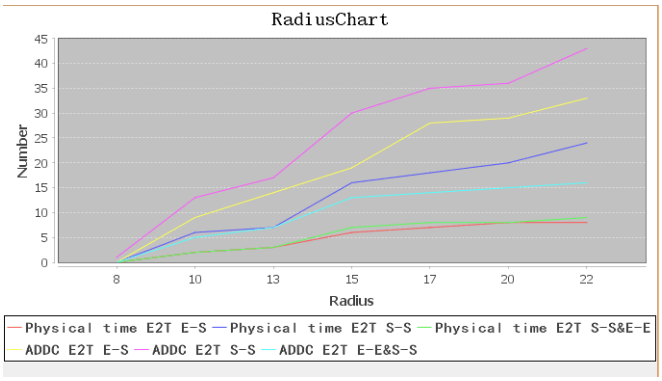

Figure 11. The number of event pattern vs. the number of groups

As shown in Figure 10, all the numbers of event patterns detected under different definitions are increased but show different features. The numbers of event pattern under $E_T^2(start - start$ is larger than that under $E_T^2(end - start)$, and the later is larger than that under $E_T^2(end - start)$ and $E_T^2(start - start$ & $end - end)$. This is because the constraint of $E_T^2(end - start)$ is more strict than $E_T^2(start - start$. If an event pattern satisfies $E_T^2(end - start)$, it also satisfies $E_T^2(start - start)$. But it is not always hold that an event pattern satisfying $E_T^2(start - start)$ can also hold under $E_T^2(end - start)$. Similarly, the constraint of $E_T^2(end - start)$ and $E_T^2(start - start$ & $end - end)$ is more strict than $E_T^2(start - start)$ and generally $E_T^2(end - start)$. It is also can be seen that the event patterns detected by detected is larger than that detected under synchronized clocks, considering the same definition. This is because that the specification of T make the detection under synchronized clocks is more strict.

5.4. 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 Figure 13 and Figure 14. As shown in Figure 13, 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

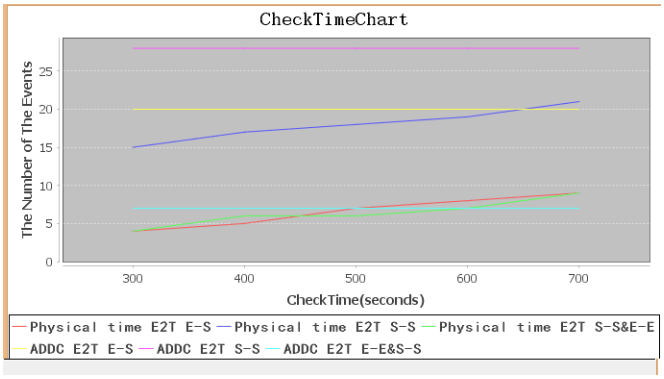

Figure 12. The number of event pattern vs. the number of groups

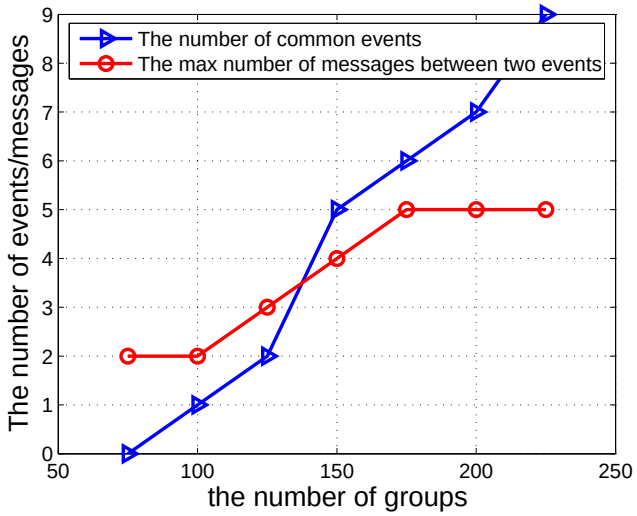

Figure 13. The number of common event pattern / the max number of messages between two events vs. the number of groups

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.

Figure 13 also shows the maximum number of messages when detecting E_T^2. It can be seen that the number of it increases 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 affect on the detection.

Figure 14 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 Figure 13. When the affecting distance increases, the common events increase, and the maximum number of messages between two events also increases. This is consistent with the uncertainty of asynchronous communications.

6. Related Works

Mobile consumption are previously investigated by the researches from different fields, including business management, marketing, psychology, computer science and so.

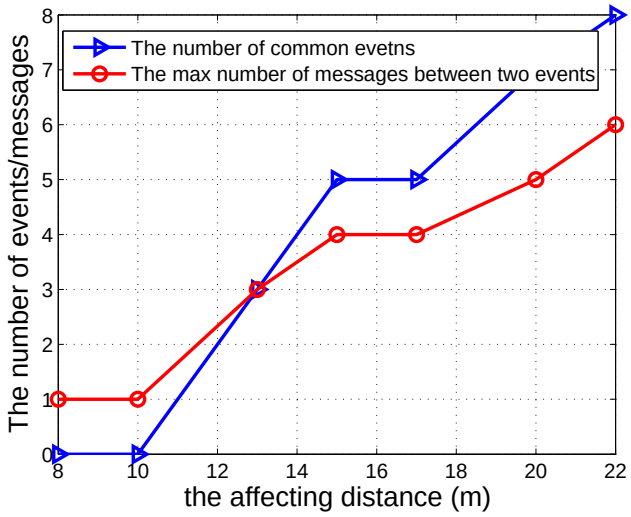

Figure 14. The number of common event pattern / the max number of messages between two events vs. the affecting distance

The works from the fields rather than computer science focus on finding out the relations between a marketing strategy and the consumption based on mobile devices. In [14,15], Ghose et al. and Molitor et al. found that location is an important factor affecting mobile consumption. After receiving the promotion information via mobile devices, the consumers are more likely to buy the goods form the neighboring retailers. In [16], Luo et al. investigated both the promotion time and location to the consumption. They found that: for the consumer that are near the promotion venue, it is better to send the promotion information at the promotion day rather than in advance; for the user that are far from the promotion venue, it is better to send the promotion information one day before the promotion day rather than at that day or two days in advance. In [17], Andrews et al. investigated how crowdedness in a space affects the advertisement on mobile devices. It is found that the crowed environment (e.g. the subway in the rush hours) lead to a more successful advertisement. It can be explained that the constraint in the physical places makes people have more interests on the advertisement. In [18], Molitor et al. investigated how the weather affects the people to accept the coupons sending from mobile applications. In [19], Harmon et al. studied the privacy issues of marketing activities based on mobile devices. In [20], Fihser pointed out that the individuals may perform group-derived behaviors that are mandated because they are part of a social role that the individual has accepted, and not necessarily because they are intrinsically satisfying. For example, as one of fans of a sports team, they may buy the gate receipts, team-licensed products, etc. Such group consumption can be affected by attractiveness of the group and the similarity between group members. However, this research is not specific to mobile group consumption.

There are many prior works on mining the rules of mobile users and predicate the user's location or path. Tseng and Tsui mine the associated rules between the movement of people and the requested services in different locations [21]. They take into account the multilevel properties of locations and service. In [22], Tseng and Lin propose an approach called SMAP-Mine based to discover patterns of users' mobile access patterns that contain both movement and service requests simultaneously. SMAP-Tree is used to organize the data efficiently in the approach. Yun and Chen mine sequential patterns in a mobile commerce environment [23]. They explore the relationship between moving and purchase patterns of users to mining the patterns. The results show that this approach has the more accurate results than using only moving or purchase data. In [24], Hadjiefthymiades and Merakos use a linear reward-penalty reinforcement learning method for location prediction. This approach need many possible transitions of the user's mobility pattern for training before prediction. In [25], Akoush and Sameh use hybrid bayesian neural network model for the predication. Anagnostopoulos

and Hadjiefthymiades further eliminate the noise (typically manifested as small-random deviations from previously seen patterns) in the movement patterns using the optimal stopping theory, and predicated the user's k-step ahead location [26]. The approach tries to keep the knowledge base with the lowest possible total spatial variance. The approach use a delay tolerant decision making mechanism such that does not require a training phase but can incrementally refine the knowledge base. In [27], Jeung et al. proposed the *trajectory pattern tree* to organize a large number of discovered trajectory patterns. The trajectory patterns are indexed in the tree for answering predictive query efficiently. A hybrid model based on trajectory pattern and mathematical motion function is built to fast predict the movement of people in the near and distant future. In [5], Lu et al. collected the users' movements and purchase transactions information in mobile commerce for mining and prediction. They propose a framework called Mobile Commerce Explorer (MCE), which improved prior works by 1) considering the similarity between shops and between items in the pattern mining and prediction 2) mining the moving patterns for individuals and 3) predicting the behaviors of consumers in terms of both the movement and purchase.

Some works also directly investigate how to promote the mobile consumption. In [4], Namiot et al. localize the consumers based on their wireless network access information and then provide them customized commercial information (e.g. deals, discounts, coupons etc.). Yang et al. collect the users' browsed websites and their locations, and then recommend new retailers to them[6]. Kanda et al. collect the users' realtime trajectory based on laser rangefinder and particle filter, and then predicate the time and venue that the most likely a consumer perform a consumption [28]. After that, the robot will be sent to provide various services. In [29], Guo et al. investigate how to use RFID technology to promote mobile-commerce. In [30], Komninos et al. build the system "me-Commerce" and point out that various of context-information are important for the mobile consumption. These works do not consider the scenarios of group consumption. In [31], Zhu et al. tackle the problem of discovering user group base on their similar movement behaviors. They propose a framework to find such groups by firstly constructing trajectory profiles of users, then deriving similarity between trajectory profiles, and finally discovering the groups. This result is useful for mobile group consumption but more investigation is needed.

Moreover, all these works are based on centralized server and synchronized clock, which suffers from performance bottleneck, one-point failure, overhead of 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 [32] for detecting the events. The lattice is invented as a tool for detecting generic events, called *relational predicates* [32,33]. As a special class of predicates, *conjunctive predicates*, specified by a conjunctive expression of local states, are also investigated [9]. In [13], Zhu et al. use 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.

7. Conclusion

In this paper, we proposed an asynchronous distributed approach to collect the data of mobile group consumption. Considering that in many scenarios the consumers are of great number and constantly move, our approach is scalable and efficient since it does not need a central server or synchronized clock. We first build the system model based on asynchronous distributed communications. 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 detect the causality relation of two/three gathering events happened in the system based on message exchange. Three definitions of the causality relation

are supported. We conduct extensive simulation to validate the proposed approach. The results show that our approach is quite effective.

Acknowledgments: This research is supported in part by Outstanding Young Academic Talents Start-up Funds of Wuhan University No. 216-410100004, the Fundamental Research Funds for the Central Universities of China No. 216-410500026, National Natural Science Foundation of China No. 61502351, Nature Science Foundation of Hubei, China No. 2015CFB340 and Shenzhen Nanshan District Project No. KC2015ZDYF0015A.

Author Contributions: Weiping Zhu, Weiran Chen and Jiaojiao Chen developed the ideas, designed the algorithms and wrote the paper. Zuoyou Li, Zejie Hu and Yue Liang implemented the algorithms and performed the evaluation.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

GCS: Consumption Simulation system
ADDC: Asynchronous Distributed Data Collection Approach

Bibliography

1. Moussaid, M.; Perozo, N.; Garnier, S.; Helbing, D.; Theraulaz, G. The Walking Behaviour of Pedestrian Social Groups and Its Impact on Crowd Dynamics. *PLoS ONE* **2010**, *5*, 521–544.
2. InMobi Insights Team. Global Mobile Media Consumption, 2013.
3. eMarketer. Mobile Ad Spend in China Hits $7 Billion This Year, 2014.
4. Namiot, D.; Schneps-Schneppe, M. About Location-aware Mobile Messages: Expert System Based on WiFi Spots. IEEE International Conference on Next Generation Mobile Applications, Services and Technologies, 2011, pp. 48–53.
5. Lu, E.C.; Lee, W.C.; Tseng, V. A Framework for Personal Mobile Commerce Pattern Mining and Prediction. *IEEE Transactions on Knowledge and Data Engineering* **2012**, *24*, 769–782.
6. Yang, W.S.; Cheng, H.C.; Dia, J.B. A Location-Aware Recommender System For Mobile Shopping Environments. *Expert Systems with Applications* **2008**, *34*, 437–455.
7. Chen, W.; Li, Z.; Liang, Y.; Chen, J.; Zhu, W. An Asynchronous Distributed Data Collection Approach for Mobile Group Consumption. Proceedings of International Conference on Identification, Information & Knowledge in the Internet of Things (IIKI), 2015, pp. 35 – 42.
8. Kshemkalyani, A.D. A Fine-Grained Modality Classification for Global Predicates. *IEEE Trans. on Parallel and Distributed Systems* **2003**, *14*, 807–816.
9. Garg, V.K.; Waldecker, B. Detection of Weak Unstable Predicates in Distributed Programs. *IEEE Trans. on Parallel and Distributed Systems* **1994**, *5*, 299–307.
10. Lamport, L. Time, clocks, and the ordering of events in a distributed system. *Communications of the ACM* **1978**, *21*, 558–565.
11. Fidge, C. Logical time in distributed computing systems. *IEEE Computer* **1991**, *24*, 28–33.
12. Kshemkalyani, A.D. Temporal interactions of intervals in distributed systems. *Journal of Computer and System Sciences* **1996**, *52*, 287–298.
13. Zhu, W.; Cao, J.; Raynal, M. Predicate Detection in Asynchronous Distributed Systems: A Probabilistic Approach. *IEEE Transactions on Computers* **2015**, *PP*, 1–1.
14. Ghose, A.; Goldfarb, A.; Han, S.P. How Is the Mobile Internet Different? Search Costs and Local Activities. *Information Systems Research* **2013**, *24*, 613–631.
15. Molitor, D.; Reichhart, P.; Spann, M.; Ghose, A. Measuring The Effectiveness of Location-based Advertising: A Randomized Field Experiment. *http://www.fox.temple.edu/cms/wp-content/uploads/2015/01/mksc-crowd.pdf* **2014**.
16. Luo, X.; Andrews, M.; Fang, Z.; Phang, C.W. Mobile Targeting. *Management Science* **2013**, *60*, 1738–1756.

17. Andrews, M.; Luo, X.; Fang, Z.; Ghose, A. Mobile Ad Effectiveness: Hyper-Contextual Targeting with Crowdedness. *Marketing Science* **2014**.
18. Molitor, D.; Reichhart, P.; Spann, M. Location-based Advertising: Measuring The Impact of Context-Specific Factors on Consumers' Choice Behavior. *Munich School of Management, Available at SSRN: http://ssrn.com/abstract=2116359* **2013**.
19. Harmon, R.; Unni, R. Perceived Effectiveness of Push vs. Pull Mobile Location-Based Advertising. *Journal of Interactive Advertising* **2007**, *7*, 28–40.
20. Fisher, R.J. Group-Derived Consumption: the Role of Similarity and Attractiveness in Identification With a Favorite Sports Team. *Advances in Consumer Research* **1998**, *25*, 283–288.
21. Tseng, S.M.; Tsui, C.F. Mining multilevel and location-aware service patterns in mobile web environments. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* **2004**, *34*, 2480–2485.
22. Tseng, V.; Lin, K. Mining sequential mobile access patterns efficiently in mobile Web systems. The 19th International Conference on Advanced Information Networking and Applications (AINA), 2005, Vol. 2, pp. 762–767.
23. Yun, C.H.; Chen, M.S. Mining Mobile Sequential Patterns in a Mobile Commerce Environment. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on* **2007**, *37*, 278–295.
24. Hadjiefthymiades, S.; Merakos, L. Proxies+ Path Prediction: Improving Web Service Provision in Wireless–Mobile Communications. *Mobile Networks and Applications* **2003**, *8*, 389–399.
25. Akoush, S.; Sameh, A. Mobile User Movement Prediction Using Bayesian Learning for Neural Networks. Proceedings of the International Conference on Wireless Communications and Mobile Computing (IWCMC), 2007, pp. 191–196.
26. Anagnostopoulos, C.; Hadjiefthymiades, S. Intelligent Trajectory Classification for Improved Movement Prediction. *Systems, Man, and Cybernetics: Systems, IEEE Transactions on* **2014**, *44*, 1301–1314.
27. Jeung, H.; Liu, Q.; Shen, H.T.; Zhou, X. A Hybrid Prediction Model for Moving Objects. IEEE 24th International Conference on Data Engineering (ICDE), 2008, pp. 70–79.
28. Kanda, T.; Glas, D.F.; Shiomi, M.; Ishiguro, H.; Hagita, N. Who will be the customer?: a social robot that anticipates people's behavior from their trajectories. *in Proc. of UbiComp* **2008**, pp. 380–389.
29. Guo, W. Research on mobile commerce based on RFID. IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI), 2008, Vol. 1, pp. 853–857.
30. Komninos, A.; Barrie, P.; Newman, J.; Landsburgh, S. Me-Commerce: an Infrastructure for Personal Predictive Mobile Commerce. Proc. of International Conference on Mobile Business (ICMB), 2006, pp. 39–39.
31. Zhu, W.Y.; Peng, W.C.; Hung, C.C.; Lei, P.R.; Chen, L.J. Exploring Sequential Probability Tree for Movement-Based Community Discovery. *IEEE Transactions on Knowledge and Data Engineering* **2014**, *26*, 2717–2730.
32. Cooper, R.; Marzullo, K. Consistent detection of global predicates. Proc. of the ACM/OCR Workshop on Parallel and Distributed Debugging, 1991, pp. 163–173.
33. Marzullo, K.; Neiger, G. Detection of global state predicates. Proc. of the 5th Workshop on Distributed Algorithms, 1991, pp. 254–272.