Performance Analysis of Virtual Time Optimistic Transaction Processing

Cong Liu [1], Wei Huang [2], Zhiguo Zhang [1]

[1] Department of Computer Science, Information Science & Technology College, Sun Yat-sen University, No. 135, Xingang Xi Road, Guangzhou, 510275, China
gzcong@hotmail.com, lnszzg@zsu.edu.cn

[2] Research center of software, Information Science & Technology College, Sun Yat-sen University, No. 135, Xingang Xi Road, Guangzhou, 510275, China
huangbodao@yahoo.com.cn

Abstract. Aiming at solving the problems in the mobile computing environment such as low bandwidth, frequent disconnection and low battery capacity, we propose an improved optimistic transaction processing method -- the virtual time optimistic transaction processing protocol. This protocol improves the performance of optimistic transaction processing by extending the concept of committability or, more specifically, by releasing the constraint of the total order relation between all transactions based on our analysis of the transaction processing approaches from a different angle. In this paper, we first explain and give the algorithm of the virtual time optimistic approach. Then we present and show the result of a simulation on the virtual time optimistic approach. Finally, we make comparison and performance analysis based on the simulation. The comparison and performance analysis show that the protocol has interesting performance gain in the metric of the number of abort.

1 Introduction

With the advent of third generation wireless infrastructure and the rapid growth of wireless communication technology such as Bluetooth and IEEE 802.11, mobile computing becomes possible. People with battery powered mobile devices can access various kinds of services at any time any place. However, existing wireless services are limited by the constraints of mobile environments such as narrow bandwidth, frequent disconnections, and limitations of the battery technology. Thus, approaches to efficiently process transactions have received considerable attention [1, 3, 4, 9, 10, 15, 17, 20].

The pessimistic approach successfully serializes all the transactions that have data dependency with each other [9]. However, it is the least efficient one in the mobile environment for the overhead of locking, waiting for locks and deadlock. The optimistic approach is therefore proposed and successfully address some of the above problems by the optimistically assuming that the data items accessed by a transaction would not likely to be updated before it commits [2, 5, 6, 7, 8, 11, 12, 13, 14, 16, 18, 19]. The virtual time optimistic approach, which is our contribution, allows that the imaginary time when a transaction performs all its operations and then commits to be a virtual time and consequently extends the concept of committability, and therefore offer an interesting performance gain.

The paper is structured as follows: Section 2 discusses from a different angle the way to handle concurrency control and show the advantage of the virtual time optimistic transaction processing approach. Section 3 provides the algorithm and proof for the virtual time optimistic approach and some complementary algorithms. Section 4 provides an simulation to do experimental comparison and performance analysis to our approach and section 5 concludes the paper.

2 Motivating Analysis

If the relations between committed transactions could be represented by a directed acyclic graph in which each transaction is a vertex and each precedency relation is an edge, they are serializable because any directed acyclic graph has at least one topological order. But it seams that a transaction cannot easily be represented by a dot because transactions do not happen in instant but have durations of execution and might interleave with each other in time. However, if certain constraints are satisfied, it is possible to imagine transaction to be an instant action. If so, transaction processing could be implemented using a directed acyclic graph. Actually, some of the transaction processing approaches could be through of as trying to do that. For example, by making the data items visited by a transaction temporarily “invisible” to the other conflicting transactions, the Strict 2PL in [3] provides the illusion that a transaction only executes on the time it commits. As another example, the optimistic approach, such as the one proposed in [16], also provides the same illusion by facilitating the local cache and installing the data items modified by the transaction into the database only when it commits if the optimistic assumptions are satisfied. The later is obviously more desirable in the mobile computing environment because it eliminates the overhead on locking, waiting for lock holders and restarting when deadlock occurs. However, both of the above approaches have the dots corresponding to a real time, that’s each dot is a spot in the time vector. Thus, the dependence relation between transactions is a total order relation. But it is possible to be a partial order because not all the transactions do have conflict operations. By not corresponding each dots of transactions to a real time, or using virtual time to represent the virtual execution instant of a transaction, we could extend the concept of committability and hence improve the overall efficiency of transaction processing.

The virtual time optimistic transaction processing approach is an extension to the traditional optimistic approach. In the traditional ones, such as in [16], timestamp is used to present the precedency relation between transactions, and the transaction commits earlier has a smaller timestamp than the transaction commits later. This constitutes a total order relation between all transactions, because any two transactions are comparable to each other in timestamp. This approach is effective, but it is not efficient however, because only those transactions have conflict operation is need to have a precedency relation between them in order to ensure valid executions and otherwise it is unnecessary. The virtual time optimistic transaction processing approach releases this constraint by presenting the partial order precedency relation between transactions in a directed acyclic graph..

3 The virtual time optimistic approach

For the purpose of simplifying the system model that our simulation is based on and for the consideration of the limitations of the mobile devices, we could make the assumptions that there is only one database server and the amount of data items that each transaction access is very small so that the limited amount of available local storage on the mobile devices can cache the data items they process and there are only short transactions. So, we use a very simple system model where there is only a single database and a number of mobile agents where transactions are running. Each data item records the number of the last transaction that updated it. Some information about the committed transaction is also maintained on the database, including the set of the data items red or written by the transaction and the directed acyclic graph used to hold the partial order precedency relation between transactions. We use the following denotation convention: *Ta* <(>) *Tb* denotes an edge in the DAG which means The virtual execution time of transaction *Ta* is earlier(later) than that of transaction *Tb*.

Algorithm 1, This algorithm is executed in the respect of an executing transaction, and any transaction that is visible to the others is committed with their information sent to the database.

Each transaction *T* maintains a DAG in the local memory to hold part of the DAG in the database.

Add *T* > *Tewd* in the local DAG if *Tewd* updates a data item *D* (executes) before *T* read *D*.

Add *T* < *Tlwd* in the local DAG if *Tlwd* updates a data item *D* (executes) after *T* read *D*, that's *Tlwd* > *Tewd*.

Add *T* > *Trd* in the local DAG if *Trd* read a data item *D* (executes) that *T* updates.

When *T* is prepared, try to add the local DAG into the global DAG in the database. If it constitutes a cycle in the global DAG, restart *T*. Otherwise, commit *T*.

When committing *T*, for each *D* updated by *T*, if *T* < *Twd*, where *Twd* is the last transaction updated *D*, ignore *T*'s update on *D*. Otherwise install *T*'s update on *D* in the database and add *T* > *Twd* on the global DAG.

Theorem 1. The The virtual time optimistic approach generates serializable execution of transactions. **Proof**: Because the global DAG is acyclic, T and the rest of the transactions have a topological order on the precedency relation, therefore they are serializable.

The above algorithms should be combined with some other techniques in practical situations, including the technique to detect incommittable transactions before they prepare and the technique to delete something obsolete from the global DAG on the database periodically.

Algorithm 2, This algorithm uses *Invalidation Report* (*IR*) to earlier abort transactions that are impossible to commits successfully before they are prepared. When an *IR* containing the read-set and write-set of a committed transaction *Tc* is disseminated to *T*, the checking is perform in *T*'s host agent as described in algorithm 1. If after the checking the local DAG is cyclic, restart *T*.

Algorithm 3, As time goes by, the DAG would occupy a great amount of disk space. This algorithm is used to delete the obsolete information. Any information represent a transaction has a life span (which is much larger than the general duration of a transaction), and the information is eliminate at the end of this time span. Using this algorithm, a change should be made to algorithm 1. That's, any prepared transaction trying to add an edge from a non-existed transaction in the global DAG should be restarted.

4 Simulation and performance evaluation

We use the aforementioned system model in our simulation in this section. This section consists of 3 parts. In the first part, we simulate the real execution of the transaction processing system to approximately testify the correctness of the virtual time optimistic transaction processing approach. In the second part, we compare the virtual time optimistic transaction processing approach with the optimistic transaction processing protocol in [16]. The third part is a performance analysis of the virtual time transaction processing approach.

The major experimental parameters are listed in the table below. In each of the performance analysis, one or two parameters vary to get the curve while the other values remain unchanged.

Table 1. Eexperimental parameters setting

Parameter Name	**Value**
Number of transaction	100
Transaction starting time	1~100
Number of data items	30
Number of mobile agents	20
Number of operation in a transaction	1~5
Percentage of write operation in a update transaction	30%
Offset of accessed data item in the subsequent operation.	1
Time to perform a read operation	3
Time to perform a mathematic operation	5~20
Time to perform invalidation checking	3
Time to transfer data	50
Time to restart a transaction	10

The method to testify the correctness of the virtual time optimistic transaction processing approach is described as follows: 1) randomly generate a number of transactions *T* and data items *D1*; 2) using the virtual time optimistic transaction processing approach to process all the above transactions T and get the DAG and the resulting values *D2* of the data items; 3) restore the data item to value *D1* and execute the transactions *T* again serially in a topological order of the above DAG and get the resulting values *D3* of the data items; 4) compare to see if *D2* and *D3* are equal. If so, the virtual time transaction processing protocol is equal to a serial execution. We repeat this test for 100,000 times and found completely correct results, thus we testified approximately that the virtual time transaction processing approach generates serializable execution of transactions

Next we will compare the virtual time optimistic transaction processing approach to the optimistic transaction processing protocol in [16]. The later is an optimistic transaction processing protocol designed to improve the overall throughput in the mobile environment. In the comparison, we use the number of abort as the metric to evaluate the performance of the different approaches. We assume all transaction would successful finished their execution if there is no other transactions. The transaction is aborted and restarted if it is found conflicting with any of the committed transaction (The DAG becomes cyclic if the transaction is committed).

Different from some of other optimistic transaction processing approaches, the virtual time optimistic transaction processing approach would not commit read only transaction autonomously, but if the last time when the transaction summits data access request to the database server can be pre-determined (for example, the transaction is summit with pre-declaration information provided by the compiler), the pre-commit request could be appended to that request and send to the server. However, the overhead to access the data through the wireless link is saved when the first operation to some data items is a write operation. More importantly, in other optimistic transaction processing approaches, a newly committed transaction must have a greater timestamp than those have committed, but in the virtual time optimistic transaction processing approach, a transaction is committable only if it keeps the DAG acyclic. So, the virtual time optimistic transaction processing approach releases the transaction from the stricter total order relation to a looser partial order relation in theory and enlarger the chance that a transaction is committable in practice. From figure 1, we can see that the virtual time optimistic transaction processing approach always has better performance than the optimistic transaction processing protocol in [16] as the number of transaction increasing from 20 to 100, and the performance remains tolerable even when the confliction on data items is relatively high (when the number of data items is 20 and the number of transactions is 100).

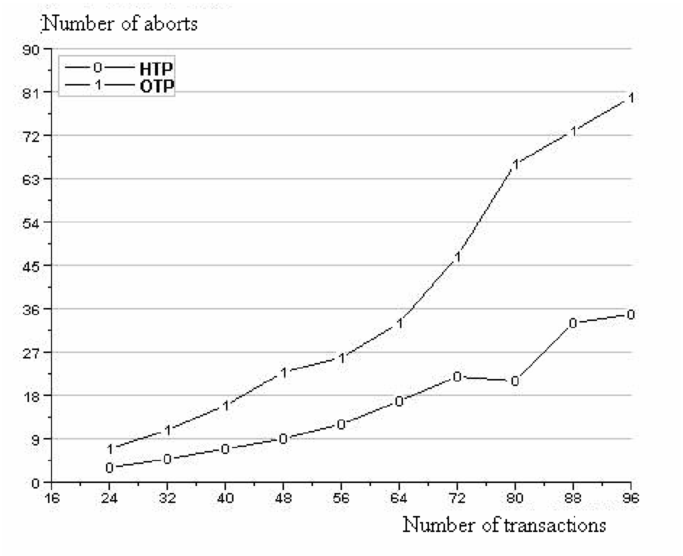

Fig. 1. The optimistic transaction processing protocol in [16] (OTP) vs. the virtual time optimistic transaction processing approach (HTP)

Finally, we will analyze the performance of the virtual time optimistic transaction processing approach using the experimental results showed in figure 2 and figure 3. As in the performance comparison, we also use as the metric of the number of aborts to evaluate the performance. There are 3 curves in each of these figures. In figure 2, these curves represents the system setting where the number of data items are different. The 3 curves in figure 2 represent 6, 12 and 18 data items respectively. In figure 3, each of these curves represents a kind of data access pattern which means the percentage of write operations in an update transaction and is showed in these figures as update rate. The 3 curves in each of these figures represent the update rate of 30, 60 and 90 respectively

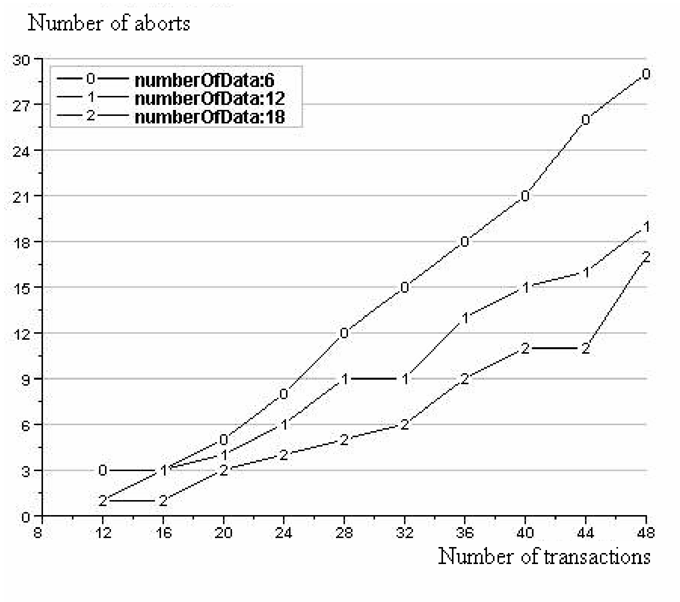

Fig. 2. Number of aborts vs. number of data items and number of transactions

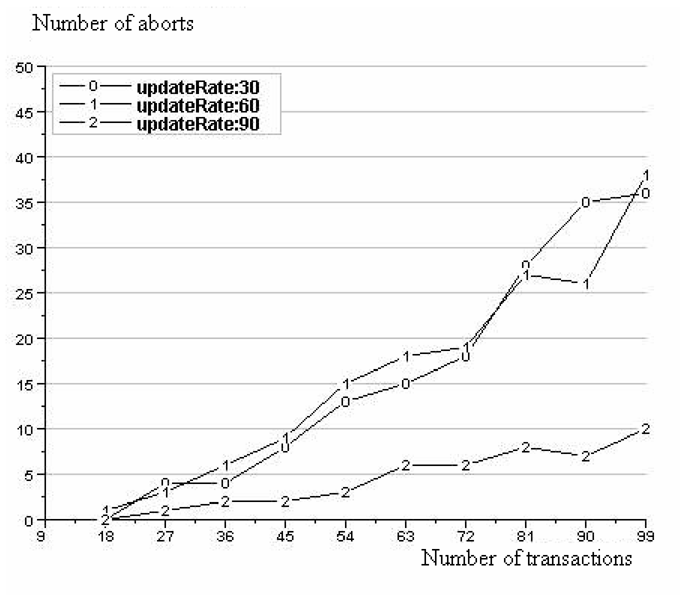

Fig. 3. Number of aborts vs. update rate and number of transactions

In figure 2, we found as we expect that the number of aborts increase as the number of transactions increases and as the number of data items decreases. That is because the chance of confliction between transactions when accessing data items increases as the number of transaction increases and as the number of data items decreases.

In figure 3, readers maybe surprising to find that the number of aborts is not always increase as the update rate increases as would be seen in the results of the other transaction processing approaches where conflict occurs between read operation and write operation and between write operations. In the virtual time optimistic transaction processing protocol, precedency relation between the virtual execution times of the transactions does not correspond to that of the real execution times nor does it to that of the commit time. So, some of the conflicts between write operations would be avoided by adjusting the precedency relation between the virtual execution time. Observation shows that write conflicts rarely happen in this approach. In the virtual time optimistic transaction processing approach, the major conflict between transactions is read-write conflict. As is suggested in figure 3, the number of aborts does not increase as the update rate increases, but it reaches its maximum value when the update rate is between 30 and 60, which means that the overall performance become worst when the rate of read operation to write operations reaches a certain proportion but not in extremes.

5 Conclusion

In this paper, we have proposed an improved optimistic transaction processing method -- the virtual time optimistic transaction processing protocol which improved the performance of optimistic transaction processing in the mobile computing environment by extending the concept of committability or by releasing the constraint of the total order relation between all transactions based on our analysis of transaction processing from a different angle. We have explained and given the algorithm of the virtual time optimistic approach, and presented and showed the result of a simulation on the virtual time optimistic approach. And finally, comparison and performance analysis based on the simulation was performed. Our next step is to make it richer and deeper on the theoretical side and also more practical than it is now.

References

1. S. Acharya, R. Alonso, M.J. Franklin and S.B. Zdonik: Broadcast Disks: Data Management for Asymmetric Communications Environments, in Proc. ACM SIGMOD International Conf. on Management of Data (1995) 199–210
2. D. Barbara: Certification Reports: Supporting Transactions in Wireless Systems, in Proc. IEEE International Conf. on Distributed Computing Systems (1997) 466–473

3. P.A. Bernstein, V. Hadzilacos, and N. Goodman: Concurrency Control and Recovery in Database Systems, Addison-Wesley, Massachusetts (1987)
4. B. Bhargava: Concurrency Control in Database Systems, IEEE Trans. on Knowledge and Data Engineering (1999) vol.11, no.1, 3–16
5. I. Chung, J. Ryu and C.-S. Hwang: Efficient Cache Management Protocol Based on Data Locality in Mobile DBMSs, in Current Issues in Databases and Information Systems, Proc. Conf. on Advances in Databases and Information Systems, Lecture Note in Computer Science, vol.1884, 51–64, Springer (2000)
6. J. Jing, A. Elmagarmid, A. Helal and A. Alonso: Bit Sequences: An Adaptive Cache Invalidation Method in Mobile Client/Server Environments, Mobile Networks and Applications (1997) vol.2, no.2, 115–127
7. A. Kahol, S. Khurana, S.K. Gupta and P.K. Srimani: An Ef.-cient Cache Maintenance Scheme for Mobile Environment, in Proc. International Conf. on Distributed Computing Systems (2000) 530–537
8. V.C.S. Lee and K.-W. Lam: Optimistic Concurrency Control in Broadcast Environments: Looking Forward at the Server and Backward at the Clients: in Proc. International Conf. on Mobile Data Access, Lecture Note in Computer Science, vol.1748, 97–106, Springer (1999)
9. S.K. Madria and B. Bhargava: A Transaction Model to Improve Data Availability in Mobile Computing, Distributed and Parallel Databases (2001) vol.10, no.2. 127–160
10. E. Pitoura and B. Bhargava: Data Consistency in Intermittently Connected Distributed Systems, IEEE Trans. On Knowledge and Data Engineering (1999) vol.11, no.6, 896–915
11. E. Pitoura and P.K. Chrysanthis: Exploiting Versions for Handling Updates in Broadcast Disks, in Proc. International Conf. on Very Large Databases (1999) 114–125
12. E. Pitoura and G. Samaras, Data Management for Mobile Computing, Kluwer, Boston, (1998)
13. M. Satyanarayanan: Mobile Information Access, IEEE Personal Communications (1996) vol.3, no.1, 26–33
14. J. Shanmugasundaram, A. Nithrakashyap and R. Sivasankaran: Efficient Concurrency Control for Broadcast Environments, in Proc. ACM SIGMOD International Conf. on Management of Data (1999) 85–96
15. K. Stathatos, N. Roussopoulos and J.S. Baras: Adaptive Data Broadcast in Hybrid Networks, in Proc. International Conf. on Very Large Data Bases (1997) 326–335
16. IlYoung Chung, Bharat Bhargava, Malika Mahoui, and Leszek Lilien: Autonomous Transaction Processing Using Data Dependency in Mobile Environments, in Proc. Workshop on Future Trends of Distributed Computing Systems (2003) 138-144
17. SangKeun Lee, SungSuk Kim: Performance Evaluation of a Predeclaration-based Transaction Processing in a Hybrid Data Delivery, in Proc. IEEE International Conference on Mobile Data Management (2004)
18. SangKeun Lee, Chong-Sun Hwang, Masaru Kitsuregawa: Using Predeclaration for Efficient Read-Only Transaction Processing in Wireless Data Broadcast, IEEE Transactions on knowledge and data engineering (2003) vol.15, no.6
19. Bettina Kemme, Fernando Pedone, Gustavo Alonso, Andre¡ä Schiper, Matthias Wiesmann: Using Optimistic Atomic Broadcast in Transaction Processing Systems, Transactions on knowledge and data engineering (2003) vol.15, no.4
20. Vasudevan Janarthanan, Purnendu Sinha: Modular Composition and Verification of Transaction Processing Protocols, Proc. the 23rd International Conference on Distributed Computing Systems, IEEE (2003)