Research Article Impact Factor: 0.621 ISSN: 2319-507X
Kshitija Patil, IJPRET, 2014; Volume 2 (9): 448-455 IJPRET

INTERNATIONAL JOURNAL OF PURE AND APPLIED RESEARCH IN ENGINEERING AND TECHNOLOGY

A PATH FOR HORIZING YOUR INNOVATIVE WORK

REVIEW OF FAULT TOLERANT STRATEGIES FOR QUERY PROCESSING IN MOBILE COMPUTING ENVIRONMENT

KSHITIJA PATIL[1], DR. VINOD PATIL[2], DR. VILAS THAKARE[3], DR. VARSHA TONDRE[4]

1. Asst. Prof., Degree College of Physical Education, H.V.P.M., Amravati(Mah.),
2. H.O.D., Shree Shivaji Science college, Akola,
3. H.O.D., Sant Gadgebaba Amravati University, Amravati (Mah.),
4. H.O.D., Brijlal Biyani Science College, Amravati (Mah.)

Accepted Date: 27/02/2014 ; Published Date: 01/05/2014

Abstract: In mobile computing environment, the devices may be moving continuously, resulting in termination of links from the network. The link may resume again by connecting to some other part of network. Therefore, it can be viewed as a dynamic type of distributed system. As there is a considerable increase in the number of components in mobile environment, there is increase in the failure probability. The query issuers and the data object are continuously roaming. Therefore, problems like limited communication range, limited battery power, and scarce communication resources generally occur in mobile environment. This paper focuses on the different fault tolerant strategies for query processing in mobile environment.

Keywords: Mobile environment, Query Processing, Mobile computing, Fault tolerance.

Corresponding Author: MS. KSHITIJA PATIL

Access Online On:

www.ijpret.com

How to Cite This Article:

Kshitija Patil, IJPRET, 2014; Volume 2 (9): 448-455

PAPER-QR CODE

INTRODUCTION

There are various aspects of query processing in mobile environment since the area of mobile computing is very broad. There are multiple network connectivity options (3G, Wi-Fi, Bluetooth, UWB, etc.), with different features. Mobile computing can be termed as a specialized class of distributed systems, where some terminals move apart from mobile network, move freely in the physical space and rejoins to a possibly different segment of a computer network in order to resume delayed kshiti_krp@rediffmail.com activities[1]. With the increase in use of mobile devices there is considerable increase in fault occurrence in query processing operation in mobile environment. To provide fault tolerance it is necessary to understand the nature of the faults that occur in mobile computing environment. Fault tolerance techniques enable systems to perform tasks in the presence of faults. The likelihood of faults grows as systems are becoming more complex and applications are requiring more resources, including execution speed, storage capacity and communication bandwidth. By considering various types of faults, this paper takes a review of different fault tolerant strategies used for query processing in mobile environment.

2. Query processing in mobile computing environment

In computer applications, mobile computing deals with the mobility of hardware, software and data on the mobile devices. The need for query processing arises to access data from these devices. For example, a mobile user looking for a taxi can fire a query to find location of taxi. The main function of a query processor is to receive a query as input, transform high-level query (typically, in relational calculus) into an equivalent lower-level query (typically, in some variation of relational algebra), optimizing this query in an executable query plan, and executes the query to get desired result [2]. The database system in a mobile environment may be centralized, distributed, or parallel systems.

3. Faults in query processing

A mobile computing system is a distributed system where some of processes are running on mobile hosts (MHs), whose location in the network changes with time. Mobile distributed systems raise new issues such as mobility, low bandwidth of wireless channels, disconnections, limited battery power and lack of reliable stable storage on mobile nodes. There is a variety of limitations occurring in mobile environment. Due to the restricted and uneven bandwidth or limited communication range, it is necessary that the mobile nodes make the most advantageous use of the connectivity when it arrives. The faults/limitations in query processing can be summarized as follows: Disconnection

- Low bandwidth

- High bandwidth variability
- Low power and resources
- Security risks
- Wide variety terminals and devices with different capabilities
- Device attributes
- Fit more functionality into single, small device

4. Fault Tolerant Strategies For Query Processing In Mobile Computing Environment

To overcome faults, several fault tolerant strategies are proposed by different authors. Some of them are discussed here:

4.1 Replication Strategies in Mobile Environments

To increase the performance of query processing, throughput replicating data at several sites is a powerful mechanism which can provide fault tolerance. In general, there are two types of replication strategies: Synchronous and Asynchronous replication. It is seen that unlike synchronous replication asynchronous replication may not keep the database consistent at every moment. If the time lag is compromised, it may result in fewer requirements of resources. Different data ownership models are proposed [4], which specifies various mechanisms that can be adopted to keep the replicas consistent. These are Workload partitioning data ownership model, Master/Slave data ownership model and Update anywhere data ownership model. These mechanisms may be adopted based on different applications implemented in mobile environments. On the basis of volume of replication and the number of transactions respective ownership models may be chosen. The update anywhere ownership model can make the replication strategies in increasing the throughput but has to be implemented as synchronous replication as any site can update data at any time.

4.2 Fault Tolerant Checkpointing Protocols for Mobile Computing Systems

There are mainly two kinds of faults: permanent and transient. Permanent faults are caused by permanent damage to one or more components and transient faults are caused by changes in environmental conditions. Permanent faults can be rectified by replacement or repair of components. Duration of transient faults is short and is difficult to detect and deal with. Thus, it becomes necessary to provide fault tolerance mainly for transient failures in distributed computers. Fault-tolerant techniques enable a system to perform tasks in the presence of faults and involves fault detection, fault location, fault containment and fault recovery. [5] Fault tolerance can be achieved through redundancy. Redundancy can be spatial or temporal. In temporal redundancy, i.e., checkpoint-restart, an application is restarted from an earlier checkpoint or recovery point after a fault. This may result in the loss of some processing and applications may not be able to meet strict timing targets. In spatial redundancy, many copies of the application execute on different processors concurrently and strict timing constraints can be met. But the cost of providing fault tolerance using spatial redundancy is quite high and may require extra hardware.

4.3 A mechanism for prioritizing query results for delay tolerance

To overcome the problem of disconnection, a continuous query processing system for intermittently connected mobile networks has been proposed **[6].** It makes the use of a delay-tolerant continuous query processor distributed across the mobile hosts. In mobile ad-hoc networks, the mobile nodes i.e. clients and servers may get disconnected. It is necessary that mobile nodes make the best possible use of connectivity when it possible due to the limited and variable bandwidth.

To be connected to the network, a node must be able to hear the transmission of at least one other node on the network and have sufficient power to function. A mechanism for prioritizing query results[6], guarantees enhanced accuracy and reduced delay in query processing. The conventional continuous query processors send the results of continuous queries instantaneously over the network, whereas the delay tolerant distributed query processor stores them in an output buffer.

4.4 Processing Continuous Range Queries with Spatiotemporal Tolerance

The efficient processing of continuous range queries over mobile objects is another technique. It proposed the definition of spatiotemporally tolerant CRQs [7] which are intuitive, well-defines query semantics in spite of data uncertainty. The algorithms can compensate all sources of uncertainty and consistently provide results within the given tolerance. Tolerant CRQs offer applications a lot of flexibility to specify their accuracy requirements precisely. They can specify the acceptable error in both space and time dimensions, independently for both sides of the query boundary. This approach effectively minimizes the reporting costs, as reports are sent only if an mobile object actually affects the query result. The energy required for receiving initialization messages is easily amortized by saving report messages throughout the query's lifetime.

4.5 Fault tolerance using cooperative caching

Query processing method that uses cooperative caching[8] makes the data items satisfying a query flow to its originator. To cope with communication bandwidth and storage constraints, this method prioritizes the data-items in terms of their value, as reflected by supply and demand. Performing Simulations based on real-life mobility traces. This approach proves to be better than existing cooperative caching strategies and an existing mobile sensor network algorithm.

The model using cooperative caching uses MARKET algorithm [8] for querying mobile P2P databases. a novel strategy is used in MARKET for a mobile peer to prioritize the reports based on their relevance. The relevance of a report depends on its size, demand (how many peers are querying it), and supply. Queries are disseminated to enable the estimation of demand. A machine learning algorithm, called MALENA, is used to enable the estimation of the supply.

4.6 Log Based Recovery with Low Overhead for Large Mobile Computing Systems

The scheme proposed in this paper employs sender based message logging along with movement based checkpointing to reduce the number of checkpoints taken by a mobile host[9]. The mobility of a node is used for deciding when it needs to take a checkpoint. The storage at a Base Transceiver Station is utilized to store the checkpoints and message logs of the mobile hosts. Also sender based logging scheme avoids the extra copying of the message to the BTS. If the message logging is combined with the underlying communication protocol, then no extra overhead is placed for logging the messages at the sender BTS.

4.7 Distributed Lock Management for Mobile Transactions

A new lock management scheme is suitable for fault tolerance in the mobile computing environment [10] in which a read unlock for an item to be executed at any copy site, regardless of whether that site is different from the copy site in which the lock is set, is allowed. The lock management scheme for the mobile computing environment, utilizes the presence of replicated copies of data items to reduce the read unlock message cost incurred by the mobility of transaction hosts over fixed networks.

4.8 Fault-Tolerance in Distributed Query

Processing

A publicly available distributed query processing system for the grid [11] can be enhanced to support fault-tolerance. It describes the implementation of a rollback-recovery protocol and presents measurements of the cost of both protocol overheads and recovery. It shows that the implementation can exhibit low overhead and can yield significant performance improvements through recovering and continuing after failure.

4.9 Efficient Recovery Scheme for Mobile Computing Environment

An efficient recovery scheme [12] for mobile computing environment is based on message logging and independent checkpointing. It is seen that with the message logging and periodic checkpointing, asynchronous recovery can be achieved even in case of multiple and concurrent failure occurrences. For the management of recovery information, such as checkpoints and message logs, the movement-based scheme is given. The Mobile Host (MH) carrying its recovery information to its current Mobile Service Station (MSS) can recover instantly in case of a failure.

The MSSs visited by the MH have to experience high failure-free cost to transfer the recovery information and access the stable storage. On the other hand, if the recovery information is dispersed over a wide range of cells, the recovery cost can be too high. The movement-based scheme considers both these costs. While the MH moves within a certain range, recovery information of the MH is not moved.

But, if the MH moves out of the range, it transfers the recovery information to nearby host. As a result, the scheme controls the transfer cost as well as the recovery cost. It is costly to maintain the exact answer of continuous spatial queries.

5. CONCLUSION

In a mobile computing environment, due to mobility the nodes may be connected or not for the intact time. Therefore, query processing in mobile computing environment suffers from a number of constraints like disconnection, variability in bandwidth, power and resource problems, security risks and a lot. Replicating data at several sites is a powerful mechanism which can provide fault tolerance by improving performance of query processing. Use of redundancy proves to be good for transient type of faults. But the cost of providing fault tolerance using spatial redundancy is quite high and may require extra hardware. Delay tolerant distributed query processing is useful for faults that occur due to disconnections. To cope with communication bandwidth and storage constraints, the method of prioritizing the data-items in terms of their value, as reflected by supply and demand proves to be best.

After the analysis of various fault tolerant strategies, the presented methods are good in terms of power consumption, cost of query processing.

6. REFERENCES

1. R. Jain and N. Krishnakumar, "An Asymmetric Cost Model for Query Processing in Mobile Computing Environments", Wireless Information Networks, The Kluwer International Series in Engineering and Computer Science Vol. 351, pp 363-377, 1996.

2. Hocine Grine, Thierry Delot, Sylvain Lecomte, “Adaptive Query Processing in Mobile Environment”, MPAC ’05, MPAC '05 Proceedings of the 3rd international workshop on Middleware for pervasive and ad-hoc computing, ACM ,pp 1-8, 2005.

3. J. M. Hellerstein, M. J. Franklin, S. Chandrasekaran, A. Deshpande, K. Hildrum, S. Madden, V. Raman, and M. A. Saha, “ Adaptive query processing: Technology in evolution”, IEEE Data Engineering Bulletin, issue 23, vol.2, pp 7–18, 2000.

4. Salman Abdul Moiz, Dr. Lakshmi Rajamani, “Replication Strategies in Mobile Environments”, BIJIT– 2010, Vol. 2, No. 1, pp 170-173, 2010.

5. Rachit Garg, Praveen Kumar, “A Review of Fault Tolerant Check pointing Protocols for Mobile Computing Systems”, International Journal of Computer Applications, Vol 3, No.2, pp 8-19,June 2010.

6. T. P. Andamuthu and Dr. P. Balasubramine, “A Delay-Tolerant Distributed Query Processing Architecture for Mobile Environment”, International Journal of Computer and Information Engineering, volume 4, issue 2, pp: 86 – 90, 2010.

7. Tobias Farrell, Kurt Rothermel, and Reynold Cheng, “Processing Continuous Range Queries with Spatiotemporal Tolerance”, IEEE Transactions On Mobile Computing, Vol. 10, No. 3, March 2011.

8. Bo Xu, Fatemeh Vafaee, Ouri Wolfson,“In-Network Query Processing in Mobile P2P Databases”, ACM GIS 09, pp 4-6, November 2009.

9. Jin Jing, Omran Bukhres, Ahmed Elmagarmid, “Distributed Lock Management for Mobile Transactions”, Proceedings of 15th International conference on Distributed Computing Systems, PP. 118- 125, 1995.

10. Parmeet kaur Jaggi, Awadhesh Kumar Singh, “Log based recovery with low overhead for large mobile computing systems”, Journal of Information Science and Engineering, Vol. 29, No. 5, pp 969-984, September 2013.

11. Jim Smith, Paul Watson, “Fault-Tolerance in Distributed Query Processing”, Database Engineering and Application Symposium, IDEAS 2005. 9th International, pp 329 – 338, July 2005.

12. Taesoon Park, Namyoon Woo and Heon Y. Yeom, “An Efficient Recovery Scheme for Mobile Computing Environment”, Proceedings, 8th international Conference on Parallel and Distributed Systems, pp 53 – 60, June 2001.

13. J. Hwang, M. Balazinska, A. Rasin, U. Cetintemel, M. Stonebraker, and S. Zdonik, "High-availability algorithms for distributed stream processing", In Proc.21st International Conference on Data Engineering (ICDE), 2005.

14. Ming-Feng Chang,Yi-Bing Lin and Shu-Chin Su, "Improving the Fault Tolerance of GSM Networks", IEEE Network , issue 1, vol. 12, pp 58-63, January/February 1998.