Volume 4, Issue 3, March 2014 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Survey on Distributed Storage Using a Mobile Infrastructure

Mr. Mahesha H.S, Mr. Parjanya C.A
M.tech Student
East West Institute of Technology,
Dept of CSE
Bangalore, Karnataka, India

Mr. Prasanna Kumar M
Assistant professor
East West Institute of Technology,
Dept of CSE
Bangalore, Karnataka, India

Abstract— As the increasing amount of data is collected in mobile wireless networks for emerging pervasive applications, data-centric storage provides energy-efficient data dissemination and organization. One of the approaches in data centric storage is that the nodes that collected data will transfer their data to other neighbouring nodes that store the similar type of data. However, when the nodes are mobile, type-based data distribution alone cannot provide robust data storage and retrieval, since the nodes that store similar types may move far away and cannot be easily reachable in the future. In order to minimize the communication overhead and achieve efficient data retrieval in data-centric mobile environments, we propose a fully distributed neighbourhood prediction scheme that utilizes past node trajectory information to determine the near likely node in the future as the best content distribute. Distributed computing is a science which solves a large problem by giving small parts of the problem to many computers to solve and then combining the solutions for the parts into a solution for the problem. As in any traditional distributed setting, reliable storage of data with concurrent read/write (query/update) accesses still appears to be a challenging problem in mobile ad hoc networks.

Keywords— Distributed storage, distributed systems, Reliable Data Storage, Replication, mobile ad hoc network (MANET), mobile storage systems

I. Introduction

Existing cloud service providers use centralized data centres to provide computation and storage services to the users.
Combining networking and storage has created a platform with numerous possibilities allowing Distributed Storage Systems (DSS) to adopt roles vast and varied which fall well beyond data storage. Networking infrastructure and distributed computing share a close relationship. Distributed systems designed to operate on the Internet need to cope with a potential user base numbering in the millions. To accommodate a large user base, distributed systems are employing more scalable decentralised Peer-to-Peer architectures over conventional centralised architectures. Distributed systems which operate across a public network need to accommodate for a variety of potentially malicious behaviour from free-riding to Denial of Service (DoS) attacks.
A distributed computer system consists of multiple software components that are on multiple computers, but run as a single system. The computers that are in a distributed system can be physically close together and connected by a local network, or they can be geographically distant and connected by a wide area network. A distributed system can consist of any number of possible configurations, such as mainframes, personal computers, workstations, minicomputers, and so on. The goal of distributed computing is to make such a network work as a single computer.

Distributed systems offer many benefits over centralized systems, including the following:
Scalability: The system can easily be expanded by adding more machines as needed.
Redundancy: Several machines can provide the same services, so if one is unavailable, work does not stop. Additionally, because many smaller machines can be used, this redundancy does not need to be prohibitively expensive.
Distributed computing systems can run on hardware that is provided by many vendors, and can use a variety of standards-based software components. Such systems are independent of the underlying software. They can run on various operating systems, and can use various communications protocols. Ensuring data persistence is a major challenge in designing a mobile storage system because of various reasons. Due to their arbitrary movement patterns, mobile devices can leave or join networks non-deterministically, making data persistence challenging. Mobile devices can shut down or fail unexpectedly. Also, mobile devices may delete some cached data contents due to lack of sufficient storage space. Wireless communications among mobile devices are error-prone. Furthermore a lot of applications in this situations are time dependant and, therefore, their connections be supposed to be perform not simply appropriately, however as well contained by their time limits. In a Distributed Database System, Data is repeatedly replicated to get better consistency and accessibility. Though one significant subject to be measured as replicating Data is the accuracy of the Replicated Data. As Nodes in MANETS are Mobile and contain some degree of Power, disentanglement might happen often, cause a group of Network separation. Mobile hosts cannot access data, which is obtainable in Mobile hosts in one division in another detachment. The subject connected to Data Replication in MANETS Databases is as.

Server Power Consumption: Servers in MANETS run on battery Power. If Servers have a small Power remains and if it replicated with numerous often-contact Data objects, then the numerous Data access requests for these hot Data will consume its Power.

Server Mobility: Because of its Mobility, Servers may from time to time shift to a position where they might not be getting in touch with by any other Servers or Clients.

Client Mobility: Client which inquiry Servers for Information is as well Mobile. Clients more often than not launch their contacts to the adjacent Servers to obtain a rapid reaction. The choice to Replicate Data substance at exacting Servers might be found on the way in frequencies of Data matter from those precise Servers. Therefore, the judgment to Replicate Data stuff at suitable Servers has to be dynamic and found on the current Network topology

Client Power: Clients in MANETS also run on battery Power. If Clients stay excessively extended for its communications fallout, it may drop its Power quickly. The Replication technique ought to be capable to Replicate Data matter at suitable Servers in such a way that a Client may be capable to contact its demanded Data substances from its adjacent Server, which has the smallest amount workload.

Time Critical Applications: Numerous MANET applications, like rescue Networks, military processes, are time dependant. Data Replication is supposed to be used to progress Data ease of access and performance of the System, thus falling the time to perform communications. Dealings with short time limit ought to be sent to the nearest Server, which contain the smallest amount workload for their implementation.

II. Related Work

a) The "classic" academic approach to providing highly available data objects, the state-machine approach , consists in synchronizing a set of server replicas, each hosting a copy of the data, to handle all updates and queries in the same way ("write all – read one"). This approach provides perfect guarantees in theory, yet is expensive to implement even in wired networks. In particular, the atomic broadcast protocol ensures that each server replica obtains all updates and in the same order, but it comes with a prohibitively large overhead. Clearly, with an underlying highly dynamic adhoc network, the synchronism required for state machine replication becomes un-practicable. Even with weaker consistency guarantees (e.g., without ordering guarantees), state machine replication leads to poor performance in adhoc networks. Further sacrificing safety guarantees such as atomicity through the use of a probabilistic dissemination of updates (e.g., probabilistic broadcast) is indeed an option, yet the incurred overhead is not low enough to address a broad application context. A more drastic shift seems to be necessary.

b) Distributed storage systems such as OceanStore use Distributed Hash Tables (DHT) to manage the mapping from a collection of keys to values autonomously without any central control. DHT-based systems [5], [6], [7], [8],attempt to reduce the latency among nodes on wide-area networks by constantly updating their routing tables to improve the lookup speed. Phoenix, on the other hand, focuses on maintaining the desired number of copies of the data items in autonomous mobile nodes within a geographic region.

c) By introducing probabilities for the intersection of individual quorums, the construction rules for these quorums are relaxed, and more freedom is left for trading protocol overhead for reliability. As this smoother tradeoff has constituted the driving force behind probabilistic quorum systems, it also turns out that the resulting reduced determinism makes such an approach more viable than a strict approach for ad hoc networks. Problem for providing reliable storage in MANETs using probabilistic quorum system [9].

d) The Random Waypoint Mobility Model is used in many prominent simulation studies of ad hoc network protocols. For example a conference setting or museum. The Gauss-Markov Mobility Model also provides movement patterns that one might expect in the real-world, if appropriate parameters are chosen. In addition, this method also forces MNs to keep away from the edges of the simulation area While the Probabilistic Random Walk Mobility Model also provides movement patterns that one might expect in the real-world, choosing appropriate parameters for the probability matrix may be difficult. The Manhatten Model appears to create realistic movements for a section of a city since it severely restricts the travelling behaviour of MNs, MNs do not have the ability to roam freely without regard to obstacles and other traffic regulations. The Reference Point Group Mobility Model (RPGM) is a generic method for handling group mobility. These are the models proposed for mobile Adhoc networks to aid in designing efficient communication protocols for networks [10],[11].

e) Phoenix, on the other hand, is more deterministic in nature. Its primary goal is to maintain the desired level of data storage redundancy fairly accurately. While Phoenix does not support data updates directly, MANET databases that require data replication [12] can be built on top of Phoenix and rely on its protocol to handle issues related to node mobility and unreliable communication channels.

f) Using error correction coding in distributed systems for fault tolerance has not been widely applied in multi-agent systems. Current multi-agent platforms like JADE and COUGAAR use replication. Replication needs a lot of storage and bandwidth support, and information theoretic results tell us that we can do better. Some information theoretic ideas are used by the CFS cooperative storage and the OceanStore. In [13], Miletic and Dewilde consider reliability of a distributed storage system using an information-theoretic approach.

g) The push-pull method for data dissemination through the three way handshake of advertisement-request-code has been used previously in sensor networks with sensed data taking the place of code. Protocols such as SPIN [3] and SPMS [4] rely on the advertisement and the request packets being much smaller than the data packets and the redundancy in the network deployments which make several nodes disinterested in any given advertisement. However, in the data

dissemination protocols, there is only suppression of the requests and the data sizes are much smaller than the entire binary code images.
h) Reliable multicast in unreliable environments, such as ad-hoc networks, can be achieved by epidemic multicast protocols based on each node gossiping the message it received to a subset of neighbours [2]. These protocols give probabilistic guarantee for the update to reach all the group members. The probability is monotonically increasing with the fan-out of each node (the number of neighbours to gossip to) and the quiescence threshold (the time after which a node will stop gossiping to its neighbours). By increasing the quiescence threshold, the reliability can be made to approach 2, which is the basic premise behind all the epidemic based code update protocols in sensor networks – Deluge, MNP, and Freshet.
i) Wangand Li [14] proposed a set of runtime algorithms to ensure the availability of centralized services to all mobile nodes in networks that are frequently partitioned and is a non-power-aware, non-real-time-aware, and partition-aware system. An ideal MANET data replication strategy should be power-aware, real-time-aware, and partition-aware. Phoenix, as an underlying storage layer can help reduce the complexity of MANET databases by handling the node mobility and communication failures in a power-efficient manner, while guaranteeing the level of redundancy.

III. Existing System

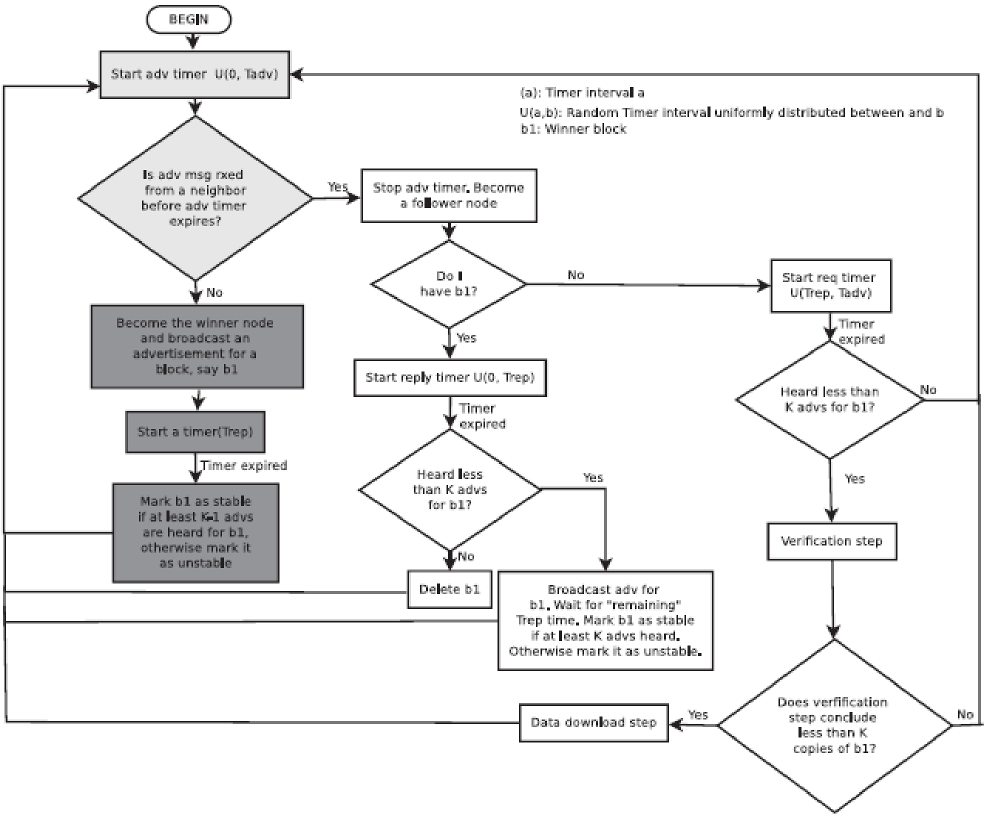

Fig 1: Flow chart showing the overview of phoenix. The shaded and un shaded blocks are the winner and the follower respectively.

When a node joins the network, it may or may not have information blocks. Each node starts an advertisement timer asynchronously—independent of the advertisement timers of other nodes. If the node does not receive any advertisement message before its advertisement timer expires, it assumes that it has won the round and broadcasts an advertisement for one of the blocks (say b1) that it currently possesses. If the node hears an advertisement message before its advertisement timer fires, it assumes that it has lost the current round and becomes a follower node. The follower node checks if it has b1—the block advertised by the winner node. If so, it starts a reply timer and when that timer expires, it broadcasts an advertisement for b1, if it has heard less than K advertisements for b1 in the current round. If it has heard K or more advertisements for b1, it deletes b1 because it knows that there are K or more nodes storing the copies of b1. On the other hand, if the follower node does not have b1, it starts a request timer and when that timer expires, it checks if it has heard K or more advertisements for b1. If so, it knows that there are K copies of b1 and, thus, does nothing and starts a new round. If it does not hear K or more advertisements for b1, it suspects that there may be less than K copies of b1 in the network. At this point, the node does not conclude that there are less than K copies because it may have missed some advertisements due to wireless communication failures. To ensure that there are indeed less than K copies of b1, it goes to the verification step, where it communicates with its neighbors to determine if they have b1. After the verification step, if the node finds that there are K or more copies of b1, it does nothing and starts a new round. Otherwise, it goes to the data download state, where it downloads b1 from one of its neighbors. Then, the node starts a new round.

IV. Proposed System

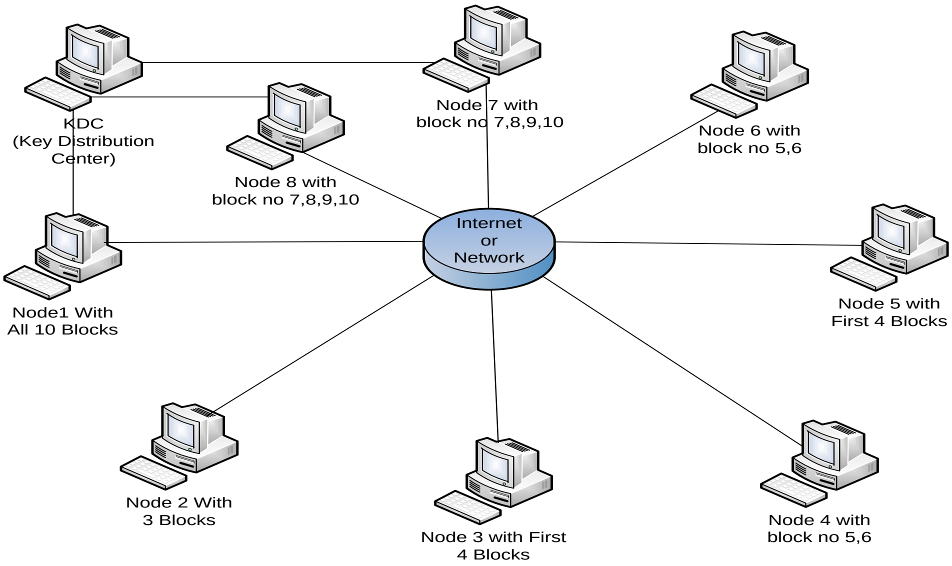

Fig2: content distribution among several nodes

The above diagram shows the practical implementation of our proposed approach with contents distributed among several nodes. In this approach we design a content distribution approach, for which we propose an algorithm called phoenix which is described below We are enhancing the application by proving the security to the algorithm by introducing key distribution center. KDC provides the service of generating and distribution public/ private keys to the set of group of nodes in the network. Each node when joining the group contacts the key distribution center and get the private key for the group using RSA Algorithm (Chinese Reminder Theorem).Then the node performs the following initiation.

Functional Requirements

The functional requirements for a system describe what the system should do. These requirements depend on the type of software being developed, the expected users of the software and the general approach taken by the organization when writing requirements. When expressed as user requirements, the requirements are usually described in a fairly abstract way. However, functional system requirements describe the system function in detail, its inputs and outputs, exceptions, and so on.

Functional requirements are as follows:

The proposed system should ensure the longevity of the stored data, even in the presence of autonomous node mobility.

The protocol proposed should scale well with the number of mobile devices and cached information content.

The protocol should be asynchronous

Non-Functional Requirement

Non-functional requirements, as the name suggests, are requirements that are not directly concerned with the specific functions delivered by the system. They may relate to emergent system properties such as reliability, response time and store occupancy. Alternatively, they may define constraints on the system such as the capabilities of I/O devices and the data representations used in system interfaces.

All nodes should be within the communication range of each other.

Does not require central control or global knowledge about the current network conditions.

Hardware Requirement

- Processor: P4 or More.
- Ram: **256 MB** or More.
- Disk Space: Minimum 800MB.

Software Requirements

- JAVA (JDK 1.6 or More)
- Netbeans IDE

V. Conclusion

Network-layer routing is an extensively researched area of ad-hoc networks. To improve network connectivity, researchers have proposed many mobility prediction schemes to predict the future availability of wireless links, for the purpose of building more stable end-to-end connections. In contrast, there are fewer studies of the effect of dynamic network topology on application-level issues, such as the provisioning of centralized service in wireless ad-hoc networks.

Although distributed applications are better suited for the adhoc network due to its peer-to-peer architecture, important network applications and services such as web servers, location information databases and network services. The main appeal of the wireless cellular and ad-hoc networks is that they allow both user mobility and unthread connectivity. However, user mobility poses significant challenges to network operations such as routing, resource management, and Quality of Service (QoS) provisioning. Node mobility leads to frequent disconnections of wireless links, hence the network topology changes dynamically. We propose a fully distributed grouping algorithm for a node to identify its stable group the group of nodes it has reliable connectivity with at run-time, using only local observations.

References

[1] Jun Luo, JeanPierre Hubaux, Patrick Th. Eugster, School of Computer and Communication Sciences Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland "PAN: Providing Reliable Storage in Mobile Ad Hoc Networks with Probabilistic Quorum Systems" June 1–3, 2003, 5005-67322

[2] J. Luo, P. T. Eugster, and J. P. Hubaux, "Route driven gossip: probabilistic reliable multicast in ad hoc networks," at the Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), pp. 2229-2239, 2003.

[3] J. Kulik, W. Heinzelman, and H. Balakrishnan, "Negotiation-based protocols for disseminating information in wireless sensor networks," Wireless Networks, vol. 8, no. 2/3, pp. 169-185, 2002.

[4] G. Khanna, S. Bagchi, and Y-S. Wu, "Fault tolerant energy aware data dissemination protocol in sensor networks," at the International Conference on Dependable Systems and Networks, pp. 795-804, 2004.

[5] S. Rhea, B. Godfrey, B. Karp, J. Kubiatowicz, S. Ratnasamy, S. Shenker, I. Stoica, and H. Yu, “OpenDHT: A Public DHT Service and Its Uses,” ACM SIGCOMM Computer Comm. Rev., vol. 35,pp. 73-84, 2005.

[6] A. Rowstron and P. Druschel, “Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems,” Proc. IFIP/ACM Int’l Conf. Distributed Systems Platforms Heidelberg (Middleware ’01), pp. 329-350, 2001.

[7] I. Stoica, R. Morris, D. Karger, M. Kaashoek, and H. Balakrishnan, “Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications,” ACM SIGCOMM Computer Comm. Rev., vol. 31, no.4, pp. 149-160, 2001.

[8] B. Zhao, J. Kubiatowicz, and A. Joseph, “Tapestry: A Fault-Tolerant Wide-Area Application Infrastructure,” ACM SIGCOMM Computer Comm. Rev., vol. 32, p. 81, 2002.

[9] D. Malkhi, M.K. Reiter, A. Wool, and R.N. Wright, “Probabilistic Quorum Systems,” Information and Computation, vol. 170, no. 2,pp. 184-206, 2001.

[10] T. Camp, J. Boleng, and V. Davies, “A Survey of Mobility Models for Adhoc Networks,” Wireless Comm. and Mobile Computing, vol. 2, pp. 483-502, 2002.

[11] X. Hong, M. Gerla, G. Pei, and C. Chiang, “A Group Mobility Model for Ad Hoc Wireless Networks,” Proc. ACM Second Int’l Workshop Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM), 1999.

[12] P. Padmanabhan, L. Gruenwald, A. Vallur, and M. Atiquzzaman, “A Survey of Data Replication Techniques for Mobile Ad Hoc Network Databases,” The VLDB J., vol. 17, no. 5, pp. 1143-1164, 2008.

[13] F. Miletic and P. Dewilde, “Data Storage in Unreliable Multi-Agent Networks,” Proc. Conf. Autonomous agents and Multiagent Systems, 2005.

[14] K. Wang and B. Li, “Efficient and Guaranteed Service Coverage in Partitionable Mobile Ad-Hoc Networks,” Proc. IEEE INFOCOM,2002.

