**COIT20257**

**Distributed Systems: Principles and Development**

**Term 2, 2020**

**Assignment 3**

**Assignment Title:**

**Some Theoretical Issues in Distributed Systems**

**Unit Coordinator, Lecturer, Tutor:**

**Wei Li**

**Submitted By:**

**Bishal Budhathoki**

**12116421**

**Table of Contents**

[**Question 1** 1](#_Toc53183605)

[**1.** **Difference between passive and active replication model.** 1](#_Toc53183606)

[**2.** **Difference between crash of server and Byzantine failure of server.** 1](#_Toc53183607)

[**3.** **If s of s+1 server crash. Explain whether the passive or active model is still fault-tolerant.** 2](#_Toc53183608)

[**4.** **If s of 2s+1 server has byzantine faults, explain whether the passive or active model is still fault-tolerant.** 2](#_Toc53183609)

[**Question 2** 2](#_Toc53183610)

[**1.** **Explain hierarchical structure/arrangement of DNS servers.** 2](#_Toc53183611)

[**2.** **If Recursive Server-controlled Navigation used, explain Workflow when client sends a name resolution request.** 3](#_Toc53183612)

[**3.** **Multi-threading strategy used in DNS server in Java.** 3](#_Toc53183613)

[**Question 3** 3](#_Toc53183614)

[**1.** **What is public and private key? Given a public key, is it possible to calculate its private key?** 3](#_Toc53183615)

[**2.** **Assume that Alice’s public key is available on a web site, describe the simplest way that Bob sends a secrete message to Alice by using Alice’s public key.** 4](#_Toc53183616)

[**3.** **Problem caused by using simple public key.** 4](#_Toc53183617)

[**4.** **Describe why digital certificate can solve the problem in step (2).** 5](#_Toc53183618)

[**Question 4** 5](#_Toc53183619)

[**References** 9](#_Toc53183620)

# **Question 1**

## **Difference between passive and active replication model.**

* The difference between passive and active replication model are given below:

|  |  |
| --- | --- |
| **Passive replication model** | **Active replication model** |
| 1. Passive replication model has only one server as a primary backup where clients requests are processed by that server (Schafer, et al., 2018). | 1. Active replication model has many servers where each client requests are processed by all the servers (Santos & Sousa, 2018). |
| 1. It cannot handle the arbitrary failures. | 1. Active replication can handle the arbitrary failures. |
| 1. In the time of failure in the system, passive replication model’s response time is late. | 1. The active replication model response time is faster than passive model in case of failure. |
| 1. For the optimization, passive model send “reads” for backup. | 1. For the optimization, active model send “reads” in specific RM. |
| 1. Crash: Replaced by one of many backups. | 1. Crash: No effect on performance. |
| 1. Byzantine Failure: Cannot survive and overhead is costly. | 1. Byzantine Failure: Can tolerate where frontend collects then compare the replies that it receives. |

## **Difference between crash of server and Byzantine failure of server.**

* Differences between crash of server and byzantine failure of server are as follow:

|  |  |
| --- | --- |
| **Crash of server** | **Byzantine failure of server** |
| 1. Crash of server occur in distributed system when the node has an omission failure and system stops to respond. | 1. Byzantine failure of server occurs when a server produces arbitrary response on random time interval (Bonomi, et al., 2016). |
| 1. The system completely stops and stop the response. | 1. System responds but don’t form respective task and produces random responses. |
| 1. In the crash failure, the server is working correctly until it stops responding. | 1. The server is working in byzantine failure but the responses and internal working is not correct. |
| 1. If a server crash, then the whole system is down, and no other server can continue from the point of failure. | 1. If a server faces Byzantine failure, the other server it communicates with will get true value and some will get false value (Tran, 2020). |

## **If s of s+1 server crash. Explain whether the passive or active model is still fault-tolerant.**

* If s of s+1 server crash, the passive model can tolerate the failure of the server. In the time of s+1 server crash, the passive replication model can acknowledge each other server as it has centralized server serves as primary backup. If s of s+1 server crash then the 1 will remains to supply the service to the clients.

## **If s of 2s+1 server has byzantine faults, explain whether the passive or active model is still fault-tolerant.**

* If s of 2s+1 server has byzantine faults, the active model is still fault-tolerant. In 2s+1 severs, s represents the number of faulty nodes. In the time of execution, one replica is used as primary and other is played as backups. At the time of execution, 2s+1 server active model can tolerate byzantine failure by giving the backups and signatures which validates along with verification of messages from different other servers.

# **Question 2**

## **Explain hierarchical structure/arrangement of DNS servers.**

* The hierarchical structure of DNS servers is like inverted tree directory where it roots of every DNS is dot. The DNS stands for Domain Name System where the IP address are converted into the domain name (Dooley & Rooney, 2017). The DNS servers are web servers which helps to translate the domain name into IP address given by users in the URL of web browser. The organizational hierarchy root zone has extension like .com, .net, .org and the geographic hierarchy root zone has extension like .au, .uk, .fr etc.

## **If Recursive Server-controlled Navigation used, explain Workflow when client sends a name resolution request.**

* Recursive server-controlled navigation is used to return the attributes from the different part of naming database where domain names are stored (Steadman & Scott-Hayward, 2019). When a client sends a name resolution request, the request start at one side of server and communicate to another server but in the other server name is not recognized which was send by the client. So, the server doesn’t have name to be resolved. To resolved the name resolution, it will again request to other single server using recursive method of continuing the search in other severs and also looks for a peer that stores a prefix of the name, until name is found or resolved.

## **Multi-threading strategy used in DNS server in Java.**

* I will use socket programming from java for multi-threading strategy which will be used in DNS server. The socket programming in java helps to communicate the two servers with each other. DNS server are used for communicating the server through domain name using the IP address and the socket programming multi-threading strategy is best for DNS server where it facilitates the connection between client and server using socket connection. To communicate over socket connection, input and output data are used in the streams. And each end point of socket can communicate, no other medium is required.

# **Question 3**

## **What is public and private key? Given a public key, is it possible to calculate its private key?**

* Public key is a sharable key and a type of lock that works along with encryption algorithm this means public key is distributed widely (Zhou, et al., 2020). Public key converts message to unread able format ("Difference Between Public Key and Private Key in Cryptography - Pediaa.Com", 2020). Private key is the type of lock that works along with decryption algorithm and only one authorized person has this key. For example: if the data is encrypted using public key every person with public can view the data but if the data is encrypted using private key the only authorized person can view the data because private key only accessible to authorized person.

Technically it is possible using the algorithm for producing public and private key pairs. But it shouldn’t be possible to derive the private key by having public key as it can cause security risk. So, with private key one can easily create public key but it is not possible to create private key in reverse.

## **Assume that Alice’s public key is available on a web site, describe the simplest way that Bob sends a secrete message to Alice by using Alice’s public key.**

* This can be achieved using an encryption algorithm by Diffie-Hellman. It is a key exchange protocol. First, Bob write his message and the public key belonging to bob will be secret from the encryption algorithm through computational method where the data will be encrypted. Then, Bob can send message to Alice. For Alice, she needs to used that her private key along with the decryption algorithm which the Bob has used.
* So basically Bob and Alice agrees on 2 numbers prime, big number say q = 23 and base value b = 5. Bob then gets a secret number z = 2 and use the formula : b^z%q which gives Z= 25%23 = 2. Similarly Alice gets a secret number p =6 and use same formula where z is replaced by p giving P= 8. Both exchange there secret number then Bob performs the following: K = P^z%q = 8^2%23 = 18. Similarly, Alice does the same where P is replaced by Z, k = Z^p%q = 2^6%23 = 18. Now since both have equal and similar number they can communicate privately (Kaczanowski, 2020).

## **Problem caused by using simple public key.**

* The potential problem caused by using simple public key is the receiver cannot have idea about the sender details because anybody can send message to the receiver whose public key is public over the website like the example of Bob and Alice on above question. Eve Dropping could be a great issue here. Third party can gain private key of individual sender and receiver ("A Deep Dive on End-to-End Encryption: How Do Public Key Encryption Systems Work?", 2020).

## **Describe why digital certificate can solve the problem in step (2).**

* Digital certificate is given by Certificate Authority that takes a certain fee from either the user or the receiver for generating the certificate for that individual. It contains Individual Name, name of certificate authority, Unique certificate number, expiry date if any, unique private key and digital signature of Certificate Authority ("The Basics of Cryptography and Digital Certificates", 2020). The digital certificate can solve the problem by:
  + Authenticating the user identity and guarantee that the individual is not a third party,
  + unauthorized access of data.
  + Certificate is stored on Hard drive
  + Certificate only contains public data like name and public key.
  + Difficult to forge the certificate.

# **Question 4**

1. If there is no concurrency control, transactions T and U may perform the following interleaving operations on objects a1, a2 and a3. What problem can be caused by the operations? Justify your answer. T: read(a2); U: read(a3); U: read(a2); T: read(a1); T: write (a2, a2-25); T: read (a3); U: write (a2, a2+33); T: write (a1, a1+52); U: write (a3, a3-26)
2. After executing the operation T: read(a2); U: read(a3); U: read(a2); T: read(a1); T: write (a2, a2-25); T: read (a3); U: write (a2, a2+33); T: write (a1, a1+52); U: write (a3, a3-26)

* T set the following values in its memory.
  + a2 = 106
  + a1 = 111
* U set the following values in its memory.
  + a3=125
  + a2=106

Next operation executes by T is T: write(a2, a2-25)

That is a2=81

The new value of a2 is not remembered by the U so when U executes the operation U:(a2,a2+33) , it uses old value of a2.

1. What requirement must be satisfied in order to avoid the problem?

To avoid the problem there are following requirements and they are:

* Two-phase locking Protocol

Two-phase locking protocol insure the permission to read or write the particular data.

* Time stamp ordering Protocol

A timestamp provides a tag that will be attached to particular transaction or particular data item and which denotes a certain time on which the transaction or the data item should be read or write.

* Multi version concurrency control

Multi version techniques store old data item to increase concurrency.

* Validation concurrency control

The effective approach is based on the assumption that the large number database operations do not conflict.

1. When the above requirement in question (1) is satisfied, what would be the correct values of a1, a2 and a3 after T and U commit?

The correct values of a1, a2 and a3 after T and U commit would be

a1=163

a2=114

a3=99

1. Give an example of possible interleaving operations that can produce the correct values of a1, a2 and a3. Note: no marks is given to this question if the operations are not interleaved.

T: read(a2); write(a2, a2-25); read(a1); read(a3); write(a1, a1+52)

U: read(a3); write(a3, a3-26); read(a2); write(a2, a2+33)

|  |  |
| --- | --- |
| T | U |
| T:read(a2) | U:read(a3) |
| T:write(a2,a2-25) | U:write(a3,a3-26) |
| T:read(a1) | U:read(a2) |
| T:read(a3) | U:write(a2,a2+33) |
| T:write(a1,a1+52) |  |

1. The following is an example to use exclusive locks to solve the problem in question (1). Give your explanation why it can solve the problem.

Exclusive locks can solve the problem because in the mechanism the lock is acquired on a data item to perform the operation. And the read locks are shared so that no data value is changed.

In given table, the T transaction lock the item a2 and read its data and does change on it.

Until T unlocks the item a2, transaction U can execute the read operation.

Because of which the exclusive locks solve the problem.

1. We rearrange T’s operations as follows and keep U’s operations unchanged. T: read(a2); read(a1); read (a3); write(a2, a2-25); write(a1, a1+52) If we use the locks as below, what problem would happen?

In the given table the problem that would arise is deadlock because both the transactions are waiting for the resources to get released.

In transaction T, it is waiting for the release of item a3 and in transaction U, it is waiting for the release of item a2. Since both the transaction have been waiting for the resources to get released from each other, so the given operations are in deadlock.

1. What will be the solution to the problem in question (6)? Give at least two different methods.

The solution to the problem is as follows:

* Wait-die

An older transaction is permitted to wait for a younger transaction, whereas a younger transaction requesting an item that is held by an older transaction is rejected and restarted in wait-die approach.

* Wound-die:

A younger transaction is permitted to wait on an older transaction, whereas an older transaction requesting an item that is held by a younger transaction is rejected in wound-die approach.

It is exactly opposite of wait-die approach.

# **References**

* Bonomi, S., Pozzo, A. D., Potop-Butucaru, M. & Tixeuil, S., 2016. Optimal mobile byzantine fault tolerant distributed storage. *PODC '16: Proceedings of the 2016 ACM Symposium on Principles of Distributed Computing,* pp. 269-278.
* Dooley, M. & Rooney, T., 2017. *Introduction to the Domain Name System (DNS).* 1 ed. s.l.:Wiley-IEEE Press.
* Santos, A. A. & Sousa, M. d., 2018. Replication Strategies for Distributed IEC 61499 Applications. *IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society.*
* Schafer, D. R., Rothermel, K. & Tariq, M. A., 2018. Replication Schemes for Highly Available Workflow Engines. *IEEE Transactions on Services Computing,* pp. 1-1.
* Steadman, J. & Scott-Hayward, S., 2019. DNSxD: Detecting Data Exfiltration Over DNS. *IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN).*
* Zhou, Y. et al., 2020. Public Key Encryption with Keyword Search in Cloud: A Survey. *Entropy,* 22(4), p. 421.
* Tran, A. (2020). *A Cursory Introduction to Byzantine Fault Tolerance and Alternative Consensus*. Medium. Retrieved 9 October 2020, from https://medium.com/@ancapalex/a-cursory-introduction-to-byzantine-fault-tolerance-and-alternative-consensus-1155a5594f18.
* Failure modes in distributed systems. Alvaro-videla.com. (2020). Retrieved 9 October 2020, from <https://alvaro-videla.com/2013/12/failure-modes-in-distributed-systems.html>.
* Difference Between Public Key and Private Key in Cryptography - Pediaa.Com. Pediaa.Com. (2020). Retrieved 9 October 2020, from <https://pediaa.com/difference-between-public-key-and-private-key-in-cryptography/>.
* Kaczanowski, M. (2020). Encryption Algorithms Explained with Examples. freeCodeCamp.org. Retrieved 9 October 2020, from <https://www.freecodecamp.org/news/understanding-encryption-algorithms/>.
* A Deep Dive on End-to-End Encryption: How Do Public Key Encryption Systems Work?. Surveillance Self-Defense. (2020). Retrieved 9 October 2020, from <https://ssd.eff.org/en/module/deep-dive-end-end-encryption-how-do-public-key-encryption-systems-work>.
* The Basics of Cryptography and Digital Certificates. WhatIsMyIPAddress.com. (2020). Retrieved 9 October 2020, from https://whatismyipaddress.com/cryptography.