Chapter 6: Synchronization

Nguyen Ngoc Lam - 20162316

Friday 5th June, 2020

Contents

1 Give two examples to demonstrate the importance and the need of synchronization mechanism between processes in distributed systems

- Distributed System is a collection of computers connected via the high speed communication network.
- The hardware and software components communicate, coordinate their actions and share their resources to other nodes by message passing.
- There is need of proper allocation of resources to preserve the state of resources and help coordinate between the several processes.
- Example: If a distributed system is not synchronized, for example computer A is 10 minutes behind computer B and both computer is on the same timezone (say ICT) with computer B is synchronized with the standard time in timezone. At 7:00 am Hanoi time, computer A try to write something to the shared resources and register the timestamp as 7:00 Hanoi time, but since it is technically 7:10 am, the system will not let the user from A to write anything because the timestamp is invalid. In fact, users from computer A cannot write anything until the administrator synchronizes that machine. Timestamp invalidation can also caused lagging when using real time application or application that shared the resources to a lot of users like online games or chat applications.
- Benefit of synchronization through example: With synchronization, when playing online games for example, since most packages have to have a timestamp or a related time series to determine the player model action sequences, with synchronization, there can be less lag and enable better quality for player. In fact, nowadays, most online games required you to synchronized before playing.

2 Compare Network Time Protocol and Berkeley algorithm

Berkeley Algorithm	Network Time Protocol
Having 1 Master to be standard	Having 3 classes with class 1 is the highest accuracy and class 3 is the lowest
Master calculate the average time and discard outliers	Class 1 is the standard. Class 2 get time from Class 1 and Class 2 server. Class 3 get time from any server when they attempted connecting to one
Send time adjustment to clients	Send timestamp to be changed into
Round trip message	One way message to dictate time

3 What is the typical characteristic of synchronization algorithm for wireless networks

Since the primary concern of all wireless applications is energy conservation, the typical characteristics of synchronization algorithm for wireless networks are :

- It must carefully regiment its frequency of resynchronization, and avoid flooding.
- In addition, the algorithm cannot typically rely on a power-hungry source of real time such as GPS.
- Another characteristic of wireless networks is unexpected and possibly frequent changes in network topology. Thus, a CSA in a wireless medium must continue to function in the face of node failures and recoveries.
- Lastly, many applications in wireless settings are local in nature. That is, only nearby nodes in the network need to participate in some activity. Thus, a desirable property for a CSA is that it closely synchronizes nodes which are nearby, while possibly allowing faraway nodes to be more loosely synchronized.

4 What is the difference between physical synchronization and logical synchronization

Physical Synchronization	Logical Synchronization
Synchronize the exact timestamp	Synchronize by order of action
Must change the clock in each systems based on a standard	No need to change the clock
Maintain the same notion of time	Keep track of information pertaining to the order of events
Expensive to maintain	Inexpensive to maintain
Inherently inaccurate	Fairly inherently accurate

5 What are the update steps of counters to implement Lamport's logical clock

- Once the logical clock function has an event, it looks at the time on that event, and compares it to the time on the clock's process
- It chooses the larger of the options, and increments it arbitrarily

6 Answer questions based on an algorithm for the physical clock synchronization

a) Is the value of T_P calculated by the above formula absolutely accurate?

- Given that there is a constant delay on the medium, the value T_P can be consider absolutely accurate because RTT is double the one way time in that condition and T_P will be equal with the timer on server S at that time.
- In practice, the value T_P is not absolutely accurate but it can be consider as accurate enough to use.

b) Let δ be the deviation of time value and min the minimum time value it takes to transmit a message one-way. Determine the value δ using only 2 variables RTT and min

- If there is no deviation of time value, $RTT = 2 * min$
- The deviation of time value is the different between RTT and two time the min. So the equation would be:

$$\delta = RTT - 2 * min$$

- δ should be positive but it can be negative when you upgrade to a better system and/or better medium. When that happened, you need to update your min value.

7 Answer the following questions in regard of using the Vector Clock concept for enforcing causal communication

a) List two conditions the receiving process use to check whether the message satisfies causality.

$$\begin{cases} V_{Pj}[i] = V_S[i] - 1 \\ V_{Pj}[k] \geq V_S[k] \forall k \in [1, 2, 3, ..., n] - \{i\} \end{cases}$$

b) Vector clock values for 4 points $X1$, $X2$, $X3$, and $X4$

$$X1 = (0, 1, 0)$$

$$X2 = (1, 1, 0)$$

$$X3 = (2, 1, 1)$$

$$X4 = (2, 1, 2)$$

c) Message will be kept at the middleware level: Message b

8 What is a mutual exclusion algorithm in a distributed system

- Mutual exclusion: Concurrent access of processes to a shared resource or data is executed in mutually exclusive manner.
- Only one process is allowed to execute the critical section (CS) at any given time.
- In a distributed system, shared variables (semaphores) or a local kernel cannot be used to implement mutual exclusion.
- Message passing is the sole means for implementing distributed mutual exclusion.
- Distributed mutual exclusion algorithms is the family of algorithms that deals with unpredictable message delays and incomplete knowledge of the system state.

9 What is the drawback of the centralized algorithm for the mutual exclusion

- If the coordinator crashes, the entire system may go down with it since it is a single point of failure.
- If processes normally block after making a request, they cannot distinguish a dead coordinator from ”permission denied” since no message comes back.
- Bottleneck when deploy on the large system with a single coordinator (scalability)

10 What is the drawback of the distributed algorithm for the mutual exclusion

- Algorithm is suitable only for small group of processes.
- It is highly complex due to the need of identify all processes needed.

11 Propose a solution for the problem of lost token in Token Ring mutual exclusion algorithm

When a process (say) P1 wants to enter into its critical section, it sends request to the coordinator. If the coordinator retains the token, it then sends the token to the requesting process (P1). After getting the token, the process will send an acknowledgment to the coordinator, and enters the critical section. During the execution of the critical section, P1 will continually send an EXISTS signal to the coordinator at certain time interval, so that, the coordinator becomes acquainted that the token is alive and it has not lost. As a reply to every EXISTS signal, the coordinator sends back an OK signal to that particular process (P1), so that the process that is executing in its critical section (P1), gets to know that the coordinator is alive also. Now, suppose, the coordinator is not receiving the EXISTS signal from that process P1. Here two cases may appear:

1. The coordinator assumes that the token has lost. Then the coordinator will regenerate a new token and sends it to that process (P1) and again it starts executing its critical section.
2. The process P1 may crash or fail while executing in it critical section and consequently, the coordinator does not receiving any EXISTS signal from P1. Hence, the coordinator will identify it as a crashed process and update the ring configuration table and send the UPDATE

signal with update information to other processes to update their own ring configuration tables.

Again it may be the case, that the process P1 (which is currently executing in its critical section), is not receiving the OK signal form the coordinator. So, P1 would assume that the coordinator is somehow crashed. At this moment, the process P1 will become the new coordinator and complete the critical section execution. The new coordinator will send a message [COORDINATOR (PID)] to every other process, that it becomes the new coordinator and send the UPDATE signal with update information, to update the ring configuration tables maintained by all other processes.
The algorithm also overcomes the overhead of token circulation in the ring. If no processes in the ring want to enter in its critical section, then there is no meaning of circulating the token throughout the ring. Rather, in this approach, the coordinator will keep the token, until any other process requests it.

1. The algorithm guarantees mutual exclusion, because at any instance of time only one process can hold the token and can enter into the critical section.
2. Since a process is permitted to enter one critical section at a time, starvation cannot occur.

12 How many messages does the system need to vote the coordinator

When P3 starts the election after node P4 and P7 was broken, it will take the system a total of 10 messages to elect a new coordinator.

- The first round, P3 sent 4 ELECTION messages to P4, P5, P6 and P7. P3 would receive 2 ANSWER messages from P5, P6.
- The second round, P5 sent 2 ELECTION messages to P6 and P7 and received 1 ANSWER message from P6.
- The final round, P6 sent 1 ELECTION message to P7 and received 0 ANSWER message thus making it the coordinator.

13 Answer questions with a system the has N nodes and each node has a status table of other nodes and the whole system on election algorithm

a) How many messages do we need to vote the coordinator

$$number\_of\_messages = 2(N-i-1)+(i+1)N$$

b) When a broken node become working again

- That broken node first broadcast a WORKING message to each of the other nodes for update their tables. The other working nodes will send a acknowledge message back for that previously broken nodes to update its table.
- If the said node has higher id than the current coordinator, the coordinator will sent the ELECTION message to said node, otherwise, the system will continue to run

14 Answer questions regrading a mutual exclusion algorithm

a) How many messages does the system need to successfully let a process use SR.

- Suppose there are k REQUEST messages on the queue already $0 \leq k$ (no upper limit due to Pi can post multiple REQUEST messages). Pi will wait k RELEASE messages.
- Pi sends $N-1$ REQUEST messages and receives $N-1$ REPLY messages.
- The total amount of messages would be:

$$k + 2(N-1)$$

b) The improvement will cut all the unnecessary REPLY messages since the purpose of the REPLY was to reorder the priority queues of all processes to timestamp order. The new algorithms can cut at most $N-1$ REPLY messages. In the worst case scenario, when ts_i is the latest, it will perform like the original algorithms.