Weekly Report

Task

- Paxos Algorithm

Main Activities

1. Paxos algorithm

1.1 Introduction

In a distributed system, how to maintain consistency is a big problem。 Paxos Algorithm which was made by Lamport in 1990 is a most important algorithm to achieve a fault-tolerant distributed system.The algorithm assume a collection of processes that can propose values.Then it can ensure only one among the proposed values is chosen. And if a value has been chosen, then processes should be able to learn the chosen value. In order to make the system safety it requires two principles as follows:

- *Safety Principle*
  - Only a value that has been proposed may be chosen,
  - Only a single value is chosen.
- *Survive Principle*
  - Will eventually choose a proposed value,
  - If a value has been chosen, then processes should be able to learn the chosen value.

Paxos algorithm obtain these principles perfectly. As long as messages are not corrupted, the consensus algorithm could keep the distributed system safety regardless agents operate at arbitrary speed and messages can take arbitrarily long to be delivered, can be duplicated, and can be lost.

1.2 Basic-Paxos

There are tow roles in the consensus algorithm, they are performed by tow classes of agents: proposers and acceptors. In addition, a single process may act as more than one agent.

- *Proposer*

  Propose the initiator, process the client request, and send the client's request to the cluster in order to determine whether the value can be approved.

- *Acceptor*

  Proposed approvers are responsible for handling the proposals received, and their response is a vote. Will store some status to decide whether to receive a value.

In a distributed system, if a server received a request from clients, it becomes a proposer. Relative to this server other servers become acceptors.Then this proposer is proposing its proposal through these phrase as follows

- *Get Log Id*

  Query majority acceptor for the maximum logID that they has been wrote by now. Then pick up the maximum result plus one as its new logID.

- *Generate ProposalID*

  According to paxos algorithm's requirement, proposalID has to be unique and monotonically increasing. Therefore, we often generate a new proposalID using timestamp together with the logID.

- *Prepare*

  (a) The proposer sends the ProposalID to a majority of acceptors which is not contain a real proposal.

  (b) If an acceptor receives a prepare request with proposalID greater than that of any prepare request to which it has already responded, then it responds to the request with a promise not to accept any more proposals numbered less than n and with the highest-numbered proposal that it has accepted.

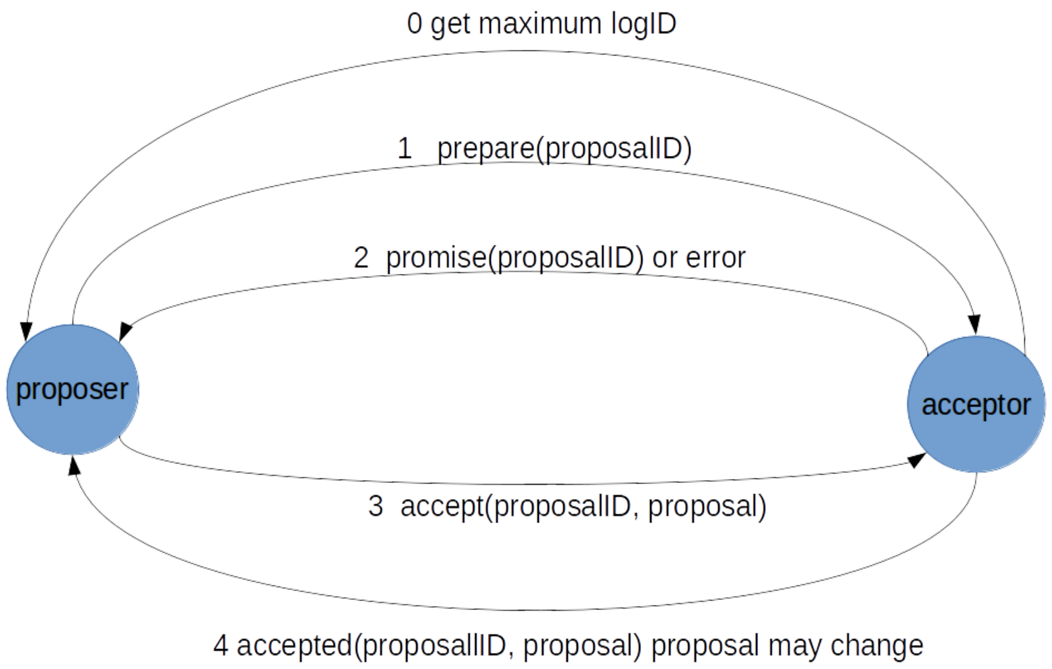
- *Accept*

(a) After proposer get to knew majority acceptors' responds, it can react according to the respond. If prepare fail, it will go back to the first phrase, otherwise it will propose its proposal together with proposalID to majority of acceptor.

(b) If an acceptor receives an accept request proposal and the proposalID has been responded to a prepare request, the acceptor will accept the proposal and send back a message of success.

(c) If the proposer receives majority with successful response, it will responds clients the result.

The above procedure can be shown in the following figure:


0 get maximum logID
1 prepare(proposalID)
2 promise(proposalID) or error
proposer
acceptor
3 accept(proposalID, proposal)
4 accepted(proposallID, proposal) proposal may change

1.3 Multi-Paxos

In the basic-paxos algorithm we have to execute prepare phrase and accept phrase, and in multi-paxos algorithm we do not need execute prepare phrase in every paxos process by selecting one leader which is the only one to propose.

- Leader Election

  Every server has a time slice. If no message was received in a time slice which indicate the current leader may fail, this server will launch a leader election tending to become a new leader.

Because the number of launching a leader election might greater than one, the leader election could be understand as a basic-paxos.

After a new leader was chosen, this leader could send proposals without prepare phrase.

2. Some Questions

- How to achieve a paxos algorithm in a real distributed system?
- What is the difference and relationship between paxos algorithm and other consensus algorithm such as raft?
- How to learn English?

References

[1] Leslie Lamport. Paxos Made Simple. ACM Transactions on Computer Systems 16, 2 (May 1998), 133-169.
[2] http://oceanbase.org.cn
[3] https://www.zhihu.com/question/19787937

Period: 2017.4.18 - 2017.4.25
Submitted by: jiacheng huang 2017.4.25