CS 7172
Parallel and Distributed Computation

Distributed Election

Kun Suo

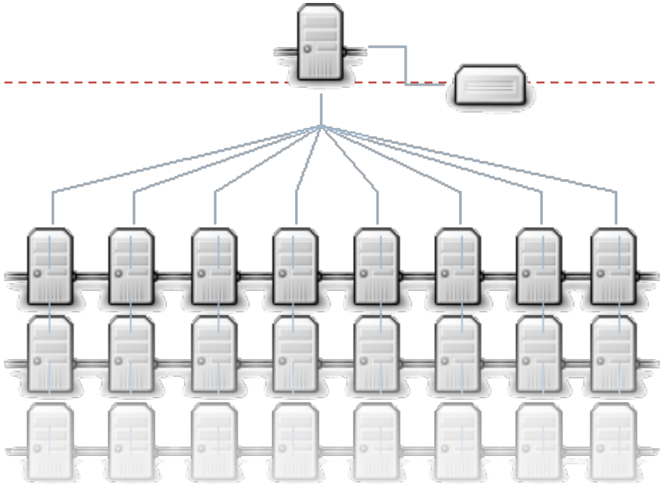
Computer Science, Kennesaw State University

https://kevinsuo.github.io/

Outline

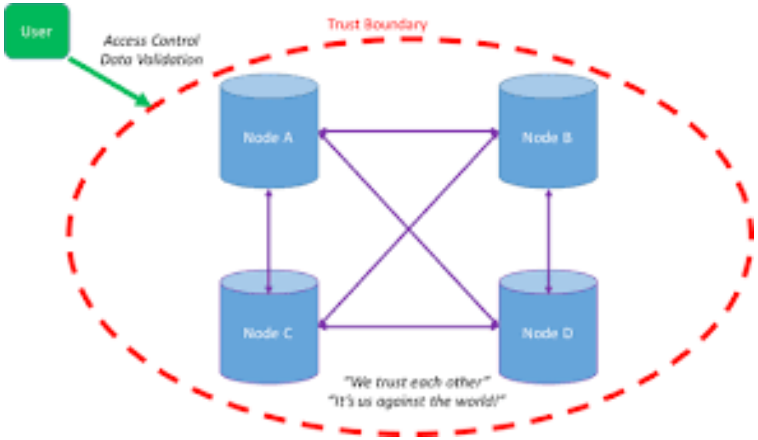
- Computer networks, primarily from an application perspective
- Protocol layering
- Client-server architecture
- End-to-end principle
- TCP
- Socket programming

Distributed Clusters



- For a cluster, how do multiple nodes work together and how are they managed?

- For example, in a database cluster, how to ensure that the data written is consistent on each node?



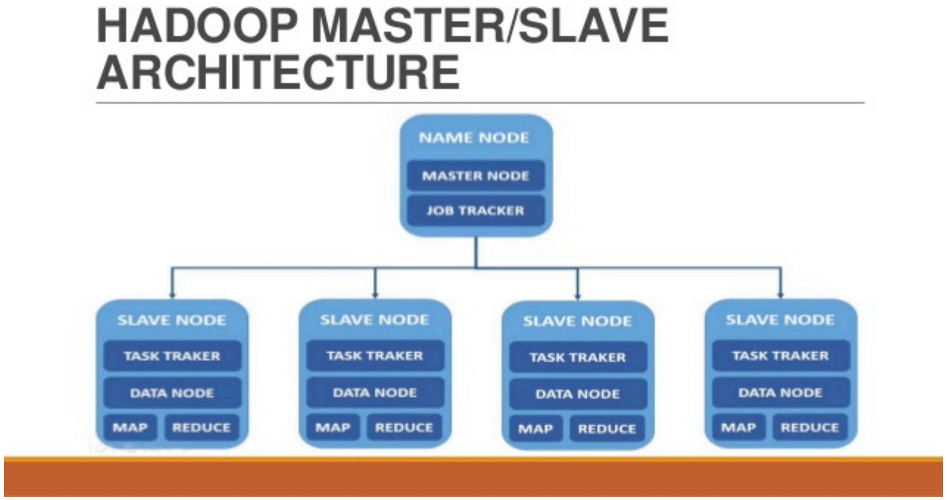
Leader in Distributed Systems

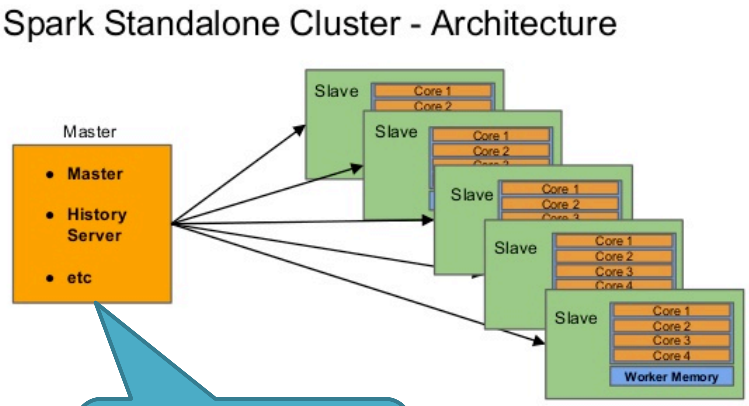
- Choose a "leader" to be responsible for scheduling and managing other nodes

- This "leader" is called the master node in the distributed, and the process of selecting "leader" is called distributed election in the distributed system

Why we need Distributed Election?

- Master node is so important in distributed system
  - Scheduling and managing other nodes

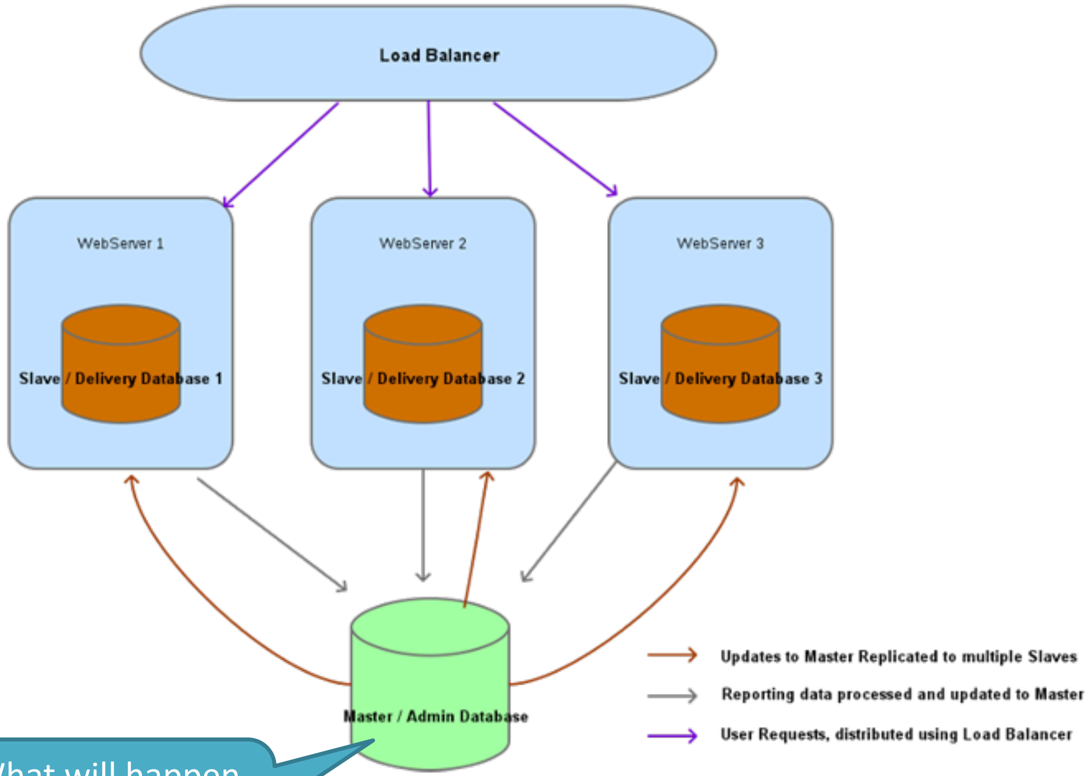




What will happen when the master node crashed?

Why we need Distributed Election?

- Failure of the master node in a distributed database cluster may cause inconsistent data on each node


Load Balancer
WebServer 1
WebServer 2
WebServer 3
Slave / Delivery Database 1
Slave / Delivery Database 2
Slave / Delivery Database 3
Master / Admin Database
Updates to Master Replicated to multiple Slaves
Reporting data processed and updated to Master
User Requests, distributed using Load Balancer

What will happen when the master node crashed?

1. Bully algorithm

- A method for dynamically electing a coordinator or leader from a group of distributed nodes.

- The node with the highest ID number from amongst the non-failed nodes is selected as the coordinator.

1. Bully algorithm

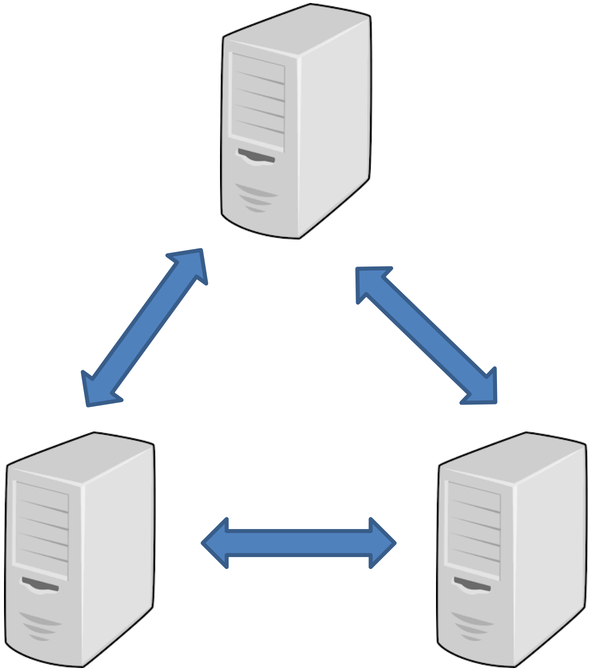
- Nodes have two types: normal nodes and master nodes.

- During initialization, all nodes are normal nodes, and have the right to become masters. However, after the election, only one node becomes the master node, and all other nodes are normal nodes.

- The master will be reelected if and only if the master node fails or loses connect with other nodes.

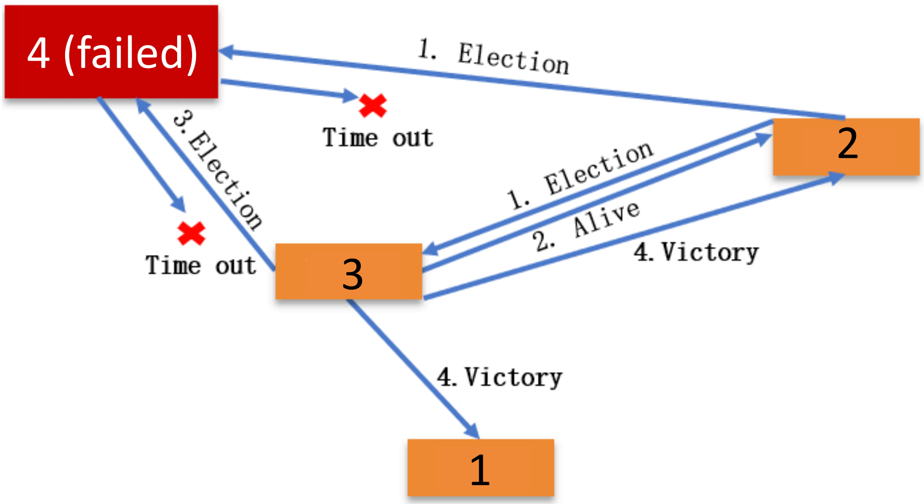
1. Bully algorithm

- During the election process of the Bully algorithm, three types of messages are needed:
    1. Election messages, which are used to initiate elections;
    2. Alive messages, which are responses to Election messages;
    3. Victory messages, which are sent by the master node that has been successfully elected.



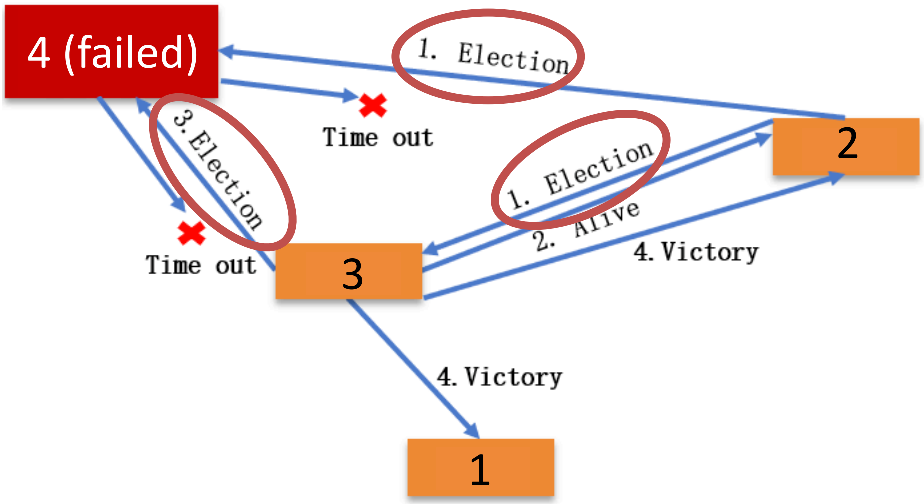
1. Bully algorithm

- How Bully algorithm works:
  1. Each node in the cluster judges whether its ID is the largest among the currently alive nodes. If so, it directly sends a Victory message to other nodes to notify it is the master node;



1. Bully algorithm

- How Bully algorithm works:
  2. If the nodes are not the one with the largest ID in the currently alive node, send an Election message to all nodes with a larger ID and wait for a reply from other nodes;

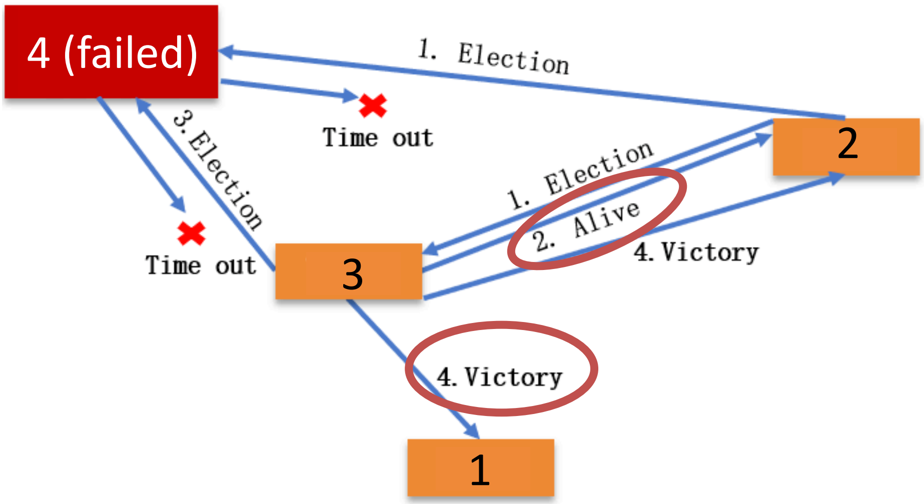


1. Bully algorithm

- How Bully algorithm works:

  3. If within the given time, the node does not receive the Alive message from other nodes, it considers itself to be the master node, and sends a Victory message to other nodes, and becomes the master node;

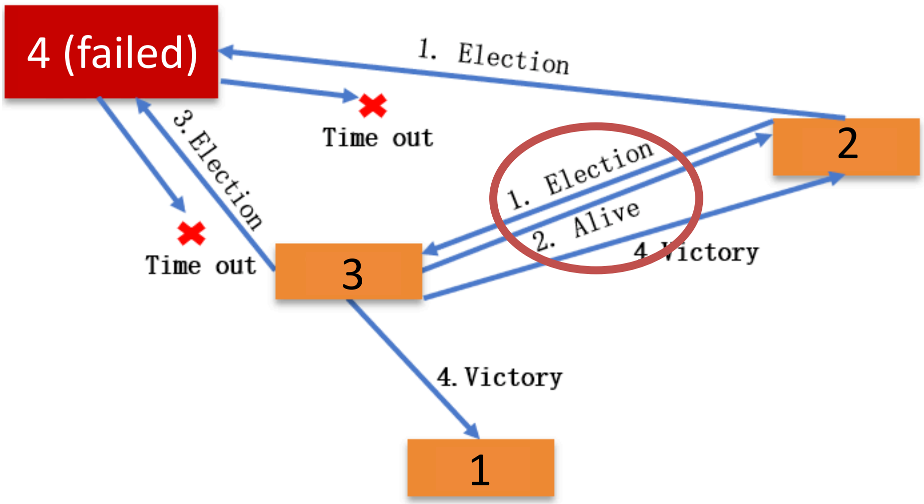
     If the node receives the Alive message from the node which has larger ID, then it waits for other nodes to send Victory messages;



1. Bully algorithm

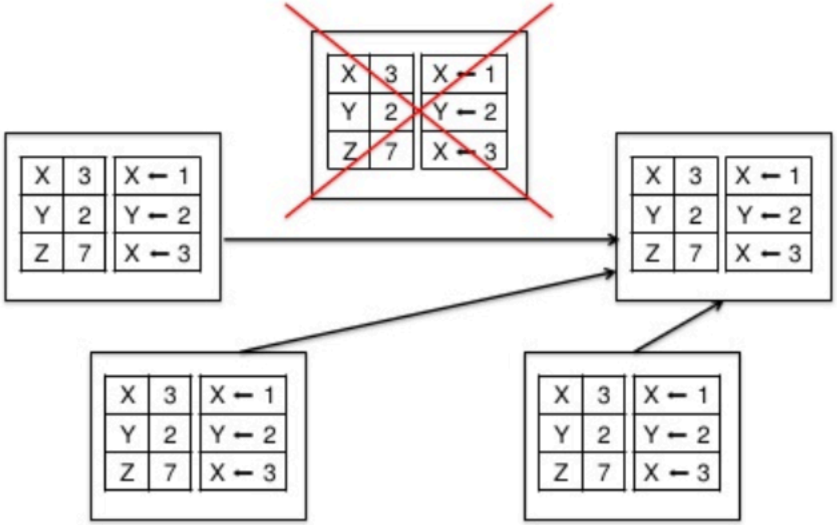
- How Bully algorithm works:

  4. If this node receives an Election message from a node with a smaller ID, it will reply with an Alive message to inform other nodes that I am larger than you and reelect the master.



Bully algorithm example in MongoDB

- How MongoDB deals with failure:
  - The node's last operation timestamp is used to represent the ID
  - The node with the latest timestamp has the largest ID, thus the live node with the latest timestamp is the master node



1. Bully algorithm

- Advantages:
  - Fast election speed
  - low algorithm complexity
  - simple and easy to implement (who lives and who has the largest ID is the master node)

1. Bully algorithm

- Disadvantages:
  - Each node needs to have global node information (all node IDs), so additional information needs to be stored.
  - New election is required when any new node that is larger than the current master node ID
  - Frequent switch over could happen when some nodes frequently join and exit the cluster

2. Raft algorithm

- Similar as the democratic voting
- The core idea is "the minority obeys the majority"
- The node with the most votes becomes the master node

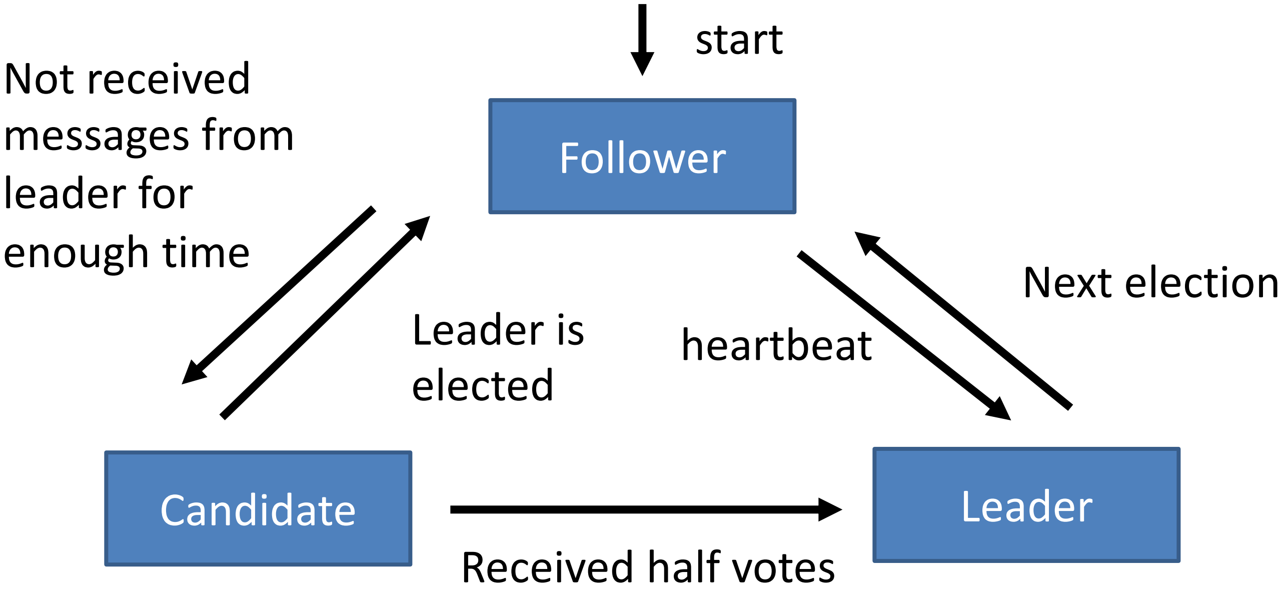
2. Raft algorithm

- Using the Raft algorithm, there are three types of roles for cluster nodes:
  - Leader: the master node, and there is only one leader at the same time, which is responsible for coordinating and managing other nodes;
  - Candidate: each node can become Candidate, and only the Candidate node can be selected as the leader;
  - Followers: can not initiate elections.

2. Raft algorithm

- How Raft algorithm works:

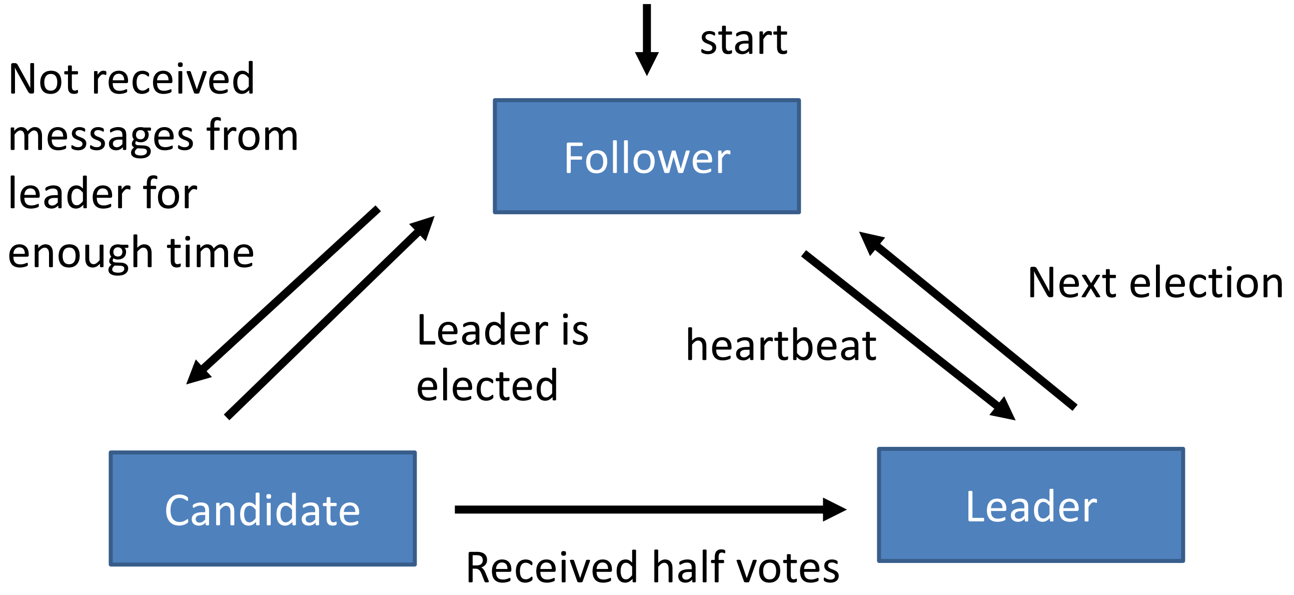
  1. Initialization, all nodes are in the Follower state.



2. Raft algorithm

- How Raft algorithm works:

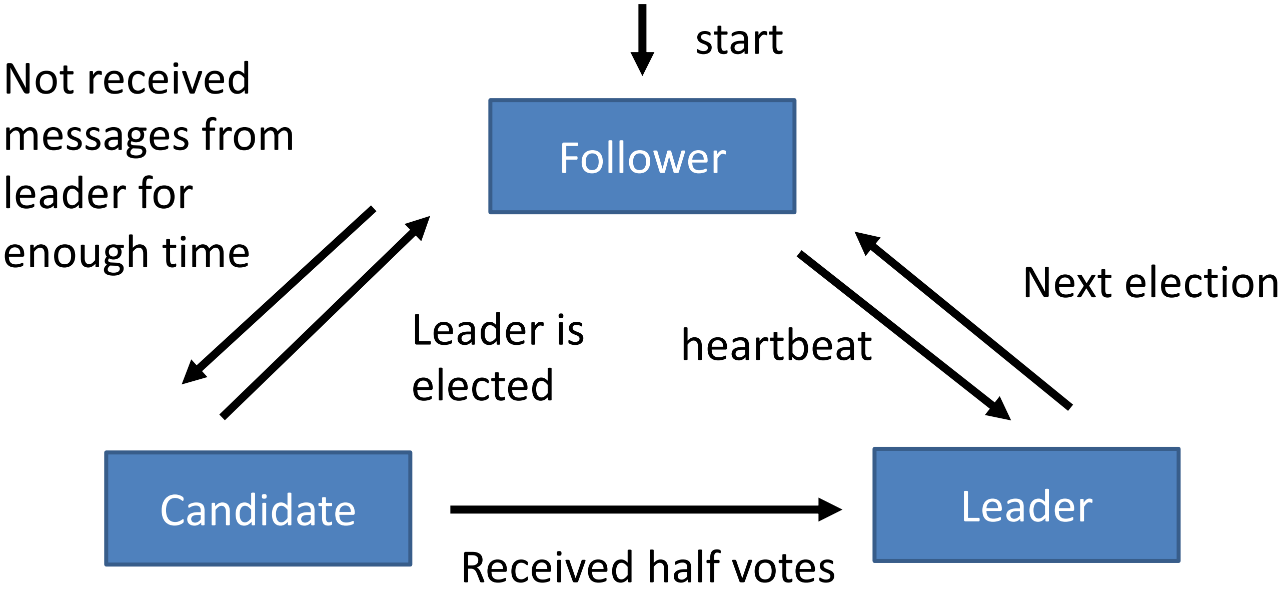
  2. At election, all nodes status change from Follower to Candidate, and send election requests to other nodes



2. Raft algorithm

- How Raft algorithm works:

  3. When nodes receive election requests, the voting starts. Every election, one node can only vote once.

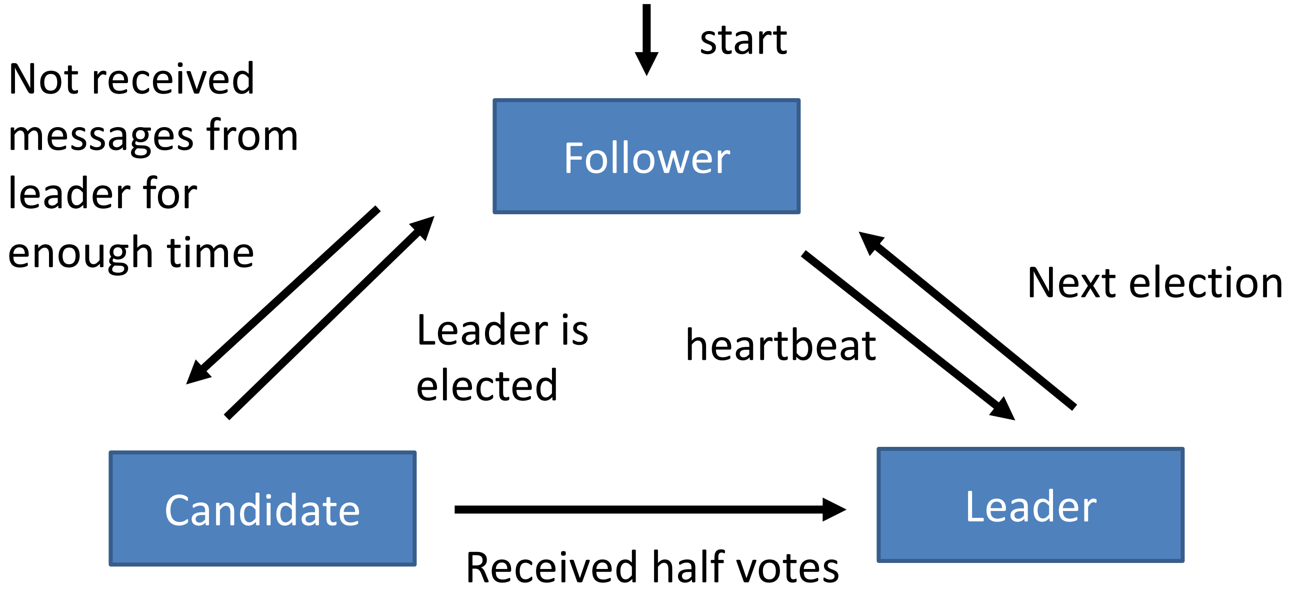


2. Raft algorithm

- How Raft algorithm works:

4. If the node that initiates the election request receives more than half of the votes, it will become the master node, and its status will be changed to Leader, and the status of other nodes will be reduced from Candidate to Follower.

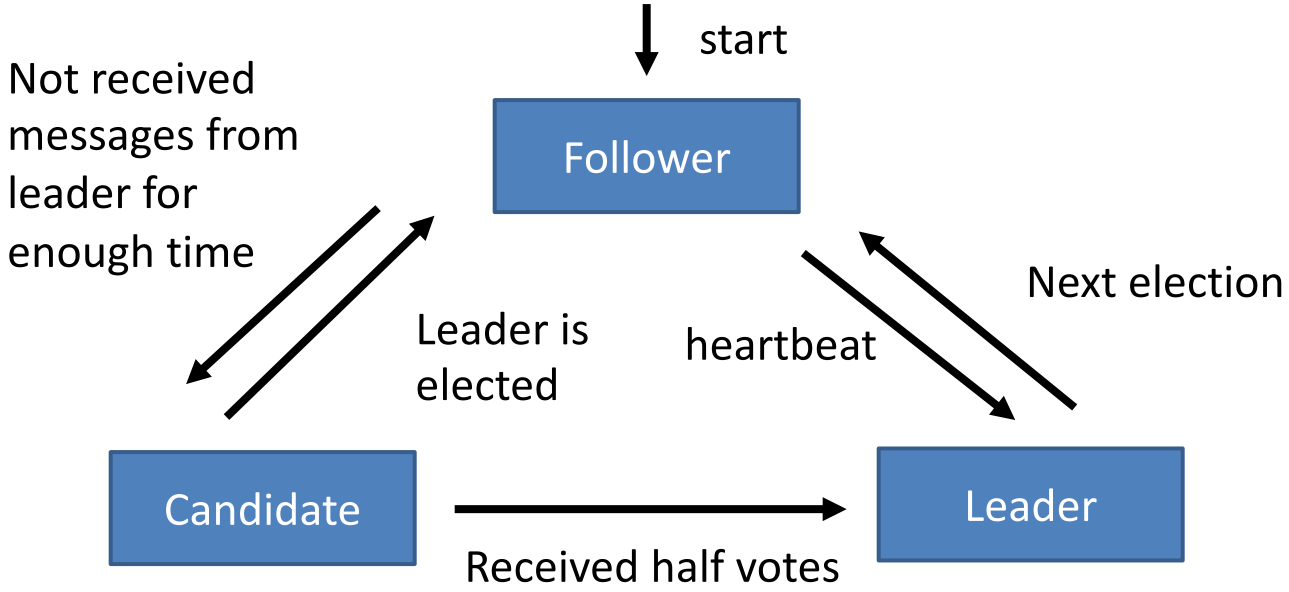
   Leader nodes and follower nodes send heartbeat packets periodically to detect whether the master node is alive.



2. Raft algorithm

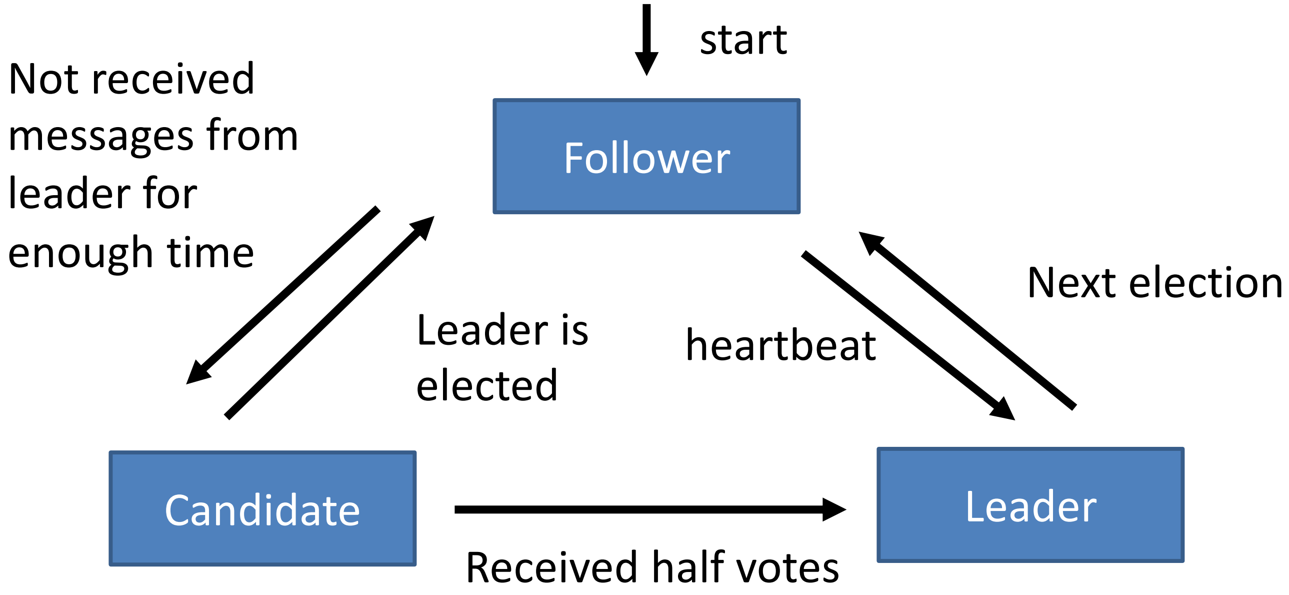
- How Raft algorithm works:

  5. When the term of the Leader node is reached, the status of the Leader node is changed from Leader to Follower, and a new round of election starts.



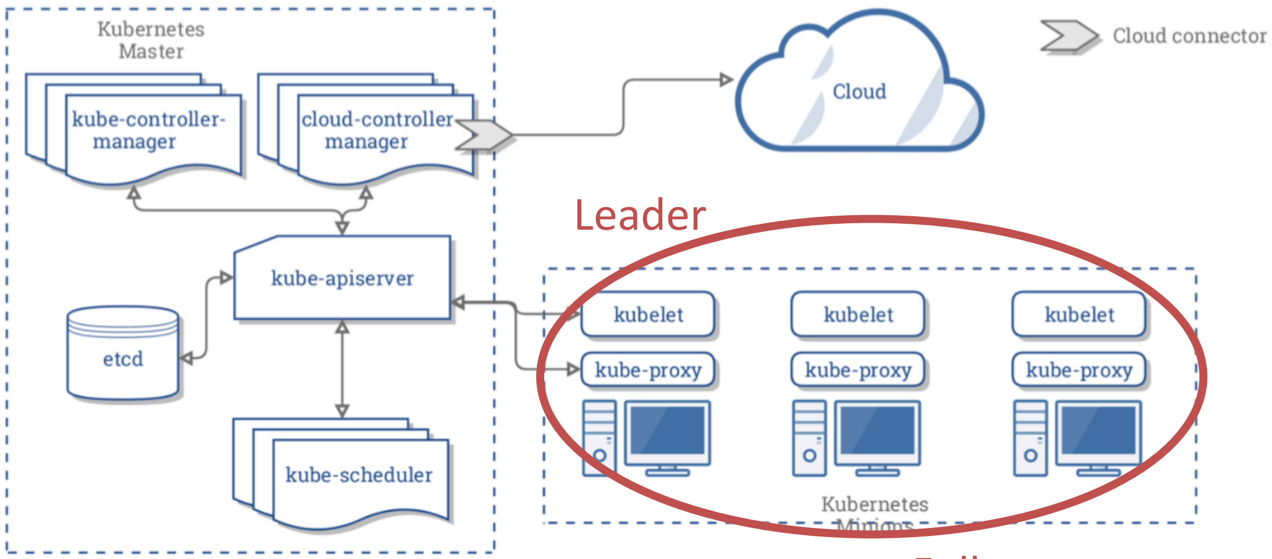
2. Raft algorithm

- When the new election starts in Raft algorithm:
  - When the leader term reaches
  - When the leader node fails or crashes



Raft algorithm example in Kubernetes

- How Kubernetes deals with data failure:
  - To ensure reliability, 3 nodes are usually deployed for data backup. One of the three nodes will be selected as the master, and the other nodes will be used as backups.



Leader

Follower

Kubernetes Architectural Overview (Retrieved from kubernetes.io)

2. Raft algorithm

- Advantages:
  - Fast election speed
  - low algorithm complexity
  - simple and easy to implement (who lives and who has the half votes is the master node)

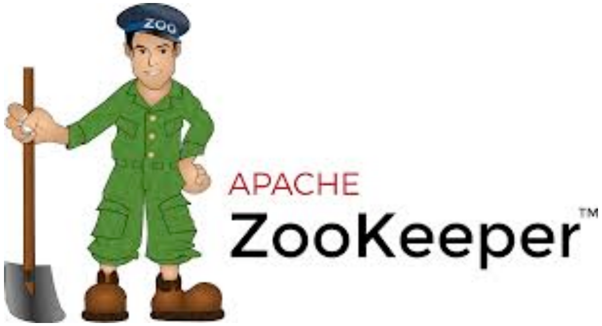
2. Raft algorithm

- Disadvantages:
  - It requires that each node in the system can communicate with each other (vote), and requires the node which has more than half of the votes to be the master, thus the communication traffic is large

3. ZAB algorithm

- ZAB (ZooKeeper Atomic Broadcast) election algorithm is designed for ZooKeeper to implement distributed election

- ZAB algorithm is an improvement on Raft algorithm



3. ZAB algorithm

- Each node in the Zookeeper cluster has three roles:
  - Leader: master node;
  - Follower: follow the leader;
  - Observer: without voting right.

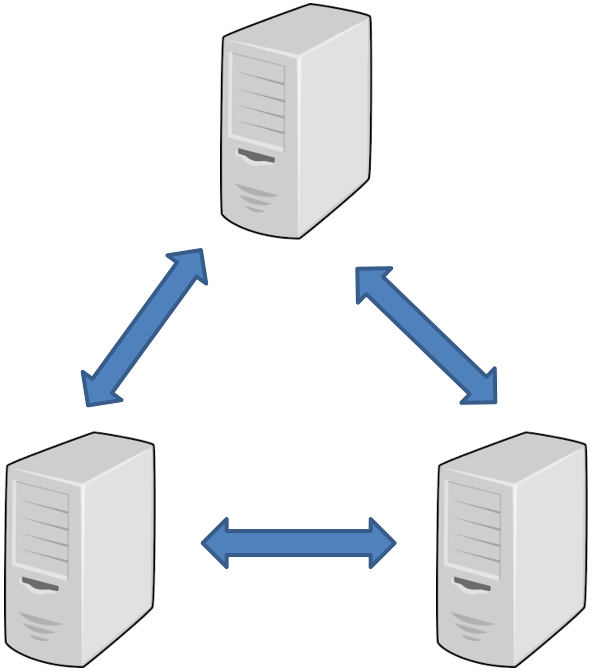
3. ZAB algorithm

- During the election, each node in the Zookeeper cluster has four states:
  1. Looking state or election state. When the node is in this state, it will consider that there is no leader in the current cluster, so it will enter the election state by itself.
  2. The Leading state: indicates that the master has been selected and the current node is the Leader.
  3. Following state: After the master has been selected in the cluster, the status of other non-master nodes is updated to Following.
  4. Observing state: indicates that the current node is the Observer, just waits and has no voting rights.

3. ZAB algorithm

- Each node has a unique triple (server_id, server_zxID, epoch):
  - server_id: represents the unique ID of this node;
  - server_zxID: represents the data ID stored by this node. The larger the data ID, the newer the data and the greater the voting weight;
  - epoch: represents the currently election round

- ZAB: the minority obeys the majority, and the node with the largest ID (both server_id and server_zxID) becomes the master.



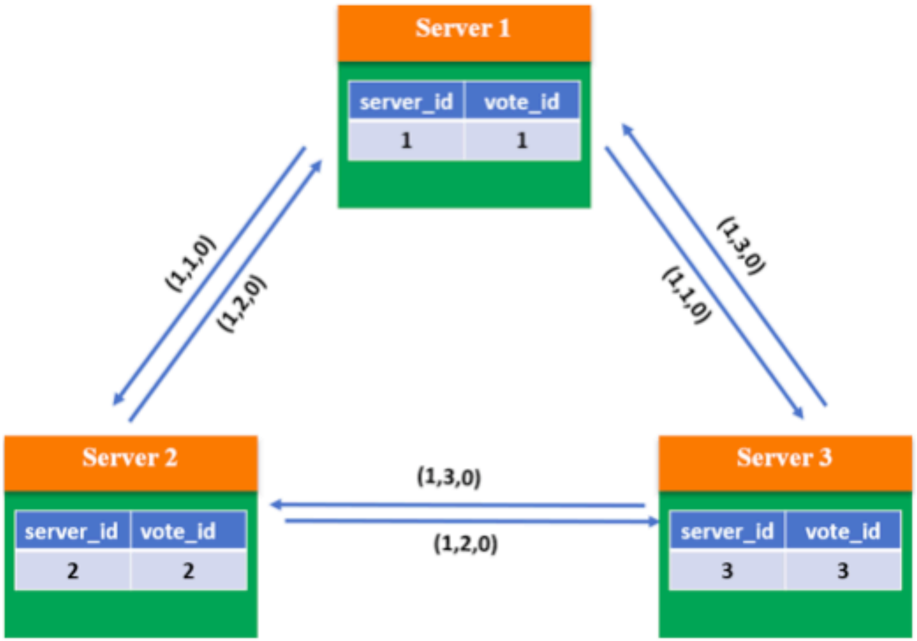
(server_id, server_zxID, epoch)

3. ZAB algorithm

- How ZAB algorithm works:

1. When the system starts, the current voting of the 3 servers is the first round of voting, epoch = 1 and zxID are all 0.

   At this time, each server selects itself and broadcasts the vote information <epoch, vote_id, vote_zxID>.



server_id	vote_id
1	1

server_id	vote_id
2	2

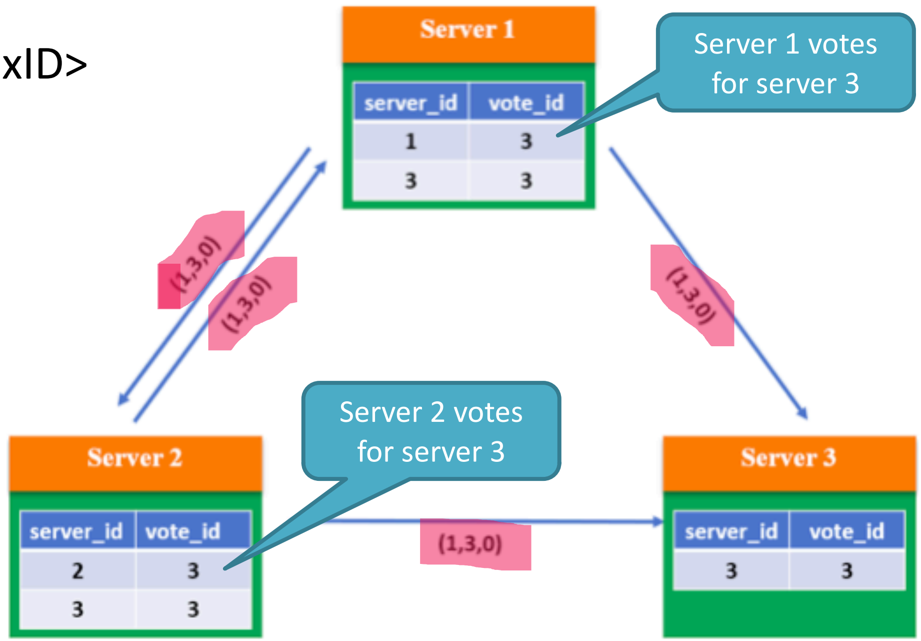
server_id	vote_id
3	3

3. ZAB algorithm

- How ZAB algorithm works:
  2. Since the epoch and zxID of the three servers are the same, the server_id is compared, and the larger one is the selection object. Therefore, Server 1 and Server 2 change the vote_id to 3, update their ballot boxes and rebroadcast their votes.

<epoch, vote_id, vote_zxID>

(1, 1, 0)
(1, 2, 0)
(1, 3, 0)



Server 1

server_id	vote_id
1	3
3	3

Server 2

server_id	vote_id
2	3
3	3

Server 3

server_id	vote_id
3	3

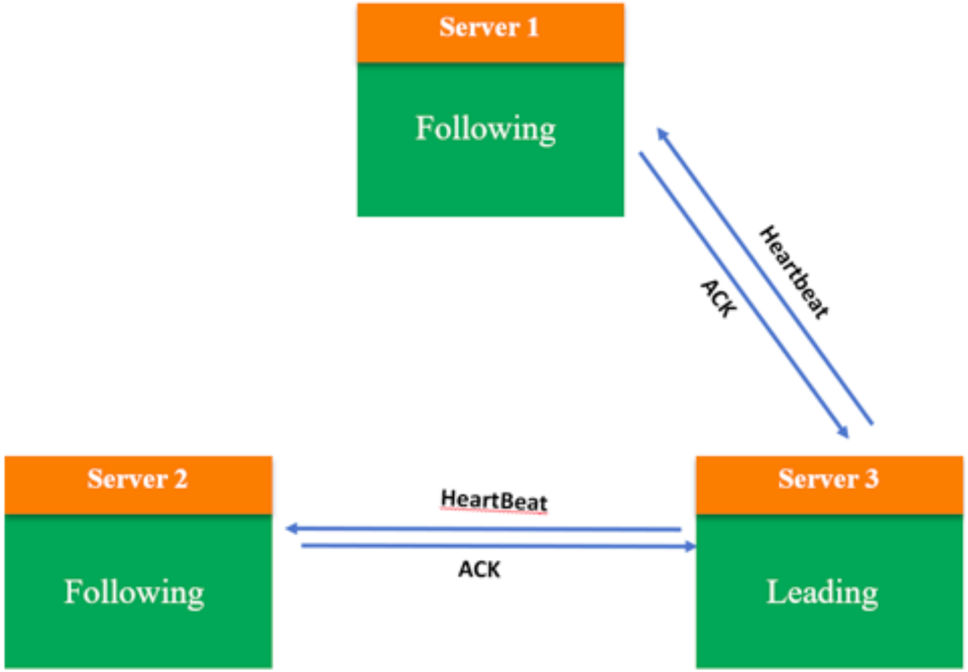
Server 1 votes for server 3

Server 2 votes for server 3

3. ZAB algorithm

- How ZAB algorithm works:

3. All servers in the system have elected Server 3, so Server 3 is elected as the Leader and is in the Leading state; Server 1 and Server 2 are in the Following state. Server 3 sends heartbeat packets to other servers and maintains the connection.



3. ZAB algorithm

- Advantages:
  - High performance
  - No special requirements for the system
  - Better stability, reelection could happen, but leader will not change frequently

3. ZAB algorithm

- Disadvantages:
  - Heavy communication:
    - Broadcast storms could happen: If there are N nodes, and each node broadcasts at the same time, the amount of information in the cluster is N * (N-1) messages.
  - Node ID and data ID is needed, which means that you need to know the Node ID and data ID of all nodes, so the election time is relatively long

Comparison

	Bully	Raft	ZAB
Election message	Alive message	Accept or reject message	Voting message <epoch, vote_id, vote_zxID>
How to elect leader	Largest ID	Get half of the votes	Node contains latest data or largest ID
How the algorithm works	When nodes find no responses from leader or leader fails, raise new election	Every node can be Candidate and selected as Leader. Every follower can only vote once in every election.	Every node at Looking state can join the election and vote many time. The leader depends on the epoch, zxID and server_ID.
Election time	Short	Short	Long
Performance	Bully < Raft < ZAB		

Most of clusters use odd number of nodes

- The idea of the election algorithm is that the minority obeys the majority, and the nodes that get more votes win.

- If a cluster use even number of nodes, the probability that two nodes have the same number of votes will be very high.

- Therefore, most of the selection algorithms usually use odd nodes.