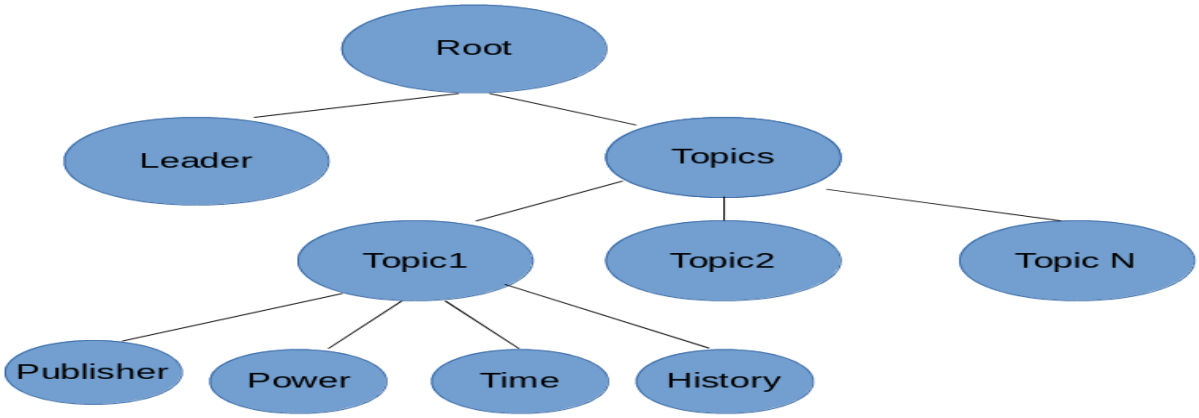
Assignment 3 Report

1 Implementation

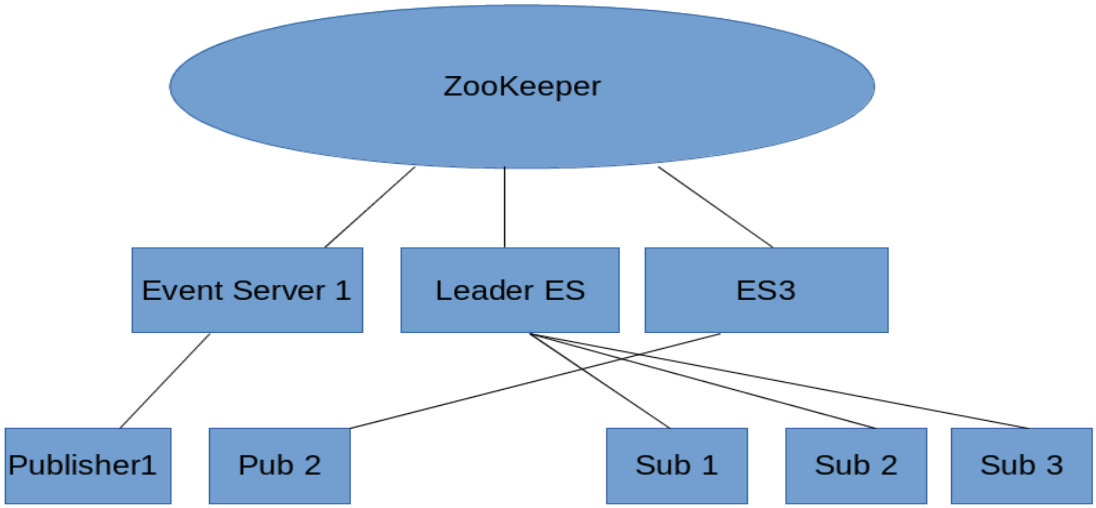
1.1 ZooKeeper Data Structure

The structure of Zoo looks as figure below. We hace a leader node to save the leader IP. For each topic, we have a node named by the topic Id, where the value of this node is the value of this topic. This node has four children: publisher, the power of this publisher, time this latest event being published and the history of this topic.



2. Network Structure

We have several servers, each one is a client connect to the zookeeper, they access, edit the data stored in zookeeper. Publishers can connect to any event service to publish event, based on the power of this publisher on this event, server will decide to update this event or not. Every ES is watching on all the topic node and leader node. Once an event (a node in zookeeper) changed, Leader ES will be notified and publish the new event to all the subscribers.



1.3 Leader Election

When the leader session (ES) stops working, zookeeper will automatically select the new leader. New leader will learn this by checking its own status. New leader will change the value of leader node to its own IP, other ES will be notified since they are watching this node. Therefore they can tell connected subscribers and publishers about the new leader.

2. End-End Measurement

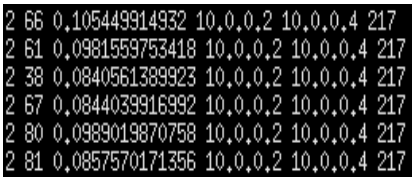


Figure3. The 3rd column is the communication time measure in seconds.

We tested 2 models: first one is 3 ES, 2 Puber, 2 Suber. Second one is 8 Puber and 10 Suber. We compare their End to End average time, listed in the table below.

#Switches	#ES	#Publisher	#Subscriber	Avg. End-End Time (sec)
1	3	2	2	0.10634
1	3	8	10	0.12457

But if we use DHT or single ES (like homework1), the time needed is much smaller (0.01 ~ 0.002). I think it is reasonable since Zookeeper make every server has the same version of data, keep the consistency among all the servers needs time.