Homework 5

Tao Yu

1 Price of several stocks

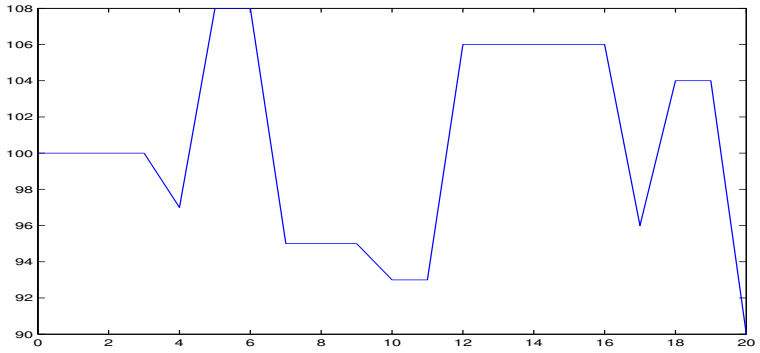

Figure 1: Stock price of Google for 20 orders. x is the number of orders, y is the price

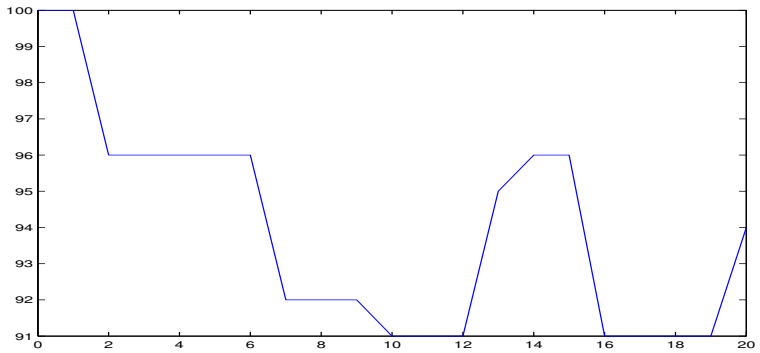

Figure 2: Stock price of Amazon for 20 orders. x is the number of orders, y is the price

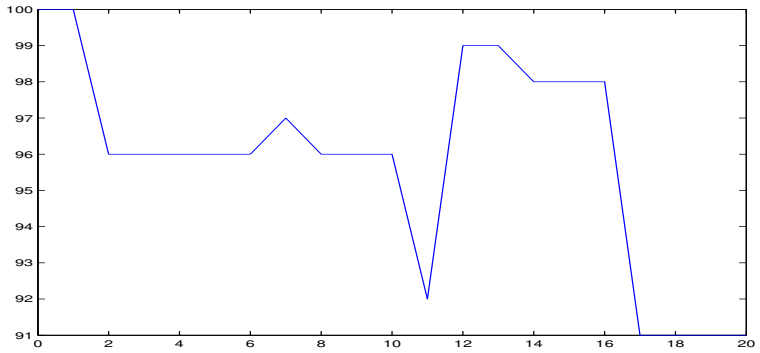

Figure 3: Stock price of Microsoft for 20 orders. x is the number of orders, y is the price

2 Questions

2.1 What are the distributed shared objects in you program?

Ans: A List of Account called clients which contains all the client information and a List of Stock called stockList which contains all the stock information.

2.2 Is your program Byzantine failure safe? Why or why not? If not, what needs to be done

Ans: No. My program won't be able to handle cases of machines that are taken over by virus, or a network that sometimes modifies the the transmitting messages. In order to be able to deal with Byzantine failures, my program needs to follow the Byzantine agreement mechanism which means that the system must reach an agreement to deal with faulty processes. However, this requires more transmission between the servers, which may make the system slower.

2.3 Is virtual synchrony in JGroups scalable? Why or why not?

Ans: Yes. Because when new member enters or leaves, virtual synchrony in JGroups ensures that messages are all transmitted to the members of the current group before a new view is installed. If a new member wants to join the group, all the transmitting message are transmitted before the new member joins the group. When the new member joins the group, it get the complete state of the group. When a member in the group wants to leave the group, all the transmitting message are transmitted before the member leaves the group so that if the transmitting messages are from the member who wants to leave, all the other members receive the messages. Therefore, the virtual synchrony in JGroups is scalable.

2.4 In order to achieve consistency between the processes, did you need additional distributed locking protocols? Why or why not?

Ans: Yes. Two-phase commit or the advanced version three-phase commit is needed. Because in my program, each process receives the message and deal with the message individually. So there is no guarantee that when one process commits the order, other processes commits the order as well. Two-phase commit or three-phase commit solves this problem by introducing Request, Vote, Global_Abort and Global_Commit message to achieve the consistency between processes.