University of Melbourne

Distributed Systems

COMP90015

Assignment 2: Group JWP

Authors:
William Campbell (wcampbell)
Justin Gunn (gunnjk)
Peng Zhao (pzhao2)

Instructor:
Prof Aaron Harwood

Student number:	*Emails*
756142	w.campbell@student.unimelb.edu.au
989347	gun-njk@student.unimelb.edu.au
899632	pzhao2@student.unimelb.edu.au

February 26, 2019

1 Introduction

The project improved upon our first iteration of a distributed group communication system. The new system is designed to provide availability and eventual consistency in the presence of server failures and network partitioning. The main protocols implemented were

- Replication of Message Queue and a global clock from the server side to achieve eventual delivery of messages in the order they were sent.
- Failure protocol to reconnect partitioned servers and continue sending replicated Message Queues.
- Recongifure login, so that users can login from anywhere

We achieved a functioning scalable and dynamic distributed application. Providing high availability and eventual consistency in its message passing service. The messages have a guarantee to be sent to all members connect to the network at that time. Additionally messages sent from a client will be delivered to receivers in the order they were sent.

2 failure model

Our system achieved high availability and eventual consistency by implementing a "self healing" tree structure in our servers. Each server uses a message queue to retrieve messages from its clients. When a client sends a message it is put in this queue. The server then processes jobs FIFO from this queue to pass them to all other servers. This queue is replicated and put in that nodes parent node, then updated each time the message queue is. Updates occur before messages are processed. If a server fails, connections to that server are re-routed to where the backup message queue is held. Then this queues messages are again replicated and pushed to that nodes parent and then processed until the queue is empty. This way all messages are eventually sent to all clients.

Whenever a server fails or a connection to a server fails (even if transiently) servers connected to that server are rerouted to that servers parent node. In the special case that the route node is killed, it puts it replicated message queue in the first server to connect to it. We implemented this logic by giving each server a dead-pool list. This stores the connection information of their parent nodes replication node. When the socket to a parent has failed, they access their dead-pool and make a run-time connection to the new server to initiate replication message queue passing. This arrangement is illustrated in figure 2 and figure 3 bellow, demonstrating initial dead-pool lists, reconnect and re-establishing dead-pool lists.

Figure 1: Initiated server tree

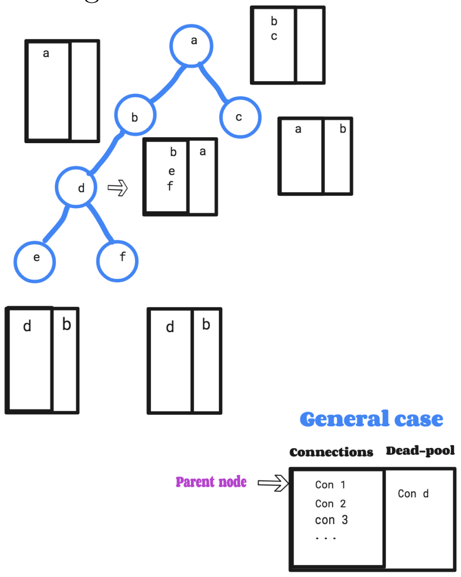
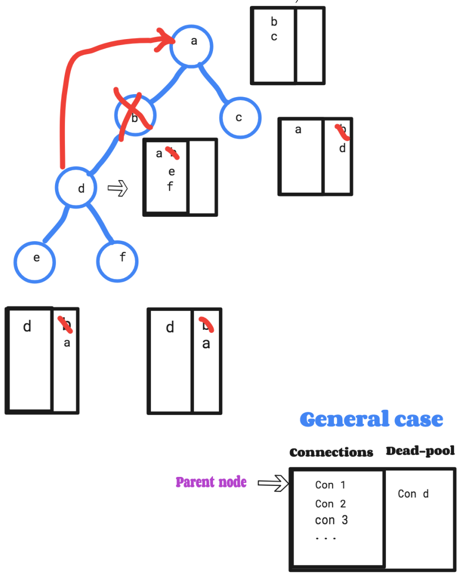


Figure 2: Server b failure, reconnect of d to a



3 Protocols and messages

AUTHENTICATION_SUCCESS

When a server receives an authentication message from another. They send back an AUTHENTICATION_SUCCESS message, containing its parent nodes information, for the servers dead-pool. If they have no parent node, the message is left blank.
Example:

"secret" : "aaa",
"port" : "3789",
"hostname" : "127.000.101"
"empty": "F"

The server will:

- Set their deadpool with this address, stored in their Settings class under backSecret, backPort and backHost
- if this server dies,stored as Settings. parentServer. They initiate a connection to their deadpool using the same protocols as the intial AUTHENTICATE, updating their deadpool in the process.

Revised client login, login from anywhere

LOGIN_REQUEST
Broadcast from a server to all other servers (between servers only) to indicate that a client is trying to login as a username with a given secret. Example:

```
"command" : "LOGIN_REQUEST",
"username" : "aaron",
"secret" : "fmnmpp3ai91qb3gc2bvs14g3ue"
```

A username must be present. The value of username may be anonymous in which case no secret field should be given (it should be ignored). Otherwise a secret must be given.
A server that receives this message will:

- Broadcast a LOGIN_DENIED to all other servers (between servers only) if the username is not known to the server and it is a terminal server within the server tree (i.e. the last server on a branch).

- Broadcast a LOGIN_ALLOWED to all other servers (between servers only) if the username and secret is known to the server. The server will record this username and secret pair in its local storage.
- INVALID_MESSAGE in any case where the message is incorrect
- Broadcast another LOGIN_REQUEST to remaining network branches and await their responses if the server is not a terminal server.

LOGIN_DENIED
Broadcast from a server to all other servers (between servers only), if the server received a LOGIN_REQUEST and to indicate that the username and password is not known, the original client should not be logged in. Example:

"command" : "LOGIN_DENIEDT",
"username" : "aaron",
"secret" : "fmnmpp3ai91qb3gc2bvs14g3ue"

LOGIN_ALLOWED
Broadcast from a server to all other servers (between servers only), if the server received a LOGIN_REQUEST and to indicate that the username and password already known, the original client should be logged in. Example: Example:

"command" : "LOGIN_ALLOWEDT",
"username" : "aaron",
"secret" : "fmnmpp3ai91qb3gc2bvs14g3ue"

LOGIN_DENIED
Broadcast from a server to all other servers (between servers only), if the server received a LOGIN_REQUEST and to indicate that the username and password is not known, the original client should not be logged in. Example:

"command" : "LOGIN_DENIEDT",
"username" : "aaron",
"secret" : "fmnmpp3ai91qb3gc2bvs14g3ue"

Clock server, time stamping for order and delivery guarantee ____________________________________
LOGIN_ALLOWED
Broadcast from a server to all other servers (between servers only), if the server received a LOGIN_REQUEST and to indicate that the username and password already known, the original client should be logged in. Example: Example:

"command" : "LOGIN_ALLOWEDT",
"username" : "aaron",
"secret" : "fmnmpp3ai91qb3gc2bvs14g3ue"

ImprovementsTo improve upon this model, we would need to implement the same message queue replication for received messages as well. However this proved above our programming skills to implement.

4 assumptions

The above implementation relies on assuming 2 failures do not happen to both a server and where its backup is at the same time. If this was not true we would need to hold additional replicas of each, however we assumed 2 failures would not happen at once for our system. Additionally our clock server is designed to never fail, this would need to be implemented using some form of RAID replication, guaranteeing that the clock is always on. This was outside our scope of abilities.

5 Functioning with failure demonstration

The "self healing tree" that we have implemented functions as described in our failure model section continues to function in the face of failure by maintaining a list of its parents parent. We have called this our dead pool (backPort and backServer in stored in settings). For every node bellow the root nodes children. This is a special case and only the 2nd child maintains a list of the 1st child's address. This allows the tree to "heal" if the root node dies.See diagrams in failure model section for further clarification. Where if one of the other nodes dies, the other child will still be connected to the root and our tree is maintained.

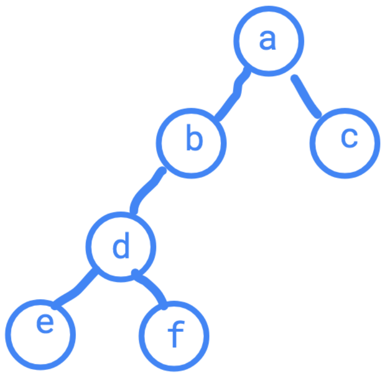
Bellow these children, each new node is passed the address of the parent of the node they connected to. Upon a nodes parents failure, they initiate a new socket with the node stored in their dead-pool. Those reconnecting a partitioned network and providing continued functionality for the system. Additionally we have implemented replicated Ques with every message a node will process. Once a message from a client comes it is put in this queue, the queue is replicated (if this is the first message) or sent to a replicated queue at this nodes parent. Meaning if one of the nodes fails mid message passing. Those messages have been copied and reside in their parent node. If they die the network will heal and reconnect, once this happens that queue will finish sending out any pending messages. This guarantees all messages sent by a client to a server will eventually reach their recipients, even in the face of server failure.

System test: Bellow are trees describing the connections that maintain system availability.

- Six servers are started up in a tree format as dictated in the bellow image. Where port numbers of each node are as follows.

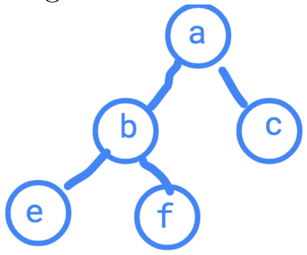
  - a = 3781
  - b = 3782
  - c = 3783
  - e = 3784
  - f = 3785
  - g = 3786

Figure 3: Six server tree.



- Server b fails, leading to d reconnecting to server a and the tree bellow.

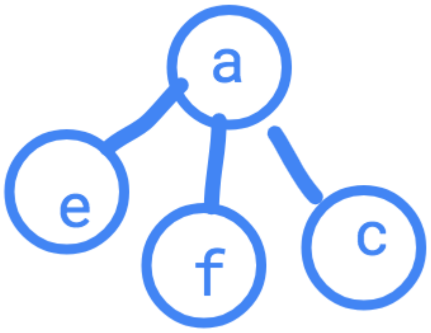
Figure 4: Six server tree.



- Server d fails lead to e and f to reconnect to a and the tree bellow.

Special cases Unfortunately the system implemented still contains some bugs and corner cases where the dead-pool list is not updated properly. Still strictly occurs when the pool of servers has grown over three. Then through failures the $2^(nd)$ server added, then the first server added are eliminated in that

Figure 5: Six server tree.



order.In this case the remaining servers will fail to have an updated dead-pool list and when the root server dies they will try to connect to that $2^(nd)$ server, however it is dead. Saying this the logic could be implemented, however we were not able to in the time span.
The above trees were established under the bellow sequence of steps running our program.

1. start a clock server to guarantee client message deliver running ClockServer and -lp 3779
2. Create the tree seen in figure 3, by starting servers using the following sequence of server start ups.
    (a) -lp 3781 -lh localhost -ms aaa
    (b) -lp 3782 -lh localhost -rh localhost -rp 3781 -s aaa -ms bbb
    (c) -lp 3783 -lh localhost -rh localhost -rp 3782 -s aaa -ms ccc d
    (d) -lp 3784 -lh localhost -rh localhost -rp 3783 -s bbb -ms ddd e

(e) -lp 3785 -lh localhost -rh localhost -rp 3784 -s ddd -ms eee f

(f) -lp 3786 -lh localhost -rh localhost -rp 3785 -s ddd -ms fff

3. Kill server d

4. Kill server b

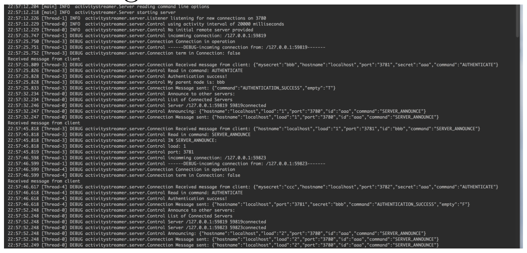


Figure 6: Server a inital.

```
22:57:12.204 [main] INFO  activitystreamer.Server reading command line options
22:57:12.218 [main] INFO  activitystreamer.Server starting server
22:57:12.226 [Thread-1] INFO  activitystreamer.server.Listener listening for new connections on 3780
22:57:12.229 [Thread-0] INFO  activitystreamer.server.Control using activity interval of 20000 milliseconds
22:57:12.229 [Thread-0] INFO  activitystreamer.server.Control No initial remote server provided
22:57:25.747 [Thread-1] DEBUG activitystreamer.server.Control incomming connection: /127.0.0.1:59819
22:57:25.750 [Thread-3] DEBUG activitystreamer.server.Connection Connection in operation
22:57:25.751 [Thread-1] DEBUG activitystreamer.server.Control ------DEBUG-incoming connection from: /127.0.0.1:59819-------
22:57:25.752 [Thread-3] DEBUG activitystreamer.server.Connection term in Connection: false
Received message from client
22:57:25.809 [Thread-3] DEBUG activitystreamer.server.Connection Received message from client: {"mysecret":"bbb","hostname":"localhost","port":"3781","secret":"aaa","command":"AUTHENTICATE"}
22:57:25.826 [Thread-3] DEBUG activitystreamer.server.Control Read in command: AUTHENTICATE
22:57:25.828 [Thread-3] DEBUG activitystreamer.server.Control Authentication success!
22:57:25.828 [Thread-3] DEBUG activitystreamer.server.Control My parent node is: bbb
22:57:25.833 [Thread-3] DEBUG activitystreamer.server.Connection Message sent: {"command":"AUTHENTICATION_SUCCESS","empty":"T"}
22:57:32.234 [Thread-0] DEBUG activitystreamer.server.Control Announce to other servers:
22:57:32.234 [Thread-0] DEBUG activitystreamer.server.Control List of Connected Servers
22:57:32.246 [Thread-0] DEBUG activitystreamer.server.Control Server /127.0.0.1:59819 59819connected
22:57:32.247 [Thread-0] DEBUG activitystreamer.server.Control Announcing: {"hostname":"localhost","load":"1","port":"3780","id":"aaa","command":"SERVER_ANNOUNCE"}
22:57:32.247 [Thread-0] DEBUG activitystreamer.server.Connection Message sent: {"hostname":"localhost","load":"1","port":"3780","id":"aaa","command":"SERVER_ANNOUNCE"}
Received message from client
22:57:45.818 [Thread-3] DEBUG activitystreamer.server.Connection Received message from client: {"hostname":"localhost","load":"1","port":"3781","id":"bbb","command":"SERVER_ANNOUNCE"}
22:57:45.818 [Thread-3] DEBUG activitystreamer.server.Control Read in command: SERVER_ANNOUNCE
22:57:45.818 [Thread-3] DEBUG activitystreamer.server.Control IN SERVER_ANNOUNCE:
22:57:45.818 [Thread-3] DEBUG activitystreamer.server.Control load: 1
22:57:45.819 [Thread-3] DEBUG activitystreamer.server.Control port: 3781
22:57:46.598 [Thread-1] DEBUG activitystreamer.server.Control incomming connection: /127.0.0.1:59823
22:57:46.599 [Thread-1] DEBUG activitystreamer.server.Control ------DEBUG-incoming connection from: /127.0.0.1:59823-------
22:57:46.599 [Thread-4] DEBUG activitystreamer.server.Connection Connection in operation
22:57:46.599 [Thread-4] DEBUG activitystreamer.server.Connection term in Connection: false
Received message from client
22:57:46.617 [Thread-4] DEBUG activitystreamer.server.Connection Received message from client: {"mysecret":"ccc","hostname":"localhost","port":"3782","secret":"aaa","command":"AUTHENTICATE"}
22:57:46.618 [Thread-4] DEBUG activitystreamer.server.Control Read in command: AUTHENTICATE
22:57:46.618 [Thread-4] DEBUG activitystreamer.server.Control Authentication success!
22:57:46.618 [Thread-4] DEBUG activitystreamer.server.Connection Message sent: {"hostname":"localhost","port":"3781","secret":"bbb","command":"AUTHENTICATION_SUCCESS","empty":"F"}
22:57:52.248 [Thread-0] DEBUG activitystreamer.server.Control Announce to other servers:
22:57:52.248 [Thread-0] DEBUG activitystreamer.server.Control List of Connected Servers
22:57:52.248 [Thread-0] DEBUG activitystreamer.server.Control Server /127.0.0.1:59819 59819connected
22:57:52.248 [Thread-0] DEBUG activitystreamer.server.Control Server /127.0.0.1:59823 59823connected
22:57:52.248 [Thread-0] DEBUG activitystreamer.server.Control Announcing: {"hostname":"localhost","load":"2","port":"3780","id":"aaa","command":"SERVER_ANNOUNCE"}
22:57:52.248 [Thread-0] DEBUG activitystreamer.server.Connection Message sent: {"hostname":"localhost","load":"2","port":"3780","id":"aaa","command":"SERVER_ANNOUNCE"}
22:57:52.249 [Thread-0] DEBUG activitystreamer.server.Connection Message sent: {"hostname":"localhost","load":"2","port":"3780","id":"aaa","command":"SERVER_ANNOUNCE"}
```

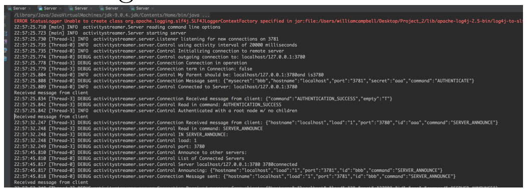


Figure 7: Server b initial.

```
/Library/Java/JavaVirtualMachines/jdk-9.0.4.jdk/Contents/Home/bin/java ...
ERROR StatusLogger Unable to create class org.apache.logging.slf4j.SLF4JLoggerContextFactory specified in jar:file:/Users/williamcampbell/Desktop/Project_2/lib/apache-log4j-2.5-bin/log4j-to-sl
22:57:25.710 [main] INFO  activitystreamer.Server reading command line options
22:57:25.723 [main] INFO  activitystreamer.Server starting server
22:57:25.730 [Thread-1] INFO  activitystreamer.server.Listener listening for new connections on 3781
22:57:25.735 [Thread-0] INFO  activitystreamer.server.Control using activity interval of 20000 milliseconds
22:57:25.735 [Thread-0] INFO  activitystreamer.server.Control Initializing connection to remote server
22:57:25.774 [Thread-0] DEBUG activitystreamer.server.Control outgoing connection to: localhost/127.0.0.1:3780
22:57:25.778 [Thread-3] DEBUG activitystreamer.server.Connection Connection in operation
22:57:25.779 [Thread-3] DEBUG activitystreamer.server.Connection term in Connection: false
22:57:25.804 [Thread-0] INFO  activitystreamer.server.Control My Parent should be: localhost/127.0.0.1:3780and is3780
22:57:25.808 [Thread-0] DEBUG activitystreamer.server.Connection Message sent: {"mysecret":"bbb","hostname":"localhost","port":"3781","secret":"aaa","command":"AUTHENTICATE"}
22:57:25.809 [Thread-0] INFO  activitystreamer.server.Control Connected to Server: localhost/127.0.0.1:3780
Received message from client
22:57:25.834 [Thread-3] DEBUG activitystreamer.server.Connection Received message from client: {"command":"AUTHENTICATION_SUCCESS","empty":"T"}
22:57:25.842 [Thread-3] DEBUG activitystreamer.server.Control Read in command: AUTHENTICATION_SUCCESS
22:57:25.842 [Thread-3] INFO  activitystreamer.server.Control Authenticated with a root node w/ no children
Received message from client
22:57:32.247 [Thread-3] DEBUG activitystreamer.server.Connection Received message from client: {"hostname":"localhost","load":"1","port":"3780","id":"aaa","command":"SERVER_ANNOUNCE"}
22:57:32.248 [Thread-3] DEBUG activitystreamer.server.Control Read in command: SERVER_ANNOUNCE
22:57:32.248 [Thread-3] DEBUG activitystreamer.server.Control IN SERVER_ANNOUNCE:
22:57:32.248 [Thread-3] DEBUG activitystreamer.server.Control load: 1
22:57:32.249 [Thread-3] DEBUG activitystreamer.server.Control port: 3780
22:57:45.810 [Thread-0] DEBUG activitystreamer.server.Control Announce to other servers:
22:57:45.810 [Thread-0] DEBUG activitystreamer.server.Control List of Connected Servers
22:57:45.817 [Thread-0] DEBUG activitystreamer.server.Control Server localhost/127.0.0.1:3780 3780connected
22:57:45.817 [Thread-0] DEBUG activitystreamer.server.Control Announcing: {"hostname":"localhost","load":"1","port":"3781","id":"bbb","command":"SERVER_ANNOUNCE"}
22:57:45.818 [Thread-0] DEBUG activitystreamer.server.Connection Message sent: {"hostname":"localhost","load":"1","port":"3781","id":"bbb","command":"SERVER_ANNOUNCE"}
Received message from client
```

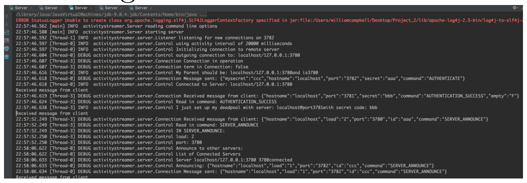


Figure 8: Server c initial.

```
/Library/Java/JavaVirtualMachines/jdk-9.0.4.jdk/Contents/Home/bin/java ...
ERROR StatusLogger Unable to create class org.apache.logging.slf4j.SLF4JLoggerContextFactory specified in jar:file:/Users/williamcampbell/Desktop/Project_2/lib/apache-log4j-2.5-bin/log4j-to-slf4j-
22:57:46.562 [main] INFO  activitystreamer.Server reading command line options
22:57:46.580 [main] INFO  activitystreamer.Server starting server
22:57:46.592 [Thread-1] INFO  activitystreamer.server.Listener listening for new connections on 3782
22:57:46.597 [Thread-0] INFO  activitystreamer.server.Control using activity interval of 20000 milliseconds
22:57:46.597 [Thread-0] INFO  activitystreamer.server.Control Initializing connection to remote server
22:57:46.604 [Thread-0] DEBUG activitystreamer.server.Control outgoing connection to: localhost/127.0.0.1:3780
22:57:46.607 [Thread-3] DEBUG activitystreamer.server.Connection Connection in operation
22:57:46.607 [Thread-3] DEBUG activitystreamer.server.Connection term in Connection: false
22:57:46.616 [Thread-0] INFO  activitystreamer.server.Control My Parent should be: localhost/127.0.0.1:3780and is3780
22:57:46.618 [Thread-0] DEBUG activitystreamer.server.Connection Message sent: {"mysecret":"ccc","hostname":"localhost","port":"3782","secret":"aaa","command":"AUTHENTICATE"}
22:57:46.618 [Thread-0] INFO  activitystreamer.server.Control Connected to Server: localhost/127.0.0.1:3780
Received message from client
22:57:46.619 [Thread-3] DEBUG activitystreamer.server.Connection Received message from client: {"hostname":"localhost","port":"3781","secret":"bbb","command":"AUTHENTICATION_SUCCESS","empty":"F"}
22:57:46.624 [Thread-3] DEBUG activitystreamer.server.Control Read in command: AUTHENTICATION_SUCCESS
22:57:46.638 [Thread-3] INFO  activitystreamer.server.Control I just set up my deadpool with server: localhost@port3781with secret code: bbb
Received message from client
22:57:52.249 [Thread-3] DEBUG activitystreamer.server.Connection Received message from client: {"hostname":"localhost","load":"2","port":"3780","id":"aaa","command":"SERVER_ANNOUNCE"}
22:57:52.249 [Thread-3] DEBUG activitystreamer.server.Control Read in command: SERVER_ANNOUNCE
22:57:52.249 [Thread-3] DEBUG activitystreamer.server.Control IN SERVER_ANNOUNCE:
22:57:52.250 [Thread-3] DEBUG activitystreamer.server.Control load: 2
22:57:52.250 [Thread-3] DEBUG activitystreamer.server.Control port: 3780
22:58:06.622 [Thread-0] DEBUG activitystreamer.server.Control Announce to other servers:
22:58:06.622 [Thread-0] DEBUG activitystreamer.server.Control List of Connected Servers
22:58:06.633 [Thread-0] DEBUG activitystreamer.server.Control Server localhost/127.0.0.1:3780 3780connected
22:58:06.633 [Thread-0] DEBUG activitystreamer.server.Control Announcing: {"hostname":"localhost","load":"1","port":"3782","id":"ccc","command":"SERVER_ANNOUNCE"}
22:58:06.634 [Thread-0] DEBUG activitystreamer.server.Connection Message sent: {"hostname":"localhost","load":"1","port":"3782","id":"ccc","command":"SERVER_ANNOUNCE"}
Received message from client
```

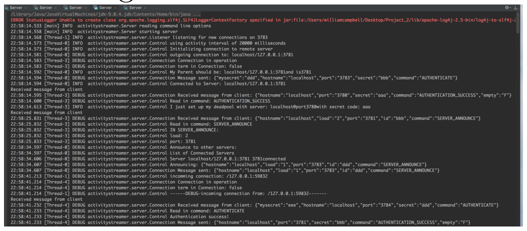


Figure 9: Server d initial.

```
/Library/Java/JavaVirtualMachines/jdk-9.0.4.jdk/Contents/Home/bin/java ...
ERROR StatusLogger Unable to create class org.apache.logging.slf4j.SLF4JLoggerContextFactory specified in jar:file:/Users/williamcampbell/Desktop/Project_2/lib/apache-log4j-2.5-bin/log4j-to-slf4j-2
22:58:14.533 [main] INFO  activitystreamer.Server reading command line options
22:58:14.558 [main] INFO  activitystreamer.Server starting server
22:58:14.568 [Thread-1] INFO  activitystreamer.server.Listener listening for new connections on 3783
22:58:14.573 [Thread-0] INFO  activitystreamer.server.Control using activity interval of 20000 milliseconds
22:58:14.573 [Thread-0] INFO  activitystreamer.server.Control Initializing connection to remote server
22:58:14.581 [Thread-0] DEBUG activitystreamer.server.Control outgoing connection to: localhost/127.0.0.1:3781
22:58:14.583 [Thread-3] DEBUG activitystreamer.server.Connection Connection in operation
22:58:14.583 [Thread-3] DEBUG activitystreamer.server.Connection term in Connection: false
22:58:14.592 [Thread-0] INFO  activitystreamer.server.Control My Parent should be: localhost/127.0.0.1:3781and is3781
22:58:14.594 [Thread-0] DEBUG activitystreamer.server.Connection Message sent: {"mysecret":"ddd","hostname":"localhost","port":"3783","secret":"bbb","command":"AUTHENTICATE"}
22:58:14.594 [Thread-0] INFO  activitystreamer.server.Control Connected to Server: localhost/127.0.0.1:3781
Received message from client
22:58:14.595 [Thread-3] DEBUG activitystreamer.server.Connection Received message from client: {"hostname":"localhost","port":"3780","secret":"aaa","command":"AUTHENTICATION_SUCCESS","empty":"F"}
22:58:14.600 [Thread-3] DEBUG activitystreamer.server.Control Read in command: AUTHENTICATION_SUCCESS
22:58:14.613 [Thread-3] INFO  activitystreamer.server.Control I just set up my deadpool with server: localhost@port3780with secret code: aaa
Received message from client
22:58:25.831 [Thread-3] DEBUG activitystreamer.server.Connection Received message from client: {"hostname":"localhost","load":"2","port":"3781","id":"bbb","command":"SERVER_ANNOUNCE"}
22:58:25.832 [Thread-3] DEBUG activitystreamer.server.Control Read in command: SERVER_ANNOUNCE
22:58:25.832 [Thread-3] DEBUG activitystreamer.server.Control IN SERVER_ANNOUNCE:
22:58:25.832 [Thread-3] DEBUG activitystreamer.server.Control load: 2
22:58:25.833 [Thread-3] DEBUG activitystreamer.server.Control port: 3781
22:58:34.597 [Thread-0] DEBUG activitystreamer.server.Control Announce to other servers:
22:58:34.597 [Thread-0] DEBUG activitystreamer.server.Control List of Connected Servers
22:58:34.606 [Thread-0] DEBUG activitystreamer.server.Control Server localhost/127.0.0.1:3781 3781connected
22:58:34.607 [Thread-0] DEBUG activitystreamer.server.Control Announcing: {"hostname":"localhost","load":"1","port":"3783","id":"ddd","command":"SERVER_ANNOUNCE"}
22:58:34.607 [Thread-0] DEBUG activitystreamer.server.Connection Message sent: {"hostname":"localhost","load":"1","port":"3783","id":"ddd","command":"SERVER_ANNOUNCE"}
22:58:41.213 [Thread-1] DEBUG activitystreamer.server.Control incomming connection: /127.0.0.1:59832
22:58:41.214 [Thread-4] DEBUG activitystreamer.server.Connection Connection in operation
22:58:41.214 [Thread-4] DEBUG activitystreamer.server.Connection term in Connection: false
22:58:41.214 [Thread-1] DEBUG activitystreamer.server.Control ------DEBUG-incoming connection from: /127.0.0.1:59832-------
Received message from client
22:58:41.232 [Thread-4] DEBUG activitystreamer.server.Connection Received message from client: {"mysecret":"eee","hostname":"localhost","port":"3784","secret":"ddd","command":"AUTHENTICATE"}
22:58:41.233 [Thread-4] DEBUG activitystreamer.server.Control Read in command: AUTHENTICATE
22:58:41.233 [Thread-4] DEBUG activitystreamer.server.Control Authentication success!
22:58:41.233 [Thread-4] DEBUG activitystreamer.server.Connection Message sent: {"hostname":"localhost","port":"3781","secret":"bbb","command":"AUTHENTICATION_SUCCESS","empty":"F"}
```

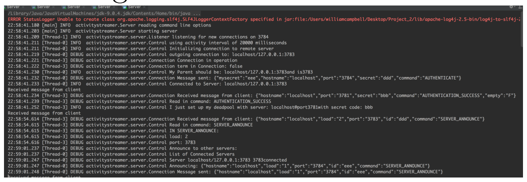


Figure 10: Server e initial.

```
/Library/Java/JavaVirtualMachines/jdk-9.0.4.jdk/Contents/Home/bin/java ...
ERROR StatusLogger Unable to create class org.apache.logging.slf4j.SLF4JLoggerContextFactory specified in jar:file:/Users/williamcampbell/Desktop/Project_2/lib/apache-log4j-2.5-bin/log4j-to-slf4j-2
22:58:41.180 [main] INFO  activitystreamer.Server reading command line options
22:58:41.203 [main] INFO  activitystreamer.Server starting server
22:58:41.209 [Thread-1] INFO  activitystreamer.server.Listener listening for new connections on 3784
22:58:41.211 [Thread-0] INFO  activitystreamer.server.Control using activity interval of 20000 milliseconds
22:58:41.211 [Thread-0] INFO  activitystreamer.server.Control Initializing connection to remote server
22:58:41.219 [Thread-0] DEBUG activitystreamer.server.Control outgoing connection to: localhost/127.0.0.1:3783
22:58:41.221 [Thread-3] DEBUG activitystreamer.server.Connection Connection in operation
22:58:41.222 [Thread-3] DEBUG activitystreamer.server.Connection term in Connection: false
22:58:41.230 [Thread-0] INFO  activitystreamer.server.Control My Parent should be: localhost/127.0.0.1:3783and is3783
22:58:41.232 [Thread-0] DEBUG activitystreamer.server.Connection Message sent: {"mysecret":"eee","hostname":"localhost","port":"3784","secret":"ddd","command":"AUTHENTICATE"}
22:58:41.233 [Thread-0] INFO  activitystreamer.server.Control Connected to Server: localhost/127.0.0.1:3783
Received message from client
22:58:41.234 [Thread-3] DEBUG activitystreamer.server.Connection Received message from client: {"hostname":"localhost","port":"3781","secret":"bbb","command":"AUTHENTICATION_SUCCESS","empty":"F"}
22:58:41.239 [Thread-3] DEBUG activitystreamer.server.Control Read in command: AUTHENTICATION_SUCCESS
22:58:41.252 [Thread-3] INFO  activitystreamer.server.Control I just set up my deadpool with server: localhost@port3781with secret code: bbb
Received message from client
22:58:54.614 [Thread-3] DEBUG activitystreamer.server.Connection Received message from client: {"hostname":"localhost","load":"2","port":"3783","id":"ddd","command":"SERVER_ANNOUNCE"}
22:58:54.615 [Thread-3] DEBUG activitystreamer.server.Control Read in command: SERVER_ANNOUNCE
22:58:54.615 [Thread-3] DEBUG activitystreamer.server.Control IN SERVER_ANNOUNCE:
22:58:54.615 [Thread-3] DEBUG activitystreamer.server.Control load: 2
22:58:54.616 [Thread-3] DEBUG activitystreamer.server.Control port: 3783
22:59:01.237 [Thread-0] DEBUG activitystreamer.server.Control Announce to other servers:
22:59:01.237 [Thread-0] DEBUG activitystreamer.server.Control List of Connected Servers
22:59:01.247 [Thread-0] DEBUG activitystreamer.server.Control Server localhost/127.0.0.1:3783 3783connected
22:59:01.247 [Thread-0] DEBUG activitystreamer.server.Control Announcing: {"hostname":"localhost","load":"1","port":"3784","id":"eee","command":"SERVER_ANNOUNCE"}
22:59:01.248 [Thread-0] DEBUG activitystreamer.server.Connection Message sent: {"hostname":"localhost","load":"1","port":"3784","id":"eee","command":"SERVER_ANNOUNCE"}
```

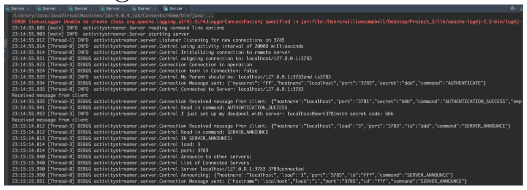


Figure 11: Server f initial.

```
/Library/Java/JavaVirtualMachines/jdk-9.0.4.jdk/Contents/Home/bin/java ...
ERROR StatusLogger Unable to create class org.apache.logging.slf4j.SLF4JLoggerContextFactory specified in jar:file:/Users/williamcampbell/Desktop/Project_2/lib/apache-log4j-2.5-bin/log4j-
23:14:55.885 [main] INFO  activitystreamer.Server reading command line options
23:14:55.903 [main] INFO  activitystreamer.Server starting server
23:14:55.912 [Thread-1] INFO  activitystreamer.server.Listener listening for new connections on 3785
23:14:55.914 [Thread-0] INFO  activitystreamer.server.Control using activity interval of 20000 milliseconds
23:14:55.914 [Thread-0] INFO  activitystreamer.server.Control Initializing connection to remote server
23:14:55.921 [Thread-0] DEBUG activitystreamer.server.Control outgoing connection to: localhost/127.0.0.1:3783
23:14:55.923 [Thread-3] DEBUG activitystreamer.server.Connection Connection in operation
23:14:55.924 [Thread-3] DEBUG activitystreamer.server.Connection term in Connection: false
23:14:55.933 [Thread-0] INFO  activitystreamer.server.Control My Parent should be: localhost/127.0.0.1:3783and is3783
23:14:55.934 [Thread-0] DEBUG activitystreamer.server.Connection Message sent: {"mysecret":"fff","hostname":"localhost","port":"3785","secret":"ddd","command":"AUTHENTICATE"}
23:14:55.935 [Thread-0] INFO  activitystreamer.server.Control Connected to Server: localhost/127.0.0.1:3783
Received message from client
23:14:55.935 [Thread-3] DEBUG activitystreamer.server.Connection Received message from client: {"hostname":"localhost","port":"3781","secret":"bbb","command":"AUTHENTICATION_SUCCESS","emp
23:14:55.941 [Thread-3] DEBUG activitystreamer.server.Control Read in command: AUTHENTICATION_SUCCESS
23:14:55.953 [Thread-3] INFO  activitystreamer.server.Control I just set up my deadpool with server: localhost@port3781with secret code: bbb
Received message from client
23:15:14.812 [Thread-3] DEBUG activitystreamer.server.Connection Received message from client: {"hostname":"localhost","load":"3","port":"3783","id":"ddd","command":"SERVER_ANNOUNCE"}
23:15:14.812 [Thread-3] DEBUG activitystreamer.server.Control Read in command: SERVER_ANNOUNCE
23:15:14.813 [Thread-3] DEBUG activitystreamer.server.Control IN SERVER_ANNOUNCE:
23:15:14.814 [Thread-3] DEBUG activitystreamer.server.Control load: 3
23:15:14.814 [Thread-3] DEBUG activitystreamer.server.Control port: 3783
23:15:15.940 [Thread-0] DEBUG activitystreamer.server.Control Announce to other servers:
23:15:15.940 [Thread-0] DEBUG activitystreamer.server.Control List of Connected Servers
23:15:15.950 [Thread-0] DEBUG activitystreamer.server.Control Server localhost/127.0.0.1:3783 3783connected
23:15:15.950 [Thread-0] DEBUG activitystreamer.server.Control Announcing: {"hostname":"localhost","load":"1","port":"3785","id":"fff","command":"SERVER_ANNOUNCE"}
23:15:15.951 [Thread-0] DEBUG activitystreamer.server.Connection Message sent: {"hostname":"localhost","load":"1","port":"3785","id":"fff","command":"SERVER_ANNOUNCE"}
```

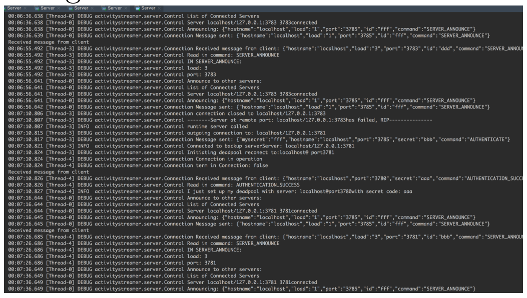
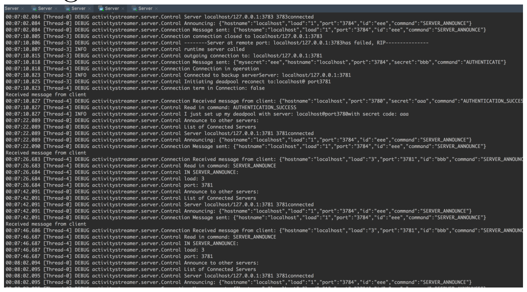


Figure 12: Server f after d death.

```
00:06:36.638 [Thread-0] DEBUG activitystreamer.server.Control List of Connected Servers
00:06:36.638 [Thread-0] DEBUG activitystreamer.server.Control Server localhost/127.0.0.1:3783 3783connected
00:06:36.638 [Thread-0] DEBUG activitystreamer.server.Control Announcing: {"hostname":"localhost","load":"1","port":"3785","id":"fff","command":"SERVER_ANNOUNCE"}
00:06:36.639 [Thread-0] DEBUG activitystreamer.server.Connection Message sent: {"hostname":"localhost","load":"1","port":"3785","id":"fff","command":"SERVER_ANNOUNCE"}
Received message from client
00:06:55.492 [Thread-3] DEBUG activitystreamer.server.Connection Received message from client: {"hostname":"localhost","load":"3","port":"3783","id":"ddd","command":"SERVER_ANNOU
00:06:55.492 [Thread-3] DEBUG activitystreamer.server.Control Read in command: SERVER_ANNOUNCE
00:06:55.492 [Thread-3] DEBUG activitystreamer.server.Control IN SERVER_ANNOUNCE:
00:06:55.492 [Thread-3] DEBUG activitystreamer.server.Control load: 3
00:06:55.492 [Thread-3] DEBUG activitystreamer.server.Control port: 3783
00:06:56.641 [Thread-0] DEBUG activitystreamer.server.Control Announce to other servers:
00:06:56.641 [Thread-0] DEBUG activitystreamer.server.Control List of Connected Servers
00:06:56.641 [Thread-0] DEBUG activitystreamer.server.Control Server localhost/127.0.0.1:3783 3783connected
00:06:56.641 [Thread-0] DEBUG activitystreamer.server.Control Announcing: {"hostname":"localhost","load":"1","port":"3785","id":"fff","command":"SERVER_ANNOUNCE"}
00:06:56.642 [Thread-0] DEBUG activitystreamer.server.Connection Message sent: {"hostname":"localhost","load":"1","port":"3785","id":"fff","command":"SERVER_ANNOUNCE"}
00:07:10.806 [Thread-3] DEBUG activitystreamer.server.Connection connection closed to localhost/127.0.0.1:3783
00:07:10.807 [Thread-3] DEBUG activitystreamer.server.Control --------Server at remote port: localhost/127.0.0.1:3783has failed, RIP---------------
00:07:10.807 [Thread-3] INFO  activitystreamer.server.Control runtime server called
00:07:10.815 [Thread-3] DEBUG activitystreamer.server.Control outgoing connection to: localhost/127.0.0.1:3781
00:07:10.817 [Thread-3] DEBUG activitystreamer.server.Connection Message sent: {"mysecret":"fff","hostname":"localhost","port":"3785","secret":"bbb","command":"AUTHENTICATE"}
00:07:10.821 [Thread-3] INFO  activitystreamer.server.Control Connected to backup serverServer: localhost/127.0.0.1:3781
00:07:10.824 [Thread-3] DEBUG activitystreamer.server.Control Initiating deadpool reconect to:localhost@ port3781
00:07:10.824 [Thread-4] DEBUG activitystreamer.server.Connection Connection in operation
00:07:10.824 [Thread-4] DEBUG activitystreamer.server.Connection term in Connection: false
Received message from client
00:07:10.826 [Thread-4] DEBUG activitystreamer.server.Connection Received message from client: {"hostname":"localhost","port":"3780","secret":"aaa","command":"AUTHENTICATION_SUCC
00:07:10.826 [Thread-4] DEBUG activitystreamer.server.Control Read in command: AUTHENTICATION_SUCCESS
00:07:10.827 [Thread-4] INFO  activitystreamer.server.Control I just set up my deadpool with server: localhost@port3780with secret code: aaa
00:07:16.644 [Thread-0] DEBUG activitystreamer.server.Control Announce to other servers:
00:07:16.644 [Thread-0] DEBUG activitystreamer.server.Control List of Connected Servers
00:07:16.644 [Thread-0] DEBUG activitystreamer.server.Control Server localhost/127.0.0.1:3781 3781connected
00:07:16.645 [Thread-0] DEBUG activitystreamer.server.Control Announcing: {"hostname":"localhost","load":"1","port":"3785","id":"fff","command":"SERVER_ANNOUNCE"}
00:07:16.645 [Thread-0] DEBUG activitystreamer.server.Connection Message sent: {"hostname":"localhost","load":"1","port":"3785","id":"fff","command":"SERVER_ANNOUNCE"}
Received message from client
00:07:26.685 [Thread-4] DEBUG activitystreamer.server.Connection Received message from client: {"hostname":"localhost","load":"3","port":"3781","id":"bbb","command":"SERVER_ANNOU
00:07:26.686 [Thread-4] DEBUG activitystreamer.server.Control Read in command: SERVER_ANNOUNCE
00:07:26.686 [Thread-4] DEBUG activitystreamer.server.Control IN SERVER_ANNOUNCE:
00:07:26.686 [Thread-4] DEBUG activitystreamer.server.Control load: 3
00:07:26.686 [Thread-4] DEBUG activitystreamer.server.Control port: 3781
00:07:36.649 [Thread-0] DEBUG activitystreamer.server.Control Announce to other servers:
00:07:36.649 [Thread-0] DEBUG activitystreamer.server.Control List of Connected Servers
00:07:36.649 [Thread-0] DEBUG activitystreamer.server.Control Server localhost/127.0.0.1:3781 3781connected
00:07:36.649 [Thread-0] DEBUG activitystreamer.server.Control Announcing: {"hostname":"localhost","load":"1","port":"3785","id":"fff","command":"SERVER_ANNOUNCE"}
```

Figure 13: Server e after d death.



```
00:07:02.084 [Thread-0] DEBUG activitystreamer.server.Control Server localhost/127.0.0.1:3783 3783connected
00:07:02.084 [Thread-0] DEBUG activitystreamer.server.Control Announcing: {"hostname":"localhost","load":"1","port":"3784","id":"eee","command":"SERVER_ANNOUNCE"}
00:07:02.084 [Thread-0] DEBUG activitystreamer.server.Connection Message sent: {"hostname":"localhost","load":"1","port":"3784","id":"eee","command":"SERVER_ANNOUNCE"}
00:07:10.805 [Thread-3] DEBUG activitystreamer.server.Connection connection closed to localhost/127.0.0.1:3783
00:07:10.806 [Thread-3] DEBUG activitystreamer.server.Control --------Server at remote port: localhost/127.0.0.1:3783has failed, RIP--------------
00:07:10.807 [Thread-3] INFO  activitystreamer.server.Control runtime server called
00:07:10.815 [Thread-3] DEBUG activitystreamer.server.Control outgoing connection to: localhost/127.0.0.1:3781
00:07:10.818 [Thread-3] DEBUG activitystreamer.server.Connection Message sent: {"mysecret":"eee","hostname":"localhost","port":"3784","secret":"bbb","command":"AUTHENTICATE"}
00:07:10.818 [Thread-4] DEBUG activitystreamer.server.Connection Connection in operation
00:07:10.823 [Thread-3] INFO  activitystreamer.server.Control Connected to backup serverServer: localhost/127.0.0.1:3781
00:07:10.825 [Thread-3] DEBUG activitystreamer.server.Control Initiating deadpool reconect to:localhost@ port3781
00:07:10.823 [Thread-4] DEBUG activitystreamer.server.Connection term in Connection: false
Received message from client
00:07:10.827 [Thread-4] DEBUG activitystreamer.server.Connection Received message from client: {"hostname":"localhost","port":"3780","secret":"aaa","command":"AUTHENTICATION_SUCCES
00:07:10.827 [Thread-4] DEBUG activitystreamer.server.Control Read in command: AUTHENTICATION_SUCCESS
00:07:10.827 [Thread-4] INFO  activitystreamer.server.Control I just set up my deadpool with server: localhost@port3780with secret code: aaa
00:07:22.089 [Thread-0] DEBUG activitystreamer.server.Control Announce to other servers:
00:07:22.089 [Thread-0] DEBUG activitystreamer.server.Control List of Connected Servers
00:07:22.089 [Thread-0] DEBUG activitystreamer.server.Control Server localhost/127.0.0.1:3781 3781connected
00:07:22.089 [Thread-0] DEBUG activitystreamer.server.Control Announcing: {"hostname":"localhost","load":"1","port":"3784","id":"eee","command":"SERVER_ANNOUNCE"}
00:07:22.090 [Thread-0] DEBUG activitystreamer.server.Connection Message sent: {"hostname":"localhost","load":"1","port":"3784","id":"eee","command":"SERVER_ANNOUNCE"}
Received message from client
00:07:26.683 [Thread-4] DEBUG activitystreamer.server.Connection Received message from client: {"hostname":"localhost","load":"3","port":"3781","id":"bbb","command":"SERVER_ANNOUNC
00:07:26.683 [Thread-4] DEBUG activitystreamer.server.Control Read in command: SERVER_ANNOUNCE
00:07:26.684 [Thread-4] DEBUG activitystreamer.server.Control IN SERVER_ANNOUNCE:
00:07:26.684 [Thread-4] DEBUG activitystreamer.server.Control load: 3
00:07:26.684 [Thread-4] DEBUG activitystreamer.server.Control port: 3781
00:07:42.091 [Thread-0] DEBUG activitystreamer.server.Control Announce to other servers:
00:07:42.091 [Thread-0] DEBUG activitystreamer.server.Control List of Connected Servers
00:07:42.091 [Thread-0] DEBUG activitystreamer.server.Control Server localhost/127.0.0.1:3781 3781connected
00:07:42.091 [Thread-0] DEBUG activitystreamer.server.Control Announcing: {"hostname":"localhost","load":"1","port":"3784","id":"eee","command":"SERVER_ANNOUNCE"}
00:07:42.091 [Thread-0] DEBUG activitystreamer.server.Connection Message sent: {"hostname":"localhost","load":"1","port":"3784","id":"eee","command":"SERVER_ANNOUNCE"}
Received message from client
00:07:46.686 [Thread-4] DEBUG activitystreamer.server.Connection Received message from client: {"hostname":"localhost","load":"3","port":"3781","id":"bbb","command":"SERVER_ANNOUNC
00:07:46.687 [Thread-4] DEBUG activitystreamer.server.Control Read in command: SERVER_ANNOUNCE
00:07:46.687 [Thread-4] DEBUG activitystreamer.server.Control IN SERVER_ANNOUNCE:
00:07:46.687 [Thread-4] DEBUG activitystreamer.server.Control load: 3
00:07:46.687 [Thread-4] DEBUG activitystreamer.server.Control port: 3781
00:08:02.094 [Thread-0] DEBUG activitystreamer.server.Control Announce to other servers:
00:08:02.095 [Thread-0] DEBUG activitystreamer.server.Control List of Connected Servers
00:08:02.095 [Thread-0] DEBUG activitystreamer.server.Control Server localhost/127.0.0.1:3781 3781connected
00:08:02.095 [Thread-0] DEBUG activitystreamer.server.Control Announcing: {"hostname":"localhost","load":"1","port":"3784","id":"eee","command":"SERVER_ANNOUNCE"}
```