

## Report - Server Side Events

In this report, I have described code along with explanation of the implementation both locally and remotely.

Joining the nodes in distributed hash table to communicate between them.

Used listenOn, listenOff and listeners to communicate between them,

Adding and retrieving bindings at both locally and remotely.

### CODE:

In POM file, server.name property has been defined (local/home) and jar file generates after run the project at specified location.

```
<properties>
  <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
  <!-- HTTP server listens on this port. -->
  <server.port.http>8080</server.port.http>
  <!-- External DNS for AWS. -->
  <server.external-dns>localhost</server.external-dns>
  <!-- Where to distribute jar files (using mvn package). -->
  <server.home>${user.home}</server.home>
  <server.testdir>${server.home}/tmp/cs549/dht-test</server.testdir>
  <server.dist>${server.testdir}</server.dist>
  <server.name>dht</server.name>
</properties>
```

In Application file, register the server side events.

```
public class Application extends ResourceConfig {
    public Application() {
        super(NodeResource.class, SseFeature.class);
        packages("edu.stevens.cs549.dhts.resource");
        // TODO register SseFeature
    }
}
```

In WebClient class file, we put listenForBinding and listenOff methods. ListenForBinding tells another node that whatever changes happen to any particular key, whichever node binds to that key will be notified.

```
public EventSource listenForBindings(NodeInfo node, int id, String skey) throws
DHTBase.Failed {
    // TODO listen for SSE subscription requests on http://.../dht/listen?key=<key>
    // On the service side, don't expect LT request or response headers for this request.
    // Note: "id" is client's id, to enable us to stop event generation at the server.
    String uri = String.format(node.addr + UriApi.LISTEN, id, skey);
    WebTarget target = listenClient.target(uri);
    EventSource eventsrc = new EventSource(target, false);
    return eventsrc;
}
```

ListenOff method is useful to get disconnect from node to given key. ListenOff to node will no longer eligible to get notified.

```
public void listenOff(NodeInfo node, int id, String skey) throws
DHTBase.Failed {
// TODO listen for SSE subscription requests on http://.../dht/listen?key=<key>
// On the service side, don't expect LT request or response headers for this request.
    String uri = String.format(node.addr + UriApi.LISTEN, id, skey);
    try {
        deleteRequest(new URI(uri));
    } catch (Exception e) {
        throw new DHTBase.Failed("listenOff error. message: " + e);
    }
}
```

deleteRequest and putRequest methods:

```
public Response putRequest(URI uri, Entity<?> entity) {
// TODO Complete.
    try{
        Response cr = client.target(uri)
            .request(MediaType.APPLICATION_XML_TYPE)
            .header(Time.TIME_STAMP, Time.advanceTime())
            .put(entity);
        processResponseTimestamp(cr);
        return cr;
    } catch (
        Exception e) {
        error("Exception during Put request: " + e);
        return null;
    }
}

public Response deleteRequest(URI uri) {
    Response cr = client.target(uri)
        .request(MediaType.APPLICATION_XML_TYPE)
        .header(Time.TIME_STAMP, Time.advanceTime())
        .delete();
    processResponseTimestamp(cr);
    return cr;
}
```

In State class file added addlisteners and remove listeners methods. Which adds if map contains the key into the node.

```
@Override
public void addListener(int id,String key,EventOutput eventopt) {
    if(outputs.containsKey(id)){
        outputs.get(id).put(key, eventopt);
    } else {
        Map<String,EventOutput> map = new HashMap<String,EventOutput>();
```

```

        map.put(key,eventopt);
        outputs.put(id,map);
    }
    if(listeners.containsKey(key)){
        listeners.get(key).add(eventopt);
    } else {
        Map<String,EventOutput> map = new HashMap<String,EventOutput>();
        SseBroadcaster b = new SseBroadcaster();
        b.add(eventopt);
        listeners.put(key,b);
    }
}

```

Simply remove nodes form the node in hash table.

```

public void removeListener(int id, String key) {
    // TODO Close the event output stream.
    if(listeners.containsKey(key)){
        listeners.get(key).remove(outputs.get(id).get(key));
    }
}

```

listenOn and listenOff methods in Dht class file.

This method register a new node for new bindings. Client send request to identify this node.

```

public void listenOn(String key, EventListener listener) throws DHTBase.Failed {
    int id = NodeKey(key);
    NodeInfo succ = this.findSuccessor(id);
    EventSource eventsrc = client.listenForBindings(succ, info.id, key);
    state.addCallback(key, eventsrc);
    eventsrc.register(listener);
    eventsrc.open();
}

```

This method stops listening to binding the node.

```

public void listenOff(String key) throws DHTBase.Failed {
    int id = NodeKey(key);
    NodeInfo succ = this.findPredecessor(id);
    client.listenOff(succ, info.id, key);
    state.removeCallback(key);
}

```

#### Local Test:

ListenOn, listenOff and listeners command to communicate in has table between nodes.

```
Terminal
keyur@ubuntu: ~/tmp/cs549/dht-test
Oct 29, 2016 5:51:58 PM edu.stevens.cs549.dhts.main.Main info
INFO: Starting state server...
Oct 29, 2016 5:51:58 PM edu.stevens.cs549.dhts.main.Main info
INFO: State server bound.
Oct 29, 2016 5:51:58 PM edu.stevens.cs549.dhts.main.Main info
INFO: Starting HTTP server.
Oct 29, 2016 5:52:00 PM org.glassfish.jersey.server.ApplicationHandler initialize
INFO: Initiating Jersey application, version Jersey: 2.1 2013-07-18 18:14:16...
Oct 29, 2016 5:52:03 PM org.glassfish.grizzly.http.server.NetworkListener start
INFO: Started listener bound to [localhost:8181]
Oct 29, 2016 5:52:03 PM org.glassfish.grizzly.http.server.HttpServer start
INFO: (HttpServer) Started.
Oct 29, 2016 5:52:03 PM edu.stevens.cs549.dhts.main.Main info
INFO: DHT Web service started with WADL available at http://localhost:8181/appl
cation.wadl
Try out http://localhost:8181/dht
Entering command-line interface...
dht<23> join http://localhost:8080/dht
dht<23> listenOn foo
dht<23> ** Binding event (new-binding): foo -> test2
** Binding event (new-binding): foo -> test3
add foo test1
dht<23>
keyur@ubuntu: ~/tmp/cs549/dht-test
Oct 29, 2016 5:52:13 PM edu.stevens.cs549.dhts.main.Main info
INFO: Starting state server...
Oct 29, 2016 5:52:13 PM edu.stevens.cs549.dhts.main.Main info
INFO: State server bound.
Oct 29, 2016 5:52:14 PM edu.stevens.cs549.dhts.main.Main info
INFO: Starting HTTP server.
Oct 29, 2016 5:52:16 PM org.glassfish.jersey.server.ApplicationHandler initialize
INFO: Initiating Jersey application, version Jersey: 2.1 2013-07-18 18:14:16...
Oct 29, 2016 5:52:18 PM org.glassfish.grizzly.http.server.NetworkListener start
INFO: Started listener bound to [localhost:8282]
Oct 29, 2016 5:52:18 PM org.glassfish.grizzly.http.server.HttpServer start
INFO: (HttpServer) Started.
Oct 29, 2016 5:52:18 PM edu.stevens.cs549.dhts.main.Main info
INFO: DHT Web service started with WADL available at http://localhost:8282/appl
cation.wadl
Try out http://localhost:8282/dht
Entering command-line interface...
dht<37> join http://localhost:8080/dht
dht<37> listenOn foo
dht<37> ** Binding event (new-binding): foo -> test2
add foo test3
** Binding event (new-binding): foo -> test3
dht<37>
keyur@ubuntu: ~/tmp/cs549/dht-test
Oct 29, 2016 5:51:43 PM edu.stevens.cs549.dhts.main.Main info
INFO: Starting state server...
Oct 29, 2016 5:51:43 PM edu.stevens.cs549.dhts.main.Main info
INFO: State server bound.
Oct 29, 2016 5:51:43 PM edu.stevens.cs549.dhts.main.Main info
INFO: Starting HTTP server.
Oct 29, 2016 5:51:45 PM org.glassfish.jersey.server.ApplicationHandler initialize
INFO: Initiating Jersey application, version Jersey: 2.1 2013-07-18 18:14:16...
Oct 29, 2016 5:51:48 PM org.glassfish.grizzly.http.server.NetworkListener start
INFO: Started listener bound to [localhost:8080]
Oct 29, 2016 5:51:48 PM org.glassfish.grizzly.http.server.HttpServer start
INFO: (HttpServer) Started.
Oct 29, 2016 5:51:48 PM edu.stevens.cs549.dhts.main.Main info
INFO: DHT Web service started with WADL available at http://localhost:8080/appl
cation.wadl
Try out http://localhost:8080/dht
Entering command-line interface...
dht<45> add foo test1
dht<45> add foo test2
dht<45>
```

## Remote Test:

Transfer dht.jar file on amazon ec2 instance using Secure Copy (Scp) and connect that instance using Secure Shell (SSH).

```
Terminal
keyur@ubuntu:~$ ssh -i /home/keyur/cs549-ssr-key-pair.pem ec2-user@ec2-54-175-41-118.compute-1.amazonaws.com
Last login: Sun Oct 30 12:47:16 2016 from c-73-150-1-99.hsd1.nj.comcast.net

Amazon Linux AMI

https://aws.amazon.com/amazon-linux-ami/2016.03-release-notes/
4 package(s) needed for security, out of 60 available
Run "sudo yum update" to apply all updates.
Amazon Linux version 2016.09 is available.
[ec2-user@ip-172-31-7-164 ~]$ exit
logout
Connection to ec2-54-175-41-118.compute-1.amazonaws.com closed.
keyur@ubuntu:~$ scp -i /home/keyur/cs549-ssr-key-pair.pem /home/keyur/tmp/cs549/dht-test/dht.jar ec2-user@ec2-54-175-41-118.compute-1.amazonaws.com:
dht.jar
100% 7190KB 2.3MB/s 00:03
keyur@ubuntu:~$

keyur@ubuntu:~$ ssh -i /home/keyur/cs549-ssr-key-pair.pem ec2-user@ec2-54-157-23-94.compute-1.amazonaws.com
Last login: Sun Oct 30 12:48:39 2016 from c-73-150-1-99.hsd1.nj.comcast.net

Amazon Linux AMI

https://aws.amazon.com/amazon-linux-ami/2016.03-release-notes/
4 package(s) needed for security, out of 60 available
Run "sudo yum update" to apply all updates.
Amazon Linux version 2016.09 is available.
[ec2-user@ip-172-31-7-163 ~]$ exit
logout
Connection to ec2-54-157-239-94.compute-1.amazonaws.com closed.
keyur@ubuntu:~$ scp -i /home/keyur/cs549-ssr-key-pair.pem /home/keyur/tmp/cs549/dht-test/dht.jar ec2-user@ec2-54-157-239-94.compute-1.amazonaws.com:
dht.jar
100% 7190KB 1.8MB/s 00:04
keyur@ubuntu:~$

ec2-user@ip-172-31-7-162:~$
https://aws.amazon.com/amazon-linux-ami/2016.03-release-notes/
4 package(s) needed for security, out of 60 available
Run "sudo yum update" to apply all updates.
Amazon Linux version 2016.09 is available.
[ec2-user@ip-172-31-7-162 ~]$ exit
logout
Connection to ec2-54-159-85-84.compute-1.amazonaws.com closed.
keyur@ubuntu:~$ scp -i /home/keyur/cs549-ssr-key-pair.pem /home/keyur/tmp/cs549/dht-test/dht.jar ec2-user@ec2-54-159-85-84.compute-1.amazonaws.com:
dht.jar
100% 7190KB 1.8MB/s 00:04
keyur@ubuntu:~$ ssh -i /home/keyur/cs549-ssr-key-pair.pem ec2-user@ec2-54-159-85-84.compute-1.amazonaws.com
Last login: Sun Oct 30 12:53:00 2016 from c-73-150-1-99.hsd1.nj.comcast.net

Amazon Linux AMI

https://aws.amazon.com/amazon-linux-ami/2016.03-release-notes/
4 package(s) needed for security, out of 60 available
Run "sudo yum update" to apply all updates.
Amazon Linux version 2016.09 is available.
[ec2-user@ip-172-31-7-162 ~]$
```

On Ec2 instances, three nodes perform DHT commands.

```
Terminal
ec2-user@ip-172-31-7-164:~$
Oct 30, 2016 12:55:02 PM edu.stevens.cs549.dhts.main.Main info
INFO: State server bound.
Oct 30, 2016 12:55:02 PM edu.stevens.cs549.dhts.main.Main info
INFO: Starting HTTP server.
Oct 30, 2016 12:55:03 PM org.glassfish.jersey.server.ApplicationHandler initialize
INFO: Initiating Jersey application, version Jersey: 2.1 2013-07-18 18:14:16...
Oct 30, 2016 12:55:05 PM org.glassfish.grizzly.http.server.NetworkListener start
INFO: Started listener bound to [172.31.7.164:9090]
Oct 30, 2016 12:55:05 PM org.glassfish.grizzly.http.server.HttpServer start
INFO: [HttpServer] Started.
Oct 30, 2016 12:55:05 PM edu.stevens.cs549.dhts.main.Main info
INFO: DHT Web service started with WADL available at http://172.31.7.164:9090/application.wadl
Try out http://172.31.7.164:9090/dht
Entering command-line interface...
dht<30> join http://172.31.7.162:9292/dht
dht<30> listenOn foo
dht<30> ** Binding event (new-binding): foo -> test3
** Binding event (new-binding): foo -> test4
Listeners defined for:
foo
dht<30> [ ]

ec2-user@ip-172-31-7-163:~$
INFO: Starting HTTP server.
Oct 30, 2016 12:55:21 PM org.glassfish.jersey.server.ApplicationHandler initialize
INFO: Initiating Jersey application, version Jersey: 2.1 2013-07-18 18:14:16...
Oct 30, 2016 12:55:23 PM org.glassfish.grizzly.http.server.NetworkListener start
INFO: Started listener bound to [172.31.7.163:9191]
Oct 30, 2016 12:55:23 PM org.glassfish.grizzly.http.server.HttpServer start
INFO: [HttpServer] Started.
Oct 30, 2016 12:55:23 PM edu.stevens.cs549.dhts.main.Main info
INFO: DHT Web service started with WADL available at http://172.31.7.163:9191/application.wadl
Try out http://172.31.7.163:9191/dht
Entering command-line interface...
dht<40> join http://172.31.7.162:9292/dht
dht<40> listenOn foo
dht<40> ** Binding event (new-binding): foo -> test3
add foo test4
** Binding event (new-binding): foo -> test4
dht<40> add foo1 temp1
dht<40> add foo1 temp2
dht<40> listeners
Listeners defined for:
foo
dht<40> [ ]

ec2-user@ip-172-31-7-162:~$
INFO: State server bound.
Oct 30, 2016 12:54:39 PM edu.stevens.cs549.dhts.main.Main info
INFO: Starting HTTP server.
Oct 30, 2016 12:54:40 PM org.glassfish.jersey.server.ApplicationHandler initialize
INFO: Initiating Jersey application, version Jersey: 2.1 2013-07-18 18:14:16...
Oct 30, 2016 12:54:42 PM org.glassfish.grizzly.http.server.NetworkListener start
INFO: Started listener bound to [172.31.7.162:9292]
Oct 30, 2016 12:54:42 PM org.glassfish.grizzly.http.server.HttpServer start
INFO: [HttpServer] Started.
Oct 30, 2016 12:54:42 PM edu.stevens.cs549.dhts.main.Main info
INFO: DHT Web service started with WADL available at http://172.31.7.162:9292/application.wadl
Try out http://172.31.7.162:9292/dht
Entering command-line interface...
dht<50> add foo test1
dht<50> add foo test2
dht<50> add foo test3
dht<50> listenOn foo1
dht<50> ** Binding event (new-binding): foo1 -> temp2
Listeners defined for:
foo1
dht<50> [ ]
```