## Report for week6,7

**Software defined measurement**

* **Goal:** create a software measurement system which can be used to analysis the data from the database in which there is bandwidth,aggregate and latency of the network. We create our algorithm to make decision of the source host and destination host for migration.
* **Process:**

Target host sets statistics

With different frequency

Preliminary

migration decision:

Source host set

Destination host set

Customized mode

Data analysis

(algorithm1)

Migration System:

Do migration

Send command to migration system

Data analysis

(algorithm2)

Database

(data collection as default mode)

Final migration decision:

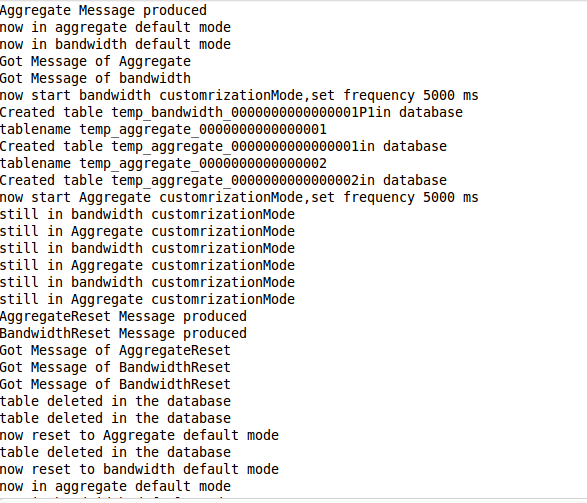
Source host

Destination host

* **Current Achievement:**

Database, statistics collection in default mode, statistics collection in Customized mode, migration System

* **Results**
* **Out put log**

****

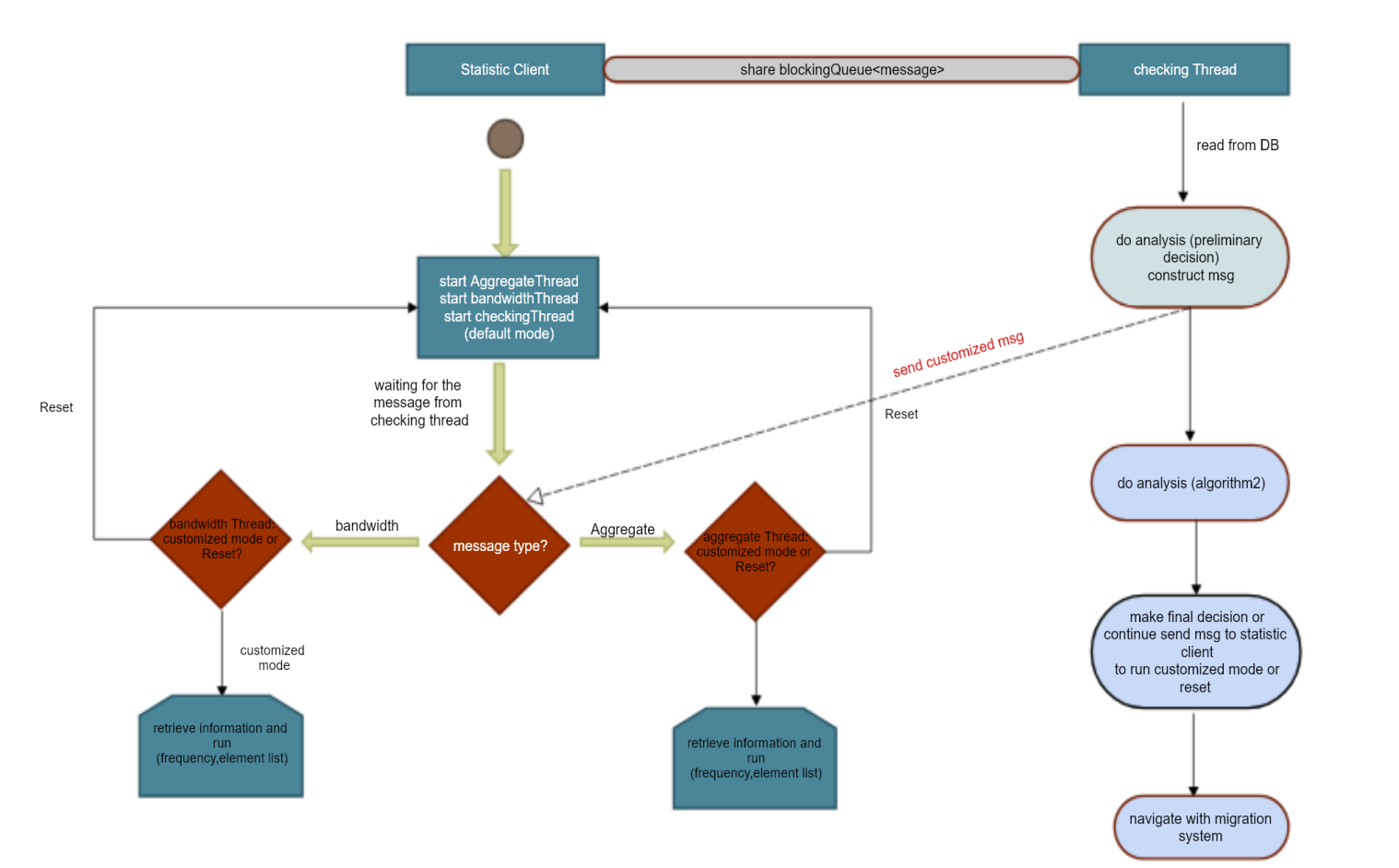
* **Temporary generated table(set the frequence 5s, one table correspond to one switch in customized mode )**

****

* **Reset Message**

The system will delete the temporary table from the database

* **Customized mode implementation details(in gray block):**



UML:

Java Class Explanation in SDN\_Statistics:

Application.java: the main thread of the system.

Package org.sdn.client:

Aggregate Thread: A thread that collects the packets aggregates statistics and store in the database. There are two modes, one is defaults mode, it collects all the switches aggregate package number and also the package difference according to the frequency specified in the configuration file. In the customized mode, it creates the temporary tables for each specified switch using a different frequency in order to monitor one switch precisely.

Bandwidth Thread: A thread that collects the bandwidth consumption per switch per port for both the send and receive speed. These statistics are stored in a database. There are two modes, one is default mode as aggregate thread, it collects all the switch ports bandwidth using the default frequency specified in the configuration file.Another mode is customize mode, it receives the port lists, and create a table for each port, monitor it using a lower frequency in order to analysis the bandwidth consumption precisely.

Checking Thread: It checks the statistics according to different policies, for random, it will directly run the policy executor. And for the bandwidth and shortest path policy, it will execute the Migration Source host Decision algorithm. When it gets the migration source host, it will run the policy executor passing migration source host as a parameter.

Statistics Client: It creates all the statistics threads and also exchange the message between checking threads and bandwidth thread, checking thread and aggregate thread.

Org.sdn.migrationPolicy

Policy Executor: it executes different policy algorithm according to different policy.

Org.sdn.system:

HostInformationCollector.java: it collect the host information and link bandwidth store them in database.

systemInitialize.java: it load the configuration file to the system.

Org.sdn.tcpClient:

tcpClient.java: It opens TCP connections to host to send start container and migration command.

Java Class in migrationServer:

ServerApplication.java: it runs the main method of the migration server.

migrationServer:

Server.java: it loads the configuration file to the system.

ClientWorker.java: Each time migration server receive a TCP connection, it will create a new client worker thread to execute the command form SDN\_Statistics or

another host.

If it receives “START” command, it will deploy the a new docker container in the host.

If it receives ”MIGRATE” command, it will receive the migration destination IP Address.

If it receives “RESTART” command, this command comes from another host and it will receives the image files to restart the container.

Decision made Process:

When Statistic Client starts running, first it retrieves the policy configured in configuration file, if it is random policy, it will start checking thread. If it is bandwidth or shortest path, it will start aggregate thread, bandwidth thread, checking thread. In the case of random policy, checking thread will directly run policy executor. In case of bandwidth and shortest path policy, checking thread will first make migration source host decision, then run policy executor.

In order to make migration source decision,it retrieves all the switches aggregate package number, and select a set of switches which both the package aggregate number and the package difference between current time and last time are exceed the threshold in the configuration file. After it selects the set of target migration source switches which running the server {S1, S2}. it will enter the second checking phase, in this phase, the aggregate thread enter customized mode, which checking thread create a table for each switch and monitor in a smaller frequency, it selected the one which the package difference is keep above 0, and continuing increase. After aggregate thread phase 1 default mode, and phase 2 customized mode.The checking thread have the target migration switch, then it check which port is consuming the most bandwidth. From this port, can get the migration source host.

Migration Source host Decision:

Algorithm1:

Select the target switch set. Factor1:aggregate package number and package number difference.

If the switch is detected by algorithm 1, it means that the switch at the moment,it experience heavy traffic. And it has the potential possibility to be a heavy switch.

Alogithm2:

Select the target host Factor2: Attached switch aggregate package number difference great than 0 constantly and the host consume the most bandwidth

After the migration source host decision is made, the policy executor can execute the migration policy.

Item Definition:

Free hosts: the host which is not running the server

Busy host: the host which is running the server

If it is Random policy, the policy executor randomly select a migration source host from the free hosts, a migration destination from the busy hosts.

If it is bandwidth policy,the policy executor will request path to each free host from floodlight. Then for each path, it calculate the available bandwidth using Max{Min(bandwidth)} function, and it select the one with the maximum bandwidth.

If it is shortest path policy, the policy executor will request the fast path to each free host from the floodlight, then it sorts the latency of all the path. The one with smallest latency will be selected.