Weekly Report - Week 9

Honam Bang and Jens Emil Gydesen

June 2, 2015

1 Project Summary

We are developing a layer 7 load balancing system, codename "Meercat", that uses several load balancers to more efficiently balance the traffic amongst servers in a data center. The idea is that, rather than having a single load balancer, we have several load balancers that communicate with each other to load balance all servers. Each load balancer is able to send traffic to any server and we will use statistics to identify the required resources for each request.

We have looked at the Ananta (L4 software load balancer) paper (reference in the Duet paper from class)[1] and also HAProxy[2].

The goal is to, by having multiple load balancers, to more the system very scalable to a high amount of servers as well as eliminating having a single point of failure.

2 Progress

In implementation part, We've implemented the multi load balancing system. Now we complete the implementation and we are on running several performance measurement. The features what we have implemented are:

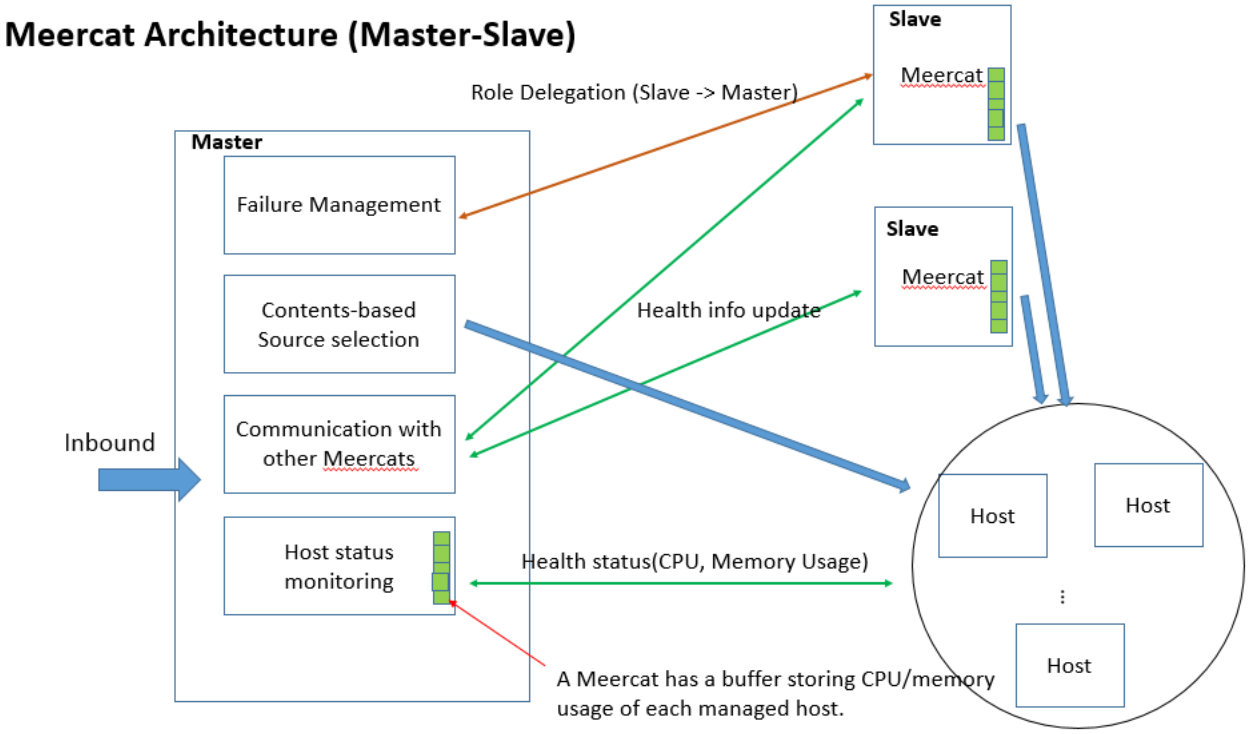
a) A Meercat communicate with all other n-1 Meercats to distribute traffic efficiently

b) A Meercat system consists of one master and multi slaves which are connected to each other.

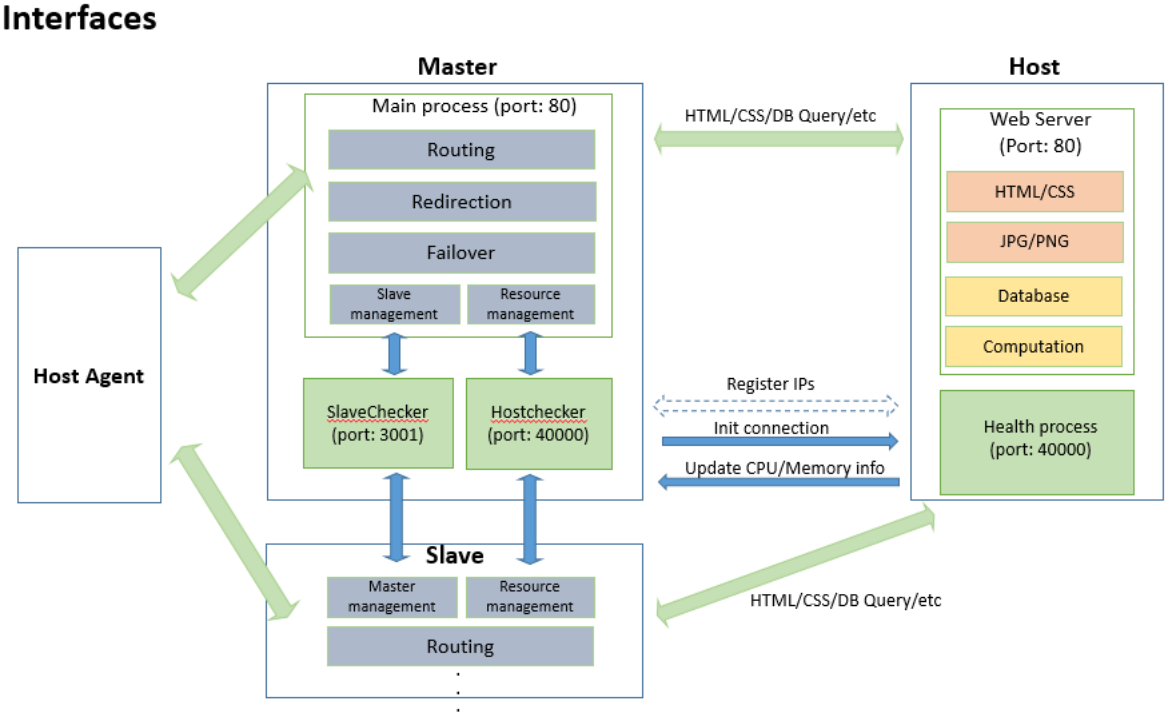
c) A Meercat system takes advantages of multi loadbalancer system. When a master fails (power-off or error) while on running, one of the other slaves automatically takes over the role.

d) A Meercat supports two routing algorithms; Round-Robin and CPU/MEM-usage-based routing. A master checks all CPU/MEM resource usage of hosts and propagates to slaves. All slaves ping to the master every seconds; if no response, the next slave becomes a master.

The communication diagram between a master and slave are described in the below figure:



The only master can listen to resource status of hosts, and can propagates these information to slaves. All slaves ping to a master in every 1 sec. If a master get power-off or down, the next slave automatically takes over the master role (failover). A Meercat has three thread to handle multi channel communication as the below figure shows.



Our current results are not shown yet as the results from our tests showed that we need to optimize/fix the way the master gets information from the hosts. As of right now, this is not happening efficiently enough and a lot requests are sent to servers before we get information about their status.

3 For Next Week

For next week, we will improve the information gathering system and measure the performance of our Meercat. We will also prove whether our initial assumption-multi load balancer system would take advantages than a single load balancer in some specific area - is correct or wrong with several metrics.

References

[1] Patel, Parveen, et al. "Ananta: Cloud scale load balancing." ACM SIGCOMM Computer Communication Review. Vol. 43. No. 4. ACM, 2013.

[2] Tarreau, Willy. "HAProxy-The Reliable, High-Performance TCP/HTTP Load Balancer." 2011-8)[2013-4]. http://haproxy. lwt. eu (2012).