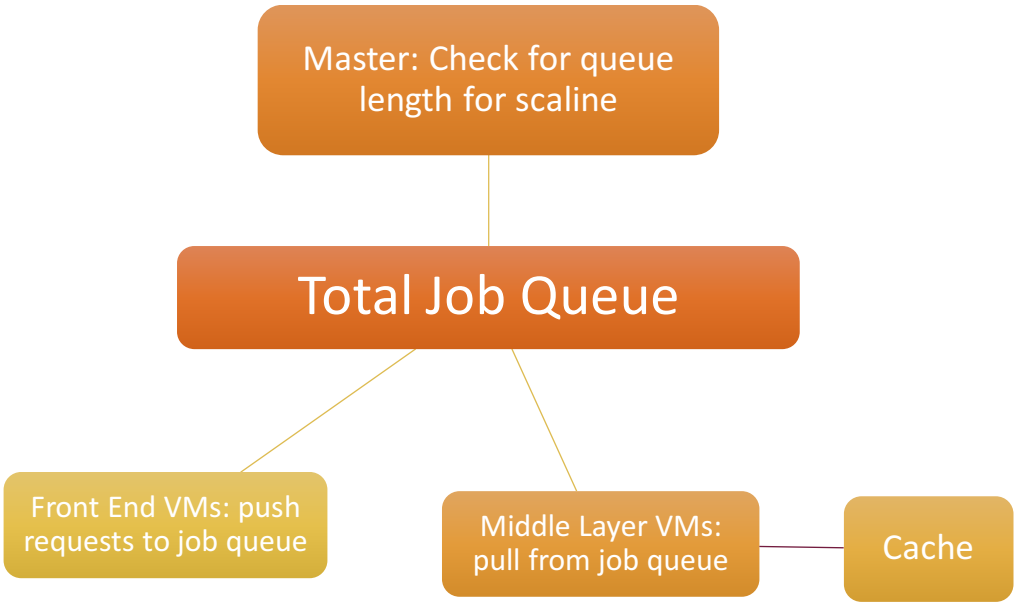
Design Notes

Yuqi Liu <yuqil@andrew.cmu.edu>



Overview of the design

1. Roles for Different Server Instances

In this project, there are four types of instances.

- Master: Master is used for managing VMs and checking for scaling out. All other VMs have to register on master. On master, it also maintains a total job queue for all system. It is in charge out periodically checking the queue length and carry out scale-out if necessary.
- Front-End VM: Front-end VM is used for acquiring requests from load balancer and push the requests to the total job queue located on the master VM. If it finds the frequency for getting a request is very low, it shuts down by itself.
- Middle-Layer VM: Middle-layer VM is used for processing requests. It pulls requests from the total job queue and process it with the help of cache. It will measure the time for getting a request, if it timeouts for three consecutive times, it will shut down itself.
- Cache: Cache all get requests. For all transactional requests, it passes to the real database.

2. Initial Settings for the Number of Servers Launched

In this project, the Initial settings for how many number of servers launched at first is calculated as listed below.

For frontend, we will firstly launch one frontend (co-locate on the master machine). Then we detect the request interval during the booting, if the request interval less than the process time (200 + 60 s), we can launch one more.

For middle layer, we firstly launch only one frontend. Then we use PROCESS_TIME / Request_Interval to estimate how many servers needed. However, we need to be conservative at first so we multiply by a weight (0.6) and let it increase with time.

We only have one cache and will put it on the master as well.

3. Scaling Policy

Scale Out: Master will compare the queue length and existing VMs to determine if it need to scale out. If the request frontend queue exceeds 3 times of the frontend number, it needs to scale out a frontend. If middle layer continuously drop two requests, it should scale out.

Scale In: If the middle layer did not get request for consecutive three times (timeout value 500s, than scale in). If the interval between two front end request exceeds twice of the frontend request process time, then scale in a front end. There is a cool down time after frontend scale out.

4. Database Cache

Database will cache all get request. But for transaction and other write operations, it will pass to backend database VM.

5. Other design decisions

(1) The master will act both as cache and a frontend in order to save VM time.