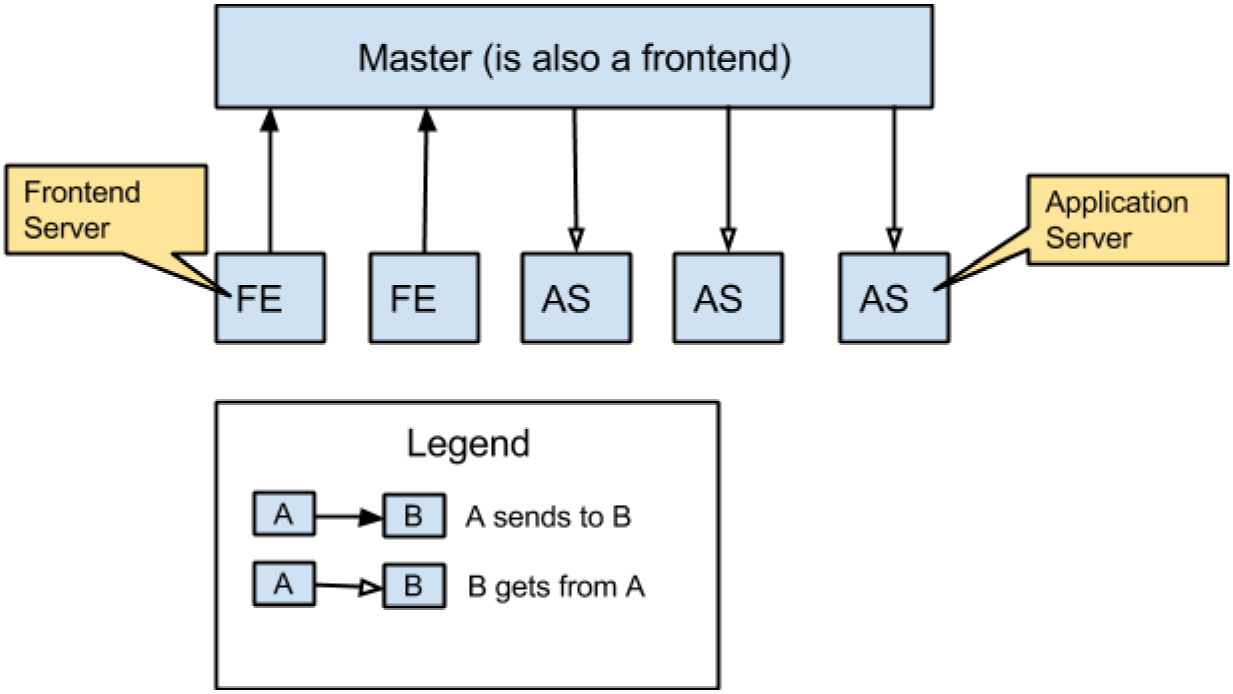
Varun Saravagi
(vsaravag)
Project 3: Design Document

The project has been designed using a Message-passing architecture (see Figure below). The initial server started by Cloud is treated as Master and it acts as our Message queue and also as a frontend. The Master has its own binding to the RMI. All the remaining frontend servers and the application servers connect to the master. The Master has a queue in which it keeps all the requests which are to be processed. The frontend servers add a request to the queue and the application servers take the request from the queue for processing.



Apart from handling the requests, the master also handles the scaling up and scaling down of the frontend and the application servers.

Application Server
Scaling up: The master starts an initial number of application servers based on the time of the day. After that, for the initial 5 seconds, it sees the intervals in which the clients are coming and starts the required number of application servers. After the initial 5 seconds have passed, it runs a maintenance cycle every 1 second and calculates the average load on each application server. The average load is calculated on the basis of the number of requests received per second. If average load is greater than the set threshold for an application server, it starts required number of application servers.

Scaling down: When the maintenance cycle for application servers is run, if the average load on an application server has less than the set threshold for more than 5 seconds, the extra application servers are shut down.

Frontend Server

Scaling up: For every set number of application servers, a new frontend server is started. This is done in two parts: for the initial 5 seconds, it is done when new application servers are started, after that a maintenance cycle for frontends is run every 1 second.

Scaling down: When the maintenance cycle for frontends is run, it shuts down the extra frontends if there are less number of application servers as are required by the frontends.

Starting up a server:

The master keeps a track of whether an application server is to be started or a frontend server in separate data structures. Whenever a server starts, it asks the master for its role and id. The master looks at the data structures and gives the server a role and an id. If both frontend and application server have to be started, first preference is given to the application servers.

Shutting down a server:

Each server binds itself to the RMI via its id. When a server needs to be shutdown, the master gets the id of the server, looks it up on the RMI and sends it the shutdown command. On receiving the shutdown command, the server stops its work and shuts itself down.

Database Cache:

The cache process is started when the master starts. The Cache binds itself to the RMI and implements the methods in the Cloud.DatabaseOps interface. It handles most of the browse requests and passes the misses and transactions to the database. The misses, once read from the database, are added to the cache. Any transaction is passed to the database. After the transaction is successful, the cache is updated so as to maintain consistency.