

Question 1

* In the bottom level, we need to have hardware resource and a resource abstraction and optimization layer to offer a virtual hardware interface.

Upper level would be an operating-system-like manager, which help to allocate the nearest resource and load-balance the hardware usage. At the same time it gives a secure and isolated way to access the system without interfering other customers.

The most upper level offers some high level APIs to public so that customers can make usage of the system, e.g. data store, program interface.

* To know the type of customer's application, is it a data intensive or a compute intensive application. And whether the applications need to utilize some hardware parallelism.

* Variety of physical disk will be used to satisfy different applications, e.g. HDD, SSD, and storage array(s). Thus application with low storage requirement will use HDD to lower the infrastructure cost; and application requiring fast disk access is assigned to the SSD device. The storage array is to create data redundancy to applications which need critical data recovery or fast read.

* First setup a private high-bandwidth network, inside the network I may use one node of hardware as a manager machine to track the usage of every node of hardware, also to do some resource allocation and pooling. At the same time it offers parallelism ability, e.g. map and reduce. One node is easy to added in because it only needs to communicate to the manager nodes and register itself. The communication is made possible by existing popular protocols, e.g. TCP, UDP.

* The new node needs first to plug into local network of the cloud, then it communicate to a configured manager machine to register. Then essential protocol daemon program is called to offer access to other machine nodes.

* First the cloud manager node allocate a node for hardware resource, then new MySQL user, new OS home directory is added with FTP access to that directory, and HTTPd is started at a new port to interact with MySQL and HTML files in that directory. In addition, an public IP is assigned and the gate host toward public network will map packet to that node as it is hosted on the public network. The customer is given the public IP assigned and MySQL hosting port and account information as well as a accessible FTP port or an interface to write HTML files on the located home directory.

Question 2

See the Question 2 directory for the script.

Discussion

1. Because configuration variables can be listed, so I choose to use a array to hold all the defined configuration variables. Instead of using associative array which is only available after Bash 4.0, here I use another array to map a variable name with its value. This makes it easy to add/delete and list configuration variables.

2. Configuration variable can also use Bash variable to implement, but doing so will make it difficult to list all the defined variable, also possible to interfere with environment variables too.

3. In terms of variable conflict, of course a module can use `cfg_var_get()` API to know if a variable name will conflict with those in other modules. But the main problem here is the way I choose to lookup from variable array in Bash is linear speed, therefore large number of module would possibly make the script much slow. By using lower level language instead of Bash and use in-memory tree lookup for variable name would be better to deal with this problem.

4. It is better in the way that it is easy to maintain by different people, if they are writing different part (module). And modules can be easily configured and enabled/disabled. Suppose all the modules will always run together, i.e. they are bind together to achieve the same functionality, in this case there is no need to separate them and call them as separate processes will make the program easy to read too.

Question 3

See the Question 3 directory for the program. You can specify the concurrency limit by passing `-l` option to the program, for example:

```
$ ./a.out -l 2
```

Question 4

See the Question 4 directory for the HTML file and a result screen shot.