Fault Tolerant Distributed Task Scheduler

Requirements

You may use the programming language of your choice.

Create three docker images:

1. *mongodb*
2. *master*: run a master program
3. *slave*: run a slave program

Slave can not access mongodb. Only master can access mongodb.

Create an offline script to generate 100 “tasks” and insert into *mongodb*.
Documents in the Task collection should contain the following fields:

```
{
  taskname: eg task1
  sleeptime: eg 60 seconds
  state: one of ['created', 'running', 'killed', 'success' ]
  host: eg slave1
}
```

You may add other fields or collections as necessary.

- Initially, only the taskname, sleeptime and state fields should be present. taskname and sleeptime should be randomly generated and state='created'.
- Every time a task “runs”, it will sleep for the specified sleeptime seconds.
- We don’t have any a priori estimate of the distribution of sleeptime, so your solution should not depend on sleeptime to decide whether the task has been killed.
- Each slave can work on one task at a time.
- Slaves know the identity of master
- Master should allow any number of slaves to join or leave.
- The scheduler should continue to operate if any component of it becomes unavailable, with the exception of mongodb.
- All tasks eventually finish. The final state in mongodb should be ‘success’.
- Killing master won't affect running tasks.
- Killing a slave will kill tasks running on it
- A killed task will be rerun from scratch (i.e. sleep the whole sleeptime).

Testing

Test with 1 copy of master and 3 copies of slave.
Verify that your system meets the above specified requirements when master or slaves are randomly killed.

Ideally, include a script to test your system in an automated fashion.

Submission

Combine the following in a zip file:

- Source code
- Mongodb entries in a .txt format
- All testing / startup scripts
- Logs from master and slaves from your testing which demonstrate the above requirements are met, and:
    - Master & slaves are re-launched after being killed
    - Task state changes: I.e. placed on a slave, running, killed, finished

Please submit the zip file to the following Dropbox:
https://www.dropbox.com/request/2UrR7Vx985iZLMVEiQCH

~~~~~
~~~~~