Design of Failure Detection

1 Introduction

When Regionserver crashed, it is too long time to notify hmaster. When hmaster know regionserver's shutdown, it is long time to fetch the hlog's lease.

Hbase is an online db, availability is very important.

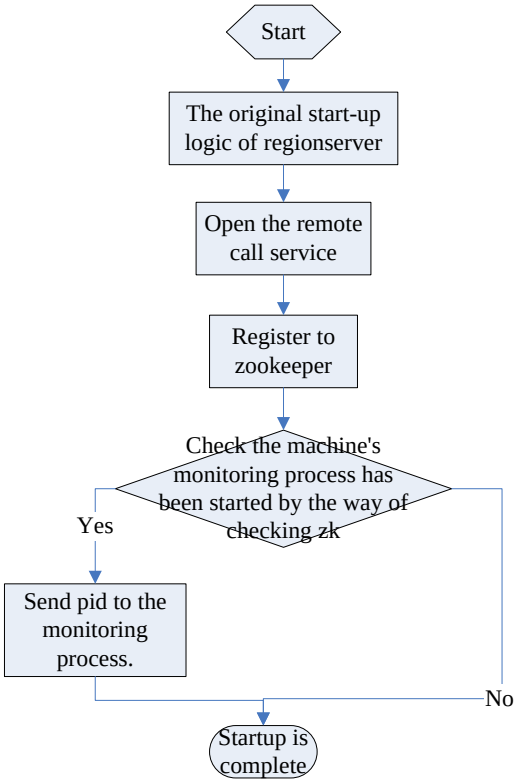
I have an idea to improve availability, monitor node to check regionserver's pid. If this pid does not exist, I think the rs has crushed down, I will delete the znode, and force close the hlog file.so the period maybe 100ms.

2 Motivation

We use a daemon process to monitor whether the RegionServer has crushed or not. If the RegionServer has crushed, the daemon-process will delete znode of the regionserver immediately.

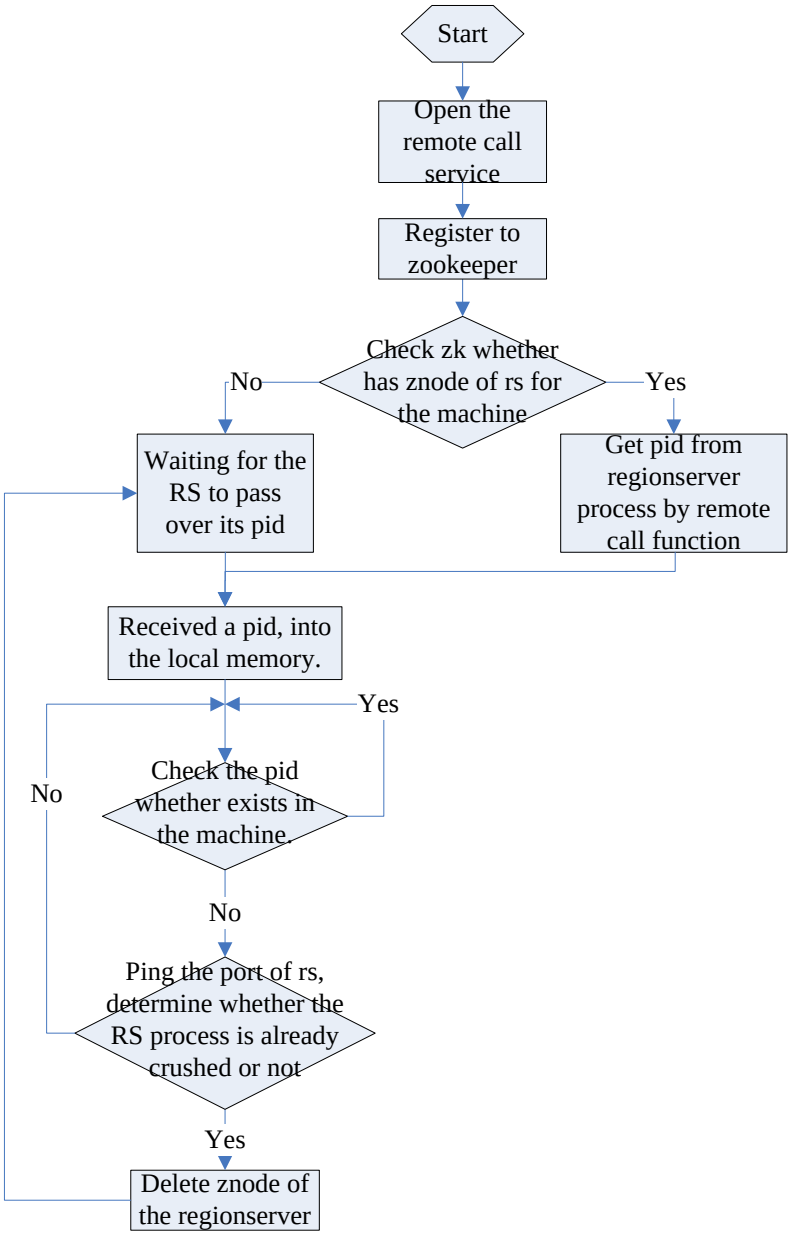
3 Flow Sheet

3.1 RegionServer Flow Sheet



(a) The original start-up logic of regionserver, this is RS's own start-up logic.
(b) Open a remote call service, provide a remote call the function for monitor. Monitor can use this function to get regionserver's pid.
(c) Register to zookeeper
(d) Check monitoring node of the machine whether exist in /hbase/monitor.
(e) If monitor has started, Send pid to the monitoring process.

3.2 Monitor Flow Sheet



(a) Open the remote call service, regionserver can send its pid to monitor process by this service.
(b) Register to zookeeper, register a monitor node in the directory '/hbase/monitor', Its format is similar to host:port
(c) Check zk whether has znode of rs for the machine, if zk has znode of rs, fetch the pid of rs,if

not waiting for the RS to pass over its pid.

(d) After getting pid of the rs in local machine, cycle to monitor the pid exists in the machine.

(e) Because checking pid circularly is not entirely reliable, we use ping the port of rs to determine whether the rs process is already crushed.

(f) When determine rs process has been crushed, delete znode of the regionserver.