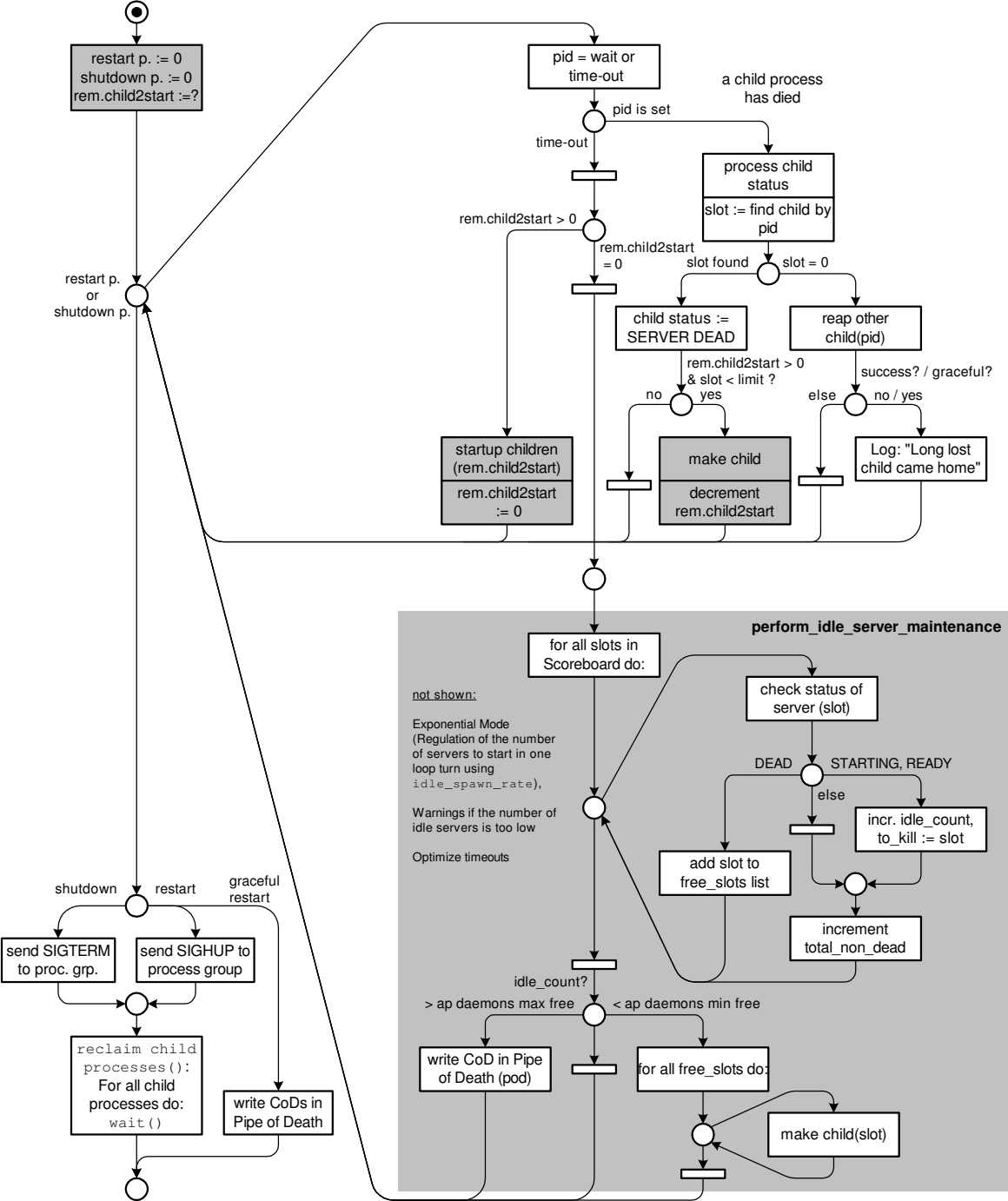
restart p. := 0
shutdown p. := 0
rem.child2start :=?
pid = wait or time-out
a child process has died
pid is set
time-out
process child status
slot := find child by pid
rem.child2start > 0
rem.child2start = 0
slot found
slot = 0
restart p. or shutdown p.
child status := SERVER DEAD
reap other child(pid)
rem.child2start > 0 & slot < limit ?
no
yes
else
success? / graceful?
no / yes
startup children (rem.child2start)
rem.child2start := 0
make child
decrement rem.child2start
Log: "Long lost child came home"
perform_idle_server_maintenance
for all slots in Scoreboard do:
not shown:
Exponential Mode (Regulation of the number of servers to start in one loop turn using idle_spawn_rate),
Warnings if the number of idle servers is too low
Optimize timeouts
check status of server (slot)
DEAD
STARTING, READY
else
incr. idle_count, to_kill := slot
add slot to free_slots list
increment total_non_dead
idle_count?
> ap daemons max free
< ap daemons min free
write CoD in Pipe of Death (pod)
for all free_slots do:
make child(slot)
shutdown
restart
graceful restart
send SIGTERM to proc. grp.
send SIGHUP to process group
reclaim child processes(): For all child processes do: wait()
write CoDs in Pipe of Death