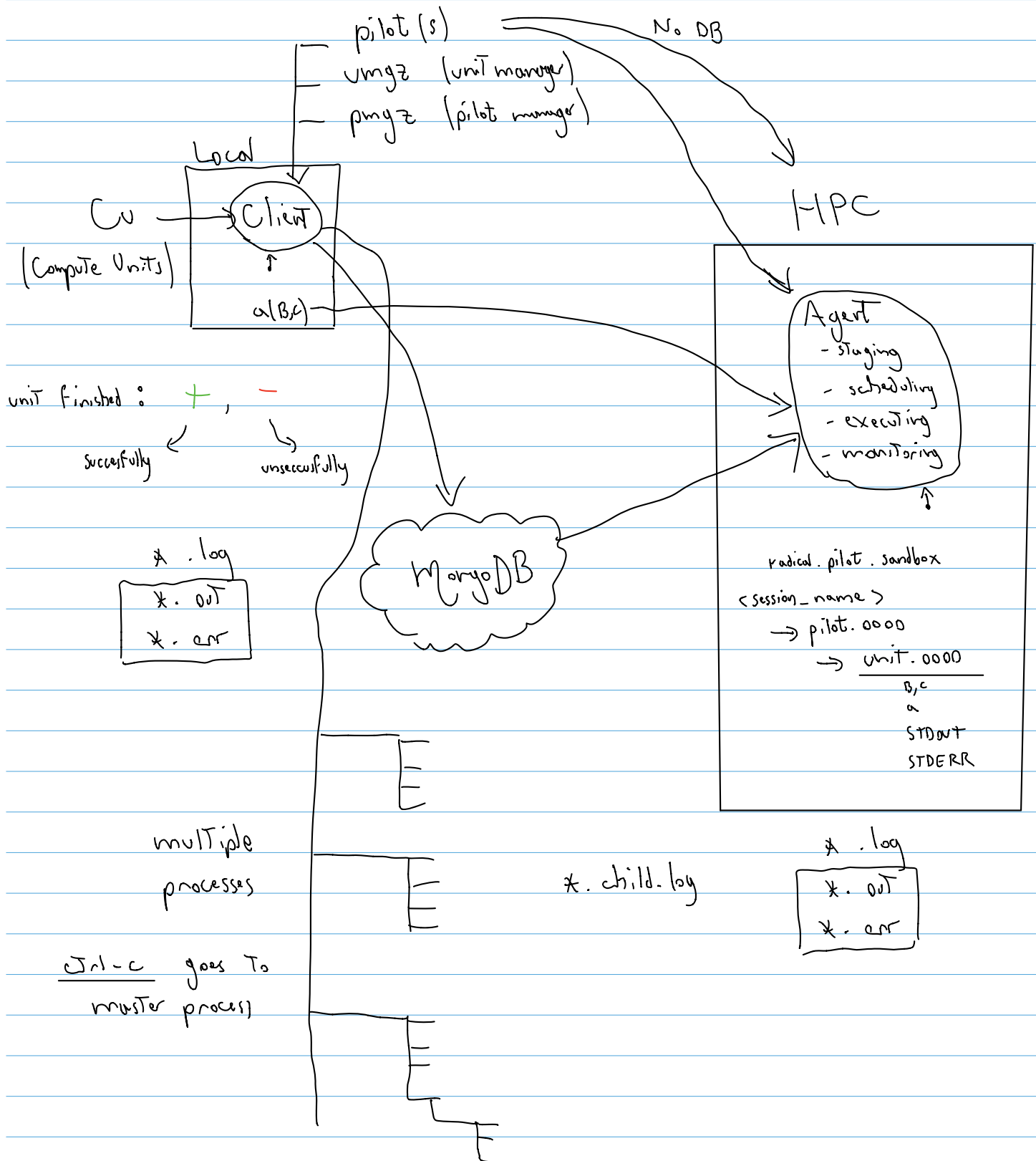


How Radical-Pilot works



if the unit Failed (-) , don't open Tickets! my problem

Because I am a member of The lab :

* _VERBOSE = DEBUG

To run experiments :

* _PROFILE = TRUE

when something fails :

grep ERROR *.log.t.m → [ERROR]: —

if grep does not catch The error , check
 *.out
 *.err

The error with The smallest Time is the one that caused The failure

Common launch pilot failed

client side	{ <ul style="list-style-type: none">- No DB- wrong # of CPU- wrong # of GPU- wrong CUE- wrong resource }	} whatever on the pilot description
-------------	--	-------------------------------------

Common errors on HPC side

- Bootstrapping (if I see no *.log files)

↳ config process (bring the env in the correct state and starts the PT process)

if on the Agent side, my python is an Anaconda distribution:

For the pd (pilot description): resource label:
end it with - anaconda

virtualenv and conda are both fully supported

When To open a Ticket

- Error of MongoDB from Agent side

Why do we use VM's?

- Always on
- Always connected online

Ctrl - c (Kill the process)

DO ONLY ONCE

why?

Cause Ctrl-c kills on the Agent side and slowly goes to the master process. wait at least 1 hr on hour to run again

see what is alive :

```
ps aux | grep *rp*
```

```
ps aux | grep *radical*
```

if clients dies, find job # on HPC and kill it so it doesn't waste resources

process vs threads ?

Ctrl-c on rp example. 0000 on bridges :

kill the parent process.
depending how parent handles kill signal

assume subprocess are not killed by OS

by the time I gain access to my terminal window, everything is fire

if I hit Ctrl-c multiple times, the master process is killed while it waits for the propagation to finish

→ processes alive → resources wasted

→ out of mem → crash

To learn more about O.S., go to : Operating System Concepts book

check your jobs running :

squeue -u Karabit