

RoboCup 3D Soccer Simulation

Organizing Committee Guide

Introduction

This document aims to provide a complete guide to the Organizing Committee (OC) of a RoboCup 3D Soccer Simulation competition. It provides a platform to transfer knowledge and experience gained from past competitions. Examples of content it should have include, but are not limited to: hardware requirements; installation instructions; scheduling guides; tips on rules; experience of extraordinary situations and rulings.

Hardware

Machines

To run a competition, the following list of machines is the *bare minimum* that is required:

- 1x Server machine; as many cores as possible (simspark using ODE-TBB can in theory use 22 cores in parallel, but 8 (or 4 hyperthreaded) cores usually suffices).
- 2x Client/Agent machines; again as many cores as possible, as each will run 11 agents.
- 1x Monitor, keyboard, mouse.
- Decent size HDs for all machines, e.g. 500GB; for the server machine to store match logfiles, for client machines to store team logfiles.

Furthermore, the following is highly advised to have things run more smoothly:

- 1x Visualization machine; mid-range GPU. If not available, the server machine should be equipped with such a GPU and used as the visualization machine.
- Monitor, keyboard and mouse for each available machine.
- 1x File server machine, to host NFS and NIS if used.

The above cluster of machines can be repeated to run multiple matches in parallel. With ~20+ teams at least 2 of these clusters is needed to be able to run all matches. An extra cluster for teams to test on is helpful as well.

Network

- To build a network, a Gigabit switch + NICs and CAT6 cables **must** be used.
- It must be possible to physically disconnect the competition network from the internet/event LAN, which must be done during matches to restrict interference. If using multiple clusters, having the ability to separate them physically may help stability, though when using NFS this means you need a mirroring file server for each cluster.
- The competition machines should have static IPs.

Set-Up

OS

- Choose a recent Linux OS that you are comfortable with, and that you know of that the simulator runs on. Most teams will be familiar with Ubuntu, other popular choice is Fedora.
- Choose a 64 bit OS.
- Have several installation methods ready, with different versions if unsure, and both on disk and on USB drive in case the machines don't support one.

Configuration

- During installation, give each machine a useful host name (e.g. server1, agent1, monitor1).
- Install required packages on all machines. On Ubuntu:

```
$ sudo apt-get install openssh-server
```

- The League Manager (see below) uses the root account to log into the different machines. Ubuntu by default disables this account, to enable it give it a password by running:

```
$ sudo passwd root
```

Do this on the server and client machines

- List all machine IP address and host names in /etc/hosts and distribute it over all machines.
- Set up passwordless SSH. First generate an SSH key on the machine and under the account from which you will run the league manager, usually the visualization machine:

```
$ ssh-keygen
```

Press <Enter> on all questions to select the default values, and to choose not to set a password. This will generate a private and public key. The latter must be distributed to root accounts on the other machines. For this, run:

```
$ ssh-copy-id root@hostname
```

where you select the right hostname. Do this at least for all the machines in the same cluster (i.e. per visualization machine for one server and two client machines). It is also useful to set this up for your own account on all machines, by running the same command without 'root@'.

NFS

NFS is used to make home directories available on all machines.

Firstly, install the server. For Ubuntu, follow the Quick-Start steps in:

<https://help.ubuntu.com/community/SettingUpNFSToHowTo>

If you put all the teams' home folders under /home/teams for instance, you may have the following line in /etc/fstab (copy of RC2013):

```
/home/teams    /export/teams    none    bind    0    0
```

and your /etc/exports may look like this (copy of RC2013):

```
/export        145.144.164.0/24(rw,fsid=0,insecure,no_subtree_check,sync)
/export/teams  145.144.164.0/24(rw,nohide,insecure,no_subtree_check,sync)
```

First get everything running without authentication and the portmap steps if you choose to apply these.

Secondly, install the clients. In theory this is only needed on the client/agent machines. Again, follow the Quick-Start steps linked to above. The following is an example of what you may have in /etc/fstab on the clients (copy of RC2013):

```
145.144.164.11:/teams    /home/teams    nfs    auto,nolock    0    0
```

The nolock option should not be needed, but could maybe prevent lockups (or maybe not). Other mount options that you may try setting are rsize and wsize to set read and write block size, where larger values may speed up things. Also see for instance:

http://www.centos.org/docs/5/html/Deployment_Guide-en-US/s1-nfs-client-config-options.html

And check 'man nfs' to see which defaults are already used. The most important way to get NFS stable however is to prevent any writing, as discussed below.

NIS

NIS is used to make user accounts available on all machines. First, again install the server, then the clients. For this, on Ubuntu follow the instructions on:

<https://help.ubuntu.com/community/SettingUpNISHowTo>

You can initially skip all steps that involve portmap.

Closing Home Directories

When agents write a lot of logging to a team's home directory they will slow down NFS and the network significantly. To prevent this, you should disable writing before each round. First, set up log directories for each team on the client machines by running this script (this and the 2 scripts below assume the team home directories are under /home/teams):

```
#!/bin/bash
sudo mkdir /teamlogs
for t in /home/teams/*
do
    sudo mkdir /teamlogs/$t
    sudo chown $t:$t /teamlogs/$t
    sudo ln -s /teamlogs/$t /home/teams/$t/log
done
```

Now when teams log to this directory, the output is written to the current machine, not to NFS.

To close down the home directories before each round, use the following script:

```
#!/bin/bash
for t in /home/teams/*
do
    chmod a-w $t -R
    chmod u+w $t/log -R
done
```

Finally, to open the home directories again when the round is finished you can use the following:

```
#!/bin/bash
for t in /home/teams/*
do
    chmod u+w $t -R
done
```

RCSServer3D

Main installation instructions are found at

http://simspark.sourceforge.net/wiki/index.php/Installation_on_Linux#Requirement

The following are a reproduction of these, with some additions specific to running a competition. This should be done on all server machines.:

- Install dependency packages. On Ubuntu:

```
$ sudo apt-get install build-essential automake libtool
subversion git cmake libfreetype6-dev libsdl-dev ruby ruby-dev
libdevil-dev libboost-dev libboost-thread-dev libboost-regex-dev
libboost-system-dev
```

- Install multi-threaded ODE:

```
$ git clone https://github.com/sgvandijk/ode-tbb.git
$ cd ode-tbb
$ ./autogen.sh
$ ./configure --enable-shared --disable-demos
--enable-double-precision --disable-asserts --enable-malloc
$ make -j && sudo make install
```

- Install simulator. Download the latest versions of simspark and rcssserver3d from <http://sourceforge.net/projects/simspark/files/>, and do step 4 of the instructions linked to above.

RoboViz

RoboViz is the preferred visualizer, install it on the visualization machine(s). Full installation and usage instructions are on its homepage:

<https://sites.google.com/site/umroboviz/usage/setup>

Proxy

Download and extract the rcssserver proxy onto the client machines from:

<http://robocup.hs-offenburg.de/html/downloads.htm>

Usage instructions are in the included README.txt file. Note that the server must run in sync mode if the proxy is used. To do this, edit the file `/usr/local/share/simspark/spark.rb`, to set `$agentSyncMode` to true. After doing this, make sure to delete an already existing `~/simspark/spark.rb` file before running the simulator again. Start the proxy on all client machines before running a group.

League Manager

Obtain a copy of the latest League Manager version; the version used for the 2013 world championship can be found at:

<http://homepages.feis.herts.ac.uk/~sv08aav/robocup.php>

To install, do the following on each machine you will start the matches from (i.e once for each cluster):

- Download and extract archive.
- Edit the file `types/3Dspark/config` as appropriate for the current cluster: set the right `NET_PREFIX`, set the right IP ending for the `SERVER`, `CLIENT1` and `CLIENT2` variables, turn off the automatic kickoff and quit if you don't want them. The default values for other variables should be sufficient.
- Make and install:

```
$ ./configure && make && sudo make install
```

Now, for each group to run, do the following:

```
$ rclm2 groupname
$ cd groupname
$ ./script/init 3Dspark
$ nano teams
[ enter list of teams in this group, by their username, one on each
line; save & exit]
$ ./script/schedule/init teams
```

To start the group, start RoboViz, and run `./script/start` in the group directory. Some pointers for during running the group:

- After each half, the simulator must be stopped. To do so, type 'end' and press enter in the terminal where the league manager is running. Alternatively, while editing `types/3Dspark/config`, set `USED_MONITOR` to 'internal', or empty; this way the league manager continues automatically when the server is killed (e.g. by pressing shift+X in RoboViz).
- If the league manager asks to run complementary match, choose 'y' (yes) if the match got interrupted erroneously (e.g server crash), or to start a second half.
- To change the schedule by hand (e.g. to rerun a match, or reorder matches), edit the file `var/schedule` in the round directory.
- Logs are compressed and stored automatically in the 'archives' directory in the group folder.