Simulation issues for Dummies - (msmtp)

I. Sticco* and F. Cornes†
Departamento de Física, Facultad de Ciencias Exactas y Naturales, Universidad de Buenos Aires, Pabellón I, Ciudad Universitaria, 1428 Buenos Aires, Argentina.

G.A. Frank‡
Universidad Tecnológica Nacional, Facultad Regional Buenos Aires, Av. Medrano 951, 1179 Buenos Aires, Argentina.
(Dated: March 11, 2016)

This article is intended as a starting point in the art of simulation. We will try to make it easy...

PACS numbers: DuMMy, 1.2.me

I. HOW TO GET AN E-MAIL NOTIFICATION WHEN A JOB ENDS

This an easy trick!

Suppose you drop a job into a cluster. The job execution is expected to last many hours or even many days. You will like recieve an e-mail right after the job finishes. You can configure that with `msmtp`.

We assume that you have a user account in the cluster (no administrator priviledges) and that the `openssl` software is installed in the system. If you are not sure about that, simply type `man openssl` in a terminal. The openssl manual should appear on screen if the software is installed. If not, ask the administrator (perhaps the `openssl` requirement does not need to be fulfilled, but we did not try it).

On the other hand, you need to open a free webmail account. We assume that your webmail account is `mycluster@gmail.com` but any other webmail account is fine. We recommend that you open an account instead of using your personal account, but this is not mandatory.

A. Step 1: install msmtp

`msmtp` is a small program that comunicates via the SMTP protocol to the Gmail server. Since you do not have administrator priviledges, you will need to download the source file. Go to

`https://sourceforge.net/projects/msmtp/files/msmtp/`

and download the latest version. You will get somthing like `msmtp-x.x.x.tar.xz`. Don't worry because it is a small file.

Drop the file into your home directory in the cluster. You will probably need a `ssh` connection. Use the command `scp` to copy the file into the cluster. From now on we will assume thar your home directory is `/home/you` and that the expression `$HOME` is an alias for your home directory (technically, `$HOME` is an environment variable).

To copy the file into the cluster, type from the terminal

```
scp ./msmtp-x.x.x.tar.xz you@cluster:/home/you
```

where `you` is your username and `cluster` is the name of the cluster (obvious). After that, log into your user directory in the cluster (say, `ssh you@cluster`) and uncompress the source file. For example,

```
tar -xf msmtp-x.x.x.tar.xz
```

You will notice that a new directory has been create called `msmtp-x.x.x`. It contains all the source files for building the binaries. Now follow the steps

```
cd msmtp-x.x.x
./configure --prefix=$HOME
make install
```

The `--prefix` option ensures that everything will be inside the limits of the `/home/you` directory.

The installation is almost done! You only need to create a configuration file. Do this by with any text editor (such a `gedit`). Your configuration file should have the name `.msmtprc` (the dot is mandatory to "hide" the file) and it should be located in `/home/you` (or `$HOME` as you wish). For security reasons, you better assign only user read/write permission to the file. Type

```
chmod 600 $HOME/.msmtprc
```

The configuration file is empty till now. Open the file (with a text editor like `gedit`) and include the following information

* ignaciosticco@gmail.com
† fercornes@gmail.com
‡ guillermo.frank@gmail.com

```
account default
host smtp.gmail.com
port 587
from mycluster@gmail.com
tls  on
tls_starttls on
tls_certcheck off
auth on
user mycluster@gmail.com
password xxxxxx
```

Save and exit. The `xxxxxx` is our secret password. The account `mycluster@gmail.com` is your recently created mail account for the cluster.

Configuration is finished, congratulations!

B. Step 2: test it

The installation process is over. You can make a simple test to ensure that everything is working fine. Type the following and see if a new mail apperas in the inbox of webmail

```
echo "Subject: hello" | msmtp mymail@gmail.com
```

where `mymail@gmail.com` is assumed to be your personal account (not the recently opened account `mycluster@gmail`). Thus, if you now check your personal account, you will find an incoming mail from `mycluster@gmail` with the subject `hello`. The body of the mail should be empty.

The above intruction is a pipe, that is, `echo` passes the `"Subject: hello"` to `msmtp`. The `|` character chains both instructions. Thus, `msmtp` knows that the string `hello` is the subject needed to complete the mail to `mymail@gmail.com`.

C. Step 3: automate it

This last step makes the use of `msmtp` more comfortable, although it is not really necesary. You can create a bash file with the following content

```
#!/bin/bash

args=("$@")

if (($# > 1)); then

  i=1
  while [ $i -lt $# ]
  do
     message+=${args[i]}
     message+=" "
     i=$[i+1]
  done

  if [[ $1 != @* ]]; then
  subject="Subject: "
  subject+="${message}"
  echo -e "${subject}" |
  /home/you/msmtp-1.6.3/src/msmtp "${args[0]}"
  fi

fi
```

(caution: the line starting with `echo` has been broken into two lines for space reasons only)

The `args` variable contains the all the arguments passed to the bash file. The first argument is `args[0]` and corresponds to the mail address where the message should be delivered. The rest of the arguments are collected into a single string (called `message`) by the `while` loop. Finally, `subject` joins the word `Subject` to the message.

This bash file is really useful, since we only need to type the address and the message. For example, if the bash file name is `myfile`, then the instruction

```
myfile  mymail@gmail.com My first message
```

will deliver the mail with the message `My first message`.

Enjoy!