GUCCI CBMSOFT/ROOT INSTALL TUTORIAL:

1. Install FairSoft:

```
      sudo apt -y install autoconf automake binutils bison build-essential bzip2 ca-certificates cmake coreutils curl debianutils file findutils flex g++ gcc gfortran git gzip hostname libbz2-dev libcurl4-openssl-dev libgsl-dev libicu-dev libfftw3-dev libprotobuf-dev libgl1-mesa-dev libglu1-mesa-dev libgrpc++-dev libgsl-dev liblzma-dev libncurses-dev libreadline-dev libsqlite3-dev libssl-dev libtbb-dev libtool libx11-dev libxerces-c-dev libxext-dev libxft-dev libxml2-dev libxmu-dev libxpm-dev libyaml-cpp-dev libzstd-dev lsb-release make patch python3-dev protobuf-compiler-grpc rsync sed sqlite3 libsqlite3-dev subversion tar unzip wget xutils-dev xz-utils

      export CMAKE_MODULE_PATH=~/Documents/FairSoft/build/Build/root/ROOTConfig.cmake          [not sure if this was needed]

      export SIMPATH=~/Documents/FairSoft/install/

      cd Documents

      git clone -b jan24_patches https://github.com/FairRootGroup/FairSoft

      cd FairSoft

      mkdir build

      cd build

      cmake -S .. -B . -C ../FairSoftConfig.cmake

      cmake --build . -j16

      make test -j16
```

Note: It tries to do three tests for different fairroot versions: test 2 (fairroot 18.8) and test 3 (fairroot 19.0) fail! only fairroot 18.6 passes all tests.

If I find the time I can check out /home/ubuntu/Documents/FairSoft/build/Testing/Temporary/LastTest.log and use "--rerun-failed --output-on-failure" to re-run the failed cases verbosely.
In the following I will choose the branch v18.6 cuz that matches the tests that have passed.

-----

2. Install FairRoot:

```
      cd ~/Documents/

      git clone -b v18.6_patches https://github.com/FairRootGroup/FairRoot.git

      cd FairRoot && mkdir build && cd build

      cmake -DCMAKE_INSTALL_PREFIX="~/Documents/FairRoot" ..

      make -j16

      sudo make install

      make test -j16
```

If it says "100% tests passed" congrats, you are almost done!

-----

3. Source scripts at startup:

```
      sudo nano ~/.profile
```

Add the following lines at the end if they do not exist already:

```
      export PATH=$PATH:/usr/local/go/bin
      export PATH="$HOME/Documents/FairSoft/install/bin:$PATH"
      source ~/Documents/FairRoot/build/config.sh
      source ~/Documents/FairRoot/bin/FairRootConfig.sh
```

Ensure that the FairRootConfig.sh is executable by running:

```
      sudo chmod +x $HOME/Documents/FairRoot/bin/FairRootConfig.sh
```

Save and reboot. Now typing ‘echo $PATH’ should return a string that contains the substring ‘fair’ or ‘sim’ (gophy queries it to quickly check whether a miner might have the required software).

——-

4. Install CbmRoot:

```
      cd ~/Documents/

      git clone https://git.cbm.gsi.de/computing/cbmroot

      cd cbmroot && mkdir build_cbmroot && cd build_cbmroot

      export SIMPATH=/home/$USER/Documents/FairSoft/install

      export FAIRROOTPATH=/home/$USER/Documents/FairRoot

      cmake ..

      make -j8

      chmod +x config.sh

      ./config.sh

      sudo nano ~/.profile
```

and add the following two lines at the end:

```
      export FAIRROOTPATH=/home/$USER/Documents/FairRoot
      source /home/$USER/Documents/cbmroot/build_cbmroot/config.sh
```

Now source the profile using:

```
      source ~/.profile
```

—

5. Other fixes:

```
      export GEOMPATH=$VMCWORKDIR/common/geometry
      mkdir -p $GEOMPATH
      cp $FAIRROOTPATH/examples/common/geometry/*.geo $GEOMPATH

      mkdir -p $VMCWORKDIR/gconfig
      cp $FAIRROOTPATH/examples/common/gconfig/* $VMCWORKDIR/gconfig

      rm ${SIMPATH}share/root/macros/rootlogon.C

       mkdir -p $VMCWORKDIR/common/gconfig

      cp /home/$USER/Documents/FairSoft/install/share/root/macros/
g4config.in $VMCWORKDIR/common/gconfig
```

Now edit the profile again:

```
      sudo nano ~/.profile
```

And after the ‘export FAIRROOTPATH…’ line add this line:

```
      export GEOMPATH=$VMCWORKDIR/common/geometry
```

Save and now source the profile again using:

```
      source ~/.profile
```

Get the simulation script currently used in gophy:

```
      cd ~/Downloads

      wget https://raw.githubusercontent.com/felix314159/gophy/refs/heads/main/simdata/runSim.C
```

You should now be able to do the following:

```
      root -l -q -b 'runSim.C(123, 14, 100, 0, 2., 0., “abcdef")'
```

This should result in the reproducible (you can hash them and delete them and run it again and hash them again) files ‘`geofile_abcdef.root`’ and ‘`mc_abcdef.root`’

In gophy, for now different parameters are supposed to be passed to such a script to run sim tasks, in the future this can be reworked to give the RA more flexibility in regards how to formulate block problems.