This works outside of slurm

sudo singularity run --bind /contrib/atlantis\_goa/AtlantisGOA\_1342:/app/model,/contrib/Alberto.Rovellini/out1:/app/model/output /contrib/atlantisCode/atlantis6665.sif

Or alternatively:

sudo singularity run --bind /contrib/atlantis\_goa/AtlantisGOA\_1342:/app/model,/contrib/Alberto.Rovellini/out1:/app/model/output /contrib/atlantisCode/atlantis6665.sif /app/model/runGOA\_Test\_v0.sh

Provided first you create the out1 folder, and provided that in the run.sh file that is in the AtlantisGOA\_1342 the argument is -d output (so name consistency). The result of running this will be that outputs will be stored in out1

Also in the Singularity recipe you need the last few lines uncommented.

Now, to run it as one slurm job. Attempting to run the same identical code but as a batch job:

#!/bin/bash

sudo singularity exec --bind /contrib/atlantis\_goa/AtlantisGOA\_1342:/app/model,/contrib/Alberto.Rovellini/out1:/app/model/output /contrib/atlantisCode/atlantis6665.sif /app/model/runGOA\_Test\_v0.sh

I get one line out output from this in slurm.out:

WARNING: could not mount /etc/localtime: not a directory

convertedXMLFileName = GOA\_run.xml

Entering the singularity container shows that the output folder was not created within the container. It shows up in the local copy on the controller (why???) although it’s empty

# 8/17/2023

I was able to run 4 simulations in parallel. That worked with the following slurm batch job order inside the currentVersion folder:

#!/bin/bash

#SBATCH --mail-type=ALL

#SBATCH --mail-user=alberto.rovellini@noaa.gov

#SBATCH --nodes=1

#SBATCH --partition=compute

#SBATCH --array=1-4

sudo mkdir -p /contrib/Alberto.Rovellini/slurm\_array2/out$SLURM\_ARRAY\_TASK\_ID

sudo singularity exec --bind /contrib/atlantis\_goa/currentVersion:/app/model,/contrib/Alberto.Rovellini/slurm\_array2/out$SLURM\_ARRAY\_TASK\_ID:/app/model/output /contrib/atlantisCode/atlantis6665Simple.sif /app/model/runGOA\_Test\_v0\_$SLURM\_ARRAY\_TASK\_ID.sh

[Alberto.Rovellini@czatlantisnwfscnoaagoa-9 currentVersion]$

Things to try now:

1. ~~Rerun this with the other Singularity container (not the “simple” one)~~
2. ~~Rerun this without the hardcoded username~~
3. ~~Rerun this with the batchjob.sh in the atlantisCode folder~~
4. ~~Rerun with 2 runs only~~

All the above worked, so I am unsure what changed since yesterday.

# 8/17/2023

Current working setup:

* Batchjob.sh in /contrib/atlantisCode/ looking like this:

#!/bin/bash

#SBATCH --mail-type=ALL

#SBATCH --mail-user=alberto.rovellini@noaa.gov

#SBATCH --nodes=1

#SBATCH --partition=compute

#SBATCH --array=1-5

sudo mkdir -p /contrib/$USER/slurm\_array/out$SLURM\_ARRAY\_TASK\_ID

sudo singularity exec --bind /contrib/atlantis\_goa/currentVersion:/app/model,/contrib/$USER/slurm\_array/out$SLURM\_ARRAY\_TASK\_ID:/app/model/output /contrib/atlantisCode/atlantis6665.sif /app/model/runGOA\_Test\_v0\_$SLURM\_ARRAY\_TASK\_ID.sh

* Individual runGOA\_Test\_v0\_$SLURM\_ARRAY\_TASK\_ID.sh in /contrib/atlantis\_goa/currentVersion with the following form:

#!/bin/bash

cd /app/model

atlantisMerged -i GOA\_cb\_summer.nc 0 -o outputGOA\_out.nc -r GOA\_runxxx.prm -f GOA\_force.prm -p GOA\_physics.prm -b GOAbioparam\_test.prm -h GOA\_harvest\_background.prm -m GOAMigrations.csv -s GOA\_Groups.csv -q GOA\_fisheries.csv -d output

* All files to run Atlantis in /contrib/atlantis\_goa/currentVersion
* Singularity container file in /contrib/atlantisCode
* Singularity recipe in /contrib/atlantisCode with the form:

Bootstrap: docker

From: ubuntu:18.04

%help

Atlantis v6665 model

%labels

Author alberto.rovellini@gmail.com, modified from Andrew Beet & Hem Nalini Morzaria Luna

%environment

TZ=UTC

DEBIAN\_FRONTEND=noninteractive

export PATH=/usr/lib/rstudio-server/bin:${PATH}

%files

/contrib/atlantisCode/trunk-6665/for\_pw/trunk/atlantis /app/atlantis

/contrib/atlantisCode/trunk-6665/for\_pw/trunk/.svn /app/.svn

#%setup

#install -Dv \

# rstudio\_auth.sh \

# ${SINGULARITY\_ROOTFS}/usr/lib/rstudio-server/bin/rstudio\_auth

#install -Dv \

# ldap\_auth.py \

# ${SINGULARITY\_ROOTFS}/usr/lib/rstudio-server/bin/ldap\_auth

%post

ln -snf /usr/share/zoneinfo/$TZ /etc/localtime && echo $TZ > /etc/timezone

apt-get update && apt-get install -yq build-essential autoconf libnetcdf-dev libxml2-dev libproj-dev subversion valgrind dos2unix nano r-base

cd /app/atlantis

aclocal && autoheader && autoconf && automake -a && ./configure --enable-rassesslink && make CFLAGS="-DACCEPT\_USE\_OF\_DEPRECATED\_PROJ\_API\_H -Wno-misleading-indentation -Wno-format -Wno-implicit-fallthrough" && make install

mkdir /app/model

%runscript

cd /app/model

./runGOA\_Test\_v0.sh

%startscript

cd /app/model

./runGOA\_Test\_v0.sh

* All output gets stored in /contrib/Alberto.Rovellini/slurm\_array/outxxx

# 8/18/2023

Set up a simple R script to analyse output. The R script is in contrib/Alberto.Rovellini and it has this form:

# test script to take Atlantis output and do something with it

library(dplyr)

library(tidyr)

arg <- commandArgs(trailingOnly=T)

runname<- arg[1]

print(runname)

#set path

atl.dir <- file.path('/contrib/Alberto.Rovellini/slurm\_array',runname,'outputGOA\_outBiomIndx.txt', fsep = '/')

# read file

dat <- read.table(atl.dir, header = T)

# do something

dat1 <- dat %>%

select(Time:DR) %>%

filter(Time == max(Time)) %>%

pivot\_longer(-Time, values\_to = 'Biomass', names\_to = 'Species') %>%

group\_by(Species) %>%

summarize(TotBiom = sum(Biomass)) %>%

ungroup() %>%

mutate(run = runname)

# write out

write.csv(dat1, paste0('/contrib/Alberto.Rovellini/terminal\_biomass', runname, '.csv'), row.names = F)

The R script is in turn called by another slurm job in the form of this bash script:

#!/bin/bash

#SBATCH --nodes=1

#SBATCH --partition=process

#SBATCH --array=1-5

Rscript --no-restore --no-save /contrib/$USER/processing\_test.R out$SLURM\_ARRAY\_TASK\_ID