

KISTI: Busan Automated Workflow

Setting Environment

From the experience, it is best to have these options in .bashrc

```
shopt -u progcomp
umask 002
alias qsub="qsub -V"
```

Set

```
export gmsim_root=/home01/hpcl1a02/gmsim
export gmsim=$gmsim_root/Environments/v21p7p20/
source $gmsim/virt_envs/python3_nurion/bin/activate
export PYTHONPATH=$gmsim/workflow
```

Install Cybershake

```
cd /scratch/hpcl1a02/gmsim/RunFolder/korea
```

If existing "korea" directory is messy, let's make a backup and install again.

```
mv korea korea.old
mkdir korea
```

Re-use the Data folder that belongs to the old "korea" directory

```
cd korea
ln -s ../korea.old/Data .
```

You need to create a file "fault_list" that contains fault name and the number of realisations for the fault

```
(python3_nurion) [hpcl1a02@login02 korea]$ cat fault_list
Busan 1r
```

Let's install cybershake. 18.5.3.1 is a Cybershake version that runs with 100m grid spacing. busan_2000_stats.ll is the station list that contains 1050 locations with 2km distance apart.

```
python $gmsim/workflow/scripts/cybershake/install_cybershake.py /scratch/hpcl1a02/gmsim/RunFolder/korea
fault_list 18.5.3.1 --stat_file Data/VMs/Busan/busan_2000_stats.ll
```



warning

Warning: During the install, you are most likely to see this error. This is due to the check against NZ lat / lon while we are doing this for Korea. Fix `install_cybershake_fault.py` and comment out line 157 temporarily.

```
Traceback (most recent call last):
  File "/home01/hpc11a02/gmsim/Environments/adv_im/workflow/scripts/cybershake/install_cybershake.py", line
  160, in <module>
    main()
  File "/home01/hpc11a02/gmsim/Environments/adv_im/workflow/scripts/cybershake/install_cybershake.py", line
  155, in main
    logger=qclogging.get_realisation_logger(logger, fault),
  File "/home01/hpc11a02/gmsim/Environments/adv_im/workflow/scripts/cybershake/install_cybershake_fault.
  py", line 157, in install_fault
    raise RuntimeError(message)
RuntimeError: Error: VM Busan failed VM extents not contained within NZVM DEM
```

Run Cybershake

Create `task_config.yaml` in "korea" directory.

```
(python3_nurion) [hpc11a02@login02 korea]$ cat > task_config.yaml
EMOD3D: ALL
HF: ALL
BB: ALL
IM_calc: ALL
clean_up: ALL
merge_ts: ALL
plot_ts: NONE
IM_plot: NONE
rrup: NONE
LF2BB: NONE
HF2BB: NONE
plot_srf: NONE
```

When it is all good, `run_cybershake.py`. Note that the second argument is the username.

```
python $gmsim/workflow/scripts/cybershake/run_cybershake.py /scratch/hpc11a02/gmsim/RunFolder/korea hpc11a02
/scratch/hpc11a02/gmsim/RunFolder/korea/task_config.yaml
```

LF

When LF is completed, you should see these output files

```
(python3_nurion) [hpc11a02@login02 Busan]$ cd LF/OutBin/Busan
Busan.e3d Busan_seis-00099.e3d Busan_xyts-00005.e3d Busan_xyts-00022.e3d Busan_xyts-00039.e3d
Busan_xyts-00064.e3d Busan_xyts-00081.e3d Busan_xyts-00098.e3d Busan_xyts-00115.e3d
Busan_seis-00067.e3d Busan_seis-00100.e3d Busan_xyts-00006.e3d Busan_xyts-00023.e3d Busan_xyts-00048.e3d
Busan_xyts-00065.e3d Busan_xyts-00082.e3d Busan_xyts-00099.e3d Busan_xyts-00116.e3d
Busan_seis-00068.e3d Busan_seis-00101.e3d Busan_xyts-00007.e3d Busan_xyts-00032.e3d Busan_xyts-00049.e3d
Busan_xyts-00066.e3d Busan_xyts-00083.e3d Busan_xyts-00100.e3d Busan_xyts-00117.e3d
Busan_seis-00069.e3d Busan_seis-00102.e3d Busan_xyts-00016.e3d Busan_xyts-00033.e3d Busan_xyts-00050.e3d
Busan_xyts-00067.e3d Busan_xyts-00084.e3d Busan_xyts-00101.e3d Busan_xyts-00118.e3d
Busan_seis-00070.e3d Busan_xyts-00000.e3d Busan_xyts-00017.e3d Busan_xyts-00034.e3d Busan_xyts-00051.e3d
Busan_xyts-00068.e3d Busan_xyts-00085.e3d Busan_xyts-00102.e3d Busan_xyts-00119.e3d
Busan_seis-00083.e3d Busan_xyts-00001.e3d Busan_xyts-00018.e3d Busan_xyts-00035.e3d Busan_xyts-00052.e3d
Busan_xyts-00069.e3d Busan_xyts-00086.e3d Busan_xyts-00103.e3d
Busan_seis-00084.e3d Busan_xyts-00002.e3d Busan_xyts-00019.e3d Busan_xyts-00036.e3d Busan_xyts-00053.e3d
Busan_xyts-00070.e3d Busan_xyts-00087.e3d Busan_xyts-00112.e3d
Busan_seis-00085.e3d Busan_xyts-00003.e3d Busan_xyts-00020.e3d Busan_xyts-00037.e3d Busan_xyts-00054.e3d
Busan_xyts-00071.e3d Busan_xyts-00096.e3d Busan_xyts-00113.e3d
Busan_seis-00086.e3d Busan_xyts-00004.e3d Busan_xyts-00021.e3d Busan_xyts-00038.e3d Busan_xyts-00055.e3d
Busan_xyts-00080.e3d Busan_xyts-00097.e3d Busan_xyts-00114.e3d
```

merge_ts

The xyts files needs to be merged into one to be able to produce an animation. The merge is done by "merge_ts". Note that this step is only needed if you want to produce an animation



warning

If PBS script for merge_ts is nowhere to be found, use below

```
#!/bin/bash
# script version: pbs
# emod3d slurm script

# Please modify this file as needed, this is just a sample
#PBS -N run_postemod3d.Busan
#PBS -V
#PBS -q normal
#PBS -A inhouse
#PBS -l select=1:ncpus=4:mpiprocs=4:ompthreads=1
#PBS -l walltime=00:05:00
#PBS -W sandbox=PRIVATE

module purge
module load craype-network-opa gcc craype-mic-knl mvapich2 cmake
export gmsim_root=/home01/hpc11a04/gmsim
export gmsim=$gmsim_root/Environments/adv_im
export PYTHONPATH=$gmsim/workflow
source $gmsim/virt_envs/python3_nurion/bin/activate

export SLURM_JOB_ID="{PBS_JOBID/.pbs/}"
ls /scratch/hpc11a02/gmsim/RunFolder/korea/Runs/Busan/Busan/LF/OutBin/*xyts*.e3d > ./list
mpirun $HOME//gmsim/tools/merge_tsP3_par filelist=./list outfile=/scratch/hpc11a02/gmsim/RunFolder/korea/Runs/Busan/Busan/LF/OutBin/Busan.e3d
```

This job should be pretty quick.

GMSimViz

Let's download Busan.e3d and Busan.srf. We will be running GMSimViz locally. I have not tested this on Nurion.

Follow the installation instruction at [GMSimViz/README.md at master · ucgmsim/GMSimViz \(github.com\)](#) and install GMSimViz and GMT

Suppose ~/busan has Busan.e3d and Busan.srf, and GMSimViz is at ~/GMSimViz

```
cd ~/GMSimViz
mpirun -n 4 bin/gmsimviz ~/busan/Busan.srf -a --crude --title "Busan Mw.5.0" --dpi 120 --downscale 1 -x ~/busan/Busan.e3d
```

WORK IN PROGRESS: On Nurion, edit SRF, E3D and TITLE below and submit the job. (Not working yet)

```
#!/bin/bash
# script version: pbs
# emod3d slurm script

# Please modify this file as needed, this is just a sample
#PBS -N gmsimviz
#PBS -V
#PBS -q normal
#PBS -A inhouse
#PBS -l select=1:ncpus=64:mpiprocs=64:ompthreads=1
#PBS -l walltime=00:05:00
#PBS -W sandbox=PRIVATE

module purge
module load craype-network-opa gcc craype-mic-knl mvapich2 cmake
export gmsim_root=/home01/hpc11a02/gmsim
export gmsim=$gmsim_root/Environments/adv_im
source $gmsim/virt_envs/python3_nurion/bin/activate
export PYTHONPATH=$gmsim/workflow

export PYTHONPATH=$PYTHONPATH:$gmsim/GMSimViz
export PATH=$PATH:$gmsim/GMSimViz/bin
export SLURM_JOB_ID="{PBS_JOBID/.pbs/}"

SRF=/scratch/hpc11a04/gmsim/RunFolder/korea/Data/Sources/Busan/Srf/Busan.srf
E3D=/scratch/hpc11a04/gmsim/RunFolder/korea/Runs/Busan/Busan/LF/OutBin/Busan.e3d
TITLE="Busan Mw.5.0 Srf Type 2"

mpirun gmsimviz $SRF -a --crude --title $TITLE --dpi 120 --downscale 1 -x $E3D
```