

# HPC Comparison chart

## NeSI Maui/Mahuika

	Maui					Mahuika																																																																					
Model	Cray XC50					Cray CS400																																																																					
Number of CPUs	18,650x2.4Ghz Skylake (1node = 80 virtual cores)					8,424 x 2.1GHz Broadwell (1node = 72 virtual cores)																																																																					
Total Memory	66.8Tb					30 Tb																																																																					
Scheduler	SLURM					SLURM																																																																					
Max num of submission per user	<table><thead><tr><th>Queue</th><th>Wall-clock limit</th><th>Nodes</th><th>CPU /Node</th><th>Max Mem /Node</th></tr></thead><tbody><tr><td>nesi_rese arch</td><td>24 h</td><td>264</td><td>40 (80)</td><td>80 or 160Gb</td></tr></tbody></table>					Queue	Wall-clock limit	Nodes	CPU /Node	Max Mem /Node	nesi_rese arch	24 h	264	40 (80)	80 or 160Gb	<table><thead><tr><th>Queue</th><th>Wall-clock limit</th><th>Nodes</th><th>CPU /Node</th><th>Max Mem /CPU</th><th>Max Mem /Node</th></tr></thead><tbody><tr><td>large</td><td>3days</td><td>226</td><td>72</td><td>1500Mb</td><td>108Gb</td></tr><tr><td>long</td><td>3weeks</td><td>69</td><td>72</td><td>1500Mb</td><td>108Gb</td></tr><tr><td>prepost</td><td>3h</td><td>5</td><td>72</td><td>6800Mb</td><td>480Gb</td></tr><tr><td>bigmem</td><td>7days</td><td>4</td><td>72</td><td>6800Mb</td><td>480Gb</td></tr><tr><td>hugemem</td><td>7days</td><td>0.5</td><td>128</td><td>30Gb</td><td>4000Gb</td></tr><tr><td>gpu</td><td>3days</td><td>4</td><td>8</td><td>13500Mb</td><td>108Gb</td></tr><tr><td>ga_bigmem</td><td>7days</td><td>1</td><td>72</td><td>6800Mb</td><td>480Gb</td></tr><tr><td>ga_huge mem</td><td>7days</td><td>1</td><td>128</td><td>30Gb</td><td>4000Gb</td></tr></tbody></table>						Queue	Wall-clock limit	Nodes	CPU /Node	Max Mem /CPU	Max Mem /Node	large	3days	226	72	1500Mb	108Gb	long	3weeks	69	72	1500Mb	108Gb	prepost	3h	5	72	6800Mb	480Gb	bigmem	7days	4	72	6800Mb	480Gb	hugemem	7days	0.5	128	30Gb	4000Gb	gpu	3days	4	8	13500Mb	108Gb	ga_bigmem	7days	1	72	6800Mb	480Gb	ga_huge mem	7days	1	128	30Gb	4000Gb
	Queue	Wall-clock limit	Nodes	CPU /Node	Max Mem /Node																																																																						
	nesi_rese arch	24 h	264	40 (80)	80 or 160Gb																																																																						
	Queue	Wall-clock limit	Nodes	CPU /Node	Max Mem /CPU	Max Mem /Node																																																																					
	large	3days	226	72	1500Mb	108Gb																																																																					
	long	3weeks	69	72	1500Mb	108Gb																																																																					
	prepost	3h	5	72	6800Mb	480Gb																																																																					
	bigmem	7days	4	72	6800Mb	480Gb																																																																					
	hugemem	7days	0.5	128	30Gb	4000Gb																																																																					
	gpu	3days	4	8	13500Mb	108Gb																																																																					
ga_bigmem	7days	1	72	6800Mb	480Gb																																																																						
ga_huge mem	7days	1	128	30Gb	4000Gb																																																																						
Max CPU request: 240 nodes = 9,600 phy.= 19,200 virt. cores																																																																											
Max Node Hours : 1200 node-hours																																																																											
eg.) requesting 240 nodes means wall clock limited to 5 hours.																																																																											
Max num of jobs (submit): 1000					Max CPU request: 576 CPUs (8 full nodes)																																																																						
					Max num of jobs (submit): 1000																																																																						
					Max Core hours per job: 20,000 hrs.																																																																						
Dev env.																																																																											
File system																																																																											
Gotchas																																																																											
Useful command	Fairshare score: nn_corehour_usage nesi00213 eg. 0.336420 out of 1.0																																																																										

## TACC Stampede2

	Stampede2 (TACC)
Model	Dell PowerEdge C6320P/C6420
Number of CPUs	367,024  Xeon Phi 7250 68C 1.4GHz
Total Memory	736Tb
Scheduler	SLURM

Max num of submission per user	KNL: 1 node 68 cores (1 socket) = 272 hyper threads BUT 64-68MPI tasks advisable * 4200 KNL nodes (96Gb+16Gb)/node SKX: 1 nodes 48 cores (= 2 sockets* 24 cores/socket) = 96 hyper threads * 1,736 nodes																																														
	<table> <tr> <th>Queue</th><th>Wall-clock limit</th><th>Max Nodes/Job</th><th>Max active jobs (running+waiting)</th></tr> <tr> <td colspan="4"><b>KNL</b></td></tr> <tr> <td>development</td><td>2h</td><td>16 (1,088 cores)</td><td>1</td></tr> <tr> <td>normal</td><td>48h</td><td>256 (17,408 cores)</td><td>50</td></tr> <tr> <td>large</td><td>48h</td><td>2048 (139,264 cores)</td><td>5</td></tr> <tr> <td>long</td><td>120h</td><td>32 (2,176 cores)</td><td>2</td></tr> <tr> <td>flat-quadrant</td><td>48h</td><td>32 (2,175 cores)</td><td>5</td></tr> <tr> <td colspan="4"><b>SKX</b></td></tr> <tr> <td>skx-dev</td><td>2h</td><td>4 (192 cores)</td><td>1</td></tr> <tr> <td>skx-normal</td><td>48h</td><td>128 (6,144 cores)</td><td>25</td></tr> <tr> <td>six-large</td><td>48h</td><td>868 (41,664 cores)</td><td>3</td></tr> </table>	Queue	Wall-clock limit	Max Nodes/Job	Max active jobs (running+waiting)	<b>KNL</b>				development	2h	16 (1,088 cores)	1	normal	48h	256 (17,408 cores)	50	large	48h	2048 (139,264 cores)	5	long	120h	32 (2,176 cores)	2	flat-quadrant	48h	32 (2,175 cores)	5	<b>SKX</b>				skx-dev	2h	4 (192 cores)	1	skx-normal	48h	128 (6,144 cores)	25	six-large	48h	868 (41,664 cores)	3		
Queue	Wall-clock limit	Max Nodes/Job	Max active jobs (running+waiting)																																												
<b>KNL</b>																																															
development	2h	16 (1,088 cores)	1																																												
normal	48h	256 (17,408 cores)	50																																												
large	48h	2048 (139,264 cores)	5																																												
long	120h	32 (2,176 cores)	2																																												
flat-quadrant	48h	32 (2,175 cores)	5																																												
<b>SKX</b>																																															
skx-dev	2h	4 (192 cores)	1																																												
skx-normal	48h	128 (6,144 cores)	25																																												
six-large	48h	868 (41,664 cores)	3																																												
	SKX is slightly more expensive than KNL																																														
Dev env.	Default compiler: Intel 18.																																														
File system	\$HOME: 10Gb (200,000 files) \$WORK: 1Tb (3mil files) : not for high IO, large files. nobackup, no purge \$SCRATCH: unlimited. nobackup, deleted if not accessed for 10 day. /nesi/project/nesi00213 == \$HOME/project /nesi/nobackup/nesi00213 == \$HOME/nobackup or \$SCRATCH/nobackp																																														

Gotchas	<h2>Building</h2> <p><b>Intel</b></p> <pre>module add fftw3/3.3.8 intel/18.0.2 impi/18.0.2 cmake/3.10.2</pre> <pre>MPI_C_LIB_NAMES = mpifort;mpi;mpigi;dl;rt;pthread</pre> <pre>MPI_dl_LIBRARY = /usr/lib64/libdl.so</pre> <pre>MPI_pthread_LIBRARY = /usr/lib64/libthread.so</pre> <pre>MPI_rt_LIBRARY = /usr/lib64/librt.so</pre> <p>By default gcc-6.5 creeps in and it attempts to build with gcc-6.5 instead of icc. Enforce it with CC=icc.</p> <p>I found "make VERBOSE=1" extremely useful to debug building issues</p> <h2>GCC</h2> <ol style="list-style-type: none"><li>1. Load the correct modules:<pre>module add git/2.24.1 cmake/3.16.1 TACC impi/17.0.3 libfabric/1.7.0 autotools/1.1 xalt/2.8 gcc/7.1.0 python3/3.6.1 hdf5/1.10.4</pre></li><li>2. Build FFTW (3.3.8)<pre>./configure --enable-float --enable-sse --enable-threads --host=x86_64-pc-linux --enable-shared --prefix=\$SOMEWHERE make all install</pre><ol style="list-style-type: none"><li>a. Now that initial set up has been completed the following commands can be used for the GCC workflow:<ol style="list-style-type: none"><li>i. activate_env /work/06833/sungbae/stampede2/Environments/stampede_gcc</li><li>ii. module restore gcc_modules</li><li>iii. export LIBRARY_PATH=\$LIBRARY_PATH:~</li></ol></li></ol></li><li>3. Build EMOD3D<pre>mkdir build cd build FTW_DIR=\$SOMEWHERE cmake ../ make</pre></li></ol> <h2>Issue</h2> <p>emod3d has a rounding error issue with icc and returns wrong "ny" failing post-emod3d test. Rob Graves fixed this by converting float to double in the function get_n1n2() in misc.c. The fix is included in 3.0.6 (On Nurion, however, this fix was found to be not enough)</p> <h2>Running</h2> <p>Project name must be CamelCase: DesignSafe-Graves</p> <p>Slurm script needs -N for number of nodes</p> <pre>#SBATCH -N 4 #SBATCH --ntasks=160</pre> <p>Instead of "srun" it uses "ibrun"</p> <h2>Workflow</h2> <p>A number of hardcoded bits assuming NeSI machine need to be updated. Check workflow and qcore "stampede" branches.</p> <p><a href="https://github.com/ucgmsim/slurm_gm_workflow/tree/stampede">https://github.com/ucgmsim/slurm_gm_workflow/tree/stampede</a></p> <p><a href="https://github.com/ucgmsim/qcore/tree/stampede">https://github.com/ucgmsim/qcore/tree/stampede</a></p>
Usage check	<pre>(python3_stampede) sungbae@stampede21(1):~\$ /usr/local/etc/taccinfo ----- Project balances for user sungbae -----   Name Avail SUs Expires       DesignSafe-Graves 19974 2020-09-30     ----- ----- Disk quotas for user sungbae -----   Disk Usage (GB) Limit %Used File Usage Limit %Used     /home1 0.8 10.0 7.82 1853 200000 0.93     /work 10.0 1024.0 0.97 55539 3000000 1.85     /scratch 11.0 0.0 0.00 4032 0 0.00   -----</pre> <p>Available 19974 SUs out of 20000.</p>

KISTI Nurion

	Nurion (KISTI)						
Model	Cray CS500						
Number of CPUs	570,020						
	Xeon Phi 7250 68C 1.4Ghz						
Total Memory							
Scheduler	PBS						
Max num of submission per user	KNL: 1 node 68 cores (1 socket) * 8305 nodes (96Gb+16Gb)/node						
	SKL: 1 node 40 cores (2 sockets * 20 cores/socket) * 132 nodes (192Gb/node)						
	Queue	Wall-clock limit	Max Nodes/Job	Max running jobs	Max active jobs (running+waiting)		
	KNL						
	exclusive	unlimited	<del>2600 (176,800 cores)</del>	<del>400</del>	<del>200</del>		
	normal (82Gb)	48h	4970 (337,960 cores)	550	600		
	long (82Gb)	120h	300	25	30		
	flat (102Gb)	48h	180	35	40		
	debug (82Gb)	48h	2 (20 avail)	2	2		
	SKL						
Dev env.	commercial	48h	118 (4720cores)	2	6		
	norm_skl	48h	118(4720cores)	15	20		
File system		Path	Size limit	inode limit	Cleanup policy	FS	Backup
	home	/home01	64GB	200K	N/A	Lustre	O
	scratch	/scratch	100TB	1M	Auto delete if not accessed over 15 days		X
	Gotchas	<p>Building EMOD3D was somewhat tricky. I ended up having my own version of CMake 3.9 (existing module has no cmake, and later versions of CMake are buggy), and fftw3 (existing module didn't have fftw3f, and CMake failed to pick up.</p> <p>Originally build with Intel tool chain, but EMOD3D had rounding error issues, and it generates incompatible random numbers (different from Maui). For best (and consistent) result, using GNU tool chain is highly recommended.</p> <p>The following modules are used.</p> <p>craype-network-opa</p> <p>gcc</p> <p>craype-mic-knl</p> <p>mvapich2</p> <p>mvapich2 is required as mpi4py doesn't seem to work properly with openmpi</p> <p>Don't bother with fftw3 module. We need to build fftw3 from scratch: only fftw3f (single) version is needed.</p> <p><b>FFTW3</b></p> <p>export MPICC='mpicc -fPIC -march=knl'</p> <p>export CC='gcc -fPIC -march=knl'</p> <p>./configure --enable-float --enable-sse --enable-threads --host=x86_64-pc-linux --enable-shared --prefix=/home01/hpc11a02/gmsim/Environments/nurion/ROOT/local/gnu</p> <p>make all install</p> <p><b>EMOD3D</b></p>					

	<pre>mkdir build  cd build  export FFTW_DIR=/home01/hpc11a02/gmsim/Environments/nurion/ROOT/local/gnu  cmake ..  cmake --build . --target all -j 8</pre> <p><b>GMT</b></p> <p>Prerequisite</p> <ul style="list-style-type: none"><li>• curl</li><li>• sqlite-snapshot-202004061816,</li><li>• zlib-1.2.11,</li><li>• libpng-1.6.37,</li><li>• tiff-4.1.0,</li><li>• GraphicsMagick-1.3.35,</li><li>• proj-7.0</li><li>• gdal-3.0.1</li></ul> <p>Except for GDAL, this works:</p> <p><code>\$HOME=/home01/x2319a02</code></p> <pre>\$ PKG_CONFIG_PATH=\$HOME/gmsim/Environments/nurion/ROOT/local/gnu/lib/pkgconfig \$ ./configure --prefix=\$HOME/gmsim/Environments/nurion/ROOT/local/gnu &amp; make all install</pre> <p>For GDAL,</p> <pre>module add netcdf  CPPFLAGS=-I\$HOME/gmsim/Environments/nurion/ROOT/local/gnu/include PKG_CONFIG_PATH=\$HOME/gmsim/Environments/nurion/ROOT/local/gnu/lib /pkgconfig ./configure --prefix=\$HOME/gmsim/Environments/nurion/ROOT/local/gnu --with-proj=\$HOME/gmsim/Environments/nurion/ROOT/local/gnu &amp; make all install</pre> <p>(Edit: I had to manually add CPPFLAGS into config.status (2022/11/25))</p> <p>For GMT,</p> <p>go to build</p> <pre>cmake -DDCW_PATH:PATH=\$HOME/gmsim/Environments/nurion/ROOT/share/dcw-gmt-1.1.4 -DGSHHG_PATH:PATH=\$HOME/gmsim/Environments/nurion/ROOT/share /gshhg-gmt-2.3.7 ../  make all install</pre> <p><b>!WARNING!</b></p> <p>"qsub" MUST be executed in \$SCRATCH directory.</p>
Usage check	<pre>isam  \$ lfs quota -h /home01  \$ lfs quota -h /scratch</pre> <p>1 gujwa = KNL 6,400 node time (100 SRU time) = 435,000 core hours</p> <p>XXX sec * 4350/3600 = core hours</p>

For details of PBS, see [PBS](#) page.