KNL Performance Comparison: *Linear GS2*

March 2017

1. Compilation, Setup and Input

Compilation

The actual code that was used in the testing is derived from the publically available GS2 code (svn checkout svn://svn.code.sf.net/p/gyrokinetics/code/ gyrokinetics-code/gs2/trunk). The main algorithm is unchanged and the build system is effectively identical to that of GS2. The main difference in the source code is simply the removal of code that's unused in linear simulations.

GS2 provides a makefile designed for Archer, to build on the front-end the following commands are used;

make WITH_EIG= USE_NEW_DIAG= depend
make WITH_EIG= USE_NEW_DIAG= gs2

When building for the KNL we add "CHIP=knl" to the make command.

The modules loaded on the front end are:

1) modules/3.2.10.2	9) packages-archer	17) cray-libsci/13.2.0	25) alps/5.2.3-2.0502.9295.14.14.ari	33) scalasca/2.2
2) eswrap/1.3.3-1.020200.1278.0	10) bolt/0.6	18) udreg/2.3.2-1.0502.9889.2.20.ari	26) rca/1.0.0-2.0502.57212.2.56.ari	34) papi/5.4.1.2
3) switch/1.0-1.0502.57058.1.58.ari	11) nano/2.2.6	19) ugni/6.0-1.0502.10245.9.9.ari	27) atp/1.8.3	35) ddt/4.0.1.0_32296
4) craype-network-aries	12) leave_time/1.0.0	20) pmi/5.0.7-1.0000.10678.155.25.ari	28) PrgEnv-intel/5.2.56	36) intel/15.0.2.164
5) craype/2.4.2	13) quickstart/1.0	21) dmapp/7.0.1-1.0502.10246.8.47.ari	29) cray-netcdf/4.3.3.1	37) cray-petsc-complex/3.5.2.1
6) pbs/12.2.401.141761	14) ack/2.14	22) gni-headers/4.0-1.0502.10317.9.2.ari	30) fftw/3.3.4.5	
7) craype-ivybridge	15) xalt/0.6.0	23) xpmem/0.1-2.0502.57015.1.15.ari	31) cube/4.3	
8) cray-mpich/7.2.6	16) epcc-tools/6.0	24) dvs/2.5_0.9.0-1.0502.1958.2.55.ari	32) scorep/1.4	

The modules loaded on the KNL system are:

1) modules/3.2.10.5	7) gni-headers/5.0.7-3.1	13) sysadm/2.2.2-3.39	19) cray-mpich/7.4.4	25) cray-netcdf/4.4.1
2) alps/6.1.6-20.1	8) dmapp/7.1.0-12.37	14) lustre-utils/2.3.4-6.74	20) pbs/default	26) fftw/3.3.4.10
3) nodestat/2.2-2.40	9) xpmem/0.1-4.5	15) Base-opts/2.1.3-2.16	21) cray-libsci/16.09.1	27) papi/5.4.3.3
4) sdb/2.2.1-3.119	10) llm/20.2.4-3.18	16) craype-mic-knl	22) pmi/5.0.10-1.0000.11050.0.0.ari	28) intel/17.0.0.098
5) udreg/2.3.2-4.6	11) nodehealth/5.2.0-5.46	17) craype-network-aries	23) atp/2.0.3	
6) ugni/6.0.12-2.1	12) system-config/2.2.18-3.38	18) craype/2.5.7	24) PrgEnv-intel/6.0.3	

Setup

Used the quad_100 memory option for all KNL runs.

Restricted study to a single node in each case with 1 thread and 1 hyperthread per MPI process. The number of MPI processes was varied from 1 to the maximum provided by a single node.

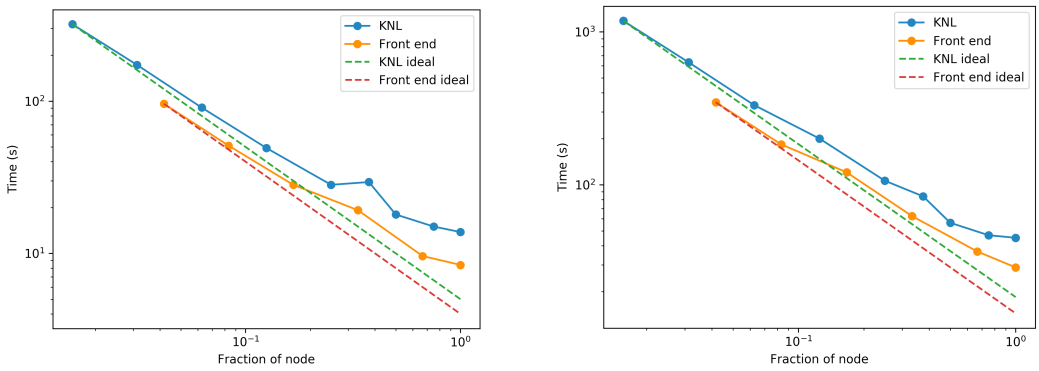
Input

The input file that was used is provided at the end of this document. This provides the input for the "collisional" runs, the "collisionless" runs set `collision_model='none' ` but are otherwise identical. This is a fairly representative "full physics" linear simulation, although larger grids are not uncommon.

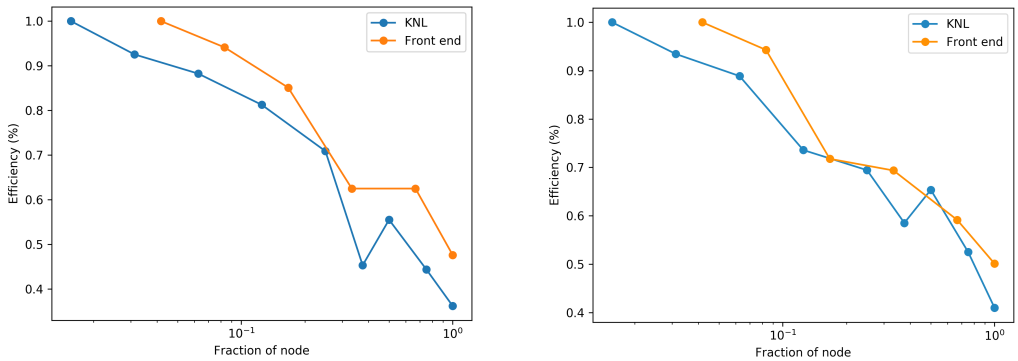
Note a few of the input variables are in different namelists for the reduced code used here as compared to GS2. It should be straightforward to transform back to a GS2 input file by referring to http://gyrokinetics.sourceforge.net/wiki/index.php/GS2_Input_Parameters.

2. Performance Data

The figures below compare the time to solution (initialisation+advance) on the KNL and front end Xeon as the number of MPI processes is varied within a single node. Rather than plotting against the number of processes directly, the x-axis is instead normalised by the number of processes available in a node (rough calculations suggest one KNL node requires roughly the same power as one Xeon node). The left hand plot is for the collisionless case whilst the right plot shows the collisional case. Also shown is the ideal scaling for each case. On the KNL the number of MPI processes used are {1,2,4,8,16,24,32,48,64} and on the front end the number of MPI processes are {1,2,4,8,16,24}.



The above plots indicate that for this linear problem the KNL runtime never falls below the front end time, for a given fraction of a node. The efficiency (ideal/actual) is shown in the plots below (left is collisionless, right is collisional). The efficiency for the KNL and Front end runs follow each other fairly closely.



The raw data for the first plots is provided in appendix 2.

3. Summary and Conclusions

Currently in these preliminary investigations it appears that these linear simulations do not benefit from the use of KNL. This is perhaps to be expected given that the code is currently pure MPI and does not take advantage of shared memory. A number of the more costly operations (a global data transpose, a matrix inversion and a matrix-vector multiplication) within the code are expected to benefit from the use of shared memory and future development to explore this will potentially offer more favourable KNL/Xeon comparisons.

Appendix 1: input file

```
&kt_grids_knobs
 grid_option = "single"
/

&kt_grids_single_parameters
 n0 = 50
 rhostar_single =  0.001084
/

&theta_grid_parameters
 ntheta = 32
 nperiod =3
 eps = 0.18 !0.18
 epsl = 2.0
 shat = 0.776
 pk = 1.4285
 shift =  0.0
 akappa = 1.0
 akappri = 0.0
 tri = 0.0
 tripri = 0.0
 qinp = 1.44
/

&theta_grid_knobs
 equilibrium_option = "eik"
/

&theta_grid_salpha_knobs
 model_option = "default"
/

&theta_grid_eik_knobs
 local_eq = .true.
 efit_eq = .false.
 gen_eq = .false.
 ppl_eq = .false.
 transp_eq = .false.
 bishop = 2
 beta_prime_input = 0.0
 s_hat_input = 0.776
 iflux =0
 irho = 2
 writelots = .true.
/

&le_grids_knobs
 ngauss = 8
 negrid = 8
/

&dist_fn_knobs
```

```
 fexp =  0.48
 bkdiff = 0.05
/

&knobs
 fphi = 1.0
 fapar = 1.0
 fbpar = 1.0
 delt = 0.075
 nstep = 10000
 beta = 0.001
 tite = 1.0
 zeff = 1.0
/

&layouts_knobs
!The following should be removed when working with GS2
 imbalance_l = .true.
 imbalance_e = .true.
 imbalance_s = .true.
 imbalance_g = .true.
/

&collisions_knobs
 collision_model = "default"!"none"
/

&species_knobs
 nspec =  2
/

&species_parameters_1
 z = 1.0
 mass = 1.0
 dens = 1.0
 temp = 1.0
 tprim = 6.92
 fprim = 2.22
 uprim = 0.0
 vnewk = 0.03
 type = 'ion'
 bess_fac = 1.0
/

&species_parameters_2
 z = -1.0
 mass = 2.74e-4
 dens = 1.0
 temp = 1.0
 tprim = 6.92
 fprim = 2.22
 uprim = 0.0
 vnewk = 1.97
 type = 'electron'
 bess_fac = 1.0
/

&init_g_knobs
 ginit_option = "default"
```

```
 chop_side = .false.
 phiinit = 1.0e-5
/

&gs2_diagnostics_knobs
 write_ascii = .false.
 write_moments = .false. !.true.
 ob_midplane = .false.
 write_final_moments = .true.
 save_for_restart = .false.
 nsave = 1000
 nwrite = 200
 navg = 100 !50
 omegatol = -1.0e-3
 omegatinst = 500.0
/
```

Appendix 2: Raw data

KNL timing data

Number of mpi processes	Time (mins) – Collisional case	Time (mins) – Collisionless case
1	19.66	5.33
2	10.52	2.88
4	5.53	1.51
8	3.34	0.82
16	1.77	0.47
24	1.40	0.49
32	0.94	0.30
48	0.78	0.25
64	0.75	0.23

Front end timing data

Number of mpi processes	Time (mins) – Collisional case	Time (mins) – Collisionless case
1	5.77	1.60
2	3.06	0.85
4	2.01	0.47
8	1.04	0.32
16	0.61	0.16
24	0.48	0.14