Installing the Columbia Physics System

A.N. Jackson & S. Booth

Revision: 1.6 Date: 2002/03/11 19:34:56

A.N.Jackson@ed.ac.uk.

Contents

1 Downloading the CPS source code

2 Compiling the code

2.1 INSTALLATION ON QCDSP

The original make structure has been retained, and the configuration files are set up to default to the target being QCDSP. Therefore, the code should compile straight away by simply running make in the phys/ directory.

2.2 INSTALLATION ELSEWHERE

You will require gmake (GNU make) to compile this program, and to run it in parallel you will require some implementation of MPI.

The code configuration is done in config.h. The compilation configuration is done in Makefile.gnu and Makefile.gnutests. The regression-testing configuration is done in tests/regression.pl.

2.2.1 The configure script

The code uses a fairly standard configure script for cross-platform configuration of the makefiles etcetera. The script should be run from the phys/ directory as "./configure". There are a large number of general configuration flags (use "./configure –help" to see them all) but only a few of these influence the behaviour of the physics code itself:

- `--enable-parallel-mpi` — Compile as a parallel code, using MPI. (default=no)
- `--enable-double-prec` — Force the code to use double precision (instead of float) throughout (default=yes)

If all goes well, you may build via the GNU make files using:

- `gmake -f Makefile.gnu [all]` — Makes everything.
- `gmake -f Makefile.gnu clean` — Cleans everything.
- `gmake -f Makefile.gnu testprogs` — Makes only the test suite.
- `gmake -f Makefile.gnu cleantests` — Cleans only the test suite.

2.2.2 Manual configuration

The current configure script handles ome things (e.g. the MPI include/linkage) in a platform (Solaris) specific manner. Therefore compilation for anything but serial gcc may require some minor tuning. The files that you may have to change are as follows:

- `config.h`

  This file contains the following flags:

  - `PARALLEL` If defined, compile for a parallel (MPI) environment.
  - `CAST_AWAY_CONST` A macro describing how the current compiler casts away the const-ness of strings (inline const char*). This is not strictly necessary and you should not have to change it.
  - `GLOBALSUM_TYPE` Precision/type for global summations. On QCDSP this was a custon double64 type, but on other platforms double should be used.
  - `LOCALCALC_TYPE` Precision/type for Floats. On QCDSP ths was a custom float type (rfloat), but a native type should be used elsewhere.
  - `INTERNAL_LOCALCALC_TYPE` Precision/type for IFloat. Should be the same as LOCALCALC_TYPE anywhere but on QCDSP, where float should be used (IFloats used to be floats in the original code).
  - `COMMS_DATASIZE` Default size of the individual items of data to be transmitted via the MPISCU. Should be consistent with the size of Float and IFloat.
  - `VERBOSE` If defined, the MPISCU will print information concerning each of the SCU calls.

  The #include of qcdio.h means that printf is overridden with qprintf, which is an implementation of printf designed to print only the information from the zeroth node. This makes the code output consistent with the default code output on QCDSP.

- `Makefile.gnu & Makefile.gnutests`
  The elements you may have to change lie between the ***COMPILATION FLAGS*** markers.

  In Makefile.gnu, this concerns the compiler name and flags (the `-I(top_dir)/nga/mpi_scu` tells the code where the MPISCU include files are).

  In Makefile.gnutests, you must choose which set of library files to compile against (depending on whether the code is serial or MPI-parallel) and define the compilation and linking flags.

- Testing Script `tests/regression.pl`
  At the start of tests/regression.pl three different examples of testing configuration are given. The first set runs the parallel QCDSP version, the second set runs the serial QCDSP version and the third set runs the non-QCDSP version.

3 Platforms

The code has been used on the following platforms.

3.1 Columbia: QCDSP

The serial and parallel versions compile straight out of the box using the ”gmake -f Makefile.gnu” and ”make” commands respectively.

3.2 EPCC: Cray T3E

For compilation the following set of commands seems to work:

```
bash$ export PATH=/opt/open/bin:$PATH
bash$ autoconf
bash$ ./configure --enable-parallel-mpi CC=CC CFLAGS="-O3 -h conform"
bash$ make -f Makefile.gnu all
```

I don’t know why you have to run autoconf on the T3E. It must be some localised abberation because it is not necessary on other platforms.

3.3 UKQCD: Ukqcd2; an Alpha

3.4 EPCC: Bronzite; a Sun workstation

3.5 EPCC: Lomond; a Sunfire 6800 SMP

3.6 EPCC: Bobcat; a Beowulf cluster