|  |
| --- |
|  |
| Building APSIM 7.9 for the USQ HPC System |
| Russell Waldron |

|  |
| --- |
|  |

Introduction

This document describes of how to compile APSIM 7.9 from the source code for a Linux High Performance Computing (HPC) environment.

The document was prepared by the University of Southern Queensland High Performance Computing team. It is based on the process developed to compile APSIM 7.7, 7.8 and 7.9 for the USQ HPC system *Fawkes*.

This document is intended for experienced programmers familiar with compiling and installing software on Linux from source.

Build Environment

The process of compilation and installation of software in a HPC environment differs from that used for a desktop Linux system. This comes about largely due to the need to provide multiple versions of the same software and to make the software available across a cluster of compute nodes.

To support multiple versions of a software package it must be built from source and installed into a directory that can be shared via a protocol such as NFS.

APSIM 7.9 was built at USQ on a Hewlett Packard Enterprise (HPE) HPC cluster running Red Hat Enterprise Linux version 6.6.

The build process described has no Red Hat or HPE environment specific features so it will be compatible with other Linux HPC environments.

Software Requirements

The USQ HPC software installation directory is /usr/local/opt/software. Other than the standard operating system packages all key APSIM dependencies were built from source.

The following packages are required to build APSIM 7.9:

**GCC-4.8.5**

This version of GCC was found to be required otherwise the APSIM build failed. The Fortran compiler associated with this version is also required. Later versions of GCC are expected to work.

**tcltk-8.6.4**

The most recent versions of tcltk should work.

**r-3.0.2**

The most recent versions of R should work.

**mono-4.8.1.1.0**

The most recent versions of Mono should work. This version of Mono was built exclusively to support APSIM. It includes the following packages:

* tiff-4.0.8
* giflib-5.1.4
* libgdiplus - latest from repository
* vbnc - Visual Basic Compiler
* libxml2-2.9.4

**p7zip\_16.0**

The most recent version of 7z should work.

Build Steps

1. Select a directory into which the APSIM 7.9 source code will be extracted and compiled. Ensure at least 5 GB of storage is available.

% cd /usr/local/opt/software/apsim

% mkdir apsim-7.9

% cd apsim-7.9

% bzip2 -cd PATH\_TO/apsim-7.9-4045.tar.bz2 | tar xvf –

1. Move APSIM files/directories to current directory.

% mv apsim/\* .

% mv apsim/.svn .

% rmdir apsim

1. Edit R configuration files.

% cd Model/RLink

% cp Makefile.linux Makefile.linux.bak

Change the following lines in Makefile.linux

R\_HOME = /usr

becomes

R\_HOME = /usr/local/opt/software/r/r-3.0.2-gnu

Change to add the fully qualified path of the mono executable to run ProcessDataTypesInterface.exe.

$(APSIM)/Model/ProcessDataTypesInterface.exe \ ../DataTypes/Datatypes.xml RDataTypes.macro

becomes

/usr/local/opt/software/mono/mono-4.8.1.0-gnu/bin/mono \ $(APSIM)/Model/ProcessDataTypesInterface.exe \ ../DataTypes/Datatypes.xml RDataTypes.macro

1. Edit TCL configuration files.

% cd ../TclLink

% cp Makefile Makefile.bak

Change the following lines in Makefile.

DOTVER=8.5

becomes

DOTVER=8.6

INCLUDES = -I /usr/include/tcl$(DOTVER)

becomes

INCLUDES = -I /usr/local/opt/software/tcltk/tcltk-8.6.4-gnu/include

Change to add the fully qualified path of the mono executable to run ProcessDataTypesInterface.exe.

$(APSIM)/Model/ProcessDataTypesInterface.exe \ ../DataTypes/Datatypes.xml TclComponent.macro

becomes

/usr/local/opt/software/mono/mono-4.8.1.0-gnu/bin/mono \

$(APSIM)/Model/ProcessDataTypesInterface.exe \

../DataTypes/Datatypes.xml TclComponent.macro

1. Edit ApsimRun.sh script.

% cd ..

% cp ApsimRun.sh ApsimRun.sh.bak

% chmod 755 ApsimRun.sh

Add the following variables to the start of the script.

APSIM\_MODEL\_PATH=/usr/local/opt/software/apsim/apsim-7.9-gnu/Model

APSIM\_EXE=$APSIM\_MODEL\_PATH/Apsim.exe

MONO=/usr/local/opt/software/mono/mono-4.8.1.0-gnu/bin/mono

LD\_LIBRARY\_PATH=/usr/local/opt/software/mono/mono-4.8.1.0-gnu/lib:\

/usr/local/opt/software/gcc/gcc-4.8.5/lib64:$APSIM\_MODEL\_PATH

Replace

"$APSIM/Apsim.exe" "`basename "$file"`"

with

$MONO $APSIM\_EXE "`basename "$file"`"

1. Edit LinuxCPP.make configuration files.

% cd Build

% cp LinuxCPP.make LinuxCPP.make.bak

Change the following lines in LinuxCPP.make.

Replace

CC=/usr/bin/g++

with

CC=/usr/local/opt/software/gcc/gcc-4.8.5/bin/g++

Replace

LIBS:= -L$(APSIM)/Model $(foreach lib,$(LIBS),-l$(lib))

with

LIBS:= -L$(APSIM)/Model -L/usr/local/opt/software/tcltk/tcltk-8.6.4-gnu/lib $(foreach lib,$(LIBS),-l$(lib)) \

1. Edit LinuxFOR.make

% cp LinuxFOR.make LinuxFOR.make.bak

Change the following lines in LinuxFOR.make.

Replace

CC=/usr/bin/g++

with

CC=/usr/local/opt/software/gcc/gcc-4.8.5/bin/g++

Replace

FC=/usr/bin/gfortran

with

FC=/usr/local/opt/software/gcc/gcc-4.8.5/bin/gfortran

1. Add a password to extract the Cotton module software using the 7z compression program.

% cd ../Cotton

% cp Makefile.unix Makefile.unix.bak

Change the following line in Makefile.unix.

Replace

7z x ./Cotton.7z -p`cat /etc/CottonPassword.txt`

with

7z x ./Cotton.7z -pPASSWORD

The password to decompress the 7z archive must be obtained from the APSIM development team.

1. Build APSIM

Configure the necessary paths and script permissions.

% cd ../Build

% MONO\_PREFIX=/usr/local/opt/software/mono/mono-4.8.1.0-gnu

% P7ZIP\_PREFIX=/usr/local/opt/software/p7zip/p7zip\_16.02-gnu

% export MONO\_PREFIX

% PATH=$MONO\_PREFIX/bin:$P7ZIP\_PREFIX/bin:$PATH

% chmod u+x ./BuildAll.sh

% ./BuildAll.sh

On a high performance machine the build will take approximately 5 to 6 minutes.

Running APSIM in a HPC Environment

APSIM was not designed to be run in a HPC environment and has behaviours that can cause problems.

The problems are:

1. Trying to use too many CPUs
2. Generating many small files

**Trying to use too many CPUs**

HPC environments require careful management of CPU resources.

The built-in APSIM job manager by default will run as many concurrent tasks as it detects CPUs present in the machine it is running on. The job control system of a HPC environment will restrict the number of available CPUs to what was requested.

The issue is best explained with an example HPC job called *littlecrop*.

1. HPC APSIM job littlecrop requests 4 CPUs because it requirements are modest.
2. The HPC job management system starts littlecrop running on a compute node with 20 CPUs.
3. APSIM detects that the compute node has 20 CPUs and starts 20 concurrent tasks even though it is restricted to running on just 4 CPUS.
4. littlejob running 20 concurrent tasks on 4 CPUS will almost certainly take longer than if it was restricted to 4 concurrent tasks on 4 CPUS.

The solution to the problem is to set the environment variable

NUMBER\_OF\_PROCESSORS. The value of the variable tells APSIM the maximum number of concurrent tasks it can start. In the littlecrop example NUMBER\_OF\_PROCESSORS would be set to 4.

**Generating many small files**

APSIM tends to produce a large number of small output files. In a HPC environment large numbers of small files generate additional overhead for the file-system as they are inefficient. There are a few steps that can be taken to manage the issue.

1. Do nothing. If an APSIM run is only generating a few hundreds or even thousands files on an occasional basis then the performance impacts will be minimal.
2. Use scratch disk. Configure the ASPIM job to write its output to a temporary (scratch) file-system. Once the job is complete only the required output files should be copied to a results directory. Do not keep unnecessary temporary files.

This approach works because the scratch disk areas will usually be faster than the file-system on to which the results are copied.

HPC System Administrators will be able to advise on the best strategy for using scratch disk.

Summary

ASPIM can without much difficulty be built and successfully run in a HPC environment.