Weather Research and Forecast (WRF) Model Port to Windows: Preliminary Report

John Michalakes

National Center for Atmospheric Research, Boulder, Colorado USA 80301

+1 303-497-8199

michalak@ucar.edu

Abstract

The Weather Research and Forecast (WRF) project is a multi-year/multi-institution collaboration to develop a next generation regional forecast model and data assimilation system for operational numerical weather prediction (NWP) and atmospheric research. Under this project – a collaboration between The National Center for Atmospheric Research (NCAR), Microsoft Corporation, Advanced Micro Devices, Inc., and The Portland Group, Inc (PGI) – a prototype version of WRF has been developed and demonstrated running in parallel using MPI on an AMD Opteron™ dual-core processor based server running Windows Compute Cluster Server 2003, Microsoft Compute Cluster Pack,. This preliminary report describes progress and issues encountered in porting WRF to an HPC cluster running Windows, using Microsoft Subsystem for UNIX-based Applications (SUA) and the PGI Fortran compiler.

1. Introduction

The Weather Research and Forecast (WRF) model [1,2] is a next-generation community weather model and data assimilation system designed and implemented from the start as a high-performance computing (HPC) application. Testing towards operational implementation is in final stages at the U.S. National Weather Service, the U.S. Air Force Weather Agency, and a number of national forecast centers in other countries. WRF is freely distributed to the atmospheric science community and presently has more than 3,000 registered users across a range of research organizations, universities, and commercial users both in the U.S. and abroad. WRF is supported across a full range of HPC systems, all of which run some variant of the UNIX operating system. Windows emergence as a viable HPC operating system broadens the

range of computational resources for atmospheric research and weather forecasting and presents a significant opportunity to the WRF user community.

This preliminary report describes the current status of and technical issues encountered in a collaborative effort between NCAR, Microsoft, AMD, and PGI to adapt WRF to the Windows Compute Cluster Server 2003 environment. The next section describes status of the effort followed by an overview of major issues encountered. Detailed notes discussing both major and more minor changes that were made to the WRF Version 2.1.2 distribution to port to Windows are provided at http://www.mmm.ucar.edu/wrf/WG2/wrf_port_notes.htm.

2. History and Status

This project to port WRF to a Microsoft-based HPC cluster was undertaken following discussions between NCAR, Microsoft, AMD, and PGI during the Supercomputing 2005 conference last November. Microsoft has provided evaluation licenses for Windows Compute Cluster Server 2003 and the Subsystem for UNIX Applications (SUA) environment for adapting UNIX applications to Windows. The SUA environment is a useful and familiar environment for building UNIX Applications on Windows, but is not required for running the WRF Model on Windows Compute Cluster Server 2003. So, Windows Server 2003 R2 was used on the development workstation as current version of SUA cannot be installed on Windows Compute Cluster Server 2003. In addition, Microsoft has provided consulting during the porting effort and coordination with AMD and The Portland Group on hardware and software for a development system located at NCAR. AMD has provided hardware in the form of four AMD Opteron™ dual-core processor based nodes and consulting. The Portland Group has provided PGI Workstation parallel Fortran, C and C++ compilers and tools for installation on the development cluster and consulting. NCAR has provided the WRF model as well as the effort and model-specific expertise for porting the model to Windows.

The WRF model has been successfully compiled, linked, and run as an MPI parallel application on the AMD Opteron™ dual-core nodes running Windows Compute Cluster Server 2003.
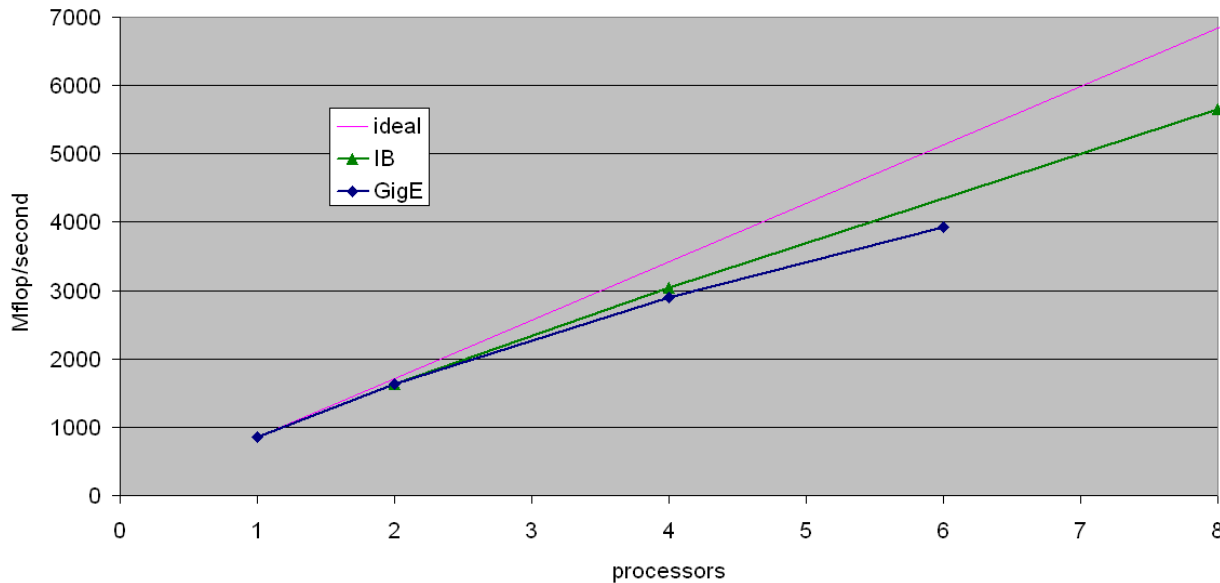

Figure 1: Performance of Windows Compute Cluster Server 2003 on WRF 12km CONUS standard benchmark case.

Figure 1 shows model performance from one processor (856 Mflop/second) to eight processors (5657 Mflop/second, equivalent to a simulation rate of 12.6 times wall-clock time for this WRF benchmark case). Parallel efficiency – that is, the percentage of ideal speedup achieved – is 82.6 percent from one to eight processors using Infiniband between nodes. For comparison, the second plot shows performance from one to six processors using a Gigabit Ethernet network. Network speed, in particularly latency, is a key factor for WRF parallel performance. WRF parallel scaling is latency-limited, as opposed to bandwidth-limited, since the model sends a large number of relatively small messages between MPI tasks during each model time step. Microsoft Compute Cluster Server 2003 supports both Infiniband and Myrinet interconnects.

 Figure 2 shows sample output from this WRF benchmark case (12km CONUS, October 2001). This output has been validated with respect to standard WRF benchmark statistical criteria and subjective meteorological evaluation.
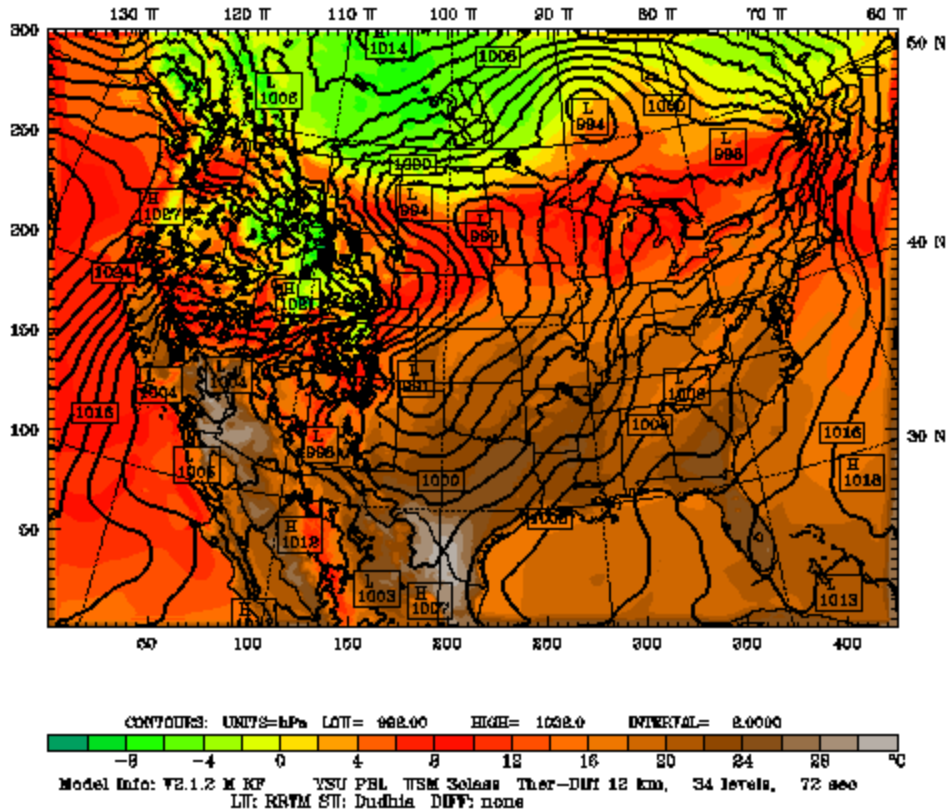
Figure 2: Output from WRF running under Windows showing surface temperature and sea-level pressure for third hour of standard 12km CONUS benchmark case

3. Overview of Major Issues

The WRF Version 2.1.2 software distribution comprises some 360-thousand lines of source code and related files (e.g. Makefiles, scripts, etc.). Of this total, fewer than 750 lines required modification. These changes, along with the need to change suffixes for file names in the WRF distribution, are described here.

3.1. File name suffixes

Suffixes in the names of the 247 Fortran source files in WRF were changed from .F to .F90 to account for case-insensitivity of the Microsoft Windows file system in our SUA setup configuration. This is because the WRF build preprocesses each Fortran source file from source files with the .F suffix) to temporary files using the .f suffix before invoking the compiler. Note that SUA can be installed with case-sensitive behavior enabled by default (see http://technet2.microsoft.com/WindowsServer/en/library/695ac415-d314-45df-b464-4c80ddc2b3bc1033.mspx?mfr=true).

3.2. Changes to WRF build mechanism

These changes involved modifications to WRF Makefiles to account for differences arising from the change of file name suffixes, as well as changes required to use different commands, path names to libraries and include files, suffixes for object and library files, etc.

The object file archiving utility under Windows is named `lib.exe`; in UNIX this is the `ar` utility. The resulting libraries use the `.lib` extension instead of the `.a` extension used under UNIX. The command line arguments and behavior also differ with respect to incremental archiving of object files to a library. Incremental archiving did not work correctly if the `/out` option was used to change the library file name extension from `.lib` to `.a`; therefore, `.lib` was used throughout and the WRF Makefiles were adjusted accordingly. PGI has added a UNIX-like ar utility in a subsequent revision of the PGI compilers, which should make this aspect of the port more seamless, but it was not available during this initial porting exercise.

Anywhere in the WRF build that an absolute path name is used – the location of the libraries and include files for NetCDF and the Microsoft implementation of MPI – needed to be specified using Windows and not SUA absolute path names. For example, it was necessary to use `C:\Program Files\Microsoft Compute Cluster Pack\Lib\AMD64\msmpi.lib` instead of `/dev/fs/C/Program Files/Microsoft Compute Cluster Pack/Lib/AMD64/msmpi.lib` for the link to work properly. Otherwise, the `pgf90.exe` compiler driver used to compile, assemble and link the model would pass filenames beginning with '/' to `link.exe`, which would interpret them as the start of a Windows command-line option. This is an artifact of using native

Windows x64 compilers and commands within the SUA environment, and would be unnecessary if the PGI compilers and toolchain were adapted to support SUA directly as a host and target environment. UNIX-style relative path names were accepted without a problem.

The standard WRF build uses the "`/lib/cpp -C -P`" command to preprocess C-style preprocessor directives. Using the Microsoft C compiler with the /C and /EP options was an acceptable substitute.

3.3. Time strings in WRF data file names

Input and output file names in WRF sometimes include an ISO standard time-string that uses colon separators between hours, minutes, and seconds; for example:

wrfout_d01_2001-10-24_00:00:00

Colons are significant in Windows file names so that this convention cannot be used. The WRF code was modified to use underscore characters instead when generating file names on a Windows platform.

3.4. Use of the wcc command

The `wcc` command provided informally by Microsoft is a script that implements UNIX-like behavior for the Windows `cl` command-level C compiler. However, the `wcc` command has some problems with the include of the file `fcntl.h`. Specifically, O_CREAT, O_WRONLY, and other control modes for the open system call are not defined (_O_CREAT and O_WRONLY are). It was a simple matter to include some extra preprocessor logic in the affected C routine in WRF that tests if O_CREAT and O_WRONLY are defined and, if not, defines then to _O_CREAT and _O_WRONLY. The wcc command is available from the Windows HPC community site on http://windowshpc.net/files/4/porting_unix_code/entry11.aspx.

3.5. Other Porting Issues

A difficulty was encountered using the Microsoft `link.exe` linker to generate the WRF executable from the PGI-compiled WRF Fortran object libraries and other necessary libraries

that are written in C. WRF calls the RSL_LITE communications library wherein calls to MPI as well as domain decomposition and other methods relating to distributed memory parallel execution are performed. Initial attempts using the `pgf90.exe` compiler driver and `link.exe` to link WRF under Windows resulted in unresolved references to the I/O routines declared in stdio.h, such as the C output routines `printf` and `fprintf` referenced in the RSL_LITE source code.

It turns out this was related to the default method of compiling the C code in RSL_LITE using `wcc` and a limitation in the current version of the PGI Fortran compiler for Windows x64. By default, `wcc` generates code in a way that assumes the resulting executable will be statically linked to the `libcmt.lib` version of *libc* rather than dynamically linked using the `msvcrt.dll` version of *libc*. The current version of the PGI Fortran compiler for Windows x64 does not support static linking of executables. To work around this problem, it was necessary to compile all RSL_LITE code using the `/MD` command-line option which specifies the resulting executable will be dynamically linked. In effect, this was an incompatibility between the default behaviors of the PGI Fortran and Microsoft CL compilers, easily resolved once the issue was well understood.

A secondary concern is that there was no obvious command-line debugger such as `dbx` or `gdb` available for use with the PGI Fortran compiler in the SUA environment. It turns out that PGI's PGDBG graphical parallel debugger is available and fully supported on Windows x64 for debugging of serial and OpenMP parallel Fortran 90 code, but it does not yet support debugging of MPI applications on Windows. All debugging of the current proof-of-concept MPI implementation of WRF was accomplished using Fortran print statements. PGI supports MPI debugging on other platforms, and is working toward adding MPI debugging capability to PGDBG for Windows Compute Cluster Server 2003, and expanding support for debugging code generated by the Microsoft CL compiler.

4. Summary and Ongoing Development

The discussion above is intended as a preliminary report on the effort to adapt the WRF model to the Windows environment. In general, while certain issues remain, it is impressive to see that

with a relatively small number of changes limited mostly to the WRF build mechanism, the Windows SUA environment and related software was successful at facilitating the transition of a large, existing UNIX HPC application such as WRF to the Windows Compute Cluster Server 2003 environment. Ongoing development activities will include extending the current development system to allow testing and benchmarking on multiple network-connected compute nodes; and benchmarking the system on larger Windows cluster configurations.

Acknowledgments

References

1. Skamarock, W., et al, "A Description of the Advanced Research WRF Version 2", NCAR Technical Memorandum, NCAR/TN-468+STR, 2005

2. www.wrf-model.org