WRF 2.0 Software

John Michalakes, David Gill, Jennifer Abernethy

National Center for Atmospheric Research, Boulder, Colorado

1. Introduction

The release of WRF version 2.0 represents a software development milestone in an effort that, begun with a blank sheet six years ago, set out to design and implement a fully-functioning, next-generation modeling system for the atmospheric research and operational NWP communities. With efficiency, portability, maintainability, and extensibility as bedrock requirements, the WRF software architecture and the WRF Advanced Software Framework (ASF) that implements it (Michalakes *et al.* 1998) have allowed incremental and reasonably rapid development towards full-functionality while maintaining overall consistency and adherence to the architecture and its interfaces. This presentation provides an overview of the features that are new or significantly updated in Version 2 of the WRF software.

2. Features in WRF 2.0

The latest release of WRF includes nesting for mesh refinement, significantly enhanced I/O, file handling, time management from the Earth System Modeling Framework (ESMF), and a new, highly-scalable parallel communication layer called RSL_LITE. A body of detailed up-to-date documentation has been produced and made available, with more in-progress. Web-based browsing of the WRF code, data structures, and in-line code documentation is now available. Efficiency and rigor of code maintenance and quality control are improved through the use of a CVS-based source-code repository and scripts for automatic regression testing of WRF in numerous configurations across multiple platforms.

Nesting

The WRF ASF supports one- and two-way nesting, described elsewhere in this workshop (Gill *et al,* 2004). The WRF ASF provides data structures for representing multiple nested domains and operations for managing the domains and coordinating their interactions over the course of a simulation. Each domain is represented by a dynamically allocated Fortran 90 Derived Data Type (DDT; also called a *structure)* containing fields that store the state and configuration data for the domain. Within the WRF driver layer, the nesting hierarchy is represented as a graph (a tree). Domain DDTs are the nodes and the parent-child nesting relationships (pointers between the DDTs) are the edges. The root node of the tree represents the top-most domain. Integrating the model involves a depth-first recursive traversal of this tree (see frame/module_integrate.F).

The set of fields that make up domain state, as well as forcing, feedback, interpolation, smoothing, and coupling operations, are specified at a high-level as entries in the WRF Registry (Michalakes, 2003). An external mesh-refinement library, RSL (Michalakes, 2000), provides lower-level infrastructure support. This includes decomposing domains over processes, keeping track of geographically coincident nest and parent domain points, and generating efficient interprocess communication to exchange data between domains. Initial parallel benchmarks on the IBM suggest that the WRF implementation incurs additional overhead of between 5 and 8 percent for nesting. This compares favorably with the design target of no more than 15 percent overhead (based on nesting performance of the parallel MM5).

I/O and File Handling

WRF 2.0 provides multiple I/O streams: default input and five auxiliary input streams, default history and five auxiliary history streams, restart, and lateral boundary. Sets of state variables are bound to streams in the WRF Registry. Attributes such as format, dataset name (with optional domain id and time stamp), initial I/O time, final I/O time, and I/O interval may be individually set in the namelist at run-time for each stream.[1] *Quilting,* asynchronous output through a set of dedicated output server tasks, may be specified for history and auxiliary history output streams. WRF also has a special stream dedicating to producing model output in WRF input form, which is used for 3DVAR cycling.

Time Management

Starts, stops, and intervals are specified in the namelist as exact integer numbers of years, months, hours, days, minutes, seconds, and/or fractions of a second (numerator and denominator are specified separately as integers). Internally, simulation time intervals and instants are represented as ESMF time manager objects. All time arithmetic involving these

[1] Periodic input of input and auxiliary input is not yet implemented in the released version.

objects is performed exactly, without drift or rounding, even for fractions of a second.

Documentation

Last year, detailed and comprehensive documentation aimed at WRF software developers and maintainers was identified as a critical need in the WRF project. In response, the WRF Training and Documentation Team[1], also known as the WRF Tiger Team, was formed. The inaugural two and a half day meeting in January included a detailed tutorial and planning sessions for producing a complete set of documents for the WRF software by the end of 2004.

In response to recommendations from the USWRP WRF Scientific Steering Committee (Bishop *et al.*, 2004), detailed subroutine-by-subroutine documentation has been implemented and is being maintained on-line.[2] The contents of these web pages are generated automatically from the WRF source code. In-line descriptions, delimited by **<DESCRIPTION> </DESCRIPTION>** tags are extracted and presented as on-line manual pages, along with detailed call tree information and call-chain information for each subroutine argument.

Supported platforms

The model has been ported and run on the following platforms. Configuration options for these are included in the *configure* mechanism provided with the WRF 2.0 code distribution:

- Compaq/HP Alpha
- Cray X1 (not fully optimized)
- HP Superdome (IA64)
- IBM SP Power4 and earlier
- Linux/Alpha
- Linux/PC (IA32/IA64)
- SGI Origin (MIPS) and Altix (IA64)
- Sun Solaris (single process and SMP only)

For Linux systems, both Portland Group and Intel compilers are supported in WRF, though finding the right combination of version and Linux system configuration may be non-trivial.

3. Ongoing Development

Development is continuing in a number of areas. Support for overlapping and moving nests is in-progress. Benchmarking and performance optimization on supported platforms, and porting to other systems such as NEC are underway. NCEP's prototype WRF/NMM dynamical core is being unified with the production version in preparation for incorporation as a second dynamics option for WRF. We are working with data assimilation groups on extending the WRF framework to support WRF 3D/4D-Var. Discussions are also underway to implement analysis and observational nudging. WRF I/O and data representations are being extended for WRF-Chem and regional climate. Prototype coupled WRF/ocean and WRF/ecosystem models have been developed. Additional ESMF functionality is being considered for use in WRF as it becomes available.

[1] http://www.mmm.ucar.edu/wrf/WG2/Tigers

[2] http://www.mmm.ucar.edu/wrf/WG2/software_2.0

Acknowledgements

Many individuals have contributed to the WRF software development, including members of Working Group 2 and the WRF Training and Documentation (Tiger) Team. Thomas Black, S.G. Gopalakrishnan (NCEP) and Jon Wolfe (NCAR) are principal developers of the prototype NMM dynamical core in WRF. Jacques Middlecoff and Daniel Schaffer (FSL), Kent Yang (NCSA), Rob Jacob (Argonne), and Matthew Bettencourt (AFRL) have contributed to implementations of the WRF I/O and model coupling API. ESMF collaborators include Cecelia Deluca, Nancy Collins, and Earl Schwab (NCAR). Jim Abeles and Jim Edwards (IBM), Gerardo Cisneros (SGI), William Gustafson (PNNL), Peter Johnsen (Cray), Mark Straka (NCSA), Craig Tierney (HPTI/FSL), and others have contributed to WRF porting, testing, and benchmarking on numerous platforms. Wei Wang and Cindy Bruyere (NCAR) provide ongoing and immeasurable support to the software development team.

References

Bishop, C.H., G. Dimego, C. Mass, and N. Surgi, "USWRP SSC WRF Working Group Report," February 1, 2004.

Gill, D., J. Michalakes, J. Dudhia, W. Skamarock, and S. G. Gopalakrishnan, "Nesting in WRF 2.0" (2004). Joint MM5/WRF Users Workshop, Boulder, Colorado, June 22-25, 2004.

Michalakes, J., WRF Design and Implementation Document, § 5, (2003). See: http://www.mmm.ucar.edu/-wrf/users/WRF_arch_03.doc

Michalakes, J.: RSL: A Parallel Runtime System Library for Regional Atmospheric Models with Nesting, in Structured Adaptive Mesh Refinement (SAMR) Grid Methods, IMA Volumes in Mathematics and its Applications (117), Springer, New York, 2000, pp. 59-74.

Michalakes, J., J. Dudhia, D. Gill, J. Klemp and W. Skamarock: Design of a next-generation regional weather research and forecast model: Towards Teracomputing, World Scientific, River Edge, New Jersey, 1998, pp. 117-124.