Performance of WRF using UPC

Hee-Sik Kim and Jong-Gwan Do*
Cray Korea

ABSTRACT: The Weather Research and Forecasting (WRF) model is a next-generation mesoscale numerical weather prediction system. We developed a Unified Parallel C (UPC) version of RSL_LITE which is a communication layer of WRF. This paper will discuss the performance of WRF using UPC on the Cray X1E system. In addition, this paper also includes some optimization techniques for WRF on the Cray X1E system.

KEYWORDS: WRF, Unified Parallel C, RSL, RSL_LITE, X1E, optimization

1. Introduction

The Weather Research and Forecasting (WRF) model has been developed as a next-generation model to advance both the research and operation of mesoscale forecasting. It is expected to replace its predecessor the Mesoscale Model Version 5 [1]. Korea Meteorological Administration also have a plan to adopt Korea WRF for its next regional model, the modified version targeting East Asia region. As the domain size is so wide, it is needed to focus on optimizing the codes. On a parallel machine such as the Cray X1E system, communication time should be considered for optimization.

The parallel implementation of WRF is based on a Runtime System Library (RSL) using Message Passing Interface (MPI) communication package. RSL is a parallel runtime system library developed at Argonne National Laboratory that is tailored to regular-grid atmospheric models with nesting [2]. Although RSL is designed well, it is too heavy. WRF provides RSL_LITE the light version of RSL. It is also scalable to very large domains as an optional new communication layer.

The Cray X1E system supports newer parallel programming models which are not library-based such as MPI but are programming language itself which is based on a global address space that allows one-side communication. Previous papers have founded that these languages such as Unified Parallel C (UPC) and Co-Array Fortran (CAF) had performance profits in various codes [3]. We could expect performance benefits in WRF with UPC either.

In this paper, we will show the method to implement a UPC version of RSL_LITE and the performance of that. We will also give some optimization techniques used in WRF.

* Jong-Gwan Do, now at HP

2. CrayX1E

The Cray X1E system is a parallel vector processor system. Each node on the X1E contains four Multi-streaming processors (MSPs) and has 16 GB of uniform memory access shared memory. It has advantages such as very fast processor inter-connected network, a global and unified memory address space, a single system image and an advanced programming environment [4].

We used KMA's Cray X1E system the fastest operational numerical weather prediction system in the world. This Cray X1E is the 2nd supercomputer of KMA which is the replacement of the old NEC SX-5/28. It was installed in two phases: the initial system (2.4 Tfolps of peak performance) in Oct, 2004 and the final system (18.5 Tflops of peak performance/ 15.7 Tflops of sustained performance) in Dec, 2005. The system is composed of the main compute server of 8 Cray X1E cabinets, login servers, pre-post servers and storage system. The TOP500.org announced that KMA's Cray X1E system ranked at No. 16 on the list published in Nov, 2005 [5].

Figure 1. KMA's Cray X1E system which comprises of 1024 MSPs.

3. WRF

3.1 Introduction: WRF

WRF project started as a joint development effort of many institutes and universities in U.S. The motivation of this effort is that it will lead to a common model framework for research and operational communities so that easily exchange information between research and operational products. WRF is used to study not only small scale weather features including storm-scale research and prediction, hurricane and tropical storm prediction and regional climate but also air-quality model, wildfire simulation and so on. Operational implementation of WRF is underway at various weather centers.

fully compressible nonhydrostatic equations with hydrostatic option
complete coriolis and curvature terms
two-way nesting with multiple nests and nest levels
mass-based terrain following coordinate
vertical grid-spacing can vary with
Arakawa C-grid staggering
Runge-Kutta 2nd and 3rd order timestep options
scalar-conserving flux form for prognostic variables
full physics options for land-surface, PBL, radiation, microphysics and cumulus parameterization

Table 1. The key features of WRF model

Generally WRF system consists of WRF model itself, preprocessors for producing initial and lateral boundary conditions for idealized, real-data and a three-dimensional variational data assimilation. In this paper, we consider WRF as

WRF model itself which version is 2.1.2. See Table 1 for a description of WRF. More detailed information can be found on WRF homepage (http:// www.wrf-model.org).

3.2 Role of RSL_LITE

Limited area models of the atmosphere for weather forecasting and climate prediction are computationally intensive applications that can benefit from the high performance and large memories of scalable distributed-memory parallel computers. Those models needs to adopt stencil-library approaches for parallelization so as to resolve, for example, the effects of complicated terrain, forcing and feedback data of multiple nested domains. The efficient implementation of these models requires attention to load balances. RSL, a representative runtime system library, is well known by using in Mesoscale Model 5. Although it is still adopted in WRF, for the convenient use WRF recommends RSL_LITE provided since version 2.1.

RSL_LITE is the light version of RSL. Its fast and efficient communication could be possible by limiting of girds and processor subdomains to rectangular geometries. Normally as subdomains are not decomposed to an irregular shape, there is no inconvenience in RSL_LITE. It is written in C and used with Fortran90 same as RSL.

4. UPC

4.1 Introduction: UPC

UPC is an extension to the C language which gives processors direct access to data located on other processors by simple reads and/or writes to data and/or arrays which are declared to be shared within a user's application code. UPC is not library-based programming tool but is language itself to support inter-processor communication and access to data distributed across the parallel machine. The feature which is that all memory is addressable could be an advantage of parallel weather and climate applications. A detailed description of the UPC is not the focus of this paper, but more details on it can be found at http://upc.gwu.edu and http://upc.lbl.gov.

We can abstract the benefits of UPC as follows; 1) The performance improvement by fast one side communication. 2) Source code brevity and clarity compared with MPI 3) The intuitive and easy usage.

4.2 UPC version of RSL_LITE

Main features of UPC version on RSL_LITE could be summarized like these.

1) Receive buffer is unchanged.
2) Send buffer is replaced by shared memory that has affinity to the local node and is allocated by "upc_alloc" and freed by "upc_local_free".
3) Shared pointer to shared memory with local affinity trick is used.
4) Shared memory which has affinity to local node can be treated as local memory so normal memory copy and pointer operation is used.
5) MPI_ISEND / MPI_IRECV / MPI_WAIT sequence is replaced by "upc_memget" on one-side communication.

5. Some Optimization techniques

In WRF version 2.1.2, RSL_LITE introduced "f_pack.F" which copies between halo area data and packed data for exchanging. We added 2-D f_pack*/f_unpack* subroutine for vectorization. Other source codes are inserted X1E specific directives to improve streaming and vectorization efficiency and are added additional rank for packed variables to better support vectorization and to aid compiler optimization.

6. Case study: CONUS benchmark

For the general experiments, we decided to run CONUS benchmark. This is one of the benchmark cases that WRF officially provides, covers the continental US with single domain. See Table 2 and Figure 2 for the specific configuration.

Resolution	425x300
Vertical levels	31 levels
Grid size (dx)	12 km
Integration time(DT)	72 secs
Forecast time	3 hour
Other features	- Eulerian Mass dynamics - Only computational part included (no initialization and file I/O)

Table 2. Description of CONUS benchmark case

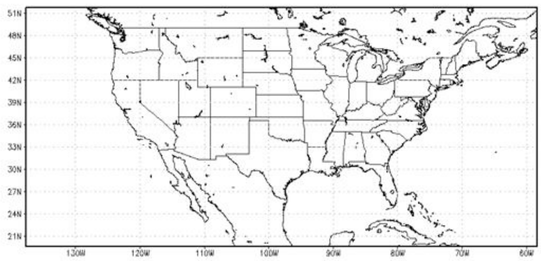

Figure 2. Domain of CONUS benchmark case

The Cray Programming Environment Version is 5.5.0.3 was used to build WRF model for this test.

We performed the same run using 1, 2, 4, 8, 16, 32, 64, 128 and 256 MSPs. The total run time was measured and was converted to Giga-Flops rate based on the total number of calculations performed. The CONUS benchmark only included computational parts, therefore there is no initialization and file I/O time. Unfortunately KMA's X1E machine which is partitioned Baram and Shinbaram, each of them consists of 512 MSPs, was not dedicated to the test. There could be some overhead in total run time. The results are shown in Figure 3.

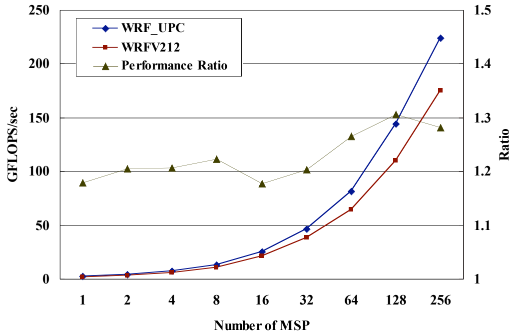

Figure 3. Performance of WRF for CONUS benchmark: The blue line designates for UPC version and the red one for original version. The green line is the performance ratio.

As can be seen in this figure, the performance of UPC version shows large improvements compared with original version. The performance ratio is between 1.1 ~ 1.3. While UPC version contains some optimizations in RSL_LITE routine mentioned chapter 5, it does not influence significantly the performance. (Not shown in this paper) One notable thing is that the performance ratio fluctuates as the numbers of MSPs invrease. We believe this is because of domain decomposition. In Figure 3, shown data from over than 16 MSPs were collected from various domain decomposition for the best performance data. In order to verify the effect of UPC as CPU increases, it is needed to compare the performance of 1-dimensional domain decomposition in Figure 4.

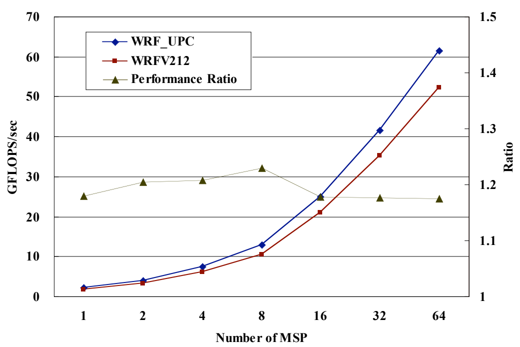

Figure 4. Same as Figure 3, except for 1-dimensional domain decomposition which means 1xN decomposition

Showing similar results in Figure 3, still good performance for a UPC version, the performance ratio dropped a little as MSPs numbers added up. We guess "upc_barrier" the global synchronization, could be the cause. As the lack of partial synchronization between UPC threads requires global synchronization, it could be bottleneck for large number of processors. Although Time limitation for the deadline prevents from investigating detail communication time, we need look into it deeply.

7. Conclusion

We have shown details about the use of Unified Parallel C within RSL_LITE on WRF model and have shown improved performance data. We will investigate possibility of further optimization in synchronization.

About the Authors

Hee-Sik Kim is Application Analyst for Cray. They have worked with atmospheric and climate models on Cray Korea Inc. He could be contacted at hskim@cray.com.

Jong-Gwan Do is former Application Analyst for Cray. E-mail contact is jonggwan@gmail.com.

References

1. Michalakes, J., J. Dudhia, D. Gill, T. Henderson, J. Klemp, W. Skamarock, and W. Wang, 2004: "The Weather Reseach and Forecast Model: Software Architecture and Performance," *Proceedings of the 11th ECMWF Workshop on the Use of High Performance Computing In Meteorology,* 25-29 October 2004. See also http://www.wrf-model.org

2. Michalakes, J., 1997: Reference Manual for RSL: A runtime system library for parallel Weather Models, *Tech. Rep. ANL-00,* Argonne National Laboratory, Argonne, Illinois.

3. A. Johnson, 2005: Unified Parallel C within Computational Fluid Dynamics Applications on the Cray X1(E), Proceedings of the 2005 Cray User Group Conference, Albuquerque New Mexico, 16-19 May 2005.

4. Cray Inc., 2004: Cray X1 Series System Overview, S-2346-25, www.cray.com/craydoc

5. Top500 Supercomputer Sites: www.top500.org