11.8 PORTING THE HIGH RESOLUTION LIMITED AREA FORECAST SYSTEM (HLAFS) ON SHARED AND DISTRIBUTED MEMORY HARDWARE PLATFORMS FOR REAL-TIME OPERATIONAL WEATHER FORECASTING IN CHINA.

Zaphiris Christidis[1]
IBM Thomas J. Watson Research Center,
Yorktown Heights, NY 10598.

Zhenghui Zhu.
China Meteorological Administration
Beijing, China

1. INTRODUCTION

Weather forecasting via numerical techniques, imposes high demands on computer processing time and other computer resources. The use of high resolution numerical techniques for the discretization of the modeling domain in space, the incorporation of more complex representations of physical processes or the exploitation of longer time scale phenomena can easily consume the resources of a large computer system. The advent of parallel processing opens a new era in numerical weather prediction. The basis of parallel computing lies in the concept of partitioning the computational domain among parallel processors. Domain decomposition techniques are usually dictated by the numerical algorithms and the nature of physical processes defined in the model. In this paper we describe the methodology and techniques used to optimize and parallelize the HLAFS Regional Weather model.

2. THE HLAFS REGIONAL MODEL

The HLAFS (High Resolution Limited Area Forecasting System) model is a mesoscale grid point weather model suitable for regional weather prediction (Yan Zhihui *et al*., 1996). It has continuously undergone changes and modifications towards the improvement of various physical processes and parameterization schemes with aim to predict flood events in continental China. HLAFS is a hydrostatic model and it employs the basic primitive equations of motion as specified on a spherical coordinate system, where the independent variables are the latitude θ, longitude λ, height σ, and time t. The primitive equations prescribe as prognostic dependent variables the east-west wind velocity component u, the north-south wind velocity component v, the temperature T, the specific humidity q, and the surface pressure, p_s. The model also includes prognostic equations for cloud and rain water, based on various microphysical processes. The rest of the dependent variables like the vertical velocity $\omega(\sigma)$ and geopotential height Φ, are prescribed via a sub-set of diagnostic equations. HLAFS employs the Arakawa "C" staggered grid, while the numerical schemes used are second order finite differences in space and time. The model incorporates the explicit leap-frog time difference scheme in combination with a time filter to advance the dependent prognostic variables in time. HLAFS uses a terrain following vertical coordinate, kinematic top and bottom boundary conditions ($\sigma=0$), and forecast lateral boundary conditions. It incorporates several physical processes (Yan Zhihui, 1999), collectively termed as "physics" involving computations to simulate a wide range of atmospheric phenomena relating to clouds and precipitation, dry and moist convection, long and short wave radiation, as well as the development and evolution of the planetary boundary layer. All these processes involve independent computations on each vertical column in the model domain. The processes termed as "dynamics" involve computations mainly in the horizontal directions. These computations treat the non-linear and linear terms in the primitive equations used for the solution of the dependent model variables.

The HLAFS model was mainly operated on a CRAY vector computer at the China Meteorological Administration (CMA), on a 181x119x20 ***I*x*J*x*K*** grid of .5 degrees of horizontal resolution (***I*** represents the number of grid points in the longitudinal direction, ***J*** the number of grid points in the latitudinal direction and ***K*** the number of height levels). Currently the model runs operationally at a .25 degree of horizontal resolution utilizing a 361x273x20 grid on a Power3 IBM RS6000 SP (http://www.ibm.com/rs6000).

3. HLAFS CODE OPTIMIZATION AND TUNING

3.1 The IBM RS6000 SP Parallel System

HLAFS was tuned and parallelized on the IBM RS6000 SP system. Tuning of the serial version of the code was performed on a single POWER3 CPU. The code was tested on CMA's 10 node SP system. Each node contains 8 CPUs interconnected via a shared memory crossbar switch. This computer architecure is known as Symmetric Multi Processor (SMP). On this system each CPU operates at 222 MHz. Different nodes can communicate with each other via a multistage interconnect network known as the High Performance Switch, (HPS). Communication is achieved

[1] Corresponding author address: Zaphiris Christidis, IBM TJ Watson Res Center, PO Box 218, Yorktown Heights, NY 10598; e-mail: zaphiri@us.ibm.com

when nodes send or receive message packets through the HPS (Distributed Memory Processing, DMP). The HPS provides a low-latency high-bandwidth communications fabric that can sustain point-to-point bandwidth of over 110 megabytes per second (MB/sec), with latencies close to 30 microseconds for a variety of applications. Each node of the SP runs its own copy of AIX (IBM's implementation of Unix). A parallel program therefore consists of a number of Unix tasks exchanging messages via the HPS. The messages can be encoded using the industry standard Message Passing Interface (MPI), which is a programming interface for FORTRAN and C, and provides with a large number of library subroutines supporting task-to-task and collective communication message passing primitives. The CPUs within each node can also communicate data via MPI, or they can share execution, using OpenMP, which facilitates shared memory programming and execution of applications (Chapman *et al.*). In this paper we will provide general details on the execution of HLAFS using the CMA's SP (denoted as NH1), as well provide benchmark results on a faster SP system consisting of 4 CPU per node, with each CPU operating at 375 MHz (WH2).

3.2 Single Node Tuning

In tuning standard FORTRAN code for the cache based POWER3 processor architecture, we took into account that basic arithmetic operations (addition, multiplication, etc.) take a single CPU cycle to execute, while a division operation can take anywhere between 16-19 CPU cycles. A total of 4 floating point independent instructions can be executed by the dual floating point unit of the CPU within a single cycle, thus delivering a peak performance of 888 Million of Floating Point Operations per second (MFLOPS) on a single NH1 CPU. A raise to a power (RAP) operation takes roughly 160 cycles to complete, while the computation of a single logarithm or an exponential function averages around 55 cycles.

It was decided to re-construct the code, by keeping the same programming structure, but eliminate redundancies, unnecessary memory usage and "dead-code" segments. The code was initially profiled, and runtime statistics were obtained in order to focus the optimization efforts to the parts and routines consuming most of the CPU power. Our first approach was to rearrange the order of the array indices in the meteorological variables from (***I,J,K***) to (***K,J,I***). This change allowed faster execution of the physics routines, as array elements in the column physics routines were accessed by a unit stride, resulting in efficient cache use. The code was altered to execute in 32 or 64 bit precision. The 32-bit version of the serial code was slightly faster to complete a 24 hour forecast than the 64 bit version. This was expected despite the fact that floating point operations on the POWER3 CPU are performed in 64 bits. The difference in timing was due to better cache utilization, as single vertical columns of 20 elements (80 bytes) in 3D arrays could fit within the 128-byte cache lines. The 32 bit version of the parallel code was faster than the 64, mainly due to the effects of MPI and OpenMP having to move twice as much data over the HPS and the shared memory respectively. In order to ensure accurate precipitation predictions, all relevant routines were executed in 64 bit precision, so errors due to numerical truncation could be avoided. Lookup tables for the saturation vapor pressures were introduced to eliminate expensive operations involving exponentials. Also, various do-loop structures and iterations were modified with the objective to achieve the minimum necessary floating point operation counts. Improved cache utilization was achieved by inverting several do-loop indices within the code, so data in 3-D arrays could be accessed with minimal memory strides. A faster but less accurate library of the mathematical intrinsic functions, called MASS, was used to further reduce the cycle count, in the standard logarithmic, exponential and RAP intrinsic functions, without any modification to the source code. Since divisions are relatively expensive to perform as opposed to multiplications, the reciprocals of selected meteorological parameters were computed and stored in memory. Consequently, divisions were replaced with multiplications whenever possible. Since several approximations were introduced, the accuracy of the numerical computations was monitored by checking the RMS errors of various horizontally integrated model quantities such as kinetic energy, momentum etc.. The RMS errors remained sufficiently small during model test simulations.

4. **PARALLEL IMPLEMENTATION**

4.1 Domain Partitioning

Due to the mixed architecture of the IBM RS6000 SP (SMP+DMP), it was decided to partition the ***J*** index among available computational nodes, such that parts of HLAFS can execute concurrently on every node. The nodes need to communicate at appropriate intervals to exchange data in order to ensure identical numerical results with the ones obtained from a single processor system. Such communication was facilitated via the use of the MPI library. The ***I*** index was allowed to be processed concurrently by CPUs within a single node. This was achieved via the use of OpenMP compiler directives within the FORTRAN code. As a result, AIX schedules "computational threads" on various CPUs within each node during runtime, where each thread executes a fraction of the ***I*** grid points on a CPU. Since there is tight coupling in the vertical direction on various physical processes like radiation, cloud/precipitation and turbulence, any decomposition over the vertical coordinate would cause high communication overhead. In this work, we focused only on decomposing the horizontal coordinates, as they simplify code development, and provide adequate parallelism on a large number of SMP nodes. I/O was performed by a single CPU attached to a local file system. This CPU was assigned less workload in order to achieve better computational load balancing among participating nodes.

5. **RESULTS**

Table 1 shows the memory requirements for HLAFS in 32/64 bit precision, low/high resolution, and MPI/OpenMP processing. It is worth mentioning that the memory usage was reduced by a factor of 3 in 64 bit precision and a factor of 5 in 32. Also, the utilization of OpenMP added only a few

Mbytes of extra memory, which could be beneficial for systems with little memory. HLAFS can run on every CPU on the system using only MPI, though the memory requirements can be tolling, causing the operating system to page excessively.

	119x181x20		273x361x20	
Mode	64-bit	32-bit	64-bit	32-bit
MPI	174MB	93MB	690MB	367MB
OpenMP	176MB	95MB	693MB	370MB

Table 1: Memory requirements of HLAFS

Table 2 illustrates HLAFS performance on 4 CMA nodes (total of 32 CPU's) for a 24 hour forecast. The best results were obtained using 2 MPI tasks per node with 4 OpenMP threads. It is worthwhile mentioning that a 72 hour forecast with a time step of 15 seconds, using 4 CMA nodes in 32 bit precision, was projected to complete in 7815 seconds, which is less than the operational criterion of 9000 seconds as imposed by CMA. In addition, the average number of floating point operations per node was obtained by using the hardware floating point performance monitor of the SP. As expected best performance was observed in 32 bit precision when running the high resolution version of the model.

	119x181x20		273x361x20	
Mode	64-bit	32-bit	64-bit	32-bit
24h forecast	589 s	501 s	3684 s	2605 s
MFLOP/node	540	605	560	730

Table 2: HLAFS Performance characteristics.

Figure 1 illustrates elapsed times in minutes for a 24 hour forecast using the high resolution 32 bit version of HLAFS for different number of nodes and computational threads on a WH2 SP system. In this case it is also evident that the best performance is obtained if 2 MPI tasks are scheduled within a single WH2 node, with 2 computational threads each.

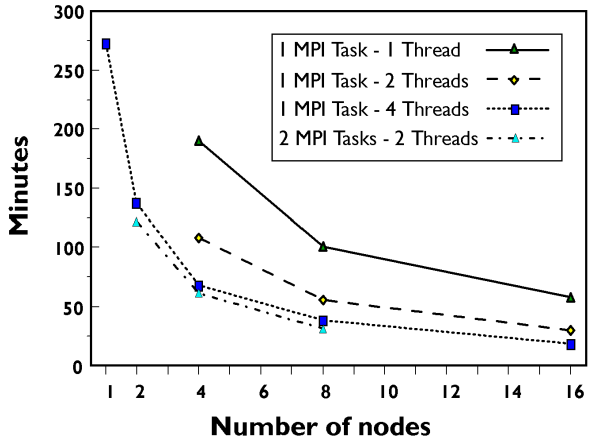

Figure 1: Elapsed times for a 24 hour forecast on WH2

6. COMMENTS

Work is still in progress towards additional tuning of the computations within the physics routines. A more efficient OpenMP implementation is currently planned, where data locality can be minimized, in order to obtain better speedups for a larger number of computational threads. Implementation of more efficient I/O schemes in order to facilitate real time visualization is near completion.

7. REFERENCES

Chapman, B., P. Mehotra and H. Zima, 1998: Enhancing OpenMP With Features for Locality Control. *Proceedings of the Eighth ECMWF Workshop on the Use of Parallel Processors in Meteorology*, Towards Teracomputing, W Zwieflhofer, N. Kreitz, eds., World Scientific Publishing, Singapore, 301-313..

Christidis, Z., J. Edwards, J. S. Snook, 1997: Regional Weather Forecasting in the 1996 Summer Olympic Games using an IBM SP *13th International Conference on Interactive and processing Systems*, Long Beech, Ca, AMS.

Christidis, Z., Jin Zhiyan, Wei Yu, Zhang Zhan, 1996: Optimization and Parallelization of the FSU Spectral Weather Forecasting Model on the IBM-SP, *Making its Mark*, G.-R. Hoffmann and N. Kreitz, eds. World Scientific, Signapore, pp 290-310.

Edwards J., J. S. Snook, and Z. Christidis, 1997: Forecasting for the 1996 Summer Olympic Games with the SMS-RAMS Parallel Model. Proceedings, *13th International Conference on Interactive and processing Systems*, Long Beech, Ca, AMS.

Guo Xiaorong, Yan Zhihui, Zhang Yuling and Chen Shoujun, 1989: BMC limited area model general description and operational results, Acta Meteor. Sin., 3: 108~118.

Yan Zhihui (1999), Forecast experiment for operational application of water loading prediction model, *Quarterly Journal of Applied Meteorology*, Vol.10, No.4: 453~461

Yan Zhihui, Guo Xiaorong, Xheng Guoan, 1996:. The limited area analysis and forecast system and its operational application. *Acta Meteorological Sinica*, 1996, 10(3): 295~308

Zhiyan J, Christidis Z., 1996: Parallel Implementation of the YH Limited Area Model on the SP2, *Making its Mark*, G.-R. Hoffmann and N. Kreitz, eds. World Scientific, Signapore, pp 290-310.