Redbooks Paper

Raj Pandian PhD

Chapter 1.

Benchmarking MOM4 on IBM @server pSeries clusters

Modular Ocean Model (MOM) is a popular code used for studying the global ocean climate system. MOM4 is the latest version of MOM. It is developed and supported by Geophysical Fluid Dynamics Laboratory (GFDL), Princeton University, New Jersey, USA. In this paper we present guidelines to benchmark MOM4 on an IBM® @server pSeries® cluster running Linux® and we include performance results.

.

How to find the source code

MOM4 source code is freely available and it can be downloaded from the Geophysical Fluid Dynamics Laboratory (GFDL) Web site: http://fms.gfdl.noaa.gov/. This site explains how to register and get approval from the administrator to download the source code and data files. For this report, we used MOM4_dec102004, which was the latest release when we started this project. MOM4 requires NetCDF and MPI libraries. Information about downloading the source code and building the NetCDF package is given in “Appendix A: Building NetCDF package” on page 6. The top_level directories in mom4 are:

- **bin**
- **doc**
- **exp**
- **src**

The directory **src** contains the source code, which is written in Fortran 90. 2D (latitudinal/longitudinal) domain decomposition and Message Passing Interface (MPI) are used for parallel processing.

Description of benchmark test cases

The directory **exp** holds the test cases: test1, test2, test3, test4, and test5. The complexity of the model configuration increases from test1 to test5. You’ll find a brief description of the test cases here: http://www.gfdl.noaa.gov/~fms/pubrel/k/mom4/doc/quickstart_guide.html

test1	Flat bottom sector model with simple physics.
test2	An extension of test1 with more physics options.
test3A	Demonstration of the open boundary condition capability of MOM4.
test3B	An extension of test3A.
test4	Global tripolar grid; coupled to GFDL sea ice model.
test5	An extension of test4 with more grid resolution and vertical levels; tracers include potential temperature, salinity, and age.

Compiling the source code

The directory **bin** has the makefile templates for different hardware platforms. The directory **exp** has the models: test1, test2, test3, test4 and test5.

For each test case, there is a script, run_mom4_test*, for compiling the source code and for running the model. Before executing this script, edit the template, mkmf.template.ibm, in the **bin** directory to change the compiler, compiler flags, loader flags, and paths to MPI and NetCDF libraries and include files. For instance, on a p710 cluster (software**:** SLES 9.1 SP1 and xlf 9.1.01) at Poughkeepsie Benchmark Center, we used the following flags. Note: /opt/pokhpc/MPICH-GM/MPICH-GM_64/include and /vol/b5p050/netcdf/include are the locations for MPI and NetCDF include files, respectively.

- FC = xlf95
- LD = mpif90
- CC = mpicc

- FFLAGS = -qextname=flush -qsuffix=f=f90:cpp=F90 -q64 -qtune=auto -qarch=auto -qmaxmem=-1 -qdpc -qrealsize=8 -qintsize=4 -O2 -qsource -I/opt/pokhpc/MPICH-GM/MPICH-GM_64/include -I/vol/b5p050/netcdf/include
- CPPFLAGS = -I/opt/pokhpc/MPICH-GM/MPICH-GM_64/include
- CFLAGS = -q64 -O2 -qtune=auto -qarch=auto -qmaxmem=-1
- LIBS = /vol/b5p050/netcdf/lib/libnetcdf.a
- LDFLAGS = -qextname=flush -q64 $(LIBS)

Now we present the details to build the executable for the benchmark case test5. The ideas can be applied to other cases as well.

After making the above changes in the template, mkmf.template.ibm, we modified the script run_mom4_test5 to set the platform to ibm and cppDefs to:

```
( "-WF,-Duse_netCDF,-Duse_libMPI,-D__aix" )
```

Also, for compiling mppnccombine.c, we changed the paths to the netcdf library and include files:

```
cc -O -o $mppnccombine -I/usr/local/include -I/vol/b5p050/netcdf/include
-L/usr/local/lib $code_dir/postprocessing/mppnccombine.c -lnetcdf
-L/vol/b5p050/netcdf/lib
```

The script run_mom4_test5 will create a subdirectory **exec** in ./mom4/exp/test5, produce a Makefile, compile the source code and build the executable fms_test5.x in **exec.**

Note 1:

We also tried the higher level compiler optimization flag –O3 with –qstrict for which the performance was about the same as –O2. See **Note 2** below for further discussion**.**

Input data for the test cases

The directory input has input files such as diag_table, field_table and namelist. Each test case has its own preprocessing data that should be in the preprocessing directory for that test case.

For example, for test5, create the directory preprocessing in ./mom4/exp/test5/. Download the data file preprocessing.tar from the Web site http://fms.gfdl.noaa.gov. Unpack it in./mom4/exp/test5/preprocessing.

For test4 and test5, create a directory called data in ./mom4/exp/ and four subdirectories in data: ice, lad, omip_mpi, and omip_ncar

Download data from the directory Data on the GFDL Web site to the corresponding subdirectory in data. Note that the data files in **omip_ncar** require about 1 GB of space and the data in **omip_mpi** is not needed for running test5. Data in ASCII, HISTORY, and RESTART is not needed for running any of the test cases.

Running the models

In ./mom4/exp/test*, there is a script to build the executable and run the model test*. As an example, for the model test5, the script, *run_mom4_test5*, is in ./mom4/exp/test5.

In addition to the changes given in the section “Compiling the source code” on page 2, set *npes* to the number of mpi tasks in the script *run_mom4_test*.* Also, depending on the benchmark requirement, change the following.

- set days = the length of the run in days
- set months = the length of the run in months

As an example, for this benchmark, we set days = 10 and months = 0.

Also, to write output at the end of 10 days instead of at the default 1 day interval, we replaced 1 by 10 in the following lines in ./mom4/exp/test5/input/diag_table.

- "ice_month", 1, "days", 1, "days", "time"
- "ocean_month", 1, "days", 1, "days", "time"
- "ocean_snap", -1, "days", 1, "days", "time"

After making the above changes, run the script *run_mom4_test5,* which will do the following.

- Compile the source code and build the executable fms_test5.x in the subdirectory exec.
- Create a working directory workdir in test5.
- Copy the input data files from test5/input, test5/preprocessing and exp/data to workdir. Copy the executable from exec to workdir.
- Run the executable in workdir using the command mpirun.

Output files

MOM4 output files will be in the working directory **workdir**.

The stdout file is 19590101.fms.out. The timing for the whole program as well as for the internal modules and the checksums for the restart files are provided at the bottom of this file. One should make sure the checksums are identical in all the runs.

Note 2:

The checksums are sensitive to the compiler optimization used. As an example, in our experiment, the checksums were different for the options –O3 (with –qstrict) and –O2, but the performance was about the same. Further, the checksums with –O2 agreed with the checksums obtained without using any optimization flag. So we chose –O2 instead of –O3 (with –qstrict) for the purposes of this report.

Troubleshooting

In this section, we present the problems encountered in running MOM4 on an IBM eServer™ pSeries cluster running Linux and the methods we used to solve them.

First, the program stopped with the following message.

```
FATAL from PE    0: diag_manager,init_output_field: file ocean_month is NOT found
in diag_table
```

We traced this message to an initialization routine src/shared/diag_manager/diag_manager.f90, and resolved the problem by recompiling that routine without the optimization flag –O2.

Next, when we ran the program using eight MPI tasks, the execution stopped with segmentation fault after reading the input data and initialization. The problem was not

because of ulimit -stack size, which was already set to unlimited. We debugged the code using gdb and found that the failure was due to stack corruption. The global arrays in MOM4 are allocated dynamically, but there are lots of local static arrays that caused stack overflow. a problem known to corrupt the stack. We identified the routines that caused the failure and changed the big static local arrays to dynamic arrays in those routines. Here is a list of the routines where we made the changes:

```
src/ice_sis/ice_model.f90
src/mom4/ocean_diag/ocean_adv_vel_diag.F90
src/shared/time_interp/time_interp_external.F90
```

Performance results

Table 1 shows the performance results for the model test5 on an OpenPOWER 720 cluster with SLES 9 Linux from SUSE, PBS Torque batch scheduler, IBM compiler XLF 9.1, and Myrinet interconnect with MPICH-GM-1.2.6 for MPI. Each node has four POWER5™ processors rated at 1.65 GHz and 8 GB of memory. We note that two MPI tasks per node gave better performance than four MPI tasks per node.

Table 1 Performance results on a p720 cluster with Myrinet interconnect.

Number of nodes	MPI tasks	Time_in_secs
4	8	2073
8	16	1601
10	20	1499
12	24	1413
14	28	1420
16	32	1466

Summary

In this paper, we presented the technical details to run MOM4 on an IBM eServer pSeries cluster running Linux. The elapsed time for the benchmark case test5 decreases as we increase the number of tasks up to 24 tasks on a p720 cluster. For more than 24 tasks, the problem size for each task gets small so that the communication time becomes more dominant than the computation time; hence, the performance degrades.

Author

Raj Pandian, Ph.D.
Application Specialist, HPC Benchmark Center
IBM Systems & Technology Group, Poughkeepsie
e-mail: pandian@us.ibm.com

Appendix A: Building NetCDF package

The source code and documentation for NetCDF package can be obtained from the Unidata Web site: http://my.unidata.ucar.edu/content/software/netcdf/index.html. Download the tar file netcdf.tar.gz for the current version. The version of the tar file used in this paper was 3.6.0-p1.

Here is a summary of the installation instructions.

- ☐ Set the environment variables for compilers and compiler flags.

  As an example, we used the following (valid for Bourne or Korn shell).

  - export CC=xlc
  - export FC=xlf
  - export F90=xlf90
  - export CXX=xlC
  - export CPPFLAGS=-D_IBMR2
  - export F90FLAGS="-O -q64 -qsuffix=f=f90 -qmaxmem=-1"
  - export CFLAGS="-O -q64 -qmaxmem=-1"
  - export CXXFLAGS=$CFLAGS
  - export FFLAGS=$CFLAGS
  - Unpack the tarball**:**

- ☐ gunzip netcdf.tar.gz

```
tar -xvf netcdf.tar
```

- ☐ To build the libraries, after changing to netcdf-3.6.0-p1/src, execute:

```
configure
make
make install
```

The libraries, libnetcdf.a and libnetcdf_c++.a, will be installed in ./netcdf-3.6.0-p1/lib and the utilities, ncdump and ncgen, will be installed in ./netcdf-3.6.0-p1/bin.

Note:

If something goes wrong, clean the leftover files before starting again. Issue:

```
make distclean
```

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:
This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

This document created or updated on April 5, 2006.

Send us your comments in one of the following ways:

- Use the online **Contact us** review redbook form found at:
  ibm.com/redbooks
- Send your comments in an email to:
  redbook@us.ibm.com
- Mail your comments to:
  IBM Corporation, International Technical Support Organization
  Dept. HYTD Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400 U.S.A.

Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

@server®	eServer™	POWER5™
@server®	pSeries®	Redbooks™
Redbooks (logo) ™	IBM®	

The following terms are trademarks of other companies:

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.