The Third Report of Teragrid Parallel WRF-HDF5 Performance

MuQun Yang prepared for Mike Folk
HDF group
April 15, 2004, the first draft
May 12,2004, revised

I. Introduction

This is the third report of Teragrid Parallel WRF-HDF5 Performance.

1. We are using SDSC TG machine to test the development version of WRF. The reason is that it was said that SDSC-TG GPFS is faster than NCSA-TG GPFS from an article at teragrid website (URL http://teragrid.ncsa.uiuc.edu/Doc/Data/rates.html#scp_alternative ; the contact person is Galen Arnold).

2. MPI compiler information:
Compiler: compiler icc, ifc 7.1,
MPICH: mpich version: mpich-gm-1.2.5..10-intel-r1

Compiling the new version of WRF is very hard at SDSC TG. I had to re-order modules inside source code to make the model compile. We believe that it is a compiler problem since it could be successfully compiled at NCSA IBM P690. The SDSC helpdesk staff also agreed with this point.

The big-endian to little-endian conversion at SDSC TG didn't work as I already reported before. So I had to use **titan** to generate the initial data for the model and move to SDSC TG to do the test. Since big output data seems always to improve parallel IO performance, I created the biggest experimental initial netcdf input file at titan; which is about 226MB. The size of the biggest array inside the file is 600*400*20=4,800,000.

3. Experiment Description:
Wall clock time of running the WRF model is used to investigate the IO performance.

In this experiment, we only investigate the output performance. So far, it is reasonable to believe that IO performance of the writer is more important since output data are much bigger when multi-timestep data are output. However, we cannot rule out the impact of a parallel reader to IO performance. With the current time and support, we will not test the performance of the parallel reader. The model reader in this experiment is always sequential NetCDF-WRF reader. What we want to compare is essentially Parallel WRF-HDF5 writer vs. Sequential

WRF-NetCDF writer. The computing module of the model is always in Parallel (MPI).

The experiment is set to be IO-bound. That is, at every timestep after the model computes, it will output the data at that timestep. The total number of timesteps is 145.

For both parallel HDF5 and Sequential NetCDF, we run the model with different number of nodes by using the same experiment setting. The number of node we choose is 4,8,16,32,64. Two processors are used for each node.

For each setting, we run three times and obtain the best of three.

II. Analysis and discussion

The attached figure shows the change of the wall-clock time of WRF model run in both Parallel HDF5 (solid line) and Sequential NetCDF(dashed line).

What we found:

1) We run three times for each setting in different number of processors, we find that under the same setting the variance of wall-clock time with the same number of processors is very small, which makes our performance result seem reliable.

2) With the same number of processors,
below 32 nodes, wall-clock time in parallel HDF5 IO module is less than that of sequential IO module.

At 64 nodes, wall-clock time in parallel HDF5 IO module is greater than that of sequential IO module.

Parallel HDF5-WRF has the smallest wall-clock time when the number of nodes is 8.

Explanations

1) Background

Wall-clock time consists of computing time, data input time, data output time.
For computing time, Since WRF is using MPI, computing time includes MPI communication time and real computing time.
For sequential NetCDF-WRF IO module, since all data in other processors will be transferred to one processor and written to the NetCDF file through that processor sequentially, MPI communication time should be the bottleneck. Furthermore, if the array is big enough, it may exceed the memory capacity and it will cause memory thrashing;

that will extremely deteriorate the performance. In our experiment, none of our dataset is that big. So the bottleneck is IO communication time transferring data from other processors to the processor used to write the data to the NetCDF file.
The sequential IO time is IO communication time plus real IO time.
For parallel HDF5-IO module, the IO time includes MPI-IO overhead plus real IO time. Each SDSC compute node is attached to the GPFS disk storage area network, so it is supposed that the configuration is fastest, so Real IO time in parallel IO is supposed to be shorter than one big sequential IO.

2) Results
With the same number of processors,
As expected, below 32 nodes, parallel HDF5 IO module uses less time than sequential NetCDF IO module.

However, two scenarios need to be investigated:

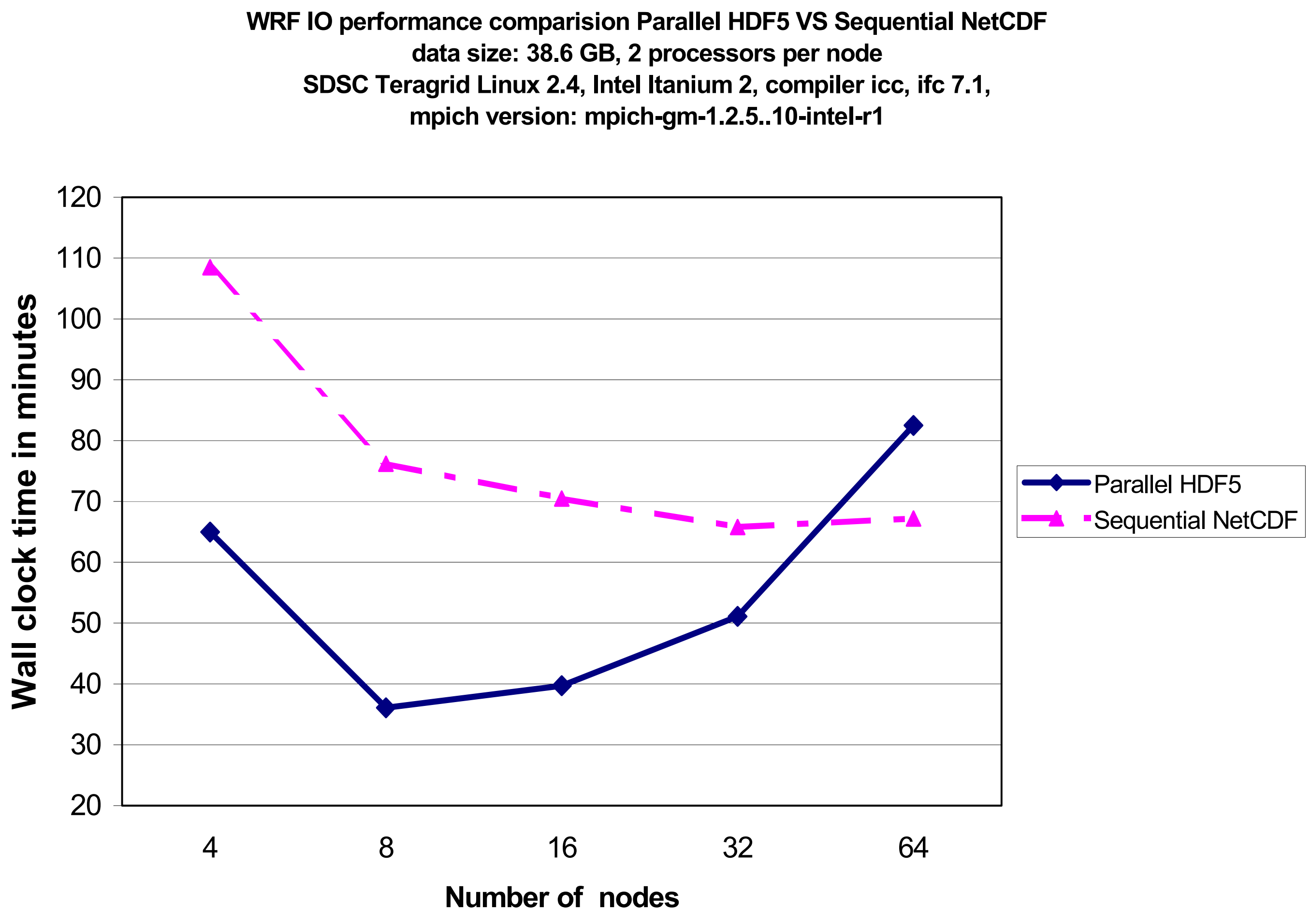
- Parallel HDF5 IO module is worse than sequential NetCDF IO module when the number of IO node is 64.
- The difference of wall clock time between parallel HDF5 IO module and sequential NetCDF IO module decreases with the increasing of the number of node when the number of node is greater than 8

From the dashed line, we can see that with the increasing of number of nodes, the wall clock time of sequential IO case decreases. We know that the only difference between parallel HDF5 IO module and sequential NetCDF IO module is the time to output the data. So the unexpected increasing of wall clock time used parallel HDF5 when the number of node is greater than 8 is in parallel IO layer.
Three factors may affect the performance in parallel IO layer:
1. Parallel HDF5
2. MPI-IO
3. Configuration of GPFS
We cannot test the third factor. We are still in the process of investigating the first and the second factors.
Future:
Of course, with more time and further funding support, we may do a through performance experiment with reader.

III. Experience with teragrid
Compared with NCSA's IBM machine, teragrid machine is hard to use.
IV. summary

- Parallel HDF5-WRF is better than sequential NetCDF-WRF when the number of nodes is less than or equal to 32.
- Parallel HDF5-WRF has the smallest wall-clock time when the number of nodes is 8.
- The performance of Parallel HDF5-WRF gets worse when the number of nodes is greater than 8.

See following figures:


WRF IO performance comparision Parallel HDF5 VS Sequential NetCDF
data size: 38.6 GB, 2 processors per node
SDSC Teragrid Linux 2.4, Intel Itanium 2, compiler icc, ifc 7.1,
mpich version: mpich-gm-1.2.5..10-intel-r1
Wall clock time in minutes
Number of nodes
Parallel HDF5
Sequential NetCDF