An HDF5-WRF module

-A performance report

MuQun Yang, Robert E. McGrath, Mike Folk

National Center for Supercomputing Applications
University of Illinois, Urbana-Champaign

ymuqun@ncsa.uiuc.edu

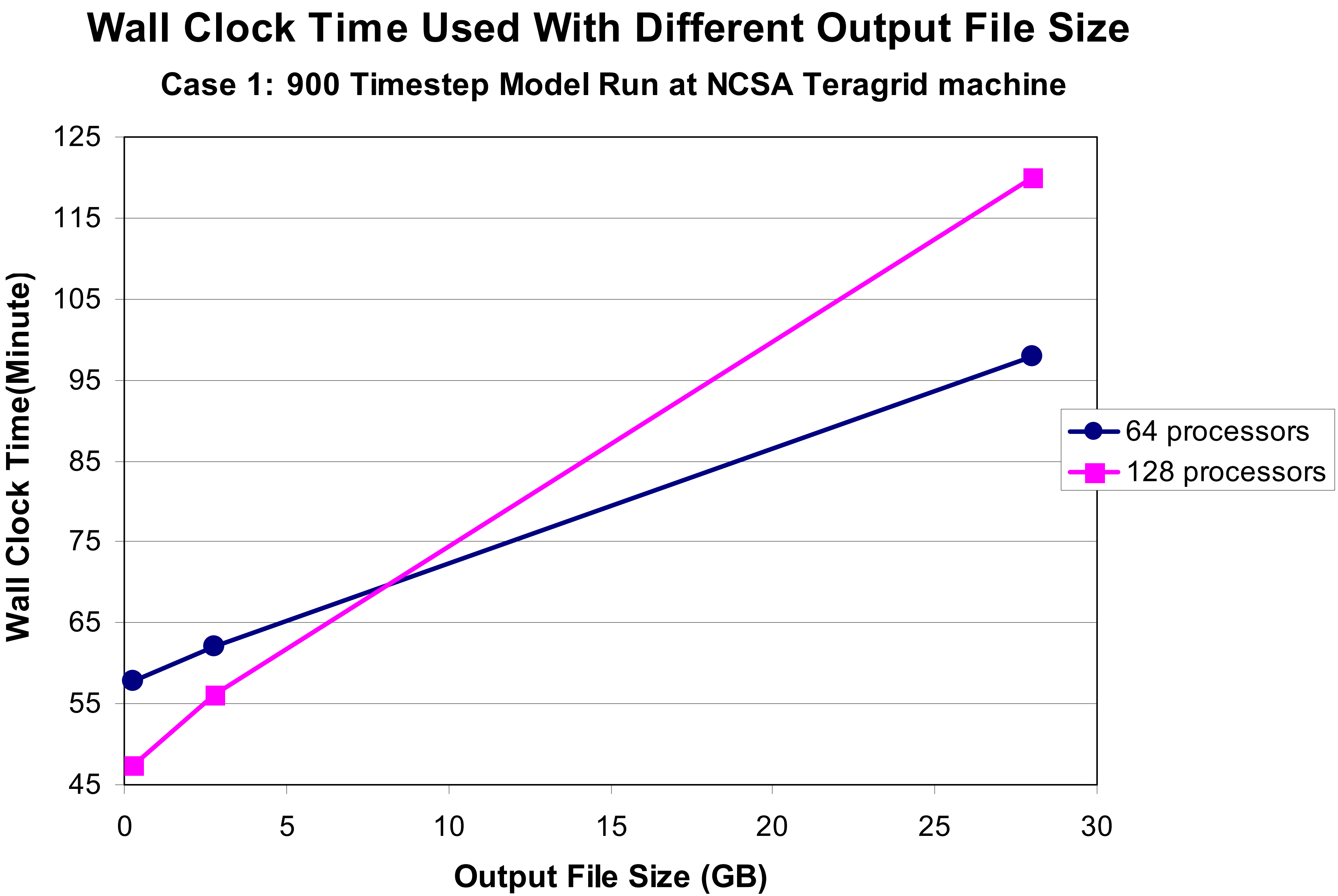
URL*: **http://hdf.ncsa.uiuc.edu/apps/WRF-ROMS/***

What is WRF?

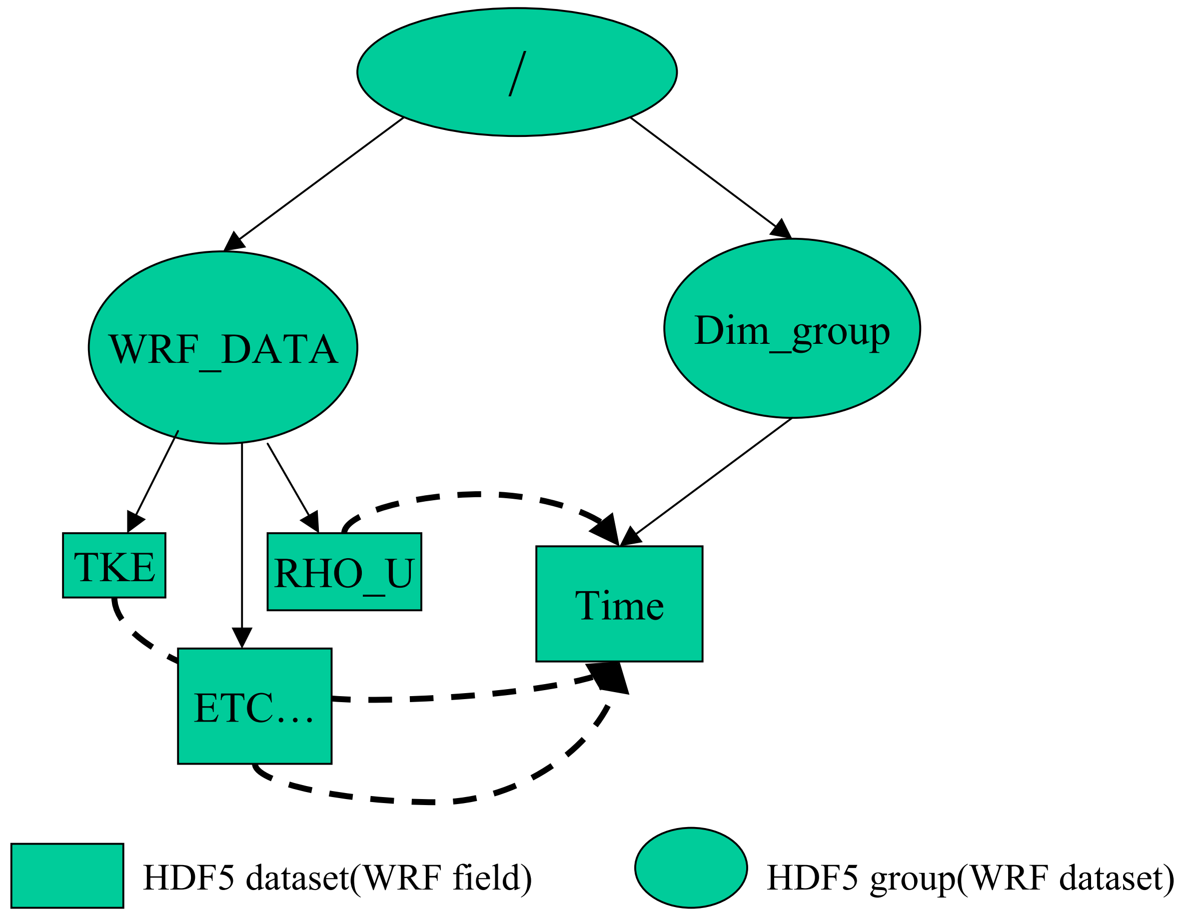
- Weather Research Forecasting Model
- Support MPI and OpenMP on parallel computing environment
- Sequential IO with NetCDF
- HDF5 able to describe model output

Why need HDF5-WRF module?

- Current NetCDF IO module becomes performance bottleneck for some WRF applications
- No compression feature is supported in the current NetCDF IO module although the output file size is large and compressible in some WRF applications


Wall Clock Time Used With Different Output File Size
Case 1: 900 Timestep Model Run at NCSA Teragrid machine
Wall Clock Time(Minute)
125
115
105
95
85
75
65
55
45
0
5
10
15
20
25
30
Output File Size (GB)
64 processors
128 processors

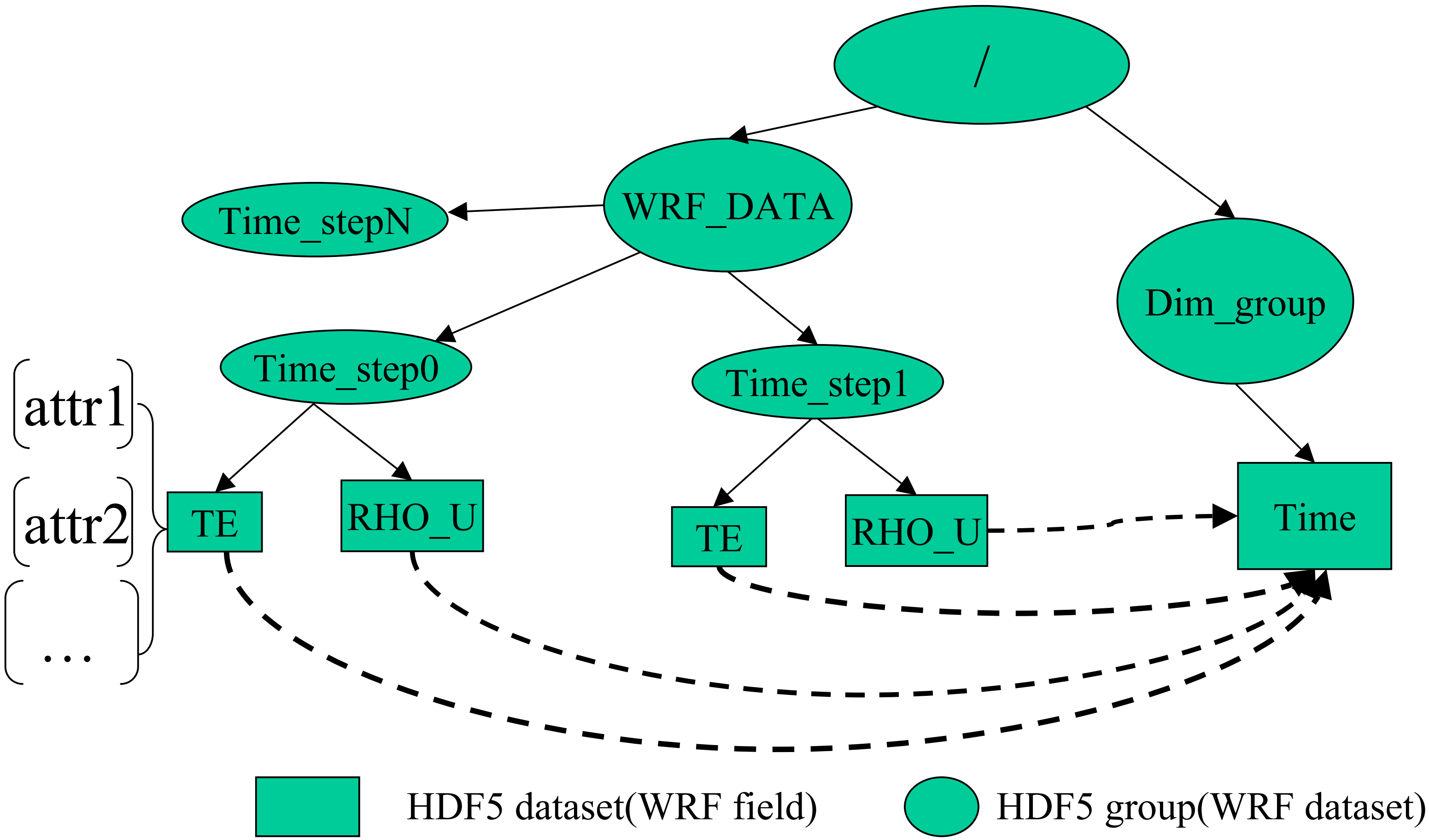
Schematic HDF5 File Structure of Sequential HDF5-WRF Output



Solid line: HDF5 datasets or sub-groups (the arrow points to) that are members of the HDF5 parent group.

Dash line: The association of one HDF5 object to another HDF5 object through dimensional scale table.

Schematic HDF5 File Structure of Parallel HDF5-WRF Output



Solid line: HDF5 datasets or sub-groups (the arrow points to) that are members of the HDF5 parent group.

Dash line: the association of one HDF5 object to another HDF5 object through dimensional table.

Methodologies

- Use real WRF applications to do performance comparisons between parallel HDF5-WRF module and NetCDF-WRF module; Wall clock time of the model run is used to measure the performance.
- Use real WRF applications to do performance comparisons between sequential HDF5-WRF module with compression features on and NetCDF-WRF module. File size and Wall clock time of the model run are used to measure the performance.

Methodologies (Continued)

Because:

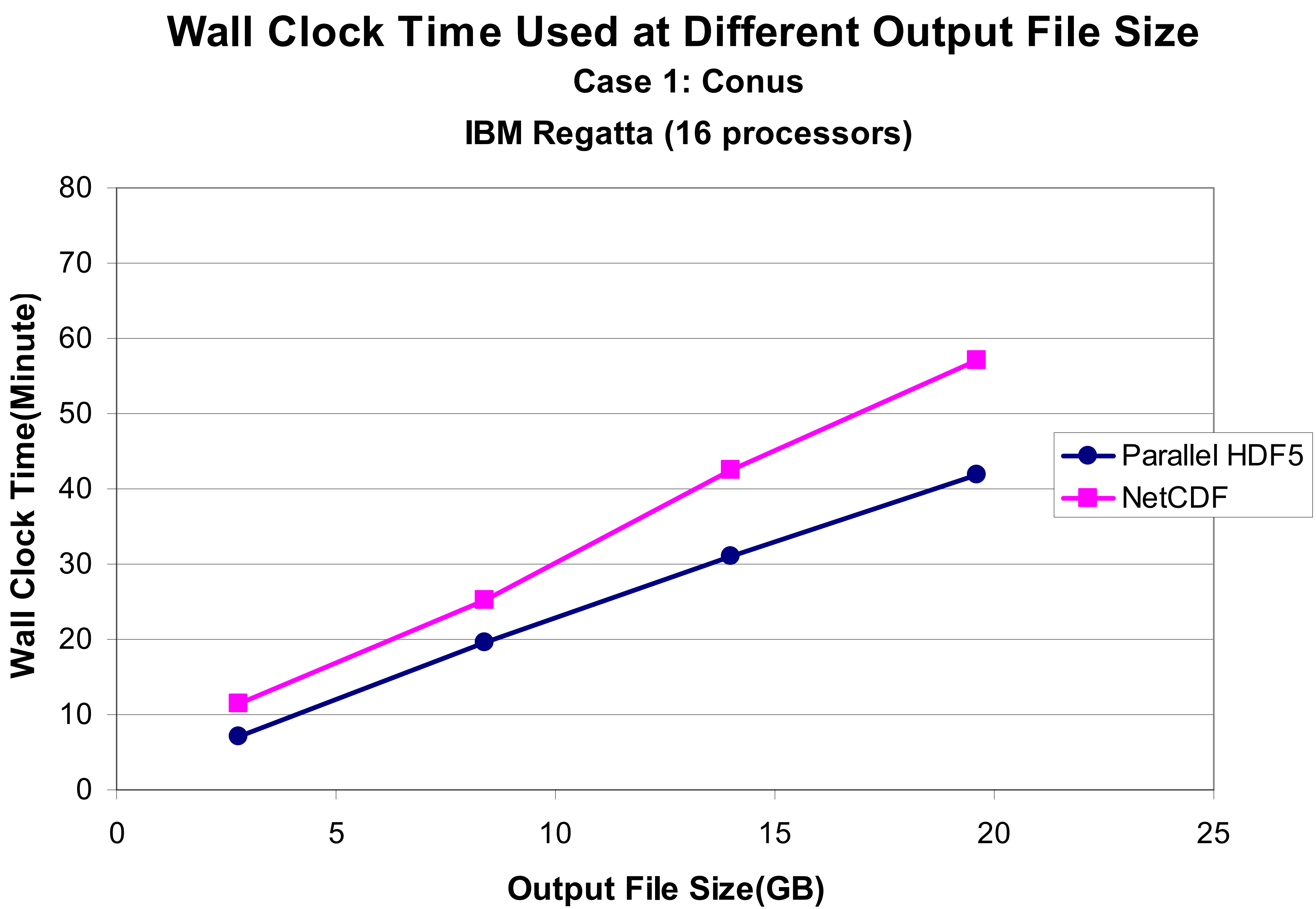
1) We want to only measure IO performance
2) Long model run is expensive
3) Model input is not a bottleneck so far

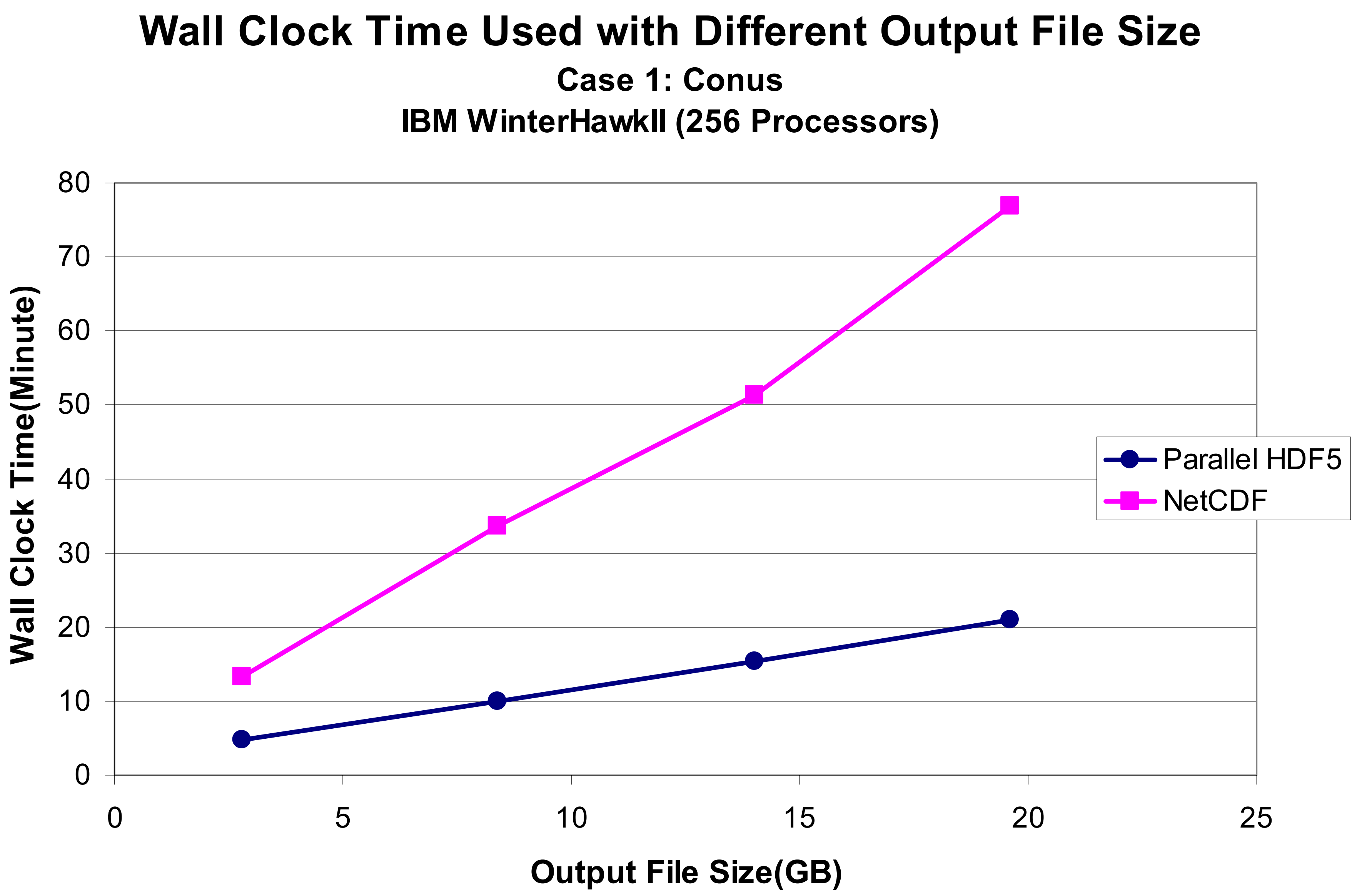
Therefore:

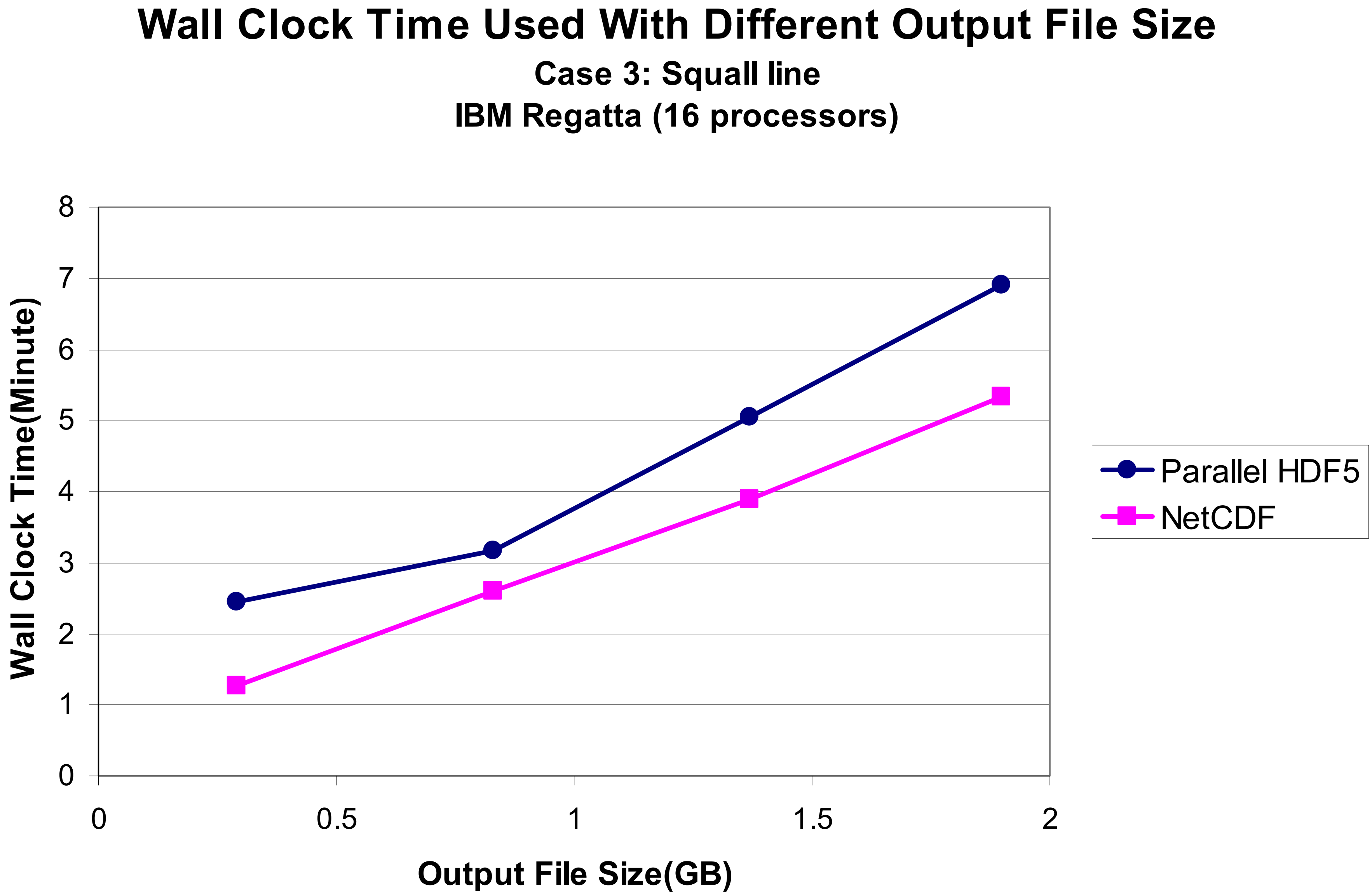
1) We focus on output performance
2) We force the model to write out data to output arrays at every time step; model outputs data at 10th,30th,50th,70th time step.

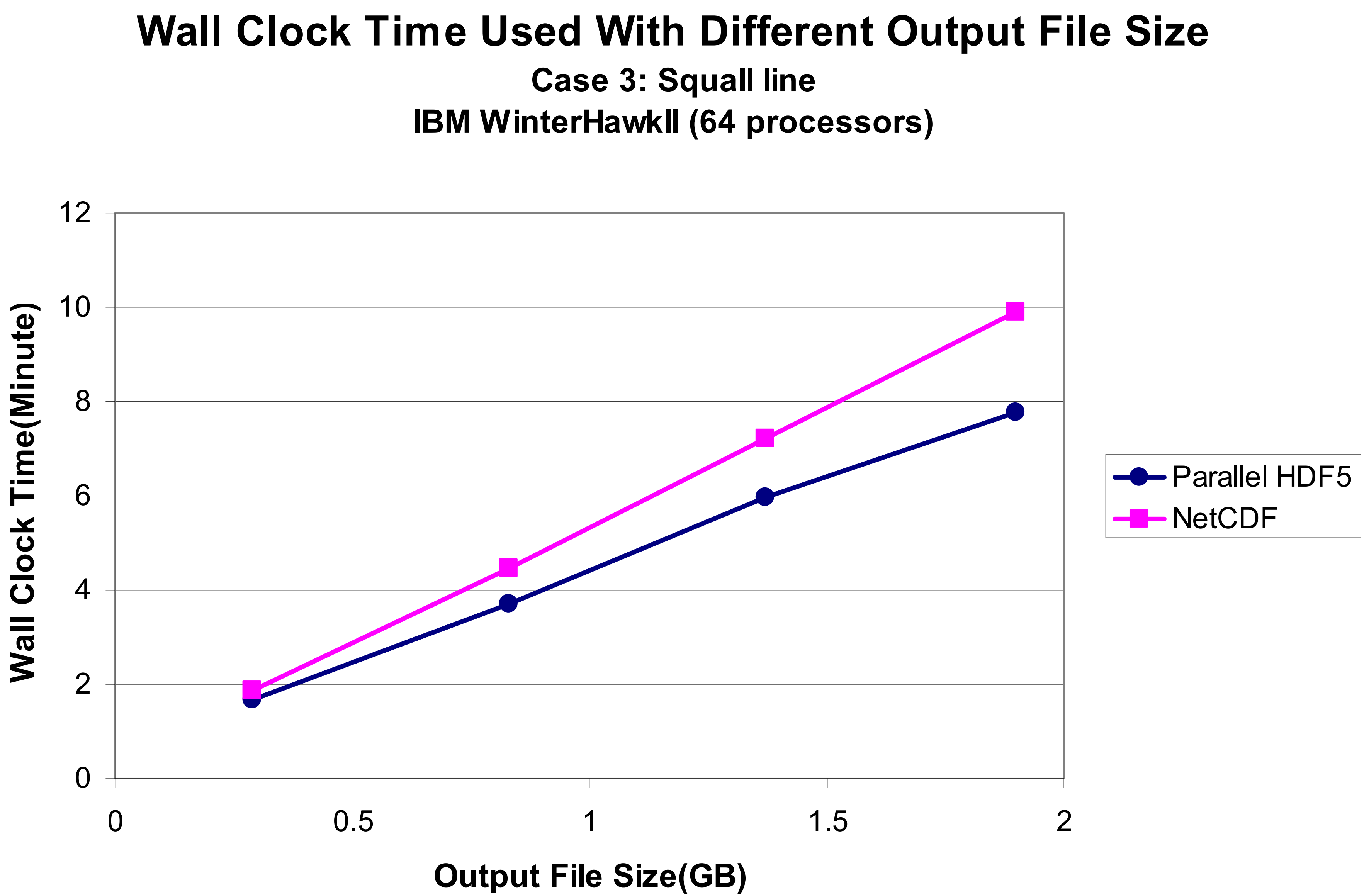
Case Descriptions

- Case 1: A real weather forecasting case from Oct. 24th, 2001 to Oct. 25th, 2001. The maximum array size is 17 MB per time step.
- Case 2: A real short-range climate forecasting case from May 2nd, 1993 to June 1st, 1993. The maximum array size is 3 MB per time step.
- Case 3: A squall-line simulation case; the maximum array size is 1.9 MB per time step.


Wall Clock Time Used at Different Output File Size
Case 1: Conus
IBM Regatta (16 processors)
Wall Clock Time(Minute)
0
10
20
30
40
50
60
70
80
0
5
10
15
20
25
Output File Size(GB)
Parallel HDF5
NetCDF


Wall Clock Time Used with Different Output File Size
Case 1: Conus
IBM WinterHawkII (256 Processors)
Wall Clock Time(Minute)
Output File Size(GB)
Parallel HDF5
NetCDF
0
10
20
30
40
50
60
70
80
0
5
10
15
20
25

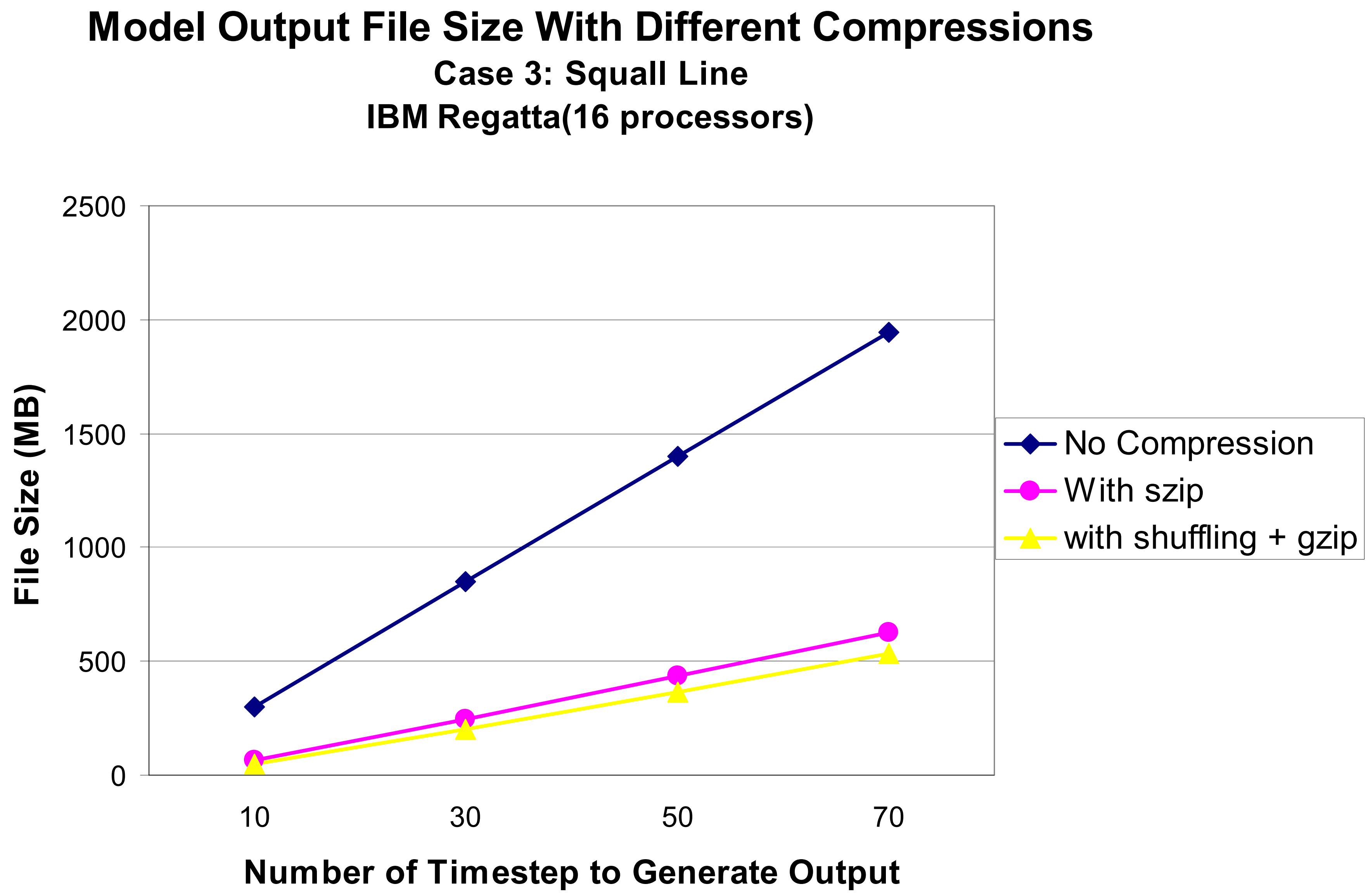

Wall Clock Time Used With Different Output File Size
Case 3: Squall line
IBM Regatta (16 processors)
Wall Clock Time(Minute)
0
1
2
3
4
5
6
7
8
0
0.5
1
1.5
2
Output File Size(GB)
Parallel HDF5
NetCDF

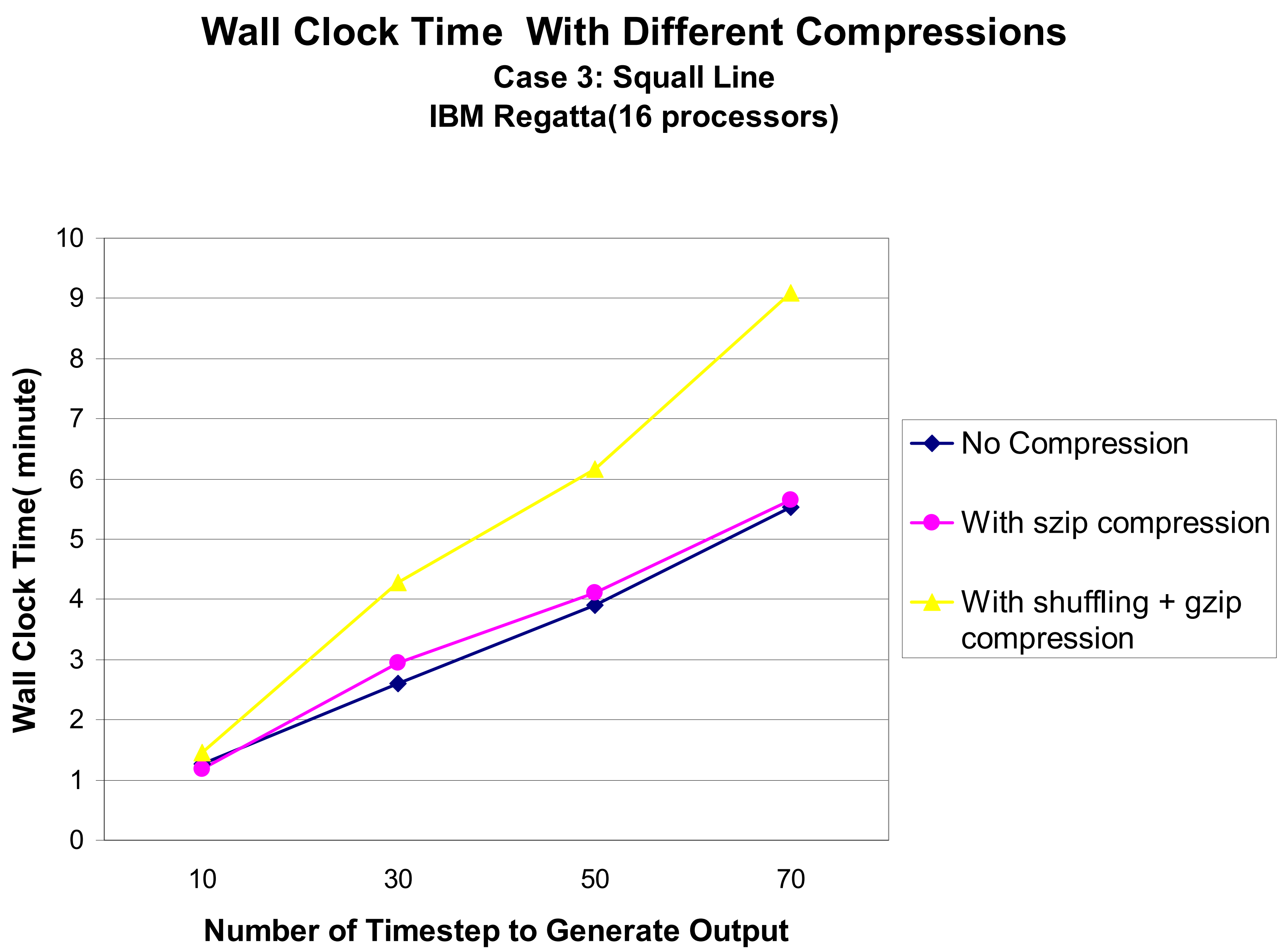


Wall Clock Time Used With Different Output File Size
Case 3: Squall line
IBM WinterHawkII (64 processors)
12
10
8
6
4
2
0
0
0.5
1
1.5
2
Wall Clock Time(Minute)
Output File Size(GB)
Parallel HDF5
NetCDF

Parallel HDF5-WRF Performance Analysis

- Parallel HDF5-WRF IO module can greatly improve WRF IO performance for **some** WRF applications.
- Usually the larger the HDF5 dataset size is and the more processors the WRF can run, the better the IO performance can be achieved.
- In order to achieve the better performance, parallel file system has to be available.


Model Output File Size With Different Compressions
Case 3: Squall Line
IBM Regatta(16 processors)
File Size (MB)
0
500
1000
1500
2000
2500
10
30
50
70
Number of Timestep to Generate Output
No Compression
With szip
with shuffling + gzip



Wall Clock Time With Different Compressions
Case 3: Squall Line
IBM Regatta(16 processors)
Wall Clock Time(minute)
0
1
2
3
4
5
6
7
8
9
10
10
30
50
70
Number of Timestep to Generate Output
No Compression
With szip compression
With shuffling + gzip compression

Sequential HDF5-WRF Performance Analysis

- SZIP compression can greatly compress the WRF data with little encoding time overhead.
- Shuffling algorithm combined with deflate compression can greatly compress the WRF data with some encoding time overhead.

Lessons learned

- Parallel HDF5-WRF IO module will not always assure good performance.
- Good performance may be achieved based on the availability of parallel file system, good size of HDF5 dataset and number of processors to use.
- Currently parallel HDF5 with chunking storage does not work well in WRF application. Improvement of the performance of parallel HDF5 with chunking storage is the key to make parallel HDF5 library more usable.

Summary

- Parallel HDF5-WRF module can greatly reduce wall clock time in WRF applications
- Sequential HDF5-WRF module can greatly reduce the WRF file size with szip or shuffling and deflate compression algorithms
- Parallel HDF5 library needs to be improved for chunking storage

Acknowledgements

- Authors want to thank John Michalakes at National Center for Atmospheric Research for providing CONUS WRF initial data, Dr. xinzhong Liang at Illinois State Water Survey for the Climate WRF initial data, Dr. Brian Jewett at Dept. of Atmospheric Sciences at University of Illinois at Urbana-Champaign for the squall line WRF initial data.
- This work is part of NSF-funded Modeling Environment for Atmospheric Discovery Expedition.