# HDF5 Performance Testing with Complex Hyperslabs

**Rishi Rakesh Sinha**
**Robert E. McGrath**
**February 24, 2005**

## Purpose:

This was a series of tests designed to test the performance of the hyperslab reading algorithm with hyperslabs more complex than the usual tests.

## Tools used:

The tool used for this was *h5_hstest*, which is being developed for the purpose of testing correctness of the hyperslab reading code. The source and run scripts for this experiment are available at:
ftp://hdf.ncsa.uiuc.edu/pub/outgoing/h5_hstest

## Setup:

The various versions and platforms on which the tests were run are shown in Table 1.

**Table 1. Versions of HDF for the two platforms**

| Platform | 5.14 | 5.16 | 5.17 |
|---|---|---|---|
| Heping (Linux) | 5-1.4.5-post9 | 5-1.6.4-snap8 | 5-1.7.45 |
| Copper (AIX5.1) | 5-1.4.5-post9 | 5-1.6.4-snap7 | 5-1.7.45 |

See Appendix 1 for the settings for each case.

The hyperslabs selected for the test were generated by h5_hstest hyperslab test tool. For an n dimensional dataset the hyperslabs are generated as follows:

- For each dimension in the data set k randomly chosen hyperplanes are selected. The number k is a user defined parameter.
- This leads to the formation of an dimensional matrix of such hyperplanes. Every two consecutive hyperplanes in each dimension create a hyperslab.
- Each of the $k^n$ hyperslabs is selected as the part of the final hyperslab with a 0.5 probability.

This hyperslab generation algorithm is used to generate two different types of hyperslabs.

- Simple Hyperslab: In this type of hyperslab, the value of k chosen is 2. This causes generation of only rectangular hyperslabs. This is tested only on copper. The details of the no of bytes of data transferred are given along with the experiment results.
- Complex Hyperslab: In this type of hyperslab, the value of k chosen is 20. This leads to formation of hyperslabs that are not simple rectangles but are of arbitrarily complicated shape. These hyperslabs were tested on heping and copper.

As in the case of simple hyperslabs the number of bytes of data transferred is given along with the experimental results.

For each of the type of hyperslab, we had three different types of data sets. Table 2 describes the three cases.

**Table 2. The test datasets (generated by *h5hstest*)**

| Case | Dataset Description |
|---|---|
| Small Datasets, Many chunks | Rank = 2,<br>Dim = 10000 * 3000,<br>Chunk size = 50 * 50<br>Compression (when used) – gzip, factor 5 |
| Normal Datasets | Rank = 2,<br>Dim = 100000 * 3000<br>Chunk Size = 500 * 500<br>Compression (when used) – gzip, factor 5 |
| Small Dataset Few Chunks | Rank = 2,<br>Dim = 10000 * 3000,<br>Chunk size = 500 * 500<br>Compression (when used) – gzip, factor 5 |

For each dataset, 3 different configurations

- Storing datasets as a contiguous block
- Storing datasets in various sized chunks.
- Storing datasets in chunks and compressing the chunks.

The time elapsed has been measured by using the time system call in the c program. The time does not include the time it takes to create and write a file. It include

- Time to open a file to Read.
- Time to generate the hyperslabs to be read.
- Time to actually read the hyperslabs.

For each of the experimental setups there were 100 different hyperslab generated in each run and these were sequentially read and checked for consistency.

**Results**

The first set of results shown are of simple hyperslabs and then the experimental results for complex hyperslabs are shown.

**For Simple Hyperslabs**
**Runs on Copper (AIX5.1)**
For following test the total number of bytes transferred is 266 MB.

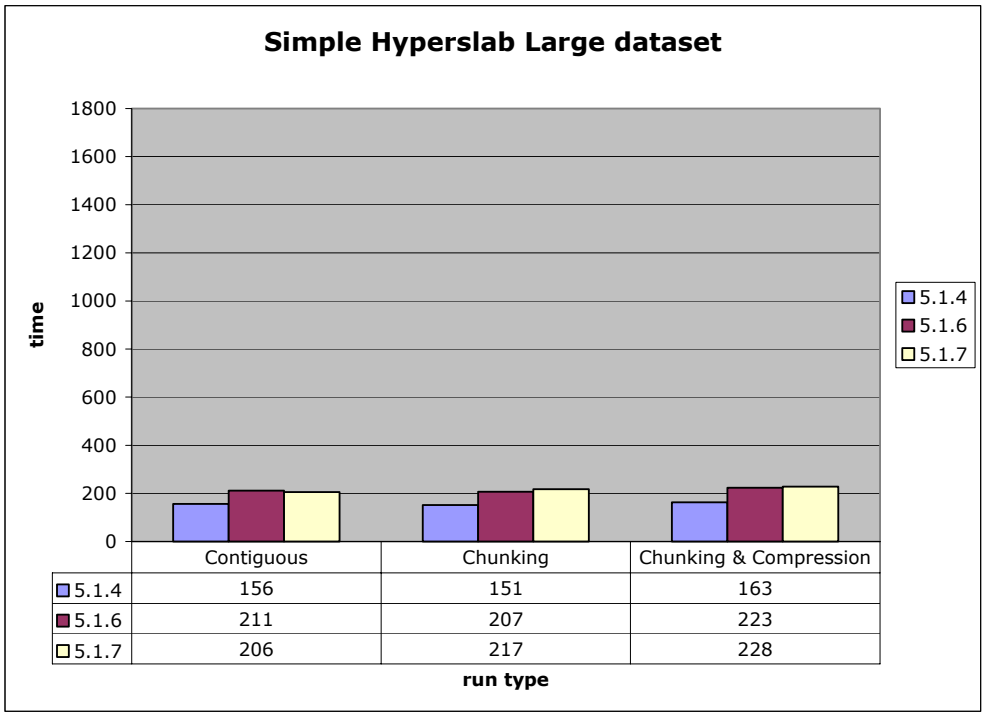



**Figure 1. Elapsed time (seconds) to read simple hyperslab.**

For the following test the number of bytes transferred is 257 MB.

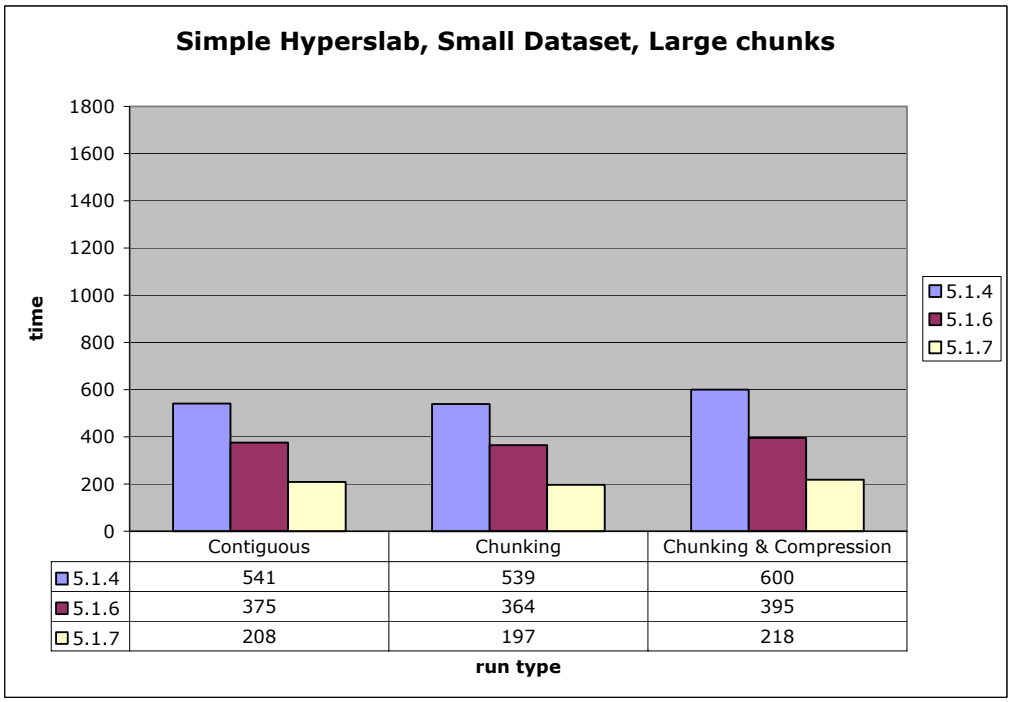



**Figure 2. Elapsed time (seconds) to read simple hyperslab.**

For the following test the number of bytes transferred is 257 MB.

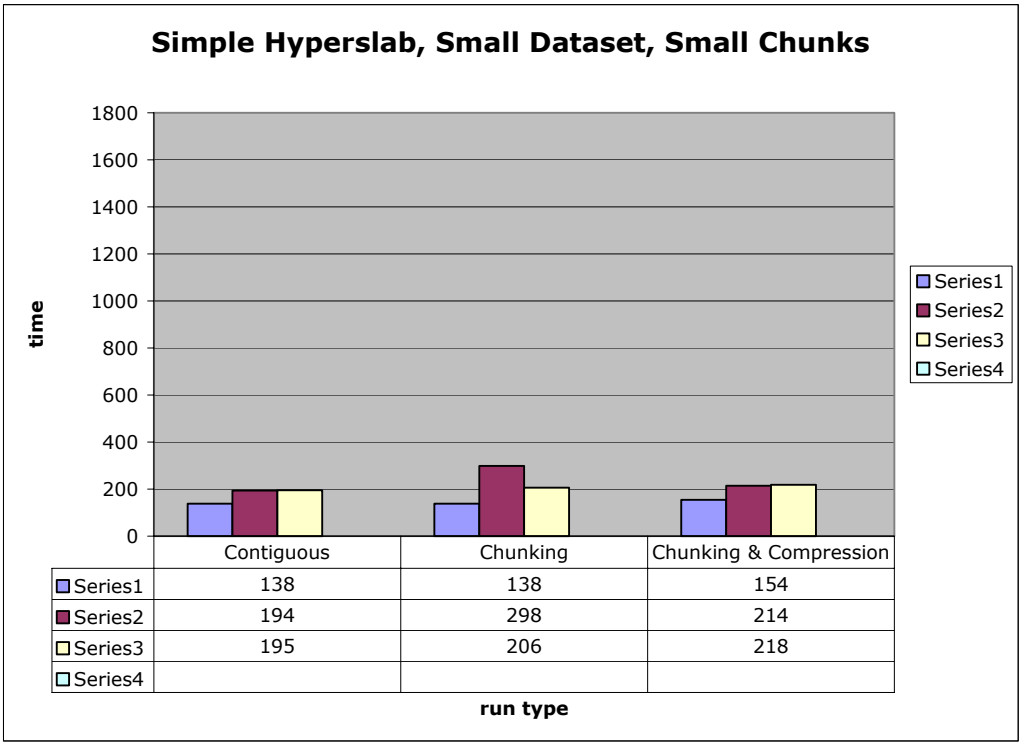



**Figure 3. Elapsed time (seconds) to read a simple hyperslab.**

## Reading complex hyperslab

## Runs on heping (Linux)

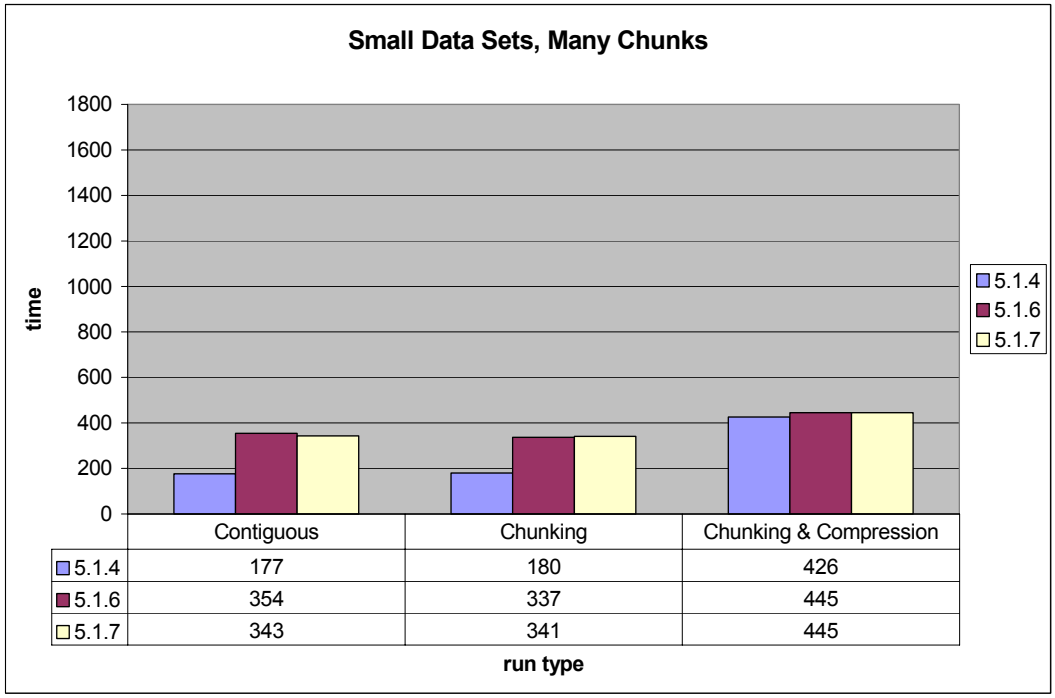



**Figure 4. Elapsed time (seconds) to read complex hyperslab.**

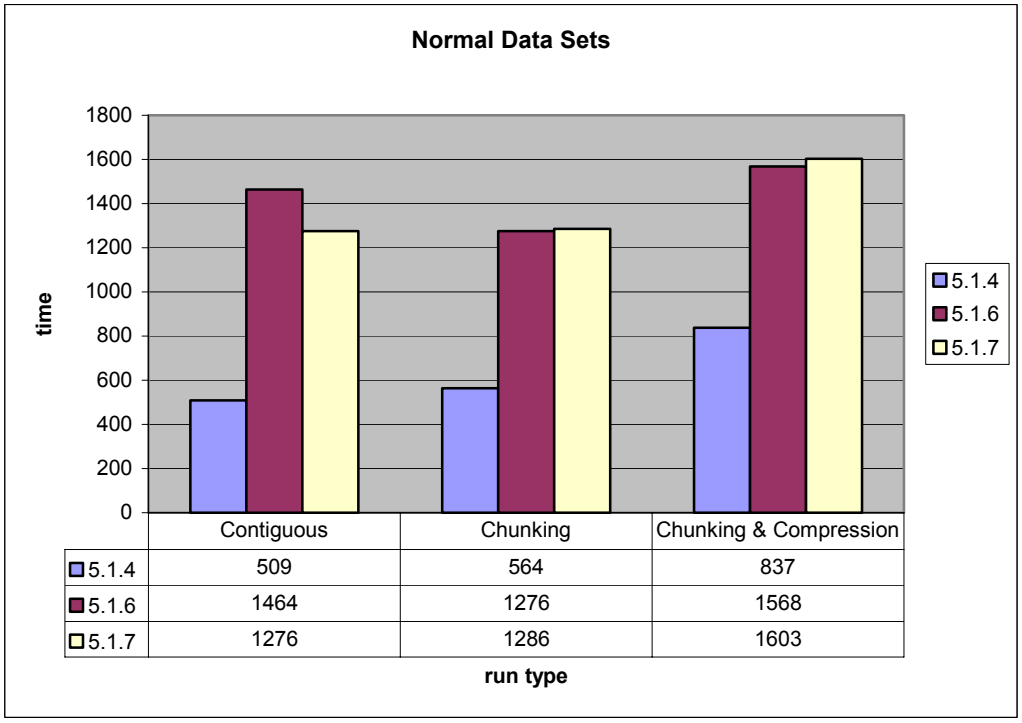



**Figure 5. Elapsed time (seconds) to read complex hyperslab.**

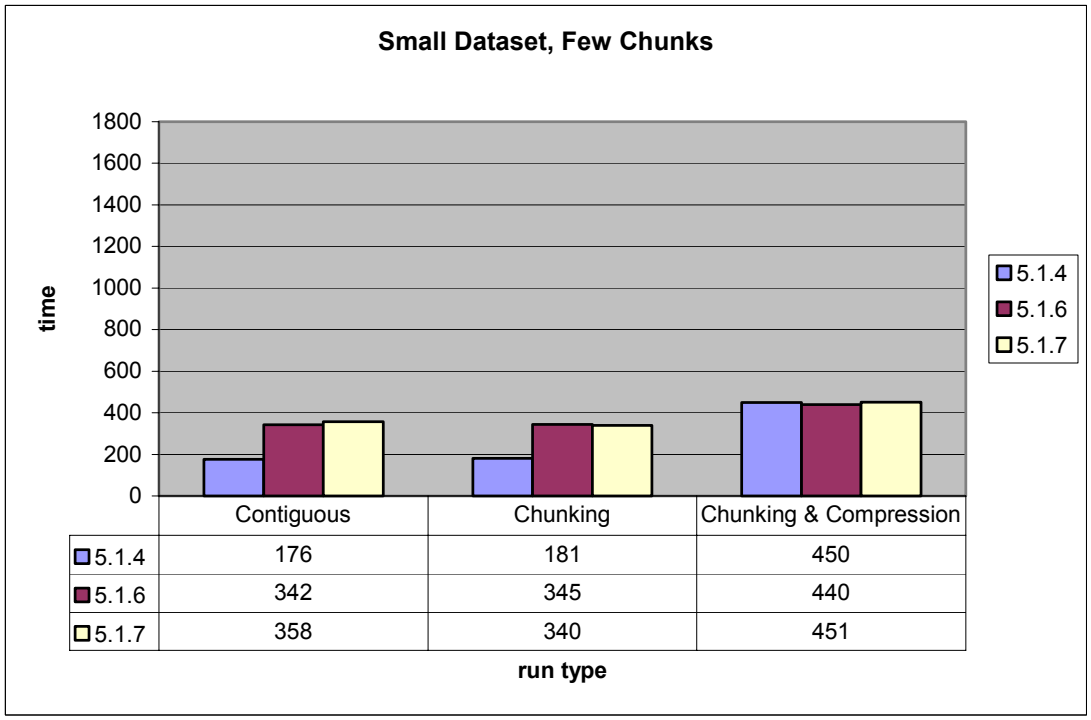
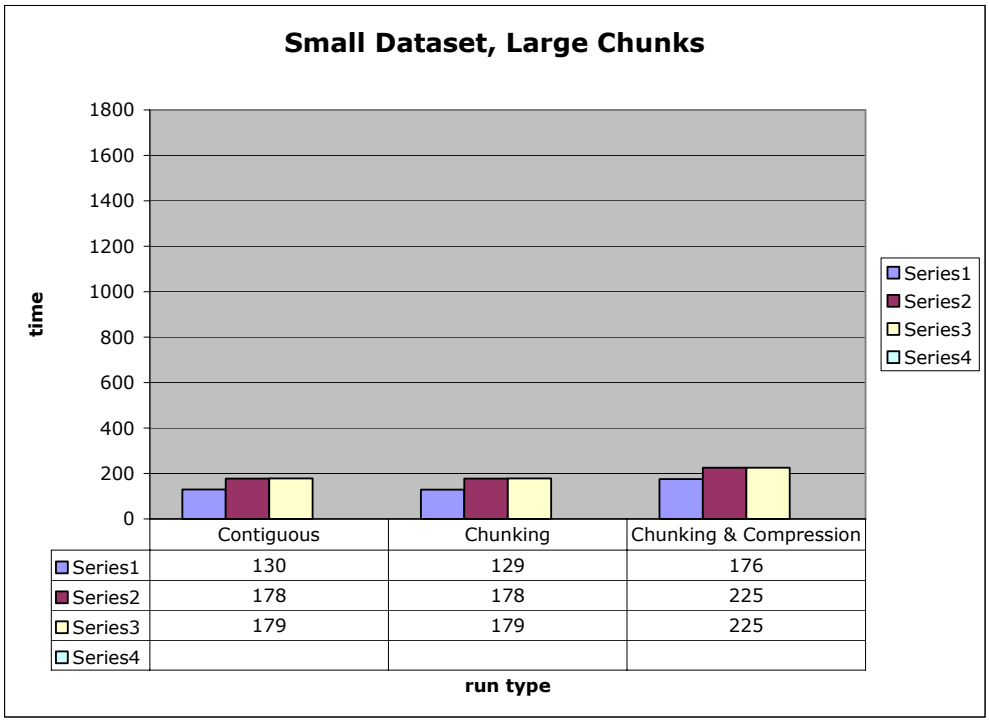



**Figure 6. Elapsed time (seconds) to read complex hyperslab.**

## Runs on Copper (AIX5.1)

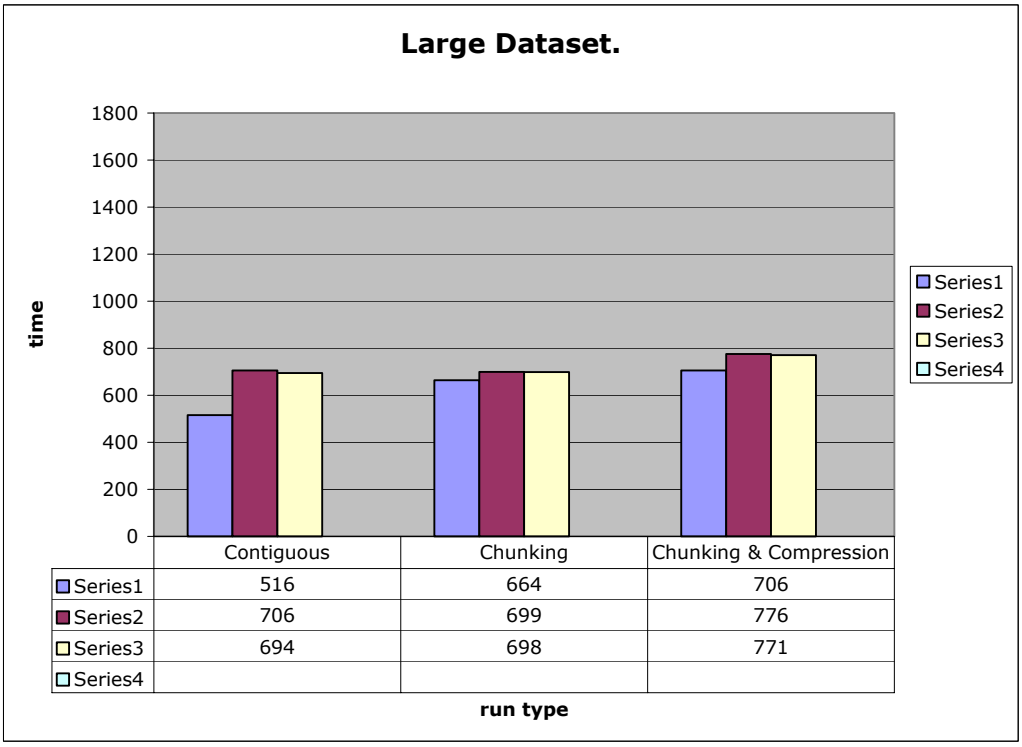
The number of bytes of data transferred is 1.07 GB.




| | Contiguous | Chunking | Chunking & Compression |
|---|---|---|---|
| Series1 | 130 | 129 | 176 |
| Series2 | 178 | 178 | 225 |
| Series3 | 179 | 179 | 225 |
| Series4 | | | |

**Figure 7. Elapsed time (seconds) to read complex hyperslab.**

The number of bytes of data transferred is 3.25 GB.




| | Contiguous | Chunking | Chunking & Compression |
|---|---|---|---|
| Series1 | 516 | 664 | 706 |
| Series2 | 706 | 699 | 776 |
| Series3 | 694 | 698 | 771 |
| Series4 | | | |

**Figure 8. Elapsed time (seconds) to read complex hyperslab.**

The number of bytes of data transferred is 1.07 GB.

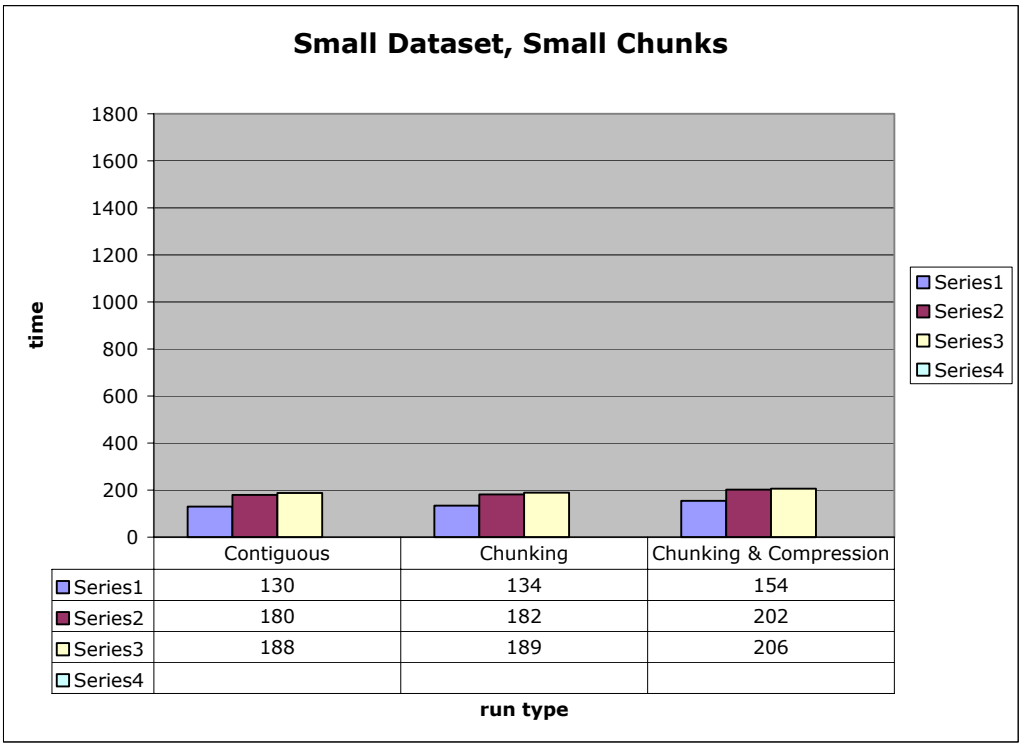



**Figure 9. Elapsed time (seconds) to read complex hyperslab.**

**Analysis:**

While for 5.1.6 chunking seems to improve performance, for 5.1.7 performance goes down with chunking. In both the cases reading the compressed data set is the most expensive. The results from the two platforms are not completely identical, the data does seem to indicate that chunking degrades performance for 5.1.7 remains.

The results for 5.1.4 shows that 5.1.6 and 5.1.7 is generally more than 100% slower than 5.1.4 on heping but on copper this number gets reduced to around 50%.

In the case of simple hyperslabs the performance improves for the later versions of HDF5. The data suggests that optimizations in the newer versions work best for simple large hyperslabs. Optimizations in the later versions may have degraded the performance of the more complex cases.

**Acknowledgements**

"This report is based upon work supported in part by a Cooperative Agreement with NASA under NASA grant NAG 5-2040 and NAG NCC5-599. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect

the views of the National Aeronautics and Space Administration.
Other support provided by NCSA and other sponsors and agencies
(http://hdf.ncsa.uiuc.edu/acknowledge.html)."

**Appendix 1**

SUMMARY OF THE HDF5.1.4 CONFIGURATION ON COPPER:
==========

HDF5 Version: 1.4.5-post9
Configured on: Wed Feb 16 09:32:11 CST 2005
Configured by: rsinha@Cu12
Configure mode: production
Host system: powerpc-ibm-aix5.1.0.0
Byte sex: big-endian
Libraries: static
Parallel support: no
Installation point: /u/ncsa/rsinha/5.1.4/
Compiler: /usr/vacpp/bin/xlc -q64
Compiler switches: -UH5_DEBUG_API -DNDEBUG -I/afs/ncsa.uiuc.edu/projects/hdf/packages/zlib/AIX5.1-64bit/include
Extra libraries: -L/afs/ncsa.uiuc.edu/projects/hdf/packages/zlib/AIX5.1-64bit/lib -lz -lm -lgpfs
Archiver: ar -X 64
Ranlib: ranlib
Debugged Packages:
API Tracing: no
File addresses: large
Configure Summary
Compiling Options:
Compilation Mode: Production
CFLAGS:
CPPFLAGS: -UH5_DEBUG_API -DNDEBUG -I/afs/ncsa.uiuc.edu/projects/hdf/packages/zlib/AIX5.1-64bit/include
LDFLAGS: -L/afs/ncsa.uiuc.edu/projects/hdf/packages/zlib/AIX5.1-64bit/lib
Debug Mode: None
Shared Libraries: No
Static Libraries: Yes
Statically Linked Executables: No
Tracing: No
Languages:
C++: No
Fortran: No
Features:

Async I/O in MPI-posix driver: No
dmalloc: No
GASS: No
GPFS: Yes
HDF5 v1.2 Compatibility: No
hsize_t: Large
Linux Large File Support (LFS): Disabled
Parallel HDF5: No
SRB: No
Stream VFD: Disabled
Threadsafety: Disabled
Zlib-compression: Yes

SUMMARY OF THE HDF5.1.6 CONFIGURATION ON COPPER:
==========

HDF5 Version: 1.6.4-snap8
Configured on: Wed Feb 16 10:13:38 CST 2005
Configured by: rsinha@Cu12
Configure mode: production
Host system: powerpc-ibm-aix5.1.0.0
Byte sex: big-endian
Libraries: static
Parallel support: no
Installation point: /u/ncsa/rsinha/5.1.6/
Compiler: /usr/vacpp/bin/xlc -q64
Compiler switches: -qlanglvl=ansi -D_LARGE_FILES -DSTDC -UH5_DEBUG_API -DNDEBUG -I/afs/ncsa.uiuc.edu/projects/hdf/packages/zlib/AIX5.1-64bit/include
Extra libraries: -L/afs/ncsa.uiuc.edu/projects/hdf/packages/zlib/AIX5.1-64bit/lib -lz -lm
Archiver: ar -X 64
Ranlib: ranlib
Debugged Packages:
API Tracing: no
File addresses: large
Configure Summary
Compiling Options:
Compilation Mode: Production
C Compiler: xlc -q64
CFLAGS: -qlanglvl=ansi -D_LARGE_FILES -DSTDC
CPPFLAGS: -UH5_DEBUG_API -DNDEBUG -I/afs/ncsa.uiuc.edu/projects/hdf/packages/zlib/AIX5.1-64bit/include

DFLAGS: -L/afs/ncsa.uiuc.edu/projects/hdf/packages/zlib/AIX5.1-64bit/lib
Debug Mode: None
Shared Libraries: No
Static Libraries: Yes
Statically Linked Executables: No
Tracing: No
Optimization Instrumentation: No
Languages:
C++: No
Fortran: No
Features:
dmalloc: No
Function Stack Tracing: Disabled
GASS: No
GPFS: No
HDF5 v1.4 Compatibility: No
hsize_t: Large
I/O filters (external): deflate
I/O filters (internal): shuffle,fletcher32
Linux Large File Support (LFS): Disabled
MPE: No
Parallel HDF5: No
SRB: No
Stream VFD: Enabled
Threadsafety: Disabled
High Level library: Disabled

SUMMARY OF THE HDF5.1.7 CONFIGURATION ON COPPER:
==========

HDF5 Version: 1.7.45
Configured on: Mon Jan 24 15:24:58 CST 2005
Configured by: rsinha@Cu12
Configure mode: development
Host system: powerpc-ibm-aix5.1.0.0
Byte sex: big-endian
Libraries: static
Parallel support: no
Installation point: /u/ncsa/rsinha/H5
Compiler: /usr/vacpp/bin/xlc -q64
Compiler switches: -qlanglvl=ansi -D_LARGE_FILES -DSTDC -g -qfullpath -DH5_DEBUG_API -DH5Z_DEBUG -DH5V_DEBUG -DH5T_DEBUG -DH5S_DEBUG -DH5P_DEBUG -DH5O_DEBUG -DH5MM_DEBUG -DH5I_DEBUG -DH5HG_DEBUG -DH5G_DEBUG -

DH5F_DEBUG -DH5E_DEBUG -DH5D_DEBUG -UNDEBUG -
I/afs/ncsa.uiuc.edu/projects/hdf/java/java4/mcgrath/copper/SZ-64bit/include -
I/afs/ncsa.uiuc.edu/projects/hdf/packages/zlib/AIX5.1-64bit/include
Extra libraries: -
L/afs/ncsa.uiuc.edu/projects/hdf/packages/zlib/AIX5.1-64bit/lib -
L/afs/ncsa.uiuc.edu/projects/hdf/java/java4/mcgrath/copper/SZ-64bit/lib -lsz -lz -lm
Archiver: ar
Ranlib: ranlib
Debugged Packages: d,e,f,g,hg,i,mm,o,p,s,t,v,z
API Tracing: yes
File addresses: large
Configure Summary
Compiling Options:
Compilation Mode: Development
C Compiler: xlc -q64
CFLAGS: -qlanglvl=ansi -D_LARGE_FILES -DSTDC -g -qfullpath
CPPFLAGS: -DH5_DEBUG_API -DH5Z_DEBUG -DH5V_DEBUG -
DH5T_DEBUG -DH5S_DEBUG -DH5P_DEBUG -DH5O_DEBUG -DH5MM_DEBUG
-DH5I_DEBUG -DH5HG_DEBUG -DH5G_DEBUG -DH5F_DEBUG -DH5E_DEBUG
-DH5D_DEBUG -UNDEBUG -
I/afs/ncsa.uiuc.edu/projects/hdf/java/java4/mcgrath/copper/SZ-64bit/include -
I/afs/ncsa.uiuc.edu/projects/hdf/packages/zlib/AIX5.1-64bit/include
LDFLAGS: -L/afs/ncsa.uiuc.edu/projects/hdf/packages/zlib/AIX5.1-
64bit/lib -L/afs/ncsa.uiuc.edu/projects/hdf/java/java4/mcgrath/copper/SZ-64bit/lib
Debug Mode: d,e,f,g,hg,i,mm,o,p,s,t,v,z
Shared Libraries: No
Static Libraries: Yes
Statically Linked Executables: No
Tracing: Yes
Optimization Instrumentation: Yes
Languages:
C++: No
Fortran: No
Features:
dmalloc: No
Flexible Parallel HDF: No
Function Stack Tracing: Enabled
GASS: No
GPFS: No
HDF5 v1.6 Compatibility: No
hsize_t: Large
I/O filters (external): deflate,szip(encoder)
I/O filters (internal): shuffle,fletcher32,nbit
Linux Large File Support (LFS): Disabled
MPE: No
Parallel HDF5: No

SRB: No
Stream VFD: Enabled
Threadsafety: Disabled
High Level library: Enabled

SUMMARY OF THE HDF5.1.4 CONFIGURATION ON HEPING
==========

HDF5 Version: 1.4.5-post9
Configured on: Mon Feb 14 09:25:33 CST 2005
Configured by: rsinha@heping
Configure mode: production
Host system: i686-pc-linux-gnu
Byte sex: little-endian
Libraries: static
Parallel support: no
Installation point: /mnt/sdt/rsinha/5.1.4
Compiler: /afs/ncsa/projects/hdf/packages/gcc-3.3.2/Linux_2.4/bin/gcc (gcc-3.3.2)
Compiler switches: -march=i686 -std=c99 -pedantic -Wall -W -Wundef -Wshadow -Wpointer-arith -Wbad-function-cast -Wcast-qual -Wcast-align -Wwrite-strings -Wconversion -Wsign-compare -Waggregate-return -Wstrict-prototypes -Wmissing-prototypes -Wmissing-declarations -Wredundant-decls -Wnested-externs -Winline -Wno-long-long -Wfloat-equal -Wmissing-format-attribute -Wpadded -O2 -fomit-frame-pointer -finline-functions -UH5_DEBUG_API -DNDEBUG -D_LARGEFILE_SOURCE -D_LARGEFILE64_SOURCE -D_FILE_OFFSET_BITS=64 -D_BSD_SOURCE
Extra libraries: -lz -lm
Archiver: ar
Ranlib: ranlib
Debugged Packages:
API Tracing: no
File addresses: large
Configure Summary
Compiling Options:
Compilation Mode: Production
CFLAGS: -march=i686 -std=c99 -pedantic -Wall -W -Wundef -Wshadow -Wpointer-arith -Wbad-function-cast -Wcast-qual -Wcast-align -Wwrite-strings -Wconversion -Wsign-compare -Waggregate-return -Wstrict-prototypes -Wmissing-prototypes -Wmissing-declarations -Wredundant-decls -Wnested-externs -Winline -Wno-long-long -Wfloat-equal -Wmissing-format-attribute -Wpadded -O2 -fomit-frame-pointer -finline-functions
CPPFLAGS: -UH5_DEBUG_API -DNDEBUG -D_LARGEFILE_SOURCE -D_LARGEFILE64_SOURCE -D_FILE_OFFSET_BITS=64 -D_BSD_SOURCE
LDFLAGS:
Debug Mode: None
Shared Libraries: No

Static Libraries: Yes
Statically Linked Executables: No
Tracing: No
Languages:
C++: No
Fortran: No
Features:
Async I/O in MPI-posix driver: No
dmalloc: No
GASS: No
GPFS: No
HDF5 v1.2 Compatibility: No
hsize_t: Large
Linux Large File Support (LFS): Enabled
Parallel HDF5: No
SRB: No
Stream VFD: Disabled
Threadsafety: Disabled
Zlib-compression: Yes

SUMMARY OF THE HDF5.1.6 CONFIGURATION ON HEPING
==========

HDF5 Version: 1.6.4-snap7
Configured on: Mon Feb 7 11:27:02 CST 2005
Configured by: rsinha@heping
Configure mode: production
Host system: i686-pc-linux-gnu
Byte sex: little-endian
Libraries: static
Parallel support: no
Installation point: /mnt/sdt/rsinha/5.1.6
Compiler: /afs/ncsa/projects/hdf/packages/gcc-3.3.2/Linux_2.4/bin/gcc (gcc-3.3.2)
Compiler switches: -march=i686 -std=c99 -pedantic -Wall -W -Wundef -Wshadow -Wpointer-arith -Wbad-function-cast -Wcast-qual -Wcast-align -Wwrite-strings -Wconversion -Wsign-compare -Waggregate-return -Wstrict-prototypes -Wmissing-prototypes -Wmissing-declarations -Wredundant-decls -Wnested-externs -Winline -Wno-long-long -Wfloat-equal -Wmissing-format-attribute -Wpadded -Wmissing-noreturn -Wpacked -Wdisabled-optimization -Wmultichar -Wendif-labels -O -fomit-frame-pointer -finline-functions -UH5_DEBUG_API -DNDEBUG -D_LARGEFILE_SOURCE -D_LARGEFILE64_SOURCE -D_FILE_OFFSET_BITS=64 -D_POSIX_SOURCE -D_BSD_SOURCE
Extra libraries: -lz -lm
Archiver: ar
Ranlib: ranlib

Debugged Packages:
API Tracing: no
File addresses: large
Configure Summary
Compiling Options:
Compilation Mode: Production
C Compiler: gcc
CFLAGS: -march=i686 -std=c99 -pedantic -Wall -W -Wundef -Wshadow -Wpointer-arith -Wbad-function-cast -Wcast-qual -Wcast-align -Wwrite-strings -Wconversion -Wsign-compare -Waggregate-return -Wstrict-prototypes -Wmissing-prototypes -Wmissing-declarations -Wredundant-decls -Wnested-externs -Winline -Wno-long-long -Wfloat-equal -Wmissing-format-attribute -Wpadded -Wmissing-noreturn -Wpacked -Wdisabled-optimization -Wmultichar -Wendif-labels -O -fomit-frame-pointer -finline-functions
CPPFLAGS: -UH5_DEBUG_API -DNDEBUG -D_LARGEFILE_SOURCE -D_LARGEFILE64_SOURCE -D_FILE_OFFSET_BITS=64 -D_POSIX_SOURCE -D_BSD_SOURCE
LDFLAGS:
Debug Mode: None
Shared Libraries: No
Static Libraries: Yes
Statically Linked Executables: No
Tracing: No
Optimization Instrumentation: No
Languages:
C++: No
Fortran: No
Features:
dmalloc: No
Function Stack Tracing: Disabled
GASS: No
GPFS: No
HDF5 v1.4 Compatibility: No
hsize_t: Large
I/O filters (external): deflate
I/O filters (internal): shuffle,fletcher32
Linux Large File Support (LFS): Enabled
MPE: No
Parallel HDF5: No
SRB: No
Stream VFD: Enabled
Threadsafety: Disabled
High Level library: Disabled

SUMMARY OF THE HDF5.1.7 CONFIGURATION ON HEPING

==========

HDF5 Version: 1.7.45
Configured on: Mon Feb 7 11:35:53 CST 2005
Configured by: rsinha@heping
Configure mode: production
Host system: i686-pc-linux-gnu
Byte sex: little-endian
Libraries: static
Parallel support: no
Installation point: /mnt/sdt/rsinha/5.1.7
Compiler: /afs/ncsa/projects/hdf/packages/gcc-3.3.2/Linux_2.4/bin/gcc (gcc-3.3.2)
Compiler switches: -march=i686 -std=c99 -pedantic -Wall -W -Wundef -Wshadow -Wpointer-arith -Wbad-function-cast -Wcast-qual -Wcast-align -Wwrite-strings -Wconversion -Wsign-compare -Waggregate-return -Wstrict-prototypes -Wmissing-prototypes -Wmissing-declarations -Wredundant-decls -Wnested-externs -Winline -Wno-long-long -Wfloat-equal -Wmissing-format-attribute -Wpadded -Wmissing-noreturn -Wpacked -Wdisabled-optimization -Wmultichar -Wendif-labels -O -fomit-frame-pointer -finline-functions -UH5_DEBUG_API -DNDEBUG -D_LARGEFILE_SOURCE -D_LARGEFILE64_SOURCE -D_FILE_OFFSET_BITS=64 -D_POSIX_SOURCE -D_BSD_SOURCE
Extra libraries: -lz -lm
Archiver: ar
Ranlib: ranlib
Debugged Packages:
API Tracing: no
File addresses: large
Configure Summary
Compiling Options:
Compilation Mode: Production
C Compiler: gcc
CFLAGS: -march=i686 -std=c99 -pedantic -Wall -W -Wundef -Wshadow -Wpointer-arith -Wbad-function-cast -Wcast-qual -Wcast-align -Wwrite-strings -Wconversion -Wsign-compare -Waggregate-return -Wstrict-prototypes -Wmissing-prototypes -Wmissing-declarations -Wredundant-decls -Wnested-externs -Winline -Wno-long-long -Wfloat-equal -Wmissing-format-attribute -Wpadded -Wmissing-noreturn -Wpacked -Wdisabled-optimization -Wmultichar -Wendif-labels -O -fomit-frame-pointer -finline-functions
CPPFLAGS: -UH5_DEBUG_API -DNDEBUG -D_LARGEFILE_SOURCE -D_LARGEFILE64_SOURCE -D_FILE_OFFSET_BITS=64 -D_POSIX_SOURCE -D_BSD_SOURCE
LDFLAGS:
Debug Mode: None
Shared Libraries: No
Static Libraries: Yes
Statically Linked Executables: No

Tracing: No
Optimization Instrumentation: No
Languages:
C++: No
Fortran: No
Features:
dmalloc: No
Flexible Parallel HDF: No
Function Stack Tracing: Disabled
GASS: No
GPFS: No
HDF5 v1.6 Compatibility: No
hsize_t: Large
I/O filters (external): deflate
I/O filters (internal): shuffle,fletcher32,nbit
Linux Large File Support (LFS): Enabled
MPE: No
Parallel HDF5: No
SRB: No
Stream VFD: Enabled
Threadsafety: Disabled
High Level library: Disabled