

FITSIO, HDF4, NetCDF, PDB and HDF5 Performance Some Benchmarks Results

Elena Pourmal

Science Data Processing Workshop

February 27, 2002

Benchmark Environment (software)

- Software
 - HDF4 r1.5
 - HDF5 1.4.3
 - NetCDF 3.5
 - FITSIO version 2.2
 - PDB version 8_7_01
 - ‘System’ benchmark uses **open**, **write**, **read** and **close** UNIX functions.
- each measurement was taken 10 times, best times were collected

Benchmark Environment (hardware)

- 2 - 550 Mhz Pentium III Xeon (Linux 2.2.18smp)
 - 1G memory
- NCSA O2K (IRIX64-6.5)
 - 195Mhz MIPS R10000
 - 14GB memory
 - Peak performance 390 MFLOPS

Benchmarks

- Creating and writing contiguous dataset; sizes vary from 2MB to 512MB
- Reading contiguous dataset; sizes vary from 2MB to 256MB
- Reading contiguous hyperslab; sizes vary from 1MB to 64MB
- Reading every second element of the hyperslab; sizes of selections vary from 0.25MB to 16MB
- Creating and writing up to 1000 1MB datasets; reading back the dataset created last

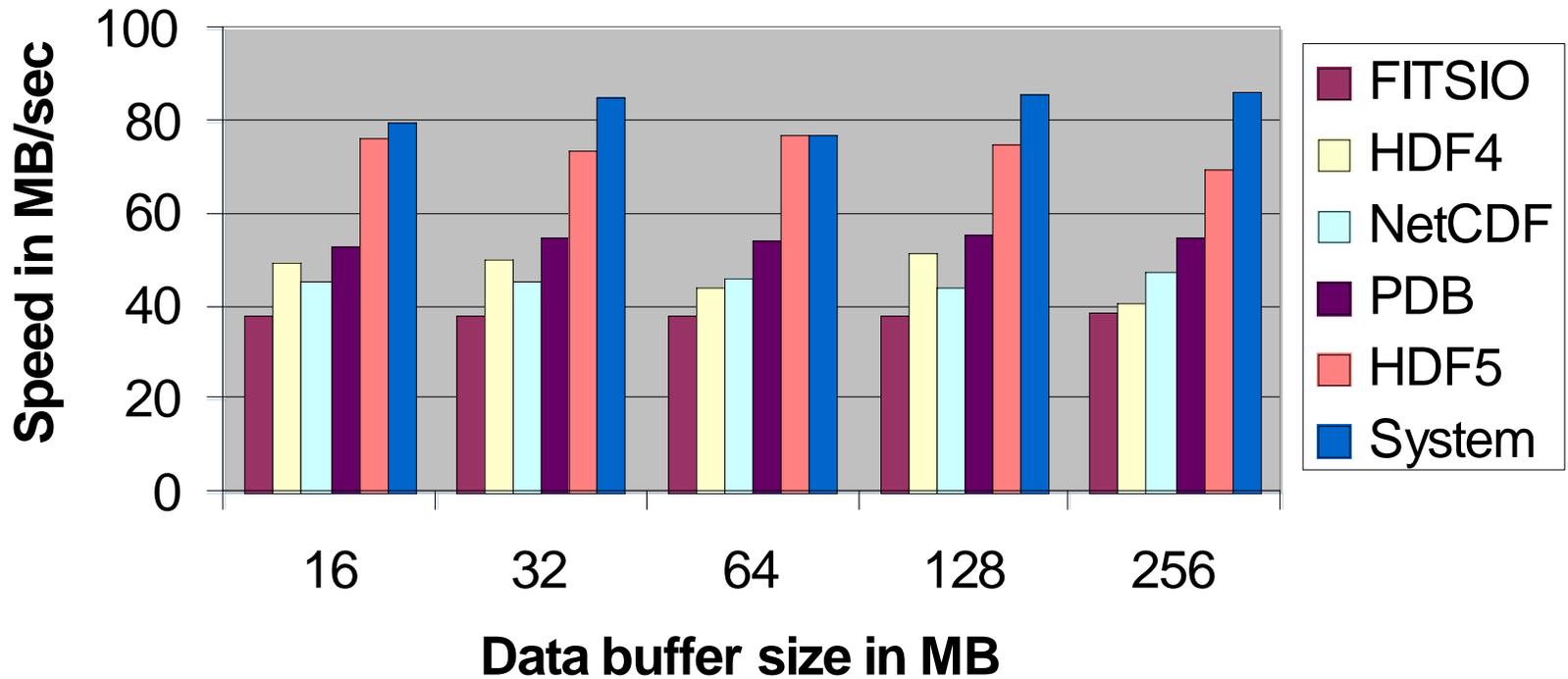
Some remarks

- “dataset” describes array stored in the FITS, HDF4, HDF5, PDB, NetCDF and UNIX binary files, i.e. “dataset” means
 - “primary array” and “extension” for FITSIO
 - “variable” for NetCDF
 - “SDS or scientific data set” for HDF4
 - “HDF5 dataset”
 - “PDB variable”
 - raw data stored in UNIX binary file
- **gettimeofday** function was used to measure time
- Speed calculated as data buffer size over time

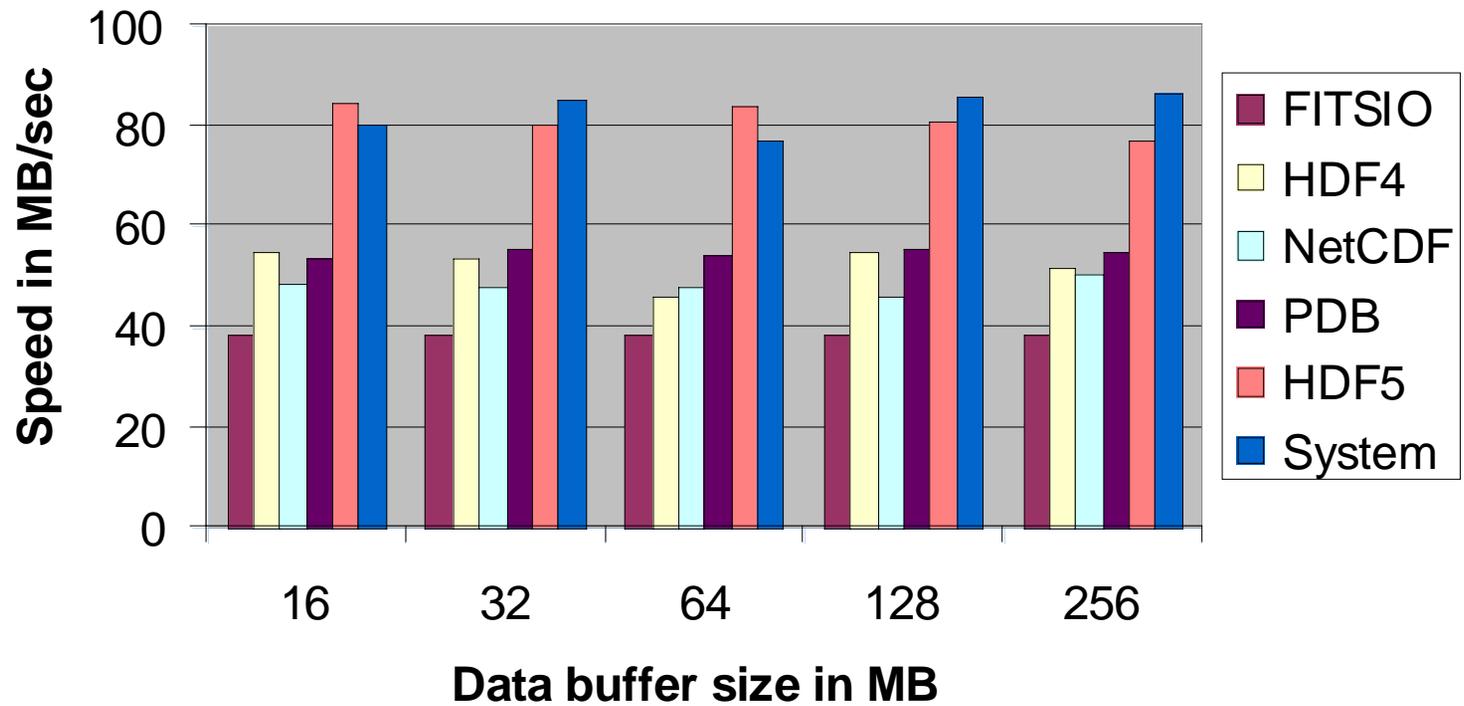
Creating and Writing Contiguous Dataset

- In this test we created a file and stored two dimensional array of short unsigned integers; size of array varied from 2MB and up to 512MB
- We measured
 - Total time to
 - create a file
 - create a dataset
 - write a dataset
 - close the dataset and the file
 - Time to write dataset only

Creating and Writing Dataset on IRIX (total time)



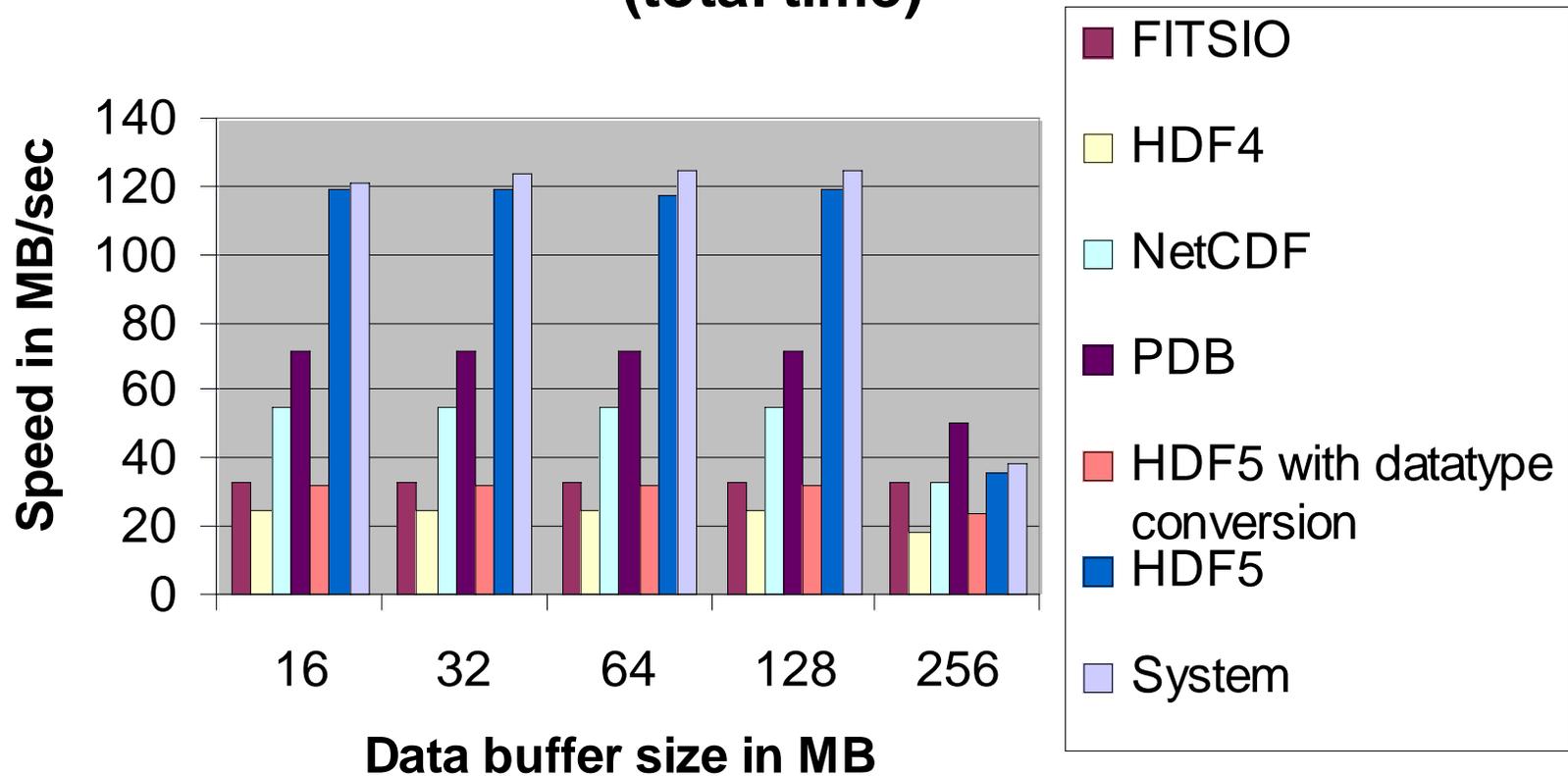
Creating and Writing Dataset on IRIX (write time only)



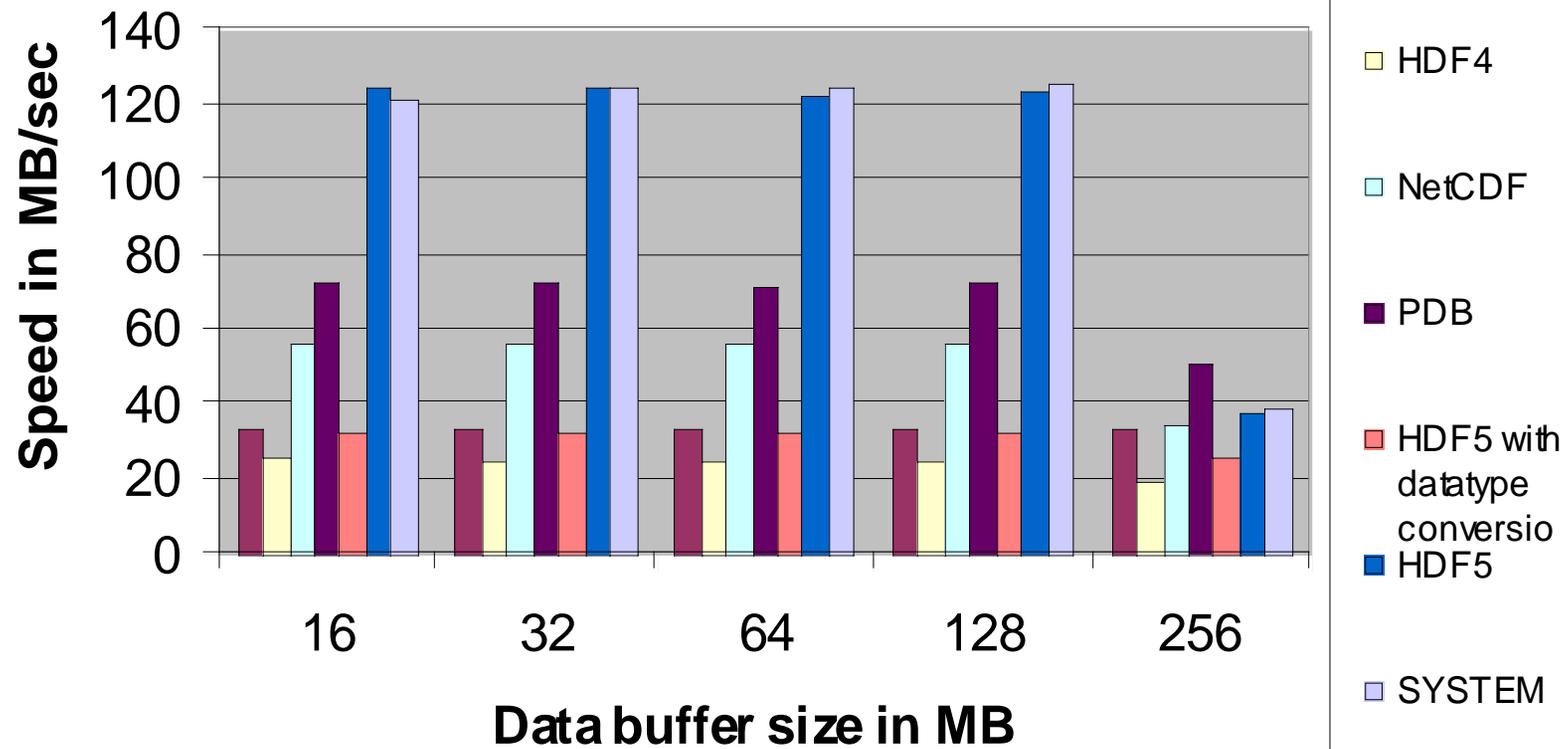
Speed ratios for writing dataset on IRIX

<i>LIB</i> <i>(Library name)</i>	<i>HDF5/LIB</i> <i>(total time ratios)</i>	<i>HDF5/LIB</i> <i>(write time ratios)</i>
System	80% - 100 %	89% -100 %
FITSIO	1.8 – 2.07	2.0 – 2.5
HDF4	1.5 – 2.5	1.5 – 1.9
NetCDF	1.2 – 1.7	1.5 – 1.7
PDB	1.3 – 1.6	1.4 – 1.9

Creating and Writing Dataset on LINUX (total time)



Creating and Writing Dataset on Linux (write time only)



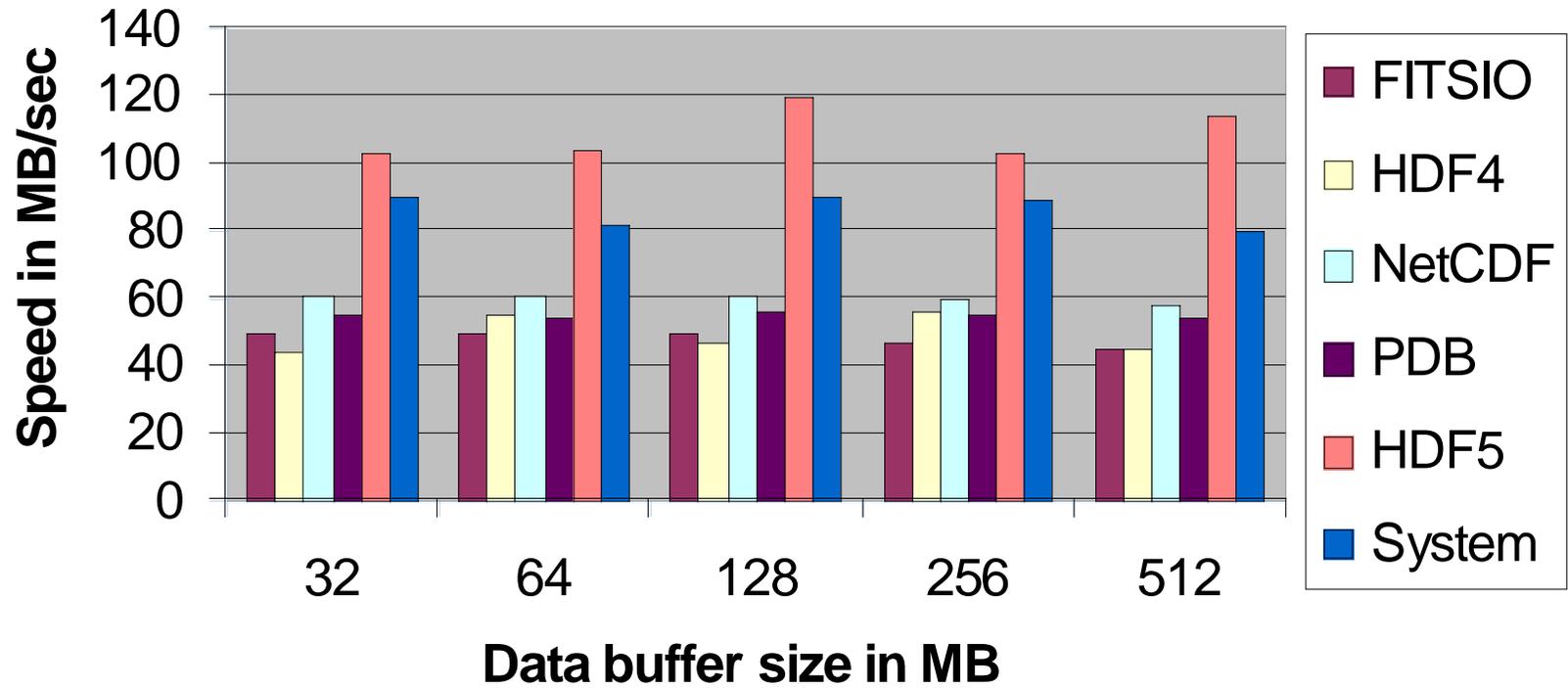
Speed ratios for writing dataset on Linux

<i>LIB</i> <i>(Library name)</i>	<i>HDF5/LIB</i> <i>(total time ratios)</i>	<i>HDF5/LIB</i> <i>(write time ratios)</i>
System	94% - 100 %	98% -100 %
FITSIO	3.5 – 3.6	3.7 – 3.8
HDF4	4.7	4.8 – 5.1
NetCDF	2.0 – 2.2	2.1 – 2.2
PDB	1.6	1.7

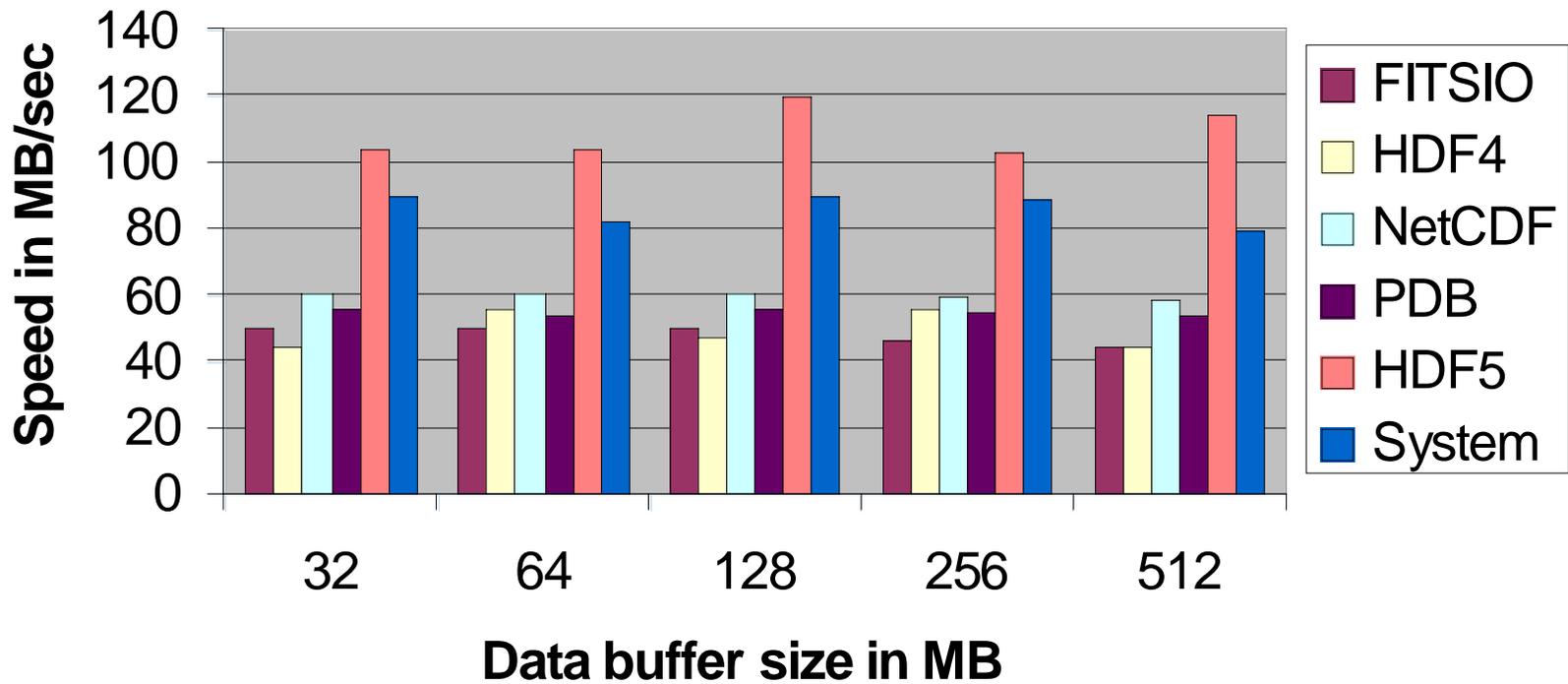
Reading Contiguous Dataset

- In this test we created two dimensional array of short unsigned integers than we read it back; size of array varied from 2MB and up to 512MB
- We measured
 - Total time to
 - open a file
 - open a dataset
 - read a dataset
 - close the dataset and the file
 - Time to read dataset only

Reading Contiguous Dataset on IRIX (total time)



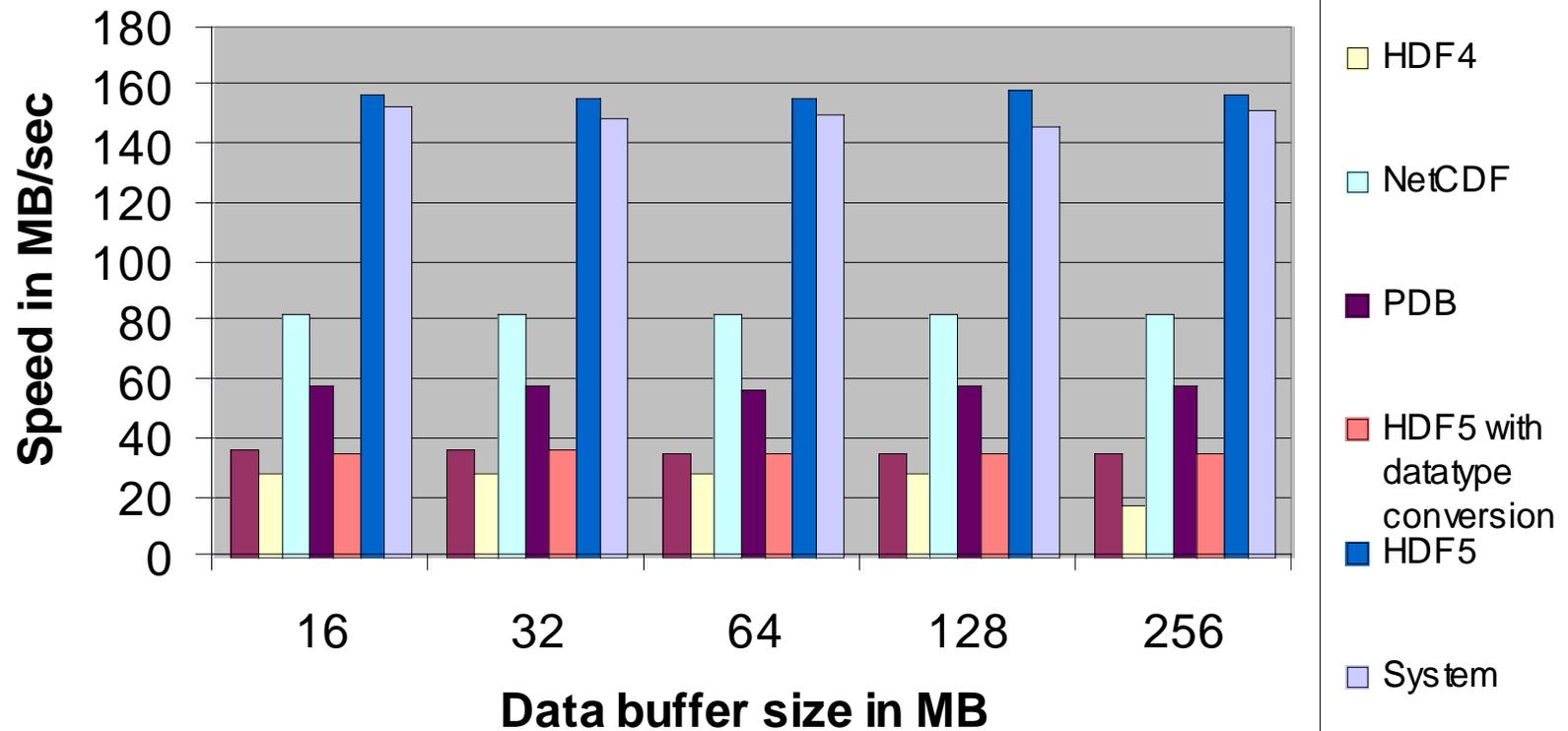
Reading Contiguous Dataset on IRIX (read time only)



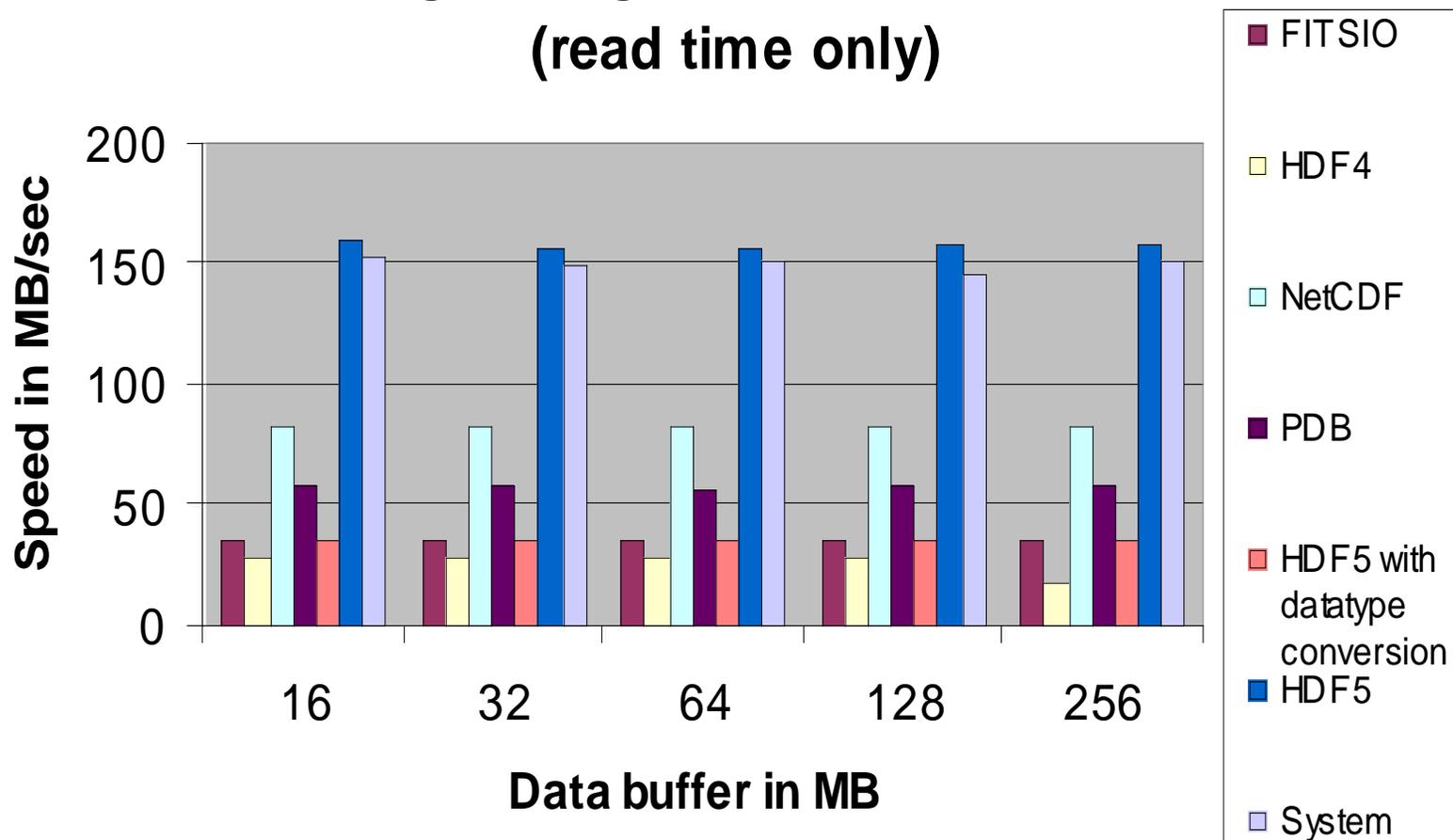
Speed ratios for reading dataset on IRIX

<i>LIB</i> <i>(Library name)</i>	<i>HDF5/LIB</i> <i>(total time ratios)</i>	<i>HDF5/LIB</i> <i>(read time ratios)</i>
System	74% - 100 %	91% -100 %
FITSIO	1.9 – 2.7	1.8 – 2.6
HDF4	1.8 –2.5	1.8 –2.5
NetCDF	1.5 – 2.2	1.7 –2.7
PDB	1.9 – 3.1	1.9 – 3.7

Reading Contiguous Dataset on Linux (total time)



Reading Contiguous Dataset on Linux (read time only)



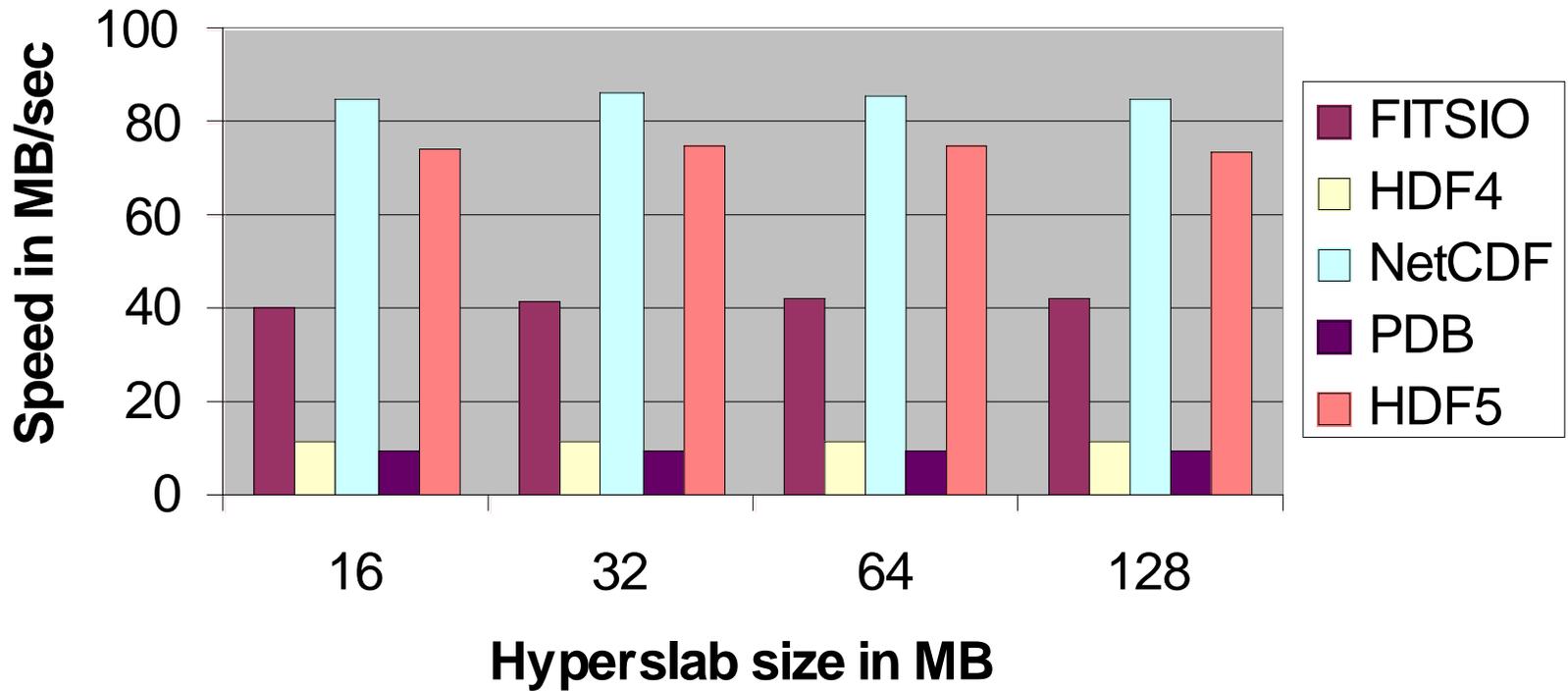
Speed ratios for reading dataset on Linux

<i>LIB</i> (<i>Library name</i>)	<i>HDF5/LIB</i> (<i>total time ratios</i>)	<i>HDF5/LIB</i> (<i>read time ratios</i>)
System	97% - 100 %	100 %
FITSIO	4.0 – 4.5	4.0 – 4.5
HDF4	5.2 – 5.5	5.5 – 5.7
NetCDF	1.8 – 1.9	1.9
PDB	2.6 – 2.8	2.7

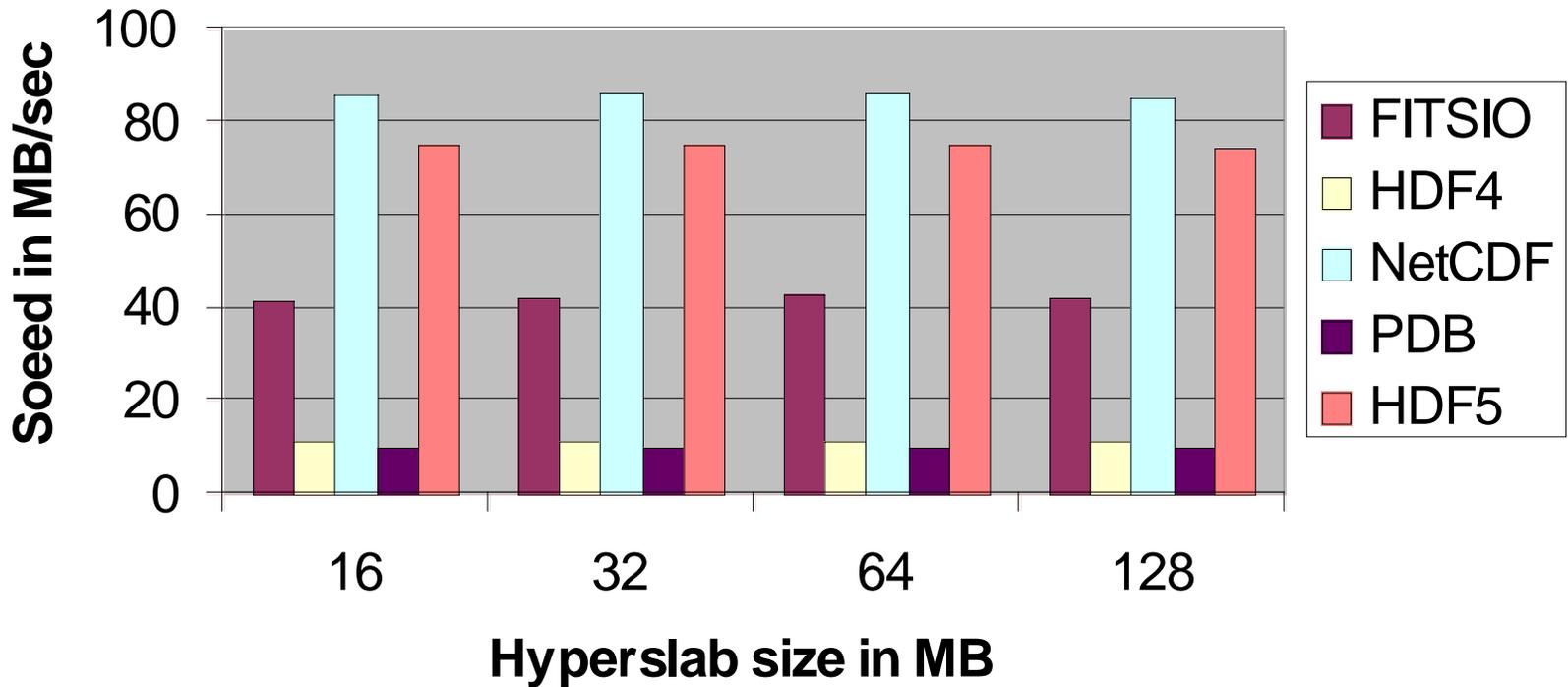
Reading Contiguous Hyperslab of the Dataset

- In this test we created two dimensional array of short unsigned integers and than read contiguous hyperslab of the dataset; size of the dataset was up to 256 MB and size of the hyperslab varied from 1MB up to 64 MB
- We measured
 - Total time to open a file, dataset, select and read hyperslab, close the dataset and the file
 - Time to read hyperslab only

Reading Contiguous Hyperslab on IRIX (total time)



Reading Contiguous Hyperslab on IRIX (read time only)



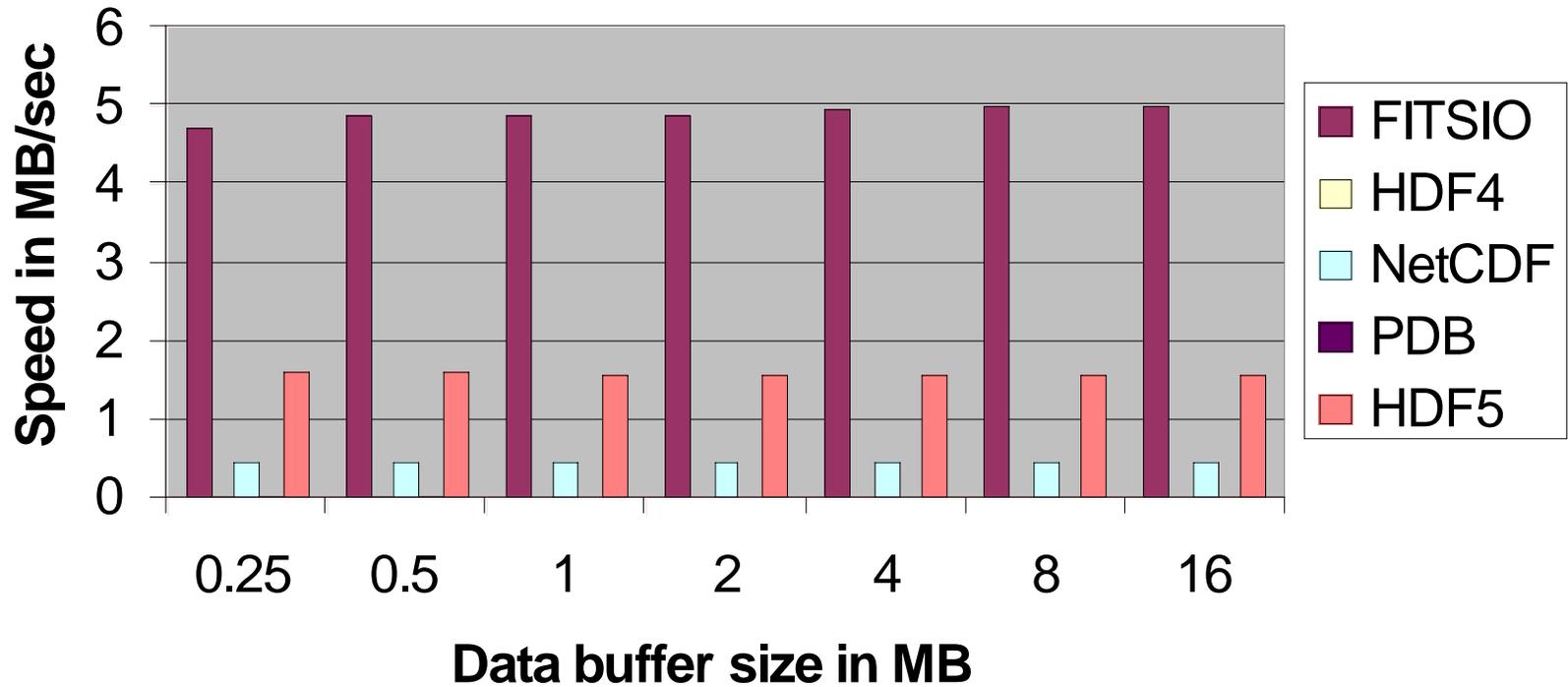
Speed ratios for reading contiguous hyperslab on IRIX

<i>LIB</i> (<i>Library name</i>)	<i>HDF5/LIB</i> (<i>total time ratios</i>)	<i>HDF5/LIB</i> (<i>read time ratios</i>)
FITSIO	1.7 – 2.1	1.8
HDF4	6.4 – 7.4	6.4 – 8.
NetCDF	0.8	0.8 – 0.9
PDB	7. - 8.	7.6 – 9.5

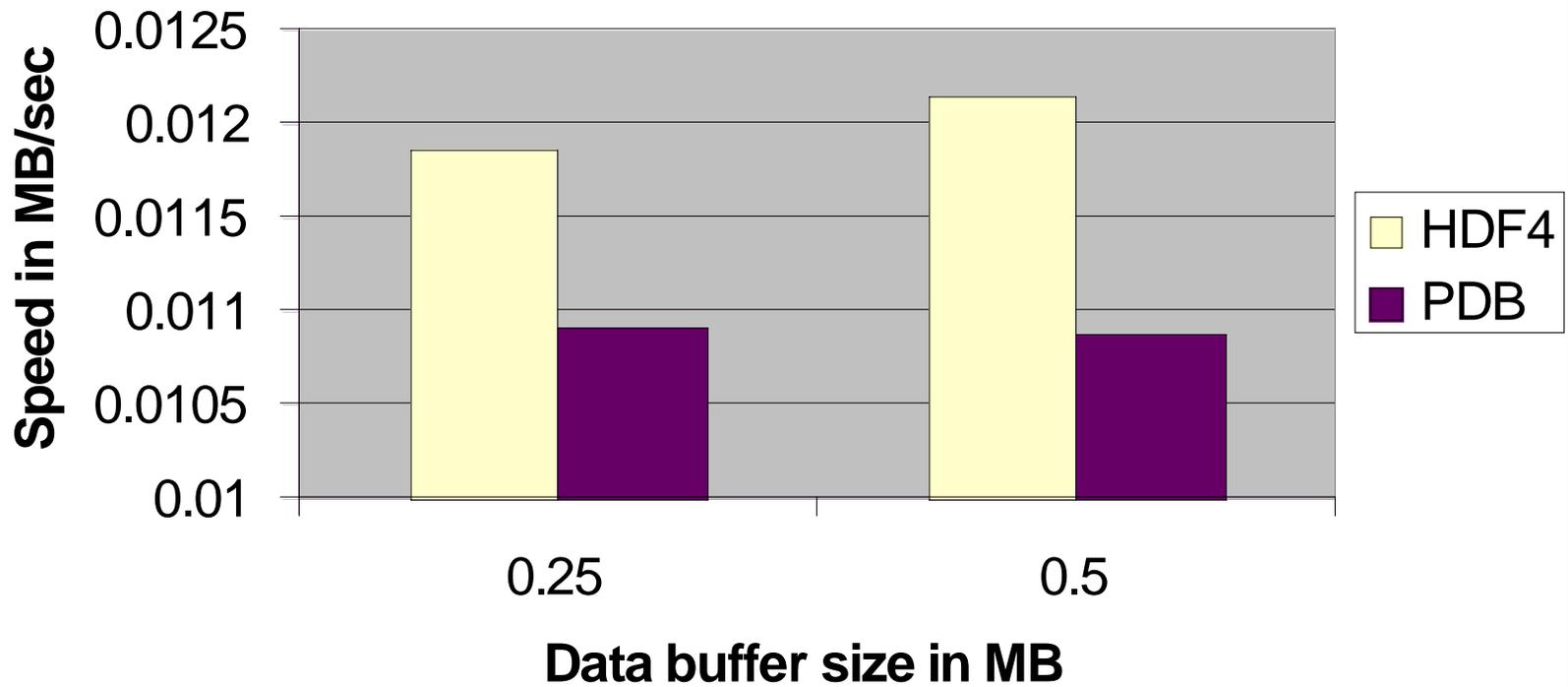
Reading Every Second Element in the Hyperslab

- In this test we created 256 MB two dimensional array of short unsigned integers; then we read read back every second element of the selected hyperslab
- We measured
 - Total time to open a file and dataset, select and read every second element of the hyperslab, close the file and dataset
 - Time to read selection only

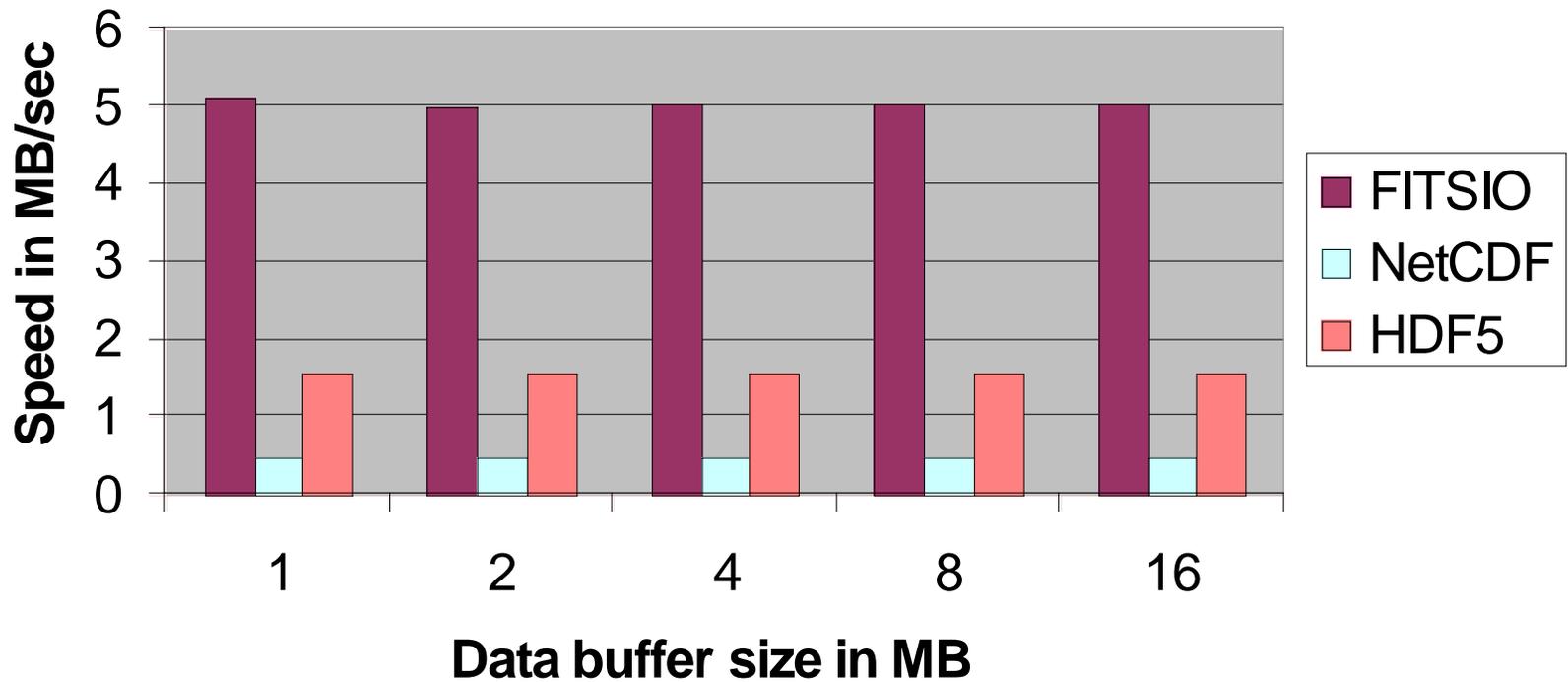
Reading Every Second Element of the Hyperslab on IRIX (total time)



Reading every Second Element on IRIX (total time)



Reading Every Second Element of the Hyperslab on IRIX (read time only)



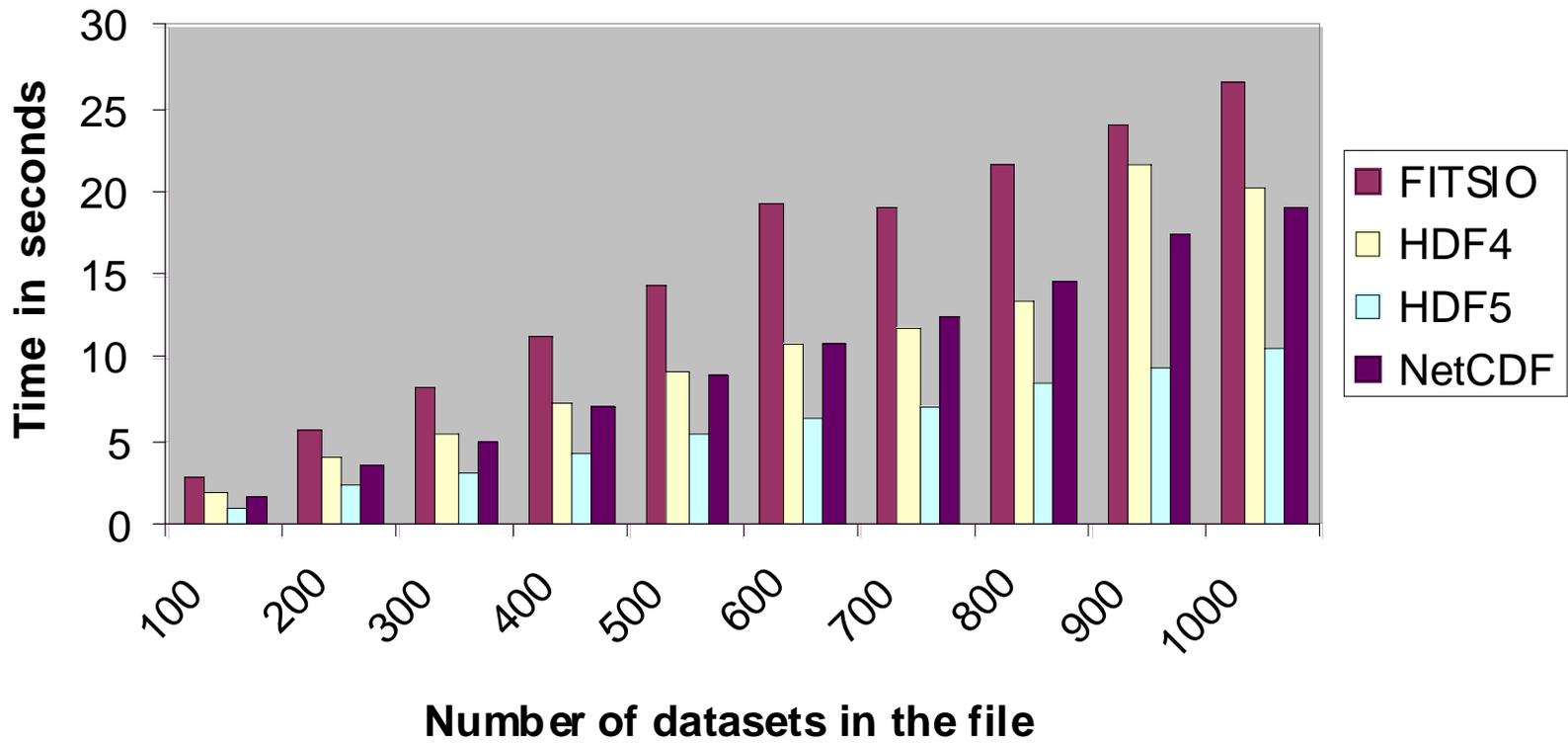
Speed ratios for reading every second element of contiguous hyperslab on IRIX

<i>LIB</i> (<i>Library name</i>)	<i>HDF5/LIB</i> (<i>total time ratios</i>)	<i>HDF5/LIB</i> (<i>read time ratios</i>)
FITSIO	0.3	0.3
HDF4	130 -136	130 - 136
NetCDF	3.4	3.4
PDB	147	147

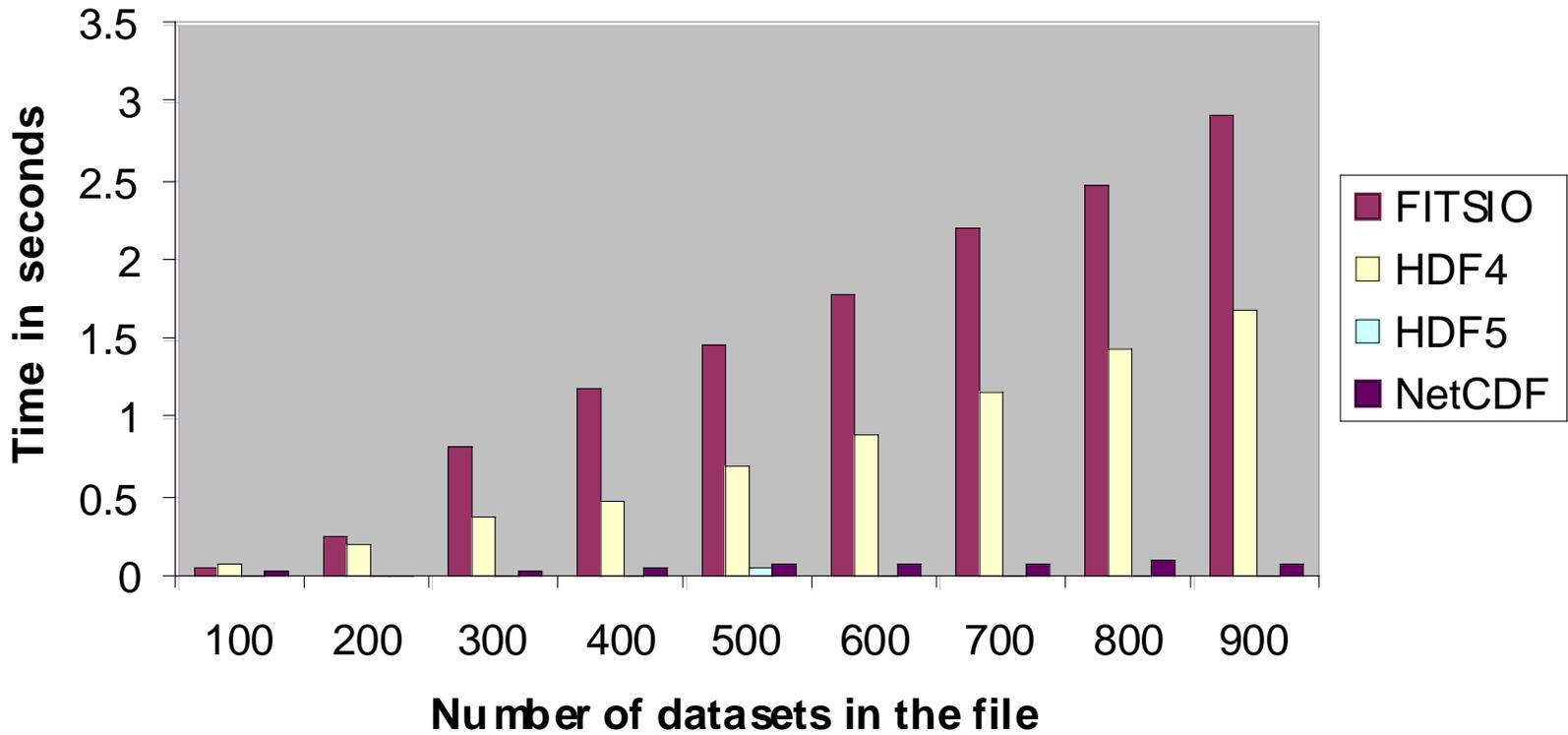
Creating and Writing Multiple Datasets

- In this test we created up to 1000 1MB two dimensional datasets of short unsigned integers; then we read the last created dataset
- We measured
 - Time to
 - create a file
 - create and write N datasets
 - close all datasets and the file
 - Time to open the file, read N-th dataset and close the file

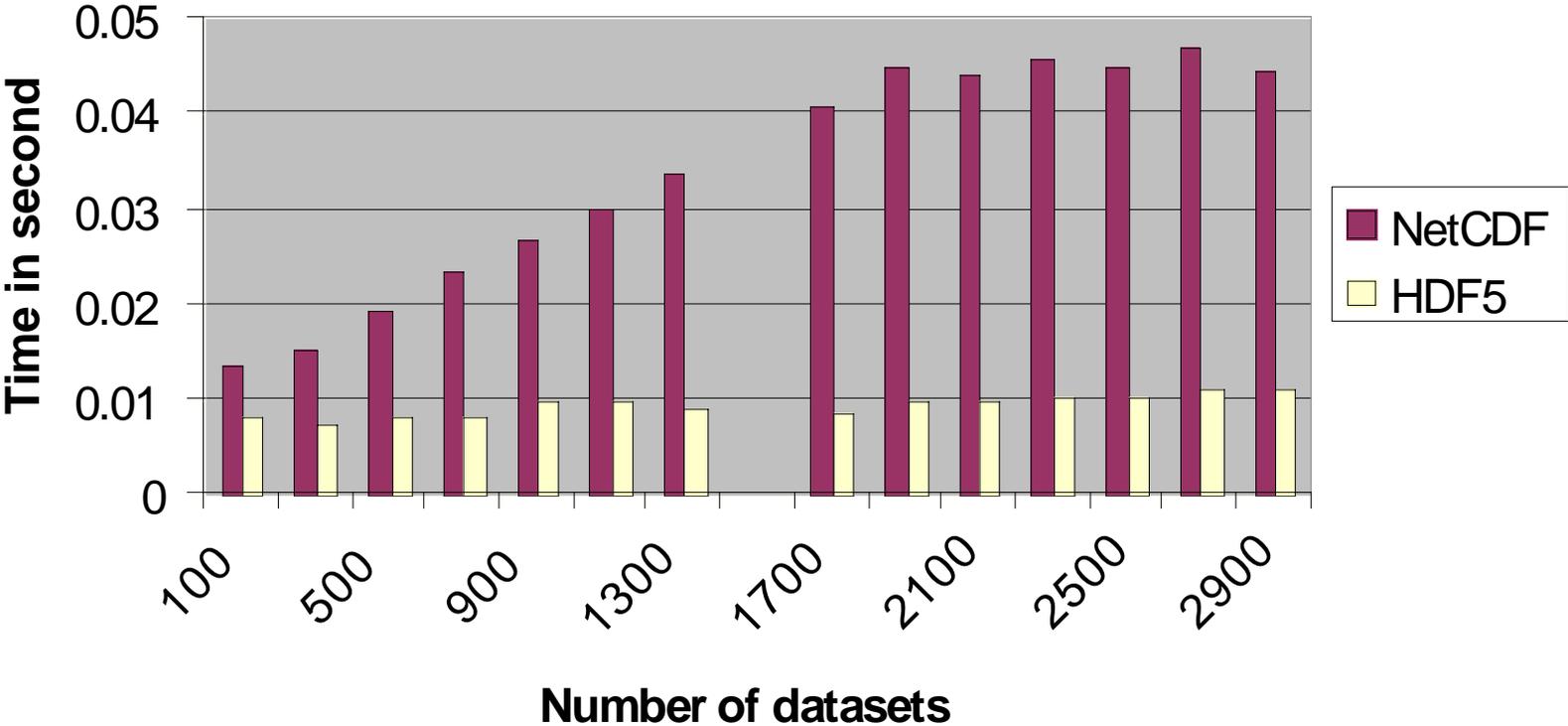
Creating and Writing Datasets on IRIX



Reading Dataset from the Files



Reading a dataset from the file on IRIX



Conclusions

- HDF5 is 1.6 - 8 times faster when performs native write/read
- HDF5 needs some tuning when datatype conversion is used
- When contiguous subsetting is used, HDF5 performs several times better than FITSIO, HDF4 and PDB and achieves about 80% of NetCDF performance
- HDF5 is an order of magnitude faster in accessing datasets within the file with many objects