# Chapter 10
# High Performance Storage for Big Data Analytics and Visualization

**Armando Fandango**
*NeuraSights, USA*

**William Rivera**
*Microsoft, USA*

## ABSTRACT

*Scientific Big Data being gathered at exascale needs to be stored, retrieved and manipulated. The storage stack for scientific Big Data includes a file system at the system level for physical organization of the data, and a file format and input/output (I/O) system at the application level for logical organization of the data; both of them of high-performance variety for exascale. The high-performance file system is designed with concurrent access, high-speed transmission and fault tolerance characteristics. High-performance file formats and I/O are designed to allow parallel and distributed applications with easy and fast access to Big Data. These specialized file formats make it easier to store and access Big Data for scientific visualization and predictive analytics. This chapter provides a brief review of the characteristics of high-performance file systems such as Lustre and GPFS, and high-performance file formats such as HDF5, NetCDF, MPI-IO, and HDFS.*

## INTRODUCTION

Input/output (I/O) system contributes to a significant amount of time and resource usage in scientific applications producing Big Data. High performance storage and I/O have become essential for business and scientific applications. Scientific applications use I/O and storage for several purposes such as storing the simulation output. The simulation output is generally recorded at multiple time points during the execution of an application, thus producing large amounts of complex scientific data. This data needs to be accessed by separate applications for further post-simulation processing and analysis. Thus, performance of storage and I/O system becomes important component for scientific Big Data processing.

DOI: 10.4018/978-1-5225-3142-5.ch010

Parallel I/O is a technique to allow concurrent access to storage by the same or different applications. However, parallel I/O relies on parallel file system otherwise parallel access to data is slowed down by sequential access to storage. There are several parallel file systems such as Lustre, GPFS (Generalized Parallel File System), and PVFS (Parallel Virtual File System), that apply different strategies in architecture and data access. Several parallel file I/O formats and libraries have also become popular at application and domain level. HDF5 and NetCDF are examples of file I/O systems that have become popular and adopted by several scientific applications for providing high performance access to complex data. Some of these application level systems are extended further by domain experts to meet challenges of their specific domains, such as H5Part for particle simulations data and HDF-EOS for earth systems data.

With the popularity of High Performance Computing (HPC) on many-core systems, multi-node clusters and accelerator cards, the scientific applications are able to perform much faster computations achieving a speedup of 10x to 100x or even more in some cases. However, I/O takes up from 10% to 40% in such applications. Classic I/O systems are slow when compared to the speedups achieved in compute. The I/O patterns of different applications require a different approach to parallelization. The goal of improving I/O performance is to increase the bandwidth size and decrease the number of I/O operations. Thus, the high performance storage systems strive to achieve this goal.

As shown in Table 1, the high-performance storage systems at several layers are available to achieve the goals of parallel and distributed I/O. At the lowest level, the High Performance File Systems are attempting to enable the parallel reads and writes by several cores, nodes, applications and processes. At the next layer level the low level libraries such as MPI-IO enable the programs to parallelize the file I/O operations. The next level of High Level Parallel I/O library builds on top of such low level I/O libraries and provides an easy way to integrate the parallelization into the application code. Some domains and applications have yet another layer of parallel I/O libraries specific to their domain. The application sits on top layer and performs faster due to the availability of parallel libraries and file system.

The scientific Big Data is generated in two kinds of workloads: checkpoint workload refers to the output generated at intermediate checkpoints in scientific computation and analysis workload refers to the output generated while analyzing the overall computation or the output of intermediate checkpoints. Both of these kinds of workloads generate I/O in bursts and spikes. With the increase in size of main memory in high performance systems, the burst I/O has also increased. Thus, High Performance File Systems and I/O have evolved to support this kind of bursts for storing scientific Big Data generated by parallel applications.

This chapter provides insight into challenges, issues and solutions in high performance storage and I/O systems. Examples of High Performance File System such as Lustre, GPFS, and OrangeFS, and examples of high performance I/O such as MPI-IO, Parallel HDF5, NetCDF5, and ADIOS are provided with brief descriptions.

## HIGH PERFORMANCE FILE SYSTEMS (HPFS)

A High-Performance File System (HPFS) is the backbone of any scientific Big Data processing facility. Generally, HPFS is deployed in High Performance Computing (HPC) clusters to allow users access the file system concurrently on multiple nodes through multiple programs. HPFS are built with goals of providing very high scalability in terms of scaling to petabytes and exabytes in data volumes to concurrent access by multiple nodes and processes to the tune of hundreds and thousands and to terabyte per