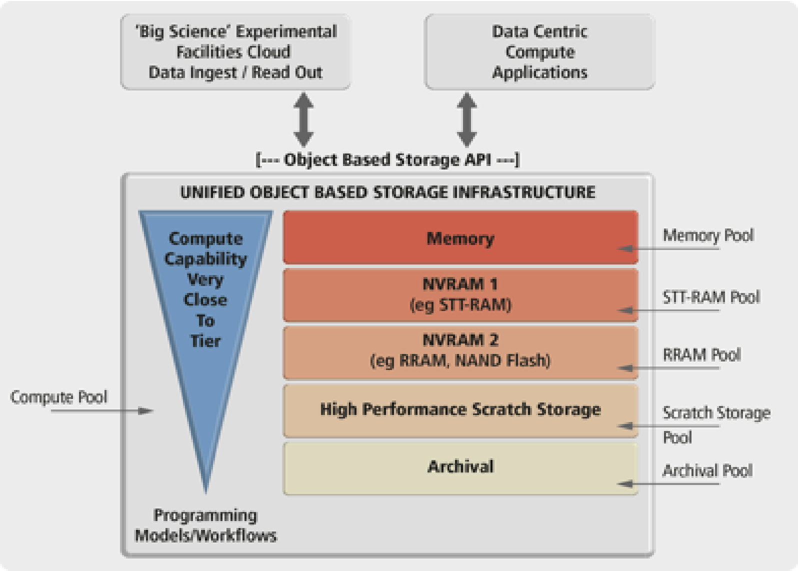
Percipient Storage: A Storage Centric Approach to BDEC

Malcolm Muggeridge and Dr Sai Narasimhamurthy
(Seagate Technology)

Abstract: *We discuss a generalised data centric computing architecture, termed "Percipient Storage" to address the unique I/O and computational requirements of BDEC systems. Percipient Storage is built around the principles of a storage system with very deep I/O hierarchies that have in-built computational capabilities drastically reducing the data movement overheads within the overall infrastructure. An advanced object storage system built from the ground up to handle I/O in the Exascale realm provides the software infrastructure for Percipient Storage.*

Introduction: Our understanding of BDEC systems are systems that would need to handle and generate science from *very high volumes* of simulation *and* instrument data. In the realm of BDEC, data may not need to be just stored to provide resiliency, but re-organised, transformed, reduced, queried, visualised, etc. The complex associations between data and predictive analysis based on these associations will need to be studied. This analysis generates scientific insights that could potentially feed back into simulation optimisation. Hence for BDEC, there is a need for a storage system that is:

a. Capable of storing and retrieving data with very high throughput and low latency at scales that are currently not possible with existing I/O solutions, considering the sheer volumes of data the I/O subsystem needs to handle.
b. Capable of running complex data processing and analytics tasks in parallel with simulations to drastically reduce the time to solution, avoiding extreme data movements in the I/O stack between compute and stored data.
c. Capable of providing direct access to vast external data sets, for example from instruments.


'Big Science' Experimental Facilities Cloud Data Ingest / Read Out
Data Centric Compute Applications
[--- Object Based Storage API ---]
UNIFIED OBJECT BASED STORAGE INFRASTRUCTURE
Compute Capability Very Close To Tier
Memory
NVRAM 1 (eg STT-RAM)
NVRAM 2 (eg RRAM, NAND Flash)
High Performance Scratch Storage
Archival
Memory Pool
STT-RAM Pool
RRAM Pool
Scratch Storage Pool
Archival Pool
Compute Pool
Programming Models/Workflows

We propose an architecture (exemplified in the figure on the left) for a storage system that can address these requirements.

The Architecture: We propose an advanced object based storage solution, with a very flexible API for applications and data sets, that is designed from the ground up, to cater for I/O workloads in the Exascale realm. The storage system will accommodate multiple storage device technologies in a multi-tiered hierarchy, with a homogenous view of data across them, with I/O performance (latency and throughput) scaling up as we go up the hierarchy, and the storage capacity scaling down as we go up the hierarchy. The solution will have the capability to run computations on data at any tier with compute capability increasing (denoted by the inverted triangle in the picture) as we go up the hierarchy. Data could reside in any of the aforementioned tiers, which have their own embedded processing capability (a) Main Memory in top tiers geared towards heavier computation, (b) Tiers of NVRAM, (c) High Performance disc-based storage, or, (d) Archival storage subsystem (low performance high capacity disc subsystem). Multiple tiers of NVRAM that could be used are exemplified by STT-RAM/STT-MRAM, RRAM/ReRAM, and NAND flash.

- The system can run *full data centric applications and data analytics tasks*, or *specific computational kernels* potentially at any tier and in parallel with computations in the main compute nodes. Having computational capability very close to the data anywhere in the hierarchy allows the storage system to have the quality of “percipience”, or the ability to perceive and extract valuable information from data wherever it resides. The system optionally can just be a plain data storage system for compute intensive simulations. Percipient storage interfaces into Big Science compute infrastructures thereby expanding their capabilities to handle BDEC workloads thus obviating “rip and replace”.

  Percipient Storage will provide interfaces to remote data sources to load data sets and to the compute nodes either creating data through simulations or performing complex analytics on stored data. These interfaces will be able to interoperate with well-known existing cloud storage interfaces such as Amazon S3, as well as more traditional HPC interfaces such as HDF5 or NetCDF. Percipient Storage will also provide APIs for third parties to build their own data management solutions as plug-ins, such as hierarchical storage management. Data hence can be moved automatically between tiers based on its dynamic usage, availability of resources, demand, user preferences and overall utilisation.

- Advanced analytics stacks that deal with more complex workflows than simple hadoop/MapReduce (Ex:Apache Flink, etc) will be able run on on top of the object API enabling functions such as data mining or predictive analysis as part of the scientific workflow. Existing programming models can interact with Percipient storage either by offloading computational kernels (MPI) , or by exploiting the expanded memory addressing possible with the inclusion of the many NVRAM tiers(PGAS) thus providing the exciting prospect of blurring the lines between traditional memory and data storage in the future. Further, innovations in system-ware (advanced debugging frameworks, etc) can easily be built on top of Percipient Storage.

Future work: Co-Designing of the Percipient storage system fitting to the needs of BDEC use cases will be very important as we continue to clarify and validate the architecture. This work is already being explored as part of Horizon 2020 proposals.