[New text for In-Memory for Vol. 6 Reference architecture]

### Platforms

The platform element consists of the logical data organization and distribution combined with the associated access APIs or methods. That organization may range from simple delimited flat files to fully distributed relational or columnar data stores. The storage mediums range from high latency robotic tape drives, to spinning magnetic media, to flash/solid state disks, to random access memory. Accordingly, the access methods may range from file access APIs to query languages such as SQL. A typical Big Data framework implementations would support either basic file system style storage or in-memory storage and one or more indexed storage approaches. This logical organization may or may not be distributed across a cluster of computing resources based on the specific Big Data system considerations.

The platform may also include data registry and metadata services along with semantic data descriptions such as formal ontologies or taxonomies.

In most aspects, the logical distribution/organization of data in Big Data storage frameworks mirrors what is common for most legacy systems. Figure XXX below shows a brief overview of data organizations approaches for Big Data.

Figure 3: Data Organization Approaches

As mentioned above many Big Data logical storage organizations leverage the common file system concept (where chunks of data are organized into a hierarchical namespace of directories) as their base and then implement various indexing methods within the individual files. This allows many of these approaches to be run both on simple local storage file systems for testing purposes or on fully distributed file systems for scale.

#### In-memory

The NIST Big Data Reference Architecture infrastructure (Figure 2) indicates physical resources required to support analytics.  However such infrastructure will vary, that is, will be optimized, for the  
data characteristics of the problem under study.  Large, but static, historical datasets with no urgent analysis time constraints would optimize the infrastructure for the Volume view of a Big Data analysis,  
while time-critical analyses such as intrusion detection or social media trend analysis would optimize the infrastructure for the Velocity characteristic.  The latter characteristic, Velocity, implies the  
necessity for extremely fast analysis and the supporting infrastructure to support it, namely, very low latency, in-memory analytics.  
  
In-memory database technologies are now experiencing increased application in the field due to the  
significant drop in memory prices and the increased scalability of modern servers and operating systems.  Yet an in-memory component of a Velocity-oriented infrastructure will require more than simply massive RAM; it will also require optimized data structures and memory access algorithms  
to fully exploit RAM performance.  Current in-memory database offerings are beginning to address this issue.

Traditional database management architectures are designed to use spinning disks as the primary storage mechanism; and the main-memory of the computing environment is relegated to providing caching of data and indexes. Many of these In-memory storage mechanisms have their roots in the massively parallel processing and super computer environments popular in the scientific community.

These approaches should not be confused with solid state (e.g. flash) disks or tiered storage systems that implement memory based storage in that those simply replicate the disk style interfaces and data structures but with faster storage medium. Actual in-memory storage systems typically eschew the overhead of file system semantics and optimize the data storage structure to minimize memory footprint and maximizing data access rates. These in-memory systems may implement general purpose relational and other NoSQL style indexing and interfaces or be completely optimized to a specific problem and data structure.

Like traditional disk based systems for Big Data these implementations frequently support horizontal distribution of data and processing across multiple independent nodes although shared memory technologies are still prevalent in specialized implementations. Unlike, traditional disk based approaches in-memory solutions and the supported applications must account for the lack of persistence of the data across system failures. Some implementations, leverage a hybrid approach involving write-through to more persistent storage to help alleviate the issue.

The advantages of in-memory approaches include faster processing of intensive analysis and reporting workloads. In-memory systems are especially good for analysis of real time data such as what might be needed for complex event processing of streams. For reporting workloads, the performance improvements can often be on the order of several hundreds of times faster especially for sparse matrix and simulation type analytics.