**Title of Pre-application:** Hierarchal Extreme Scale Knowledge Management
**Principal Investigator:** Scott Klasky, Group Leader, ORNL, 854-241-9980, klasky@ornl.gov
**Funding Opportunity Announcement Number: DE-FOA-0001338**
**List of all co-PIs and Key/Senior Personnel**
Hasan Abbasi, Oak Ridge National Laboratory
Mark Ainsworth, Oak Ridge National Laboratory JFA, Brown University
Matthew Curry, Sandia National Laboratory
Qing Gary Liu, Oak Ridge National Laboratory
Jay Lofstead, Sandia National Laboratory
Kimmy Mu, Oak Ridge National Laboratory
Carlos Malzahn, U. Cal. Santa Cruz
Manish Parashar, Rutgers University, Oak Ridge National Laboratory
Sudharshan Vazhkudai, Oak Ridge National Laboratory
Lee Ward, Sandia National Laboratory

**Objectives:** Exascale scientific discovery will be severely bottlenecked without sufficient new research into managing and storing the large amounts of data that will be produced during the simulation, and analyzed for months afterwards. Our goal in this project is to address the associated I/O and storage challenges on the future storage landscape, and expedite insights into mission critical scientific processes. To that end, we will build on the capabilities offered by the Sirocco peer-to-peer file system under development at Sandia and the object storage and annotation expertise of UC Santa Cruz to develop a unique application aware data management scheme. It will build on the existing multi-tier storage aware approach offered by Sirroco to incorporate new functionality and extended object metadata and APIs for 1) matching available storage bandwidth across the storage hierarchy with data importance employing application aware data compression (both lossy with specified error bounds and lossless), 2) a selectable performance/quality tradeoff when reading data, and 3) incorporate reader prioritization for data annotation, placement, and data quality when writing data. The major challenge in this project is metadata management connecting different object quality levels and connecting the user with the best possible outcomes given the current system state. The new user APIs required to interact with this rich metadata system will drive effective use of the entire middleware and storage stack. The project is compriesed of a team with strong expertise in I/O middleware (ORNL, Rutgers), file system (SNL, UCSC) and storage (UCSC), and connect and coordinate these key storage components in a seamless fashion.

Our approach can most easily be understood by considering a road navigation program. Similar to the navigation program, we will offer a coarse grained, fast overview of a data set with progressively more detail in areas of interest noted by data features and less data in areas without. These progressively more detailed data views require more storage space and time to retrieve. By incorporating middleware informed of both how to identify relevant data features and the intended use locations, it will be able to selectively store data at different quality levels, with bounded errors in the case of lossy compression, and allow a user to select what data to retrieve by specifying both an acceptable retrieval timeframe and quality required. If the request cannot be fulfilled, the storage system can negotiate with the user providing information about the quality and timeframes available for a subsequent request.

Our objective here is to reduce the time to knowledge, an end-to-end metric relevant to the scientific discovery process. Beyond the traditional high volume I/O pattern of checkpoint/restart, we will address the challenges posed by other essential data access patterns in the knowledge gathering process. Through a deeper insight into the scientific process we will encode and utilize accuracy and errors as optimization parameters.

For example, scientific simulations contain approximations, as do measurements from observations and experiments, and depending on the goal, these approximations are acceptable. We can leverage this observation to optimize the presentation of data to the user and to implement various tradeoffs. For example, users can ask for information within a given accuracy bound, allowing us to offer a mode for re-computation vs. data storage and reterival.

**: please list other things from OLCF**.

**Key Technical Approach:** Our overall technical approach is based on an application-aware runtime realization of tradeoffs in data representation, data placement and data access. We will allow users to "plug-in" their knowledge about the data, not as bytes but as motifs, allowing the I/O and storage system to understand user intentions and the relationships between data objects, as well as data access and transport patterns. This will facilitate efficient mapping of data from user space, such as a multi-dimensional array, onto various storage tiers. Rather than focus on simple data compression, we will incorporate a multitude of techniques to reduce data on the faster yet smaller storage tiers, and keep the less reduced information in the lower storage tiers.

We will support additional data access modes as well. **First**, within a given accuracy bounds, offer a mode for re-computation from data stored in a "fast" tier to reduce data latencies. **Second**, exploratory analysis operations frequently entail an overall data set view followed by targeted data exploration based on identified features. We will offer support for a configurable data access mode to support these sorts of analysis accesses. An "overview" access mode that gives a quick, approximate within error bounds, data view that can guide feature selection offering rapid coarse-grained data exploration without requiring loading the complete, detailed data set from storage. Based on the granularity requested, the accuracy and size of the data returned can be adjusted. At the most extreme setting, the original data can be retrieved at the time cost of moving the potentially huge data quantity. **Third**, to ensure available storage for subsequent operations, we will offer automatic data migration based on user annotations for required data lifetimes using monitoring and learning techniques. Unlike existing approaches, this will be tempered both by the user annotations and through learned access patterns. While past access patterns may not indicate future access because the simulation run purpose may have changed, we are focused on scalability where runs are subsequently larger as the simulation prepares for a capability run. By learning from the output and access patterns during this run sequence, we can accurate decide how to place and organize data for the critical capability runs. Fourth, we anticipate storing multiple data copies, each compressed in different ways according to the underlying media, some of these copies will disappear based on storage pressures, but data persistence will be maintained according to user specifications. Assuming a relatively low latency cache layer before a tape system, we can offer exploratory data access reserving pulling data from tape to just the data required. This will save scientists time and make data stored on tape usable without long delays.

Our research efforts will be heavily focused on the need to, in a coordinated manner, adapt data and metadata retention policies to the dynamic resource balancing that will need to take place between the application, OS/R, and hardware.

The success of this project will provide insights into how to build autonomic middleware and storage layers which can interact well with each other, and take user-provided hints. Today, data is reduced by application scientist who have limited information on what the storage layer can provide. They often make compromises based on this limited knowledge and either tune their output for writing or reading performance. This data then gets moved to other locations, and much of the tuning is lost when the data is read back during their post processing. Furthermore, there is a limited set of operations which users will be able to stage to other staging nodes for real-time-reduction and visualization.