Title of Pre-application: Extreme Scale Hierarchal Knowledge Management

Principal Investigator: Scott Klasky, Group Leader, ORNL, 854-241-9980, klasky@ornl.gov

Funding Opportunity Announcement Number: DE-FOA-0001338

Co-PIs and Key/Senior Personnel

Hasan Abbasi, ORNL	Mark Ainsworth, ORNL JFA, Brown University
Matthew Curry, SNL	Qing Liu, ORNL
Jay Lofstead, ORNL	Kimmy Mu, ORNL
Carlos Malzahn, U. Cal. Santa Cruz	Manish Parashar, Rutgers University, ORNL
Feiyi Wang, ORNL	Lee Ward, SNL

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 in the context of current and emerging storage landscapes, and expedite insights into mission critical scientific processes. To that end, we will build on the capabilities offered by ADIOS and DataSpaces that provide I/O abstractions and services, the Sirocco peer-to-peer file system at Sandia and the object storage and annotation expertise of UC Santa Cruz, to explore application and multi-tier storage aware data management solutions. This project brings together 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 solutions will provide novel functionalities and APIs for 1) specifying, at the application level, data annotations that enable the qualification of the relative importance and utility of data objects, and enable them to be mapped at runtime to appropriate storage capabilities across the storage hierarchy; 2) specifying selectable performance/quality/cost tradeoffs from both the application and system perspectives; 3) evaluating these tradeoffs at runtime during data placement and movement, and executing the resultant policies in an autonomic system using models, heuristics and continuous learning; and 4) leveraging techniques such as application-aware data compression, re-computation at the requisite level of accuracy, and I/O prioritization to enforce these policies.

Our objective here is to reduce the time to knowledge, not just for a single application or workflow, but for the entire workload in the system. This is an end-to-end system wide metric relevant to the scientific discovery process. We will explore beyond the traditional I/O pattern of checkpoint/restart, and 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. Finally, we will take the knowledge from the storage system to provide vital feedback to the middleware so that the best possible decisions can be autonomically between the user intentions and the available system resources. Our overall goal is to ensure that optimizations can be made across 1) a single application, 2) an ensemble of applications, and 3) the entire suite of applications which are running on the system.

Our objective here is to reduce the time to knowledge, not just for a single application or workflow, but for the entire workload in the system. This is an end-to-end system wide metric relevant to the scientific discovery process. We will explore beyond the traditional high volume I/O pattern of checkpoint/restart, and 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. Ultimately, we will take the knowledge from the storage system to provide vital feedback to the middleware so that the best possible decisions can be autonomically between the user intentions and the available system resources. We will test our prototypes on current and future DOE system with many of today's applications, including the s XGC1, GTC, QMCPack, and SpecFM3D simulations.

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 plug-in their knowledge about the data, represented not as bytes but as motifs, allowing the I/O and storage system to understand user intentions, the relationships between data objects, and also data access and transport patterns. This will facilitate efficient mapping of data from user space onto various storage tiers, and application-guided data reductions.

Incorporating awareness of the data content and user intent into our decision making and data management process is integral to our approach. For example, we will offer a coarse grained, quick overview of a data set with progressively more detail in areas of interest such as those containing features, and less detail in areas without. These progressively more detailed data views require more storage space and time to retrieve. By defining mechanisms for applications to provide this knowledge and incorporating its awareness into our middleware and storage allow it will be able to selectively store data at different levels, and allow a user to select what data to retrieve by specifying both an acceptable timeframe and accuracy.

The I/O and storage system prototype will enable highly flexible data access modes that are common in scientific applications and associated analysis. **First**, the storage system will offer the ability to directly store different portions of a single dataset–even a single variable at differnt levels of the storage hierarchy and apply different data transformation operations to each portion. For example, full data can be queued to tape while a highly compressed version intended for high-level analytical views can be stored in NVM or even RAM within the storage hierarchy. The storage system will support the metadata and plug-ins reuired to support this sort of data storage approach. **Second**, new storage access APIs will be developed that understand the notion of a time and data quality for requested data. The storage system, through the knowledge of both the data quality stored in differeht storage hierarchy tiers and the approximate time to retrieve data from the different tiers, the user can manage the tradeoff between data precision and time to retrieve. This storage system support will be managed through middleware insulating the user as much as possible from the new APIs. The storage system must also support plug-ins to potentially decompress or expand data to the original size. Both lossless and lossy with error bounds style storage is assumed. The time factor must take into account how long this operation takes for servicing the user request. **Third**, the storage system will offer annotations within the metadata and support for both predictive and reactive data placement and migration. While past access patterns may not indicate future access because the simulation run purpose may have changed, we are focused on scalability where subsequent runs are larger as the simulation prepares for a capability run. By learning from the output and access patterns during this run sequence, we can accurately decide how to place and organize data for the critical capability runs. The reactive mechanisms will consider the space and performance characteristics, as well as the requested data fidelity, to determine where and how to pull and store data sets. In some cases, a full fidelity will be pulled from tape to disk and no further to preserve 100% data fidelity. In others, a lossy-compressed within error bounds version may be generated and stored in local node RAM for very fast access with an acceptable error. **Fourth**, to effectively manage data storage capacities, the storage system's capability for storing multiple, potentially different data quality versions of the same data will be leveraged to offer eviction and migration. The main challenge of this approach is to keep in mind the intentional placement decisions made to optimize future data access while ensuring proper system operation by not exhausting any tier inappropriately. If a single application has been allocated the entire machine for a capability run, completely consuming resources on the machine is expected. For smaller runs, these allocations must be balanced to support the diverse workload.

A key challenge in this project is defining and maintaining the metadata connecting different object quality and utility levels, and using this metadata to most appropriately manage the placement and access of data object based on user/application intent/constraints and storage system state. 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. This will be managed against the needs of individual simulations as well as what's happening on the entire system. This will reduce the time to knowledge by allowing user and system knowledge into the SSIO software layers.