Project Summary

Many problems at the forefront of science, engineering, medicine, and the social sciences, are increasingly complex and interdisciplinary due to the plethora of data sources and computational methods available today. A common feature across many of these problem domains is the amount and diversity of data and computation that must be integrated to yield insights. Such data is increasingly large-scale and distributed, arising from sensors, scientific instruments, scientific simulations and Internet data clouds. Applications need access to these disparate datasets and to the methods that operate on them. One solution is to "pour" the data and methods into a single platform (e.g., a cloud) for integration and execution. However, for many complex data-intensive applications, moving the data may have restrictions due to ownership or size issues. Furthermore, different computational methods may run best on different platforms rather than within a uniform cloud, and applications are also increasingly integrating third-party services (e.g. GoogleEarth). Therefore, we believe data-intensive applications need first-class support for distributed multi-platform execution.

Today, data-intensive computing is in its infancy. Point solutions exist for single platforms and a limited set of application patterns (e.g. MapReduce) that are tied to low-level infrastructure (e.g. GFS on a cluster). The **lack** of infrastructure independence and extensibility restrict the re-use of existing implementations. The **lack** of support for distributed data has reduced the set of potential applications further and reduced effectiveness of national-scale cyberinfrastructure such as the TeraGrid for distributed applications. The **lack** of support for distributed versions of well-used patterns, and support for the expression of new patterns is another major obstacle. Collectively, these are all barriers to realizing the full potential of data-intensive science.

We propose a layered *framework* approach to enable a richer set of data-intensive applications to run on multi-platform systems. The key idea is a cross-cutting set of properties that exist at each framework level: *affinity*, *fault-tolerance*, *patterns*, and the first-class support for distributed data. We identify four abstractions at the application and runtime framework layers needed to realize these properties: *Pilot-Job*, *Pilot-Store*, *Compute-Store*, and *Data-Store*.

We propose to show how these abstractions can realize the above properties within our layered framework and how they can support distributed data-intensive applications on multi-platform systems. Given the extensive team experience with scientific applications and multi-platform systems, we plan an applications-driven approach. Analysis of applications will drive the construction of our framework and help us understand how to best support affinity, fault-tolerance, and patterns, in a general way. We have selected a diverse set of data-intensive scientific applications for this proposal: mpiBLAST, LIGO data analysis, and Montage. We will evaluate our framework and applications on a disparate set of available platforms: clusters, commercial clouds, TeraGrid, and open cloud testbeds.

Intellectual Merit: This project will develop a layered framework model to support distributed data-intensive applications. Abstractions are proposed to realize a set of cross-cutting properties critical to such applications on multi-platform systems: affinity, patterns, and fault-tolerance. An applications-driven research agenda will guide the development of the frameworks and validate our approach.

Broader Impact: This project will enable new data-intensive science by making it easier to construct multi-platform applications to integrate data and computation. Framework software will be put in the public domain. Significant education opportunities and programs will be provided for under-served groups, high-school students, and undergraduates through LSU's role in the Cybertools program.

Key Words: Data-intensive Computing, Distributed applications; Resource Management; Performance; Fault tolerance