Final Technical Report

Chase Qishi Wu
University of Memphis

Michelle Mengxia Zhu
Southern Illinois University, Carbondale

1. Project Objectives

A large number of high-performance computing (HPC)-based scientific applications feature experiment/computing/analysis workflows that can be modeled as simple as a linear pipeline or as complex as a directed acyclic graph. Oftentimes, these workflows operate on a colossal amount of data of extreme scales and must be executed in distributed environments with heterogeneous computing and networking resources.

Unfortunately, even if a computing workflow is well streamlined and configured, there still exists a bottleneck component (node or link) within the workflow, which could significantly limit the performance seen by end users if a static and ad-hoc system configuration is adopted as many commercial or open-source software tools do. Moreover, the system performance deteriorates even much faster as the size of scientific datasets rapidly grows. This project aims to identify such bottleneck components and optimize the execution of scientific computing workflows in heterogeneous network environments with guaranteed end-to-end latency, throughput, stability, and fault tolerance to facilitate and accelerate scientific discovery.

There are three different types of large-scale scientific datasets being generated in broad science communities: (i) real environmental data (satellite climate data for climate science, multimodal sensor data, etc.); (ii) simulation data (climate modeling, astrophysics, combustion, earth simulator, etc.); (iii) experimental data (Spallation Neutron Source, Large Hadron Collider, etc.). No matter which type of data is considered, one important task would be to develop end-to-end solutions for distributed users who are not located at the same site as the data source. To be more specific, researchers need to analyze, synthesize, organize, and transfer a colossal amount of data and it is crucial to streamline their computing workflows so that “their time is not eaten up by slow and repetitive chores”.

We will develop an application-support computing platform that integrates various component techniques to enable automated workflow streamlining and optimization. The ultimate goal of this project is to improve the productivity of scientists and the utilization of precious computing and networking resources by automating and optimizing scientific computing workflows in heterogeneous high-performance network environments. System components and end user requirements from several DOE-funded science projects such as climate modeling and high energy physics will be analyzed and supported via a set of customized software packages adapted from our general-purpose toolkits.

2. Revised Research and Development Plan and Schedule

The PIs are Prof. Mengxia Zhu at Southern Illinois University, Carbondale (SIUC) and Prof. Qishi Wu at University of Memphis (UM), both of whom have decades of extensive software development experience and strong research background in various computing and networking areas directly related to the research and development tasks in this project. Based on our prior results and a comprehensive comparison, we will choose C/C++ as the major programming language and Linux as the major development platform for this project. Several other programming languages will be used as well, for example, Java for AJAX web-based interface design and Fortran for scientific computations. The team members will also work very closely with application scientists throughout the 3-year project period and beyond to mature the software for production use. The following are a list of research and development tasks:

Year 1

1. Investigate and understand the networking and computing needs for DOE's mission-critical large-scale scientific applications in various disciplines. Particularly, both PIs have long-term partnerships and direct interactions with scientists in astrophysics and combustion research, and moreover, we have initiated contact with the climate modeling group at ORNL. We have made plans to visit ORNL and meet with scientists there on a regular basis to discuss and elucidate their application-specific computing workflows and performance requirements. (SIUC, UM)
2. Design and develop a preliminary software framework to support network-intensive scientific applications and determine its architecture, components, and functionalities. This preliminary framework features fixed workflow mapping or configuration schemes and manual deployment of computing modules for testing purposes. (UM, SIUC)
3. Design an event-driven control flow structure and define appropriate data structures, event types, and message formats for network path establishment and maintenance as well as module deployment, execution and cleanup to support distributed workflow configuration. (UM)
4. Design and implement a metadatabase describing the properties of various scientific datasets and a web-based graphical user interface using AJAX technology for smooth delivering and rendering of final results and convenient user access. (SIUC)
5. Construct analytical models to characterize the properties (such as complexities, memory requirements, etc.) of computing modules and inter-module dependencies in virtual task graphs or workflows as well as the properties (such as capabilities, reliabilities, etc.) of computer nodes and network paths in underlying overlay computer networks. Particularly, we will thoroughly examine application-specific computing techniques such as Principal Component Analysis, random matrix theory-based correlation computation, multivariate spatio-temporal clustering, eco-region identification in climate modeling and science, as well as visualization and computation monitoring and steering in other simulation-based scientific applications. (UM, SIUC)
6. Develop and implement performance characterization methods including multivariate least squares and artificial neural networks to estimate and predict module execution time in shared dynamic computing environments based on the analytical cost models. (SIUC)
7. Develop and implement regression-based active bandwidth measurement methods to estimate and predict data transfer time over shared layer 2 or 3 network paths. Note that the accuracy and timeliness of these time estimates for both data transfer and module execution are critical to support adaptive workflow reconfiguration. (UM)
8. Investigate the network services provided by OSCARS in ESnet. With OSCARS, we can either set up layer 3 (IP packets) paths or layer 2 (Ethernet VLAN packets) Virtual Circuits. For the later, it would require some coordination with the end-sites to configure the VLAN. We have been discussing with the OSCARS group and will work with them to set up VLAN channels for end users at various DOE sites including ORNL. (UM)

Main deliverables and milestones:

1. An event-driven control structure
2. A universal web-based user interface
3. A set of analytical cost models and performance characterization methods to estimate and predict module execution time with focus on climate modeling and high energy physics
4. An active bandwidth measurement software package to estimate and predict data transfer time over layer 2 or 3 network paths
5. Configuration and testing of IP paths (layer 3) and VLAN (layer 2) Virtual Circuits using OSCARS to reach end hosts at ORNL

Year 2

1. Refine the design of the workflow configuration framework to make it flexible for automatic module deployment and adaptive workflow reconfiguration. Suggestions, comments, and inputs from application users will assist in the framework refinement. (SIUC, UM)
2. Conduct an exhaustive categorization of computing workflow optimization problems based on optimization objectives, mapping constraints, network topologies, and resource uniformities and investigate their computational complexity. (UM)
3. Design and implement a set of efficient linear pipeline configuration methods for minimum end-to-end delay or maximum frame rate to support workflows with a linear arrangement of computing modules. (SIUC)
4. Design and implement a set of efficient heuristic graph mapping algorithms by expanding pipeline mapping algorithms and incorporating the critical path concept for general computing workflow optimization. (UM)
5. Develop a simulation-based formal method and implement a multi-threaded software program for workflow execution simulation. Due to the enormous scale of problems, structural complicacy of tasks, and wide distribution, vast diversity, and high dynamics of resources, implementing and deploying a massively distributed practical application in a real large network environment is a formidably expensive, time-consuming, and labor-intensive process. This simulation program will enable us to quickly and accurately test and evaluate the proposed solution especially the workflow optimization schemes and system stability before real deployment. (UM)
6. Build a prototype system by integrating the web-based user interface, event-driven control structure, time prediction components, and pipeline and graph mapping schemes into the refined workflow configuration framework. (SIUC)
7. Demonstrate the prototype system on a testbed involving supercomputers, PC clusters, and workstations connected through the campus network or the Internet (including ESnet). The demonstration will target scientific applications in climate modeling and high energy physics and the functions to be demonstrated include dynamic workflow partitioning and network mapping, remote visualization, and computational monitoring and steering. (UM, SIUC)

8. Summarize the preliminary results and release the first version of software package, whose open source license will be listed by the Open Source Initiative as an approved open source license. (SIUC)

Main deliverables and milestones:

1. A refined workflow configuration framework
2. A set of efficient and scalable algorithms to compute workflow mapping schemes
3. A prototype application-support system demonstrated over a testbed network
4. A simulation software package
5. First release of the system

Year 3

1. Implement, test, and evaluate component techniques developed in Years 1 and 2, and integrate them into a complete application/network interface solution within the workflow configuration framework. (SIUC, UM)
2. Investigate the stability issue of distributed computing workflows in heterogeneous network environments and determine the system steady state under various mapping constraints. (UM)
3. Investigate the system reliability issue and develop effective and efficient mechanisms to maximize system reliability while meeting end-to-end performance requirements. (SIUC)
4. Revisit the optimization, stability, and reliability problems using time-varying cost models and extend the mapping solutions and steady state results to the time-varying settings. (UM, SIUC)
5. Investigate different task scheduling policies (fair share, proportional share, etc.) to maximize end-to-end workflow performance. (UM)
6. Evaluate the performance of new and existing pipeline and graph workflow mapping algorithms based on implementation and experiments. (SIUC)
7. Test, measure, and compare the performance of the proposed optimization solutions and existing algorithms in the simulation program, verify the validity of analytical cost models, and improve mapping algorithms and refine cost models based on the simulation-based performance measurements. (UM, SIUC)
8. Integrate the refined workflow optimization algorithms with performance guarantees on end-to-end latency, throughput, stability, and reliability into the final framework. (SIUC)
9. Perform extensive experiments for network-intensive distributed scientific applications including climate modeling and high energy physics. (SIUC, UM)
10. Summarize the system design and implementation, algorithm evaluation and comparison, performance measurement, and experimental results and release the final system for public use. (SIUC)

Main deliverables and milestones:

1. Stability results for workflow mapping
2. A set of refined workflow mapping algorithms with guaranteed reliability and fault tolerance
3. Conducting large-scale experiments in real wide-area networks including OSCARS within ESnet

4. A set of software libraries containing API functions used by end users to automate and execute computing workflows
5. Final release of the system

3. Management Plan

This project is based on a close collaboration between SIUC and UM. The project team will participate in weekly teleconferences and monthly face-to-face meetings. Scientists will be invited to attend these meetings to discuss their specific application needs and interact with the project team. The computing technologies may be appropriately adapted and scoped based on the dynamic needs posed by different scientific applications. We will publish the results in the technical domains in peer-reviewed, open literature.

We will establish a dedicated website at SIUC to introduce the workflow configuration framework, report the research and development progress, and disseminate the software packages to the broad research community in a timely manner. We will also provide a public forum on this website for the general community of researchers to initiate communications with the project team directly.

In order for researchers to benefit the most from the proposed work, we will make our results widely available to broad scientific research communities. User friendly interfaces will be designed and developed to provide the latest high performance computing and networking technologies as an easy-to-use integrated middleware system to promote the adoption of our system in a wide spectrum of science projects. We will interact with scientists to identify the component technologies, develop comprehensive solutions, incorporate the solutions into the system, and make the system available to a wide range of users.