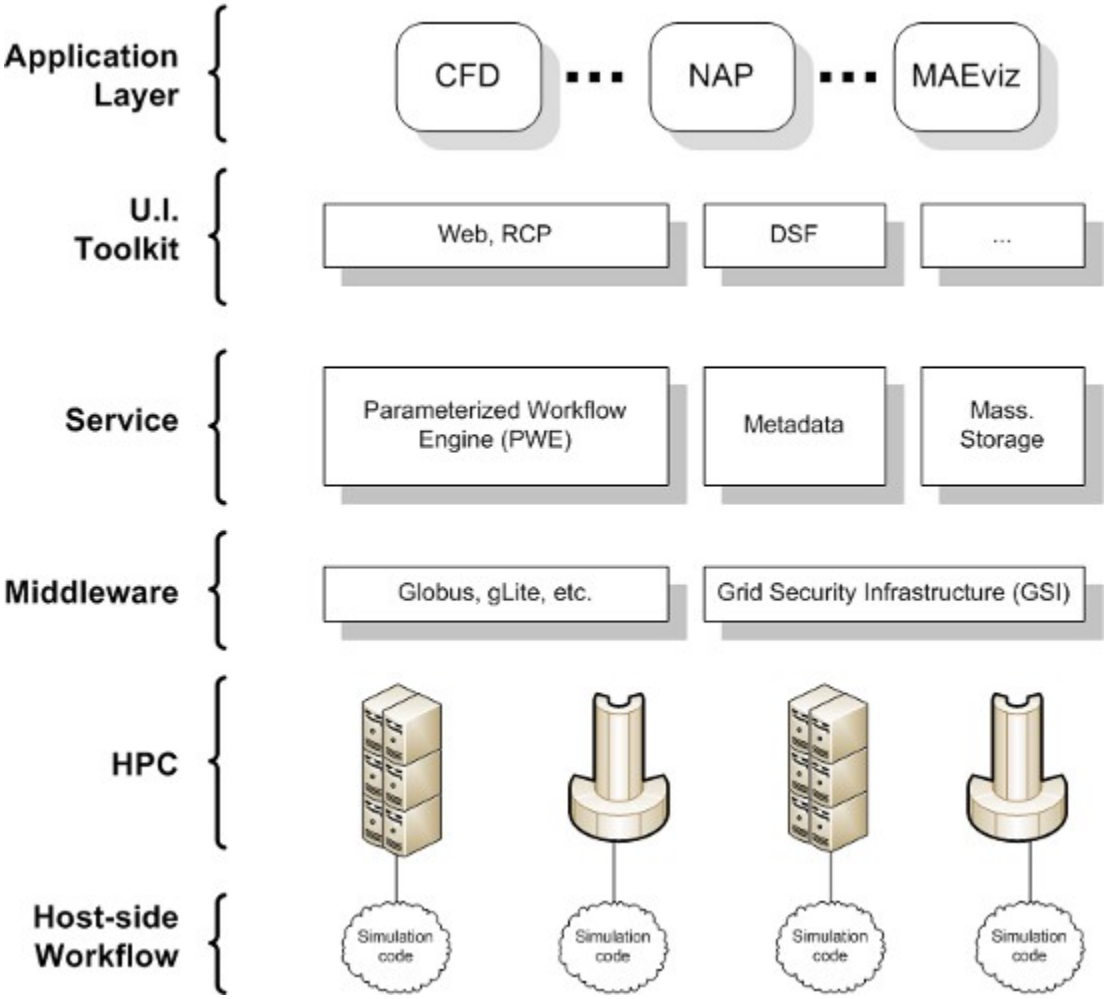
Architecture

System Architecture

In this section, the overall architecture for generating, editing, submitting, and monitoring high-performance computing workflows in an HPC environment is discussed. Each component shown in the figure below will be explained and identified whether it is in the scope of work for this project or not. The explanation of the components which are not in the scope of this project still remains for better understanding of possible future tasks and projects.



These technologies have been used extensively at NCSA for various projects and some are used widely in HPC applications; however, for those who are unfamiliar with any of these technologies, more detail will be provided in the sections that follow.

Application Layer

Applications are building using UI toolkits and can consist of both web and stand-alone applications

UI Toolkits

UI Toolkits provide the tools and widgets for building user interfaces of both web and stand-alone applications. Bard, Siege and the Digital Synthesis Framework are customizable UI Toolkits developed at NCSA that target the needs of different application types. The next few sections will describe each toolkit in more detail and provide examples of applications that use them.

Bard

Bard is a non domain specific toolkit for building extensible, configurable, and semantically aware applications with a consistent user interface. The Bard Analysis Framework uses Ogrescript, a scripting language for configuring, launching, monitoring and managing jobs on the desktop as well as in high-performance computing environments. The Analysis Framework User Interface helps facilitate the definition and creation of new analysis workflows and to explore new scientific possibilities by creating additional workflows from existing components. Bard delivers the exploratory capabilities which allow users to visualize results in 2D and 3D, create charts and tables, and publish reports. Bard also has a data catalog that gives users the ability to import, export, explore, and share data. Bard utilizes several other open source projects including Tupelo and the Eclipse Rich Client Platform (RCP).

Previous and Existing Uses

Currently Bard exists as the underlying framework upon which MAEviz [1] was built. MAEviz is an open source earthquake disaster management application that extends the Bard framework and provides a set of domain-specific plugins for the earthquake science community. MAEviz allows multiple scientific disciplines to work together to understand system interdependencies, validate results and present findings in a unified manner for the earthquake science domain. MAEviz helps bridge the time-from-discovery gap among researchers, practitioners, and decision makers. The MAEviz project has an intuitive graphical user interface that allows users to visually interact with workflows providing a better understanding of the inputs, outputs and readiness of the system for execution.

It is also under active development for the use in additional NCSA projects, such as the Medici project, as both a RCP (desktop) and web client.

Planned Extensions and Modifications

Several of the tasks proposed for this project will be modifying and extending the Bard platform to support HPC in general and the needs of KISTI in particular. Also, effort will be spent on integrating the current framework with other existing and developing projects to provide a common set of tools upon which consistent and dynamic scientific applications for both desktop and the web can be developed.

Siege

Siege [3] is intended to serve as a general-purpose work environment for application scientists wanting to run distributed workflows on production resources (such as on the Teragrid). Its current incarnation provides low level access to the creation of and submission of new or existing workflows as well as a set of sophisticated status monitoring tools. Seige has been under development for about five years and was initially developed as part of the Linked Environments for Atmospheric Discovery (LEAD) [2] program.

Siege will also play an important role in the workflow management infrastructure being developed for Blue Waters.

Siege is currently used on a regular basis by researchers from the University of Illinois' Atmospheric Sciences Department. To date, approximately 2,000 testing and production workflows have been submitted to the PWE via the Siege interface.

Siege was used as part of the 2006 Unidata Workshop. For more details see the report submitted to the American Meteorological Society.

Planned Extension and Modifications

The current version of Siege requires an intimate understanding of the inner workings of Siege, PWE, and the backend HPCs that the workflows will execute upon. This proposal calls for a far more user friendly user interface that must hide the complexity of the underlying system while not limiting access to the extensive capabilities provided.

The desktop version will be rewritten based upon the Bard toolkit as described above. A new product, Bard HPC, will be created with Siege as the initial desktop client. Common features of Bard HPC may also be used to create a web based job submission and monitoring interface.

Digital Synthesis Framework (DSF)

The Digital Synthesis Framework (DSF) provides a coherent framework for dynamically publishing visual analysis environments based on underlying observational and modeled information. The concept of a synthesis framework involves core capabilities for integrating data from multiple sources, enabling on-demand execution of scientific workflows, and the association of data outputs with multiple visualization and analysis widgets in a dynamically generated web application. In the DSF, NCSA's Cyberintegrator workflow environment is used to integrate data sources and invoke modeling modules. When the workflow is complete, it can be saved and run repeatedly as a service. A publication service allows the workflow outputs (which may be observational data or model outputs) to be associated with visualization widgets and embedded into a dynamically generated scenario viewer web application. The application can display data outputs from completed workflows or can trigger new workflows on demand. Along with maps, graphs, tables, and other displays, the application can display provenance information and links to associated reference material. As a concrete example, we present one of the TRECC pilot projects to incorporate a model predicting hypoxic (low oxygen) conditions in Corpus Christi Bay, Texas based on information from sensors deployed in and around the bay, into a hypoxia forecast web application [4].

Service Layer

Parameterized Workflow Engine (PWE)

The PWE is a workflow management system that has been incrementally developed over a period of eight years supporting high-performance applications in meteorology, chemical engineering, astronomy and even the analysis of financial markets. The entire process, while lengthy, has had a consistent set of goals.

- To assemble a loosely coupled set of components with well-defined functions such that we could extend them or compose into them other components as the need arose;
- To build an infrastructure above the lower middleware layer which would be general enough to suit a wide range of applications and not be tightly bound to the requirements of any single domain (along with this goal goes, needless to say, the requirement that application-level code remain insulated from this infrastructure; that is, that it need not be recompiled to work in our environment);
- o place high priority on productive and efficient interaction with the resource disposition typical of HPC centers, which more specifically means traditional batch queue managers; hence we have imposed a fundamental asynchronicity on the manner in which the highest level of the execution process must be handled (for instance, it would not make sense in this light that the component responsible for the execution logic hold the entire graph in memory for the duration of the workflow);
- To interact with existing resources such as mass storage systems and HPCs while requiring no modifications, additions, or installations on the remote resources.

Previous and Existing Uses

The PWE is currently used along with the Siege user interface.

Planned Extension and Modifications

The PWE is a robust, general system for the management of workflows and little effort should be required to modify the existing system to support the use cases provided for by this proposal. Logical extensions, however, may be required to suit the particular needs of both the back-end HPC systems and any Grid-middleware required accessing those systems.

Data Management

Scientific workflows can require an immense amount of archival storage. For this proposal, these storage resources must be identified and the appropriate access and security privileges must be given to both the developers and the end users. These resources must be able to be accessible by both the back end resources so jobs can store the data as well as by the client applications, whether they are web-based or desktop-based.

Previous and Existing Uses

Identify the resources that will be used on this project.

Planned Extensions and Modifications

It is assumed that the configuration and administration of these resources lie outside the purview of this proposal.

Metadata Management

Managing information about and generated by scientific applications is critical to the long term success of any project launching large number of jobs onto HPC resources. Merely storing the data is typically a straightforward process, but finding a particular data set in the black hole that can be a mass storage system can be frustrating and time consuming. To ensure the success of this project, care must be taken to develop a system to track the inputs, outputs, and provenance information generated by the scientific workflows so users can easily find and retrieve their data when needed.

Previous and Existing Uses

The use of metadata for management of scientific processes exists in many incarnations. At NCSA we are concentrating on the use of the Resource Description Framework (RDF) as a model for tracking this metadata, such as provenance and user submitted metadata (e.g. tagging and annotations.)

Planned Extensions and Modifications

This component is not in the scope of this project.

Event Management

The existing service stack uses both synchronous Java RMI calls as well as asynchronous Java Messaging Service (JMS) events to handle interprocess communications. A custom Event Service was developed to manage archival event storage as well providing a means for desktop clients which may go offline for extended periods to not miss any events.

Previous and Existing Uses

Both Java RMI and JMS are widely used across the globe. The custom Event Service is used along with PWE and Siege.

Planned Extensions and Modifications

The event service will not be extended or modified for this project.

Grid Middleware Layer

The primary mechanism for accessing the HPC resources with be via the PWE. In order to manage the submission of a potentially large amount of workflows over a potentially large number of geographically distributed resources, the PWE bypasses most of the Grid middleware specific functionality such as JobManagers. Instead, it relies heavily upon a few core components such as Grid Security Infrastructure (GSI) and GridFTP. This allows a more efficient use of both the backend resources (e.g. not overloading either the cluster headnode or the queuing systems) as well as allowing the PWE to support very large (thousands of nodes) workflows.

We plan no modifications or extension to the Grid Middleware Layer.

Globus/gLite/etc

The Globus Toolkit is a widely used Grid middleware project and is the toolkit running on the PRAGMA resources.

The gLite is the middleware stack for grid computing used by the CERN Large Hadron Collider (LHC) experiments and a very large variety of scientific domains.

Grid Security Infrastructure (GSI)

The Grid Security Infrastructure (GSI), formerly called the Globus Security Infrastructure, is a specification for secret, tamper-proof, delegatable communication between software in a grid computing environment. Secure, authenticatable communication is enabled using asymmetric encryption.

Host-side Workflow

Elf/Ogrescript

Elf is a robust container designed to support scripted applications. Elf's native scripting language is Ogrescript, but it can be extended to run other kinds of scripts (e.g. Perl or Python) as well.

Previous and Existing Uses

Elf/Ogrescript is currently used along with the PWE and the Siege user interface. See Bard section above for current usage.

Planned Extensions and Modifications

The core functionality of Elf/Ogrescript should require no changes for this project. Custom Elf scripts will be developed and tested for each HPC code required for ths project as consistent with the design of Elf/Ogrescript.

References

[1] James Myers, Terry McLaren, Chris Navarro, Jong Lee, Nathan Tolbert, B.F. Spencer, Amr Elnashai (2008), "MAEviz: Bridging the Time-from-discovery Gap between Seismic Research and Decision Making", UK e-Science AHM. Edinburgh, UK. Sept 8-11, 2008.

[2] Alameda, J., Hampton, S., Jewett, B., Rossi, A., Wilhelmson, B. (2006), "Ensemble Broker Service Oriented Architecture for LEAD", 22nd International Conference on Interactive Information Processing Systems for Meteorology, Oceanography, and Hydrology. Atlanta, GA. January 28 - Feburary 2, 2006.

[3] Alameda, J., Wilhelmson, R., Rossi, A., Hampton, S., Jewett, B., "Siege: A Graphical User Interface to Enable Management of Large Numbers of Weather Simulations"

[4] James D. Myers, Joe Futrelle, Jeff Gaynor, Joel Plutchak, Peter Bajcsy, Jason Kastner, Kailash Kotwani, Jong Sung Lee, Luigi Marini, Rob Kooper, Robert E. McGrath, Terry McLaren, Alejandro Rodriguez, Yong Liu (2008), "Embedding Data in Knowledge Spaces". Presented at Workshop 9: The Global Data Centric View, UK e-Science AHM. Edinburgh, UK. Sept 8-11, 2008.