

## **SIEGE: A GRAPHICAL USER INTERFACE TO ENABLE MANAGEMENT OF LARGE NUMBERS OF WEATHER SIMULATIONS**

Jay Alameda, Robert Wilhelmson, Albert Rossi, Shawn Hampton  
National Center for Supercomputing Applications, University of Illinois at Urbana-Champaign,  
Urbana, IL 61801

Brian Jewett  
Department of Atmospheric Sciences, University of Illinois at Urbana-Champaign, Urbana, IL 61801

Thomas Baltzer, Anne Wilson  
Unidata Program Center, UCAR, Boulder, CO

### **1. INTRODUCTION**

In the LEAD project, we have focused on the most challenging users of numerical weather prediction, namely, the atmospheric science researchers, who are prone to use their own tools, their own modified versions of community codes such as the Weather Research and Forecasting (WRF) model, and are typically comfortable with elaborate shell scripts to perform the work they find to be necessary to succeed, to drive our development efforts. We will describe our graphical user interface, Siege, which we can adapt to a multiplicity of user scenarios, and is coupled to a multi-level workflow engine, to handle both the challenges of ensemble description and execution, as well as the detailed patterns of workflow on each computational resource; services to support the peculiarities of each platform being used to do the modeling (such as on TeraGrid), and the use of an RDF triple store and message bus together as the backbone of our notification, logging, and metadata infrastructure. The design of our ensemble management software and services attempts to come to grips with lack of control of elements surrounding and supporting the environment; we achieve this through multiple mechanisms including using the OSGI plug-in architecture, as well as the use of RDF triples as our finest-grain descriptive element. This combination, we believe, is an important,

---

*Corresponding author address:* Jay Alameda,  
National Center for Supercomputing  
Applications, University of Illinois at Urbana-  
Champaign, Urbana, IL, 61801:  
[jalameda@ncsa.uiuc.edu](mailto:jalameda@ncsa.uiuc.edu)

stepping stone to building a cyberenvironment which aims to provides flexibility and ease of use far beyond the current range of typical problem solving environments.

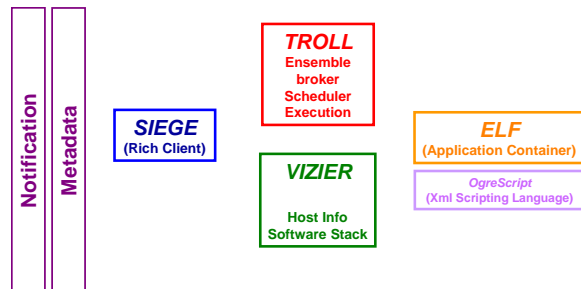
### **2. PROBLEM DESCRIPTION**

Computational atmospheric science researchers present a methodology of work that taxes the ability of conventional web portals to cleanly solve. For instance, the advanced researcher will often want to work with either a modified community code such as the Weather Research and Forecasting (WRF) [WRF] code, or their own code designed to solve the problems of interest. This causes problems for portals that provide access to high end computational resources such as those provided by TeraGrid [TeraGrid], as normally resource providers balk at providing a community user account unfettered access to their resources without full vetting of the user (which is normally done in the context of conventional peer-reviewed TeraGrid allocations). This prevents the user from using their own code on the resource, through a conventional web portal. Also, the researcher typically would have their own allocation to use; so using a community allocation would not allow the user to consume the resources awarded through the normal peer-review process. Such researchers often want to easily move data to and from their workstation, for data preparation tasks and post-simulation analyses which can be most easily accomplished on their desktop compute platform. They also want to be able to cleanly extend the computational experiments they are performing, either by adding parameters that need to be examined, or organizing their results by attributes newly

discovered to be important, or changing the pattern of work to be accomplished based on new hypotheses and insights into the computational problem. Thus, for this class of users, being able to easily modify the orchestration needed to accomplish the desired science is essential.

### 3. SIEGE: DESKTOP ACCESS TO A SERVICE-ORIENTED ARCHITECTURE.

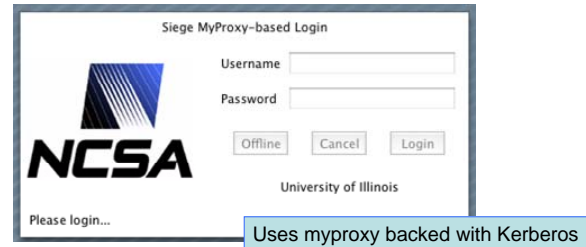
To properly address the requirements briefly outlined in section 2, we developed a complementary service-oriented architecture, driven by a desktop client that we denote as "Siege"[Siege]. The service-oriented architecture that we use in the system, depicted in figure 1, shows the desktop client, Siege, which interacts with the Troll ensemble broker stack and Vizier information services, to deploy applications controlled by the remote application container, ELF, which implements our own scripting language, OgreScript. Tying the system together are the notification systems, currently using the Java Messaging Service (JMS) [JMS] channel ActiveMQ [ActiveMQ], and



**Figure 1: The Siege Service-Oriented Architecture for Ensemble Brokering** a metadata system (we are planning to integrate myLEAD into the JMS channel, to be able to create LEAD metadata objects [LMS] from metadata events published to the channel).

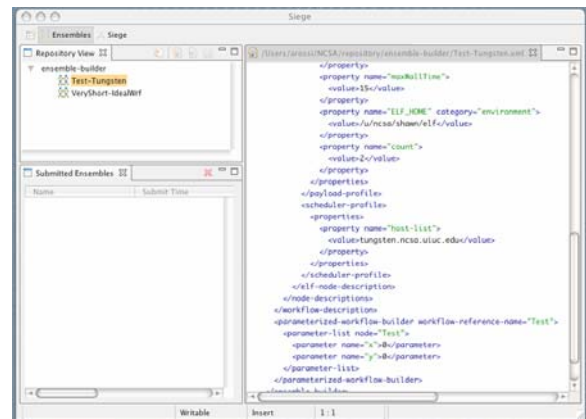
The Siege desktop client is built using the Eclipse Rich Client Platform (RCP) [RCP], which has many advantages including ease of interface mockup, pluggable modules, and well-defined extension points. We are also using RCP to build our services, as it provides a nice modularity to better manage dependencies, especially on third-party libraries such as the jglobus [jglobus] client libraries that we use to access grid capabilities such as job submission and file management.

With Siege, the user authenticates to a myproxy server backed with a Kerberos realm [myproxy] to allow delegation of short term grid credentials to the Siege client by mere use of a user's Kerberos login at NCSA or on TeraGrid. This login, which is granted as normal part of a user's allocation at NCSA, allows users seamless access to computational resources through Siege, by use of their familiar Kerberos login (as depicted in Figure 2).



**Figure 2: Siege MyProxy Login, currently using Kerberos authentication for NCSA and TeraGrid.**

Siege provides simple mechanisms for describing, executing and monitoring workflows. For instance, a user can directly edit the xml description of the workflow, as depicted in

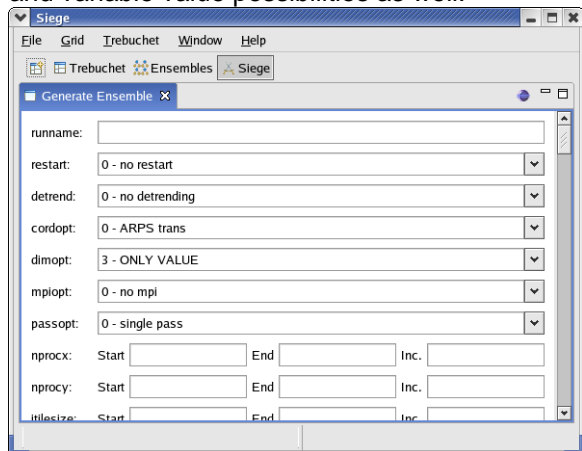


**Figure 3: Direct manipulation of XML-based ensemble description.**

Figure 3. One such user prefers this, as the user can directly manipulate ranges of parameters to be explored in a parameter space study. The XML description of the ensemble is expanded into the full graph of execution at job submission time.

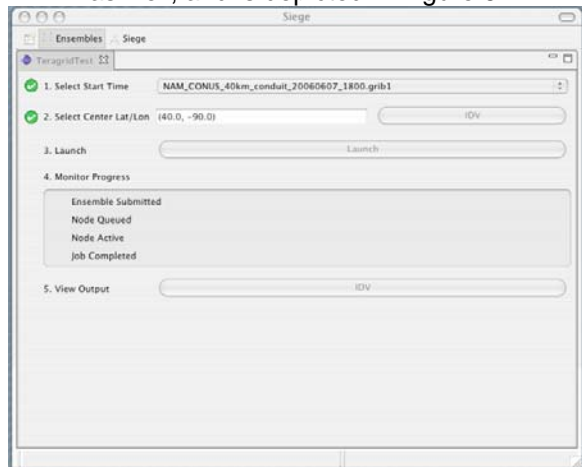
We are also prototyping user interfaces which guide the user to describe ranges and intervals of variables to be studied, in this case for a research weather code. The prototype, shown

in Figure 4, depicts indication of default ranges and variable value possibilities as well.



**Figure 4: Ensemble Generator Interface for a weather research code**

Finally, we have prototyped a specific interface which is designed to allow the user to use Unidata's Integrated Data Viewer (IDV) [IDV] to select the center of a domain to model the atmosphere using WRF, and then simply launch and monitor the resulting model on TeraGrid resources. This interface, the use of which is described in paper 8B.6, was integrated to allow straightforward visualization of the model results in IDV as well, and is depicted in Figure 5.



**Figure 5: Simplified Unidata Workshop facility, which is integrated with the Unidata IDV, and allows simple selection of domain center latitude and longitude, as well as the NAM dataset for model initialization.**

## 5. RESULTS

We have been able to use the Siege user interface and service architecture to manage

runs made in the course of the Unidata workshop in July 2006 [UnidataWorkshop], as well as to support one of the coauthors (Jewett) work in understanding parametric sensitivities for interacting thunderstorms. Through this, we have observed that we have a number of issues to address with respect to scalability and robustness – in no particular order:

- in-memory service state is hard to manage, we need to persist all such state to a relational database so that services can be brought back up seamlessly without loss of information
- gridFTP servers appear to have an issue handling many clients, with difficult-to-diagnose failures as a result
- the web services in the stack need to be few in number, as well as as close to transactional as possible, with small items being used per interaction rather than complex objects

But, even with the issues identified for resolution, we were able to successfully support the modeling efforts of the workshop attendees as well as our coauthor, and have charted a path for improved performance and scalability of the underlying service stack. We also have shown that the Eclipse Rich Client Platform provides a powerful, flexible alternative user interface which integrates well with a user's desktop platform. The use of JMS for messaging has also proven to be a good choice, for its inherent flexibility and malleability to a variety of other messaging systems, as well as its performance, and finally, the ability to perform cluster-local orchestration readily facilitates local monitoring of batch processes.

## 6.0 ACKNOWLEDGEMENTS

LEAD is a Large Information Technology Research (ITR) Grant funded by the National Science Foundation under the following Cooperative Agreements: ATM-0331594 (University of Oklahoma), ATM-0331591 (Colorado State University), ATM-0331574 (Millersville University), ATM-0331480 (Indiana University), ATM-0331579 (University of Alabama in Huntsville), ATM03-31586 (Howard University), ATM-0331587 (University Corporation for Atmospheric Research), and ATM-0331578 (University of Illinois at Urbana-Champaign, with a sub-contract to the University of North Carolina). This work has also been supported by SCI03-30554, SCI04-38712, and

SCI96-19019. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the National Science Foundation.

## 7. REFERENCES

[WRF] The Weather Research&Forecasting Website, <http://www.wrf-model.org/index.php>.

[TeraGrid] TeraGrid, <http://www.teragrid.org/>.

[Siege] Siege – MRD-Public – Confluence, <http://torcida.ncsa.uiuc.edu:8080/confluence/display/MRDPUB/Siege>

[JMS] Java Message Service, <http://java.sun.com/products/jms/>

[ActiveMQ] ActiveMQ Home, <http://www.activemq.org/site/home.html>

[LMS] LEAD Metadata Schema Repository, <http://www.extreme.indiana.edu/rescat/metadata/>.

[RCP], Rich Client Platform, [http://wiki.eclipse.org/index.php/Rich\\_Client\\_Platform](http://wiki.eclipse.org/index.php/Rich_Client_Platform).

[jglobus] CoG jglobus, [http://dev.globus.org/wiki/CoG\\_jglobus](http://dev.globus.org/wiki/CoG_jglobus).

[myproxy] Myproxy Credential Management Service, <http://grid.ncsa.uiuc.edu/myproxy/>

[IDV] Unidata Integrated Data Viewer (IDV), <http://www.unidata.ucar.edu/software/idv/>.

[UnidataWorkshop] 2006 Unidata Users Workshop: Expanding the Use of Models as Educational Tools in the Atmospheric & Related Sciences, <http://www.unidata.ucar.edu/community/2006workshop/>