How the NCSA/LEAD Workflow Engine Manages Complex Workflows

Jay C. Alameda¹, Albert L. Rossi¹, Shawn D. Hampton¹, Brian F. Jewett², and Robert B. Wilhelmson¹,²

Univ. of Illinois: ¹Nat'l Center for Supercomputing Applications (NCSA) ²Department of Atmospheric Sciences

System Architecture

Includes Rich Client Platform front-end (SIEGE), workflow and Information services (PWE/VIZIER), resource-resident application container and scripting language (ELF/Ogrescript), message bus and relay agents.

PWE Internals

The Parametric Workflow Engine manages workflow state via a fixed number of threaded queues; state is persisted to a database for asynchronous handling. PWE receives status updates directly from ELF as well as through the monitoring or polling of the computational resources.

ELF “Glide-In”

PWE expands, configures and submits parametric nodes through the glide-in container mechanism, distributing parametric configurations via a tuple-space service. When this container begins to run, it retrieves these configurations and launches their corresponding members (e.g., WRF jobs).

Siege (I)

After submitting a workflow, the PWE perspective allows drilling down to get individual node-state information.

Siege (II)

Information concerning users and resources can be conveniently configured through the Vizier perspective show below.

Example of the Workflow Description XML (partial) submitted to PWE. These are stored locally on the user’s machine and can be edited through Siege.

Workflow engine development and testing has been supported by LEAD, a NSF ITR (ATM-0331578 and others), and SCI03-30554, SCI04-38712, and SCI96-19019.

Research efforts within the LEAD (Linked Environments for Atmospheric Discovery) program include workflow orchestration and fault tolerance for use with WRF; data mining; and on-demand and adaptive computing. For more info: <leadproject.org>