A Workflow Management System for Automated Weather and Climate Simulations

Christopher W Harrop¹, Ligia Bernardet¹, Mark Govett¹, Jeff S Smith¹,³, Stephen Weygandt¹

NOAA - Earth System Research Laboratory - Global Systems Division¹, Cooperative Institute for Research in Environmental Sciences², Cooperative Institute for Research in the Atmosphere³, Systems Research Group⁴

Introduction
One of the challenges of modeling and simulating weather and climate phenomena is the onerous process of running the various codes that are required to carry out a simulation. There are several factors that contribute to this difficulty:

- An end-to-end run of a simulation often requires running a large number of components.
- The simulation components can have complex interdependencies and runtime requirements.
- Simulations that run in real-time must complete before the resulting forecasts become irrelevant.
- Many experiments require 100’s or even 1000’s of end-to-end runs.
- High Performance Computing systems are notoriously difficult to use and are not always reliable.
- Monitoring and tasking simulation progress is labor-intensive and prone to error.
- Ad hoc scripting methods for end-to-end simulation orchestration are not feasible.

A Conceptual View of a Realistic Complex Large-Scale Workflow

The Workflow Management System

Definition: A Scientific Workflow Management System is software that helps scientists compose scientific workflows and automate their execution.

Definition: A scientific workflow is a collection of data and computational tasks along with a description of their interrelationships and runtime requirements.

A workflow management system provides scientists with two key functions:

1. A mechanism for composing workflows, and
2. An engine for managing workflow execution.

Composing Workflows

We have devised a custom XML-based language for composing workflows. The language is simple, non-erotic, and uses terms meaningful to most scientists. It consists of a set of "tags" that are used to describe workflow structure. Scientists use an editor to arrange the tags and set the tags' attributes to define the structure of their particular workflow.

The Workflow Management Engine

The Workflow Manager has a layered, object-oriented architecture. Layers communicate with each other via standard interfaces. Support for additional batch systems is currently available by implementing a new batch system class in the Batch System Layer.

The workflow XML document is prepared by the scientist using his or her editor of choice.

The workflow XML document, instantiates Task, Cycle, Batch System, and other objects, maintains a list of active cycles, runs the workflow by invoking each Task object's run method for each active cycle.

Maintains Task state, renames files, runs tasks by instantiating Job objects to do the tasks' actual work, and invokes Job object methods to manipulate and query the status of batch jobs.

Maintains Job state, invokes Batch System object methods to submit, query, and cancel batch jobs.

The workflow engine submits jobs to the HPC system. Once submitted, the batch system on the supercomputer is solely responsible for scheduling the jobs. The Workflow Manager can submit, monitor, and cancel jobs, but it relies on the underlying batch queuing system to schedule their execution.

Using the Workflow Manager

The algorithm implemented by the workflow manager is described below. Each time it runs, it detects task failures and newly satisfied task dependencies and starts or renews tasks accordingly. Thus, repeated invocations (usually as a cron job) of the workflow manager make incremental progress until the workflow is complete. The current state of the workflow is maintained in permanent storage between runs, and each run usually finishes in a few seconds.

Command Line Usage:

```
$ workflowmgr -b -xml=/path/to/workflow.xml -user=root -n=startWorkflow
```

Crontab Usage:

```
*/2 * * * * workflowmgr -b -xml=/path/to/workflow.xml -user=root -n=startWorkflow
```

A Core Component of WRF Portal

Our workflow manager also provides core functionality for WRF Portal. WRF Portal is a Java application that simplifies model testing by providing modelers with an intuitive graphical user interface for composing, running, and monitoring workflows. Using WRF Portal, scientists can easily create, organize, and track large numbers of model configurations, model runs, and visualize the results. The diagram below shows how the workflow manager is used by WRF Portal to coordinate the execution of simulation components.

Practical Experience

Our workflow manager is designed to solve the practical simulation management problems that scientists face on a daily basis. This means we focus on solving the problems that scientists encounter during their real-world modeling experiences rather than focusing only on “grand challenge” applications. As a result of this philosophy, our system has played a key role in the success of several real-world modeling applications. Below is a list of some recent experiments that made use of our workflow manager.

<table>
<thead>
<tr>
<th>Experiment Name</th>
<th>Type</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>WRF Test Plan</td>
<td>Remispective</td>
<td>Remote</td>
</tr>
<tr>
<td>WRF Rapid Refresh Core Test</td>
<td>Remispective</td>
<td>Realtime</td>
</tr>
<tr>
<td>DTV Weather Forecasting Experiment (DWFEX)</td>
<td>Realtime</td>
<td>Remote</td>
</tr>
<tr>
<td>NMM5-COSYS</td>
<td>Realtime</td>
<td>Remote</td>
</tr>
<tr>
<td>Terrain-induced Ecosystem (T-ICEX)</td>
<td>Remispective &amp; Realtime</td>
<td>Realtime</td>
</tr>
<tr>
<td>Rapid Refresh Development</td>
<td>Realtime</td>
<td>Remote</td>
</tr>
<tr>
<td>Rapid Update Cycle Benchmark</td>
<td>Realtime</td>
<td>Remote</td>
</tr>
</tbody>
</table>

Conclusions

- The success of our system has been demonstrated by its practical use in several real-world experiments.
- A focus on solving daily practical issues results in a greater amount of “buy in” by the end-user scientists.
- Many scientists now consider our system to be a crucial tool for conducting their experiments.
- Our system has dramatically increased the reliability and efficiency of several experiments.
- We have plans to make the following improvements:
  - Add workflow language constructs to make it easier to create and maintain a large number of nearly identical tasks.
  - Add quality of service workflow language constructs to enable the specification of start and end deadlines for real-time task execution.
  - Generalize task runtime property specification to allow workflow documents to be portable across different batch systems.
  - Implement the capability to “roll back” a workflow so that a scientist can make changes and resume tasks that have already completed successfully (currently it is only possible to resume tasks that have failed).