The Trident Scientific Workflow Workbench

R.S. Barga, J. Jackson, N. Araujo, D. Guo, N. Gautam, Y. Simmhan
Microsoft Research, Microsoft Corp.
barga@microsoft.com

Abstract

We introduce Trident, a scientific workflow workbench that is built on top of a commercial workflow system to leverage existing functionality. Trident is being developed in collaboration with the scientific community for use in a number of ongoing escience projects that make use of scientific workflows.

Introduction

In designing Trident, our goal was to leverage existing functionality of a commercial workflow management system to the extent possible and focus our development efforts only on functionality required to support scientific workflow. The result is a smaller code base to maintain going forward, improving sustainability and manageability of the project, and an improved understanding of requirements unique to scientific workflow.

Trident is implemented on top of Windows Workflow (WF) [1], a workflow enactment engine included at no additional cost in the Windows operating system. All *activities* in WF are derived from *System.Workflow.ComponentModel.Activity* base class. The Windows WF extensible development model enables the creation of domain specific activities which can then be used to compose workflows that are useful and understandable by domain scientists.

The key elements of the Trident architecture, illustrated in Figure 1, include a visual composer and library that enable scientists to visually author a workflow using a catalog of existing activities and complete workflows. The Trident registry serves as a catalog of known data sets, services, workflows and activities, and compute resources, as well as maintaining state for all active workflows. An execution engine supports launching workflows remotely and according to a schedule. Admin tools are provided to allow users to register and manage computational resources, publish workflows for external use, and track all workflows currently running or recently completed. Users can also schedule and queue workflow execution based on time, resource availability, etc. A set of community tools includes a web service that enables users to launch workflows from any web browser and a repository that facilitates the publishing and sharing of workflows and workflow results with other scientists which integrates with myExperiement [2]. At the lowest level of Trident is a data access layer that abstracts the actual storage service that is in use from the running workflows. The data access layer is extensible and currently Trident supports a default XML store and SQL Server for local storage, and Amazon S3 [3] and SQL Server Data Services (SSDS) [4] for cloud storage.

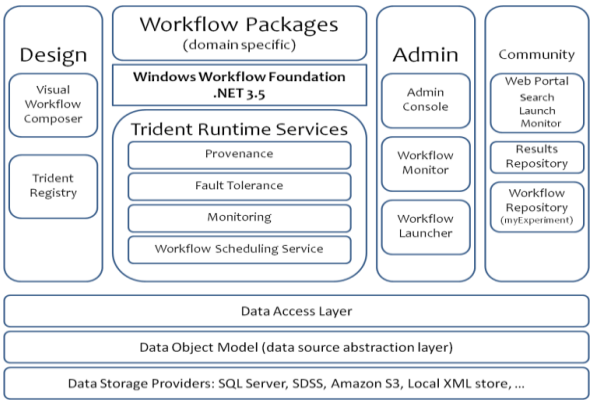

Fig. 1. Diagram of the Trident logical architecture.

WF provides several runtime services which can be used as required by attaching the service implementation to the workflow runtime. Two of the most useful for our implementation of Trident are:

- Tracking service: This service enables event based tracking of a running workflow through the use of extensible tracking profiles.
- Persistence service: This service allows the workflow executor to serialize and restore the entire working state of an in-progress workflow, allowing the executor to pause and resume workflows and archive intermediate state to any capable storage device.

Additional services can be constructed to run alongside these basic services. Our current implementation of Trident includes a service for automatic provenance capture, a monitoring service that listens for events pertaining to machine utilization, resources available, etc., a service that schedules workflows on HPC clusters, and a fault-tolerance and recovery service for workflows.

What will be demonstrated?

We will demonstrate how Trident implements actual workflows for both Pan-STARRS [5] and NEPTUNE [6] projects. In addition, an accompanying poster will outline exactly how we leverage a commercial workflow enactment engine to support scientific workflows in general. Highlights of the demonstration are described in the remainder of this submission.

Features of the Trident Registry

The Trident registry consists of a series of modules and abstractions to provide great flexibility to the scientists as to where they actually store their data. During our demo we will illustrate the following

- Trident allows the user to dynamically select where to store data (results) output from a workflow, such as SQL Server, Amazon S3, SSDS, etc.
- A **data provider** abstraction that allows actual data contents to be referenced to external entities, allowing scientists to host their data anywhere (external data stores, community databases or servers, etc).
- Strong typing of objects referenced and stored by the Registry reduces runtime issues common to software development, leading to a more robust system
- Programming APIs that allow workflows being executed to record experiment results in the registry in a consistent and organized way. Scientists or services can later navigate through this data to implement new functionality.

Scheduling Workflows in Trident

In our demo we will illustrate that Trident provides:

- The ability to schedule workflows to run on any machine, or collection of machines, from a single and easy to use console (local or web based).
- Ability to schedule entire workflows and individual activities on an HPC cluster.
- Scheduling that takes workflow compute and data requirements into consideration, and is aware of resource utilization (CPU, databases, disks, I/O, memory) within a cluster to optimize scheduling and support job priorities
- Ability to pause, resume, stop and restart specific workflows and entire queues on specific machines
- Ability to recover from failures and take corrective actions when workflow execution does not go as expected

Provenance and Monitoring

Trident adds a publication/subscription mechanism called the Blackboard that utilizes custom and built-in WF tracking services to provide extensible workflow monitoring and provenance support. This model allows for both evolutionary and runtime provenance and enables:

- Customizable logging for analysis and recovery
- Reporting and visualizations of intermediate data products from a running workflow
- Provenance record capture either locally or in the cloud
- Fault tolerance messaging and repair
- Workflow execution monitoring with resource usage analysis and intelligent completion estimates

Web Services and Portal

In addition to providing client application tools to facilitate scientific workflows, a library of web services are included that allows access to Trident's key features, including access to repositories of workflows, ability to launch and monitor workflows remotely, and integration with repositories and scientific networking sites outside of Trident. While workflow execution must still be done in a .Net capable environment, these web services allow access to the features of Trident from any platform connected to the internet. Oceanographers at the University of Washington have already integrated their underwater visualization tool COVE [7] with Trident workflows using this mechanism.

Trident includes a web portal written in Silverlight [8] that allows scientists to launch and manage workflows from any internet location. The portal works with a variety of browsers running on Windows, Max OS, or Linux.

REFERENCES

[1] Microsoft Windows Workflow Foundation (WinWF) http://en.wikipedia.org/wiki/Windows_Workflow_Foundation.

[2] MyExperiment, http://www.myexperiment.org.

[3] Amazon S3 Web Service http://aws.**amazon**.com/**s3.**

[4] Microsoft SQL Server Data Services (SSDS) www.microsoft.com/**sql**/dataservices/default.mspx.

[5] Pan-STARRS – Panoramic Survey Telescope & Rapid Response System, http://pan-starrs.ifa.hawaii.edu/public/

[6] Project Neptune http://www.neptune.washington.edu/.

[7] COVE Oceanographic Visualization Workbench http://www.cs.washington.edu/homes/keithg/oceans.html.

[8] Microsoft Silverlight http://silverlight.net/.