DOME and Federated Scheduling

Brian Lynn
September 2009

Introduction

A short while ago there were discussions between the DOME project and the GPO regarding the implications of a federated resource management scheme on DOME and its framework, ORCA. It was agreed that we would resume the discussion when the new GENI fiscal year began. The purpose of this paper is to reinitiate that dialogue. We realize that there are other federation discussions taking place, and that the framework architects are already addressing many of the issues. The objective of this paper is to complement those discussions and to provide a perspective from one substrate.

DOME Overview

At the highest level there are two things that need to be identified when a user schedules time on the DOME testbed:

- Buses
- Experiments

Currently DOME's granularity is 100% of the buses, i.e., you reserve all of the buses or none. There are discussions regarding allowing users to reserve a portion of the testbed, but that is not relevant within the scope of this paper.

An experiment is defined by DOME as a collection of one or two user-defined files representing disk partitions that will be mounted by a VM. Experiments have the following properties:

- Experiments are transient; they can be created or destroyed on demand.
- Their usage is restricted. Experiments are associated with a user. User B must not be able to schedule a bus with User A's experiments.

The above characteristics make an experiment different from a component such as the buses, whose properties are known in advance and can be represented in a reasonably static configuration file.

DOME may be an exception in its association of experiments with the reservation of the component. The configuration of other testbeds is often limited, such as selecting one of the component's predefined VM images. Most testbeds assume that a user will be able to install experiments or apply further customization, such as the installation of user-specific software or a customized kernel, after the component has been allocated. An underlying assumption is that the user will have almost unrestricted access to the component, and that the time and effort involved in customizing the testbed environment will not prohibitively impact the ability of a researcher to perform his or her experiments.

But DOME's environment is different from most testbeds.

- Network connectivity is not always available and is often unreliable. We have measured the 3G network to have on average less that 90% availability. Numerous disconnections can occur, even while still achieving 90% availability.
- IP addresses are dynamically assigned by the cellular network and typically fire-walled, disallowing inbound connections. Tunneling over reverse ssh connections is feasible, but it requires coordination with a host.
- The quality of the link varies. Even though we have 3G modems they are often required to connect using EDGE or GPRS technology.
- The buses are not always on. Even during peak hours the full complement of buses is not operable, buses don't run in the middle of the night, buses are turned off between shifts, and anti-idling laws require that buses sometimes be turned off at stops.
- Experiments often have small windows of productivity, such as when two buses are within proximity to each other. It's critical that any allocated time be used efficiently.

To work within the limitations described above and to maximize the utilization of DOME resources, DOME has implemented the following.

- A "DOME portal" has been provided to stage experiments, interact with buses, and provide an “always on” interface to ORCA.
- Users log into VMs, install software and customize the VM's configuration prior to scheduling time on the DOME testbed. This is done using stationary or "yarded" bricks. Once a user has readied a VM, the VM's disk images are saved and uploaded to the DOME portal. The disk images are loaded as partitions to a user's VM on a bus.
- Experiments (disk images) are proactively downloaded onto the buses. DOME does not wait until an experiment is scheduled before copying the partitions onto a bus.
- If a bus is scheduled to run an experiment but the full experiment is not on the bus, the VM is not started. Actually, the experiment cannot be started because all of its disk drives cannot be mounted. Instead, the bus will run a set of "default experiments," collecting longitudinal data until the software is downloaded.

DOME Scheduling

In order to schedule time on the DOME testbed a user must use the DOME portal. The steps are summarized below.

- The user uploads the validated disk partitions to the DOME portal. The portal places the files in a location that the buses will be able to access.

- The user creates an experiment. An experiment identifies which partition files will be used by a VM.
    - Any file associated with a defined experiment will be downloaded by the buses, space permitting. Experiments that are not scheduled are given a lower download priority than those that are scheduled.
- Once an experiment has been defined it can be scheduled. From the DOME portal a user specifies the experiment to be run, how much of the testbed to reserve (currently 100%), the duration (the minimum slice is 1 hour), and the earliest start time (to allow for download/staging time or to target peak operation).
    - Once the user requests access to the testbed, the DOME portal registers a pending lease in its database and sends an XML-RPC request to the DieselNet controller (an ORCA package). Included in the request is information such as the experiment identifier, the duration and early start time.
    - The DieselNet controller does not submit the request to ORCA until two conditions are met: (1) the current time is not before the early start time, and (2) the resources are not allocated to another experiment. The DieselNet controller holds onto the lease request until it knows ORCA will immediately honor the request so that the controller can implement canceling of requests. This logic will be removed once ORCA supports canceling tickets, but the net result is that the DieselNet controller is currently responsible for the scheduling policy.
    - When the controller submits the request into ORCA it attaches the XML-RPC parameters as opaque values to the request. When the lease begins, ORCA invokes an ant script marking the "join" event. The ant script passes the opaque values to the DieselNet handler.
    - The DieselNet handler sends an XML-RPC message to the DOME portal noting the start of the lease. The parameters to the XML-RPC method are the opaque values.
    - The DOME portal changes the state of the pending lease to active.
    - Buses periodically query the DOME portal for active leases. Each bus notes the lease and a VM is launched with the specified experiment.
- When the lease expires ORCA again executes an ant script, this time marking a "leave" event.
    - The ant script invokes the DieselNet handler, again with the opaque values.
    - The DieselNet handler sends an XML-RPC message to the DOME portal to indicate the lease has terminated.
    - Buses query the portal and see that the lease is no longer active. The VM is terminated.

Federated Scheduling

A goal of GENI is to federate the reservation of testbed resources. The vision that has been presented to us is a researcher sitting down at a computer and reserving resources on multiple testbeds, with the underlying control framework(s) ensuring that the tickets for the resources be honored, that the slices for each testbed occur simultaneously, and that communication channels exist between the testbeds.

In this section we discuss, from the perspective of DOME, what we believe would be required to allow a control framework to reserve access to DOME in conjunction with non-DOME testbeds. By no means are we trying to dictate how things should work; we are simply providing input based on our implementation and experiences. The other objectives, ensuring that tickets be honored and establishing communication channels, are beyond the scope of this paper. We use the existing interaction between DOME and ORCA as the starting point. The assumption in the discussion below is that some third-party application exists to reserve simultaneous access to multiple testbeds.

Uploading DOME VM Partitions

This should continue to operate as it does today. A user would log onto the DOME portal and upload files to the DOME server. There is no need to delegate this functionality to a surrogate. A necessary change would be that logging into the DOME portal should involve authentication with some authority, e.g., ORCA.

Creating DOME Experiments

The creation of an experiment should continue to be done via the DOME portal. DOME experiments are, of course, specific to the DOME testbed. Since we need to associate an experiment with a DOME testbed reservation request, the experiment needs to somehow be visible to and represented by the third-party application. Our current thoughts are that DOME should be able to present the experiment and dependency to ORCA, and that ORCA would provide the information to the third-party application. Unless there are compelling reasons, it seems that the third-party application should only know of DOME's surrogate, ORCA. In addition, we would want to be able to assign an access control list to the experiment, to restrict access to certain users. Also required is the ability to remove the visibility of the experiment should a user delete it.

Scheduling Access to the DOME Testbed

The ability to initiate scheduling of the testbed would need to be delegated to the third-party application. The DieselNet controller XML-RPC facility might continue to exist as a convenience interface for the DOME portal, but it should not be the interface contacted by third-party applications. There is no means to express additional dependencies of non-DOME components, nor does it make sense to add such functionality to the DieselNet controller. Our view is that ORCA would need to provide a substrate-independent interface and authenticate the third-party application.

Even though the controller's XML-RPC interface may no longer be used, there is still significant value to having a DieselNet controller. Specifically, the state event notifications (onTicket, onLease, etc.) should continue to exist. The controller could learn about the resource requests through these state events. This information could be used, for example, to prioritize the downloading of experiments to buses.

Since there is a need to cancel requests, there is also a need to identify and reference requests.

Because experiments are limited to certain users, the authentication chain would need to be part of the reservation request. A lease should not be granted for an experiment that the requester does not have permission to use. The earlier the request fails, the better.

Joins and Leaves

The join and leave operations initiated by the DieselNet handler should be able to remain as they are. The framework would continue to interface with the DOME portal rather than directly to the buses.