Virtualization Pipeline For Testing and Validating Lustre Environments

Johnathan Martinez | Mentor: Thomas Bertschinger, Jarrett Crews | HPC-INF

Introduction

WHAT IS LUSTRE?

Lustre is an open-source, massively parallel file system designed for high-performance computing environments. It excels at handling large-scale data by allowing clients to leverage bandwidth across multiple servers simultaneously. This architecture is crucial for I/O-intensive workloads like modeling and simulation, making Lustre an essential part of the HPC ecosystem.

A typical Lustre environment consists of three key components: Metadata Servers (MDS) that manage file attributes and directory structure, Object Storage Servers (OSS) that store the actual file data, and clients that access the file system.

TESTING CHALLENGES

While parallel file systems offer linear I/O scaling, they introduce complexity through inter-server communication requirements. Testing Lustre requires configuring multiple components (client, MDS, OSS) for each test environment. This pipeline simplifies this process by creating virtual machines (VMs) to simulate a Lustre environment, enabling safe validation before production deployment.

Build Pipeline

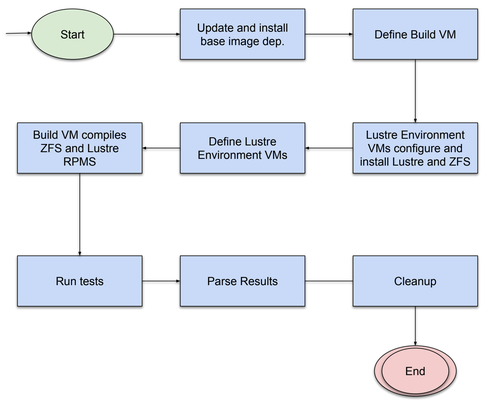

Figure 1: Illustrates the process of building and installing Lustre packages

Go Implementation

WHY GO?

Go serves as the foundation for the pipeline implementation, chosen for its robust concurrency model, strong type system, and comprehensive error handling capabilities. The language enables the creation of a reliable infrastructure orchestration layer while maintaining clean automation logic and error management.

LIBVIRT-GO

A low-level Go package is utilized where it provides direct bindings to the libvirt API. This allows the application to communicate directly with the libvirt service, enabling control of virtual machine lifecycle operations.

LIBVIRTXML

A high-level Go package offers structured representations of libvirt's XML configurations, simplifying the creation and manipulation of definitions for virtual machine domains, network configurations, and storage pools and volumes

CODING ARCHITECTURE

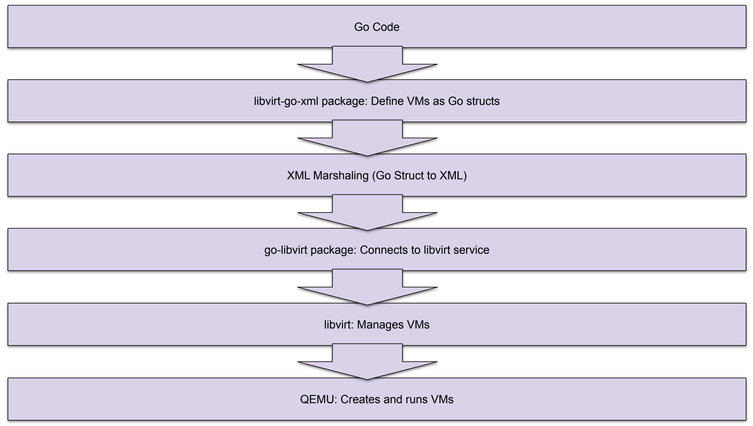

Figure 2: Shows the interaction between Go packages and hypervisor to make guests

PRIMARY CHALLENGE

ERROR RECOVERY

The primary challenge in this pipeline implementation is robust error recovery. When errors occur, the application must carefully unwind all previously completed steps in reverse order. This complexity arises because libvirt definitions persist independently of the program's execution state. Failed pipeline runs leave behind artifacts such as VM instances, storage volumes, and networks in the libvirt environment. Subsequent pipeline runs will encounter these past resources, causing cascading failures.

Virtual Infrastructure

STORAGE PROVISIONING

Libvirt enables the allocation of dedicated storage for VMs through storage pools. These pools are partitioned into volumes that are attached to VMs as block devices.

For the pipeline, a pre-configured base image with the required operating system serves as backing storage for any volume created. When a new VM is provisioned, a storage volume is created from this template and assigned to the guest machine

Each guest VM automatically mounts a shared file system that's configured through libvirt XML definitions upon creation of a guest. This shared storage architecture serves two critical functions: it enables the execution of bash scripts directly inside the VMs without manual file transfers, and it allows a centralized location to collection logging information related to the state of each guest.

NETWORKING

The application environment uses libvirt's integrated dnsmasq service to automatically assign IP addresses to VMs via DHCP. This ensures each virtual machine receives a unique network identity upon creation, enabling seamless communication between the pipeline controller and guests. In addition, it facilitates communication between Lustre components i.e., MDS OSS, and clients.

VIRTUAL MACHINE LAYOUT

VMs are created with specific roles that mirror a typical Lustre environment, plus an additional specialized build role. The environment consists of:

1. Metadata Server (MDS) VM
2. Object Storage Server (OSS) VM
3. Client VM
4. Build VM

The Build VM serves as a centralized compilation environment, generating Lustre and ZFS RPM packages on a shared filesystem. This shared storage architecture allows all other VMs to access and install these packages according to their specific roles, eliminating redundant compilation work and ensuring consistency across the environment.

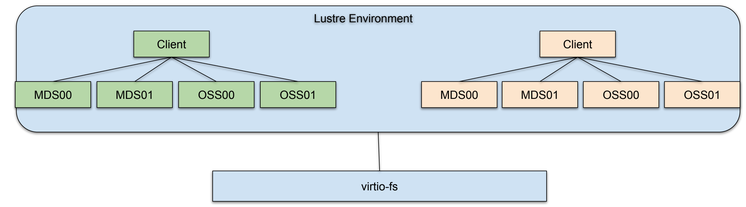

Figure 3: Depicts the Lustre environment being set up, along with how scripts are shared between servers

Los Alamos NATIONAL LABORATORY
NNSA National Nuclear Security Administration
Managed by Triad National Security, LLC, for the U.S. Department of Energy's NNSA.
ROSY ID: 3da09f09 LA-UR-25-28261
NEW MEXICO TECH SCIENCE • ENGINEERING • RESEARCH UNIVERSITY