About Novell ZENworks Orchestrator

Novell ZENworks Orchestrator (ZO) is a tool set that provides heterogeneous management of physical machines and many types of Virtual Machines. Novell ZENworks Orchestrator also adds the ability to orchestrate the environment by writing scripts that automate the provisioning of the machines and performing work.

Novell ZENworks Orchestrator is built as an enterprise server with capabilities to support extremely large computing environments. It includes all the services required to provide an enhanced grid computing application server environment. Below is a high-level graphic view of the Orchestrator Architecture:

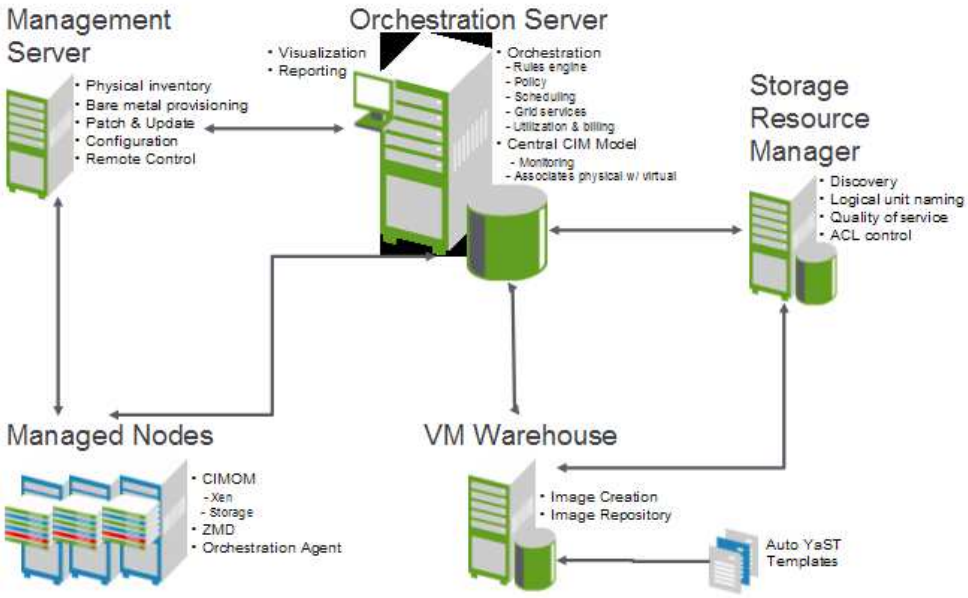

Figure 2: Novell ZENworks Architecture

ZENworks Orchestrator provides:

- Policy-based control system
- Constraint based job assignment & rules execution
- Reservation of resources
- Pre-emption of lower priority tasks for critical tasks
- Scheduled and event driven tasks
- Deploys and moves workloads dynamically
- Heuristic learning
- Pre-position workloads based on history
- Task to resource optimization
- Extensible Job Definition
- Python-based job definition language
- XML based Policy definition language

Novell ZENworks Orchestrator uses intelligent automation to manage heterogeneous virtual machines to align IT to business requirements, control costs, and minimize your risks, thus giving you more control and flexibility to align business objectives to IT value.

Novell ZENworks Orchestrator can be used for:

- Data Center Server Consolidation
- Workload / Performance Management
- Capacity On-demand
- Business Continuity / Disaster Recovery
- High Performance Computing (HPC)
- Development & Testing Environment Automation
- Dynamically reallocate and provision workloads

A simple view of the ZENworks Orchestrator software architecture is diagrammed below, showing the Orchestrator software components:

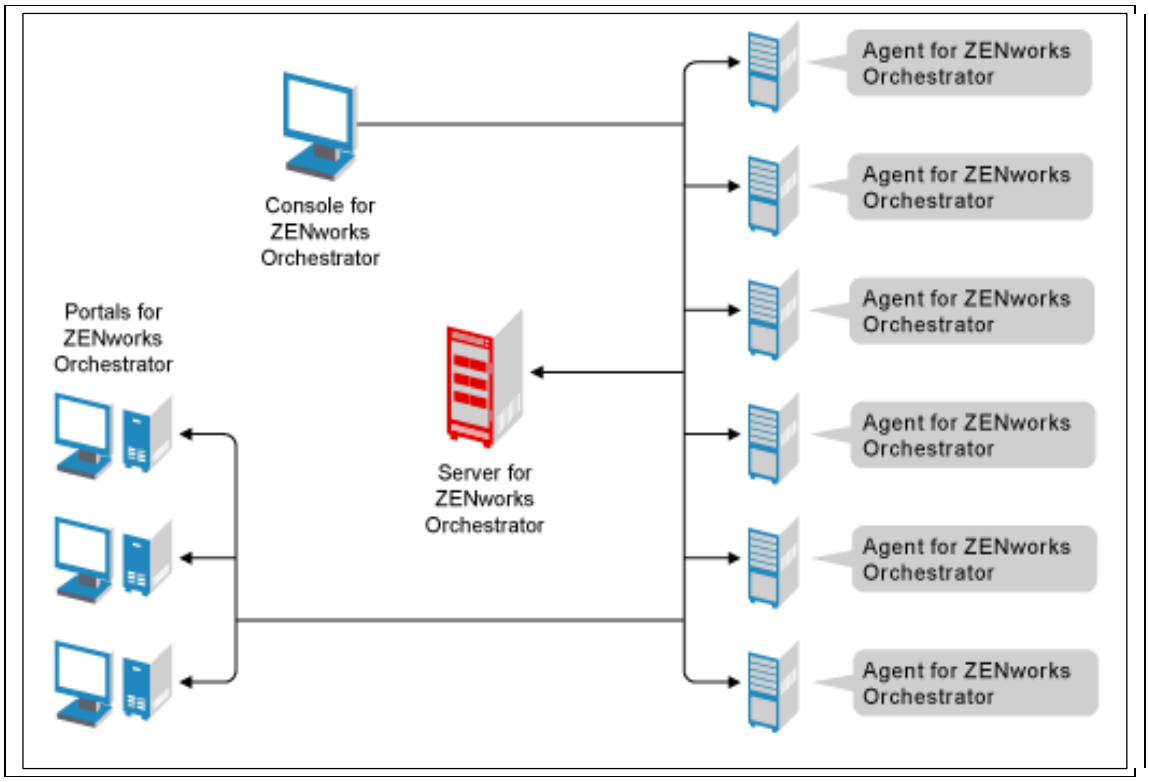

Figure 3: Novell ZENworks Orchestrator Software Architecture

The Novell ZENworks Orchestrator Server (ZOS Server) is the gateway between enterprise applications and resource servers. The server has two primary functions:

1. To manage the resource servers

2. To manage jobs submitted from applications to run on the computing resource

In the first function, the server manages the computing resources by collecting, maintaining, and updating their status availability, service cost, and other facts. Changes to the computing resources can be made either by the administrator or by the computing resource owner if there is one. The individual who makes changes (administrator or owner) is not important in this first function. What is important is that changes are discovered and the computing resource attributes are kept up-to-date.

The second function of the server is to run application requests—called jobs—on the computing resources. Because the server is capable of handling multiple jobs, it uses a policy-based broker and scheduler to decide when and how a job should run on the managed computing resources. The decisions are based on many controlled factors, including the number of computing resources, their cost, and a variety of other factors as requested by the application, but managed under policy constraints set up by the server administrator. The server runs the job and provides all the job's output responses back to the requesting application. The server handles and responds to the application with any error conditions, includes security, faults, outages, etc., that it might encounter. It provides failover capabilities to allow jobs to continue if computing resources and network conditions degrade.

The Novell ZENworks Orchestrator Agent (ZOS Agent) is installed on all computing resources that are to be managed. It runs applications under the management of the ZOS Server.

The Novell ZENworks Orchestrator Clients (ZOS Clients) let a computing resource administrator troubleshoot, initiate, change, or shut down server functions for Novell ZENworks Orchestrator and its computing resources. The clients also monitor all managed computing resource job activity and provide facilities to manage application jobs. When you install the ZOS Clients on a computing resource, you are installing the following tools:

- zos command line interface
- zosadmin command line interface
- Novell ZENworks Orchestrator Console (optional)
- Java* SDK (toolkit)

The Novell ZENworks Orchestrator Console is a graphical user interface running on Java. It provides a way for the Orchestrator administrator to troubleshoot and to initiate, change, or shut down the functioning of the ZOS Server and its resources. It also functions as a monitor of all Orchestrator job activity, and it provides an interface for managing Orchestrator jobs as well as:

- Administrative and Operational Control
  - Administers ad-hoc (manual), programmatic, and dynamic provisioning of virtual resources to fixed or shared resources within a data center

- Architectural and Engineering Development
  - Provides an integrated development environment (IDE) GUI for creating policy extensions
  - Enables 3rd party application integration for utility based SOA applications
  - Full command line interface support

The Novell ZENworks Orchestrator User Portal is available as a thin client. It provides end users with the ability to administer their jobs among the managed computing resources.

ZENworks Virtual Machine Management (VM Management) can be licensed as an add-on Management Pack to the basic ZENworks Orchestrator components and provides the ability to confidently employ virtualization in your data centers. From VMware to Microsoft to Xen, this policy-based solution automates the process of deploying and managing virtual data center assets, as well as dynamically provisioning workloads and ensuring business continuity.

The Virtual Machine (VM) Management part of the Orchestrator consists of provisioning adapters for different kinds of virtual machines, an image warehouse, and an image creation interface.

VM images can be created from a graphical interface, with manual configuration or using control files like for example AutoYaST for SUSE Linux Enterprise systems.

The VM images are then checked into the image warehouse, where they are put under version control and made available for provisioning

The provisioning adapters can be triggered through the graphical interface, Job scripts, or by requirement for services, and will make the VM image available directly from the image warehouse on a manually selected or eligible available VM host. For system-level backups, the image can be checked into the warehouse from the Orchestrator to create a new image version.

ZENworks Virtual Machine Management can be used for:

- Legacy Operations
- Temporary Application Operations
- Server Maintenance without Interrupting Service
- Checkpoint Virtual Machine State
- Roll Back To Best Known State
- Restore Service in the Event of a Fault on Virtualized Container
- Dynamic Machine Configuration and Reconfiguration
- Maintain Guaranteed Performance during Spike in Workload
- Licensing

A major feature of Novell ZENworks Orchestrator is the ability to manage the full Virtual Machine Lifecycle. The graphic below illustrates the VMM Lifecycle:

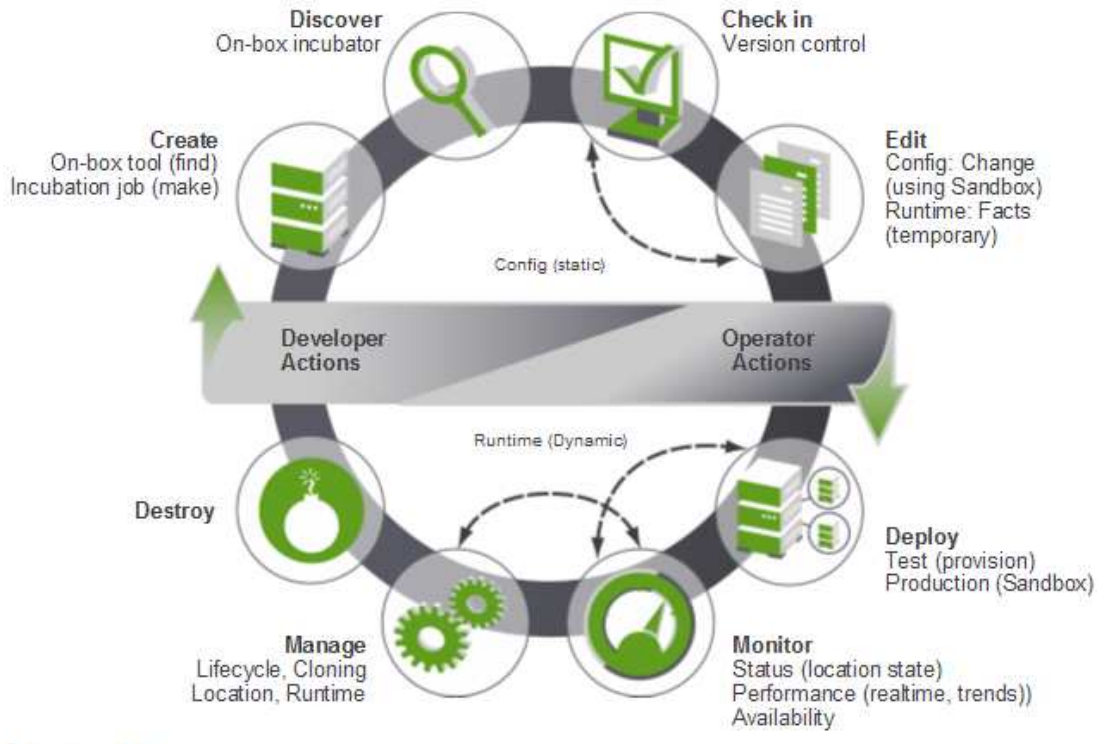

Figure 4: Virtual Machine Lifecycle

Virtual Machine Lifecycle offers the following:

- Powerful multi-vendor dynamic virtual machine automation services
- Resource Management
  - Discovery of existing virtual machine hosts, stores version and configuration details
  - Virtual machine images are discovered and brought under resource control
  - Discovery of virtual machine images including XEN and VMDK formats
  - Discovers off-line and on-line VM's & templates
- Inventory
  - Maintains a library of virtual machine images and hosts
  - Like physical resources, VM's have ‘facts’ describing attributes, and can be grouped
- Heterogeneous virtual machine management
  - Supports VMware, Xen, and Microsoft VM's
  - Deployment, re-deployment/rollback
  - Associates physical, virtual, and storage compute nodes
- Cluster-aware virtualization
  - H/A, DR, Warm-site replication
- Extensible framework
  - VMM provides abstracts native interfaces for performing actions on the hypervisor
  - Designed to quickly create VM providers by end user customers for other hypervisor's

High-Performance Computing (HPC) Management provides grid-based management of Java applications and enables workloads to be distributed for parallel execution. This includes automated high-performance multicast data distribution which can move and copy large volumes of data to remote resources for processing, thus enabling high performance computing (HPC) workloads to be distributed across your computing resources to gain maximum efficiency.

How ZO communicates with its components

The following diagram illustrates how the various components of Novell ZENworks Orchestrator communicate with the ZOS Server. An explanation for each communication link follows the diagram.

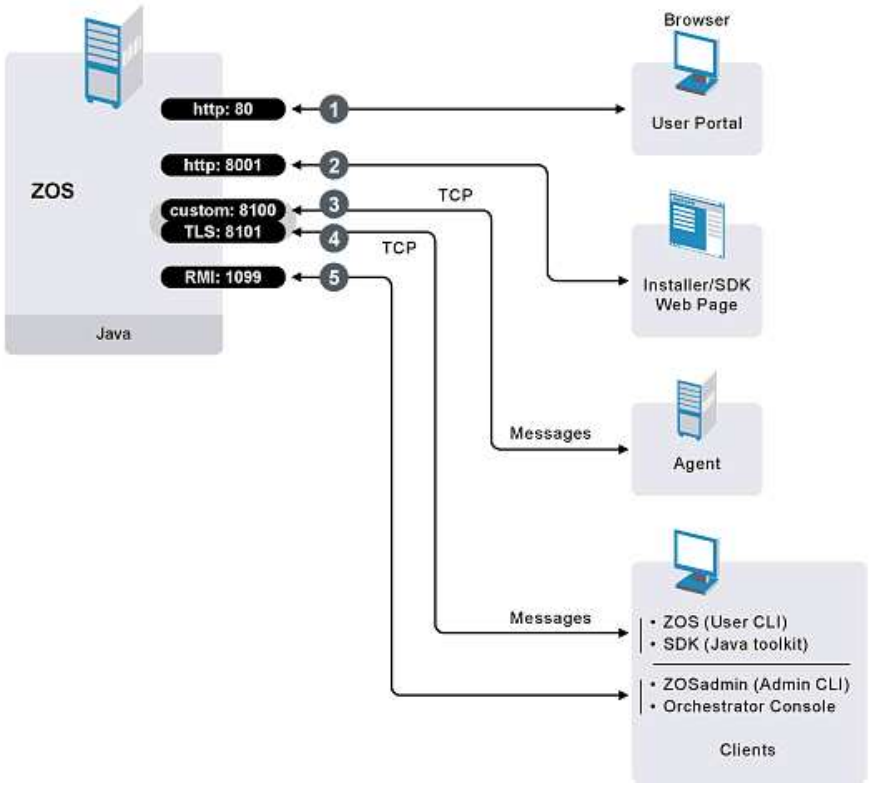

Figure 5: Communication with Novell ZENworks Orchestrator Server

Novell ZENworks Orchestrator Concepts and Terms

At a high level, the Novell ZENworks Orchestrator landscape appears as follows:

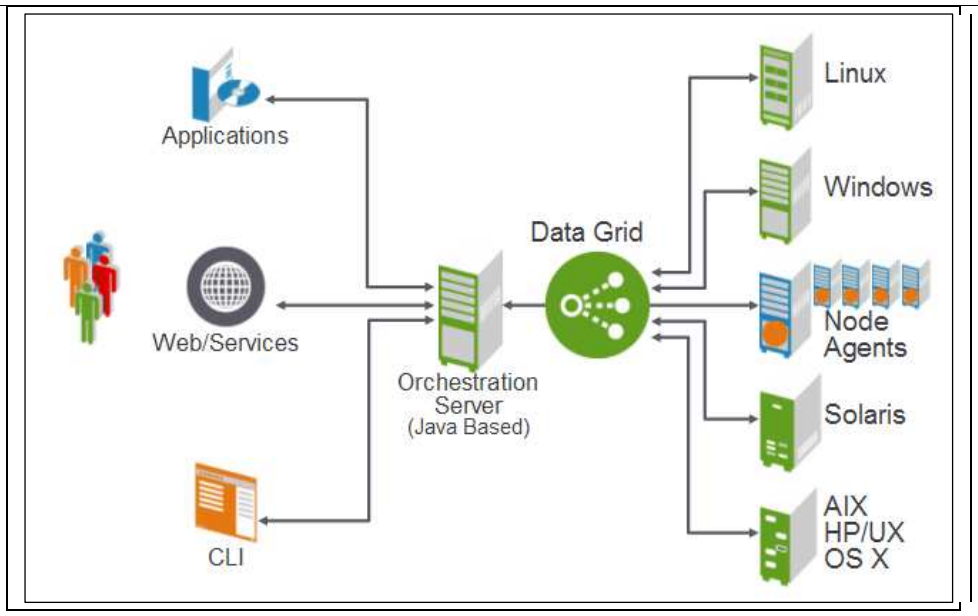

Figure 6: Novell ZENworks Orchestrator Landscape

The Grid

When the Novell ZENworks Orchestrator Server (ZOS Server) is installed, an entity called a grid is created, which is basically an administrative domain container. A grid contains all of the objects in network or data center that the Orchestrator is responsible to monitor and manage, including Users, Resources, and Jobs.

Typical Use of the Grid

In general, the everyday use of the grid and its computational components follows this sequence:

1. Jobs containing applications programmed by developers are created. A job component includes the application code, instructions for running the job, and policies to apply to the job.
2. The ZOS Server is started. It identifies all the available computing resources.
3. The administrator logs in to the server and deploys jobs to it so that users can run these jobs. The administrator also creates and distributes user logins.
4. Users log in to the server and search for deployed jobs they want to run.
5. When a user selects a job to run, he or she runs the job based on selected options; for example, he or she might select when to run the job, how many computing resources the job should run on, the type of computing resources to be used, and so on.
6. Users monitor and control their own jobs by using certain functions, such as cancelling a job, pausing a job, or even changing the priority of a job if they have rights to do so.
7. Administrators can monitor and control all of the running jobs. They can dynamically change how a job runs before or while it is executing, they can change job priorities, or they can turn computing resources off or on.

The graphic below illustrates the relationship of the grid types to the overall architecture (applications, etc.)

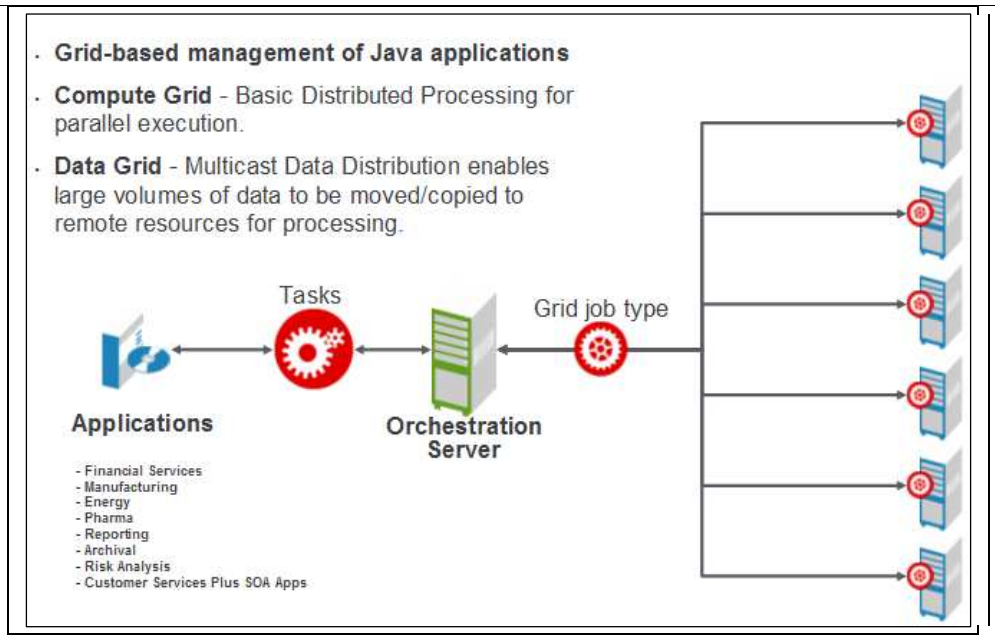

Figure 7: Grid Types

Users

For Novell ZENworks Orchestrator, a user is an individual who authenticates to the ZOS Server for the purpose of managing (that is, running, monitoring, cancelling, pausing, stopping, or starting) a deployed job. The Orchestrator administrator can use the Orchestrator Console to identify users who are running jobs and to monitor the jobs that are currently running or that have run during the current server session.

Resources

For Novell ZENworks Orchestrator, a resource refers to a computing node somewhere in your network (that is, your grid or data center network), which has a ZENworks Orchestrator Agent (ZOS Agent) installed on it. When the ZOS Agent is installed on a resource, communication between the agent and the ZENworks Orchestrator Server (ZOS Server) is established and the computing resource can be discovered and can begin performing jobs that are assigned to it by the ZOS Server.

Jobs

For Novell ZENworks Orchestrator, a job is an application module that can run on some or all of the resources. Job logic is written in Python, and can include instructions and policies that dictate how, when, and where that job runs. In addition, the job can embed instructions that dictate any processes or applications that the resource needs to launch.

The ZENworks Orchestrator is a system that allows making complex, distributed services independent from the systems where they are running. Orchestrator Jobs can manage distributed applications on a cluster of machines that may be virtual or physical, controlling and monitoring start, stop, restart and recovery through a distributed scripting engine with policy control and constraint-based resource scheduling.

Resources for specific services like virtual machines can be provisioned “manually” through the Orchestrator interfaces, “explicitly” through job scripts, or “implicitly” on demand through the policy-controlled provisioning engine in the Orchestrator.

The Job Description Language is Python, a modern, object-oriented scripting language that is augmented with a huge object library for the Orchestrator-specific functionality.

Job scripts can be written and managed outside the Orchestrator, then deployed automatically to one or more Orchestrator engines so that they are synchronized for quality assurance and disaster recovery functionality.

ZENworks Orchestrator Console

Both the grid administrator and the job developer need to have access to and use the ZENworks Orchestrator Console. The administrator needs to use the console to perform any management functions, such as creating user accounts and managing ZOS Server activities. The developer uses the console to access the JDL editor for creating or modifying jobs and policies.

The following figure shows the general areas on the console interface that are referred to in this guide.

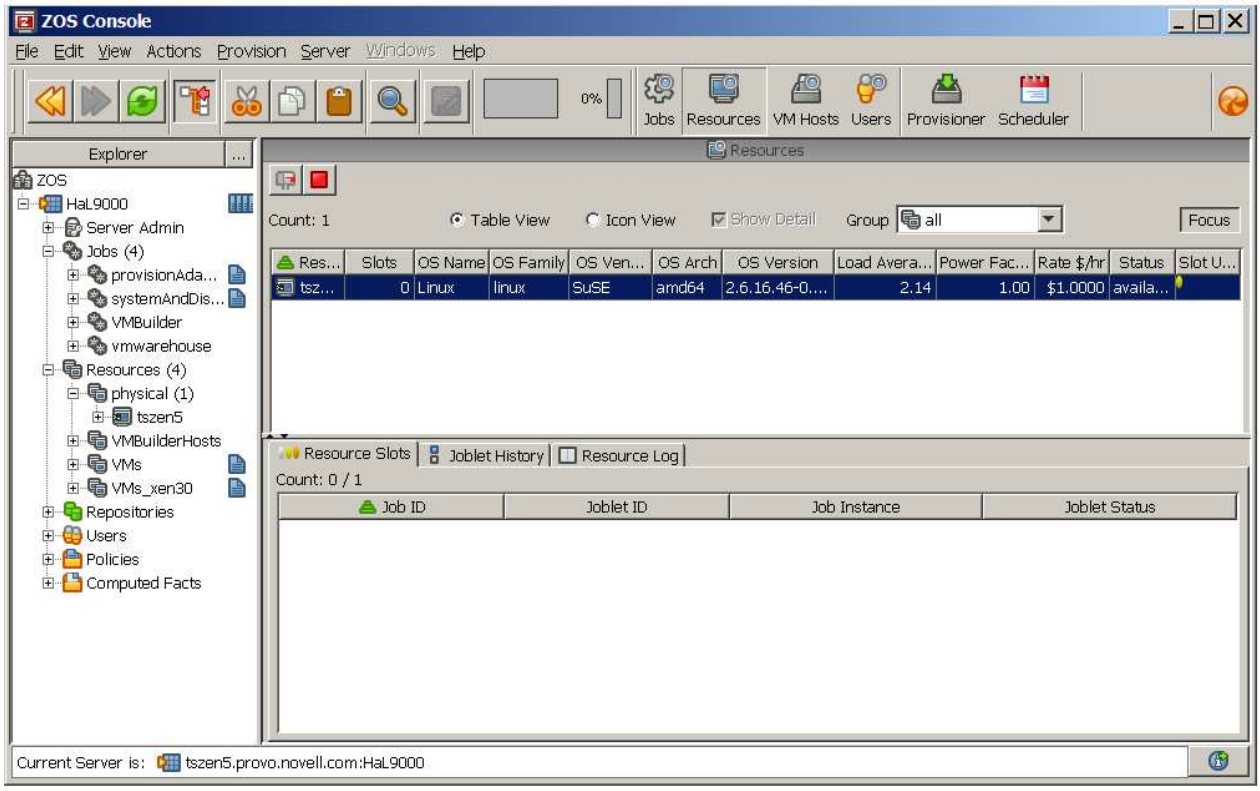

Figure 8: Novell ZENworks Orchestrator Console