Xen Management with SmartFrog
On-Demand Supply of Heterogeneous, Synchronized Execution Environments

Xavier Gréhant$^{1,2}$, Olivier Pernet$^3$, Sverre Jarp$^1$, Isabelle Demeure$^2$, and Peter Toft$^4$

$^1$ CERN openlab, Geneva, Switzerland
$^2$ École Nationale Supérieure des Télécommunications, Paris, France
$^3$ ENSIMAG, Grenoble, France
$^4$ Hewlett-Packard Laboratories, Bristol, UK

xag@cern.ch

Abstract. Applications to be executed on multipurpose Grids frequently have very specific resource requirements (platform, kernel, operating systems, libraries, memory, CPU, etc.) and need to be delegated part of the resource control. Typical Grid sites offer a limited range of resource types, inhibiting the range of applications that can be supported; and Grid node managers are bound to maintain their servers according to users requirements. To address these problems, we introduce SmartDomains, which combines the high performance virtual machine technology provided by Xen, with automatic deployment of Xen virtual machines using the SmartFrog configuration and deployment framework. SmartDomains automatically deploys distributed, synchronized pools of custom-configured Xen virtual machines and manages them through their lifecycle as a single coherent distributed execution environment. SmartDomains uses a representation of the complete distributed resources specifications, including information about how to sequence their creation and removal. We discuss SmartDomains test cases at CERN for distributed testbeds and Grid execution nodes.

1 Introduction

Although virtual machine (VM) technology has been around for four decades or more [12], there has been a resurgence of interest in recent years as virtualization has become practical on commodity hardware [2]. It follows that grid resource management tools will evolve to embrace support for virtual resources and those that do not will risk obsolescence. Our contribution to this evolution is a tool called SmartDomains (SD), which automatically supplies custom-configured, distributed virtual execution environments targeted at running batch Grid jobs or conducting system tests. In these contexts, it is important to keep independent the activity on the utilized resources and the maintenance of the backing hardware. SD provides fast, integrated VM control and a representation of resources to decouple administration and usage. A comparable resource control plane was
developed in COD [4] for nodes provisioning in Beowulf clusters, and in PlanetLab [5] for managing networked applications’ points of presence. Scientific production Grids (e.g. LCG, EGEE, OSG) do not yet optimise back-end resource usage with platform virtualization [6], which would add extra dimensions of flexibility in terms of resource configuration and fractional physical resource allocation (as illustrated with Tycoon [7]) although Globus Virtual Workspaces [8] already make interactions with the VM Monitors (VMMs) a Grid service.

The goal of SD is to provide a simple-to-use yet powerful mechanism for describing a required set of virtual machine resources, and a fully-automated deployment system to create VMs according to the supplied description. The automated deployment engine is a peer-to-peer distributed layer that takes in resource request descriptions and realises them with the requested sequencing. The same deployment engine is used to deactivate resource requests and release their resources, again according to the specified sequencing. Our experience with this approach leads us to believe that it makes it easier for resource administrators to prepare and control virtual resources, and for developers to create new functionality.

In this paper we discuss SD from three perspectives: In section 2 we explain the component technologies used in the SD system. SD usage is explained in section 3 along with a comparison of other systems using virtualization for resource management, and performance measurements are presented. In section 4 we illustrate the benefits of SD for resource administration, and its contributions to research and development in resource management systems.

2 SmartDomains, a Novel Approach

SD builds on Xen [9] for virtualization, and on SmartFrog [10,11] for the resource description and deployment mechanisms. Using virtualization in batch execution environments offers resource consumers and resource providers the following benefits:

- Software compatibility: By creating a library of customized VM images, VMs can easily replicate a very wide range of resource configurations, satisfying the specific needs of a wide range of applications.
- Resource sharing and performance isolation: By running multiple VMs on the same physical machine, fractional resources can be allocated, with fine-grained control over the resource consumption of each VM.
- Failure isolation: VM failures do not affect the physical node nor other VMs.

We chose the Xen virtualization technology [9] for its high performance, openness, advanced features (live VM migration, for example) and growing popularity. Xen allows VMs to run at near native speed, which is critical for high-performance computing applications. However, the SD approach could be applied to other virtualization technologies such as VMWare [12], or in-kernel virtualization approaches such as KVM [13]. The resource description and deployment mechanism in SD is built using SmartFrog (SF), a Java-based framework for the configuration, deployment and management of distributed software.