Hypervisor-Based Server Virtualization

Eduardo Correia
Christchurch Polytechnic Institute of Technology, New Zealand

INTRODUCTION

Virtualization has a long history that dates back to the nineteen sixties (Hoopes, 2011; Hinson & MacIsaac, 2011). It was then that the Cambridge Scientific Center developed the CP-40, a system based on fully virtualized hardware. This led to the VM/370, which was capable of running many virtual machines, all managed by something called a "virtual machine monitor" (Williams and Garcia, 2007; Hoopes, 2009). These earlier forms of virtualization, though, were implemented specifically for the mainframe market. Today this technology is most closely associated with the company that brought virtualization to x86 systems approximately thirty years after the CP-40 and popularized its commercial use: VMware. At first, virtualization was used largely in test and development environments but it soon became clear that the conventional installation of the operating system directly on hardware is wasteful, especially considering the capability of modern hardware and typically low server workloads. Other organizations have recognized the value of virtualization and since developed their own virtualization platforms, including Microsoft, Citrix and the Open Virtualization Alliance (OVA).

These platforms all involve abstracting the hardware that is presented through a hypervisor to guest systems otherwise known as "virtual machines," so enabling more than one system to run on a single physical computer. It is the hypervisor that abstracts the physical hardware and presents it to each virtual machine as well as manages access to the actual hardware. Once a system is stored as a collection of discrete files, it is independent of and detachable from the underlying hardware. This provides opportunities to make copies of entire systems, and given real impetus to notions of agility, availability and fault tolerance, and underlies the widespread promotion and adoption of cloud environments. In fact, this abstraction highlights the rigidity of traditional systems running directly on physical hardware compared to the elasticity of virtualized, software-defined resources.

This article discusses the nature of virtual machines before exploring the basic architecture of hypervisor-based server virtualization. It goes on to explain the role of management software commonly used to manage hosts, resources and virtual machines, and finally explores the future of virtualization in the context of a proliferation of devices and their ecosystem.

BACKGROUND

The hypervisor sets up isolated execution environments, each of which has its own set of abstracted hardware resources such as memory, disk, processor and network adapter(s). It is this bundling of abstracted hardware that is really a virtual machine capable of sharing physical hardware with other virtual machines because it believes it has exclusive access to the hardware. The hypervisor manages the way virtual machines access hardware (Hales, Eiler, & Jones, 2013), and provides the virtual machines with the illusion that they are in fact running on physical hardware, which they are not (Arrasjid, Epping, & Kaplan, 2010). More recently changes are being made to the operating systems themselves to make them aware that they are running on virtual rather than physical hardware, so as to improve performance of virtual machines and remove some of the overhead associated with virtualization.

As Microsoft (2012a) put it, the hypervisor is really the layer of software that interacts with the hardware and lies below one or more operating systems. These guest operating systems, or virtual machines as they are more commonly known, are in every way the same as machines installed on physical computers except that they run on the virtualized hardware presented to them by the hypervisor. As a result of these isolated

DOI: 10.4018/978-1-4666-5888-2.ch112

execution environments one virtual machine can fail without it affecting other running virtual machines. Naturally dependencies may exist, whereby a web server might require a database server and if the database server fails, so will the delivery of data to the web server. This, though, has nothing to virtualization and would occur whether machines are virtualized or not (Williams & García, 2007).

A virtual machine is emulated hardware stored as a set of discrete files. Typically it contains a configuration file that specifies the abstracted hardware of the virtual machine and its basic configuration. The files store the contents of the virtual machine. One file stores the contents of memory, another the bios, and yet other files contain system-related data from snapshots to the configuration and contents of one or more virtual disks (VMware, 2011). As Figure 1 shows, the virtual machine called UX01 comprises of a number of files. The configuration of the virtual machine is stored in UX01.vmx but there is place for additional configuration in UX01.vmxf. UX01.vmdk, contains the configuration of the virtual disk of the virtual machine, whereas the hidden UX01-flat.vmdk is where the data on the virtual disk is actually stored. The BIOS of the virtual machine is stored in UX01.nvram and the state of the suspended UX01 is stored in UX01-382c065e.vmss. The UX01.vmsd file stores the content of virtual machine snapshots, whereas the vmware.log file stores logged events related to the virtual machine. These system archives old vmware.log files, seen here in the vmware-1x.log files. In the case of Hyper-V the configuration details of a virtual machine is contained in an.xml file, the contents of a virtual hard disk in a.vhd or.vhdx file, and so on (Armstrong, 2010). Although the names of these files are different to those found in ESXi, and the Microsoft hypervisor interacts with them somewhat differently, the same principle applies: a virtual machine ultimately takes the form of a set of discrete files, making it far more flexible and easier to manage than traditional systems that are not virtualized.

VIRTUALIZATION ARCHITECTURE

Two types of hypervisors exist: type 1 hypervisors (also known as bare-metal hypervisors) and type 2 hypervisors (also known as hosted hypervisors) (Goldberg, 1973). VMware Workstation, Microsoft Virtual PC and other hosted hypervisors require a conventional host operating system. They are installed as an application and can only access the hardware through either a Windows or Linux host operating system, but these workstations can be used to run other conventional applications. Bare-metal hypervisors, on the other hand, are associated with backend server infrastructure, although recently Microsoft have released Hyper-V as part of its client operating system, Windows 8, and managed to bring some benefits of server virtualization to the desktop. VMware ESX and later ESXi Server, Microsoft Hyper-V, Citrix Xen Server and other bare-metal hypervisors run directly on the physical hardware instead of making calls by way of a conventional host operating system. As a result, a bare-metal hypervisor

Figure 1. Files in an ESXi virtual machine