Chapter 16

Internet-Based Virtual Computing Infrastructure for Cloud Computing

James Hardy
University of Derby, UK

Cui Lei
Beihang University, China

Lu Liu
University of Derby, UK

Jianxin Li
Beihang University, China

ABSTRACT

Virtualisation is massively important in computing and continues to develop. This chapter discusses and evaluates the virtualisation technologies and in particular, a state-of-art system called iVIC (the Internet-based Virtual Computing) developed by Beihang University, China as it provides an all-in-one example of many of the major headline Cloud Computing titles of SaaS, IaaS, and HaaS. The chapter considers several virtualization packages which are either commercial, community, or experimental, before focusing on iVIC, a virtual machine cloning system that may be beneficial in a learning or office environment. The chapter introduces a test environment which is used to assess the performance of the iVIC process and the virtual machines created. Power requirements of virtual, as opposed to physical machines, are compared and evaluated. The chapter closes with conclusions regarding virtualisation and iVIC.

1. INTRODUCTION

Virtualisation has been used for many years; however its importance has increased significantly in recent times. The underlying reasons why virtualisation is undertaken have also diversified. It is commonplace for cost reduction, increased power efficiency and operational flexibility to be cited as the prime drivers but, whilst these points are clearly very important, testing, repeatability and massive parallelism should not be overlooked as potential motivators.

In this chapter, a state-of-art virtualisation system - iVIC (the Internet-based Virtual Comput-
Internet-Based Virtual Computing Infrastructure for Cloud Computing

Two or more iVIC hosts can be linked over a network and will provide some degree of VM load balancing; alternatively a single iVIC host can be used as a complete “Cloud in a box”. The system creates new fully networked “user level” virtual machines as a cluster, as opposed to single entities, and is “on request” rather than “on demand”. Each virtual machine in a powered on cluster is individually accessible over VNC, can be powered on or off and can be executing any compatible software at the request of the user. The entire cluster can be granted or denied access to the physical network which does not affect VNC to the virtual machines or the networking between machines located on separate hosts.

The main objective of this chapter is to discuss virtualisation and a specific virtualisation cloning platform known as iVIC. Particular emphasis is placed on underpinning technologies and system functions.

A comparison of several virtualisation packages is included and intended to show that there is no single method to create a virtual machine, no single business ethic or financial motivator and no guarantee of continued growth and success without significant exposure and general acceptance.

The next section of this chapter considers the background and objectives of virtualisation and also introduces some of the terminology used. Several virtualisation packages are discussed in general terms in following section, including the objective and target use of the packages. Two sections focus on the iVIC system, presenting by example the guide to the experimental setup and the results obtained in testing. Conclusions and future considerations are given in the final sections. There are two appendices containing a sample xml file and potential modifications to guest machines to overcome a networking issue.

2. VIRTUALISATION

2.1 Virtualisation History

Virtualisation has been in use for several decades. Some of the earliest references that have been obtained are dated from 1964 (Varian, 1997) and formed part of the operating system of IBM CP-40 for the S/360 mainframe.

Many of the reasons and principles of the early IBM “pseudo machines” such as separation of user environments and OS backwards compatibility are still of particular importance in today. As an example, Microsoft provides backward compatibility by including “XP Mode” with Windows 7 to “reduce possible operational downtime by extending the life of existing software” (Microsoft, 2011). XP Mode is a virtual machine “appliance” with an XP operating system.

The concept of operating system backwards compatibility can be expanded beyond purely software and into the realms of hardware. Hardware emulation allows software intended for a specific hardware platform to function on completely different hardware. Early examples are “games console emulators”, where computers such as the Z80 based Sinclair ZX range from the very early 1980s were emulated on 6502 based Acorn and consequently Mac, DOS and several Windows variants (Scherrer, 2011). Although there is an apparently eternal interest in “retrospective gaming”, the original motivators for these emulators were purely commercial from user community: to extend the life of game software purchased for a system that very quickly became obsolete.

Emulation can also be used to reduce development time and cross-system expertise. Many programs intended for Windows can be executed directly on Linux systems using “compatibility layer” technologies such as Wine (WineHQ, 2011). If wine is available, any “.exe” programs started in a Linux environment first invoke Wine.