

# OpenNEbula

## 安裝與測試技術文件

郭文傑 [rock@nchc.org.tw](mailto:rock@nchc.org.tw)

國家高速網路與計算中心

格網技術組



National Applied Research Laboratories

National Center for High-Performance Computing

## 版本

日期	版本	內容修改
2009/04/21	初版	初稿
2009/04/24	二版	修改格式(左右對齊)和內容(root SSH key exchange)

## 目錄

第一章 前言

第二章 相關技術探討

2.1 Xen

2.2 OpenNEbula

第三章 OpenNEbula 安裝和測試

3.1 安裝

3.2 測試

第四章 結論

參考文獻



National Applied Research Laboratories

**National Center for High-Performance Computing**

# 第一章 前言

雲端運算是門值得研究的領域，著名的私人企業（搜尋引擎公司 Google 和全球最大的網路書城 Amazon）將本身的核心技術透過雲端來提供服務給一般使用者使用，例如：Google 在其本身的 Google App Engine 上架構了 Google Docs, Google Gmail, Google Reader 等就成了許多使用者每天都需使用到的服務，而 Amazon 則提供比 Google 更底層的服務，它將虛擬主機和儲存裝置作為服務提供給使用者來使用，讓使用者透過瀏覽器能隨時隨地存取到雲端的服務。

在如此的背景下，本組擬研發一套似 Amazon EC2 的系統架構來提供基礎平台的雲端服務，為了達成此項目標，從底層的技術逐漸建構起，包含了(1)採用 Lustre 作為雲端儲存機制，(2)使用 Xen 來虛擬化實體資源，(3)佈署成 Xen Cluster，(4)使用 OpenNEbula 提供虛擬技術平台的管理介面，而本技術文件便是著重在 OpenNEbula 的測試說明。

本技術文件的內容的第二章著重於相關的技術探討，第三章則包含了 OpenNEbula 的安裝和測試部份，第四章則為結論。



National Applied Research Laboratories

**National Center for High-Performance Computing**

## 第二章 相關技術探討

### 2.1 Xen

Xen 為一開放原始碼的虛擬化技術，Xen 支援多種 CPU 架構 (x86、x86\_64、IA64、PowerPC 等)，採用的虛擬化技術稱為『Para-Virtualization』，中文稱為準虛擬化或半虛擬化，它的主要概念在於改善早期使用的 Full-Virtulaziation (Vmware 的產品和 Sun 的 VirtualBox 都是屬於完全虛擬化) 的效能。

Full-Virtualization 的優點在於對使用者很簡單和直覺，但缺點就是效能損失約 20~30%，而 Xen 採用的 Para-Virtualization 可讓效能減少損失到 2~8%，但缺點就是虛擬機器的核心必須被修改。但 Xen 也可以透過 Intel VT-x 和 AMD-V(SVM) 的處理器來做到 Full-Virtulaziation，適用於當虛擬機器的核心無法修改時 (像如果需要安裝 Window 時)。

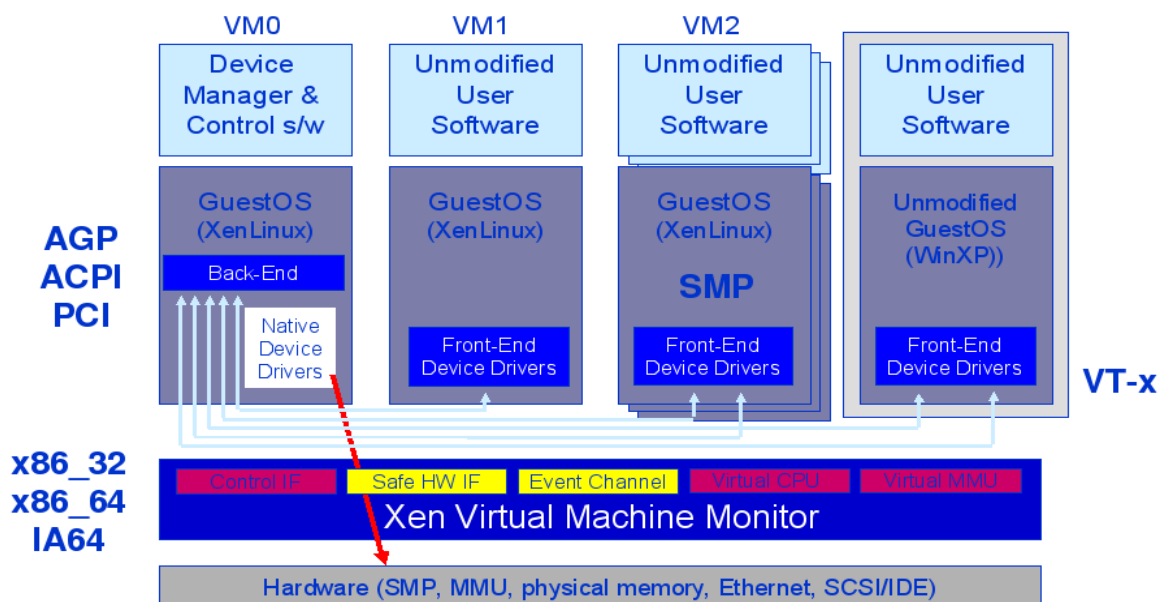


圖 2.1 Xen 運作架構圖

(取自 Xen 官網)



National Applied Research Laboratories

National Center for High-Performance Computing

圖 2.1 為 Xen 的運作架構圖，Xen Virtual Machine Monitor(也可稱為 Hypervisor)負責管理硬體，為了將實體硬體資源虛擬化，它修改了記憶體管理和 CPU 和 I/O 的介面，而本機稱為 domain 0 (簡稱 dom0)和虛擬機器 (domain U、domU)則運作再 Hypervisor 之上，dom0 和 domU 若是需要操作硬體資源時，皆需透過 Hypervisor 來控制硬體。

## 2.2 OpenNEbula

OpenNEbula(ONE) 是開放原始碼的虛擬基礎設備引擎用來動態佈署虛擬機器在一群實體資源上，OpenNEbula 最大的特色在於將虛擬平台從單一實體機器到一群實體資源。目前最新的版本為 1.2 版，可支援 Xen、KVM 也可透過 ONE 來控制 Amazon EC2 上的機器。

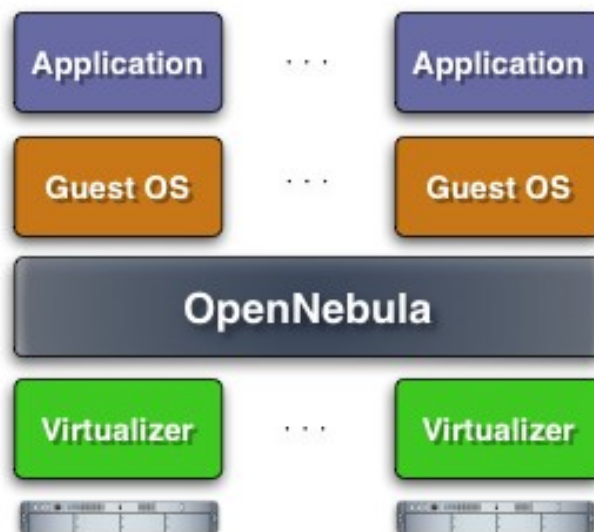


圖 2.2 OpenNEbula 架構圖

(取自 OpenNEbula 官網)

圖 2.2 是 OpenNEbula 的架構，它的概念是在 Virtualizer(Xen 或 KVM ) 上實現一個虛擬層，再此層實作管理的機制，包含了管理每臺實體機器的 Virtualizer 和實體機器上的虛擬機器。當 one start 時會產生 2 個 Daemon : (1)oned (2)mm\_sched 和 3 個 Driver : (1)one\_im\_ssh.rb (2)one\_vmm\_xen.rb (3)one\_tm.nfs.rb，ONE 預設使用的 Scheduler 是採用 Rank Scheduling Policy，若使用者的需要更複雜的排程，可以使用第 3 方提供的 Haizea 來實現。



National Applied Research Laboratories

**National Center for High-Performance Computing**

# 第三章 OpenNEbula 安裝和測試

## 3.1 安裝

安裝 OpenNEbula 前，必需注意(1)已經有佈署成 Xen Cluster 環境了，若是目前只有一臺單機安裝好，可以使用 DRBL 和 Clonezilla 來快速大量佈署 Xen Cluster 環境，(2)使用的共享儲存裝置(本組的雲端平台使用 Lustre 來作為共享的儲存裝置，但再本技術文件中採用簡單的 NFS)和網路架構為何，(3)所有的 node 皆已經設定 root 的 SSH Key exchange。

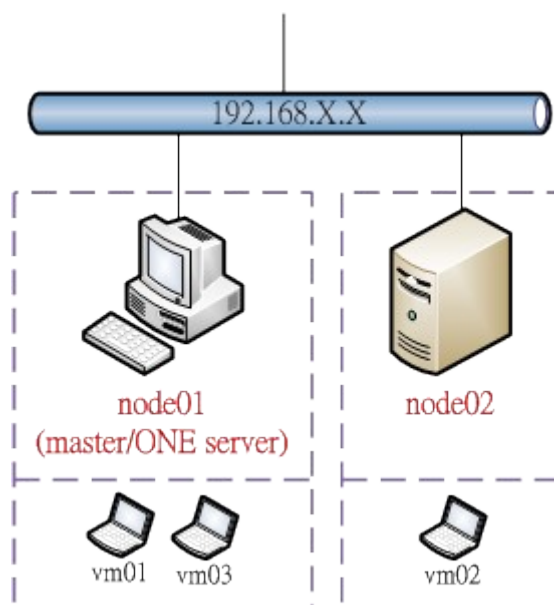


圖 3.1 本範例系統架構圖

圖 3.1 為本測試的系統架構圖，node01 作為 OpenNEbula 伺服器，再本範例中 node01 除了當 ONE 伺服器外，也安裝 Xen 來提供虛擬機器的服務，node02 則為 Xen 的機器，透過 ONE Server 可以動態的再 node01 和 node02 開啓虛擬機器。



National Applied Research Laboratories

National Center for High-Performance Computing

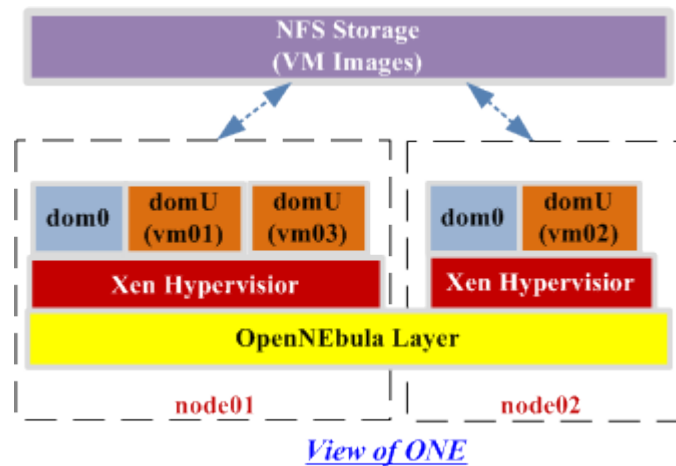


圖 3.2 本範例 ONE 架構圖

圖 3.2 為本範例系統的 ONE 架構圖，node01 和 node02 透過 OpenNEbula Layer 來管理虛擬機器，為了實現 Migration 和 Image 的統一存取區，本範例使用 NFS 作為共享的儲存裝置。

### ● 3.1.1 安裝 OpenNEbula 所需的相關套件

```
$ sudo aptitude install ruby sqlite3 libsqlite3-0 libsqlite3-dev libsqlite3-ruby libxmlrpc-c3
libxmlrpc-c3-dev scons g++ flex bison
```

### ● 3.1.2 下載 OpenNEbula 和安裝

```
#### 在 Master 上執行 (ONE 只需安裝在 Master 上) ####
$ cd
$ wget http://opennebula.org/files/one-1.2.0.tar.gz
$ tar zxvf one-1.2.0.tar.gz
$ cd one-1.2.0
$ scons
$ sudo mkdir /home/one

#### 配置 ONE 環境變數 ####
$ vim ~/.profile
export ONE_LOCATION=/home/one
export ONE_XMLRPC=http://localhost:2633/RPC2
export PATH=$PATH:$ONE_LOCATION/bin
$ source ~/.profile

$ sudo ./install.sh -d /home/one
```



National Applied Research Laboratories

National Center for High-Performance Computing

### ● 3.1.3 編輯 OpenNEbula 設定檔

```
#### 在 node01 上編輯，將 160~164 行註解掉，打開 169~173 行 ####
$ cd /home/one
$ vim etc/oned.conf
23 VM_POLLING_INTERVAL = 1
160 #TM_MAD = [
161 #name = "tm_ssh",
162 #executable = "one_tm",
163 #arguments = "tm_ssh/tm_ssh.conf",
164 #default = "tm_ssh/tm_ssh.conf" ]

169 TM_MAD = [
170 name = "tm_nfs",
171 executable = "one_tm",
172 arguments = "tm_ssh/tm_nfs.conf",
173 default = "tm_ssh/tm_nfs.conf" ]
```

### ● 3.1.4 啓動 one (OpenNEbula daemon)

- 啓動前需注意所有 node 都已 xend start
- 所以有 node 都可讓 master 的 root 免密碼登入

```
#### 在 master 上執行 ####
$ one start
```

### ● 3.1.5 將所有實體 node 增加到 OpenNEbula Pool

```
#### 在 master 上執行 ####
$ onehost add node01 im_xen vmm_xen tm_nfs
$ onehost add node02 im_xen vmm_xen tm_nfs
$ onehost add node03 im_xen vmm_xen tm_nfs
.....

#### 列出所有已增加的實體機器 ####
$ sudo onehost list
```

### ● 3.1.6 使用 OpenNEbula 開啓一個 VM

```
#### 在 master 上執行 ####
#### 編輯虛擬機器 xen01 的設定檔 ####
$ vim /home/domains/xen01.one
NAME = xen01
CPU = 1
MEMORY = 156
```



National Applied Research Laboratories

**National Center for High-Performance Computing**



```
OS = [ kernel    = /boot/vmlinuz-2.6.22.9-rock,
        initrd   = /boot/initrd.img-2.6.22.9-rock,
        root     = sda2,
        kernel_cmd = "ro xencons=tty console=tty1" ]
DISK = [ source  = /home/domains/xen01/disk.img,
        clone    = no,
        target   = sda2,
        readonly = no ]
DISK = [ type     = swap,
        size     = 512,
        target   = "sda1",
        readonly = "no" ]
NIC = [ mac="" ]
```

```
#### 使用 ONE 來開啓 xen01 ####
$ sudo onevm create xen01.one
$ sudo onevm deploy xen01 node02

#### 列出虛擬機器的狀態 ####
$ sudo onevm list
```

## 3.2 測試

此章節主要測試再 OpenNEbula 上使用 Migration 和 Live Migration，所謂的 Live Migration 主要能讓一台正在執行任務的虛擬機器，不需關機，變可將此臺虛擬機器移轉到另一臺實體機器上繼續運行，不會讓正在執行的任務停掉，導致需要重新再執行一次任務。此項特色可用於，當虛擬機器正跑在一臺實體機器上，但此台實體機器剛好要進行維護或是伸級時，但虛擬機器正在提供服務給他人或是正在執行重要的任務，無法隨便將虛擬幾器中斷掉，此時可以將此臺虛擬機器 Live Migrate 到另一臺實體機器上繼續運行。

### ● 3.2.1 測試 Migration (offline Migration)

```
#### 在 master 上執行 ####
$ cd /home/domains
$ onevm create xen01.one
$ onevm deploy xen01 node01
$ onevm list

#### 將 xen01 Migrate 到 node02 (設設 xen01 已經跑在 node01) ####
$ onevm migrate xen01 node02
```



National Applied Research Laboratories

National Center for High-Performance Computing

```
$ onevm list

#### 在 node02 登入 xen01，當 xen01 編譯程式時，將其 migrate 到 node01，看編譯過程式
是否中斷 ####
$ xm console one-X
$ aptitude install build-essential
$ wget http://www.kernel.org/pub/linux/kernel/v2.6/linux-2.6.29.1.tar.bz2
$ tar jxvf linux-2.6.29.1.tar.bz2
$ cd linux-2.6.29.1
$ make allmodconfig
$ make
CTRL+]

#### 在 master (ONE Server) 上，執行 Migrate 動作 ####
$ onevm migrate xen01 node01

#### 登入到 xen01 檢查 make 程序是否中斷 ####
$ xm console one-X
$ top
```

### ● 3.2.2 測試 Live Migration

```
#### 檢查 xen01 目前跑在哪一台實體機器上(假定 xen01 跑在 node01 時) ####
$ onevm list

#### 在 node01 登入 xen01，並讓 xen01 執行編譯核心的任務 ####
$ xm console one-X
$ cd linux-2.6.29.1
$ make allmodconfig
$ make clean
$ make

#### 將正在編譯中的 xen01 live migrate 到 node02 ####
$ onevm livemigrate xen01 node02
$ onevm list

#### 在 node01 登入到 xen01，並檢查 make 程序是否中斷 ####
$ xm console one-X
$ top
```



National Applied Research Laboratories

**National Center for High-Performance Computing**

## 第四章 結論

現階段本組的虛擬技術實驗平台已完成了(1)採用 Lustre 作為雲端儲存機制，(2)使用 Xen 來虛擬化實體資源，(3)佈署成 Xen Cluster，(4)使用 OpenNEbula 提供虛擬技術平台的管理介面，在 OpenNEbula 的測試上，目前已掌握了基礎功能的運作，但由於 OpenNEbula 的開發經驗尚不是很悠久，有些功能還未實現，像是提供一致的 domU 登入介面，onevm list 和 xen list 出現的虛擬機器名稱不一樣 OpenNEbula 的 domU 設定檔必須自行手動 v 修改，這些部份可能得等 OpenNEbula 之後的發展。

未來考量到虛擬技術平台的負載平衡，OpenNEbula 預設的排程是使用 Rank Scheduling Policy，而比較複雜的排程則需透過 Haizea 來實作，未來重點會放在虛擬機器的排程，和如何透過網頁的方式，讓使用者可以方便的透過網頁來控制虛擬機器。



## 參考文獻

- [Xen](#)
- [OpenNEbula](#)
- [DRBL](#)
- [Lustre](#)
- [格網技術組-Grid Architecture](#)
- [格網技術組-Cloud Computing](#)



National Applied Research Laboratories

**National Center for High-Performance Computing**