課程回顧（二）

國立臺灣師範大學物理學系 陳俊明

chunming@ntnu.edu.tw

平行檔案系統

- 以分散式讀寫資料的方式，避免所有的磁碟讀寫集中在單一硬碟或是單一伺服器上，通常由數個儲存節點所組成。

Lustre

GlusterFS

BeeGFS

OrangeFS

ceph

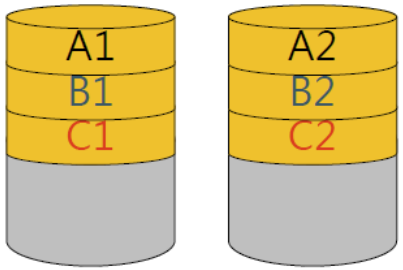
- Open Source
  - Lustre
  - Glusterfs
  - BeeGFS
  - OrangeFS
  - Ceph

IBM Spectrum Scale

- Enterprise
  - IBM GPFS
  - DELL Isilon OneFS

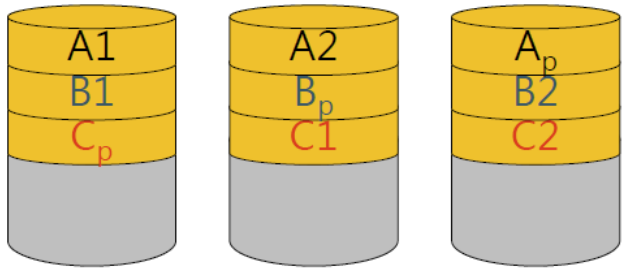
磁碟陣列-RAID(基本型)

RAID0



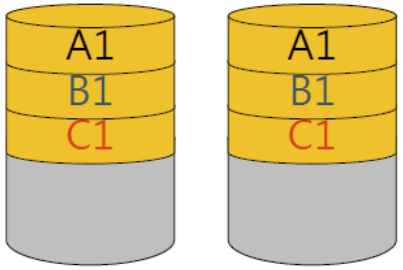
$Size = \mathrm{N} \times \min(S_1, S_2 \ldots S_N)$

RAID5



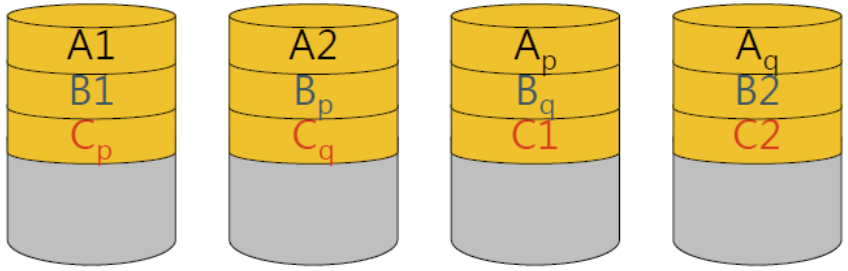
$Size = (N-1) \times \min(S_1, S_2, S_3, \ldots S_N)$

RAID1



$Size = \min(S_1, S_2 \ldots S_N)$

RAID6



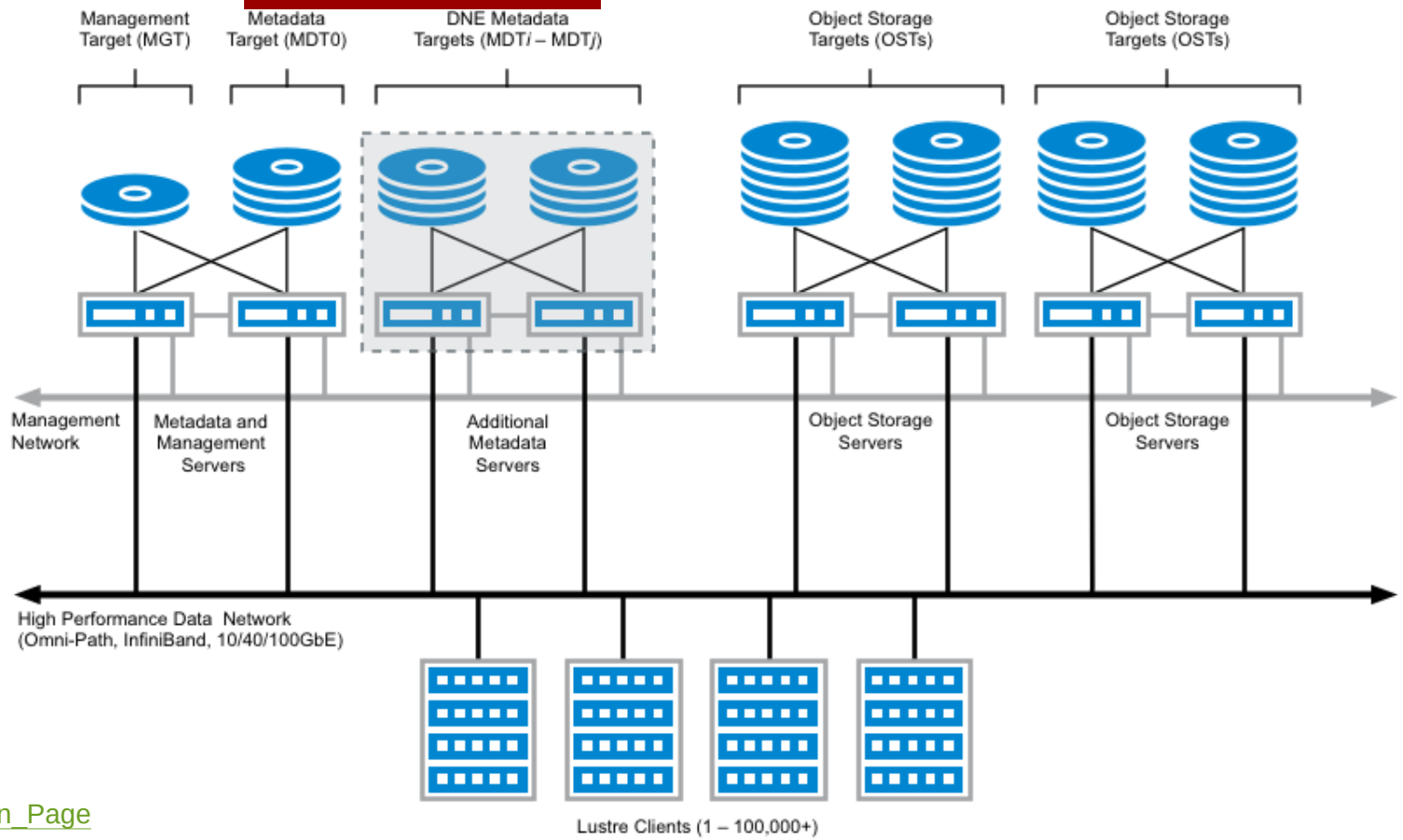
$Size = (N-2) \times \min(S_1, S_2, S_3, S_4 \ldots S_N)$

Lustre 架構圖

MDT : where is which files ?

OST-1, OST-2, OST-n : data files

MGT : System Management Info about OSTs and clients



圖片來源：http://wiki.lustre.org/Main_Page

Lustre基本架構

- 管理伺服器 MGS（Management Server)
- Lustre文件系统架構
  - 描述資料伺服器 MDS (Metadata Servers)
  - 物件儲存伺服器 OSS（Object Storage Servers)
  - Lustre 用戶端
- Lustre 網路通訊 LNet（Lustre Networking)

Lustre 建制策略

- MGT, MDT需做RAID1以擁有最安全的資料保護機制
- MGT, MDT可採用SSD或NVMe以降低存取瓶頸
- OST需做RAID6+Hot Spare以保護資料
- Client最好使用低延遲的高速網路

Lustre 維護

- 更新 Kernel 並重新編譯安裝 Lustre Client
- 日常檢查
  - 檢查 MDS 及 OSS 的硬體狀態
  - 檢查 MDS 及 OSS 上的負載、記憶體及系統容量 (lfs df -h, lfs df -hi)
  - 檢查 MDS 及 OSS 的網路狀態
  - OSS 負載過高時，登入 OSS 找尋負載來源 (lctl get_param ost.OSS.ost_io.req_history)
- 強制重開機時（負載過高導致無法登入MDS、OSS或當機）
  - 需在 MDS 或 OSS 上執行 e2fsck -f /dev/sdX 檢查 MDT 或 OST
  - 掛載 MDT 或 OST（lctl dl, 使用 dmesg 確認 kernel 訊息）
  - 等待 Client 端回復連線

PBS – 組成元件

- commands
  以命令列方式，讓透過socket來讓使用者進行提交(submit)、監督(monitor)、修改 (modify)和刪除(delete) 工作
- pbs_server
  接收、產生、管理及保護使用者的批次工作
- pbs_mom
  接收pbs_server 給予的批次工作，並呼叫對應的程式來執行，完成後將結果回報給pbs_server
- pbs_sched
  負責排程工作、資源分配及節點管理

設定 Queues

指令	說明
qmgr -c "del queue batch"	刪除Queues
qmgr -c "create queue workq queue_type=execution“ qmgr -c "set queue workq enabled = True“ qmgr -c "set queue workq started = True"	建立Queues
qmgr -c "set queue workq resources_default.walltime=1:00:00"	運算最長時間
qmgr -c "set queue workq max_running=10"	設定Queues最多同時作業數量
qmgr -c "set queue workq max_user_queuable=20"	設定使用者最多提交作業數量
qmgr -c "set queue workq max_user_run=5"	設定使用者最多同時作業數量
qmgr -c "set queue workq resources_default.nodes=1"	設定Queues預設的nodes數量
qmgr -c "set queue workq resources_max.nodes=4"	設定Queues最大使用的nodes數量

設定 Queues

指令	說明
qmgr -c "set queue workq acl_hosts=cn1+cn2+…“ qmgr -c "set queue workq acl_host_enable=true"	設定Queues使用的nodes群組
qmgr -c "set queue workq acl_users = user1" qmgr -c "set queue workq acl_users += user2" qmgr -c "set queue workq enabled=true"	指定Queues可以使用的帳號
qmgr -c "s q queue priority = 100"	設定Queues的優先等級

設定 Queues 範例

- 建立Queue

```
qmgr -c "create queue workq"
qmgr -c "set queue workq queue_type = Execution"
qmgr -c "set queue workq started = True"
qmgr -c "set queue workq resources_default.nodes = 1"
qmgr -c "set queue workq resources_default.walltime = 01:00:00"
qmgr -c "set queue workq resources_default.neednodes = workq"
qmgr -c "set queue workq enabled = True"
qmgr -c "set queue workq started = True"
```

Submit Job 範例

- 一般的提交

```
[user1@master ~]$ qsub submit.sh
```

- 指定 queue 名稱

```
user1@master ~]$ qsub -q vip submit.sh
```

- 指定執行時間

```
$ qsub -l walltime=24:00:00 submit.sh
```

- 指定 Job 名稱

```
$ qsub -N hello submit.sh
```

Submit Job 範例

```
#!/bin/bash
#PBS -N job
#PBS -o folder path
#PBS -e folder path
#PBS -q workq
#PBS -l nodes=1:ppn=2
cd $PBS_O_WORKDIR

module purge
source /opt/intel/oneapi/setvars.sh intel64
module load “modulefiles

if [ -n "$PBS_NODEFILE" ]; then
                if [ -f $PBS_NODEFILE ]; then
                echo "Nodes used for this job:"
                cat ${PBS_NODEFILE}
                NPROCS=`wc -l < $PBS_NODEFILE`
                fi
fi

echo "Starting on `hostname` at `date`"
mpirun -hostfile $PBS_NODEFILE -n $NPROCS $HOME/hpl/bin/xhpl > HPL.out
```

Jobs 執行監控及操作

- 查詢所有 Jobs 狀態

```
$ qstat
```

- 查詢所有 Jobs 更多狀態

```
$ qstat -a
```

- 顯示某特定 Job 詳細狀態

```
$ qstat -f JOBID
```

- 查詢某使用者 Jobs 狀態

```
$ qstat -u USERID
```

- 查詢 Jobs 歷史紀錄

```
$ qstat -x
```

Jobs 執行監控及操作

- 查詢某使用者歷史紀錄

```
$ qstat -xu USERID
```

- 顯示某 Job 使用哪些機器

```
$ qstat -xf JOBID | grep exec_host
```

- 顯示系統目前的 queues 狀態

```
$ qstat -Q
```

- 顯示系統目前的 queues 詳細狀態

```
$ qstat -fQ
```

- 顯示系統目前的某 queue 詳細狀態

```
$ qstat -fQ workq
```

Jobs 執行監控及操作

- 暫停某 Job 執行

```
$ qhold JOBID
```

- 釋放某 Job 狀態

```
$ qrls JOBID
```

- 停止執行某 Job

```
$ qdel JOBID
```

- 強制停止某 Job

```
# qdel -W force JOBID
```

- 執行某 Job

```
# qrun JOBID
```

Jobs 執行監控及操作

- 變更某 Job 優先權

```
# qalter JOBID -p 150
```

- 變更某 Job 使用時間

```
# qalter JOBID -l walltime=02:00:00
```

- 重新跑某 Job

```
# qrerun JOBID
```

Nodes 維護及操作

- 列出某 node 的狀態

```
# pbsnodes cn1
```

- 列出有狀況的 nodes

```
# pbsnodes -l
```

- 使某 node 離線進行維護

```
# pbsnodes -o cn1
```

- 恢復某 node 運算服務

```
# pbsnodes -c cn1
```

設定檔及 logs 路徑

- PBS 伺服器使用紀錄路徑

```
/var/spool/torque/server_priv/accounting
```

- PBS 伺服器 logs 路徑

```
/var/spool/torque/server_logs
```

- PBS 排程設定檔

```
/var/spool/torque/sched_priv/sched_config
```

- PBS 排程 logs 路徑

```
/var/spool/torque/sched_logs
```

設定檔及 logs 路徑

- PBS 運算節點設定檔

```
/var/spool/torque/mom_priv/config
```

- PBS 運算節點 logs 路徑

```
/var/spool/torque/mom_logs
```

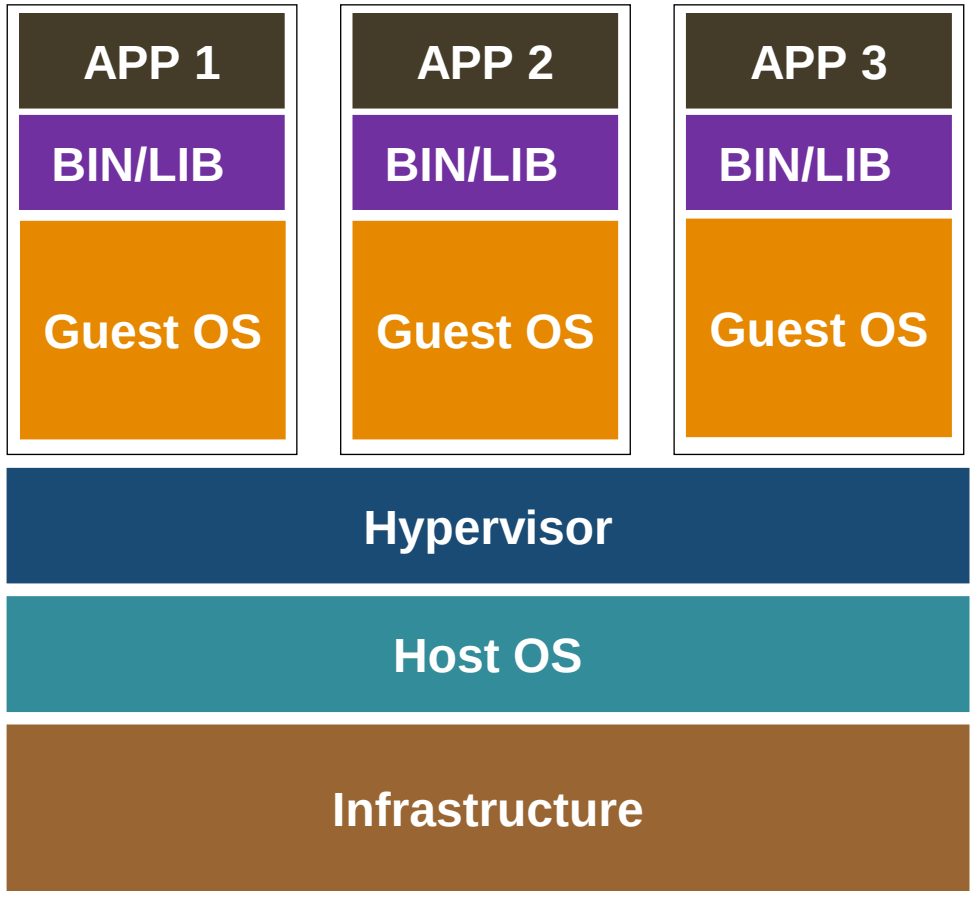
什麼是容器 (Containers) ?

Container 是作業系統輕量級虛擬化技術，用來封裝應用程式和應用程式所依存的函示庫於獨立環境。容器提供標準化輕量級封裝及部署程式的方式，所以可以擺脫基礎設施 (Infrastructure) 的差異在不同地方進行執行。

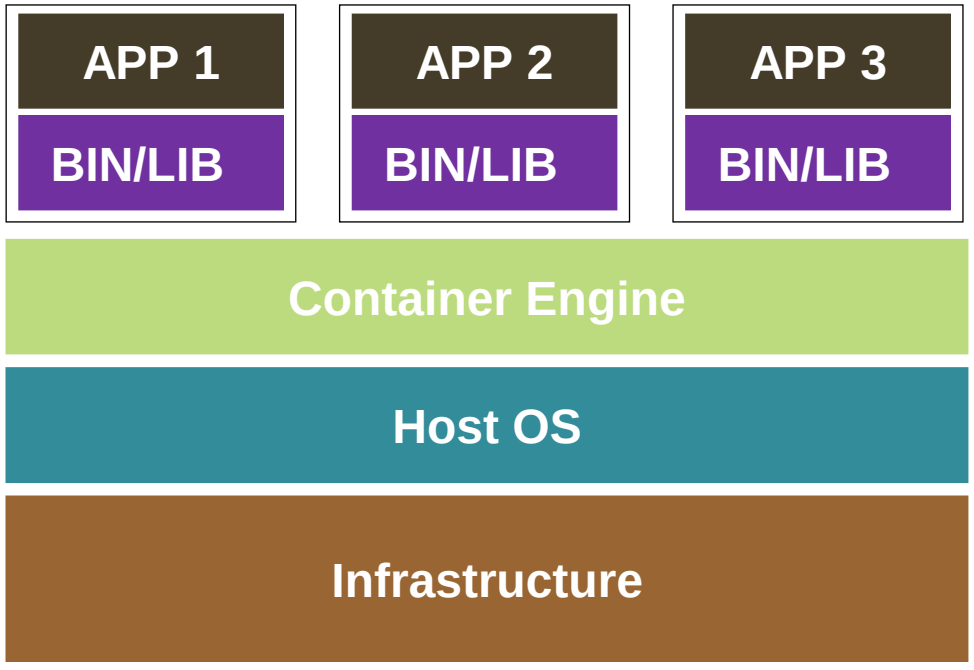
早在 1982 年，Unix 內建的 chroot 機制就是一種 Container 的技術，但直到2013 年，dotCloud 這家公司釋出了一套將 Container 標準化的套裝平台 Docker，才真正開始了 Container 的發展。

目前市面上所有主流的Linux發行版本(如RedHat、CentOS、 Ubuntu 、SLES、OpenSUSE)、虛擬化的龍頭VMware、主流的雲端服務平台(如Amazon AWS、Microsoft Azure、Google Cloud)以及開放原始碼的雲端架構OpenStack都支援Docker。

Container VS VM



Virtual Machine



Container

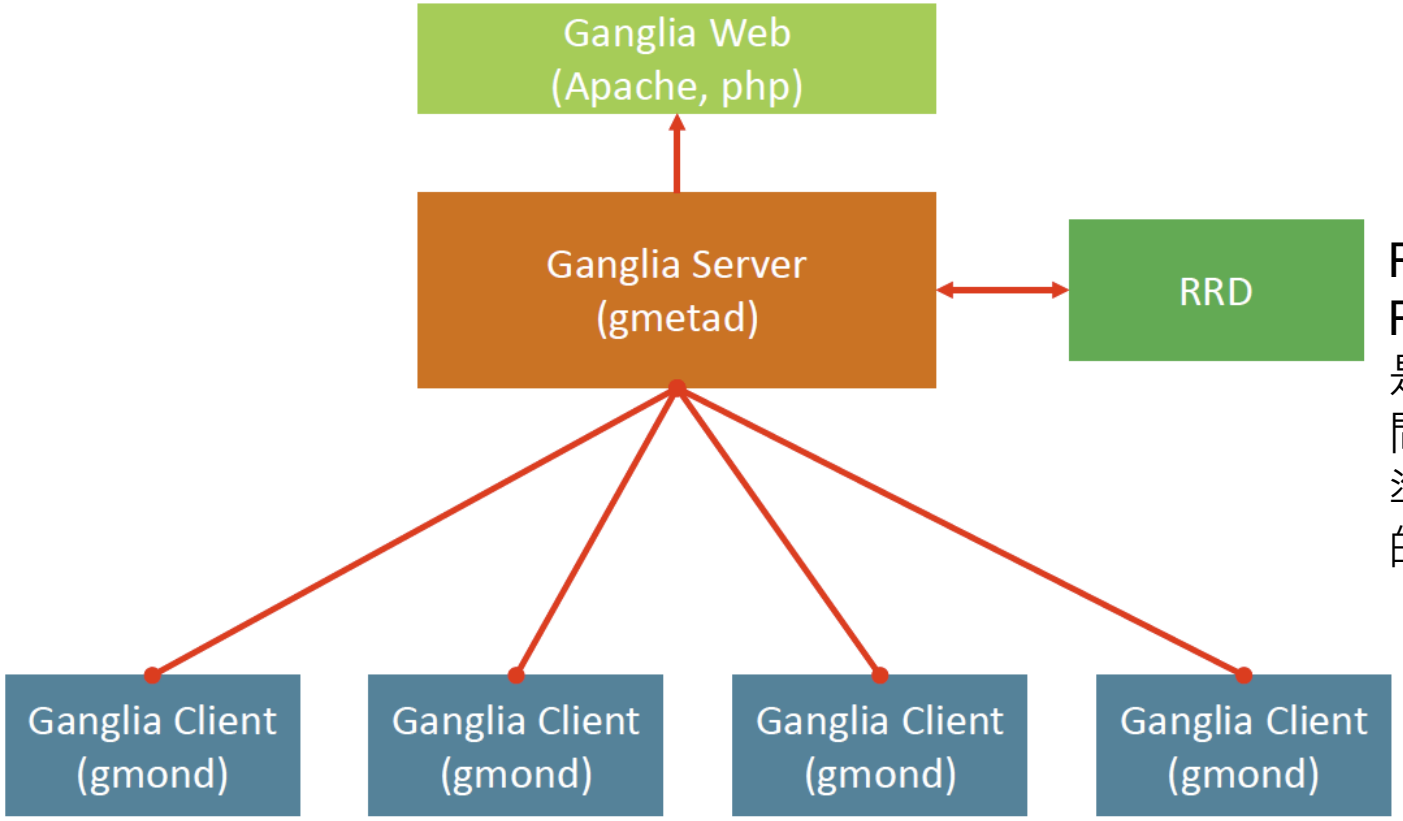
使用 Container 原因

- 快速部署：可更快速的更換版本，且不影響使用者操作
- 適合開發：方便更換不同函式庫或環境版本，可以方便開發者在容器內因應不同環境進行開發
- 使用環境差異大：不同的使用者需要不同環境或作業系統
- 方便移動：封裝好的映像檔 (image) 可以移動套不同平台環境上使用
- 資源充分使用：提高使用效率且容器執行效能直逼裸機效能
- 節省資金：搭配資源管理把忙碌的節點容器進行轉移，可提高機器運轉率

Ganglia Monitoring System

Gangia是一套OpenSource的Cluster監控系統，可幫助管理人員迅速且簡單的瞭解Cluster中各個節點的狀態，並可查詢記錄

Ganglia的架構



RRDtool (Round Robin Database Tool)是開源工具，是用時間序列資料為業界標準、高效能資料紀錄的繪圖系統

Parallel Distributed Shell - pdsh

- pdsh (parallel distributed shell)可以以平行的方式對指定的節點送出指令操作
- 下載pdsh source code

```
[root@master ~]# git clone https://github.com/chaos/pdsh.git
```

- 編譯並安裝

```
[root@master ~]# cd pdsh
[root@master ~]# ./bootstrap
[root@master ~]# ./configure --with-ssh
[root@master ~]# make && make install
```

- 透過 pdsh 在計算節點執行校時指令

```
[root@master ~]# pdsh -w ssh:cn[1-2] "ntpdate -u master"
```

管理上常用的指令

- echo
- date
- find
- xargs
- Filter commands
  - cat, tac, grep, cut, head, tail, nl, join, split, sort, tr, uniq, wc, sed, awk
- test
- kill
- ps

其他管理指令及工具

- lscpu, lspci, lsscsi
- blkid
- parted, fdisk
- mkfs
- fsck
- htop
- iftop
- wireshark

Shell Scripts

電腦程式使用的文字檔案，內容由一連串的 shell 命令組成，經由 Unix Shell 直譯其內容後運作。被當成是一種手稿語言來設計，其運作方式與直譯語言相當，由 Unix shell 扮演命令行直譯器的角色，在讀取 shell 指令碼之後，依序執行其中的 shell 命令，之後輸出結果。利用 shell 指令碼可以進行系統管理，檔案操作等。

六個字「執行程式列表」

節錄：https://zh.wikipedia.org/wiki/Shell%E8%84%9A%E6%9C%AC

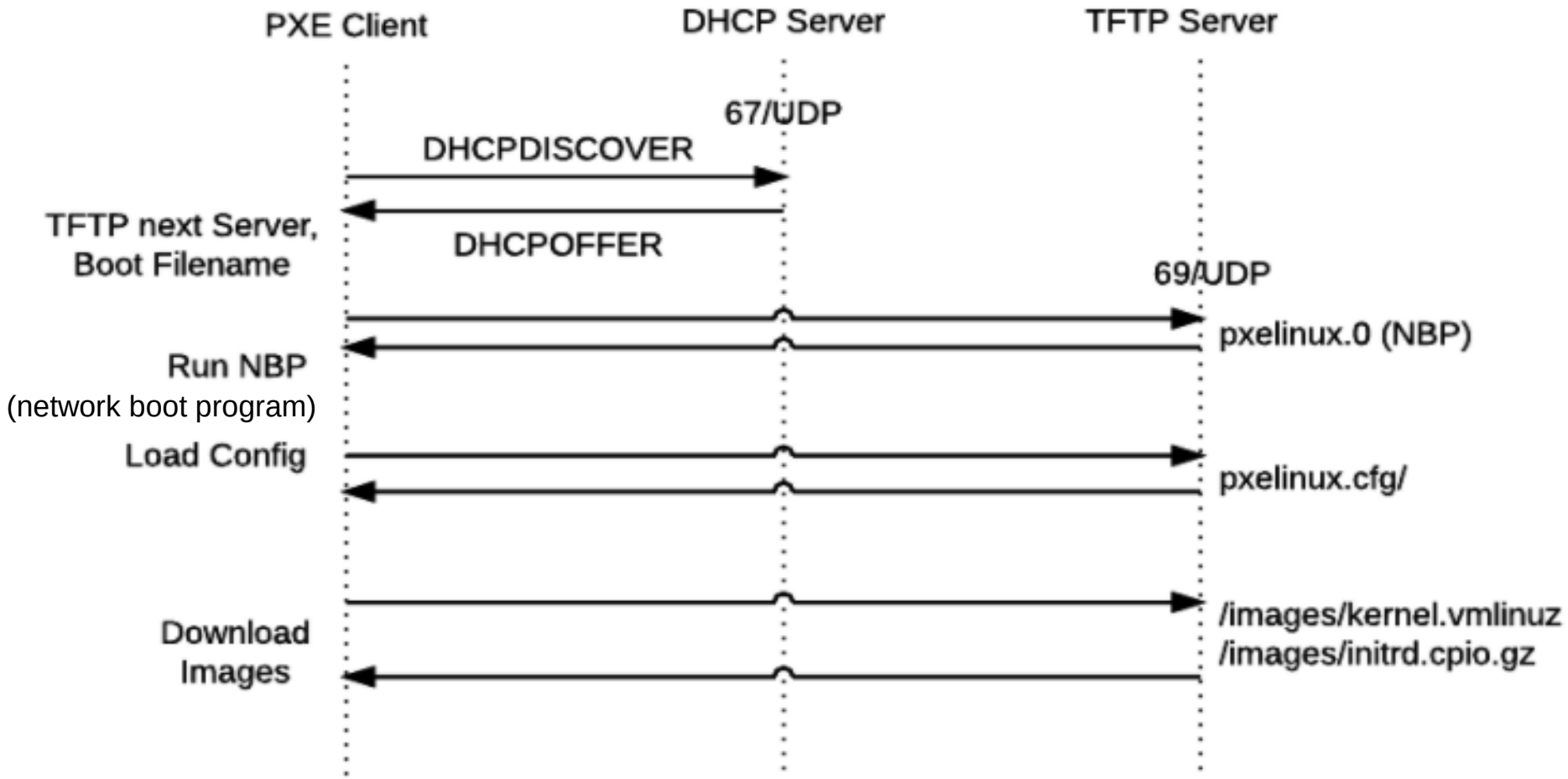
Crontab 讓Shell Scripts定時執行

```
# vi /etc/crontab
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root
HOME=/

# For details see man 4 crontabs

# Example of job definition:
# .---------------- minute (0 - 59)
# |  .------------- hour (0 - 23)
# |  |  .---------- day of month (1 - 31)
# |  |  |  .------- month (1 - 12) OR jan,feb,mar,apr …
# |  |  |  |  .---- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat
# |  |  |  |  |
# *  *  *  *  * user-name command to be executed
```

作業系統部署工具-Cobbler

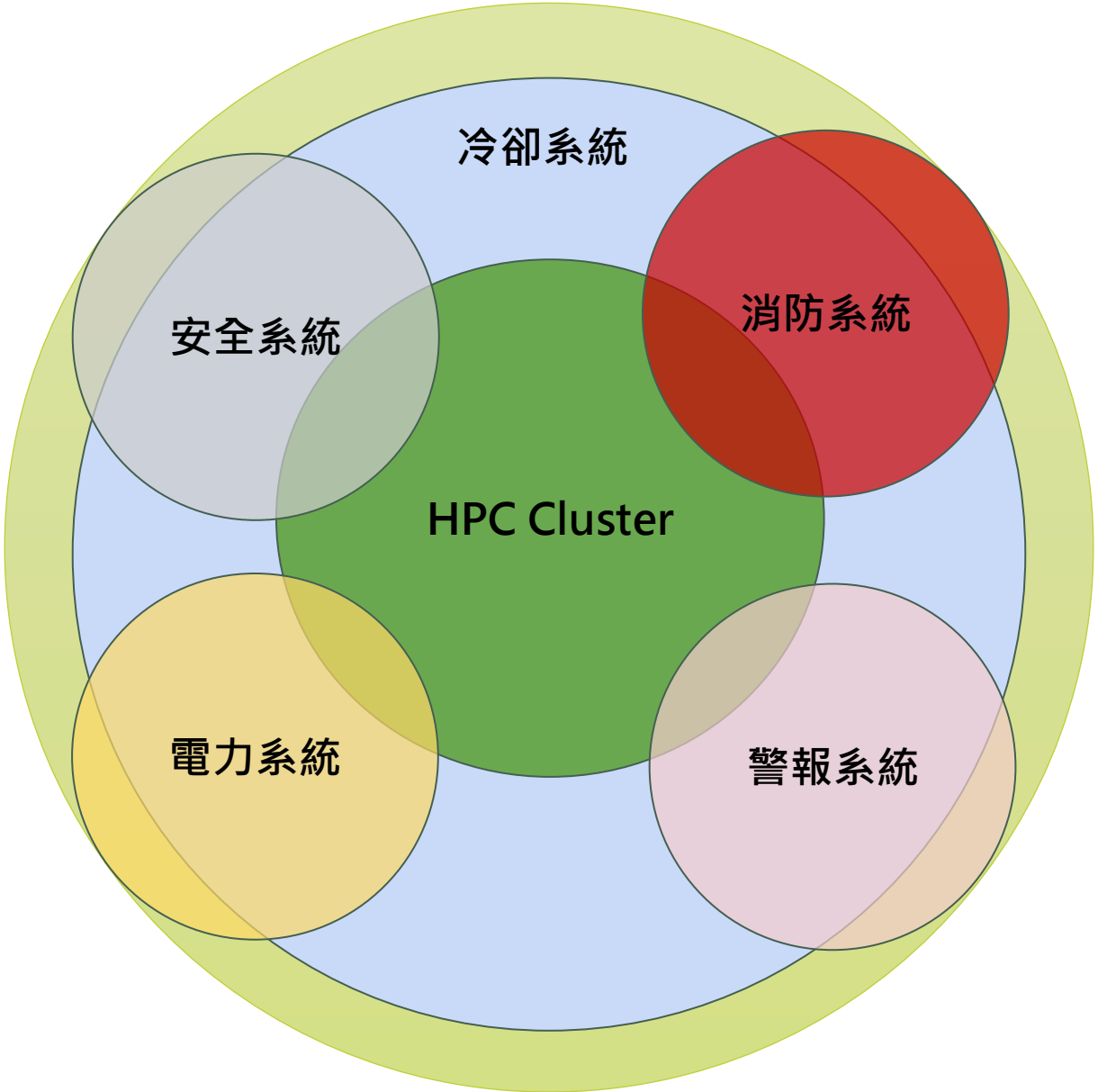
- Cobbler is a Linux installation server that allows for rapid setup of network installation environments. It glues together and automates many associated Linux tasks so you do not have to hop between many various commands and applications when deploying new systems, and, in some cases, changing existing ones. Cobbler can help with provisioning, managing DNS and DHCP, package updates, power management, configuration management orchestration, and much more.
- Cobbler 3.2.x – python3
- Web: https://cobbler.github.io/

PXE Boot 流程

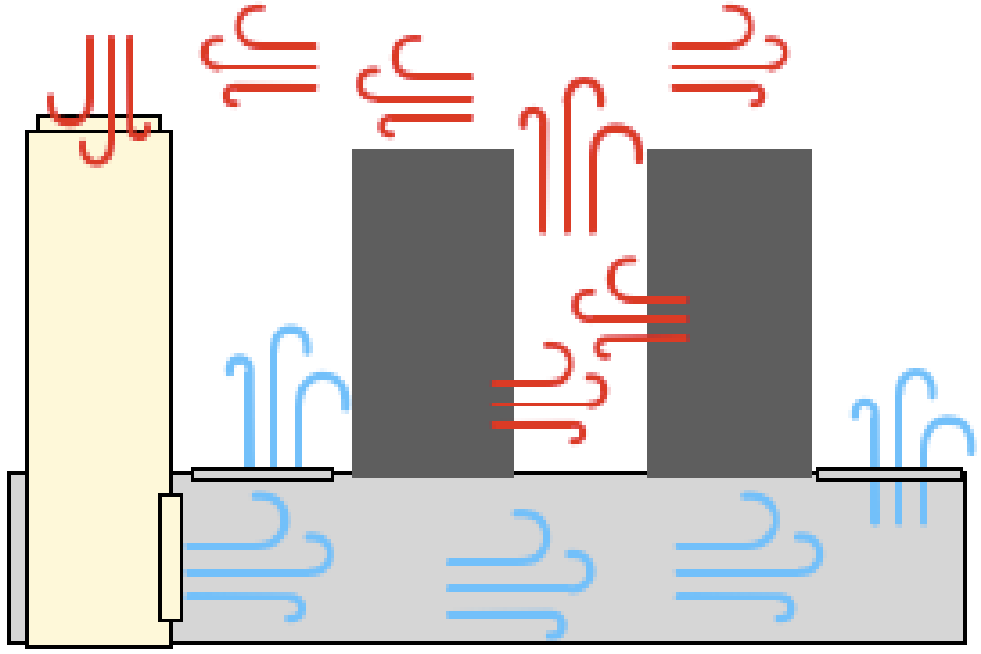


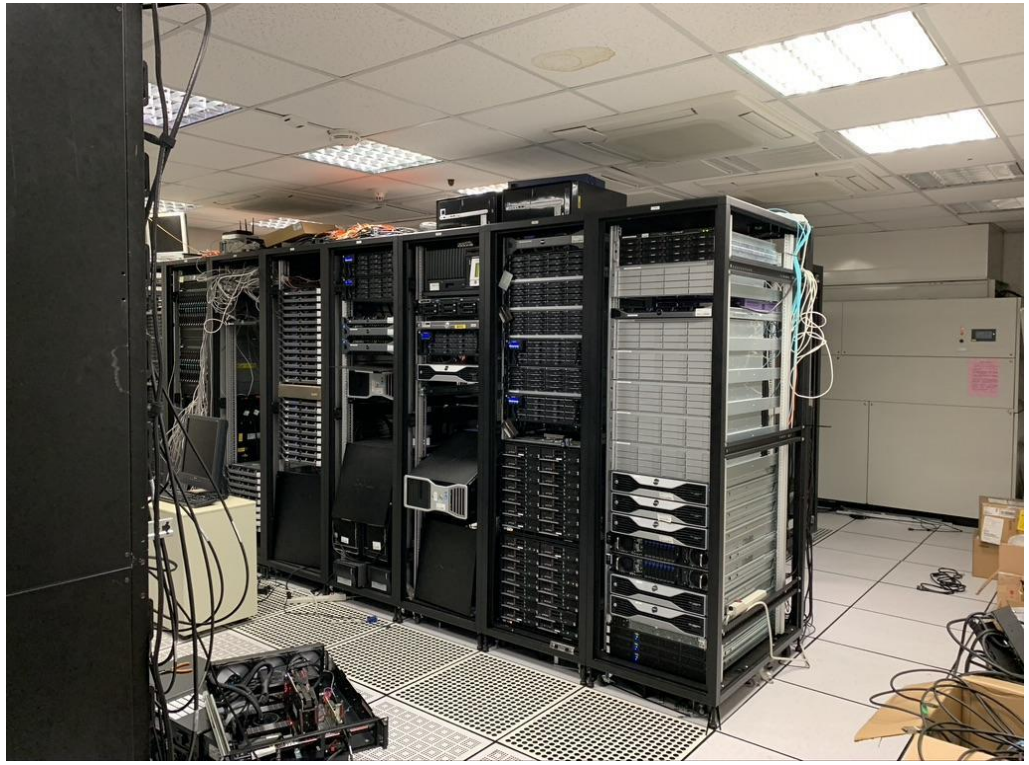
圖片來源：https://coreos.com/matchbox/docs/latest/network-booting.html

HPC 周邊硬體環境

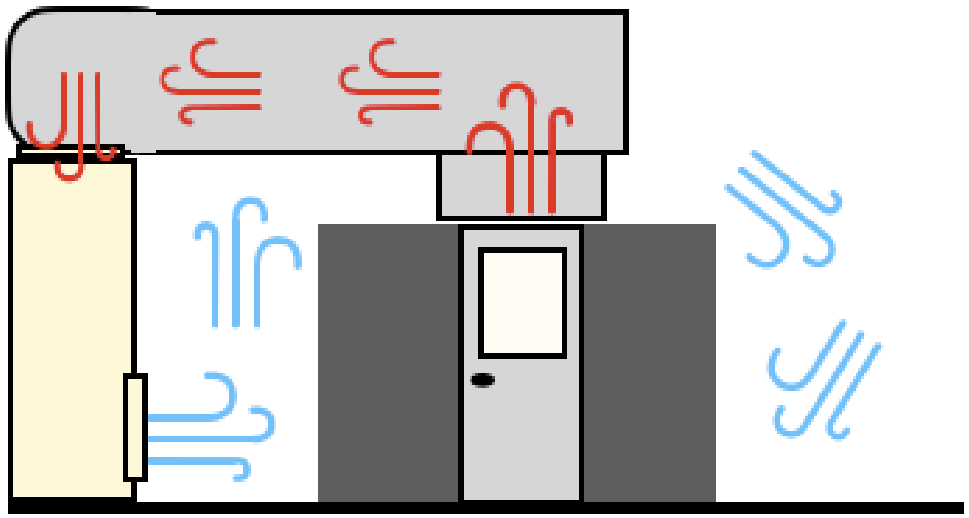


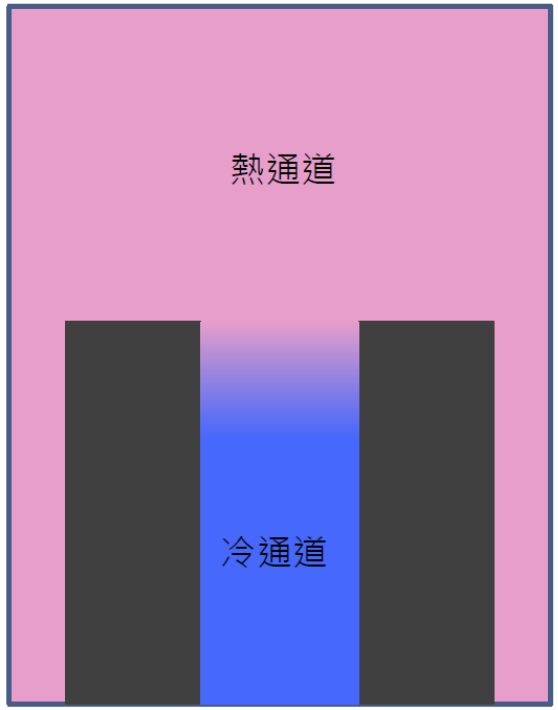
冷卻系統-房間級制冷

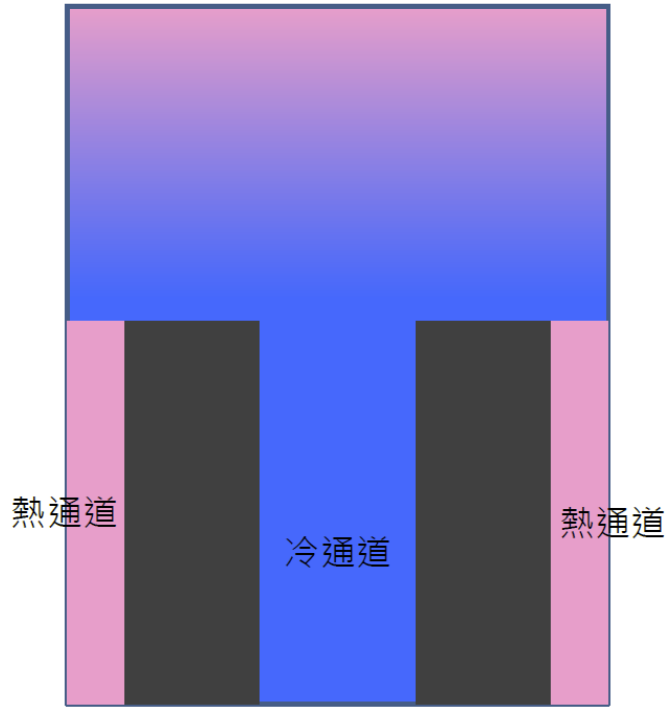




冷卻系統-行級制冷







冷卻系統-機櫃制冷

圖片來源：資料中心房間級、行級和機櫃級製冷方式的選擇

冷卻系統-水冷散熱系統

冷水背板

- 常溫水冷卻
- 冰水冷卻
- 冷媒冷卻

圖片來源：氣象局導入水冷式冷卻背板解決高運算散熱挑戰

冷卻系統-自然進氣

- 低能源消耗
- 需注意空氣濕度
- 需過濾空氣中懸浮微粒

在台灣無法使用自然進氣冷卻

2013年的趣聞
https://www.ettoday.net/news/20130611/221111.htm

臉書機房養出「室內雲朵」　工程師臉綠：開始下雨了…

國際中心／綜合報導

臉書(Facebook)在美國奧勒岡州的第一數據中心曾經出現個令人哭笑不得的大麻煩，機房裡面形成真正的「室內雲朵」，更糟的是，它還在伺服器上方「下雨」。**工程師說，「有那麼幾分鐘，機房裡有兩朵雲，一朵營運社群網路、一朵從上頭澆水……」**

▼圖為藝術家Berndnaut Smilde製造的室內雲。(圖／取自berndnaut.nl)

基礎設施主任傑伊派瑞克(Jay Parikh)說，「我接到一通電話，工程師告訴我『傑伊，數據中心裡有一朵雲！』我當時想說，是在室外嗎？但對方堅持在機房裡。**接下來我聽見一陣騷動，電話那頭無助地說『它開始下雨了！』」**

冷卻系統-水冷散熱系統

HP Apollo 8000

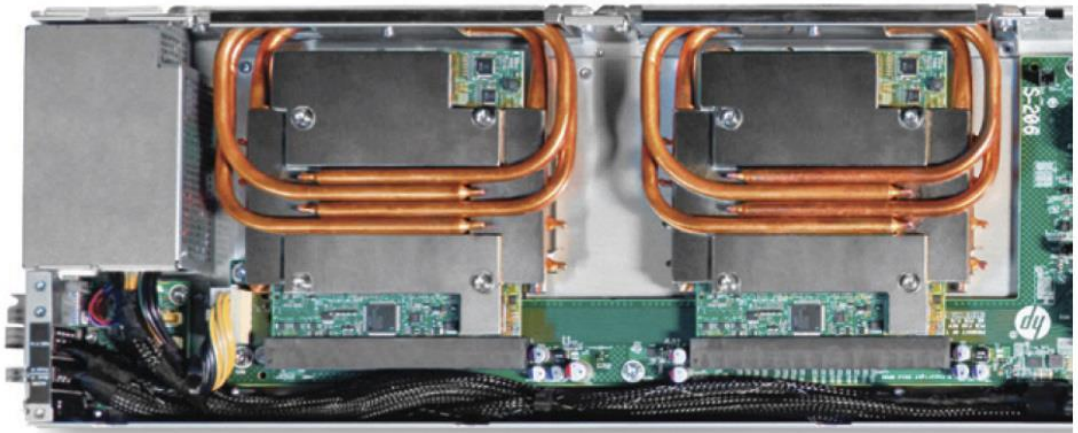


Photo captured from data sheet of HP Apollo 8000 system.

FUJITSU PRIMERGY CX400 M4

Photo captured from Fujitsu website.

CoolIT systems - Direct Contact Liquid Cooling

Photos captured from “Rack DCLC Product Guide” and “CoolIT Rack DCLC in HPC”.

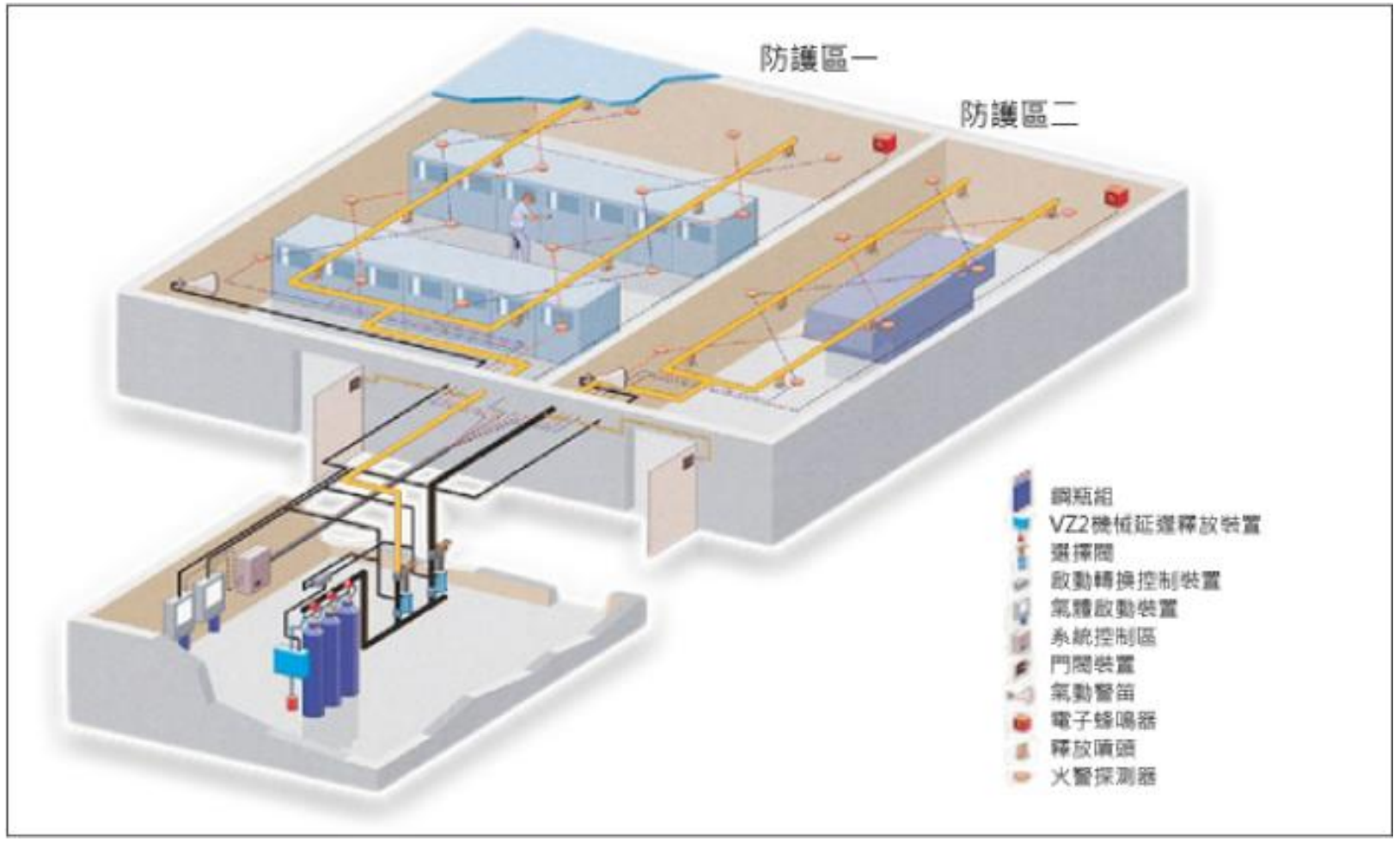
電力系統

- 不斷電系統 (UPS)
  - 容量以VA計算
  - 穩壓
  - 防止突波
  - 預防斷電
- 機架配電單元(PDU)
  - 以機架為單位配電
  - 穩壓
  - 過載跳脫
  - 監控單位設備用電(監控型)

圖片來源：
Vertiv (https://www.vertivco.com/zh-TW)
https://www.computerage.com.au/products/apc-rack-pdu-2g-metered-zerou-20a-208v-1045841

消防系統

- 自動化滅火
- 機械延遲釋放裝置
- 滅火警示
- 氣體滅火



圖片來源：中興保全|消防安全解決方案

警報系統

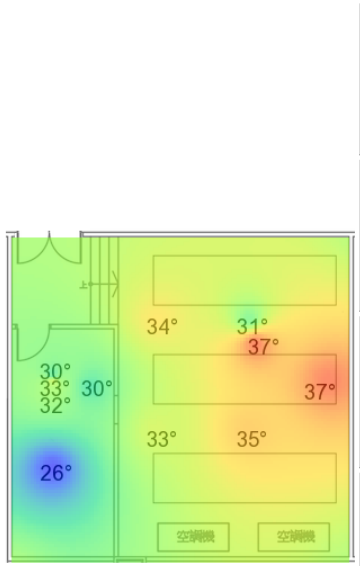
- 電力異常警報
- 溫度異常警報
- 環境設備異常警報
- IT 設備異常警報
- 安全警報

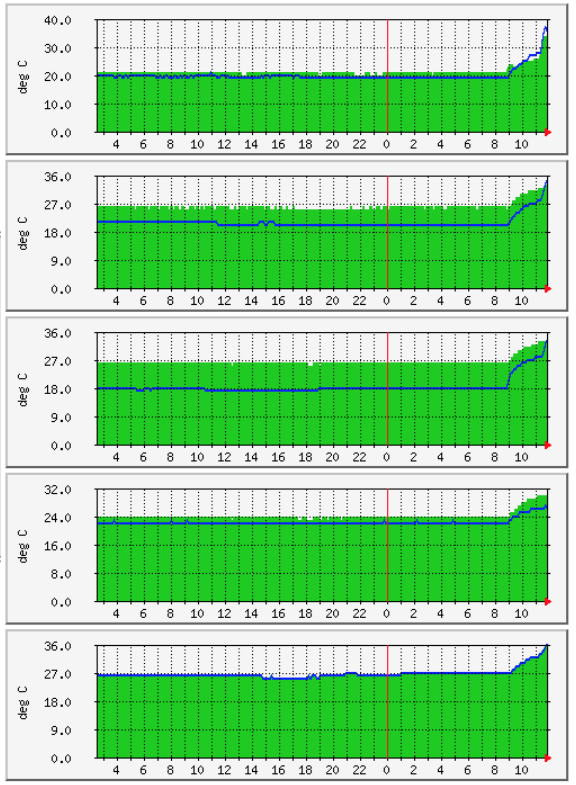
安全系統

- 環境監視
- 環境影像錄影
- 自動報警
- 安全發報

環控系統

集中控管所有機房環境設備及記錄，並可控制環境設備





無人值守機房

管理人員可由遠端處理大部份的機房狀況，不需留守在機房

Aten CN8000

Aten KL116VN

圖片來源：ATEN KVM Switch 多電腦切換器的領導品牌，提供個人與企業全方位的解決方案