分散式檔案系統 OpenAFS 的實現與研究

期間：96 年 9 月 18 日至 97 年 6 月 4 日止

系別：電機系　　學生：施谷燁 林昌儒 林家毓 劉力仁

一、背景

由於在課堂上需要用到在伺服器上的軟體，但若太多人使用伺服器會負荷過大，而使操作速度明顯下降，為解決此一問題，我們發現可以利用分散式檔案系統的優點來克服，故我們選擇了在國外大學頗為普遍的基礎服務，OpenAFS 來解決此一問題，而此服務更可以進一步提供使用者在任何電腦上使用其儲存在伺服器上的檔案，提供便捷的檔案服務。

OpenAFS 全名為 Open Andrew File System，是一種分散式網路檔案系統，能在區域及廣域網路中，提供有效分享檔案及系統資源的服務。原本為卡內基美濃大學 Andrew 計畫的一部份，始於 1983 年，其設計概念是希望整合校園內的計算資源，利用低階電腦整合計算環境、減少工作站負荷及浪費所發展而成。

分散式系統的優點如下：
經濟效益：整體的效能比大型電腦為佳
速度：整體加成性的運算能力比單一的電腦
支援分散作業：有些應用系統必須分散作業
可靠性高：部分電腦故障不會造成系統癱瘓
擴充性佳：可逐漸加強運算能力而不致產生大的影響

OpenAFS 除了繼承上述的優點之外，更改善了 Network File System 的缺點，改進其在大網域上的低效率。因為 AFS 提供了客戶端的快取的機制，使得檔案存取效率大幅提高，有效減低網路流量、頻寬，且 AFS 檔案系統內的檔名對全世界的 AFS 系統而言是唯一的，從單一機器就能存取全世界的 AFS 檔案系統，不會造成檔案命名重複的問題。

類別	AFS	NFS
檔案名稱	唯一檔案名稱	不同檔案名稱
檔案位置	自動	掛載點
效能	客戶端快取機制	無快取
可擴充性	小到非常大	小到中等
安全性	特優於廣域網路 使用Kerberos 5 ACLs	區域網路內為佳 使用者帳號未加密
可及性	資料及AFS資訊	無ACLs
備份	隨時自動備份	無複製
檔案編排	使用volumes(檔案分組)	需使用UNIX工具
系統管理	不隨client影響 每一client端皆可	隨文件變動 Client端需更新 需連上server

此圖譯自 Transarc Corp.。分別比較 AFS 與 NFS 的優缺點

OpenAFS 特色：

Use Volume

可視為一個 AFS 系統內的檔案分割。一個 Volume 包含了許多檔案和資料夾。

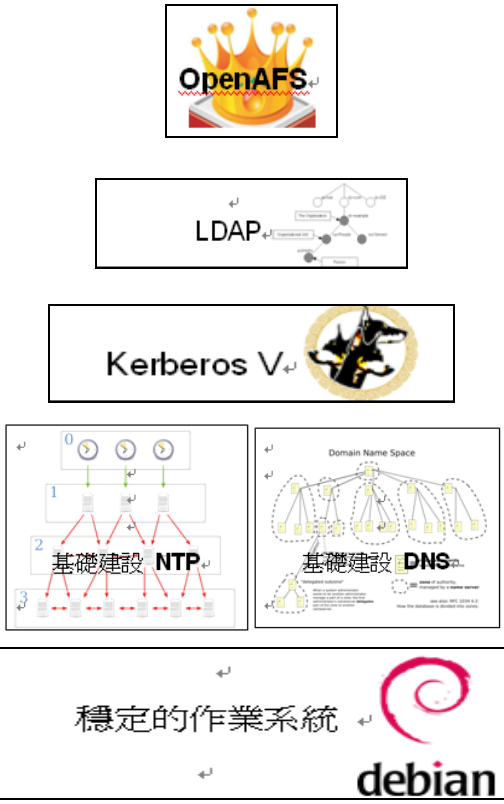
Volume 的大小比實體硬碟的分割小，其原因在於此大小可以在不同的檔案伺服器間傳送、複製，如不同實體的伺服器中有相同的 Volume，就可以達到負載分散的功能。且 Volume 在複製過程中不會造成使用者的中斷，僅僅在結束或開始時造成些許的速度延緩。

Transparent to user

使用者不需要知道檔案的實體位置在哪一台實體的機器上，伺服器會告知使用者檔案可從哪台伺服器取得。不同於 NFS 的檔案系統，使用者需要知道知道檔案在哪台機器上的詳細位置。AFS 提供了使用者的便利，不管檔案在哪台機器上，都會由 Database Server 來告訴使用者所需要的檔案在哪一個實體機器中的哪一 Volume，使用者就能直接拿到資料。

二、方法與結果

在實現 OpenAFS 前仍需要許多基礎服務，如 NTP 時間同步服務、DNS 名稱解析服務、Kerberos 安全認證服務、LDAP 目錄服務，在此四個服務之上方能建置 OpenAFS 檔案系統。如下圖所示：



作業系統選擇

我們選擇 Debian Linux 作為我們伺服器的作業系統，因為相較其他 Linux 版本而言，Debian 相對穩定、擁有較長時間的套件支援且屬於免費發行的版本，故適合當伺服器的作業系統。

2.1 實驗環境

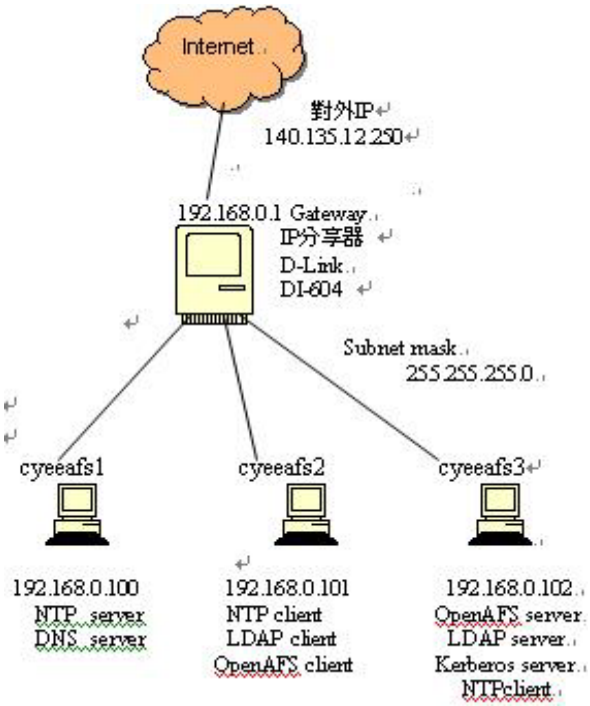
作業系統： Debian linux 2.6.18-5-686 #1 SMP i686 GNU/Linux CPU : Pentium 4 1.6GHz			
Hostname	RAM (Mbs)	SWAP (Mbs)	Debian版本
cyeeafs1	640	1333	2.6.18-5-686
cyeeafs2	640	1333	2.6.18-6-686
cyeeafs3	256	956	2.6.18-5-686

OpenAFS 軟體：OpenAFS-1.4.2-6etch1

LDAP 軟體：2.3.30-5+etch1。V1.9 版本

Kerberos 5：1.4.4-7etch5

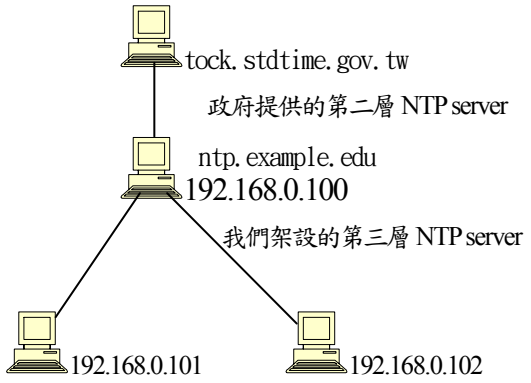
2.2 私有網路架構



我們一開始使用私有子網路的概念建置我們的網路環境，減少設定名稱伺服器的麻煩、及降低安全性的風險。

2.3 NTP 時間伺服器

時間伺服器主要功能是要維持各伺服器間的時間同步，使得 Kerberos 安全認證服務中的"票據"不會因為各台機器的時間不同，導致"時間戳記"相距過大而認證失敗。



上圖為我們的 NTP 架構

在另外兩台 NTP client 端輸入#ntpq –p 觀察其時間同步的狀況如下頁所示：

```
remote : *ntp.example.edu 上一層的NTP Server
refid  : 220.130.158.71
                參考的上一層 NTP 主機的位址
st : 3          指此client 端屬於第幾階層
t  : u
when  :424      秒前曾做過時間更新
poll  :1024     下一次更新在幾秒鐘之後
reach :377 已經向上層 NTP 伺服器要求更新的次數
delay :3.553    網路傳輸過程當中延遲的時間
offset:-2.912   時間補償的結果
jitter:4.529 Linux 系統時間與 BIOS 硬體時間的
差異時間
```

上述的單位都是 10^{-6} 次方秒，所以我們從灰底中的數據可得知即使在網路狀況不穩定的狀況下，電腦之間最少也可以達到 10^{-3} 次方秒內的精確度，確實符合我們時間同步的要求。

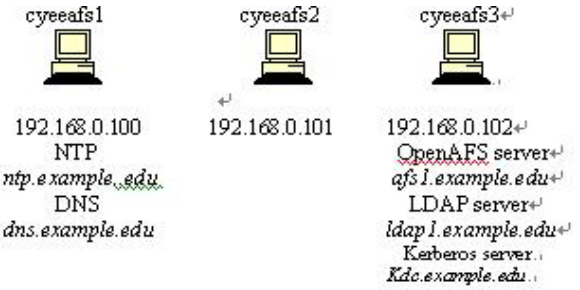
2.4 DNS 名稱伺服器架設

DNS 全名為 Domain name service，因為電腦只認得 IP address，但卻不容易被記憶，故此服務即是將 IP address 轉換成人類容易記憶的網路名稱。

如我們要到 tw.yahoo.com 網站時會先 query DNS Server 取得相對應的 IP address 為 203.84.202.164，此時方為電腦認得，始可連線。

因為我們在架設 OpenAFS Servers 時需要有對外容易記憶的網路名稱。

下圖為本系統所有 Server 的網址名稱



以上設定需要在 DNS server 中更改，/etc/named.conf 設定檔，也須將 client 端的 /etc/resolv.conf 內 nameserver 改成 192.168.0.100。

在 client 端輸入#host –v afs1.example.edu 則可得右上方之結果，故 DNS server 運作正常。

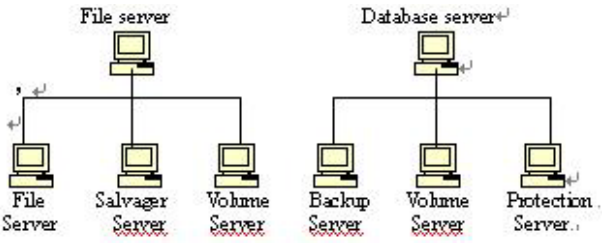
DNS query afs1.example.edu 的結果：

```
;; QUESTION SECTION:
;afs1.example.edu.   ←問的問題   IN      A
;; ANSWER SECTION:
afs1.example.edu.       900     IN      A
192.168.0.102       ←得到的答案IP
```

2.5 OpenAFS 實現過程

Server 架構

OpenAFS 檔案系統中主要分為 File Servers 和 Database Servers，以及每一台實體伺服器上皆會有的 Bos Server。



Bos Server—Basic Overseer Server，監控在此台伺服器上的 AFS 所有程序執行正確，若有錯誤則協助重新執行程序。

File Server—管理檔案、資料夾儲存在哪個 Volume 中，若使用者被正確的授權，就能由此獲得檔案。

Salvager Server—若在 File Server 上 Volume 內的資料毀損，提供救援服務。

Volume Server—管理此台伺服器內 Volume 的建置、移動、複製和刪除。

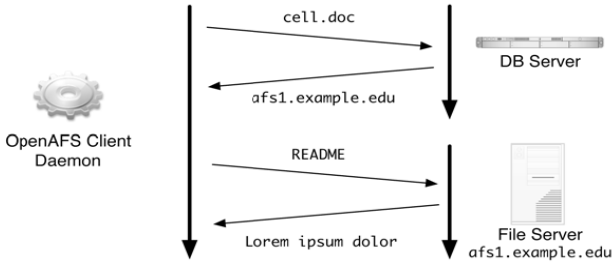
Database Servers—含有 Backup Server、Volume Location Server 和 Protection Server。

Backup Server—管理所有在 Database Servers(含以下兩種 Server)上的備份程序，儲存成 backup database。

Volume Location Server—追蹤所有 Volume 在哪台伺服器上有，儲存成 Volume Location Database。

Protection Server—管理使用者和群組對 AFS 的使用權限。

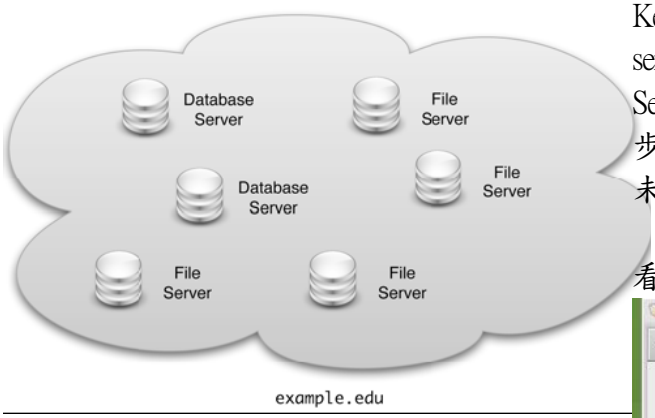
我們用圖來解釋此 query 達成的過程



此圖取自 Distributed Services with OpenAFS

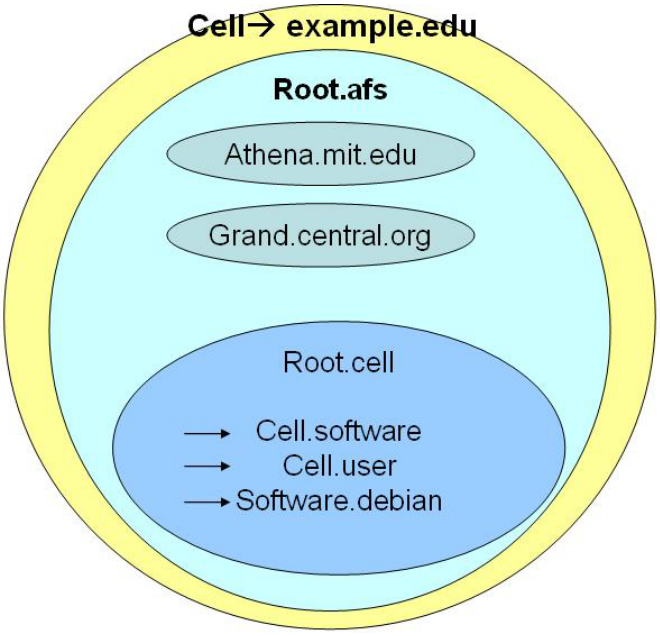
Cell 概念

Cell 是由 AFS 中很多的 Database、File servers 組合而成。例如一個組織所建立的一個 AFS 檔案系統即可視為一個 Cell。以下是概念圖 cell→example.edu：



此圖取自 Distributed Services with OpenAFS

我們在 OpenAFS 中的 Cell 結構：



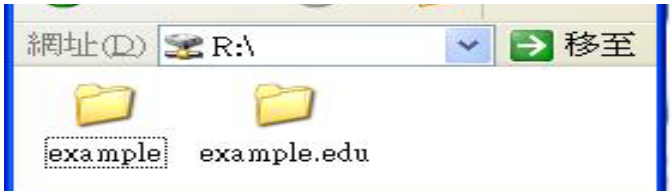
其中 root.afs 包含了所有額外掛載上來的 foreign Cell 如 anthena.mit.edu(麻省理工學院的 AFS Cell)，以及自己的 cell，root.cell。

而 ⟶ 下指的是我們 local 端 cell 內的 volume 名稱，可以用 volume Server 的命令#vol listtvol afs1 來證明。

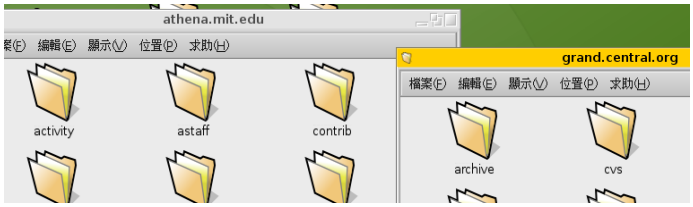
值得特別注意的是，當我們建好 volume 後要記得把他 mount 到/afs/example.edu 下，以免無法使用。且任一個 volume 都可以設定最大儲存空間，例如此功能便能限制 cell.user 使用者儲存檔案的上限。

除了上述最複雜的創建 volume 之外，更包含了一開始的建立快取資料夾、使用 Kerberos 來認證 AFS 的權證，及設立 database server 等步驟，至此我們的 File Server 和 Database Server 已大致完成，除了建立 volume 此複雜的步驟外，其餘過程因過於冗長，故在此文章中未提及。

由於以上實現的結果，我們可以在系統中看到以下的顯示



從使用者的觀點而言，可以看到所有的檔案，而且可以連結到其他的 cell 中去使用很方便。



從上面兩個 MIT 與 GRAND 的內可以發現，雖然我們不知道其檔案確實的位置在哪台伺服器上，但對使用者而言卻是全部自動的，且相當方便，只是在速度上因為距離過於遙遠而有些緩慢，但也可以顯示出 AFS 在廣域網路上的可用性。

三、討論

OpenAFS 相對較常使用的 NFS 來說是較複雜的架構，尤其是包含了 Kerberos 認證系統和 LDAP 資料夾管理系統，但也相對的提供了較好的服務。解決了 NFS 在過多使用者登入使用下造成的效率低落問題，此問題可由擴充 AFS 的 Sever 數量來解決，AFS 會自動將使用者導向較不忙碌且有一樣檔案的伺服器上，而其快取功能更將常用的檔案存在客戶端電腦中，不會每次使用者要使用檔案時，都到 AFS 的 Server 上取得，適度的減少了網路流量。

遇到的問題：

在 LDAP 中，因為安全性的問題，我們不將密碼儲存在設定檔下，不論是明文或是單向加密過的密文，而統一使用 Kerberos 系統，但此嘗試卻是認證失敗，如下所示：

```
afsl:/# ldapsearch -LLL
SASL/GSSAPI authentication started
ldap_sasl_interactive_bind_s: Local error (-2)
        additional info: SASL(-1): generic
failure: GSSAPI Error: Miscellaneous failure
(Server not found in Kerberos database)
```

上述說明了 SSAL/GSSAPI 認證確實有開始執行，但出現 Server not found in Kerberos database 的訊息，說明可能是設定檔或是 kerberos 中的 principle 有問題導致，此認證失敗連帶導致在 OpenAFS client 端的讀取權限也造成問題，產生使用者的存取問題

在解決上述兩問題之後，我們試圖建立的分散系統也大致完成，僅剩下備份、救援伺服器尚未實現。

四、結論

4.1 支援多種作業軟體試用於校園環境

AFS 支援不同作業系統，我們分別以 Windows 作業系統和 Linux 系統來嘗試，而結果均可以運作。在透過 kerberos 系統認證之後，在兩個不同的作業系統中，可以看到 example.edu 及 grand.central.org 和 athena.mit.edu 三個資料夾，代表我們可以在任何安裝 OpenAFS client 端程式的電腦上，不論其為何系統，都不會影響我們來使用 AFS 的檔案服務。

對於一個擁有眾多不同作業系統電腦的單位而言，管理及使用 AFS 的服務是相當方便的。

4.2 Transparent to user

此部份則可由以下證明，我們在兩台不同的"檔案"伺服器上有兩個不同的 volume，但在客戶端的觀點而言，使用者不需要知道檔案實際在哪個電腦上，從資料夾裡面看都是一樣的，使用者不用重新設定檔案所在電腦的 IP 來取得檔案。

五、未來展望

由於目前我們使用的仍是私有網路，將來需要對外公開以便使用，故其相關伺服器的 IP 都需要做變更，而 AFS 的 cell Name 也須重新命名以加入全球的 AFS 公開使用。在備援服務上，像是 NTP、DNS、Kerberos、LDAP 及 AFS 中的 file server、database server 等等，都需要因應此分散式服務客戶的多寡來增加備援，如產生第二台、第三台相同服務的伺服器，以免因單一伺服器毀壞造成服務終止，而其與原主要伺服器間的同步問題及檔案伺服器間 volume 的置放、移動問題仍需要去了解及研究，以免正式上線後管理上造成問題。

在使用者方面，只要電腦上有 AFS 的客戶端安裝的話，則使用者即可以在任何地方存取它需要的檔案，甚至是在家裡！例如使用者僅需要輸入他的學號跟密碼後，便能登入屬於它個人的資料夾，裡面有他的作業、個人設定等資料，不用換了一個地方就需要記憶另一組使用電腦的密碼，而管理者也可以針對他的資料做容量上限的管理，對於使用軟體部分，也可以針對資料夾的權限存取來做控制，如免費軟體的權限是對全部使用者公開的，而某些軟體僅限定給某系的學生擁有登入使用的權利，如學號的開頭。如此便可以減少記憶多組電腦密碼的困擾，及妥善的管理及使用軟體及檔案，達到我們原本實現此一系統的目的。