# 行政院國家科學委員會專題研究計畫　成果報告

## 植基於 Linux 之高效能安全叢集檔案系統之設計與實作(I)

計畫類別：個別型計畫
計畫編號：NSC91-2213-E-002-105-
執行期間：91 年 08 月 01 日至 92 年 07 月 31 日
執行單位：國立臺灣大學電機工程學系暨研究所

計畫主持人：雷欽隆

計畫參與人員：邱瀚輝、蔡攀龍、李俊頡

報告類型：精簡報告

處理方式：本計畫可公開查詢

中　華　民　國 92 年 12 月 8 日

# 行政院國家科學委員會專題研究計畫成果報告

# 植基於Linux之高效能安全叢集檔案系統之設計與實作(I)

# Design and Implementation of a Secure and High Performance Linux-Based Cluster File System (I)

計畫編號：NSC 91-2213-E-002-105
執行期間：91年8月1日至92年7月31日
主持人：雷欽隆　台大電機系教授

## 一　中文摘要

本計畫的主要目的是提出一個新的叢集檔案系統架構，除了具有現今叢集檔案系統的大部分優點之外，並選擇Linux作業系統平台加以實作，以期滿足今日科技產業的需求。我們的叢集檔案系統除了要能支援一般桌上型個人電腦上常見的作業系統之外，也需能支援如行動裝置這種計算能力低、傳輸頻寬小的平台。爲了達到這個目的，我們需要克服的重要研究課題包括：高效能網路傳輸機制，以降低通訊成本與達到支援行動裝置的目的；一致性模型，以解決檔案共享的問題；檔案系統容錯，以增加系統可靠度；物件檔案系統，以增加系統的可用性；以角色爲基礎的存取控制機制，以增加管理上的彈性與使用上的便利；檔案系統與網路傳輸用的加密模組，以增加資料儲存與傳輸的安全性等等。有了這個高效能的叢集檔案系統之後，除了可以解決在不同平台間共享檔案的問題，也可以拿來作爲伺服器叢集的基礎架構，在上面建置分散式資料庫伺服器、全球資訊網伺服器等等。還可以拿來作爲協同計算環境的平台，利用叢集檔案系統的網路傳輸與資料同步的能力，來達到支援協同計算的目的。

**關鍵詞**：叢集檔案系統、一致性模型、檔案系統容錯、物件檔案系統、以角色爲基礎的存取控制、加密檔案系統

**Abstract**

The main goals of this project are to propose a new cluster file system architecture which consists of features from existing systems and select Linux as the platform to implement on in order to meet the needs of current technology industry. Our cluster file system not only supports operating systems that runs on desktop personal computers, but also platforms which have low computing powers and narrow communicating bandwidth, such as mobile devices. To meet this goal, the topics that we have to solve including: high performance network transmission schemes to reduce the communication cost and to support mobile computing devices, consistency models to solve file-sharing problems, file system fault-tolerance mechanisms to improve the system reliabilities, an object-based file system to increase the system usability, a role-based access control scheme to improve flexibility of administration and convenience of use, and encryption modules for file systems and underlying network protocols to increase the security of data storage and network transmission, and so on. Once this high performance cluster file system is ready, we are not only able to solve the cross-platform file sharing problem, but also use this system as an infrastructure for server clusters, such as distributed database servers, world-wide web servers, and so on. Also, it may be used as a platform that a collaborative computing environment can build on. By taking advantage of the capability for network transmission and data consistency of the

cluster file system, we may reach the goal to support collaborative computing.

**Keywords:** Cluster File System, Consistency Models, Fault Tolerant, Object-Based File System, Role-Based Access Control, Encrypted File System.

## 二 緣由與目的

隨著網際網路的盛行，以及微處理機能力的大幅提昇，使得以計算機網路爲主體的分散式計算，愈來愈受到資訊界的重視。昂貴的大型主機與工作站平台，已經不再是解決計算能力需求的首要選擇。

一個分散式計算平台，可以不用架構於分散式檔案系統之上。在這個組態下，有些平台是將資料存放在專屬的檔案伺服器上，由伺服器來處理檔案共用與資料同步的問題。有些平台則是將所要處理的問題切割，每個客戶端機器只負責保管自己專屬的資料區塊，等到計算完畢再與其他客戶端機器整合。

專屬的檔案伺服器不僅價格與維護成本高昂，而且容易變成系統中的單一錯誤發生點。當檔案伺服器需要離線進行維護工作，或者是發生錯誤時，整個計算工作將會因此而被迫中斷。如果採用將資料切割再發放給客戶端機器這種做法的話，則會大大地限制這個計算平台的應用範圍。

所以，分散式檔案系統對於一個分散式計算環境而言，是必須的。然而，在資訊界專注於計算能力與計算方式的改進之際，關於分散式儲存媒介的發展似乎是被忽視了。

在作業系統這個領域裡，分散式檔案系統的研究已經行之有年了。一個分散式檔案系統大致上來說應該具有以下的特色：

1. 簡化系統使用與管理需求間的區隔動作。
2. 簡化使用者間檔案資料的共享操作。
3. 系統對使用者或應用程式都具有高度的透通性。

整體來看，分散式檔案系統採用客戶端與伺服器的架構，將檔案存放的位置予以透明化，使用快取緩衝區來加速客戶端對資料的重複讀取動作，提供與資料庫系統比較起來相對而言較弱的資料一致性模型，以及提供一個很接近本地端檔案系統的程式設計與使用者介面。

現有的分散式檔案系統隨著時間不斷地改進，不斷地提升系統成熟度與穩定度。然而，它的系統架構卻不見得能夠跟得上時代。當新的計算平台（例如行動裝置）想要採用現有的分散式檔案系統時，各式各樣的問題就會浮現出來。

因此，本計畫的目的，是藉由低通訊成本的網路傳輸機制，與高效能的資料同步機制，來建置一個高效能的Linux叢集檔案系統。在這個Linux叢集檔案系統建置完成之後，除了可以簡化使用者之間檔案分享的動作之外，還可以支援協同計算工作環境。

此外，也可以利用這個叢集檔案系統來架設伺服器叢集，例如：分散式資料庫伺服器可以使用叢集檔案系統的資料一致性語意來確保交易管理的正確性；全球資訊網伺服器可以利用叢集檔案系統來取代檔案系統鏡射（mirroring）或是反向代理伺服器（reverse proxy servers）的設置等等。

## 三 結果與討論

在經過了一個年度的執行後，本計畫大致上已經完成了相關參考文獻的研讀，也針對了一些現有檔案系統的優缺點進行分析與比較，並且提出了大略的系統架構設計。以下將主要結果條列之，並加以討論：

### （一）現存網路檔案系統及分散式檔案系統之比較

由昇陽公司所主導的NFS網路檔案系統，在UNIX及相容的作業系統間被廣泛地使用。對於一般使用者以及應用程式來說，NFS達到了一定程度的透通性要求。但是就管理的角度來看，客戶端主機的管理者必須詳細指明需要掛載的遠端伺服器主機名稱、以及其所提供的可掛載目錄名稱，方能順利完成掛載遠端NFS分享的操作。如果伺服器端變更了可掛載目錄的名稱，客戶端也要跟著更改，否則操作就會失敗。有些UNIX相容系統在使用NFS時會順便啓用自動掛載服務，例如：AutoFS等等。但是這還是不能解決管理者必須指定遠端NFS分享目錄名稱的問題。NFS也缺乏容錯的能力，也只適用UNIX及相容作業系統。

SMB/CIFS是微軟視窗作業系統上行之有年的網路檔案系統。它可以將主機的資源分享給網路上其他的SMB/CIFS客戶端主機。雖然在視窗作業系統上，GUI圖形化的操作十分方便，但是它有著跟NFS類似的問題：必須正確指定伺服器SMB/CIFS分享的名稱、缺乏容錯能力等等。此外，SMB/CIFS的效能不高也是一向爲人所詬病的一個缺點。Samba是一個在UNIX及相容作業系統上的SMB/CIFS通訊協定的實作，它有客戶端程式可以存取伺服器所分享的資源，也有伺服器程式可以將

本身的資源分享給其他SMB/CIFS客戶端。不過，由於UNIX與視窗作業系統對使用者認證與存取控制的機制不同，使得管理者在進行安全性方面的設定時常常不免感到困擾。

由美國卡內基美濃大學所研發的AFS與Coda分散式檔案系統，算是比較接近我們理想中的產品。AFS是第一個在客戶端設置持續性快取緩衝區的檔案系統，也就是說，快取緩衝區裡面的檔案與目錄資料不會隨著重開機而消失。AFS有統一的檔案命名空間，客戶端不需要知道檔案的實體存放位置，只需要知道檔案是屬於哪一個儲存區域，系統會自動查詢屬於該區域的管理伺服器來獲得檔案眞正存放的位置。AFS採用Kerberos來對客戶端進行認證，並輔以存取控制清單的機制來確保安全性。

目前產業界在網路儲存媒介方面的研發主要有兩個方向：網路連結儲存裝置（NAS）與高容量儲存網路（SAN）。NAS是透過網路檔案系統將儲存空間分享給客戶端程式；SAN則是透過光纖等高速連線，將一個以上的磁碟機「掛載」在本地端。作業系統存取這些SAN磁碟機的方式就跟使用直接裝置於主機內部的磁碟機類似，也是使用傳統檔案系統來存取。這兩種儲存媒介雖然有些也提供各種功能與特色，例如：容錯、硬體RAID、高傳輸量、高存取速度、GUI圖形化管理介面等等，但是要將它們實際運用於叢集運算上還是有好長的一段路要走。

## （二）相關著作及研究成果的研讀

叢集檔案系統是應用於叢集計算環境中的分散式檔案系統。叢集檔案系統所強調的主要是兩個特性：透通性以及容錯性。在透通性方面，叢集檔案系統有著離散性與多樣性這兩個主要的特徵。在容錯性方面，系統需要提供多個副本，以及系統日誌。雖然副本會引出資料一致性的問題，不過它除了滿足容錯的需求之外，還可以提供不同客戶端對不同副本的同時存取能力，所以對於叢集檔案系統來說，這是必須的。系統日誌可以提供錯誤回溯、以及整批大量寫入的功能，對於交易式爲主的叢集檔案系統來說，這更是一項不可或缺的功能。叢集檔案系統要能兼具透通性和容錯性，才能符合叢集計算環境單一系統映像以及高可利用性的要求。

一個檔案系統，不論是傳統檔案系統或者是叢集檔案系統，大致上皆可以分成四個部分：

1. 目錄服務：負責名稱解析、檔案的新增與刪除。
2. 認證服務：管理使用者能力清單與檔案存取控制列表。
3. 檔案服務：包括檔案的讀寫、屬性的取得與設置、同時控制、以及副本管理等等。
4. 系統服務：實體裝置、快取、與資料區塊的管理。

在整合式的叢集計算環境當中，叢集檔案系統的目錄服務即爲網路節點資源管理的一部份；檔案服務是與叢集系統的系統備援、容錯、錯誤回復等機制配合；而系統服務則需與底層的網路傳輸機制相結合。至於客戶端程式，則是利用系統所提供的API來進行檔案系統的各項操作。

叢集檔案系統的研究課題，主要有下列幾項：

1. 平行式輸出入處理（Parallel I/O）：將原本伺服器與客戶端一對一的傳輸，變成多對多的傳輸，以提高系統輸出入的效能。
2. 軟體RAID：利用資料交錯與資料分群的機制，將許多小磁碟機模擬成一個大的虛擬磁碟機，並且具有資料偵錯與修正的功能，達到高可利用性的目的。
3. 快取的一致性（Cache Coherence）：快取可以節省頻寬以及降低使用者等候時間。然而，不同副本間的資料同步與一致性問題，除了會影響快取所帶來的效能提升，還有可能會造成錯誤的運算結果。
4. 資料預取（Data Prefetching）：資料預取是根據使用者過去的行爲，先將稍後可能會用到資料傳回，以減少使用者等候時間。資料預取並不會節省頻寬。如果所使用的策略不當，反而會造成網路頻寬浪費以及快取污染。
5. 系統介面（Interface）：這裡指的是API，以及所提供給上層應用程式環境的系統概觀。叢集檔案系統系統必須要提供單一系統映像以及單一輸出入定址空間，如此使用者以及應用程式設計師才能感覺到叢集檔案系統所提供的便利與實用性。

## （三）訂定功能清單及詳盡系統規格

爲了要建構一個緊密耦合的叢集檔案系統，本計畫將著重於資料快取、資料預取、資料壓縮、檔案鏡射及錯誤回復等關鍵技術的研發。以下是系統的規格以及所具備的功能清單：

1. 以區塊（Block-based）爲最小存取單位。
2. 高速網路通訊協定的研發。

3. 平行式輸出入處理（Parallel I/O）。

4. 負載平衡（Load Balancing）。

5. 以角色爲基礎的存取控制（Role-based Access Control）。

6. 容錯（Fault-Tolerance）。

7. 持續性快取（Persistent Cache）。

8. 資料預取（Prefetching）。

9. 圖形化管理介面（GUI Administration）。

以下列出的幾項，是目前少有實作的一些額外功能。本計畫會評估其可行性，在不影響系統架構的完整性，以及系統效能的前提下，予以實作：

1. 資料壓縮（Compression）。

2. 資料加密（Encryption）。

3. 物件導向（Object-based）。

## 四　計畫成果自評

在上一個年度之中，我們進行了相關文獻的研讀工作、比較了一些現有網路檔案系統的特色與優缺點、也初步地完成了我們的叢集檔案系統架構。接下來，我們將根據所訂定的工作項目與進度，繼續推動本計畫的進行，完成這個叢集檔案系統的實作、測試、與效能評估。

## 五　參考文獻


[1] Coda File System. URL:<http://www.coda.cs.cmu.edu/>, Feb. 2002.

[2] SAMBA Web Pages. URL:<http://www.samba.org/>, Feb. 2002.

[3] P.J. Braam. "The Coda Distributed File System," Linux Journal, 50:46-51, Jun. 1998.

[4] R. Buyya. "High Performance Cluster Computing: Architectures and Systems," Prentice Hall PTR, New Jersey, 1999.

[5] J.J. Kistler and M. Satyanarayanan. "Transparent Disconnected Operation for Fault-Tolerance," 4th ACM SIGOPS European Workshop, Bologna, Italy, Sep. 1990.

[6] J.J. Kistler and M. Satyanarayanan. "Disconnected Operation in the Coda File System," ACM Transactions on Computer Systems, 10(1):3-25, Feb. 1992.

[7] W. Liu and et al. "An Effective File Migration Algorithm in Cluster File Systems," In Proceedings of International Workshops on Parallel Processing, 1:329-335, 2000.

[8] D. McDonell and et al. "Security for Distributed Object-Oriented Systems," In Proceedings of DARPA Information Survivability Conference & Exposition II (DISCEX '01), 1:264-278, 2001.

[9] E. Miller and et al. "Strong Security for Distributed File Systems," IEEE International Conference on Performance, Computing, and Communications, 1:34-40, 2001.

[10] M.J. Moyer and M. Abamad. "Generalized Role-Based Access Control," 21st International Conference on Distributed Computing Systems, 1: 391-398, 2001.

[11] L.B. Mummert and M. Satyanarayanan. "Variable Granularity Cache Coherence," Operating Systems Review, 28(1):55-60, Jan. 1994.

[12] M. Satyanarayanan. "Integrating Security in a Large Distributed System," ACM Transactions on Computer Systems, 7(3):247-280, Aug. 1989.

[13] M. Satyanarayanan and et al. "Coda: A Highly Available File System for a Distributed Workstation Environment," IEEE Transactions on Computers, 39(4):447-459, Apr. 1990.

[14] M. Satyanarayanan. "Scalable, Secure, and Highly Available Distributed File Access," IEEE Computer, 23(5):9-21, May 1990.

[15] M. Satyanarayanan. "The Influence of Scale on Distributed File System Design," IEEE Transactions on Software Engineering, 18(1):1-8, Jan. 1992.

[16] M. Spasojevic and M. Satyanarayanan. "An Empirical Study of a Wide-Area Distributed File System," ACM Transactions on Computer Systems, 14(2):200-222, May 1996.