行政院及所屬各機關出國報告

（出國類別：實習 ）

赴美國實習
『實習分散式網際網路服務及新應用之開發』報告

服務機關 ： 中華電信股份有限公司

數據通信分公司

出國人 ：職 稱 姓 名

助理工程師 邱國龍

出國地點 ： 美國舊金山

出國期間 ： 91年12月1日 至 90年12月14日

報告日期 ： 92年1月

系統識別號:C09105397

公 務 出 國 報 告 提 要

頁數: 40　含附件: 否

報告名稱:

實習分散式網際網路服務及新應用之開發

主辦機關:

中華電信數據通信分公司

聯絡人／電話:

／

出國人員:

邱國龍　中華電信數據通信分公司　網際網路處　助理工程師

出國類別: 實習

出國地區: 美國

出國期間: 民國 91 年 12 月 01 日 -民國 91 年 12 月 14 日

報告日期: 民國 92 年 02 月 25 日

分類號/目: H6／電信　H6／電信

關鍵詞: 效能,調校

內容摘要: 如何將現行的Sun伺服器之效能加強，又不會影響運作中的系統，同時僅需花費些許的費用，就可應用系統調校的技術。

本文電子檔已上傳至出國報告資訊網

目次：

壹、實習之目的

為因應客戶端網路設備的寬頻化所帶來的服務系統資源不足的問題，而此現象又無法以簡單的系統擴充就能得到完善的解決前提下，我們必須要在提供客戶相同的服務與可彈性擴充系統、服務功能的情形，重新對現有系統與未來系統做一審慎評估。在評估之前，如何讓現有系統能運作得更順利、未來系統也可充分發揮其效益是本報告將要闡述的重點----系統效能調教與組態之更動。

本次實習的課程為『Solaris system performance management』，為期五天。

貳、實習行程及課程

職奉派至美國舊金山實習『實習分散式網際網路服務及新應用之開發』，實習時間自民國九十一年十二月一日至九十一年十二月十四日為期十四天。本次實習課程計有：

IEEE802.11技術與產品發表 （5天）

Solaris system performance management課程研習（5天）

參、Solaris系統效能管理

系統效能調教的目的是希望能讓系統資源利用度達到最佳化、增進系統效能、協助系統管理者辨認系統真正需要升級之所在。我們將在以下章節中，將調校的概念與技術，依CPU、記憶體、I/O與匯流排乃至檔案系統的順序，搭配其運作概念，並配合各種系統分析與調校工具，逐項詳細說明。

3.1 效能管理基礎與工具使用說明

系統調校前，有幾個觀念需要知道：

1. 取捨問題：系統效能調校會讓子系統間的關聯性浮現出來，這時候我們就必須要有所取捨，才能讓調校的目標達成。關聯性有兩種：一為CPU、記憶體、I/O，另一為可靠度、效能、成本。例如，為增加記憶體使用空間，我們將換頁的臨界點降低，但這將導致換頁的次數變頻繁了。或者你為了增進系統效能，擴充了記憶體或CPU，雖能提昇效能，但也會導致整體的系統成本增加。

2. 效能概念：圖3.1為效能概念模型，其包含了系統與一些不同的外部變數，從這個模型中可清楚知道系統的效能將會被各種因素所影響。

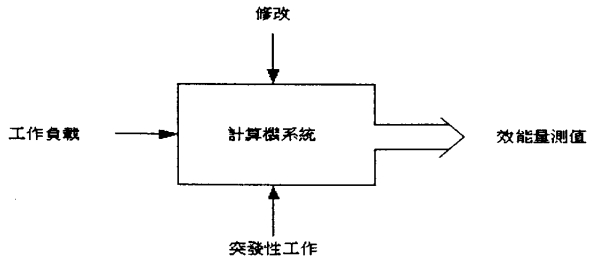

圖3.1效能概念模型

工作負載：指的是使用任何系統資源所產生的負載，亦是影響系統效能的原因。包含有CPU的使用、記憶體的消耗、網路使用、尖峰時間的使用與經常性的資料庫存取。

突發性工作：如資料備份、執行效能監測工作等導致系統不正常運作等事情。

修改：更改硬、軟體資源所導致的效能影響，如加修正程式等。

3. 基本調校程序：

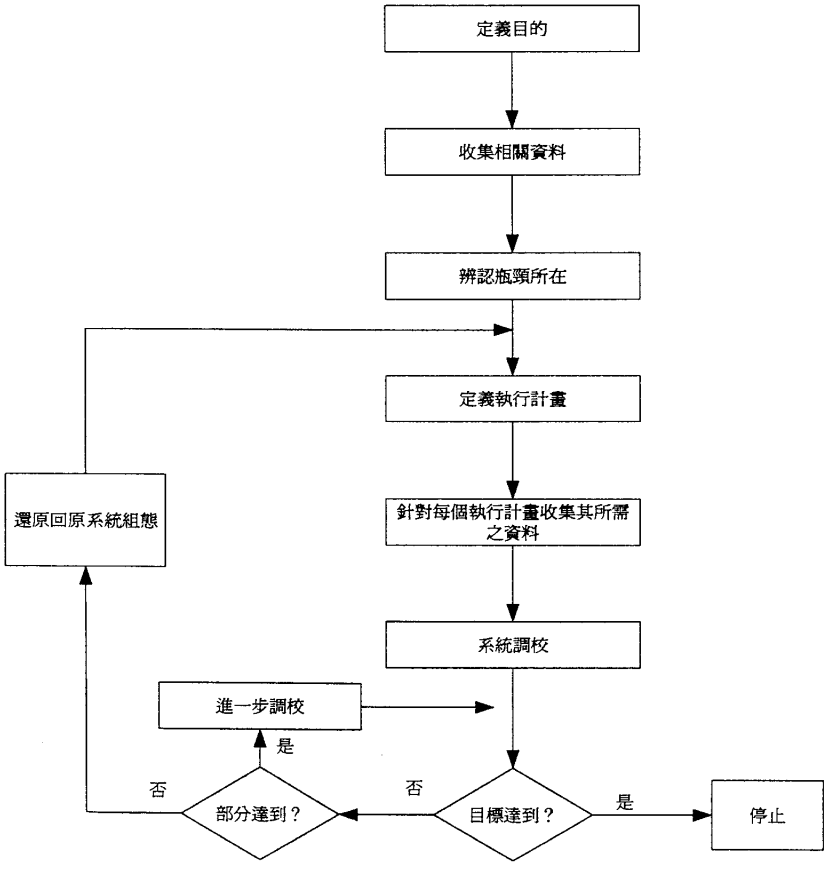

圖3.2基本調校程序

由圖3.2所示，整個程序是由定義目的與收集、辨認瓶頸所需之資料開始，同時將所得之量測值作為調校之參考。調整後的系統效能在與原效能做比較以找出真正的瓶頸問題所在，若順利找出問題，那麼就一步一步慢慢修改系統直至達成我們既定的效能目標。倘若一直無法改進，系統則需要進一步整體升級。

有一個重要的觀念：調校是一個持續性的工作；意即只調整系統效能問題之一的參數，並不能有效解決效能問題，而是要持續性地調整所有子系統。通常，為移除系統的瓶頸問題都會導致另一種瓶頸問題的發生。

工具說明

下面所列的指令是Solaris所提供之系統效能監測工具

- 虛擬記憶體運作之統計：vmstat、memtool工具(含pmem,memps,memtool,mem)、SE工具
- 與I/O有關之統計：iostat、busstat
- 處理器的運作統計：mpstat、cpustat
- 與網路相關的統計：netstat
- 與網路檔案系統相關的統計：nfsstat
- 行程運作之統計：/usr/proc/bin目錄下的工具
- 綜觀系統運作之效能：sar、Sun Management Center(Sun MC) software

另有系統參數檢查之指令，

- sysdef：可列出目前系統參數之定義值，所列出的可調參數是最常調整的。
- mdb：可在系統運作的時候作參數調整
- ndd：調整TCP/IP驅動程式的參數值

當系統需要調校的時候，可執行指令或修改系統檔案：

- /etc/system ：提供靜態的系統調校參數之儲存位置，修改數值後需要重新開機始生效。
- mdb：修改運作中的系統參數。
- ndd：修改TCP/IP相關的驅動程式之系統參數。

3.2 Processes and threads與 CPU scheduling之監控

一個行程的運作情形與系統的組態是息息相關的，如何監視一個行程的執行過程，又能夠明確發現問題之所在是本節的重點。同時我們也將說明CPU的運作與一些系統參數設定之關係，諸如排程等。

行程調校參數

在這裡，我們不再贅述行程的相關定義，我們將針對有關行程可調校的部分做一說明。下表是與行程相關的調校參數，所有參數的設定與maxusers及max_nprocs息息相關，而且均可設定在/etc/system檔案中。

參數	預設值	最小值	最大值
maxusers	使用者記憶空間之百萬位元數或小於2048	8	2048(4096)
max_nprocs	16×maxusers+10	266	maxpid
maxuprc	max_nprocs-5	1	29995
pidmax	30000	266	999999

表3.1 與行程相關之調校參數

因max_nprocs參數會因pidmax改變而改變，更顯得pidmax的重要性，所以我們也為此說明了maxpid相關的演算法，

```
if   pidmax   is  >   maxpid(30000)
then      maxpid   =   pidmax
else if   pidmax   is  <   reserved_procs(5)
     then      maxpid   =   MAX_MAXPID(999,999)
     else if   reserved_procs(5)  <   pidmax   <   maxpid
          then      maxpid   =   pidmax
```

其中，reserved_procs是保留給root所執行的行程使用，pidmax為最大的PID(行程編號)，maxpid之值則是開機時由上列的演算法求得的。

行程狀態

行程有開始也有結束，如圖3.3所示。每個方塊中所代表的均為

一特定狀態，如SIDL、SRUN等。其中zombie代表此行程是由父行程喚起，但當其結束時卻無法回覆執行結果所殘留的狀態， 在Solaris9作業系統中，可用/usr/proc/bin中的preap指令清除。

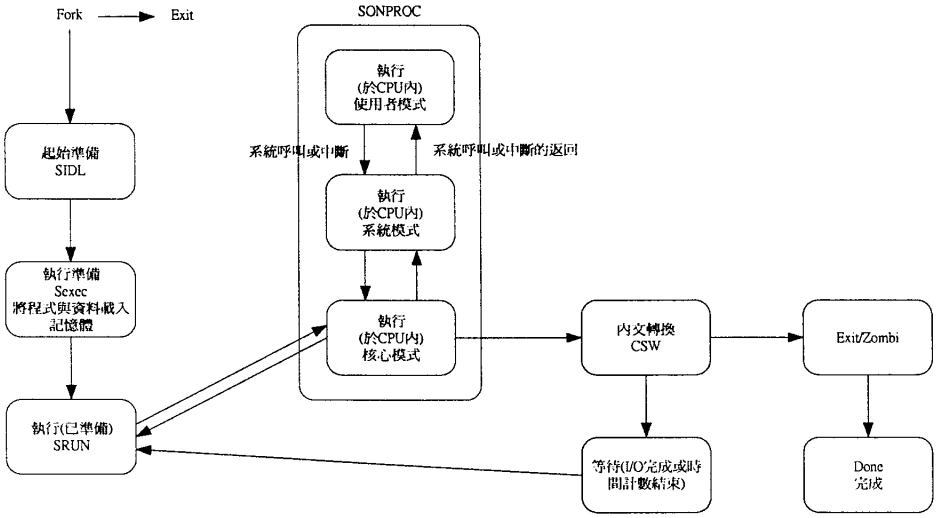

圖3.3行程狀態圖

行程監視工具

行程監視工具可用來觀察行程執行的過程、使用系統資源的多寡等，工具如下面所列

觀察閉鎖狀態：不良的閉鎖狀態會導致死結(deadlock)、競賽(race condition)的發生，甚至會大大的影響系統的效能，我們可用lockstat指令來得知此閉鎖運作的情形。

```
Lockstat -H -D 10 sleep 1
Adaptive mtex hold: 4468 events in 1.009 seconds (4429 events/sec)
Count    indv    cuml    rcnt     nsec Lock    Caller
-------------------------------------------------------
124  3%  3%  1.00     525 0x300001a9000    timeout_common+0x114
122  3%  6%  1.00     277 0x300001a9000    callout_execute+0x84
```

```
101  2%  8%  1.00     429 0x300001a9000    softint+0x4c
100  2%  12% 1.00     356 thread_free_lock     clock_0x1ec
186  2%  14% 1.00     269 0x300001a6000    callout_schedule+0x50
```

查看時鐘常式：使用vmstat –i可觀察時鐘常式的中斷頻率

```
#vmstat  -i
interrupt        total   rate
--------------------------------
clock    27622896     100
hmec0    780010       2
qfec0         0       0
qfec0         0       0
qfec0         0       0
qfec0         0       0
fdc0          0       0
--------------------------------
total         28403013     102
```

愈快的rate愈適合即時(RT)的作業系統

基礎監視工具：/usr/bin/ps與/usr/ucb/ps，這兩種監視工具有其不同之處，不僅參數不同，連輸出結果亦不同。

CPU排程

系統時程安排目的在分散CPU工作負載，時程安排的好壞影響CPU工作效能甚大，因此我們要針對系統應用的性質來選擇適合的排程類別，讓CPU工作的效率最高。

排程的類別有五種，分述如下：

分時(TS)：行程的優先權會因CPU的使用之多寡動態被調整，擁有相同優先權的行程會一起切割CPU時間，而使用

CPU較多的工作被賦優先權則較I/O較多的工作為低。

互動式(IA)：以視窗為主的工作通常會被賦予此類別，期能達到較佳的效能。如OpenWindows 或CDE。

系統(SYS)：使用在系統的執行緒上，如paging、fsflush、sched等行程才能設為此類別。

固定優先權(FX)：顧名思義，即行程的優先權並不會有任何的更動。

即時(RT)：一個RT的執行緒在系統運作中有著最高的優先權，並會一直在CPU中執行直到執行結束。

我們可端視所應用之系統的需求選擇適合的類別。

如何顯示並更動排程的參數？可執行指令dispadmin

dispadmin -c 類別 類別補充

例：

```
#dispadmin -c TS -g
#Time Sharing Dispatcher Configuration
RES=1000
#ts_quantum ts_tgexp     ts_slpret ts_maxwait     ts_lwait PRIORTY    LEVEL
200             0             50        0             50        #     0
200             0             50        0             50        #     0
.....................
```

如何更改排程中行程的行為呢？可用priocntl指令變更。Priocntl可做到下列三點

1. 列出目前的排程類別
2. 顯示行程的排程參數
3. 產生一行程

例1：

```
#priocntl -l
CONFIGURED CLASSES
    ================
SYS (System Class)
TS (Time Sharing)
        Configured TS User Priority Range: -60 through 60
IA (Interactive)
        Configured IA User Priority Range: -60 through 60
```

例2：

```
#priocntl -d -I pid 1
TIME SHARING PROCESSES
PID     TSUPRILIM       TSUPPRI
1           0               0
```

例3：

```
#priocntl -e -c TS -p 20 find / -name core -print
```

CPU的控制與監視

在Solaris的開放環境中，用以監控CPU的指令有

mpstat：顯示各個CPU的運作統計

psradm：可啟動(-n)或停止(-f)CPU運作

psrinfo：顯示處理器運作與否，若有啟動則回1否則為0

prtconf：顯示裝置的組態，可得知許多細項資料，如主機板、處理器、記憶體等

prtdiag： 列出系統的硬體組態與測試情形

psrset：設定處理器群組

sysdef：顯示系統組態

3.3 記憶體的監視與調校

記憶體的功用是加快CPU資料的存取、減低重複資料讀取動作以期加快CPU的運作效率。從種類可分快取(cache)與實體記憶體，前者一般是置於CPU模組或CPU內部中，目前又可分L1與L2快取；後者裝置為傳統之記憶體。這兩種不同的記憶體其效能與成本相差非常大，由於快取位於CPU的核心範圍內，製作難度高，因此其成本高但效能最好；傳統記憶體則相反。

表3.2中說明L1與L2快取記憶體的不同處

快取名稱	大小	位置
L1或內部快取	4kbytes-40kbytes	於微處理器內
L2或外部快取或Ecache	0.5Mbytes-8Mbytes	與微處理器分屬兩個獨立的晶片

表3.2 L1與L2快取記憶體之特性

圖3.4為一SUN主機之快取架構圖，其中L1快取內採用哈佛快取架構，該架構是為了提昇系統效能而將指令快取(I$)與資料快取(D$)分開，又因其具有雙匯流排架構，故可同時接受雙重的指令，達到雙倍的速度。

在L2快取中資料未發現者定義為次要錯誤(minor fault)，在memory中未被發現者則為主要錯誤(major fault)。這兩個數值可以從mpstat的指令中觀察初步運作的統計結果，所代表的欄位分別是

minf、mjf。

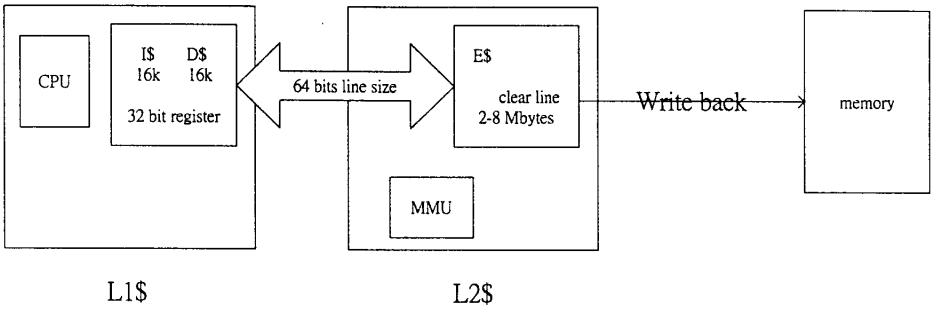

圖3.4 快取架構圖

上圖中，對於更動過的資料所採的更新方式為write-back，即此資料僅需在記憶體不足的情形下再寫回下一階快取即可，此種方式相當適合大量資料處理。另一種為write-through，更動過的資料是隨即被寫回下一階，這種方式適合頻寬不足與資料需要同步的情形使用。

快取的運作方式：

當你對系統發出資料請求時，快取控制器會使用一種標籤用來搜尋所需的資料，這標籤可以是資料儲存的位置或可用以辨識資料之特性。所謂的快取命中，表示資料是存在於快取中；而未命中時，快取管理器會將需求傳遞給下一階快取，且此管理器必須等到真正的資料得到之後才算完成此動作，也才會執行下一指令。

究竟命中率的多寡影響系統效能有多少？我們可從圖3.5清楚看出，命中率86%與100%其系統效能相差了約90%，因此為提昇系統效能，我們必須盡量提高快取命中率。

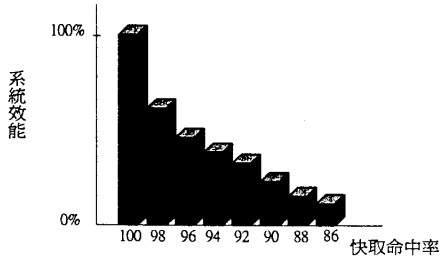

圖3.5 快取命中率與系統效能

例如，在網路檔案系統(NFS)中，當使用者數增加時，其效能就會相對地降低，我們可以利用vmstat –s檢查目錄名稱查尋快取(Domain name lookup cache)，發現命中率即使有96%，卻僅有約50%的系統效能。

位址快取亦分為兩種：一為虛擬位址快取，此快取置於MMU(記憶體管理單元)之前，負責儲存經由MMU轉換後的虛擬位址，優點為執行效率較佳，但缺點是當程式在易換後需全部重新清除內容。另一種為實體位址快取，此快取位於MMU之前，負責儲存實體位址，優缺點與前述相反。

記憶體監視

透過記憶體監視工具可以知悉如下的資料：

實體記憶體大小(PM)

檔案緩衝記憶體大小(FC)

核心記憶體大小(KM)

應用程式記憶體大小(AM)

可用的記憶體大小(FM)

PM = FC + KM + AM + FM

其中，實體記憶體的大小是固定的，末項之方程式為各組成的相關式。我們可以利用Memtool工具中的prtmem指令輕易地得知系統目前的記憶體使用情形。舉例如下

```
#/opt/RMCmem/bin/prtmem
Total memory:          982 Megabytes
Kernel memory:         118 Megabytes
Application:           106 Megabytes
Executable & libs:     24 Megabytes
```

File Cache:	95 Megabytes
Free, file cache:	534 Megabytes
Free, free	102 Megabytes

檔案緩衝記憶體用以儲存最常被讀取或寫入的檔案，以減少磁碟存取；核心記憶體則負責存放核心碼，而其大小會隨者驅動程式或核心模組的更動而變化，而且核心記憶體是不會被易換出去，您亦可使用sar –k l 的指令觀察核心記憶體的配置情形。

除上述的prtmem指令可以查之記憶體的使用分配情形外，vmstat亦可顯示部分的記憶體使用情形或換頁之統計報表。

行程位址空間

圖3.6中，顯示一行程被執行時系統配置虛擬記憶體的情形

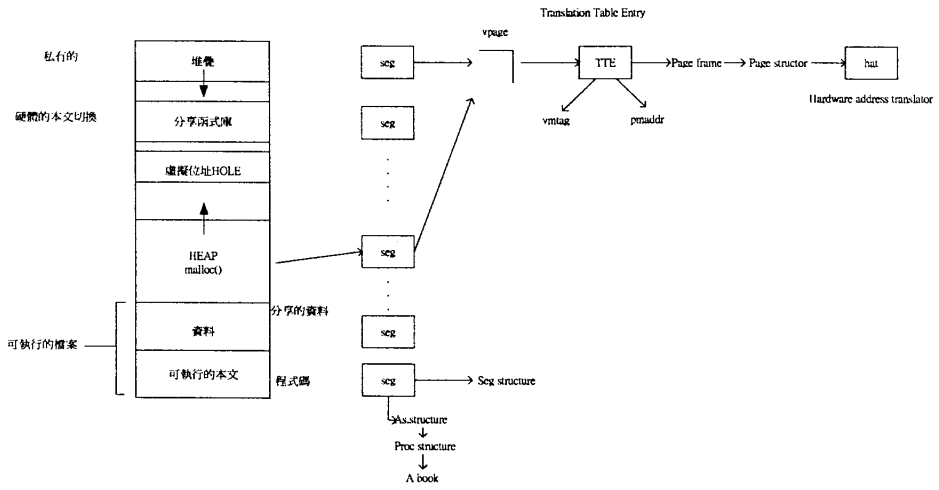

圖3.6行程被執行時系統配置虛擬記憶體的情形

說明：堆疊是用來給存放行程執行特殊常式時所存放資料，所佔空間則端視常式之需求，執行緒則可相互共享堆疊內的資料；共享程式庫區則存放行程間共用的常式，此區塊為獨立的、與位址不相關的；HEAP則是行程執行時所另外要求配置的記憶

區塊；資料區塊存放執行碼所需要的全域資料、常數與靜態變數之資料；可執行的本文則存放可執行的指令，從執行檔中獲取。

換頁掃描行程

當系統使用虛擬記憶體運作時，主記憶體就變成磁碟的快取，因此也就要執行移進移出的動作，也就是換頁(Paging)。換頁是為確保記憶體空間尚在安全的範圍內。為達到此目的，對於頁面的有效性就必須依賴換頁掃描行程來確保，但由於有時記憶體的使用又超乎常理，以致於記憶體不足，因此就有一置換程式將最少使用的記憶體頁面移至磁碟中，此確保有效的記憶體空間。

以下我們將針對換頁掃描行程做初步說明，以建立正確概念，才能配合記憶體之調校策略。圖3.7為換頁掃描頻率與有效的記憶體頁面之數量間的關係

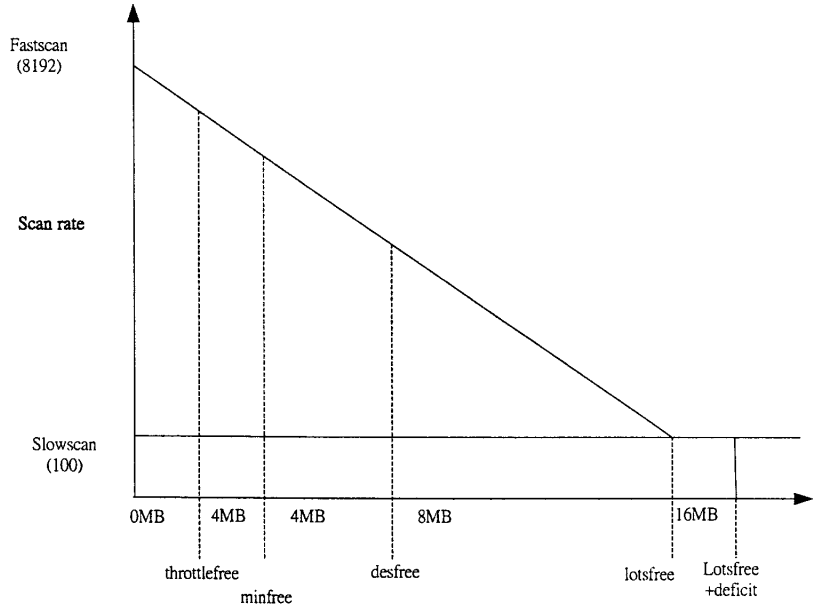

圖3.7換頁掃描頻率與有效的記憶體頁面之數量間的關係

lotsfree：換頁行程的掃描器始點，當可用的頁面少於此數值

時，則該行程開始掃描；此值一般為主記憶體的64分之一。

slowscan：換頁行程開始掃描時的掃描頻率；一般為記憶體頁面數的百分之一。

desfree：當可用的記憶體空間低於此值時，置換(swapping)行程會開始動作以騰出更多的可用空間；預設值為lotsfree的二分之一。

fastscan：當可用的記憶空間低於minfree時，換頁行程會將掃描速度加快成此數值；此數值與系統架構相關。

minfree：同上；預設值為desfree的二分之一。

deficit：提高行程的記憶體頁面之可用數，以防止應用程式執行時產生的暫時性記憶體不足現象。

磁碟檔案快取

在Solaris開放的環境裡，檔案系統的快取是屬於虛擬記憶體之一部份，因此它可以使用記憶體中的可用頁面。圖3.8為檔案系統快取架構圖

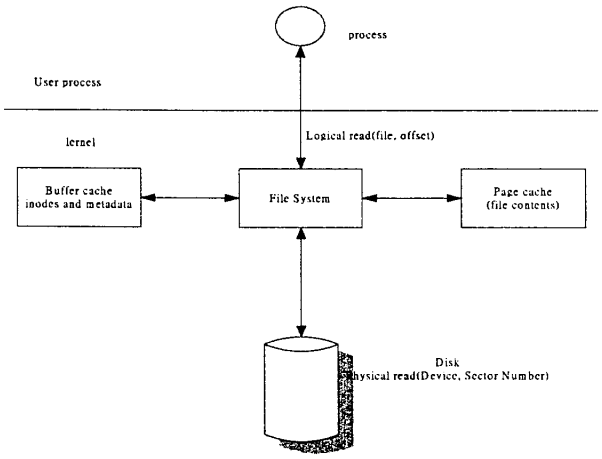

圖3.8檔案系統快取架構圖

頁面快取(page cache)負責暫存檔案的內容以供程式取用，同

時亦提供檔案在磁碟中的位移位置，就不必至磁碟中搜尋。緩衝快取(buffer cache)則負責儲存inode資料與檔案的中間資料(metadata)，其快取的大小會隨檔案內容的多寡而變動，上限由核心參數bufhwm所限制。一個inode的大小為300bytes，檔案的中間資料則每2Gbytes暫存1Mbytes。舉例說明，如果一個資料庫有100個檔案，全部的檔案大小為100Gbytes，但同時存取的量為50Gbytes，則其所需要的buffer cache為100x300bytes＋[50/2]x 1Mbytes=25030kbytes。你可以使用sar –b指令去觀察此快取的命中率。

以下所列的為觀察記憶體運作情形的指令

觀察可用的記憶體數量：vmstat、sar –r

觀察swap運作情形：sar –w

觀察行程使用記憶體的情形：pmap

觀察系統換頁情形：sar –p

3.4 匯流排及I/O的管理與調校

本節說明的部分較偏向硬體的設定及其使用之規劃，由於在匯流排與I/O的規劃之前必須對系統與I/O設備的規格要有相當了解，因此，為使系統的運作效能提高，首先要做的就是先明瞭所有的硬體規格與運作特性。

SUN伺服器產品所使用的匯流排種類與規格有：

Gigaplane-XB Bus:up to 12.8Bytes/sec

Sbus:20-25MHz, 32bit or 64bit

PCI Bus:33MHz or 66MHz and 32bit or 64bit

IDE Bus: for disk or CDROM

Sun Fireplane Bus:maximum of 9.6Gbytes/sec and maximum of 18 address bus.

我們可以用prtdiag或prtconf指令檢查硬體的組態，包含I/O介面卡的資訊。若要觀察匯流排運作情形，可用busstat指令，其參數及其意義請參閱表3.2

參數	意義
-a	以絕對計數值表示
-e	顯示PIC的值
-h	顯示有用的資訊
-l	列出支援效能計數器的裝置
-n	不要顯示標頭
-r<deivce><instance>	顯示<device>與<instance>之PIC值
-w <device> <instance> [,pic0=event] [,picn=event]	利用指定的程式來計數指定的事件

表3.2 busstat指令之參數及其意義

外掛裝置的匯流排有下列幾種：

SCSI Bus：需注意匯流排速度與寬度，請參閱表3.3與表3.4

名稱	最大速度
Asynchronous	4Mbytes/sec
Synchronous	5 Mbytes/sec
Fast	10 Mbytes/sec

Sun Enterprise Ultra 20	20 Mbytes/sec
Sun Enterprise Ultra 40	40 Mbytes/sec
Sun Enterprise Ultra 80	80 Mbytes/sec

表3.3 SCSI Bus之規格(1)

型式	寬度	最大傳輸率	電纜型式
Narrow	8-bit	5 Mbytes/sec	50conductor
Wide	16-bit	10 Mbytes/sec	68conductor

表3.4 SCSI Bus之規格(2)

Fibre channel：上面相容的協定有SCSI、IP、ATM、IEEE802.2、TCP，其標準速率為1.06Gbytes，最長的距離為10公里。

如何做I/O效能計劃

任何的I/O運作特性都會衝擊到匯流排的流量，一個好的檔案系統規劃需將資料分散並同時平衡磁碟與控制器與提昇效能，所以你必須依系統的負載特性規劃磁碟組態；另一方面，依資料操作的特性又可分循序讀取與隨機讀取，不同的讀取方式需要不同的磁碟快取規劃。

如何增進I/O運作效能

要增進I/O運作效能，簡單的做法可依下列三點為之：

1. 提昇匯流排頻寬，不僅加快資料傳送時間，亦可增加IOPS。

2. 提昇記憶體容量，可增加系統處理資料效能。

3. 提昇快取容量，使系統可同時傳送較多的請求給裝置。

I/O副系統調教

副系統調校主要在減少I/O運作的數量，以提昇整體的系統流量。所以如何降低I/O運作的數量? 方式如下：

1. 盡可能一次請求較大的區塊資料。

2. 正確的調校主記憶體。

3. 適當地調整檔案快取大小。

4. 使用寫入刪除，降低I/O數。

3.5檔案系統的管理與調校

決定何種型態檔案系統的主要因素之一是檔案系統效能，而評估效能時，有兩個參數需列入考慮：

1. 資料存取速度

2. 復原機制

執行何種應用程式類別將會影響檔案系統的運作效能，若依應用程式的運用資料特性可分為資料導向型或型態導向型；資料導向型的應用程式多半不需要太多建立或刪除檔案的需求，主要僅是管理大量資料的需求，如資料庫等。若以型態導向類的應用程式則需要管理大量的小檔案，如電子郵件系統、人員profiles等。

若依應用程式存取型態而言，可分為循序-可同時下多個I/O請求以提昇校能，與隨機-每個I/O請求均是相關性很低的小資料區塊。

調校的方法

檔案系統效能的增進會使整體系統效能有明顯的改變，下面將說明各種調校的觀念與方法：

1. 資料預讀取：此方式可預測目前所讀取的區塊與該區塊相鄰的區塊是否要先讀取以增進效能。預讀取的區塊的大小可由系統參數maxcontig調整，其預設值為該裝置的最大傳輸量，在設定前先查詢清楚以利最佳化。

2. 設定SCSI傳輸大小：系統的SCSI驅動程式中最大的SCSI傳輸大小為128Kbytes，大於此數量的資料區塊將被切割，若需要調整的切割大小，可設定修改系統參數於/etc/system如下

   set maxphys=1048576

3. 設定叢集大小：newfs –C 16，每個叢大小為8Kbytes，則叢集大小為128Kbytes，越大的叢集大小對循序讀取性的資料操作更有利。

4. 修改叢集大小：tunefs –a 8 /dev/rdsk/cXtXdXsX

5. 檔案系統寫入之控制：作業系統一般將寫入的請求先予以儲存並回應給應用程式，以利應用程式的繼續執行，或稱之為延遲式的非同步寫入或寫入延遲。UFS中的參數maxcontig定義多少的寫入資料被群組後才會再真正寫入磁碟中。同時你亦可啟動寫入節流機制，該機制可控制每個檔案等待被寫入的384Kbytes資料暫存限制的總和限制值。可分為高標與低標，一般之高標為記憶體的1/64而低標為1/128，可將其設定在/etc/system中

   set ufs_WRITES=1 (enable write-throttle mechanism)
   set ufs_fs_HW=167777216
   set ufs:ufs_LW=8388608

肆、實例研究

上述之系統效能調校的概念與技術，若能以一實例逐一說明，相信更容易融會貫通且一以貫之。本章節將對一特定的實例，說明如何判斷並調整現行運作系統。

這個例子是一位客戶請求Sun Microsystem協助其找出系統的問題與瓶頸，並建議改善的方法。他是在Sun的設備上執行SAS(Statistical analysis software)應用程式。

在開始效能調校之前，我們必須先收集以下資訊：

1. 系統組態
2. 系統上執行的應用程式
3. 使用者社群：如使用者的數量、位置及其工作的型態等等

在調校之初，對於相關的系統運作資料需收集完整，而且每一次只能調整一種，以便觀察其效能的變化，不可一次調校多種參數。圖4.1為系統組態

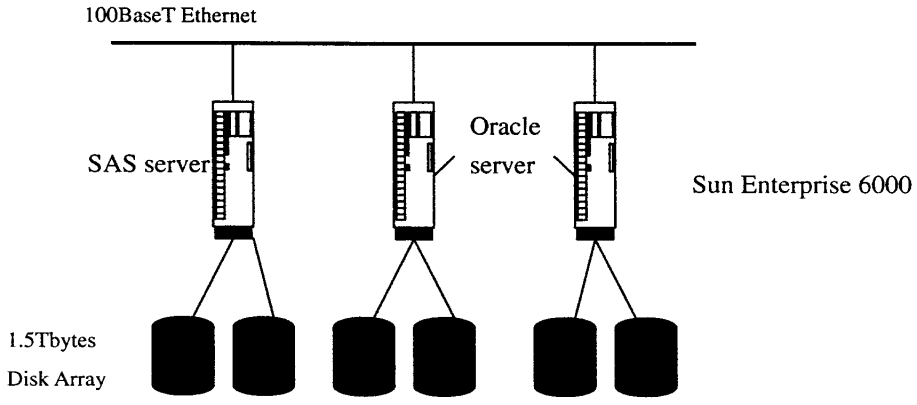

圖4.1系統組態

整個系統包含三部Sun Enterprise 6000伺服器，其中兩部執行Oracle 關聯式資料庫管理系統、另一部執行SAS應用程式。每一部伺服器均連接至兩組不同的儲存陣列，每個儲存陣列具有1.5Tbytes的影射(mirrored)儲存容量。

目前，在SAS伺服器中已建立200個使用者，每個使用者透過100BaseT之Ethernet，經由Switch連接至伺服器所屬的LAN上，來使用伺服器所提供的服務。這200個使用者在系統中均有排定時程，用以查詢Oracle database中資料、進行資料分析並輸出結果報告。系統的排程是由5:30 a.m.開始，夜間並不執行任何工作。但是，當使用者在50人時，系統運作還正常，人數若逐漸遞增時，效能就開始降低。在這情形下，Oralce database的查詢速度仍舊正常，SAS伺服器的效能就會因使用者的遞增而有明顯的下降，執行時間也越來越長。

系統組態：

利用prtdiag工具，可獲知系統的硬體完整資訊

```
# /usr/platform/sun4u/sbin/prtdiag

System Configuration: Sun Microsystems sun4u 16-slot Sun Enterprise 6000
System clock frequency: 84MHz
Memory size: 6144Mb
```

CPUx

Brd	CPU	Module	Run MHz	Ecache MB	CPU Impl.	CPU Mask
0	0	0	336	4.0	US-II	2.0
0	1	1	336	4.0	US-II	2.0
2	4	0	336	4.0	US-II	2.0
2	5	1	336	4.0	US-II	2.0
4	8	0	336	4.0	US-II	2.0
4	9	1	336	4.0	US-II	2.0
6	12	0	336	4.0	US-II	2.0
6	13	1	336	4.0	US-II	2.0

• • •
• • •

Memory

Brd	Bank	MB	Status	.Condition	Speed	Intrlv. Factor	Intrlv. With
0	0	1024	Active	OK	60ns	4-way	A
0	1	1024	Active	OK	60ns	2-way	B
2	0	1024	Active	OK	60ns	4-way	A
2	1	1024	Active	OK	60ns	2-way	B
4	0	1024	Active	OK	60ns	4-way	A
6	0	1024	Active	OK	60ns	4-way	A

• • •
• • •

IO Cards

Brd	Bus Type	Freq MHz	Slot	Name	Model
1	SBus	25	0	nf	SUNW,….
1	SBus	25	1	fca	FC
1	SBus	25	2	fca	FC
1	SBus	25	3	SUNW,hme	
1	SBus	25	3	SUNW,fas/sd (block)	
1	SBus	25	13	SUNW,socal/sf (scsi-3)	501-3060
3	Sbus	25	0	QLGC,isp/sd (block)	QLGC,ISP1000
3	Sbus	25	1	fca	FC
3	Sbus	25	2	fca	FC
3	SBus	25	3	SUNW,hme	
3	SBus	25	3	SUNW,fas/sd (block)	
3	SBus	25	13	SUNW,socal/sf (scsi-3)	501-3060
5	Sbus	25	0	QLGC,isp/sd (block)	QLGC,ISP1000
5	SBus	25	3	SUNW,hme	
5	SBus	25	3	SUNW,fas/sd (block)	
5	SBus	25	13	SUNW,socal/sf (scsi-3)	501-3060
7	Sbus	25	0	fca	FC
7	Sbus	25	2	QLGC,isp/sd (block)	QLGC,ISP1000
7	SBus	25	3	SUNW,hme	
7	SBus	25	3	SUNW,fas/sd (block)	
7	SBus	25	13	SUNW,socal/sf (scsi-3)	501-3060

No failures found in System

```
No System Faults found
==========
#
```

大部分嚴重的瓶頸問題都與I/O有關，可能會發生的狀況如”系統匯流排超載”、”週邊裝置匯流排超載”或”硬碟超載”等等。從以上的資料分析，對於E6000主機來說，每一塊主機板有兩個Sbus，共有200Mbyes/sec；E6000的系統匯流排頻寬為2.6Gbytes/sec；而四塊主機板之頻寬共為800Mbytes/sec，並未超過系統容許值。

在主機板1上連接許多的週邊裝置，可能會發生超載

- nf-FDDI at 100Mbps(or 12.5Mbytes/sec)的現象
- fca-Fibre Channel controller at 40 Mbytes/sec
- fca-Fibre Channel controller at 40 Mbytes/sec
- SUNW,hme-Fast/Wide SCSU at 20 Mbytes/sec
- SUNW,fas/sd-Fast/Wide SCSI at 20 Mbytes/sec
- SUNW,socal/sf-Serial Optical Channel at 100 Mbytes/sec

總和所有的頻寬為225Mbytes/sec，已超過Sbus的最大頻寬。因此我們將一片fca移至主機板5，這樣使得其頻寬為192.5Mbytes/sec，而主機板1為185Mbytes/sec。

磁碟的瓶頸：

一個I/O的運作必須等到I/O的請求被送至裝置後才會開始，而此請求必須等到起始器可存取匯流排後才能被傳送。假如匯流排忙碌時，起始器必須等待；這段期間，I/O請求會被儲存在驅動程式佇列中，任何一個讀取或寫入的指令會先被存入驅動程式的等待佇列中直

到SCSI bus與disk都準備就緒才會被執行。

等待的I/O請求須等到其前面的請求被執行後才會執行。因此，過大的等待的I/O請求數量，會易使裝置或匯流排產生超載的現象，而我們可以使用以下的技術解決此問題：採用標籤佇列、換較快的匯流排、將速度慢的裝置移至其他匯流排。

我們可以使用iostat指令來觀察整體I/O狀態

```
#iostat -x
                                                    extended device statistics
device    r/s  w/s   kr/s   kw/s   wait  aciv  svc_t  %w  %b
sd0       9.4  2.1   79.2   57.0   0.4   0.5   75.0   1   9
sd1       2.5  3.1   62.3   140.4  0.0   0.2   34.0   0   5
sd2       3.1  3.8   121.4  166.9  0.1   1.6   246.8  1   21
sd3       1.4  0.2   71.0   8.4    0.0   0.0   25.6   0   3
sd4       3.1  3.6   120.0  164.0  0.0   1.0   156.2  1   18
sd5       1.4  0.2   69.9   8.4    0.0   0.0   25.6   0   3
sd6       3.0  3.7   118.0  163.9  0.1   1.6   258.9  2   21
```

從以上的報告中，svc_t的時間顯然高了許多，而且s0,s2,s4,s6的%b>5，這都是可改進的部分，但目前這種運作狀況堪稱良好，並不是十分急迫要調整。若要改善此狀況，可先由降低I/O請求數量著手，這裡可以先降低cache flush的次數，有兩個參數可供調整

- tune_t_fsflushr 喚起fsflush daemon的時間，預設值為5秒。
- Autoup 每個週期要執行多久，預設值為30秒。

由於SAS程式所讀取的Oracle database資料之重複性很高，又不會更改資料，所以flush時間並不用太頻繁。可在/etc/system檔案中將這兩個參數設為

```
set     tune_t_fsflushr=10
set     autoup=120
```

同時，一個blocked的行程，也是磁碟瓶頸的現象，如果系統的

blocked行程多過執行行程(run process)，顯然CPU需花費較多的時間來等待I/O的完成。我們可以利用vmstat的指令來獲取系統運作資訊。

```
#vmstat 2
procsmemory                  page              disk      faults    cpu
r b w   swap  free  re  mf  pi  po  fr  de  sr  s0 s1 s2 s3 in  sy  cs  us ….
 1  1  34 ……..…1528……..
 0  0  71 ……..…4220……..
 0  1  71 ……..…4347……..
 0  2  71 ……..…4467……..
 0  1  71 ……..…4287……..
```

從以上的報表中可得之，較多的行程在等待I/O的完成。因此，持續對disk的副系統進行調校，平衡系統匯流排與disk裝置的使用、減少I/O的請求(如增加inode的快取大小)將有助於系統效能的提昇。

CPU估算：

為蒐集系統運作資料，系統之accounting功能需被啟動。啟動後，我們可以使用sar –u之指令觀察CPU運作之情形，如下所列

```
#sar -u
SunOS pansco-sdm 5.6 Generic_15181-13 sun4u 07/15/99
00:00:01        %u        %sys        %wio        %idle
01:00:01        5         8           62          26
02:00:01        0         3           1           95
……..…..
06:00:01        10        13          5           72
07:00:03        37        53          4           6
08:00:05        33        61          5           0
……..…
09:20:02        33        54          12          1
09:40:01        32        48          20          0
10:00:01        35        48          17          0
10:20:01        26        32          39          3

Average         16        26          12          45
```

從以上的報表中，從7點開始，CPU的idle 比率已趨近於0，且

09:20起等待I/O完成的比率亦大幅升高，再次確認I/O裝置為瓶頸問題之一。同時，為確保CPU運作正常，使用mpstat指令來觀察各個CPU運作狀況

```
#mpstat 2
CPU minf mjf xcal intr ithr csw icsw migr smtx srw syscl usr sys wt idl
0 ..... 39
1 ..... 36
4 ..... 38
5 ..... 37
8 ..... 44
9 ..... 44
12 ..... 45
13 ..... 42
CPU minf mjf xcal intr ithr csw icsw migr smtx srw syscl usr sys wt idl
0 ..... 18
1 ..... 19
4 ..... 0
5 ..... 3
8 ..... 16
9 ..... 22
12 ..... 26
13 ..... 16
```

因每個CPU仍然有許多idle時間，因此並不需要增加其他CPU模組。

記憶體估算：

實體記憶體的大小可從prtdiag或prtconf指令得知。而目前的系統其實體記憶體的大小為6Gbytes

```
#prtconf
System Configuration:  Sun Mircosystems sun4u
Memory size:6144 Megabytes
.....
```

其中，分配給kernel使用的大小，可由sar –k指令所得之alloc欄位

得知

```
#sar -k
SunOS pansco-sdm 5.6 Generic_15181-13 sun4u 07/15/99
00:00:01 sml_mem alloc fail   lg_mem alloc fail ovsz_alloc fail
01:00:01 68993024 51091032 0  228032512 150674684 0 .. 2458432  0
02:00:01 ·····
·····..
Average 62946236 36593635 0  142180352 86621538 .0··· 30758775 0
```

分配給kernel使用的記憶體，含sml_mem, lg_mem, ovsz_alloc共有153Mbytes。對於一般的程式來說，也可以pmap指令得知某依行程使用的記憶體知情形

```
#pmap -x 3639
3639 sas
Address  Kbytes  Resident Shared  Private Permissions  Mapped File
0000200      8       8       -       8       read     [anon]
·····..
EFFF4000 48      16      -       16  read/write/exec      [stack]
-------- ------  ------  ------  ------
total Kb 20376   9904    6712    3192
```

得知SAS行程使用的真實記憶體有9904Lbytes，其中分享式記憶體佔6712Kbytes供shared library使用、3192Kbytes則為此行程的自有記憶體大小。計算200個使用者所需之記憶體為638400Kbytes(638Mbytes)，而系統有6Gbytes，故足以因應此服務之所需。

DNLC(Domain Name Lookup Cache)負責快取最近最常被參考的目錄內容名稱與其相關的vnode資料，從sar –a指令中的namei/s欄位即可得知運作狀況。

```
#sar -a
SunOS pansco-sdm 5.6 Generic_15181-13 sun4u 07/15/99

00:00:01      iget/s   namei/s  dirbk/s
```

```
01:00:01        1         240       1079
08:40:01       58         296        609

Average         3         209        982
```

假如在DNLC中找不到目錄名稱，iget就會被呼叫去得到檔案與目錄的inode，這也是大部分iget的呼叫起因於DNLC miss的原因。在上面所列的報表之iget/s欄位及是代表被查詢的inode並未被快取起來的次數。Vmstat –s指令可得知DNLC報查詢的命中率多寡

```
#vmstat -s
1156 swap ins
         916       swap outs
        2312       pages swapped in
       16226       pages swapped out

   . 156855213     total name lookups (cache hits 78%)

#
```

一般而言，90%以上的命中率會有較佳的系統效能(不必做太多 disk I/O)，目前只有78%，因此亟待改進。Inode 的快取大小可經由參數 ufs_ninode予以調整，可從netstat –k指令中得知其運作的詳細狀況

```
#netstat -k
kstat_types:
raw 0 name=value 1 interrupt 2 i/o 3 event_timer 4
segmap:
fault ...  faulta ... getmap ... get_use ...  get_reclaim ...get_reuse ...

inode_cache:
size 3607 maxsize 17498 hits 770740 misses 591785 ...kmem frees 92065 maxsize
reached 21216...
#
```

maxisize的值等於ufs_ninode就是inode快取的數量，maxsize reached值為實際被參考的inode數量，若maxsize reached>maxsize代表快取的大小已不足所需，需要調整。

而且，我們亦可利用指令sar -g觀察inode flushed現象

```
#sar -g
SunOS pansco-sdm 5.6 Generic_15181-13 sun4u 07/15/99
00:00:01 pgout/s  ppgout/s  pgfree/s  pgscan/s  %ufs_ipf
01:00:01 0.16     0.26      1087.63   1055.39   0.00
02:00:01 0.16     0.27      77.66     83.49     0.00
03:00:01 0.15     0.27      82.27     72.65     0.00
.....
08:00:05 176.68   1383.28   6086.32   4932.94   0.25
08:20:02 205.35   1614.82   6049.20   4643.43   0.22
08:40:01 188.46   1497.26   5477.24   4238.56   0.17
.....
09:20:02 113.15   883.76    4027.47   3246.40   0.10
.....

Average  50.05    392.96    2052.27   1709.92   0.07
```

任何ufs_ipf非為0表示其inode快取太小以致於無法應付目前的負荷，因為此欄位的意義是flushed inode的比率。為增加DNLC快取的大小，並增加命中率，可修改系統兩個參數ncsize、ufs_ninode。我們將其設在/etc/system中，如下

```
set ufs_ninode=20000
set ncsize=20000
```

UFS緩衝快取是專門儲存 inode、cylinder groups與間接區塊。一般而言，其值為是系統所預設，是記憶體的2%

```
#sysdef|grep bufhwm
126189568 maximum memory allowed in buffer cache (bufhwm)
```

因為此緩衝器只有暫存inode與metadata，所以它並不需要一個非常大的記憶空間，其實你只需要給每個inode 300bytes、每2Gbytes檔

案存取1Mbytes的緩衝空間。那究竟多大的值才符合需求呢？ 舉例說明，假使你現在有一資料庫含有100個檔案，全部所需的儲存空間為100Gbytes，你估計同時間每個檔案有500Mbytes被讀取，那麼你需要有30Kbytes(100x300bytes)給inode使用 、((100x500Mbytes)/2Gbytes)x 1Mbytes=25Mbytes。所以建議在/etc/system中加入

```
set bufhwm-28000
```

並使用sar –b指令觀察快取命中率。

最佳觀察RAM的使用情形是由vmstat指令的scan rate(sr)欄位得知，如果此值在30秒內超過200page/sec的話，表示記憶體已有不足的現象。

```
#vmstat 2
procsmemory                    page                disk      faults    cpu
r  b  w    swap  free  re  mf  pi  po  fr  de  sr  s0 sl s2 s3 in  sy  cs  us ….
 1  1  34 ……….2439….…1528……….
 0  0  71 ……….11392…4220……….
 0  1  71 ……….12948…4347……….
 0  2  71 ……….20432…4467……….
 0  1  71 ……….20100…4287……….
```

由於scan rate(sr)遠高過200pages/sec，且pageout(po)5 之數值亦十分高，在在顯示記憶體短缺的現象。為解決此問題，我們可以調整lotsfree參數使系統充分保留所需之記憶空間供SAS應用程式所需。建議此系統之lotsfree設為256Mbytes，故在/etc/system中加入

```
set lotsfree=0x10000000
```

或者，為確保應用程式執行順利，不受檔案快取的影響，可啟動priority paging功能，這功能會在檔案快取的部分予以設限，使其不會因檔案系統的I/O導致應用程式換頁的現象。在/etc/system中設定如下

```
set priority_paging=1
```

檔案系統調校

檔案系統的效能與應用程式有者高度的相關性，因此在設定檔案系統之初，我們必須先對應用程式有充分的認識。

SAS基本上是一個決策支援系統，而其特性是每筆I/O之資料量大(64Kbytes-1Mbytes)、大部份的I/O均為循序存取。為符合此一需求，我們將檔案系統的Cluster大小調整為512

#newfs –C 512 /dev/rdsk/xxx

又由於SCSI驅動程式的限制，最大的SCSI傳輸量為128Kbytes，因此過大的request將會被切割而降低系統效能，所以我們需要對maxphys予以調整，在/etc/system加入如下的參數設定

set maxphys=16777216

這裡要再一次的提醒，所有的系統調整，必須依步驟逐步試驗，並記錄結果，保留有效的部分並討論，無效果的方面也要確認後再還原。所有的調校步驟是一個一再重複的程序，這是必經之路，不可不慎，尤其針對運作中的系統。

伍、實習心得與結語

個人在維運HiNet基礎應用服務系統的五年中，經歷過許多的系統上的嚴厲挑戰。每次面臨困難的問題與障礙，總得再次地審視問題以釐清發生問題的原因並尋求解決之道；然而，並不是每一次都是如此幸運且順利地解決，甚或有些問題至今仍無答案。但是為什麼會造成如此情形呢?究竟是什麼關鍵點讓問題無法釐清?這些問題，終於在我參加這次的出國受訓有了初步的答案，經由一位在SUN工作經驗非常豐富的老師 Jake Kanmore告訴我們許多系統上不廣為人知與不易習得的知識與經驗，讓自己對系統的認知也更深、更有能力探究問題發生的原因。

其實，在這五年的過程中，由於工作範圍較窄且無法有機會與相關的資深系統工程師交換心得或更進一步的派訓來提昇技能，使得技術的領域一直無法有長足的進步。幸運的是，經由這次的訓練，學得在台灣無法學的視野與技術，算是邁開不小的一步，收穫非常多，所以很感謝這次公司所給予的機會與相關協助的長官、同事與朋友。

從受訓的過程中老師不斷地提醒，習得求知或解決問題的方法，是必須先從運作的基本原理了解、洞悉後，才能確實習取整體系統知識、融會貫通，以確定問題之所在，否則不僅無法了解全貌，更會使真正的問題無法被發現，更埋下隱憂，不可不慎。

陸、附錄

參考資料與網站

1. Sun Performnce and tuning(Java and the Internet), Adrian Cockcroft and Richard Petit. (2nd Edition) ISBN 0-13-095249-4
2. solaris Tunable Parameters Reference Manual, on docs.sun.com web site(PDF fuke downloadable).
3. Solaris Internals(Core Kernel Architecture), Jim Mauro and Richard McDougall. ISBN 0-13-022496-0. An excellent book to obtain more details on how solaris works.
4. Unix Internals(The New Frontiers), Uresh Vahalia. ISBN 0-13-10908-2. Covers various flavors of Unix.
5. System Performance Tuning, Mike Loukides. ISBN 0-937175-60-9. O'Reilly book
6. Oracle Performance Tuning. 2nd edition. <rk Curry and Peter Corrigan. ISBN 1-56592-237-9. O'Rilly book.
7. RCP/IP Illustrated Bolime 1, W. Richard Stevens. ISBN 0-201-63346-9.
8. Configuration and Capacity Planning for Solaris Servers, Brian Wong. ISBN 0-13349-952-9.
9. Fibre Channel For Sans, Alan Brenner. ISBN 0-07137-413-2. " A new 2001 book".
10. Resource Management, McDougall, Cockcroft, Vargas, Etc. Sun Blueprints. ISBN 0-13-025855-5.
11. www.ancot.com <http://www.ancot.com> - you can request a free copy of "The basics of SCSI" or "What is Fiber Channel".
12. docs.sun.com-search for the Solaris tunables manual.
13. Sunsolve.sun.com-white papers on tuning and setting kernel parameters.
14. configuring & tuning Datases on Solaris, A Packer. ISBN 0-13-083417-3.
15. Plotting software/graphics tools:

    Rrd.tool.com

    ww.sarge.org

    www.izone.org <http://www.izone.org>

    www.orcaware.com <http://www.orcaware.com>
16. sunworld.com SunWorld Online
17. sun.icsnet.com Inofficial Guide to Solaris

18. sunsolve.sun.com Sun tech info and patches
19. geek-girl.com Solaris technical information
20. spinweb.net/solaris/adm/ Security information
21. ugu.com Unix guru I iverse
22. umbc8.umbc.edu/~vijay/solaris/solaris.html Solaris tips and tricks
23. eis.com/html/listmain.html Solaris on Intel
24. sunfreeware.com Solaris Freeware in pkg format; personal copy of Solaris 7