



Sun™ Grid Engine，企業版 5.3 軟體管理基礎

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054 U.S.A.
650-960-1300

文件編號：816-7484-10
2002年7月，修訂版A

請將您對本文件的意見寄到：docfeedback@sun.com

Copyright 2002 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054 U.S.A. 版權所有。

Sun Microsystems, Inc. 擁有本文件內說明的產品所含之技術的相關智慧財產權。特別是，且無限制地，這些智慧財產權可包含一或多項 <http://www.sun.com/patents> 中列示的美國專利，以及一或多項在美國或其他國家的專利或申請中的專利。

本文件以及其所屬的產品按照限制其使用、複製、分發和反編譯的授權可進行分發。未經 Sun 及其授權許可頒發機構的書面授權，不得以任何方式、任何形式複製本產品或本文件的任何部分。

協力廠商軟體，包括字型技術，由 Sun 供應商提供許可和版權。

本產品的某些部分從 Berkeley BSD 系統衍生而來，經 University of California 許可授權。UNIX 是在美國和其他國家的註冊商標，經 X/Open Company, Ltd. 獨家許可授權。

Sun、Sun Microsystems、Sun 標誌、AnswerBook2、docs.sun.com 和 Solaris 是 Sun Microsystems, Inc. 在美國和其他國家的商標或註冊商標。

所有的 SPARC 商標都按授權許可使用，是 SPARC International, Inc. 在美國和其他國家的商標或註冊商標。具有 SPARC 商標的產品都基於 Sun Microsystems, Inc. 開發的架構。

OPEN LOOK 和 Sun™ 圖形使用者介面是 Sun Microsystems, Inc. 為其使用者和授權許可持有人開發的。Sun 承認 Xerox 在為電腦行業研究和開發可視或圖形使用者介面方面所作出的先行努力。Sun 以非獨佔方式從 Xerox 獲得 Xerox 圖形使用者介面的授權許可，該授權許可涵蓋實施 OPEN LOOK GUI 且遵守 Sun 的書面許可協議的授權許可持有人。

本資料按「現有形式」提供，不承擔明確或隱含的條件、陳述和保證，包括對特定目的或非侵害性的商業活動和適用性的任何隱含保證，除非這種不承擔責任的聲明是不合法的。



目錄

關於格線 (Grid) 運算和 Sun Grid Engine，企業版 5.3 軟體	2
分配資源管理的角色	2
策略、主機和常駐程式	3
一般管理作業的指令行指示	5
▼ 如何從主機新增或移除管理權限	5
▼ 如何新增執行主機	5
▼ 如何移除執行主機	6
▼ 如何新增或移除提交主機	6
▼ 如何顯示目前主機的名稱	6
▼ 如何管理佇列	7
▼ 如何變更主要主機	7
▼ 如何設定備用主要主機	8
將 Script 和檔案使用於管理作業	9
▼ 如何使用檔案新增或修改物件	9
▼ 如何使用檔案修改佇列、主機和環境	10
▼ 如何使用檔案修改全域配置或排程器	14
▼ 如何微調格線環境	15
策略	17

功能策略設定 17

- ▼ 如何建立以使用者為主之功能排程 17
- ▼ 如何建立以專案為主之功能排程 18
- ▼ 如何建立以部門為主之功能排程 18
- ▼ 如何於每項專案中，與 FCFS 建立以專案為主之共享樹排程 19
- ▼ 如何於每項專案中，建立以專案為主之共享樹排程，並平等分享資源予每位使用者 19
- ▼ 如何於每項專案中，建立以專案為主之共享樹排程及單一使用者之共享資源 20

一般問題疑難排解 21

問題診斷 26

沒有分派擱置的作業 26

在錯誤狀態 E 報告的作業或佇列 27

Sun Grid Engine，企業版 5.3 軟體 管理基礎

本文件是做為協助 Sun™ Grid Engine，企業版 5.3 系統管理者快速參考手冊之用。除了與這類管理相關的一般作業的指示外，本文件還包含一般的格線運算和以及特定的 Sun Grid Engine，企業版 5.3 產品的簡短討論。本文件總結出數個微調和疑難排解的提示。

將此手冊當成《*Sun Grid Engine，企業版 5.3 管理和使用者指南*》的補充資料（而非替代品），其中您可以找到 Sun Grid Engine，企業版 5.3 概念和程序的詳細說明。

本文件包含完成以下作業的指令。

- 第 5 頁的「如何從主機新增或移除管理權限」
- 第 5 頁的「如何新增執行主機」
- 第 6 頁的「如何移除執行主機」
- 第 6 頁的「如何新增或移除提交主機」
- 第 6 頁的「如何顯示目前主機的名稱」
- 第 7 頁的「如何管理佇列」
- 第 7 頁的「如何變更主要主機」
- 第 8 頁的「如何設定備用主要主機」
- 第 9 頁的「如何使用檔案新增或修改物件」
- 第 10 頁的「如何使用檔案修改佇列、主機和環境」
- 第 14 頁的「如何使用檔案修改全域配置或排程器」
- 第 15 頁的「如何微調格線環境」
- 第 17 頁的「如何建立以使用者為主之功能排程」
- 第 18 頁的「如何建立以專案為主之功能排程」
- 第 18 頁的「如何建立以部門為主之功能排程」
- 第 19 頁的「如何於每項專案中，與 FCFS 建立以專案為主之共享樹排程」
- 第 19 頁的「如何於每項專案中，建立以專案為主之共享樹排程，並平等分享資源予每位使用者」
- 第 20 頁的「如何於每項專案中，建立以專案為主之共享樹排程及單一使用者之共享資源」

- 第 21 頁的「一般問題疑難排解」
- 第 26 頁的「問題診斷」

注意 – 本文件中大部分的資料原本都出現在 Sun Grid Engine 專案網站的「How-To」一節中。該網站因為經常更新，所以其對 Sun Grid Engine，企業版 5.3 系統管理員有特殊的價值，並值得經常上線查詢資料。網站的 URL 為：
<http://gridengine.sunsource.net/project/gridengine/howto/howto.html>

關於格線 (Grid) 運算和 Sun Grid Engine，企業版 5.3 軟體

若您不熟悉格線 (*grid*) 運算管理，您是屬於絕大多數專業電腦管理員中的一個。因為啟動軟體技術到現在還不到十年的光景，所以很少人能夠宣稱自己真的是有經驗的格線運算環境管理員。根據格線運算領域的先趨 Ian Foster 的說法，與格線運算相關的概念直到 1995 年才有人「探索」。因此，即使最早採用格線技術的人的經驗也不到十年。

可是，不管你的格線運算經驗多麼地貧乏，您還是有各種使用格線的經驗 — 雖然是不同種類的格線。每次您「開啓」電子開關，您就在使用格線的資源。在大部分的工業化國家中，提供給家庭和企業的電力是由數個將其產品供應給一般的資源儲存區（其又稱為格線）的獨立電源廠商所組成的「電力格線」所供給的。

資源儲存區對消費者和廠商都有益處。電力廠商可以其最飽和量來運轉電廠並藉此獲益，即使當地的消費者可能不需要電廠所產生的所有電源。例如，居住在離城市很遠的鄉村地區的消費者通常不會需要原本電廠所產生那麼多的電力。然而，電力格線的安排可以讓電廠將「過多」的容量賣到格線中，因此可以服務某些電廠電力供應不足之地區的消費者。

將低利用資源用於生產並傳送這些資源到需要者的手中（不管這些需求的所在位置）的能力是電力格線最明顯的特性。格線運算概念的先趨藉由採用「格線」名稱，建立了一個適當的類比。格線運算的基本目標是儲存數百個 — 甚至數千個 — 閒置和近乎閒置電腦的低利用資源，並使該結合的、過度的運算能力能符合運算密集消費者的需求。

分配資源管理的角色

架構細分後，全在傳輸的細節中。電源格線透過連接將格線廠商成員和電力消費者的家庭、工廠、農場和辦公室等最終目標之間的電線來管理儲存資源。電力在連線並供入電源格線後就會根據需求而流動。「開啓」電燈開關，電力就會從某處的電廠，透過電源

格線，流到燈泡中。格線運算環境中的運算資源也可以稱為共同「連線」，雖然是透過網路技術：主要為網際網路以及企業內部網路。但是您不能打開網路電腦上的開關，就期望使用運算格線的資源或貢獻資源給運算格線。

只有在安裝特殊的**中介軟體**的軟體之後，您才能啟動對運算格線的存取。中介軟體是介於電腦作業系統，如 Sun 的 Solaris™ 作業環境或 Linux 作業系統，和應用程式軟體，如三度空間圖形描繪程式，之間的軟體階層。Sun Grid Engine 軟體是啟動參予運算格線的中介軟體。

之前「開放來源」格線運算專案的實際結果，Sun Grid Engine 是在多重使用者和多重運算資源 — 通常位於遠端 — 進行調解的**分配資源管理**軟體，其提供使用者更多不同的運算能力，以處理複雜的作業。透過 Sun Grid Engine 軟體，遠距的使用者可以使用加入格線的閒置運算資源，如此可增加使用者和資源的生產力。例如，原本只以實際容量的四分之一執行的運算資源在加入格線後顯示以近乎百分之百的容量執行。

運算裝置說明「分配資源管理」一詞的「分配資源」部分。Sun Grid Engine 軟體說明「管理」一詞不可或缺的部分。沒有這樣的管理，就會導致混亂。做為格線管理員，Sun Grid Engine 接受使用者提交的作業，並根據資源管理**策略**在適當的系統上排定執行的時程，其中策略是由組織的技術和管理人員所設定的。

沒有兩個格線是相同的；一種尺寸不能符合所有的情況。共有三個主要格線類別，階層從單一系統到使用數千個處理器的超級電腦類別的運算中心 (compute farm)：

- **叢集格線 (Cluster grid)**，由基礎的 Sun Grid Engine 5.3 軟體啟動，是最簡單的格線，包含一起運作的電腦主機，提供使用者在單一專案或部門中單一的存取點。
Sun Grid Engine，企業版 5.3 軟體延伸此基本模式來建立其他兩個較複雜且功能更為強大的格線類別。
- **園區格線 (campus grid)** 能使組織裡的多重專案或部門分享計算資源。組織可以使用園區格線來處理各式各樣的作業，從週期的企業過程到產生資料、資料探勘和更多的作業。
- **全域格線 (Global grid)** 為跨越組織界線的園區格線 (campus grid) 的集合，以建立龐大的虛擬系統。使用者能存取遠超過其組織範圍內能使用資源之計算能力。

Sun Grid Engine，企業版 5.3 軟體提供園區格線所需的電源和靈活性。此產品對由基礎的 Sun Grid Engine 軟體所啟動的現存叢集格線非常有用，因為它促成平穩的轉變，並藉由聯合園區上所有現存的 Sun Grid Engine 叢集格線來建立園區。本產品是企業園區第一次對格線運算模式採取行動的好的開始。

策略、主機和常駐程式

Sun Grid Engine，企業版 5.3 軟體會妥善安排以您（系統管理員）所維護之企業資源策略為基礎的運算能力的傳送。Sun Grid Engine，企業版 5.3 系統使用這些策略檢查園區格線裡的可用運算資源、收集這些資源，然後以將橫跨園區格線用法最佳化的方式自動配置並傳送資源。

若要在園區格線裡啓動合作，使用格線的專案所有人必須協商策略、對獨一無二的專案需求的手動置換策略能有靈活性、並使策略能自動受到監視且執行。

做為管理員，您可以定義高階的使用策略，根據任何適合站點的事物來自訂。以下有四項策略，如欲詳細解說，請參閱《*Sun Grid Engine，企業版 5.3 管理和使用者指南*》。

- 功能性策略
- 共享為基礎策略
- 截止期限策略
- 覆寫策略

Sun Grid Engine，企業版 5.3 策略管理會自動控制叢集內共享資源的使用，以達成您的目標。高度優先權的作業會優先分配並在與其他低優先權的作業爭取資源時會獲得較大的 CPU 使用權。Sun Grid Engine，企業版 5.3 會監視所有作業的進度並依照策略所定義的目標調整作業相關的優先權。

不論您站點的使用規則是鬆還是緊，Sun Grid Engine，企業版 5.3 策略模組都會容納它們。這個以策略為基礎的資源分配在累積的時期期間（如一星期、一個月或一季）給予每個使用者、團體、部門和所有的專案系統資源的配置共享。

有四類型的主機對 Sun Grid Engine，企業版 5.3 系統是十分重要的。

- 主要主機
- 執行主機
- 管理主機
- 提交主機

此外，建議做為管理員的您建立**備用主要主機**。雖然主要主機上只有一個，但叢集內的其他機器還是可以指定為備用主要主機，以提供更大的可用性。備用主要主機會持續監視主要主機，並會在主要主機故障時自動並透明地假設控制。已經在叢集的作業不會受主要主機故障的影響。

有關各主機詳細解說，請參閱《*Sun Grid Engine，企業版 5.3 管理和使用者指南*》。

四個常駐程式提供 Sun Grid Engine，企業版 5.3 功能。

- `sgc_qmaster` 是主要常駐程式。
- `sgc_schedd` 是排程器常駐程式。
- `sgc_execd` 是執行常駐程式。
- `sgc_commd` 是通訊常駐程式。

有關各常駐程式詳細解說，請參閱《*Sun Grid Engine，企業版 5.3 管理和使用者指南*》。

本文件的剩餘部分提供如何完成最常見的 Sun Grid Engine，企業版 5.3 管理作業的指示。若要取得未在此提到的完成作業之額外背景資訊和指示，請參閱《*Sun Grid Engine，企業版 5.3 管理和使用者指南*》、《*Sun Grid Engine 5.3 及 Sun Grid Engine，企業版 5.3 參考手冊*》、或是線上說明手冊。

一般管理作業的指令行指示

注意 – 以下的指示群組是以只在指令行使用指令做為根據。所有的指令在 Sun Grid Engine，企業版 5.3 圖形使用者介面，QMON 也有對應的功能。欲得知如何使用 QMON 來完成以下工作，請參閱 《Sun Grid Engine，企業版 5.3 管理和使用者指南》。

▼ 如何從主機新增或移除管理權限

- 若要新增管理權限，請輸入以下指令。

```
qconf -ah
```

- 若要移除管理權限，請輸入以下指令。

```
qconf -dh
```

▼ 如何新增執行主機

1. 輸入以下指令使新主機成為管理主機。

```
qconf -ah
```

2. 以新主機上的 root 使用者身分從 \$SGE_ROOT 執行以下的 script。

```
install_execd
```

▼ 如何移除執行主機

- 輸入以下指令以刪除與該主機相關的佇列。

```
qconf -dq
```

- 輸入以下指令以刪除該主機。

```
qconf -de
```

▼ 如何新增或移除提交主機

- 若要指定主機為提交主機，請在主機指令行輸入以下指令。

```
qconf -as
```

- 若要移除指定為提交主機主機，請在主機指令行輸入以下指令。

```
qconf -ds
```

▼ 如何顯示目前主機的名稱

- 若要顯示管理主機的名稱，請輸入以下指令。

```
qconf -sh
```

- 若要顯示提交主機的名稱，請輸入以下指令。

```
qconf -ss
```

- 若要顯示執行主機的名稱，請輸入以下指令。

```
qconf -sel
```

▼ 如何管理佇列

- 若要新增佇列，請輸入以下指令。

```
qconf -aq
```

- 若要從檔案新增佇列，請輸入以下指令。

```
qconf -Aq
```

- 若要刪除佇列，請輸入以下指令。

```
qconf -dq
```

- 若要修改佇列，請輸入以下指令。

```
qconf -mq
```

- 若要變更一個以上的佇列之單一屬性，請輸入以下指令。

```
qconf -mqattr
```

▼ 如何變更主要主機

1. 輸入以下指令以停用目前主要主機上的主要和排程常駐程式。

```
qconf -ks -km
```

2. 根據以下的指導方針，修改 `$SGE_ROOT/default/common/act_qmaster` 檔。

- a. 在 `act_qmaster` 檔案中，用新的主要主機名稱取代目前的主機名稱。

該名稱應該和 `gethostname` 公用程式所傳回的名稱相同。若要取得該名稱，請在新的主要主機上輸入以下指令。

```
$SGE_ROOT/utilbin/$ARCH/gethostname
```

- b. 使用 `gethostname` 公用程式所傳回的名稱取代 `act_qmaster` 檔案裡的舊有名稱。
3. 在新的主要主機上，執行以下的 `script`。

```
$SGE_ROOT/default/common/sge5
```

這將會在新的主要主機上啟動 `sge_qmaster` 和 `sge_schedd`。

▼ 如何設定備用主要主機

1. 根據以下的指導方針，建立 `shadow_masters` 檔。
 - 在 `$SGE_ROOT/default/common` 建立檔案。
 - 將主要的主要主機名稱做為第一行，然後列出選擇的其他主機，以您所要的順序假定主要的責任。例如：

```
system% cat shadow_masters
host1
host2
host3
```

在上面的範例中，`host1` 是主要的主要主機。假如 `host1` 故障的話，`host2` 會在過了約 10 分鐘左右進行接管，成為主要伺服器。如果 `host2` 接著故障的話，`host3` 就會進行接管。

2. 驗證正確權限。

所有的主要備用主機都必須有 `qmaster` 多工緩衝處理目錄的讀取 / 寫入權限。
3. 使用 `sge5` 啟動 `script` 來啟動備用常駐程式。做為每個主機上的 `root` 使用者，請輸入以下指令。

```
$SGE_ROOT/default/common/sge5 -shadowd
```

在您完成這些步驟後，Sun Grid Engine，企業版 5.3 叢集的主要備用會在作用中。

將 Script 和檔案使用於管理作業

注意 – 您可以使用 QMON 圖形使用者介面來執行所有的 Sun Grid Engine，企業版 5.3 管理作業，而使用該介面的附加的好處是您可以知道全部的系統能力。但是，您也可以透過發出給 shell 提示符號的指令和從 shell script 呼叫的指令來管理 Sun Grid Engine，企業版 5.3 格線。許多有經驗的管理員覺得這會是較有彈性、更加迅速及功能更強的變更設定值的方法。以下三套指示處理這類的方法。

▼ 如何使用檔案新增或修改物件

- 根據您在檔案建立的規格，使用 `qconf` 指令新增或修改物件。

該指令的語法如下：`qconf -{A,M}<物件> <檔案名稱>`

在以上的語法中，`-A` 表示新增，`-M` 表示修改。`<物件>` 變數可以是以下的任何事物。

- `c` – 複雜的
- `ckpt` – 檢查點環境
- `e` – 執行主機
- `p` – 平行環境
- `q` – 佇列
- `u` – 使用者集

您可以和 `qconf` 指令 — `qconf -s<物件>` — 的顯示選項一起使用這個選項，來取用現存的物件、進行修改，並更新現存物件或建立新的物件。

範例

以下為修改檢查點環境的遷移指令的 shell script 範例。

```
#!/bin/sh
# ckptmod.sh: modify the migration command
# of a checkpointing environment
# Usage: ckptmod.sh <checkpoint-env-name> <full-path-to-
command>
TMPFILE=/tmp/ckptmod.$$

CKPT=$1
MIGMETHOD=$2

qconf -sckpt $CKPT | grep -v '^migr_command' > $TMPFILE
echo "migr_command $MIGMETHOD" >> $TMPFILE
qconf -Mckpt $TMPFILE
rm $TMPFILE
```

▼ 如何使用檔案修改佇列、主機和環境

您可以根據指令的引數，以兩個明確的方式之一使用 `qconf` 指令或結合其他指令，從指令行修改獨立的佇列、主機和平行與檢查點環境。以下的指示涵蓋兩項。

- 如果您已經準備好檔案，請輸入 `qconf` 指令和適當的選項。

該指令的語法如下；請特別注意字母選項的大小寫，因為大寫和小寫的差異是有不同意義的。

```
qconf -M{q,e,p,ckpt} < 檔案名稱 >
```

這些選項有以下的意義。

- -M – 從現存的檔案修改，< 檔案名稱 >
- q – 佇列
- e – 執行主機
- p – 平行環境
- ckpt – 檢查點環境

- 如果您尚未準備好檔案，請輸入 `qconf` 指令和適當的選項。

該指令的語法如下；請特別注意字母選項的大小寫，因為大寫和小寫的差異是有不同意義的。

```
qconf -m{q,e,p,ckpt} < 物件名稱 >
```

這些選項有以下的意義。

- -m – 藉由開啓文字編輯器進行修改，以建立修改檔案。
- q – 佇列
- e – 執行主機
- p – 平行環境
- ckpt – 檢查點環境

討論

指令間的差異 — 第一個指令的大寫 M 和第二個指令的小寫 m — 可控制指令的結果。-M 和 -m 選項都是表示修改的意思，但是大寫的 M 意謂著修改是來自現存的檔案，而小寫的 m 則沒有這種意思。小寫的 -m 會在編輯器中開啓一個暫存檔，而在您儲存對這個檔案的任何變更並退出編輯器時，系統會立刻反應這些變更。

然而，當您想要一次變更許多物件，或您想要非互動地變更物件配置時，請使用 qconf -...attr 指令集。

這樣的一套指令會根據檔案中的規格進行修改：

```
qconf - {A,M,R,D} attr queue | exechost | pe | ckpt < 檔案名稱 >
```

對應的指令集會根據指令行上的規格進行修改：

```
qconf - {a,m,r,d} attr queue | exechost | pe | ckpt < 屬性 > < 數值 > < 佇列清單 > | < 主機清單 >
```

在兩套的指令中，這些選項是表示以下的事物：

- -A/a – 新增屬性
- -M/m – 修改屬性
- -R/r – 取代屬性
- -D/d – 刪除屬性
- < 屬性 > – 要變更的佇列或主機屬性
- < 數值 > – 受影響的屬性數值
- < 檔案名稱 > – 包含屬性數值配對的檔案

a、m 和 d 選項能讓您在數值清單中的獨立數值上作業，而 r 將用在指令行或檔案中指定的新數值取代整個數值清單。

範例

- 請將 tcf27-e019.q 的佇列類型變更到 batch-only:

```
% qconf -rattr queue qtype batch tcf27-e019.q
```

- 根據檔案，new.cfg 的內容修改佇列類型和 tcf27-e019.q 的 shell 啟動運作方式：

```
% cat new.cfg
qtype batch interactive checkpointing
shell_start_mode unix_behavior
% qconf -Rattr queue new.cfg tcf27-e019.q
```

- 將名為 storage 和 license 的聚合體 (complex) 附接到主機上，tcf27-e019：

```
% qconf -rattr exechost complex_list storage,license tcf27-e019
```

- 增加數值為 1000M、名為 scratch1 的資源，以及數值為 2、名為 long 的資源：

```
% qconf -rattr exechost complex_values scratch1=1000M,long=2 tcf27-e019
```

- 附接數值為 4、名為 short 的資源至主機。

```
% qconf -aattr exechost complex_values short=4 tcf27-e019
```

- 將 scratch1 的數值變更至 500M，但其他的數值不變。

```
% qconf -mattr exechost complex_values scratch1=500M tcf27-e019
```

- 刪除資源，long：

```
% qconf -dattr exechost complex_values long tcf27-e019
```

- 將 tcf27-b011.q 加到檢查點環境，sph，的佇列清單中：

```
% qconf -aattr ckpt queue_list tcf27-b011.q sph
```

- 將平行環境 make 中的插槽數目變更為 50：

```
% qconf -mattr pe slots 50 make
```

使用 qselect 指令鎖定佇列為目標

qselect 指令輸出佇列清單。若使用選項呼叫，它只會列出符合特定規格的佇列。您可以結合 qconf -...attr 指令集好好利用這個指令，鎖定您想要修改的特定佇列。

範例

- 列出 Linux 機器上所有的佇列：

```
% qselect -l arch=glinux
```

- 列出有兩個 CPU 的機器上的所有佇列：

```
% qselect -l num_proc=2
```

- 列出所有 4 CPU 64 位元 Solaris 機器上的所有佇列：

```
% qselect -l arch=solaris64,num_proc=4
```

- 列出提供應用程式授權的佇列（先前配置的）：

```
% qselect -l app_lic=TRUE
```

您可以使用單一指令行結合 qselect 和 qconf，來做大範圍的變更。要進行這項工作，請將整個 qselect 指令放到反斜線 (/) 之內，並在 qconf 指令行上使用該指令代替 *< 佇列清單 >* 變數。

範例

- 在 Solaris 機器的所有佇列上，將 prolog script 設定為 sol_prolog.sh：

```
% qconf -mattr queue prolog /usr/local/scripts/sol_prolog.sh `qselect -l arch=solaris`
```

- 在兩個處理器系統的所有佇列上，將屬性，fluent_license，設為 2：

```
% qconf -mattr queue complex_values fluent_license=2 `qselect -l num_proc=2`
```

聯合 qselect 指令使用 qconf 指令會提供最有彈性的 Sun Grid Engine，企業版 5.3 佇列配置自動化的方式，讓您能建立自訂的管理 script。

▼ 如何使用檔案修改全域配置或排程器

- 若要變更全域配置，請根據下一節的指導，使用 qconf -mconf 指令。
- 若要變更排程器，請根據下一節的指導，使用 qconf -msconf 指令。

討論

這兩個指令都在編輯器中開啓暫存檔。當您離開編輯器時，系統就會處理您儲存在暫存檔中的任何變更，並會立即生效。用於開啓暫存檔的編輯器程式是 EDITOR 環境變數所指定的那個。如果沒有定義變數，則會使用 vi。

您可以利用 EDITOR 環境變數將 qconf -m... 指令的運作方式自動化。變更此變數的數值，來指向修改名稱為第一個引數所授與的檔案的程式。在此程式修改暫存檔並推出之後，系統將在修改中讀取並立刻更新。

注意 - 如果檔案的修改時間在編輯作業後沒有改變，則系統有時候會錯誤假定檔案還沒被修改。因此，您應該在寫入檔案之前插入一個 sleep 1 的指示，以確保不同的修改時間。

範例

以下範例的 script 會修改排程器的排程間隔時間。

```
#!/bin/ksh
# sched_int.sh: modify the schedule interval
# usage: sched_int.sh <n>, where <n> is
# the new interval, in seconds. n < 60

TMPFILE=/tmp/sched_int.$$
if [ $MOD_SGE_SCHED_INT ]; then
    grep -v schedule_interval $1 > $TMPFILE
    echo "schedule_interval 0:0:$MOD_SGE_SCHED_INT" >> $TMPFILE
# sleep to ensure modification time changes
    sleep 1
    mv $TMPFILE $1
else
    export EDITOR=$0
    export MOD_SGE_SCHED_INT=$1
    qconf -msconf
fi
```

上述的 script 會修改 EDITOR 環境使其指向自己，然後會呼叫 qconf -msconf 指令。第二個 script 的巢狀呼叫 (nested invocation) 然後會修改第一個引數所指定的暫存檔，之後就會退出。Sun Grid Engine，企業版 5.3 然後系統會自動在變更中讀取，而 script 的第一次呼叫就會中止。以上的技巧可以結合任何 qconf -m... 指令使用。然而，其對排程器和全域配置的管理特別有用，因為沒有其他的方式將此自動化。

▼ 如何微調格線環境

Sun Grid Engine，企業版 5.3 是個全功能、通用的分配資源管理 (DRM) 工具。系統的排程器元件支援大範圍、不同的運算中心情境。若要從您的運算環境中取得最大的效能，檢視哪些功能已經啟動以及您需要哪些功能來解決負載管理問題是值得的。停用這裡的某些功能可能會在叢集的總處理能力上對效能有所助益。

排程器監視作業

排程器監視作業會幫您找出沒有分派某些作業的原因。然而，在任何時刻提供此資訊給所有的作業可能會造成資源耗損，而且通常是不必要的。

- 若要停用排程器監視作業，請在排程器配置，sched_conf (5)，將 schedd_job_info 設為 false。

完成的作業

如果是陣列作業，在 `qmaster` 的完成作業清單可能會變得很大。將其關閉會同時節省記憶體並增加 `qstat` 過程的速度，因為 `qstat` 過程也會取得完成作業的清單。

- 要關閉完成作業清單功能，請在全域配置，`sge_conf(5)`，中將 `finished_jobs` 設為 0。

作業驗證

在作業提交時間的強制驗證可能會是非常重要的程序，可以防止無法分派的作業永遠停留在擱置的狀態。然而，要驗證作業也是件相當花時間的作業，特別是在有多樣的執行節點和耗損資源，而每個使用者有自己的作業簡介的異質環境中。在只有兩個不同作業的同執環境中，通常會忽略一般的作業驗證。

- 若要停用作業驗證，請將 `qsub(1)` 選項，`-wn`，加到全叢集的預設要求中（請參閱 `sge_request(5)`）。

載入臨界值和暫停臨界值

如果您要刻意超額配置 (`oversubscribe`) 您的機器而您需要預防過量的系統載入的機制的話，您就需要載入臨界值。暫停臨界值也同用於此。另一個需要載入臨界值的例子就是當執行節點仍開放給不受 Sun Grid Engine，企業版 5.3 系統控制的互動式載入，而您要預防節點過度載入。

如果運算中心比較單一目的取向 — 例如，每個運算節點的 CPU 只由單一佇列插槽所代表，而這些節點沒有預期互動式載入 — 則您可以忽略 `load_thresholds`。

- 要停用兩個臨界值，請將 `load_thresholds` 設為 `none`，並將 `suspend_thresholds` 設為 `none`（請參閱 `queue_conf(5)`）。

載入調整

載入調整用來在分派作業後虛擬增加測量過的載入。若為超額配置機器，則此機制是有效的，以達成和載入臨界值一致。如果不需要載入調整，您應該將其關閉，因為載入調整會將一些與分類主機和載入臨界值相關的額外工作加到排程器上。

- 若要停用載入調整，請在排程器配置，`sched_conf(5)`，中將 `job_load_adjustments` 設定為 `none`，並將 `load_adjustment_decay_time` 設定為 0。

依需求排程

Sun Grid Engine，企業版 5.3 系統的預設值是在固定的排程間隔時間啟動排程執行（請參閱 `schedd_conf(5)` 的 `schedule_interval`）。固定間隔時間的優點是它們可以限制 `qmaster/scheduler` 的 `cpu` 時間的消耗。缺點是調節排程器，人為造成有限的總處理能力。許多運算中心有特別為 `qmaster/scheduler` 所用的機器，而在這樣的設定中，沒有調節排程器的理由。

- 使用全域叢集設定，`sge_conf(5)`，的 `schedd_params` - 節中的 `FLUSH_SUBMIT_SEC` 和 `FLUSH_FINISH_SEC` 設定值來配置依需求排程。

若已啟動依需求排程，運算中心的總處理能力只會受到管理 `qmaster/scheduler` 的機器之能力的限制。

策略

Sun Grid Engine，企業版 5.3 軟體最重要的特徵之一為由系統管理員建立策略，以分配運算能力。請參閱第 3 頁的「策略、主機和常駐程式」一節以了解背景資訊。

下一節會討論如何為您站點建立基本的「功能性」及「共享」策略設定。而「截止期限」和「置換」策略設定，及其它「功能」和「共享」策略設定的指令，都包含在《*Sun Grid Engine，企業版 5.3 管理和使用者指南*》一書中。

功能策略設定

本節所提及的設定全以功能策略為主策略。此設定可確保所有使用者、專案、或部門隨時皆可得到定義之共享。當系統分配工作給閒置資源時，會傾向分配給比預估佔用資源較少之使用者、專案、或部門的工作。

同時，還可保證不浪費完整資源，因為未使用之共享部分會分配給需要的使用者、專案、及部門，而過去的資源消耗並不列入考慮之中。

▼ 如何建立以使用者為主之功能排程

此設定之目的是為了建立結合於 Sun Grid Engine，企業版 5.3 叢集中，所有資源之特定共享作業予不同的使用者。FCFS 排程是用於計劃同一使用者的工作。

1. 在全域配置 (`sge_conf(5)`) `schedd_params` - 節中，使用 `SHARE_FUNCTIONAL_SHARES=1`。
2. 指定功能票卷的數量（例如：1000000）於排序器配置 (`sched_conf(5)`) 中。

3. 新增每位與排程相關之使用者 (user (5)) 。

4. 分派功能共享給每位使用者。

並依比例分配。例如：

- 使用者 A (10)
- 使用者 B (20)
- 使用者 C (20)
- 使用者 D (50)

▼ 如何建立以專案為主之功能排程

此設定的目的是為了建立結合在 Sun Grid Engine，企業版 5.3 叢集內所有資源之特定分享作業予不同的專案。FCFS 排程是用於計劃同一專案之工作。

1. 在全域配置 (sge_conf (5))schedd_params 一節中，使用 SHARE_FUNCTIONAL_SHARES=1 。

2. 指定功能票卷的數量（例如：1000000）於排序器配置 (sched_conf (5)) 中。

3. 新增每個與排程相關之專案 (user (5)) 。

您可以藉由 acl 及 xacl 功能來控制存取至更高權限之專案。詳細功能說明請參閱 《Sun Grid Engine，企業版 5.3 管理和使用者指南》或 《Sun Grid Engine 5.3 及 Sun Grid Engine，企業版 5.3 參考手冊》。

4. 分派功能共享給每個專案。

並依比例分配。例如：

- 專案 A (55)
- 專案 B (45)

▼ 如何建立以部門為主之功能排程

此設定的目的是為了建立結合在 Sun Grid Engine，企業版 5.3 叢集內所有資源之特定分享作業予不同的部門。FCFS 排程是用於計劃同一部門的工作。

1. 在全域配置 (sge_conf (5))schedd_params 一節中，使用 SHARE_FUNCTIONAL_SHARES=1 。

2. 指定功能票卷的數量（例如：1000000）於排序器配置 (sched_conf (5)) 中。

3. 新增每個與排程有關之部門。

4. 分派功能共享給每個部門。

並依比例分配。例如：

- 部門 A (90)
- 部門 B (5)
- 部門 C (5)

本節所提及的設定全以共享樹策略為主策略。此設定可確保隨著時間改變，共享樹內配置之實例仍可得到定義之共享。當系統分配工作給閒置資源時，會傾向分配給過去比預估（共享樹共享）佔用資源較少之共享樹「分枝」的工作。同時，還保證不會浪費完整資源，因為未使用之共享部分仍會分派給其它與共享樹分枝相關之工作。

▼ 如何於每項專案中，與 FCFS 建立以專案為主之共享樹排程

此設定的目的是為了確保即使時間改變，不同的專案仍可得到結合在 Sun Grid Engine，企業版 5.3 叢集內所有資源之特定分享作業。FCFS 排程是用於計劃同一專案的工作。

注意 - 本節有些指令（與其它節不同）需要使用到 qmon 圖形使用者界面。欲瞭解 qmon 請參閱《Sun Grid Engine，企業版 5.3 管理和使用者指南》。

1. 指定共享樹票卷的數量（例如：1000000）於排序器配置 (sched_conf(5)) 中。
2. 新增每個與排程相關之專案 (user(5))。
3. 利用 qmon 圖形使用者界面來設定共享樹，以反映所有與排程相關之專案結構，是為節點。
4. 分派共享樹共享給各個專案。

請參閱以下簡易構造範例：

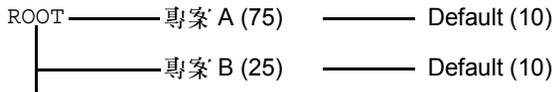


▼ 如何於每項專案中，建立以專案為主之共享樹排程，並平等分享資源予每位使用者

此設定的目的是為了確保即使時間改變，不同的專案仍可得到結合在 Sun Grid Engine，企業版 5.3 叢集內所有資源之特定分享作業。在同一專案中，能否平等共享資源為各工作人員之主要目標。

1. 指定共享樹票卷的數量（例如：1000000）於排序器配置 (sched_conf(5)) 中。
2. 新增每位與排程相關之使用者 (user(5))。
3. 新增每個與排程相關之專案 (user(5))。
4. 利用 qmon 圖形使用者界面來設定共享樹，以反映所有與排程相關之專案結構，是為節點。
5. 分派共享樹共享給各個專案。

請參閱以下簡易架構範例：



6. 在每個專案底下，新增使用者 default，是為共享樹之樹葉。

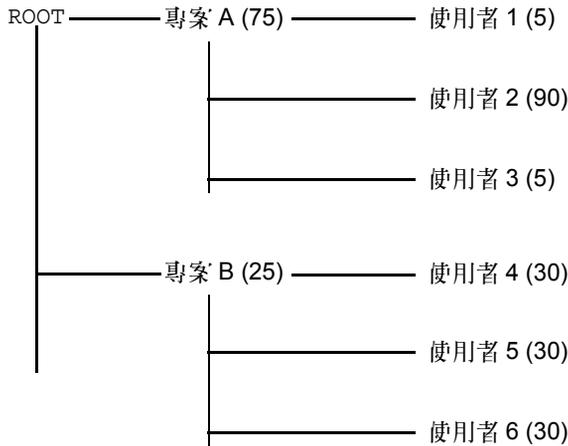
▼ 如何於每項專案中，建立以專案為主之共享樹排程及單一使用者之共享資源

此設定的目的是為了確保即使時間改變，不同的專案仍可得到結合在 Sun Grid Engine，企業版 5.3 叢集內所有資源之特定分享作業。建立單一使用者共享資源分配是必要的。

1. 指定共享樹票卷的數量（例如：1000000）於排序器配置 (sched_conf(5)) 中。
2. 新增每位與排程相關之使用者 (user(5))。
3. 新增每個與排程相關之專案 (project(5))。
4. 利用 qmon 圖形使用者界面來設定共享樹，以反映所有與排程相關之專案結構，是為節點。

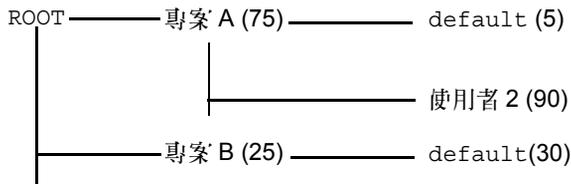
5. 分派共享樹共享給各個專案。

新增使用者至各專案中，是為共享樹之樹葉，且使用者並可存取分派單一共享資源。
請參閱以下簡易架構範例：



討論

若僅有少數使用者需分派單一共享，您可以與專案節點底下之單一使用者結合，來指定使用者 default。舉例來說，以上提及之共享樹架構就可以濃縮成以下版本：



一般問題疑難排解

使用下一節可以幫您診斷並回應一般問題的原因。

- 問題 – 您作業的輸出檔陳述 Warning: no access to tty; thus no job control in this shell...
 - 可能的原因 – 您的登入檔案中一或數個含有 stty 指令。這些指令只有在終端機存在時才有用。

- 可能的解決方法 – 在 Sun Grid Engine，企業版 5.3 批次作業中，沒有終端機和這些作業有關。您必須從登入檔案將所有的 `stty` 指令移除，或是將其加上大括弧，並附上於處理前會檢查終端機的 `if` 陳述。以下為一範例。

```

/bin/csh:
stty -g          # checks terminal status
if ($status == 0) # succeeds if a terminal is present
< 放置所有 stty 指令在此 >
endif

```

- 問題 – 作業標準錯誤日誌檔案陳述：``tty': Ambiguous`。然而，使用者的 shell（其在作業 script 被呼叫）沒有與 `tty` 有相關的事物。
 - 可能的原因 – `shell_start_mode` 根據預設值是 `posix_compliant`；因此，所有的作業 script 與在佇列定義中指定的 shell 一同執行，而不是與作業 script 第一行所指定的那個執行。
 - 可能的解決方法 – 將 `-S` 旗號用於 `qsub` 指令，或將 `shell_start_mode` 變更為 `unix_behavior`。
- 問題 – 您可以從指令行執行作業 script，但當您透過 `qsub` 指令執行時就會失敗。
 - 可能原因 – 可能您的作業有設定過程限制。若要對此進行測試，請寫入一個執行 `limit` 和 `limit -h` 功能的測試 script。在 shell 提示符號互動執行，並透過 `qsub` 指令執行來比較結果。
 - 可能的解決方法 – 確認移除任何在 shell 中設定限制的配置檔中的指令。
- 問題 – 執行主機回報負載 99.99。
 - 可能原因 – 有三種可能：
 1. `execd` 常駐程式並未在主機上執行。
 2. 預設領域界定錯誤。
 3. `qmaster` 主機判讀的執行主機名稱，與執行主機自行判讀的不同。
 - 可能的解決方法 – 依原因不同，遵照以下方法解決問題。（將「可能原因」的編號搭配以下解決方法編號。）
 1. 以 `root` 身份執行 `$SGE_ROOT/default/common/`rcsge` script`，於執行主機啓動 `execd` 常駐程式。
 2. 以 Sun Grid Engine 企業版管理員身份執行 `qconf -mconf` 指令，並改變 `default_domain` 變數為 `none`。
 3. 若您是使用 DNS 解析您電腦叢集中的主機名稱，請配置 `/etc/hosts` 和 NIS 以回復成完全符合之領域名稱 (FQDN: fully qualified domain name) 的主機名稱（當然，您仍可以定義並使用較短的別名；如，`168.0.0.1 myhost.dom.com myhost`）。

若您沒有使用 DNS，則請確定您所有的 `/etc/hosts` 檔案和 NIS 對照表一致；如，`168.0.0.1 myhost.corp myhost` 或 `168.0.0.1 myhost`。

- 問題 – 每隔 30 秒，一個與下面相似的警告會列印至
< 儲存格 >/spool/<host>/messages：

```
Tue Jan 23 21:20:46 2001|execd|meta|W|local
configuration meta not defined - using global configuration
```

但是 < 儲存格 >/common/local_conf/ 中的每個主機都有檔案，而每個都有 FDQN。

- 可能的原因 – 決定在您的機器，meta，的主機名稱傳回簡短的名稱；可是在您的主要機器上，傳回的是附有 FQDN 的 meta。
- 可能的解決方法 – 確認您所有的 `/etc/hosts` 檔和 NIS 表格在這方面是一致的。在此範例中，在主機，meta，的 `/etc/hosts` 檔案中可能有類似下面的一行錯誤文字：

```
168.0.0.1 meta meta.your.domain
```

但此行應該是：

```
168.0.0.1 meta.your.domain meta。
```

- 問題 – 通常您會在常駐程式的 messages 檔案中看到 CHECKSUM ERROR、WRITE ERROR 或 READ ERROR 訊息。
 - 可能的原因 – 只要這些訊息不出現在一秒鐘的間隔時間內（它們通常一天出現一到三十次），您就不須對此問題採取任何行動。
- 問題 – 作業在特定的佇列上結束，並在 qmaster/messages 傳回以下訊息：

```
Wed Mar 28 10:57:15 2001|qmaster|masterhost|I|job 490.1
finished on host exechost
```

可是您卻在執行主機的执行host/messages 檔中看到以下的錯誤訊息：

```
Wed Mar 28 10:57:15 2001|execd|exechost|E|can't find directory
"active_jobs/490.1" for reaping job 490.1
```

```
Wed Mar 28 10:57:15 2001|execd|exechost|E|can't remove
directory
"active_jobs/490.1": opendir(active_jobs/490.1)
failed:Input/output error
```

- 可能的原因 – \$SGE_ROOT 目錄，其為自動裝載，已經遭解除裝載，而造成 sge_execd 常駐程式遺失它的 cwd。
- 可能的解決方法 – 將本端多工緩衝處理目錄使用在 execd 主機上。使用 qmon 或 qconf 指令設定參數，execd_spool_dir。
- 問題 – 在使用 qrsh 公用程式提交互動作業時，您會接到以下的錯誤訊息：

```
% qrsh -l mem_free=1G error: error: no suitable queues
```

但是，使用 qsub 公用程式，批次作業還是可以使用佇列，而佇列也可以使用 qhost -l mem_free=1G 和 qstat -f -l mem_free=1G 查詢。

- 可能的原因 – 訊息，error: no suitable queues，可能來自 -w e submit 選項，其根據預設值活動於互動作業中（如 qrsh）（請查詢 qrsh(1) 中的 -w e）。如果 qmaster 不確定作業會根據目前的叢集配置而分派的話，則此選項會造成 submit 指令失敗。這個機制的目的是如果無法給予作業要求的話，這些作業要求就會事先被拒絕。
- 可能的解決方法 – 在這個例子中，mem_free 配置為可消耗的資源，但您沒有指定每個主機可用的記憶體數量。這個檢查刻意不將記憶體載入數值列入考慮，因為它們會變化，所以不能視為叢集配置的一部份。要克服這一點，您可以執行以下事項之一。

通常藉由置換 qrsh 預設設定值，-w e，來略過此項檢查，其由 -w n 明確地提交。您也可以將此放進 \$SGE_ROOT/<儲存格>/common/cod_request。

若您想管理 mem_free 為可消耗的資源，請使用 qconf -me <主機名稱> 在 host_conf(5) 的 complex_values 指定主機的 mem_free 容量。

若您不想管理 mem_free 為可消耗的資源，請使用 qconf -mc 主機在 complex(5) 的 consumable 欄位中將其再度變成不可消耗的資源。

- 問題 – qrsh 不會分派到它所在的同一個節點。從 qsh shell：

```
host2 [49]% qrsh -inherit host2 hostname
error: executing task of job 1 failed:

host2 [50]% qrsh -inherit host4 hostname
host4
```

- 可能的原因 – gid_range 不足。它應該定義為一個範圍，而不是單一號碼。Sun Grid Engine，企業版 5.3 系統指派一個明確的 gid 給主機上的每個作業。
- 可能的解決方法 – 使用 qconf -mconf 或 qmon 圖形使用者介面調整 gid_range。建議的範圍如下。

```
gid_range                20000-20100
```

- 問題 – `qrsh -inherit -v` 用於平行作業裡時無法運作。您會收到以下的訊息。

```
cannot get connection to "qlogin_starter"
```

- 可能的原因 – 這個問題隨著巢狀 `qrsh` 呼叫一起發生，並歸咎於 `-v` 開關。第一個 `qrsh -inherit` 呼叫會設定環境變數，`TASK_ID`（平行作業內緊密整合作業的 `id`）。第二個 `qrsh -inherit` 呼叫則會使用該環境變數來註冊此作業，當其嘗試使用已經在第一個作業執行的 `id` 來啓動作業時，該作業就會失敗。
- 可能的解決方法 – 您可以事先解除 `TASK_ID` 的設定，在您呼叫 `qrsh -inherit` 之前，或是選擇不要使用 `-v` 開關，而使用 `-v`，並只匯出您真正需要的環境變數。
- 問題 – `qrsh` 似乎完全無法運作。您收到類似下面的錯誤訊息。

```
host2$ qrsh -verbose hostname
local configuration host2 not defined - using global
configuration
waiting for interactive job to be scheduled ...
Your interactive job 88 has been successfully scheduled.
Establishing /share/gridware/utilbin/solaris64/rsh session to
host exehost ...
rcmd: socket: Permission denied
/share/gridware/utilbin/solaris64/rsh exited with exit code 1
reading exit code from shepherd ...
error: error waiting on socket for client to connect:
Interrupted system call
error: error reading return code of remote command
cleaning up after abnormal exit of
/share/gridware/utilbin/solaris64/rsh
host2$
```

- 可能的原因 – `qrsh` 的權限沒有正確設定。
- 可能的解決方法 – 檢查以下檔案的權限，這些檔案位於 `$SGE_ROOT/utilbin/`。（請注意，`rlogin` 和 `rsh` 需為 `setuid`，並為 `root` 所擁有。）

```
-r-s--x--x 1 root root 28856 Sep 18 06:00 rlogin*
-r-s--x--x 1 root root 19808 Sep 18 06:00 rsh*
-rwxr-xr-x 1 sgeadmin adm 128160 Sep 18 06:00 rshd*
```

注意 – `$SGE_ROOT` 目錄也需要以 `setuid` 選項裝載在 NFS。如果是從提交客戶端使用 `nosuid` 裝載的話，則 `qrsh`（及其相關指令）將無法運作。

- 問題 – 當您嘗試啟動分配製造 (distributed make)，qmake 會退出並產生以下的錯誤訊息。

```
qrsh_starter: executing child process qmake failed: No such
file or directory
```

- 可能的原因 – Sun Grid Engine，企業版 5.3 系統會在執行主機上啟動 qmake 的實例。如果 Sun Grid Engine，企業版 5.3 環境（特別是 PATH 變數）沒有在使用者的 shell 資源檔中設定的話 (.profile/.cshrc)，則該 qmake 呼叫會失敗。
- 可能的解決方法 – 請使用 -v 選項將 PATH 環境變數匯出至 qmake 作業。典型的 qmake 呼叫如下。

```
qmake -v PATH -cwd -pe make 2-10 --
```

- 問題 – 使用 qmake 公用程式時，您收到以下的錯誤訊息。

```
waiting for interactive job to be scheduled ...timeout (4 s)
expired while waiting on socket fd 5
```

```
Your "qrsh" request could not be scheduled, try again later.
```

- 可能的原因 – 呼叫 qmake 的 shell 裡的 ARCH 環境變數設定可能不正確。
- 可能的解決方法 – 將 ARCH 變數正確設定為符合叢集內可用主機的支援數值，或是在提交時間指定正確數值；例如，qmake -v ARCH=solaris64 ...

問題診斷

Sun Grid Engine，企業版 5.3 系統提供您數個協助您診斷問題的回報方法。以下各節簡述其使用方法。

沒有分派擱置的作業

有時候，擱置的作業很明顯的能夠執行，但卻沒有獲得分派。若要診斷出原因，Sun Grid Engine，企業版 5.3 會提供一組公用程式和選項，qstat -j <作業 id> 和 qalter -w v <作業 id>。

- qstat -j <作業 id>

啓動時，`qstat -j < 作業 id >` 會提供使用者上一次排程作業執行時，沒有分派某個特定作業原因清單。您可以啓動或停用此監視作業，因為它會在 `schedd` 常駐程式和 `qmaster` 之間造成不必要的通訊零碎空間（請參閱 `sched_conf(5)` 中 `schedd_job_info` 下面的資訊）。以下爲作業 `id` 爲 `242059` 的樣本輸出。

```
% qstat -j 242059
scheduling info: queue "fangorn.q" dropped because it is temporarily not available
queue "lolek.q" dropped because it is temporarily not available
queue "balrog.q" dropped because it is temporarily not available
queue "saruman.q" dropped because it is full
cannot run in queue "bilbur.q" because it is not contained in its hard
queue list (-q)
cannot run in queue "dwain.q" because it is not contained in its hard
queue list (-q)
has no permission for host "ori"
```

該資訊是直接由 `schedd` 常駐程式產生的，並將目前叢集的使用考慮在內。有時候這不是您真正感到興趣的事物；例如，如果所有的佇列插槽都被其他使用者的作業所佔用的話，則您感興趣的作業不會產生詳細的訊息。

■ `qalter -w v < 作業 id >`

此指令原則列出作業無法分派的原因。要達到這個目的，需執行乾燥 (`dry`) 排程執行。乾燥排程執行的特別之處是所有可耗損資源（也是插槽）都視爲能完全爲該作業所用。同樣地，所以的載入數值都因爲它們會變化所以被忽略。

在錯誤狀態 E 報告的作業或佇列

作業或佇列錯誤在 `qstat` 輸出中是以大寫的 `E` 表示。作業在 Sun Grid Engine，企業版 5.3 系統嘗試在佇列執行作業時進入錯誤狀態，但會特別是因爲作業的原因而故障。佇列在 Sun Grid Engine，企業版 5.3 系統嘗試在佇列執行作業時進入錯誤狀態，但會特別是因爲佇列的原因而故障。

Sun Grid Engine，企業版 5.3 系統提供使用者和管理員一套在發生作業執行錯誤時收集偵錯資訊的可能性。因爲佇列和作業錯誤狀態都是由失敗的作業執行所產生，所以偵錯可能性適用於兩種類型的錯誤狀態。

■ 使用者中斷郵件 (abort mail)

如果使用 `submit` 選項，`-m a` 提交作業，中斷郵件會傳送到以 `-M 使用者[@ 主機]` 選項指定的地址去。中斷郵件包含作業錯誤的偵錯資訊，並爲推薦使用者使用的資訊來源。

■ `qacct` 帳號 (accounting)

若沒有中斷郵件，則使用者可以從 Sun Grid Engine，企業版 5.3 系統的作業帳號功能執行 `qacct -j` 指令來收集作業錯誤的資訊。

- 管理員中斷郵件

管理員可藉由指定適當的電子郵件地址來命令作業執行問題的管理員郵件（請參閱 `sge_conf(5)` 中 `administrator_mail` 下面的資訊）。管理員郵件包含比使用者中斷郵件更詳盡的偵錯資訊，如果經常有作業執行錯誤的話，該郵件是個建議的方法。

- 訊息檔案

如果沒有管理員郵件，則應該先研究 `qmaster messages` 檔。您可以搜尋適當的作業 ID 來找出與某些作業相關的日誌紀錄 (`loggings`)。在預設的安裝中，`qmaster messages` 檔是 `$SGE_ROOT/default/spool/qmaster/messages`。

您有時可以從作業啟動的 `execd` 常駐程式訊息中找到額外的資訊。使用 `qacct -j <作業 id>` 找出作業啟動的主機，並在 `$SGE_ROOT/default/spool/<主機>/messages` 搜尋作業 id。