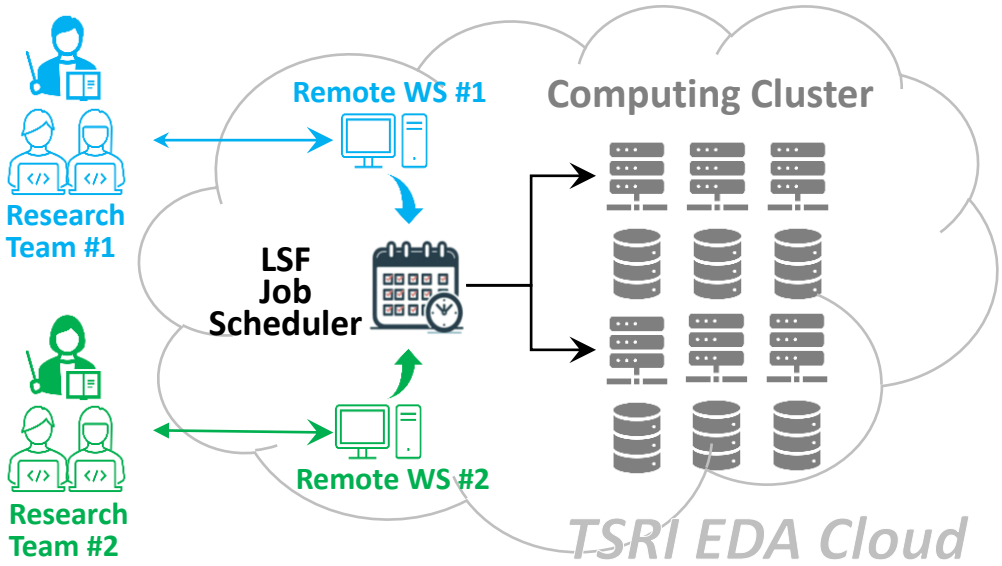
EDA 軟體啟動與執行 in TSRI EDA Cloud 2.0 使用說明

前言



使用運算叢集在 EDA Cloud 2.0 上執行 EDA軟體

TSRI EDA Cloud 2.0 是採用叢集式運算 (Cluster Computing) 的雲端運算環境，使用者執行 EDA 軟體的方式如上圖所示。首先由遠端裝置 (僅限已在 TSRI 註冊 IP 位址的裝置) 登入到 EDA Cloud 2.0 上的遠端工作站 (Remote WS)，遠端工作站的虛擬桌面 (Virtual Desktop) 會顯示在使用者的遠端裝置上，讓使用者能夠下指令執行 EDA 軟體。TSRI 為每一個研究團隊 (每一位教授) 配置一個遠端工作站，其僅用來作為 EDA Cloud 2.0 登入或備援之用，配置之運算資源極少，雖然亦可執行 EDA 軟體，但建議將 EDA 軟體送至有大量共享資源的運算叢集 (Computing Cluster) 執行。使用者在遠端工作站使用專用指令，將 EDA 軟體指令送至工作排程器 (Job Scheduler)，工作排程器會依據運算叢集的資源狀況，排定工作次序與時程。EDA Cloud 2.0 上的運算叢集由所有使用者共享，TSRI 會依據實際使用狀況動態調整使用資源限制，盡可能讓所有研究團隊公平使用運算資源。

若要直接在登入的遠端工作站執行 EDA 軟體，執行原始指令即可，但如前述原因，因配置資源極少，TSRI 不建議此作法。本文旨在說明如何透過工作排程器提交 EDA 軟體指令，在批次模式 (batch) 或互動模式 (interactive) 下執行 EDA 軟體以進行 IC 設計研發。由於 EDA 軟體通常具有多種執行介面 (例如純文字介面、純圖形介面、圖形與文字混合介面等)，其與工作排程器的兩種工作提交模式可以有多重的組合，即同一個 EDA 軟體可以根據使用者需求而有多種的執行方式，本文將針對常用的組合進行說明。若仍有使用上的問題，可先參考 Part 5: 線上查詢與參考網站 內文資訊自行找尋解答，若仍無法獲得解答，請洽本中心客服。以下是工作排程器的工

作提交模式和 EDA 軟體執行介面組合的簡要說明。

- 批次模式 (batch mode，背景執行，文字視窗被釋出且不再呈現軟體執行資訊)，可與下列 EDA 軟體的執行介面並用：

  - 純文字介面：執行過程完全不需再與使用者互動。
  - 純圖形介面：軟體開啟後，完全藉由圖形介面與使用者互動。

- 互動模式 (interactive mode，前景執行，文字視窗持續呈現軟體執行資訊)，可與幾乎所有 EDA 軟體的執行介面並用：

  - 可以看到即時訊息以利偵錯 (debug)。
  - 若 EDA 軟體執行需要文字與圖形混合介面，一定要使用互動模式。

請務必在每次 EDA Cloud 遠端工作站建立新桌面之後 (即非連線至既有桌面)，於任一文字視窗先執行一次 **xhost +** 指令，以允許遠端顯示。否則，EDA 軟體將會產生 cannot open display 等錯誤訊息並立即終止。故凡遭遇經由 LSF 執行的圖形介面無法顯示的狀況，皆可嘗試上述方法做故障排除。

另需提醒使用者，勿陷入執行緒 (thread) 越多越好的迷思。這是因為執行緒的加速效益會因為 EDA 軟體特性而有所差異：若超過臨界值，通常 CP 值會鈍化而增益趨緩。所以，某些 EDA 軟體 (例如 HSpice 或 Spectre) 甚至會主動縮減執行緒的數量。在 EDA Cloud 學員可以參考下列建議值操作，並觀察 log 檔中的實際執行緒使用數據做適度調整。至於細節或進階選項，可詳閱各 EDA 軟體使用手冊。

- 單一交談式工作 (如線路繪製、手工佈局、波形或偵錯工作等等) 以 1 ~ 2 個執行緒為宜，例如 Virtuoso 或 Custom Compiler。而其所衍生的模擬或驗證運算子工作，建議應另行設定並以批次模式再提交執行，方能藉由專屬的執行緒等運算資源，提高工作效率。
- 單一數位模擬工作僅需 1 個執行緒，而數位合成或時序分析建議使用 8 ~ 32 個執行緒。
- 單一類比模擬或佈局驗證相關工作，建議使用 8 ~ 48 個執行緒。如果設計案較為複雜，使用者可自行評估是否使用到上限值。
- 至於大型自動佈局、變異分析或蒙特卡羅分析，也可以拆分成數個子工作再配合上述的緒值，執行效率會更好，也比較容易被排程器分派執行。

雖然 EDA 軟體是透過工作排程器提交到運算叢集執行，但執行後產生的資料檔案，使用者可以在提交指令或指定的工作目錄中即時查看，不需其他額外的動作。

目錄

Part 1: 提交 EDA 軟體指令至運算叢集執行 (基礎篇)

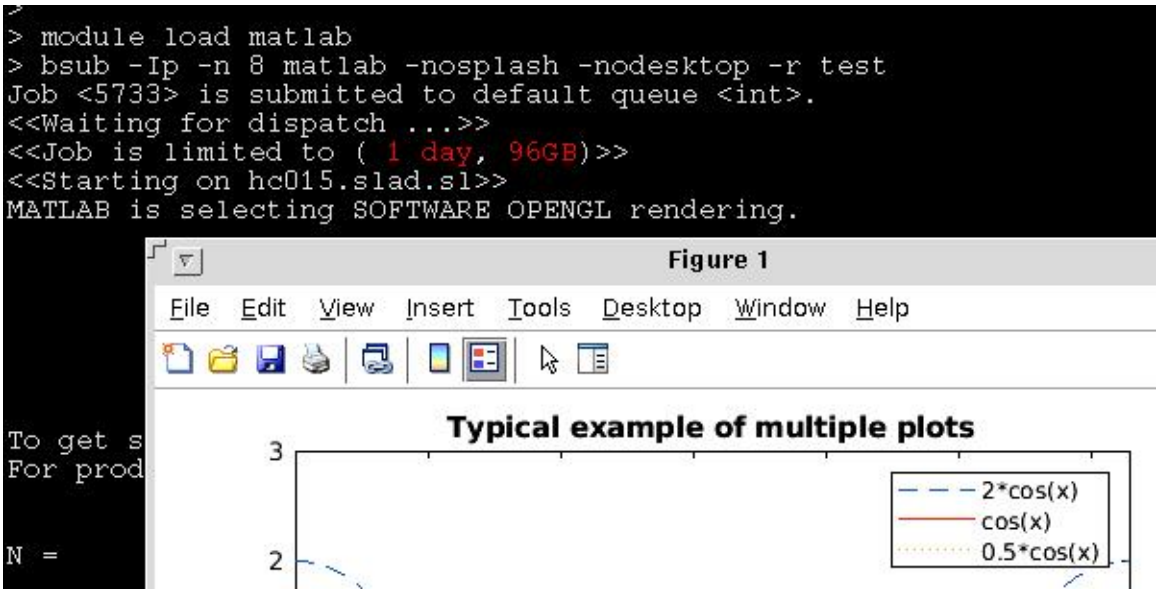
執行需要圖形與文字混用介面的 EDA 軟體

目前已知在開啟圖形介面後仍需保留文字介面的常用 EDA 軟體如下列清單 (但不限於此)。現以 Matlab 與 Design Compiler 的單工多緒舉例說明，餘請類推。執行 EDA 軟體前須先以 GNU Modules 設定軟體環境，GNU Modules 使用請參照 Part 3: 使用 GNU Modules 軟體管理 EDA 軟體啟動環境設定。

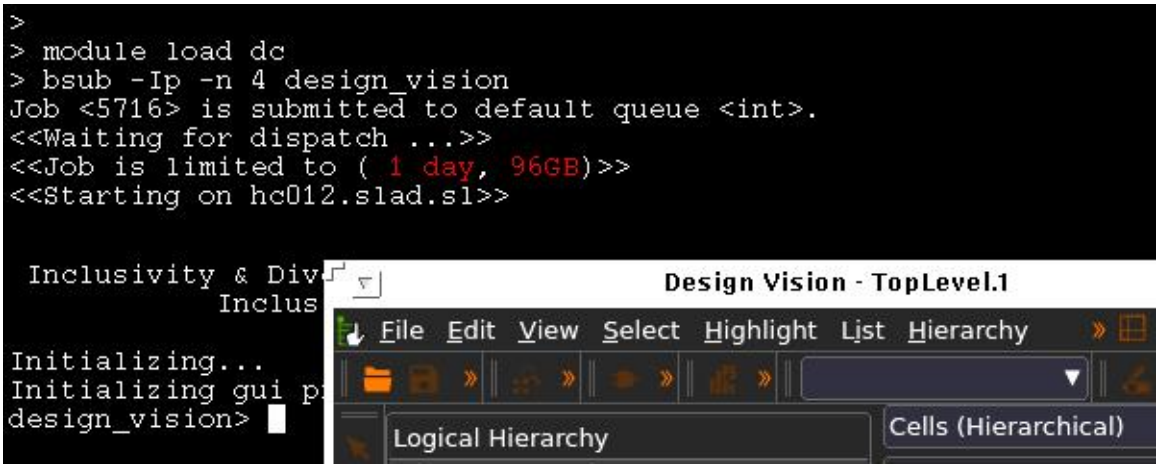
- Cadence ：Innovus
- Mathworks ：Matlab
- Synopsys ：Design Compiler (DC)、Primetime (PT) 與 IC Compiler II (ICC2)

1. 批次模式無法與 EDA 軟體圖形與文字混用介面並用。
2. 使用 bsub -Ip 指令以互動模式執行，命令檔 (script) 中不可含有 exit 指令。

> module load matlab
> bsub -Ip matlab
> bsub -Ip -n 8 matlab -nosplansh -nodesktop -r test (test.m 在運算之後才繪圖)



> module load dc (或 module load synthesis)
> bsub -Ip -n 4 dc_shell-t -f do_gui.tcl (在命令檔中啟動圖形介面)
> bsub -Ip -n 4 dc_shell-t -gui -f no_exit.tcl (以圖形介面執行命令檔)
> bsub -Ip -n 4 design_vision (直接開啟圖形介面)



執行純文字介面的 EDA 軟體

絕大部分的 EDA 軟體都支援以純文字的方式執行 (如下列清單)，可搭配批次模式於背景執行，有經驗的工程師通常會使用此種模式。設計初期或偵錯階段，可搭配互動模式以即時顯示 EDA 軟體執行訊息。以下列舉數例，餘請類推。

- 模擬器　：XCelium、VCS、Matlab、Spectre 或 HSpice
- 實作驗證軟體：DC、PT、Formality、ICC2、Innovus、Calibre、Star RC 或　ICV
- 檔案轉換器　：fsdb2vcd 或 strmin
- 文字工具　　：vim、emacs (文字模式)、nano 或 diff

1. 使用 bsub 指令以批次模式執行。需確保軟體能自我結束，或命令檔需包含 exit 指令。執行過程中可以使用 bsub -Ip bpeek [-f] *job-ID* 來查詢輸出訊息。例如：

> module load hspice
> bsub -q bat.l -n 4　hspice test.sp -mt 4　　(HSpice 預定值會自我結束)

> module load xcelium
> bsub　ncverilog　-f　vlog.f　　(ncverilog 預定值會自我結束)

> module load vcs verdi_vcs　　(同時載入兩個設定檔)
> vcs　-full64　-debug_access　-R　-f vlog.f　　(編譯後跑模擬並允許產生 fsdb 檔)

> module load dc;　　bsub -n 4　dc_shell-t -f script.tcl

```
>  module load pt;              bsub -n 4   pt_shell -f script.tcl
>  module load fm;              bsub -n 4   fm_shell -f script.tcl
>  module load calibre;         bsub -q bat.l -n 8   calibre   -drc   -hier   -turbo 8   drc.rule
>  module load matlab;          bsub -q bat.l -n 4   matlab -nosplansh -nodesktop -r script
```

2. 使用 bsub -Ip 指令以互動模式執行，命令檔有無 exit 指令皆可。

```
>  module load xcelium;         bsub -Ip   ncverilog   -f   vlog.f
>  module load vcs;             bsub -Ip   -full64   -debug_access   -R   -f vlog.f
>  module load hspice;          bsub -Ip -n 4   hspice   test.sp   -mt 4
>  module load calibre;         bsub -Ip -n 8   calibre   -drc   -hier   -turbo 8   drc.rule
```

```
> module load calibre
> bsub -Ip -n 8 calibre -drc -hier -turbo 8 drc.run
Job <5718> is submitted to default queue <int>.
<<Waiting for dispatch ...>>
<<Job is limited to ( 1 day, 96GB)>>
<<Starting on hc014.slad.sl>>
//   Calibre v2023.1_34.18     Thu Mar 2 10:21:38 PST 2023
//   Calibre Utility Library    v0-10_13-2017-1     Tue Nov 8 14:13:5
//   Litho Libraries v2023.1_34.18  Thu Mar 2 10:21:38 PST 2023
//
```

凡互動式文字介面軟體，如文字編輯器，僅能於互動模式執行，如：

```
>  bsub -Ip   vim   big-file.log
```

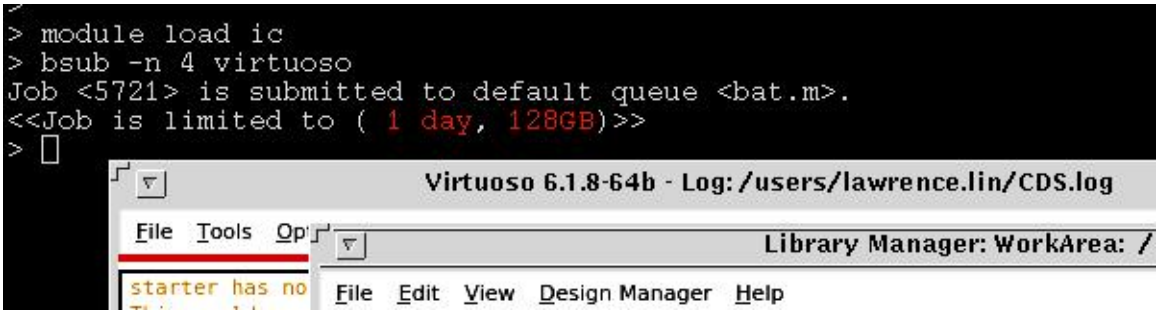
執行純圖形介面的 EDA 軟體

此處所言之純圖形介面為所有的互動皆於圖形介面上進行，完全不需透過原文字視窗。使用者可依自有需求或喜好，以批次或互動模式執行。以下列舉數例，餘請類推。

1. 使用 bsub 指令以批次模式執行 (原文字視窗會被釋放，可執行其他指令)：

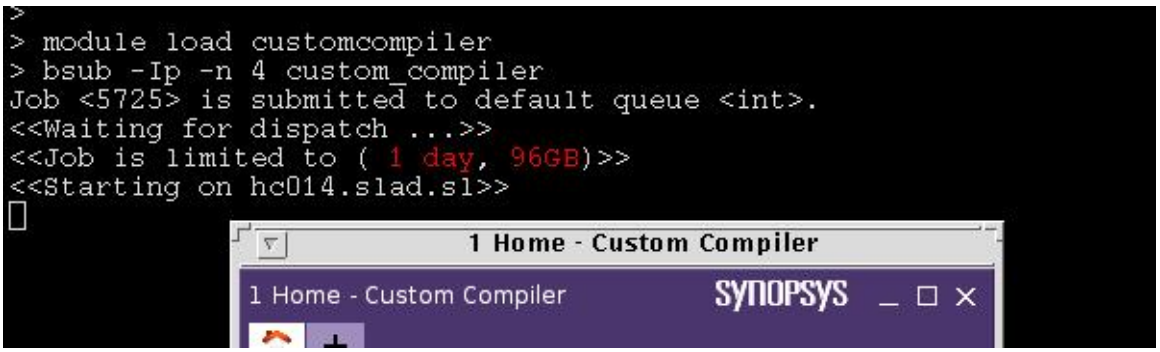
```
>  module load ads;             bsub ads
>  module load laker_oa;        bsub laker
>  module load customexplorer;  bsub cx
>  module load verdi_vcs;       bsub verdi
>  module load xcelium;         bsub simvision     (新版的 Verdi 已不支援 XCelium)
>  module load calibre;         bsub calibrerve
>  module load calibre;         bsub -n 4 calibre -gui          (勿加 -drc 或 -lvs 等)
>  module load customcompiler;  bsub -n 4 custom_compiler       (以 4 核執行外掛軟體)
```

> module load ic; bsub -n 4 virtuoso (以 4 核執行外掛軟體)

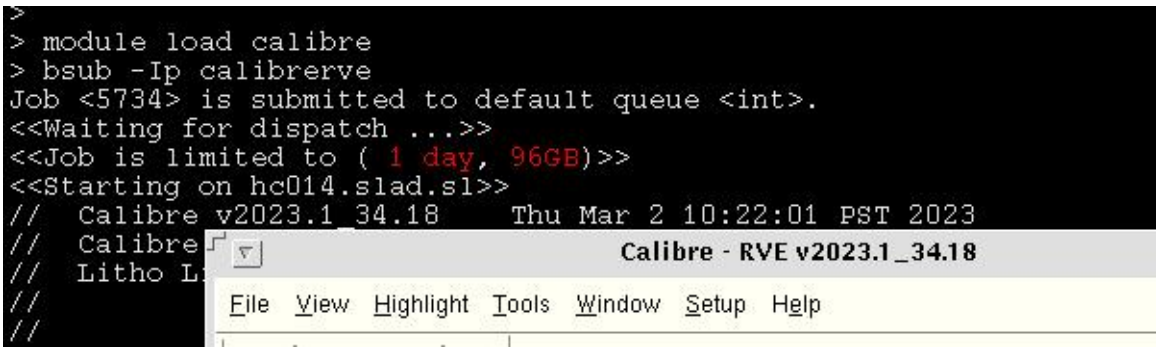


2. 使用 bsub -Ip 指令以互動模式執行 (原文字視窗不會被釋放，持續顯示軟體執行訊息)：

> module load ic; bsub -Ip -n 2 virtuoso
> module load laker_oa; bsub -Ip laker
> module load customcompiler; bsub -Ip -n 2 custom_compiler



> module load calibre; bsub -Ip calibrerve



執行未知介面形態的軟體

若無法確定 EDA 軟體的執行介面，請以互動模式提交 EDA 軟體先做測試。例如：

> module load freeware; bsub -Ip tkdiff (Tk 介面的 diff，比較兩個文字檔)

另有極少數的軟體會自動隱身於背景執行而無法與 LSF 配合使用，請於登入機台直接執行此類軟體即可。例如微軟的 Visual Studio Code 文字編輯器：

```
>  module load freeware;  code                    (code 會立即退至背景執行)
```

其他提交 EDA 軟體指令須知

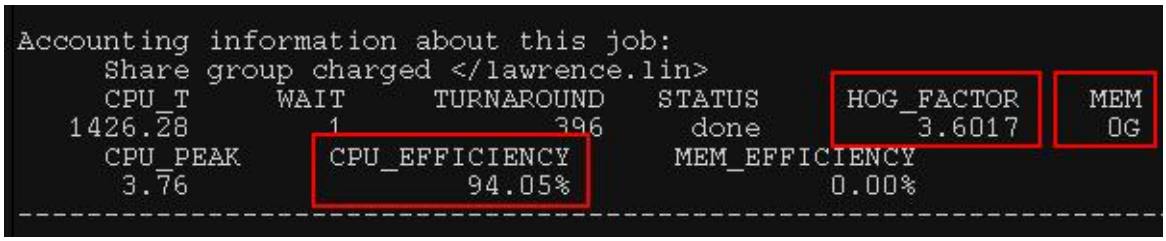
工作排程器在提交 EDA 軟體指令至運算叢集執行時，會送至使用者指定的佇列 (queue)。目前 EAD Cloud 2.0 提供的佇列 (queue) 如下表，請使用提交參數 -q *queue* 自行切換。

Queue Name (*: default)	Weight	MAX slot per user	MAX DRAM per job	Auto kill	使 用 時 機
int *	70	128	128 GB	2 days	互動模式
bat.m *	50	128	250 GB	1 day	批次模式，中小型工作
bat.l	40	128	250 GB	14 days	批次模式，大型工作

- 目前中心允許的記憶體使用上限值為 250GB，而最久的執行時間是 14 天。使用者可多利用 bat.l 佇列，並以多機多工多緒模式來提升運算效益，有效減少等待時間。
- EDA 軟體指令執行時間若超過 Auto Kill 欄位所標示的時限，指令執行會自動終止 (無任何事先警告，請自行查詢 log 或 rep 檔案)。
- Max slot per user 為可同時運用的工作槽數量上限。目前一個槽配置一個 CPU core。

提交 (submit) 工作的步驟與技巧：

- 務必先瞭解工作的特性，尤其是運算資源的需求、平行運算的功能以及與外掛 EDA 軟體的整合等等。
- 依照上述的需求選擇合適的佇列 (queue)。
- 依照上述的需求選擇合適的排程器提交參數，亦可開發自動化程式輔助。
- 善用自動產生的 log 與 rep 報告檔，以便偵錯或改善執行效率。
  - rep 報告檔內含各種運算資源使用摘要，可做為下次提交參數的參考。
  - 執行 bacct -l *job-ID* 可以查詢已完成工作的運算資源使用分析。其中 HOG_FACTOR 值應越接近所申請的 CPU core 數越好；而 CPU_EFFICIENCY 值應越接近 100% 越好。否則，可能代表軟體的平行運算效益不高，或所申請的 core 過量而浪費了。下圖為使用 4 core 並取得良好平行運算效益的範例。



- 請自行刪除過時的報告檔。

提交 (submit) EDA 軟體指令的範例 (最常使用的 bsub 參數是 -Ip，-n 與 -q)：

- (懶人法) 為 Virtuoso 申請兩個 core，以便稍後給各種外掛軟體使用 (但切勿給予過多的核心數，以免大部分的時間都處於閒置的狀態而浪費了)。詳細說明請見進階篇，參考範例檔案置放於 /lsf/examples/lsf_virtuoso 目錄下。

  > module load spectre calibre ic
  > bsub -n 2 virtuoso

- (懶人法) 為 Custom Compiler 申請兩個 core，以便稍後給各種外掛軟體使用。

  > module load hspice primewave icv starrcxt customcompiler
  > bsub -n 2 custom_compiler

- 申請同一台主機的 4 core 給 4 緒使用。而若程式執行超過 12 小時，則自動刪除。

  > module load hspice
  > bsub -q bat.l -n 4 -W 12:00 -- hspice test.sp -mt 4

- 申請同一台主機的 8 core 給使用 8 緒的 DC 自動化程式，且該主機必須仍有 128GB 可用記憶題。而若程式執行超過 32 小時，則自動刪除。

  > module load dc
  > bsub -q bat.l -n 8 -R “select[mem>128]” -W 32:00 “dc_script.pl | tee dc.log”

- 目前波形顯示軟體與數位模擬器軟體皆未支援多緒，所以提交參數相對簡單。

  > module load xcelium
  > bsub simvision
  > bsub ncverilog -f vlog.f

Part 2: 提交 EDA 軟體指令至運算叢集執行 (進階篇)

凡欲從 EDA 軟體 (主工作，main/master job) 再啟動外掛第三方 EDA 軟體 (子工作，slave job)，皆建議使用者 “自行” 設定 EDA 軟體與排程器的連結，以便主工作能經由排程器提交子工作而分派執行。藉此，子工作將會被排程器視為獨立的工作，而給予專屬的運算資源。

- 除非模擬較小 (少於 250 個 device)，因為使用排程器會有相當的 overhead。
- 一般訓練課程也不需要整合 (LAB 較為簡易)，僅有在 EDA Could 2.0 跑專案才需要。
- 通常為類比或後段軟體，如 Virtuoso、Custom Compiler、Primetime 或 Calibre。

若未先整合，排程器會將主、子工作視為一體，而僅給予一份運算資源。

例如，若未整合而在 Virtuoso 中執行包含 4 個緒的 Spectre，則 5 個運算將會全擠在下方黃色主機的某顆 CPU core 裏，僅能共享 100% 的 CPU 時間，而非所期望的 500%。

現提供兩種解決方法供使用者參考，請自行判斷與利用：

1. 在提交主工作時，先預估包含主、子工作的 CPU core 使用上限量，再以此量設定主工作的提交參數 -n *N* 值。而子工作則不需任何額外設定 (或僅須啟動多緒模式做加速運算)。這種方法的優點是簡單快速，而缺點是只能使用同一台主機以及資源使用率較低。例如：

   - 執行 bsub -n 4 virtuoso 後，所有外掛軟體將 (只) 能分享所申請的 4 個 core。

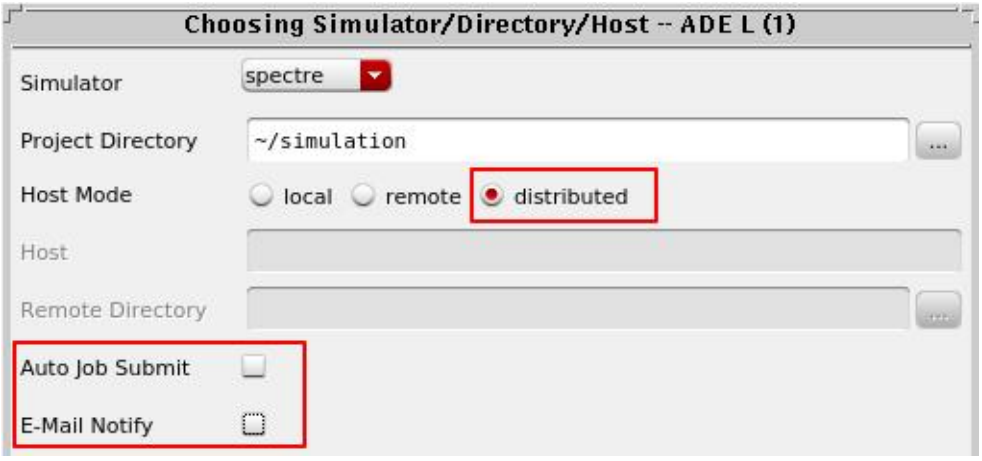
2. 另一種方法即如前言所提，是經由 EDA 軟體對於排程器的整合支援，在設定參數之後，由主工作主動向排程器提交子工作執行。這種方法的優點是可以藉由更高的平行運算層級再提升整體的運算效益，而缺點是需花時間瞭解方法並做設定。

現舉例數種外掛軟體與排程器的整合方法，方便使用者有通盤的瞭解。而使用者仍需參考所使用的軟體版本的使用手冊，才能發覺版本的差異而做出正確的設定。

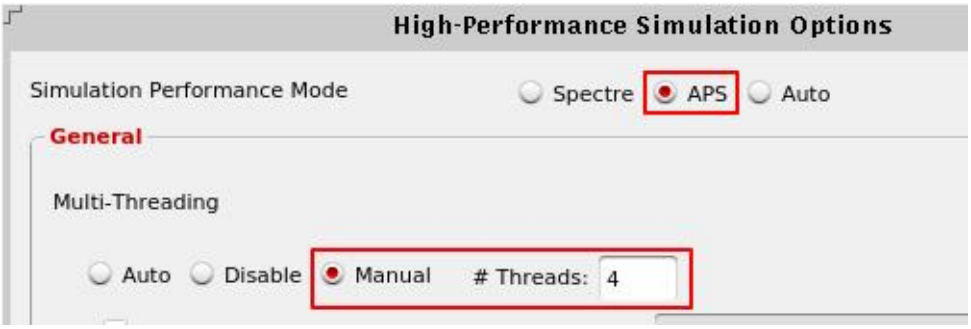
於 Virtuoso ADE L 執行 Spectre 的設定

- 所需之 LBS_BASE_SYSTEM 變數已於登入時自動啟用，使用者不須再設定。
- 請執行 bsub virtuoso 後，開啟設計單元的 ADE L 介面。
- ADE L 可設定 Spectre 以多緒方式加速運算，適合單一的模擬運算工作。

➢ Setup ➔ Simulator/Director/Host
✧ 於 Host Mode 中選擇 distributed (勿啟用 Auto Job Submit 或 email)。

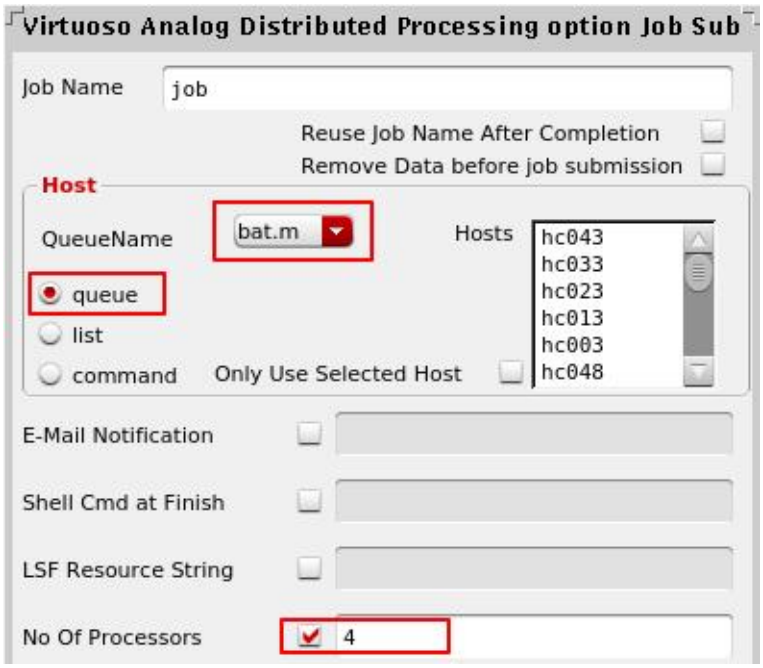


➢ Setup ➔ High-Performance Simulation Options
✧ 選擇 APS 後於 Multi-Threading 選擇 Manual，並於 Threads 輸入數量。



➢ Setup ➔ Environment
✧ 點選 Spectre 的 Run with 64-bit Binary。

- Simulation ➔ Netlist and Run
  - ✧ 在 Host 區域中點選 queue，並於 QueueName 中選擇 bat.m 或 bat.l。
  - ✧ 於 No. Of Processors 中設定所需的數量 (即上方的緒數量 *N*)。



- 其餘步驟不變。

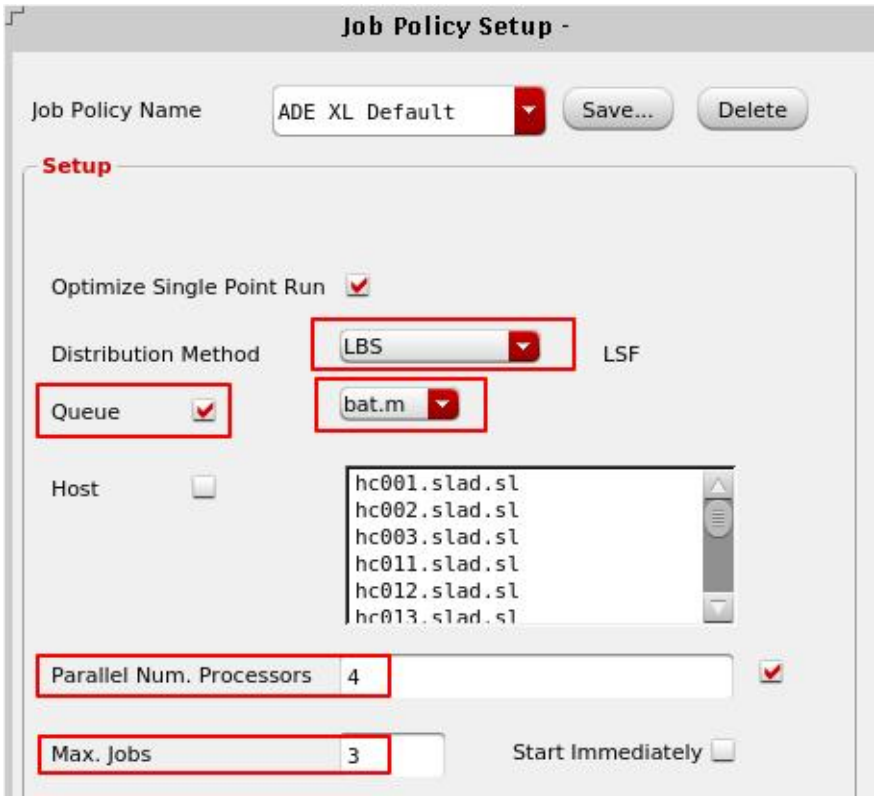
  - Tools ➔ Job Monitor 可查詢 job ID 與執行過程簡介。
  - Simulation ➔ Output Log 可查詢 Spectre 的 log。

於 Virtuoso ADE XL 執行 Spectre 的設定

- 所需之 LBS_BASE_SYSTEM 變數已於登入時自動啟用，使用者不須再設定。
- 請執行 bsub virtuoso 後，開啟設計單元的 ADE XL 介面。
- ADE XL 可設定 Spectre 以多機多工多緒方式加速工作，適合如 Monte Carlo。

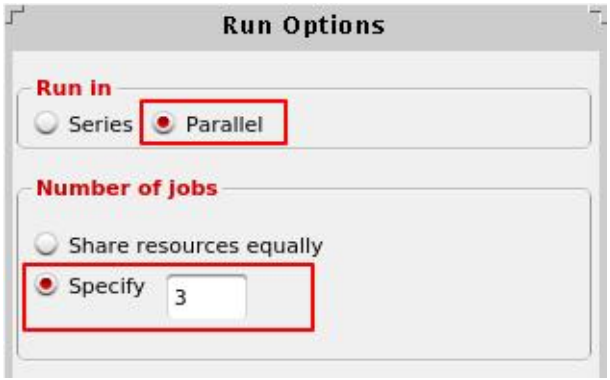
  - Options ➔ Job Setup

    - ✧ 於 Distribution Method 中選擇 LBS，之後會帶出 LSF 與相關資訊。
    - ✧ 點選 Queue 的選項，並選擇 bat.m 或 bat.l (勿選 “int”)。
    - ✧ 輸入 Parallel Num. Processors 的數量 *N* (即每個子工作所需的緒數)。
    - ✧ 輸入 Max. Jobs 的數量 *M* (衍生的子工作數量，將經由 LSF 再分派至其他機台執行)，並取消 Start Immediately。



- Options ➔ Run Options

  - 點選 Run In 中的 Parallel 選項。
  - 輸入 Number of jobs ➔ Specify 的數量 *M* (即所需的子工作數量)。



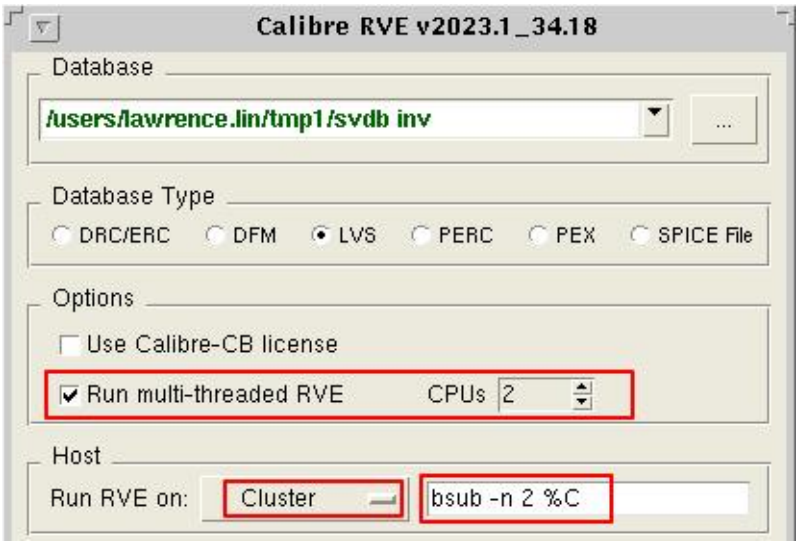
- 其餘步驟不變

  - Tools ➔ Job Monitor 可查詢 job ID 與執行過程簡介
  - 或於 Virtuoso console 中尋找 ADE XL 的 log 訊息

於 Virtuoso 執行 Calibre 的設定

- 需事先載入 Calibre 的軟體環境以便外掛 (即 module load calibre)。
- 請執行 bsub virtuoso 後，帶出設計單元的 layout view。
- 現以兩核心執行 RVE 舉例，請點選上方命令列 Calibre ➔ Run RVE 打開圖形介面：

  - 在 Options 中點選 Run multi-threaded RVE，並點選右側 CPUs 的數量為 2。
  - 於 Host 中的 Run RVE on 選項中點選 Cluster，再於右側輸入 bsub -n 2 %C。
  - 此方法為啓動獨立運作的 RVE。若需與當下的 Layout Editor 連線以顯示錯誤位置，仍需由如下方 mmDRC 範例中圖形介面的 “Start RVE” 功能啓動。

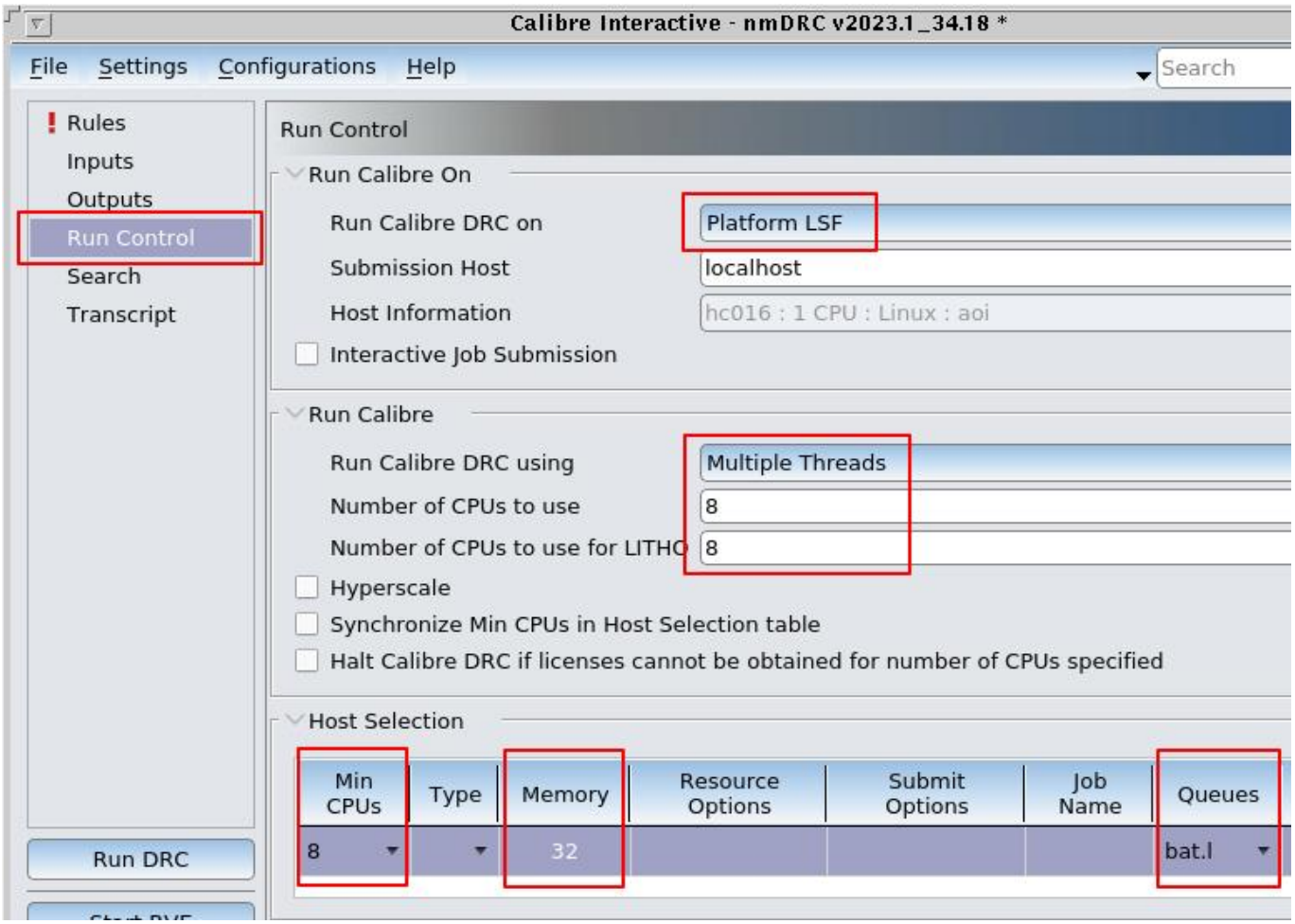


- 另以 mmDRC 舉例，請開啟 layout view 後點選 Calibre ➔ Run mmDRC 以打開圖形介面。之後點選左側的 Run Control，再於右側的 Run Calibre DRC on 選項中選擇 Platform LSF，即可看到設定畫面。而執行後將自動啟動 RVE。

  - 取消 Interactive Job Submission。
  - 選擇 Multiple Threads，並設定適當的緒數量 (建議量為 4 ~ 8)。
  - 續設定 Host Selection。應對應上方的緒數量設定所需的 CPU 與記憶體數量 (GB，勿輸入單位)，並建議選擇 bat.l 佇列。
  - 因需使用倍數的記憶體，故不建議在大型設計上啟用 -hyper 參數。若工作因記憶體使用超量而被 LSF 刪除，請檢視並停用此參數後再重跑。

在 Virtuoso 執行 Calibre 所看到的圖形介面，與單獨執行 Calibre 是相同的。所以使用者可以比照上方關於 mmDRC 的說明，應用在單獨執行 Calibre 的工作上。單獨執行 Calibre 的指令範例如下 (因稍後主工作會經由排程軟體提交 DRC 等子工作，故僅需一個 CPU core)：

> module load calibre; bsub -q bat.l calibre -gui



於 Virtuoso 執行 IC Validator 的設定

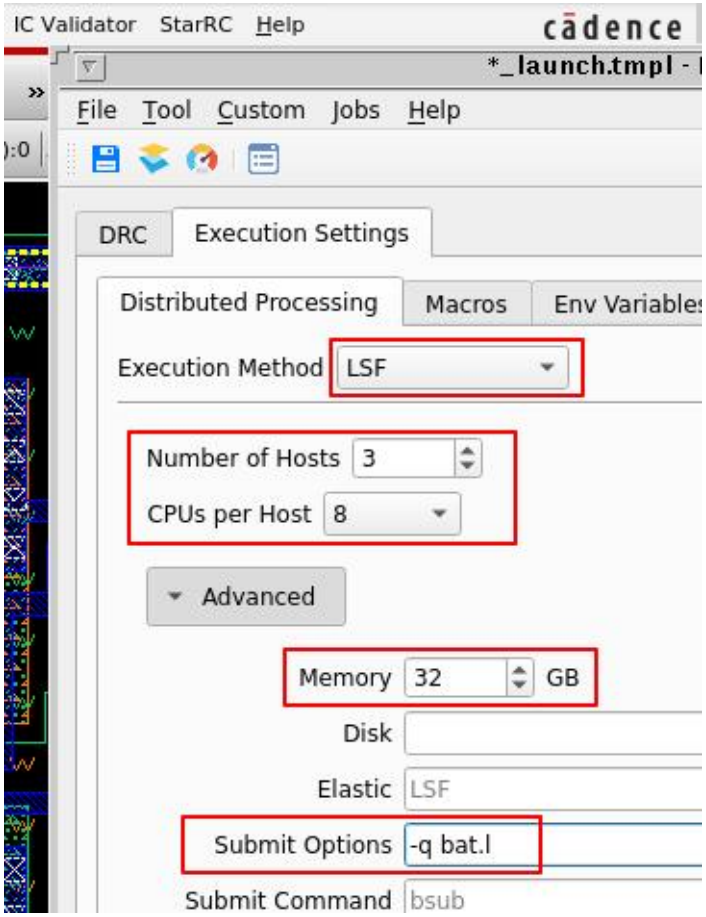
- 需事先載入 ICV 的軟體環境以便外掛 (即 module load icv)。
- 請執行 bsub virtuoso 後，帶出設計單元的 layout view。
- 以 DRC 舉例，請選擇 IC Validator ➔ DRC 打開圖形介面。之後點選 Execution Settings 即可看到設定畫面。

  - ➢ 在 Execution Method 選擇 LSF。

    - ✧ 在 Number of Hosts 輸入所需的子工作數量。
    - ✧ 在 CPU per Host 輸入每個子工作所需的緒數量。
    - ✧ 請注意：上述設定所產生的工作槽總量 (即兩數的乘積) 切勿超過使用者的上限，否則工作將永遠不會被分派。

  - ➢ 點選並展開 Advanced。

- ✧ 在 Memory 輸入每個子工作所需的記憶體預估量。建議可由 rep 檔回推。
- ✧ 在 Submit Options 中可切換至 bat.l 佇列，以取得更多的運算資源。



於 Virtuoso 執行 Star RC 的設定

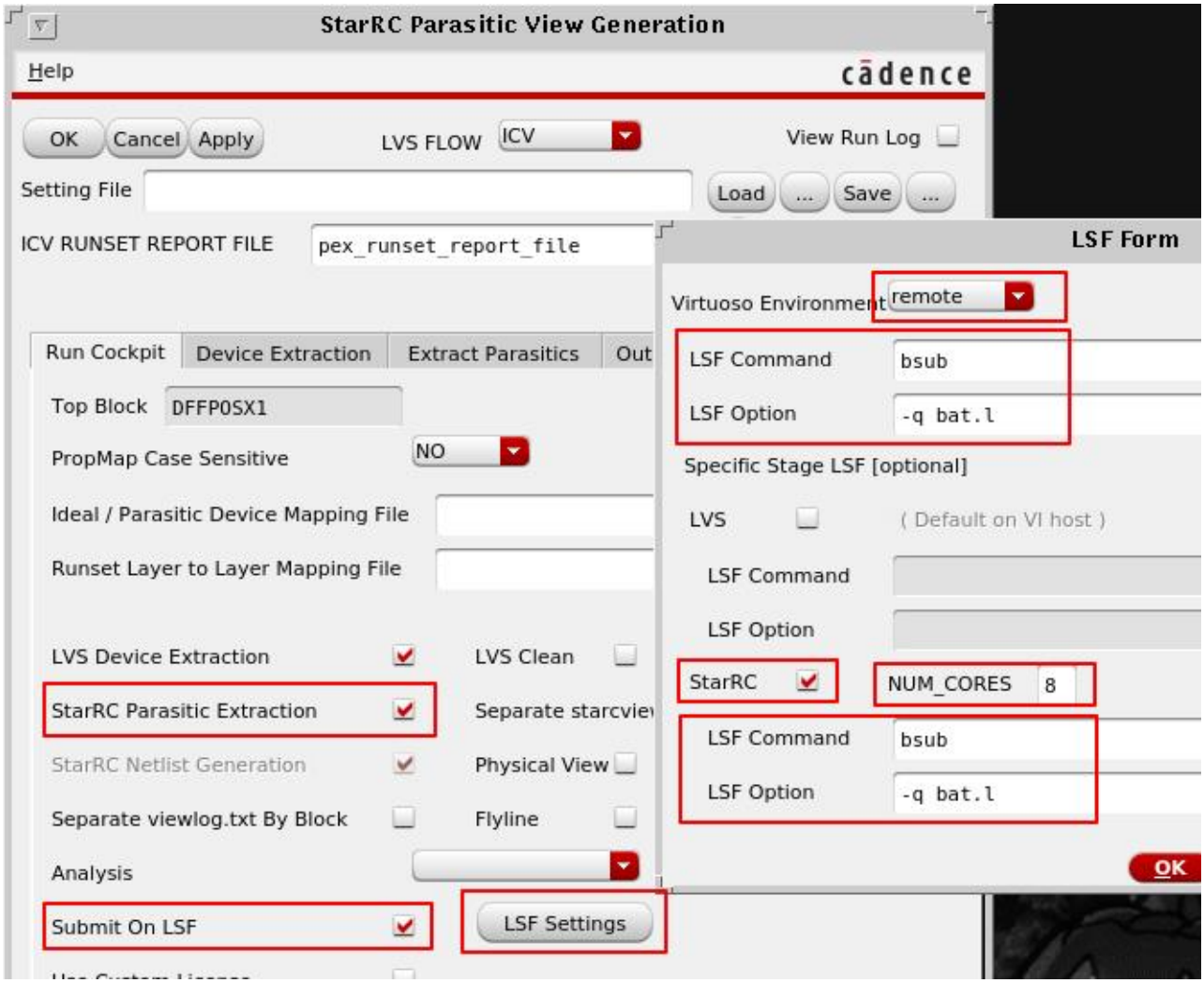
- 需事先載入 Star RC 的軟體環境以便外掛 (即 module load starrcxt)。
- 請執行 bsub virtuoso 後，帶出設計單元的 layout view。
- 以 PGC 舉例。請選擇 StarRC ➔ Parasitic Generation Cockpit 打開圖形介面。之後點選 Run Cockpit 即可看到設定畫面。

  - ➢ 點選 StarRC Parasitic Extraction。
  - ➢ 點選 Submit On LSF 之後 LSF Settings 的按鈕才會出現。
  - ➢ 點擊 LSF Settings 打開 LSF Form 做設定。
  - ➢ 在 LSF Form 中的 Virtuoso Environment 選擇 remote。

    - ✧ 於的 LSF Command 中輸入提交指令 bsub。
    - ✧ 於 LSF Option 中輸入提交指令參數，如 -q bat.l，或自行調整。

➢ 在 LSF Form 中點選 StarRC，允許設定 StarRC 專屬的提交方法。

✧ 於 NUM_CORES 中指定做 partition 的數量，亦即子工作的數量，如 8。
✧ 於其下的 LSF Command 中輸入提交指令 bsub。
✧ 於 LSF Option 中輸入提交指令參數，建議使用 -q bat.l 即可。而依照使用者手冊，每個子工作並沒有多緒的功能，故請勿加入 -n 參數。



關閉 Virtuoso 數分鐘後，仍可於排程器看見該工作的問題

這是因為 ADE 所啟動的常駐程式 (cdsfrb_lsf，與排程器溝通之用) 會延遲結束所造成的現象。據 Cadence 文件顯示，應於 30 分鐘後可自動結束。但若使用者想逕行移除 (因為會佔據工作槽)，可先使用 bjobs 取得該 Virtuoso 的 job ID 之後，使用 *bkill job-ID* 刪除即可。

Cadence Spectre 的平行運算設定

Spectre 內含三種支援 LSF 的平行運算工作模式 (~~baseline~~，APS，X 與 XPS)，而且功能強大，請閱讀使用者手冊後選擇最適合的工作模式。而 Spectre 對於 LSF 的支援有兩個與其他 EDA 軟體非常不一樣的地方，可能會令使用者困惑，故在此一併解說：

1. Spectre 的參數 +mt=lsf 會自動參考 LSF 的提交參數 -n N 值來設定子工作的緒數量。
2. Spectre 的參數 +mp 會使用 fork() 於本機衍生第一個子工作，並非都經由 LSF 提交。

- 現以 APS 單機單工 4 緒機制為例 (+aps 加 +mt，單一模擬工作適用僅此模式)：

  > bsub -q bat.l -n 4 spectre +aps +mt=lsf test.scs…

- 另以 APS 多機 3 工 4 緒機制為例 (+aps 加 +mp 加 +mt，適用變異或蒙特卡羅)

  > bsub -q bat.l -n 4 spectre +aps +mp=3 +mt=lsf test.scs…

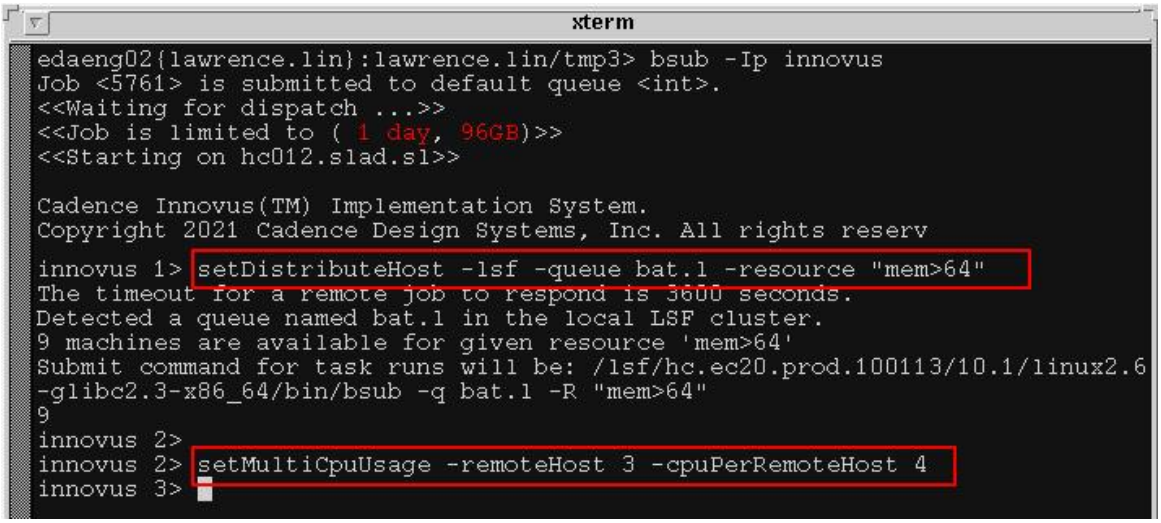
Cadence Innovus 的平行運算設定

- 圖形介面請於 Multiple CPU Processing Form 中設定。
- 以本機多緒模式 (local mode) 的 8 個緒舉例，需先輸入下列兩行，再啟動運算：

  ```
  setDistributeHost -local
  setMultiCpuUsage -localCpu 8
  ```

- 而遠端多工模式 (remote mode) 的範例如下 (啟動各具有 4 緒的 3 個子工作)：

  ```
  setDistributeHost -lsf -queue bat.l -resource “mem>64”
  setMultiCpuUsage -remoteHost 3 -cpuPerRemoteHost 4
  ```

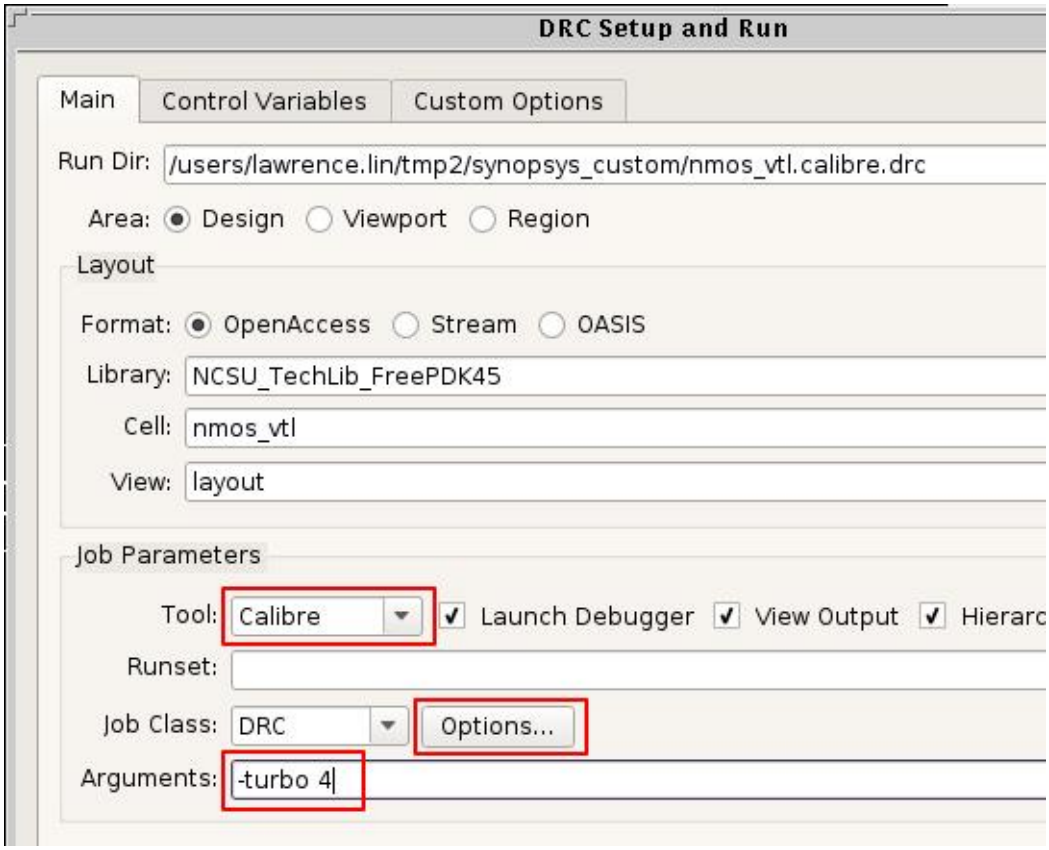


- Innovus 允許同時混用 local 與 remote 模式，故需特別留意主工作的提交參數：
  - 凡使用 local 模式，皆需為主工作申請 (*N*+1) 個核心。如：
    - bsub -q bat.l -n 5 innovus…　　(申請 5 核心，其中 4 個給子工作)
  - 而純 remote 模式是經由排程器分派子工作，所以主工作不需多核心。如：
    - bsub -q bat.l innovus…　　(只需一個核心給主工作即可)

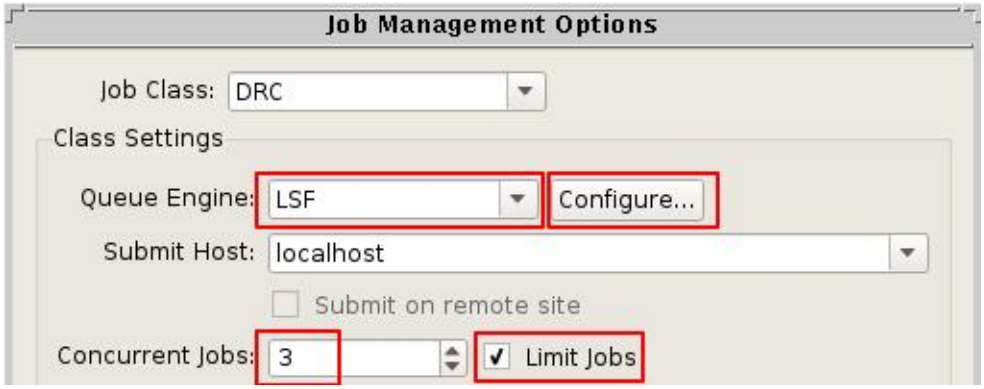
於 Custom Compiler 執行外掛軟體的設定

Custom Compiler 對於執行外掛軟體的設定較為簡單，而且使用者介面也較為一致。現以執行多機 3 工 4 緒的 Calibre DRC 工作舉例，餘如 HSpice 或 IC Validator 等請類推。

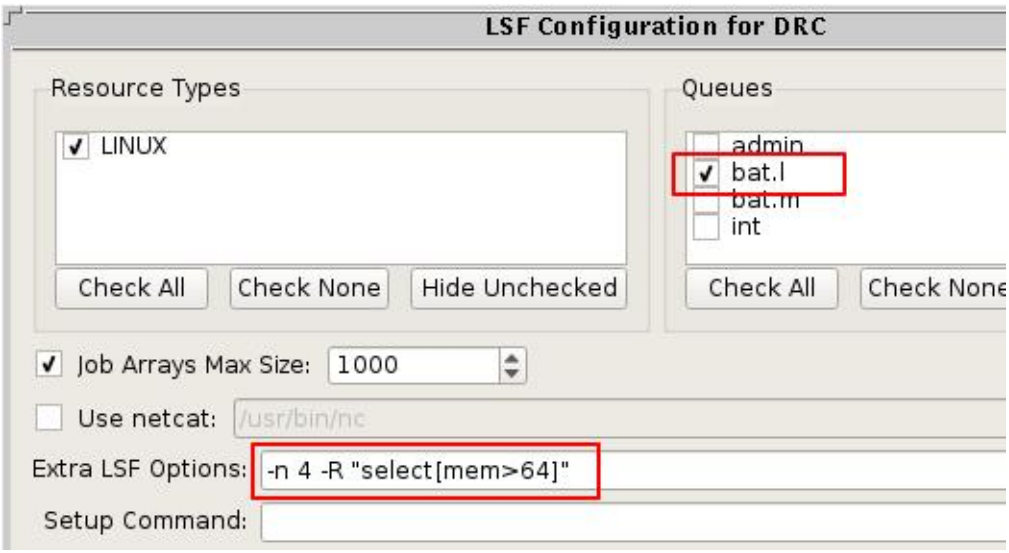
- 須先載入外掛軟體的軟體環境，再執行 Custom Compiler。
  - module load calibre customcompiler
  - bsub custom_compiler
- 於 layout view 功能表選擇 Verification ➔ DRC ➔ Setup and Run…帶出對話視窗。
  - 點擊於 DRC Setup and Run 對話視窗點的 Main 對話頁：



- ✧ 於 Tool 選擇 Calibre。
- ✧ 於 Arguments 輸入所需的 -turbo 數量，此例為 4 緒。若使用 ICV ，則需輸入 ICV 的對應參數。
- ✧ 點擊於 Job Class 右方的 Options，以帶出對話視窗。



- ▪ 先在 Queue Engine 中選擇 LSF。
- ▪ 點擊於 Queue Engine 右方的 Configure… 以設定 LSF。建議值如下，並請自行調整，以配合 Calibre 的 -turbo 數量與設計佈局資料量的大小。以此例而言，Calibre 將會使用到 4x3=12 個 core 來加速運算。



  - 再於 Concurrent Jobs 中設定所需的子工作數量 (此例為 3)，並點選 Limit Jobs 以免佔用使用者所有的工作槽。

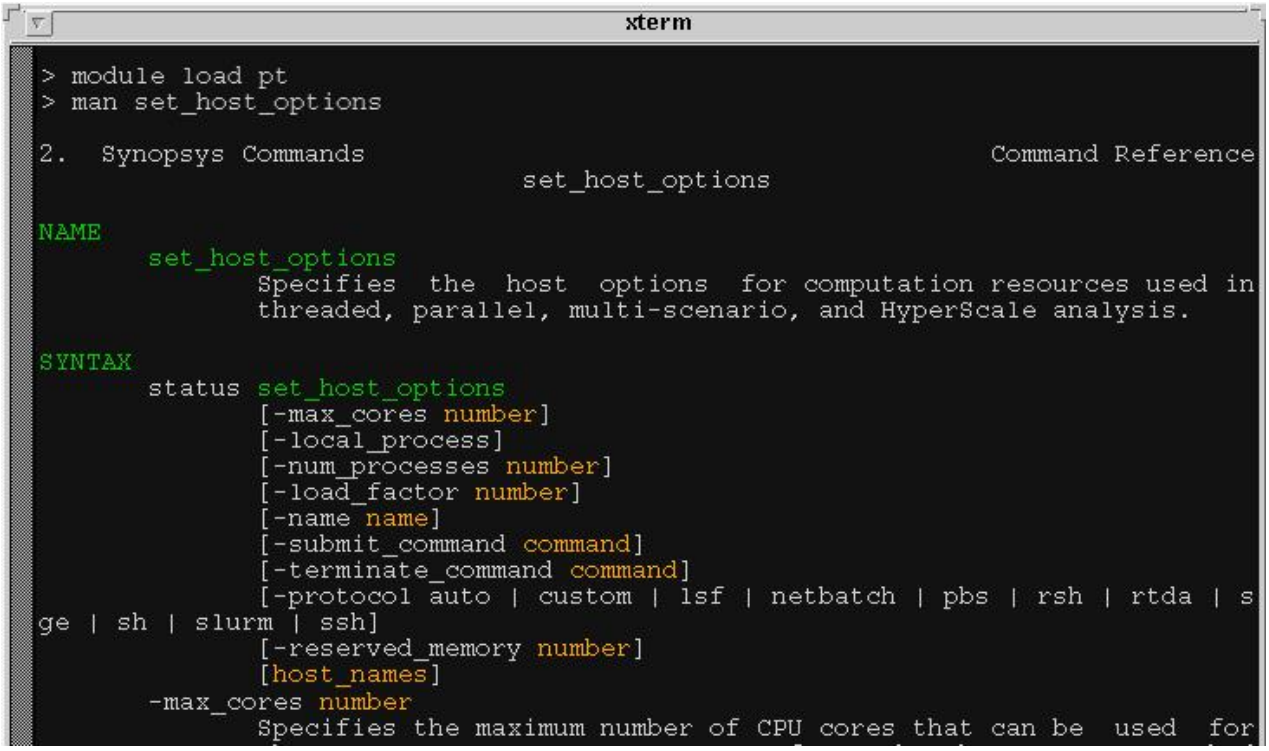
- 查看 DRC 執行過程的方法不變。而使用者也可經由排程器的 rep 檔瞭解運算資源的使用狀況。

Synopsys 軟體慣用的平行運算設定

大部分的 Synopsys 軟體都採用 set_host_options 函數來設定平行運算的機制，但每個軟體所提供的參數卻不盡相同，故請務必參考使用手冊來設定這個函數。例如：

- Design Compiler 僅支援 local mode，即僅有 -max_cores 參數。
- Primetime 的 -protocol 參數在 ICC2 變成 -submit_protocol 參數。
- 雖然 Synopsys 的軟體皆不允許 local mode 與 distributed mode 並存，但上節所言之 Innovus 對於主工作提交參數 -n 的設定方法，仍可應用於 Synopsys 軟體。請參考並套用。

在載入 EDA 軟體環境後，可執行 man set_host_options 取得線上使用說明。但請務必留意 MANPATH 的前後次序，以免抓到其他軟體的 man page 而造成設定錯誤。例如：



Synopsys HSpice 的平行運算設定

HSpice 提供三種平行運算機制 (但皆非經由 set_host_options 函數)，現說明如下。

而為了縮減模擬檔案內容進而節省組員 (組群) 的硬碟空間，請務必於 HSpice 輸入檔中先啟用 .option post=2 probe print 功能，再利用 .probe 或 .print 設定所需要的訊號或資訊即可。

- 本機單工多緒機制 (-mt，單一模擬工作適用僅此模式)。
  - 在此模式下 HSpice 僅利用多緒做加速運算，故僅需 *N* 個 CPU core。
  - 指令範例： bsub -n *N* hspice test.sp -mt *N*…
- 本機多工多緒機制 (-mp 加 -mt，適用變異 (alter) 或蒙特卡羅，overhead 較重)。
  - 主工作自行於本機衍生與管理多個多緒子工作，共需 *NxM* 個 CPU core。
  - 指令範例：

    > bsub -n *NxM* hspice test.sp **-mp *M*** -dpconfig mp.cfg -mt *N*…

    而 mp.cfg 檔案內容如下，請自行建立檔案：

    1|localhost|-1|/tmp|SH|sh

- 多機多工多緒機制 (-dp 加 -mt，適用變異 (alter) 或蒙特卡羅，overhead 最重)。

  - 主工作利用排程器提交各包含 *N* 個緒的 *M* 個子工作來加速運算。而因為此模式可以使用多台主機，故適用於較大型的平行運算工作，如 Monte Carlo。且在此模式中，主工作僅扮演 job coordinator 的角色，故僅需一個 CPU core。
  - 指令範例：

    > bsub hspice test.sp **-dp *M*** -dpconfig dp.cfg -mt *N*…

    而 dp.cfg 檔案內容如下，請設定緒數 -n 以與 HSpice 的 -mt 參數相等)：

    1||-1|/tmp|LSF|bsub -q bat.l -n *N*

HSpice 三種平行運算機制的效益提升案例。

- 提升單一運算工作效能的案例：

平行運算機制	使用的 core (slot) 數量	執 行 時 間	平行運算的效益
無	1	9:03	1 X
-mt 2	2	5:10	1.75 X
-mt 4	4	3:34	2.53 X

- 提升多個運算工作效能的案例 (此例執行三個 ALTER 子工作)：

平行運算機制	使用的 core (slot) 數量	執 行 時 間	平行運算的效益
無	1	27:02	1 X
-mp 3 -mt 4	12 (同機台)	3:52	7.04 X
-dp 3 -mt 4	12 (跨機台)	3:54	6.98 X

Part 3: 使用 GNU Modules 軟體管理 EDA 軟體啟動環境設定

在 EDA Cloud 2.0 環境中，GNU Modules 與以 csh 為基礎的 source file 會同時並存，以確保既有教材的運作。使用者可自行選用。而在登入後，系統會自動載入 GNU Modules 的環境，使用者立即可以操作使用。

GNU Modules 的優點

- 支援查詢、載入、轉換或卸下軟體的環境設定。
- 支援設定 module 之間的相關或互斥關係，防止環境設定的衝突。
- 支援各種 UNIX shell (如 bash、csh、ksh、sh、tcsh 與 zsh)。
  - 在各種 shell 中的指令與操作方法皆相同。
- 支援各種 script language (如 Lisp、Perl、Python、Ruby 與 TCL)。
  - 直接將軟體環境載入至 run-time shell 中，既方便操作又可提升執行效率。
- 提供與業界相仿的設計環境，日後學員可與業界無縫接軌。
  - GNU Modules 廣為學界與業界所使用，學員入業界可以直接學以致用。
  - 如國網中心的超級電腦，以及國內外的晶片設計或製造公司。

GNU Modules 的常用指令

GNU Modules 指令	用 途 與 說 明
module avail [*tool*]	列出所有 (或包含特定字) 的軟體與版本
module disp *tool*[/version]	顯示預設 (或指定) 版本軟體的設定內容
module help *tool*[/version]	顯示預設 (或指定) 版本軟體的須知事項
module load *tool*[/version]	載入預設 (或指定) 版本軟體的設定，載入後即可使用軟體
module switch *tool*[/version]	切換至預設 (或指定) 版本軟體的設定
module unload *tool*	卸下軟體的設定
module list	列出目前已載入的軟體設定
module purge	卸下所有已載入的軟體設定 (但以 sticky 模式載入者除外)
module path *tool*[/version]	顯示預設 (或指定) 版本軟體的 module 檔案位置

GNU Modules 指令操作範例

```
>
> module avail
---------------------------------- /cad/utility/modulefiles/Agilent ----------------------------------
ads/2020u2  empro/2017  gg/2017

---------------------------------- /cad/utility/modulefiles/Cadence ----------------------------------
assura/231_04.17.001  ic/23.10.030       integrand/63.10.000  pvs/22.22.000       ssv/22.12.000
assura/618_04.16.113  ic/23.10.060       jasper/2023.06       quantus/21.20.000   stratus/23.01.002
conformal/23.10.200   icadvm/18.10.130   liberate/23.10.084   quantus/22.11.000   verisium/23.03.001
genus/21.17.000       icadvm/20.10.170   mmsim/15.10.801      spectre/20.10.155   vmanager/23.03.003
ic/06.18.180          incisiv/15.20.086  modus/22.11.000      spectre/21.10.389   xcelium/23.03.004
ic/06.18.320          innovus/21.17.000  pegasus/22.23.000    spectre/23.10.275
ic/06.18.350          innovus/21.18.000  pegasus/23.12.000    ssv/21.13.000

---------------------------------- /cad/utility/modulefiles/Mathworks ----------------------------------
matlab/R2022b

------------------------------ /cad/utility/modulefiles/Siemens-Mentor -------------------------------
calibre/aoi/2023.1_34.18  modelsim/2023.1     questavip/2023.1_1  tessent/2023.1
catapult/2023.1           questasim/2023.1_1  tanner/2022.2u3

---------------------------------- /cad/utility/modulefiles/Synopsys ----------------------------------
cosmos_scope/2019.06        icc2/2020.09-sp3      primesim/2023.12       vcformal/2023.03
```

```
>
> module avail ic
---------------------------------- /cad/utility/modulefiles/Cadence ----------------------------------
ic/06.18.180  ic/06.18.350  ic/23.10.060      icadvm/20.10.170
ic/06.18.320  ic/23.10.030  icadvm/18.10.130

---------------------------------- /cad/utility/modulefiles/Synopsys ----------------------------------
icc2/2020.09-sp3  icc2/2022.03-sp2  icv/2022.12  icv/2022.12-sp4  icv/2023.12-sp4  icv_wb/2023.09
```

```
>
> module help lsf
----------------------------------------------------------------------
Module Specific Help for /cad/utility/modulefiles/Utility/lsf/10.01.13:


*** You don't need to load up any module for LSF ***

            Welcome to use the LSF in EDA Cloud 2.0

1. Be sure to run "xhost +" once in any terminal after logging in, or your
   GUI will fail due to the "can't open display" error.

2. Always learn your LSF usage by running "busers" or "bjobs".
```

```
>
> module disp dc
----------------------------------------------------------------------
/cad/utility/modulefiles/Synopsys/dc/2022.12:

module-whatis   {Synopsys Design Compiler -- RTL synthesis solution}
conflict        dc synthesis
setenv          SYNOPSYS /cad/synopsys/synthesis/2022.12
setenv          SYNOPSYS_SYN_ROOT /cad/synopsys/synthesis/2022.12
set-alias       dv design_vision
prepend-path    PATH /cad/synopsys/synthesis/2022.12/icc2/bin
prepend-path    PATH /cad/synopsys/synthesis/2022.12/bin
prepend-path    MANPATH /cad/synopsys/synthesis/2022.12/doc/syn/man
----------------------------------------------------------------------
>
```

```
>
> module load dc
> which dc_shell-t
/cad/synopsys/synthesis/2022.12/bin/dc_shell-t
>
> module unload dc
> which dc_shell-t
dc_shell-t: Command not found.
>
```

預先於個人帳戶載入 EDA 軟體的方法

1. 編輯 ~/.cshrc 或 ~/.bashrc 並直接將 module load *tool/version* 寫入即可。
2. 較晚載入的軟體，其設定值會置於變數 (如 path) 中較前方的位置。
3. 使用者仍需注意勿載入過多的 EDA 軟體，以免互相衝突或造成困擾。例如，若同時載入 DC 與 PT，則 man set_host_options 指令所看到的內容，會因為載入的次序而不同 (即 MANPATH 的次序)，因而產生混淆。

```
49 #
50 #============================================================
51 #
52 # Load up my frequently used tools in advance.
53
54 module load tkdiff
55 module load ic/06.18.320
56 module load spectre/21.10.389
57
```

對於自動化程式的支援

對於自動化程式的設計而言，GNU Modules 提供了相當完善的支援。凡低階如 csh/tcsh 與 sh/bash，或高階如 Perl 與 Python 等語言，都可以藉由 GNU Modules 將 EDA 軟體環境直接載入 run-time shell 中後直接操作，既方便又快速。現以 bash 與 Perl 舉例說明 VCS 的簡易 wrapper 作法如下：

```
#!/bin/bash
set -e
. /cad/utility/modules/init/bash

verVcs='vcs/2023.03-sp2'
verVerdi='verdi/2023.03-sp2'

module unload    vcs verdi
module load      $verVcs $verVerdi

fsdbPliPath="${VERDI_HOME}/share/PLI/VCS/linux64"
if [ -z "${LD_LIBRARY_PATH}" ]; then
    export LD_LIBRARY_PATH="${fsdbPliPath}"
else
    export LD_LIBRARY_PATH="${fsdbPliPath}:${LD_LIBRARY_PATH}"
fi

exec vcs $@ 2>&1 | tee vcs.log
```

```
#!/cad/utility/perl/bin/perl -w
require "/cad/utility/modules/init/perl.pm";

$verVcs     = 'vcs/2023.03-sp2';
$verVerdi   = 'verdi/2023.03-sp2';

module('unload', 'vcs', 'verdi');
module('load', $verVcs, $verVerdi);

$fsdbPliPath    = "$ENV{'VERDI_HOME'}/share/PLI/VCS/linux64";
$ENV{'LD_LIBRARY_PATH'} =
    defined($ENV{'LD_LIBRARY_PATH'}) ?
        $fsdbPliPath . ':' . $ENV{'LD_LIBRARY_PATH'} :
        $fsdbPliPath;

exec("vcs", @ARGV);
```

視窗出現 TCL 版本衝突的錯誤訊息

若視窗出現下列 TCL 版本衝突的錯誤訊息且無法操作 module 指令時，使用者可以採取以下方法自行排除問題：

```
version conflict for package "Tcl": have 8.6.X, need exactly 8.6.13
```

1. 以點選方式開啟新視窗 (勿從原視窗再衍生新視窗)，通常可以解決此一問題。
2. 若新視窗仍出現相同錯誤訊息，請將 ~/.cshrc，~/.tcshrc 或 .bashrc 先更名與登出後，再重新登入。之後再對 ~/.cshrc 中的命令逐一分析，以排除造成衝突的指令。
3. 請勿於同視窗中載入過多的 EDA 軟體環境，以免造成 DLL 的衝突。如有需要，建議可分散於不同的視窗或 run-time shell 中，載入與執行不同階段所需的 EDA 軟體。

Part 4: 了解 EDA Cloud 2.0 的工作排程器 (IBM Spectrum LSF)

為方便運用叢集中數量龐大的運算伺服器，使用者必須熟悉 LSF 的操作方法。尤其是熟悉各種 EDA 軟體與 LSF 連結的技巧，方能藉由 LSF 來充分發揮 EDA 軟體內建跨平台平行運算的威力，進而縮短運算時間而提高工作效率。

在登入之後，系統會自動載入 LSF 軟體環境，使用者立即可以操作使用。

LSF 的優點

- 使用者工作 (job) 執行效率的最佳化，提高個人生產力。
- 自動平衡伺服器負載，與管理可運用之運算資源。
- 記錄與分析工作、使用者與運算資源的使用狀況。
- 提供與業界相仿的設計環境，日後學員可與業界無縫接軌。
  - LSF 為三大 EDA 軟體廠商支援度最高的商業排程器。
  - LSF 廣為國內外晶片設計或製造公司所使用，學員入業界可以直接學以致用。

LSF 的基礎與運作概念

常見的 LSF 名詞如下：

- 叢集 (cluster)　：LSF 的基本構成單位，目前 EDA Cloud 2.0 只設立一個叢集
- 主機 (host)　：叢集中的管理或運算伺服器
- 工作 (job)　：使用者提交給 LSF 執行的指令 (目前一個工作配置一個工作槽)
- 工作槽 (slot)　：為 LSF 分派工作的基本單位 (目前一個工作槽配置一顆核心)
- 核心 (core)　：指主機呈現的 CPU core (種類與數量)
- 佇列 (queue)　：儲存工作的容器，目前 EDA Cloud 2.0 有三個序列可供使用

常見的 LSF 工作性質分類如下：

- 批次性工作 (batch job)
  - 背景執行，且執行過程可能會被 relocate 或 suspend。需等待排程與分派。

- 互動性工作 (interactive job)
  - 前景執行，且執行過程不會被 suspend。無需排程且會立即被分派。

常見的 LSF 工作狀態分類如下：

- PEND ：已提交 (submit) 且被接受，目前正於佇列中 (queue) 等待分派的工作
- RUN ：已被分派 (dispatch) 給主機，目前正執行中的工作
- *SUSP ：已被分派，但目前被 LSF 或使用者暫停 (suspend) 的工作
- DONE ：工作以零結束值完成 (正常結束)
- EXIT ：工作以非零結束值完成 (通常代表有錯誤發生，需檢查)
- UNKWN ：主機之間失去聯絡。

LSF 的工作提交指令 bsub 介紹：

- 使用者僅需使用 bsub 指令即可提交各種類型的工作，語法如下：

  bsub [*bsub-options*] app-command [*app-options*]

  - 凡 bsub 的參數一定要放在應用程式命令的左方，且切勿與應用程式參數混淆。
  - 如有需要，可以使用兩個連續的短破折號加以區隔。例如：

    bsub [*bsub-options*] -- app-command [*app-options*]

- 在 LSF 分派工作之後，一律會顯示 job ID 與其他訊息，以便後續追蹤。

  - 此訊息無法以參數形式予以停用或格式化，自動化程式需自行捕捉與處理。
  - 請務必檢查所顯示的執行時間與記憶體使用量的上限值。若發現參數有誤或該佇列 (queue) 無法滿足運算需求，請立即刪除該工作並更新參數重新提交。

- 凡使用內建多工或多緒平行運算能力的 EDA 軟體，皆需自行規劃與設定 EDA 軟體的平行運算參數之後，再決定 -n *N* 的參數值以提交主工作。
- 輸入的指令中切勿使用 & 符號再執行其他程式，否則排程器會因為無法追蹤背景程式，而產生錯誤的判斷。也請勿提交 alias 指令，因為 alias 不會被衍生環境所繼承。
- 使用 -R “select[mem>*N*]” 參數僅能找出條件適合的機台，但記憶體不會實際保留。
- 如需同時提交多個程式 (如 regression) 時，可自行開發 script 代勞
- 如果應用程式參數較多，或工作流程較為複雜 (例如以多管道 (“|”) 功能抓取執行訊息: “script2 | tee script2.log”)，可以使用兩種方法解決：

  - 將工作流程寫成一個傳統的 script，再以上述之傳統方法提交該 script。
  - 或以 template 方式 (稍後介紹) 開發 script，再以 pipeline 模式 (即 < 符號) 提交。如：

    bsub < my_job.sh

提交指令 bsub 的常用參數介紹：

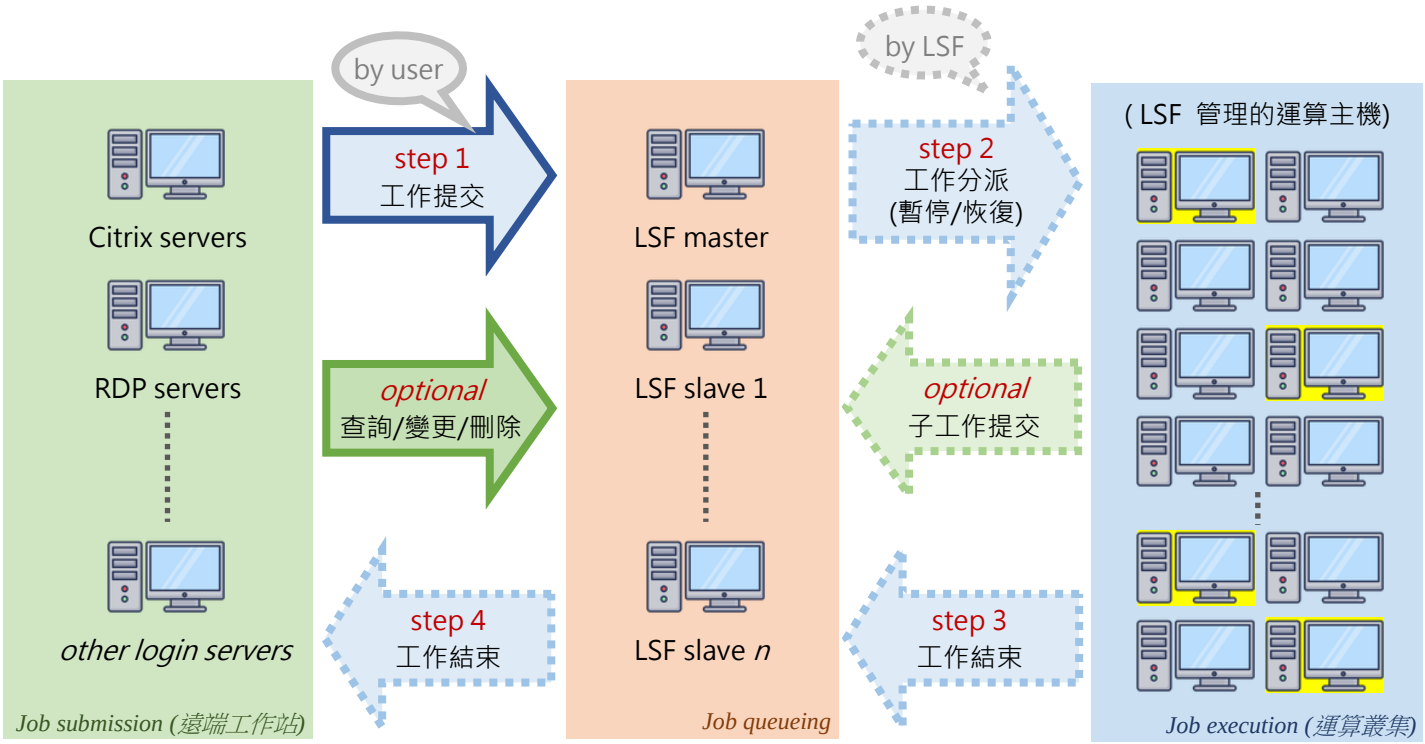
Option	用　途	說 明 與 範 例
-q *queue*	指定 queue	> bsub -q bat.l my_dc.bat
-Ip	指定為交談模式，並建立虛擬螢幕。	> bsub -Ip design_vision > bsub -Ip vim big-file
-J *job_name*	指定 job name。	> bsub -J dc_try dc_shell-t…
-n N	指定所需要的 slot (core) 數量 。	> bsub -n 4 hspice -mt 4… > bsub -n 8 dc_shell-t …
-R “select[mem>*N*]”	選擇尚有 N GB 記憶體的主機(但記憶體不會被保留)。	> bsub -R “select[mem>64]” verdi
-o logfile	指定 stdout 與 stderr 輸出檔(如果未使用 -e)。	> bsub -o my_dc.log my_dc.bat > bsub -o “dc_%J.log” my_dc.bat
-cwd work_dir	指定工作目錄。	> bsub -cwd ./project1 dc_shell-t…
-W [hour:]min	指定執行時間的上限 (逾時十分鐘後會觸發 auto kill)。	> bsub -W 120 dc_shell-t… > bsub -W 48:00 hspice
-K	在指令結束後才會釋放視窗，只能在 batch mode 使用。	> bsub -K make compile

常用的 LSF 控制與查詢指令：

Command	用　途	說 明 與 範 例
busers	顯示 job slot 的使用狀況。	> busers (顯示使用者個人的使用狀況) > busers *group* (顯示組群的使用狀況)
bjobs	顯示工作 (job) 的狀況。	> bjobs (顯示使用者所有尚未完成的工作) > bjobs -l [*job-id*] (顯示完整資訊)
bkill	刪除工作。	> bkill *job-id* (刪除 *job-id* 的工作) > bkill 0 (zero，刪除使用者所有尚未完成的工作)
bqueues	顯示 queue 的規範或狀況。	> bqueues (顯示簡要資訊) > bqueues -l [*queue*] (顯示完整資訊)
bsub -Ip bpeek	顯示批次工作的即時輸出訊息。需以交談方式提交執行。	> bpeek -f *job-id* (以 tail 顯示即時輸出) > bpeek -J *job-name* (顯示即時輸出)
bacct / bhist	顯示已執行完畢的工作摘要(用於偵錯與執行效能)。	> bacct [-l] *job-id* (顯示工作摘要) > bhist [-l] *job-id* (顯示排程摘要)

LSF 的工作流程示意圖：

- 簡而言之，使用者只要專注於工作提交即可，其餘工作皆由排程器代勞。



使用模板 (template) 將工作提交自動化

```
#!/bin/bash

#BSUB -q bat.1
#BSUB -n 8
#BSUB -R "select[mem>32]"
#BSUB -o out.%J.log
#BSUB -J my_dc_job

echo "Job ID : $LSB_JOBID"
echo "Job Execution Host : $LSB_HOSTS"
echo "Job Sub. Directory : $LS_SUBCWD"

. /cad/utility/modules/init/bash
module load dc
pre_task.sh
dc_shell-t...
```

- 模板 (template) 的好處，在於可將整個流程寫在一起，並記錄所採用的提交參數。除了可以避免日後遺忘疏漏之外，工作提交也會變得更容易。
- 模板可以使用 sh/bash，csh/tcsh，Perl 或 Python 來撰寫，使用者可以自行選用。

  - 將提交參數以註解形式定義在模板的前方，之後再撰寫工作內容。
  - 可以搭配 GNU Modules 載入 EDA 軟體環境，一氣呵成，不須事前先預載。
  - 最後執行 bsub < *template* 即可提交工作。

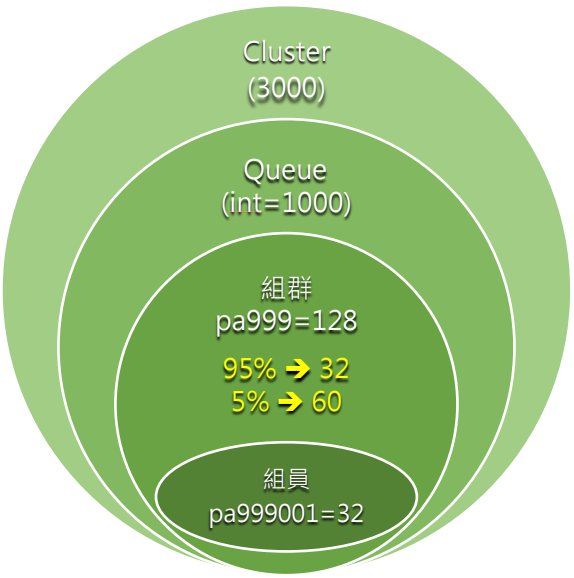
- 缺點是無法在 template 後方加入命令列參數。

  - 故無法經由命令列參數來改變程式的行為。
  - 建議同一工作可以開發批次用或互動用的兩個版本，交替使用。

- 使用者可參考 /lsf/examples 的範例，自行開發適用的版本。

組群與組員的工作槽 (job slot) 的數量配置關係

- 叢集工作槽數量為排程系統可運用的上限值。若超過，則所有後續的工作皆需等候。
- 佇列工作槽數量為該佇列可運用的上限值。若超過，則該佇列後續的工作皆需等候。
- 組群工作槽數量為該組群 (相同 GID) 可同時運用的上限值。若超過，則所有組員後續的工作皆需等候。此值可以使用 buser *group-name* 指令查詢。
- 組員工作槽數量為組員可同時運用的上限值。若超過，則該員後續的工作皆需等候。此值可以使用 buser 指令查詢。



Part 5: 線上查詢與參考網站

- EDA Cloud 2.0 帳號申請，請參閱官網的 EDACloud2.0 帳號申請說明 檔案。
- EDA Cloud 2.0 登入步驟與設定，請參閱官網的 EDACloud2.0 操作手冊 檔案。
- EDA Cloud 2.0 一般性操作問題，請參考官網的 EDACloud2.0 使用常見問題 檔案。
- 製程檔案皆置於 EDA Cloud 2.0 的 /process 目錄，可參閱官網的使用說明檔案。
- 與排程器相關的參考範例皆置於 EDA Cloud 2.0 的 /lsf/examples 目錄。
- 線上執行 man bsub (或其他 LSF 指令) 可查詢詳細的參數說明。
- 線上執行 module help *tool* 可查詢簡要資訊，如 module help ic 或 module help hspice。
- 若 EDA 原廠有支援，線上執行 man *tool* 即可取得與 LSF 連結的方法，如 man innovus 或 man set_host_options。若無，請參考使用手冊或搜尋網際網路。

- Reference sites for LSF
  - LSF at Technical University of Denmark
  - LSF at Icahn School of Medicine
  - LSF at NC State University
  - LSF at Livermore Computing's HPC

- Reference sites for GNU Modules
  - GNU Modules at 創進一號
  - GNU Modules at lmod.readthedocs.io
  - GNU Modules at Technical University of Denmark

(全文完)