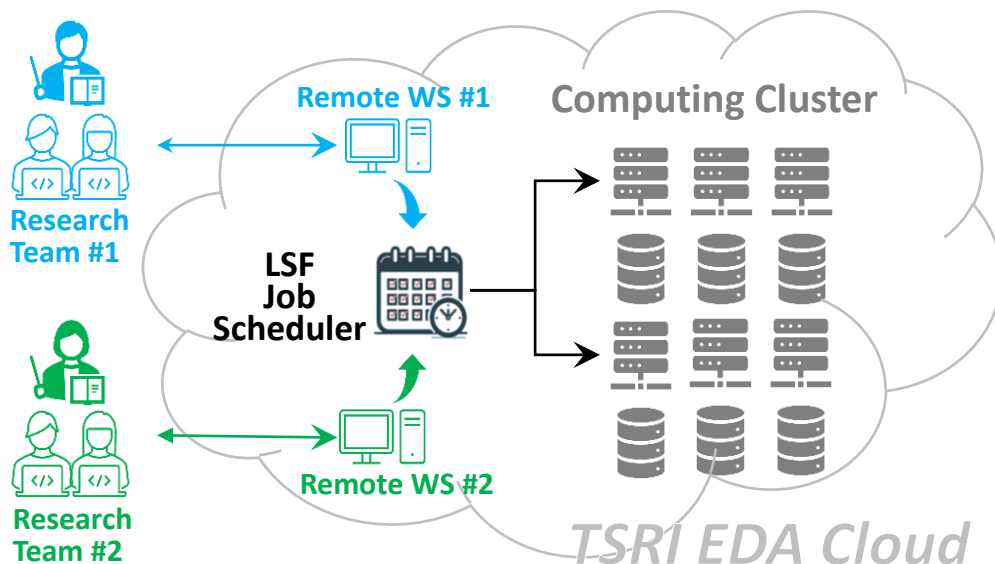


EDA 軟體啟動與執行 in TSRI EDA Cloud 2.0 快速指引

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，可先參考第五章“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。

使用者亦需留意勿陷入執行緒 (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

前言.....	1
EDA 軟體啟動.....	4
步驟一：新桌面需執行一次 “xhost +” 以允許遠端顯示	4
步驟二：使用 GNU Modules (建議) 或 source file 載入 EDA 軟體環境	4
步驟三：提交指令給 LSF 工作排程器	5
視窗出現 TCL 版本衝突訊息的解決方法	5
執行未知介面形態的軟體.....	6
無法與 LSF 配合使用的軟體	6
EDA Cloud 1.0 R/Q 指令的對應指令	7
查看工作的執行狀態.....	11
LSF 指令執行注意事項	11
線上查詢與參考網站.....	13

EDA 軟體啟動

步驟一：新桌面需執行一次 “xhost +” 以允許遠端顯示

請務必在每次 EDA Cloud 遠端工作站**建立新桌面** (desktop) 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 (remote display)。否則，EDA 軟體執行將會產生 cannot open display 的錯誤訊息並終止執行。

步驟二：使用 GNU Modules (建議) 或 source file 載入 EDA 軟體環境

- 使用 GNU Modules。

- 先執行 **module avail** 列出所有可用的 EDA 軟體與版本。

- ✧ 執行 **module avail keyword** 可列出模組名稱中包含關鍵字的所有軟體與版本。
- ✧ 執行 **module disp tool** 可先瞭解軟體的設定內容。
- ✧ 執行 **module help tool** 可先瞭解軟體的使用方法摘要 (不一定有提供)。

- 再執行 **module load tool[/version]** 即可載入 EDA 軟體環境設定，例如：

- > **module load calibre** (載入 Siemens Calibre 的預設版本)
- > **module load customcompiler** (載入 Synopsys Custom Compiler)
- > **module load hspice** (載入 Synopsys HSPICE)
- > **module load ic** (載入 Cadence IC)
- > **module load synthesis** (載入 Synopsys Design Compiler)
- > **module load verdi_vcs/2023.03** (載入 Synopsys Verdi 2023.03 版本)

- 使用基於 Csh 的 source file。

- 需找出所需軟體的 source file 的絕對路徑，然後直接 **source** 即可。例如：

- > **source /cad/synopsys/CIC/synthesis.csh**
- > **source /cad/cadence/CIC/IC.csh**
- > **source /cad/mentor/CIC/calibre.csh**

步驟三：提交指令給 LSF 工作排程器

使用者需提交 EDA 軟體指令給 LSF 工作排程器，再由排程器送交運算叢集 (computing cluster) 執行或放在序列 (queue) 中等待執行。工作提交模式分為兩種：

- 互動模式 (interactive mode)。

以 **bsub -Ip** 提交。軟體將於前景執行，適合需於原文字視窗與軟體互動的情境。例如：

```
> bsub -Ip virtuoso
> bsub -Ip design_vision
> bsub -Ip icc2_shell -gui
> bsub -Ip matlab
> bsub -Ip vcs -R -f vlog.f
> bsub -Ip ncverilog -f vlog.f
> bsub -Ip -n 4 hspice test.sp -mt 4
```

- 批次模式(batch mode)。

以 **bsub** 提交。軟體將於背景執行，而提交後可於原文字視窗中繼續其他操作或提交其他指令。程式或命令檔中請勿再使用 **&** (背景執行符號)，以免排程器誤判而造成錯誤。例如：

```
> bsub Verdi
> bsub cx
> bsub simvision
> bsub ncverilog -f vlog.f
> bsub vcs -R -f vlog.f
> bsub -n 4 pt_shell -f nogui.tcl
> bsub -n 4 calibre -drc -hier -turbo 4 drc.run (4 核心運算，搭配 Calibre 的 4 個緒)
> bsub -n 2 virtuoso (2 核心運算，另一核心給外掛軟體使用)
> bsub -n 4 dc_shell-t -f nogui.tcl (純文字模式執行 Design Compiler)
> bsub -n 4 hspice test.sp -mt 4 (4 核心運算，搭配 HSpice 的 4 個緒)
```

視窗出現 TCL 版本衝突訊息的解決方法

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。

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)
```

無法與 LSF 配合使用的軟體

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)
```

EDA Cloud 1.0 R/Q 指令的對應指令

所列出之對應指令以批模式為主。如欲使用**互動模式**，請自行附加提交參數 **-Ip** 即可。

EDA Cloud 1.0	EDA Cloud 2.0
Rncverilog Rirun Rrxrun	<ul style="list-style-type: none"> > module load xcelium > bsub ncverilog > bsub irun > bsub xrun <p>新版 Verdi 已不再支援 XCelium，請改用 Simvision 來顯示波形與偵錯。</p>
Rvcs	<ul style="list-style-type: none"> > module load vcs; bsub vcs <p>如需輸出 FSDB，請另載入 verdi_vcs 模組，並添加兩個 VCS 參數：-full64 與 -debug_access。若欲於編譯後直接跑模擬，則需再添加 -R 參數。即：</p> <ul style="list-style-type: none"> > module load vcs verdi_vcs > bsub vcs -full64 -debug_access -R...
Rvsim	<ul style="list-style-type: none"> > module load questasim; bsub vsim
Rverdi RnWave Rfsdb2vcd	<ul style="list-style-type: none"> > module load verdi_vcs > bsub verdi 或 bsub nWave > bsub fsdb2vcd
RnLint (以 Spyglass 取代)	<ul style="list-style-type: none"> > module load spyglass > bsub spyglass -shell 或 bsub spyglass (GUI)
Rdc_shell Rdc_shell-t Rdv	<ul style="list-style-type: none"> > module load synthesis 或 module load dc > bsub dc_shell 或 bsub dc_shell-t > bsub -Ip design_vision
Rformality Rfm_shell	<ul style="list-style-type: none"> > module load formality 或 module load fm > bsub fm_shell 或 bsub formality (GUI)
Rprimetime Rpt_shell	<ul style="list-style-type: none"> > module load primetime 或 module load pt > bsub pt_shell 或 bsub -lp pt_shell -gui (GUI)
Ricc_shell Ricc2_shell	<ul style="list-style-type: none"> > module load icc2 > bsub icc_shell 或 bsub -Ip icc_shell -gui (GUI) > bsub icc2_shell 或 bsub -Ip icc2_shell -gui (GUI)

EDA Cloud 1.0	EDA Cloud 2.0
Rvirtuoso	> module load ic; bsub virtuoso
Rcustomcompiler	> module load customcompiler > bsub custom_compiler
Rlaker_oa	> module load laker_oa; bsub laker
Rhspice	> module load hspice; bsub hspice
Rspectre	> module load spectre; bsub spectre
Rcalibre Rcalibreve	> module load calibre > bsub calibre 或 bsub calibreve 因 Calibre 的 -hyper 參數會使用較多的記憶體，故凡工作因此而被 LSF 刪除時，請移除此參數後再重新執行。
Rcx	> module load customexplorer; bsub cx
Ricv	> module load icv > bsub icv 或 bsub icv_vue
RStarXtract	> module load starrcxt > bsub StarXtract 或 bsub -Ip starrc_shell
Rinnovus	> module load innovus > bsub innovus -no_gui (不啟動 GUI) > bsub -Ip innovus (啟動 GUI 或需以文字模式互動)
Rmilkyway	> module load milkyway; bsub Milkyway
Rtmax	> module load tmax; bsub tmax
Qentry	若設計案較大且記憶體使用量已超過上限值 (250GB) 而失敗，請檢查並停用 Calibre 的 -hyper 參數，即可降低記憶體使用量。 DRC 與 LVS 子功能 請參考 dsd_EDACloud2_FC_*.pdf 來填寫對話視窗的內容。(包含 full custom 的 LPE 流程) Add dummy (metal fill) 子功能 請參考 rundrc 檔案說明。此檔案通常位於製程的 IP/CBDK-core*/CIC/Calibre/drc 目錄內。

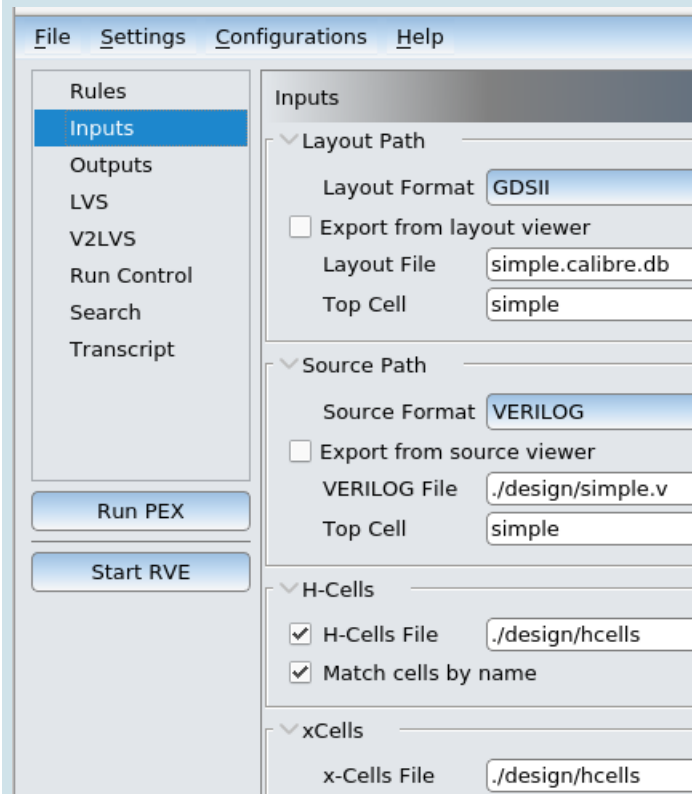
LPE 子功能 (特指 cell based 的設計流程)

下文所提及的 hard macro IP 是指設計使用到的所有 std. core/IO cell、PLL 或 memory block 等 leaf cell。請自行找出相關檔案的絕對路徑，並依照檔案格式逐一建立輸入檔備用。

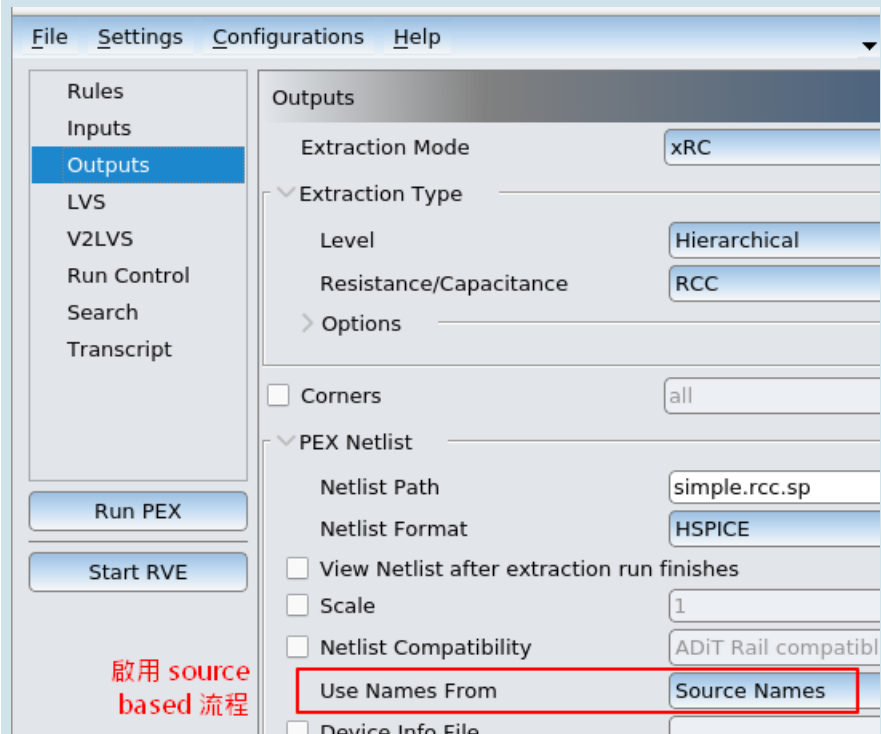
1. LPE 所需的 rule file 通常位於製程的 TECH/1_TSRI/PDK/Calibre/rcx 目錄內。
2. Verilog netlist：ARP 完成後的檔案，此範例為 simple.v。
3. Verilog Library file：hard macro IP 的 Verilog model。檔案通常位於製程的 IP/CBDK*/CIC/Verilog 目錄內。
4. include SPICE file：hard macro IP 的 spice netlist。檔案通常位於 IP/CBDK*/orig_lib/TSMCHOME/digital/Back_End/spice 目錄內。
5. H-Cells file：hard macro IP 的 cell list。檔案中每一行描述個別 cell 的 source_subckt_name 與 layout_instant_name 即可。
6. 請自行設定 V2LVS 中 hard macro IP 所需要的 global power/ground netname，TSMC 的 P/G 為 VDD/VSS，UMC 的 P/G 為 VCC/GND，範例 P/G 為 VCC/VSS。

PEX 對話視窗填寫範例

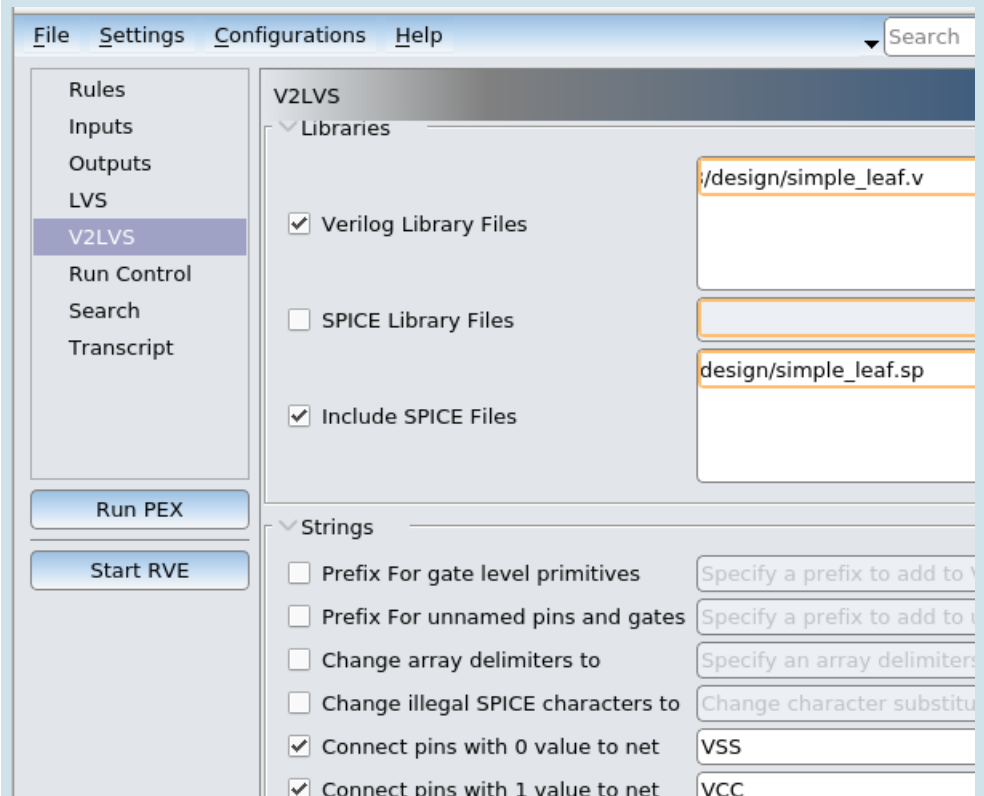
Input 對話視窗 (source format 務必選擇 Verilog)：



Output 對話視窗 (本範例選用 "Hierarchical+RCC"。Cell-based 設計案建議採用 "Hierarchical+C+CC" 或 "Hierarchical+RC") :



V2LVS 對話視窗 (source format 設定為 Verilog 時才會顯示) :



各種 Memory compiler

> 以 `bsub [-Ip]` 提交包含絕對路徑的指令即可

查看工作的執行狀態

常用的 LSF 工作排程器指令如下：

- 工作提交：bsub
- 執行過程：bjobs、bkill 或 bpeek (因防火牆因素，需提交至 queue 中執行)
- 執行完畢：bacct 或 bhist
- 任何時間：bjobs、busers、bqueues 或 module help lsf

指令	用途	說明與範例
busers	顯示使用者的 slot (CPU core) 資源使用狀態	1. busers (顯示使用者個人的使用狀態) 2. busers group-name (顯示組群的使用狀態)
bjobs	顯示尚未完成的工作狀態	1. bjobs (顯示使用者所有尚未完成的工作) 2. bjobs [-l] [job-id] (顯示完整資訊)
bkill	刪除工作	1. bkill job-id (刪除 job-id 的工作) 2. bkill -b 0 (刪除使用者所有尚未完成的工作)
bqueues	顯示執行佇列的種類、規範與現況	1. bqueues (顯示簡要資訊) 2. 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。	1. bpeek job-id 2. bpeek -f job-id (以 tail 顯示即時輸出)
bacct bhist	顯示已執行完畢工作的運算資源使用摘要	1. bacct [-l] job-ID 2. bhist [-l] job-ID

LSF 指令執行注意事項

- 提交參數 **-n N** 是用來申請 CPU 核心資源，需與 EDA 軟體的多執行緒參數一致。
- 以 LSF 執行 EDA 軟體，工作目錄即為提交指令當下的目錄 (除非經由提交參數更改)，執行結果會直接置於工作目錄中。
- 所有 EDA 軟體皆可使用**互動模式**執行，但最多僅能使用 **16** 核心。
 - 批次模式可使用較多核心與記憶體，請多加利用。
 - 但不建議給予交談式軟體 (如 RVE、Virtuoso 或 Custom Compiler 等等) 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。只有純運算的軟體才能充分利用多核心資源的威力。

- 以 LSF 提交 EDA 軟體執行指令，將自動產生 log 與 rep 檔。
 - 自動產生 `job_JID.rep` 檔，內容為 LSF 工作執行摘要，包含工作的起訖時間，與所使用的核心數與記憶體總量等等。
 - 批次模式會自動產生 `job_JID.log` 檔，內為執行過程所有訊息，方便查詢與偵錯。
 - 請使用者自行刪除已不再需要的 log 與 rep 檔

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)