

第十一章 計算環境的設定與計算效能

計算環境的設定對於計算效率常有重要的影響，以下針對 Gaussian 程式使用上常用到的設定做一簡單介紹。

I. Gaussian 09 input file 格式範例 (在 Unix 系統中副檔名預設為 .com)

```
%mem=1000M          (設定計算時所使用的記憶體量)
%NprocShared=4      (設定計算時所使用的核心數)
%chk=H2O            (設定存取 checkpoint file 的檔名，內容包含結構與波函數等計算結果)
#MP2/aug-cc-pVTZ OPT FREQ      (理論方法及 keywords，可多行)
(空一行)
Geometry optimization and frequencies calculation  (計算工作的註解欄，可多行)
(空一行)
0 1                (charge 和 multiplicity，multiplicity = 2S+1，S 為分子的電子自旋量子數)
H                  (這裡開始輸入分子的結構，這裡是水的 Z-matrix 結構)
O 1 R
H 2 R 1 A
(空一行)
R = 0.98
A = 109.5
(空一行)
其他輸入資料 (可多行)
(空一行)
```

II. Gaussian 軟體免費線上手冊 (目前有 09 與 16 版本)

Gaussian 16 使用手冊: <http://gaussian.com/man/>

Gaussian IOP 使用手冊: <http://gaussian.com/iops/>

Gaussian 09 手冊可由 Gaussian 官網取得：

1. 先到 Gaussian 官網：<http://gaussian.com/>
2. 選擇 **Support** 項目，會顯示選單，在 **Historical Documents** 區域底下有個 **G09 Keyword pages** 選項，點擊 **G09 Keyword pages** 選項，下載一個壓縮檔。
3. 解壓縮 Gaussian 09 的網頁檔案後，可直接用瀏覽器開啟。

III. 設定記憶體的使用量

首先第一個問題是記憶體的使用量，**Gaussian 09** 的記憶體該使用多少呢？以下我們以聯苯(biphenyl, $C_{12}H_{10}$)為測試對象，使用常見的 DFT 理論 B3LYP 搭配 aug-cc-pVTZ 基底，來測試結構最佳化(keyword : OPT)，在使用不同記憶體量下所需要時間。

測試所使用的電腦配備為

(1) 中央處理器為 Intel(R) Core(TM) i7-8700K CPU @ 4.50GHz
總記憶體量為 65536 MB (64GB)

使用 6 核心計算(%NProcShared=6)，搭配不同記憶體量的計算時間：

記憶體使用量(MB)	Real Time (s)
256	139
512	177
1024	144
2048	145
4096	137
8192	179
32768 (32GB)	111
65536 MB (64GB)	113

測試結果顯示使用大於 32GB 開始有較明顯的減少時間，主要是因為 Gaussian09 在計算上能量收斂計算預設為 SCF=direct，部分站存檔直接計算以減少讀寫硬碟，記憶體使用 32GB 預設就會轉為 SCF=incore，把大部分的暫存檔全部存到記憶體裡面，因此計算時間會減少，大約使用 50% 的記憶體就可以達到最佳的計算效率，更多記憶體並沒有幫助，反而可能會因為占用作業系統需要的記憶體，造成作業系統反應會變緩慢，增加計算工作的困擾。針對 Gaussian 09 的計算，如果沒有其他考量，一般建議可將記憶體設定在系統記憶體總量的 50% 左右。

IV. 設定計算使用的核心數

大部分人在執行複雜的計算工作時，都會使用多核心 CPU 進行平行化計算以增加計算效率，**以下我們探討 Gaussian 09 的平行化效率**？我們同樣針對吡嗪(Pyrazine， $C_4H_4N_2$)分子，使用 B3LYP 搭配 aug-cc-pVTZ 基底做結構最佳化計算，測試在使用不同核心數下計算所需的時間 (計算時記憶體設定為 32768 MB)，測試一使用二種 6 核心處理器，測試二則是使用 8 核心處理器

測試一所使用的電腦配備如下：

(1) 中央處理器為 Intel(R) Core(TM) i7-6800K CPU @ 3.40GHz

總記憶體量為 65536 MB (64GB)

CPU 核心數	Real Time (s)	平行化效能 (以單核心基準)
1	531	
2	280	1.9
3	198	2.7
4	159	3.3
5	134	4.0
6	120	4.4

(2) 中央處理器為 Intel(R) Core(TM) i7-8700K CPU @ 4.50GHz

總記憶體量為 65536 MB (64GB)

CPU 核心數	Real Time (s)	平行化效能 (以單核心基準)
1	344	
2	185	1.9
3	137	2.5
4	117	2.9
5	109	3.2
6	105	3.3

可以看到6核心的工作站電腦計算速度的提升由單核提升至雙核最為明顯 (速度變為 1.9 倍)，以 Intel(R) Core(TM) i7-6800K CPU 來看而使用六核心速度約為單核心的 4.4 倍、雙核心的 2.3 倍、三核心的 1.7 倍，雖然 i7-8700K CPU 平行化效能並沒有比 i7-6800K CPU 好，但是整體來說因為時脈提高還是減少了計算時間，單顆核心來說效能提升約 50%。

測試二所使用的電腦配備為

(1) 中央處理器為 Intel(R) Core(TM) i7-7820X CPU @ 3.60GHz

總記憶體量為 65536 MB (64GB)

CPU 核心數	Real Time (s)	平行化效能 (以單核心基準)
1	535	
2	296	1.8
4	169	3.2
8	112	4.8

不管是 6 核心或 8 核心的工作站電腦計算速度由單核提升至雙核最為明顯 (速度變為 1.8~1.9 倍)，而使用八核心速度約為單核心的 4.8 倍、雙核心的 2.6 倍、四核心的 1.5 倍，計算速度亦有不錯的提升，不過從 6 核心 CPU 到 8 核心 CPU，平行化效能並沒差異太多，以相同 6 核心的電腦來看，較新的 CPU 會因為單核心時脈提高而減少計算時間(單核心加快約 1.5 倍)，但是平行化效率不見得有蠻明顯的改善，一般建議在 Linux 工作站上使用 4-8 核心進行計算，CPU 時脈的提高還是改善效率較重要的因素，在 Windows 的環境中須使用 Multi-Core 版本才可使用多核心計算。

接下來我們比較國網中心台灣杉(Taiwania)與桌上型工作站電腦的效能差異，測試的分子為吡嗪 (Pyrazine) 並使用 B3LYP 方法搭配 aug-cc-pVTZ 基底函數做結構最佳化，記憶體使用量設定為 8 GB，所使用的電腦配備為

1. 桌上型工作站: 中央處理器為 Intel(R) Core(TM) i7-8700K CPU @ 4.50GHz，總記憶體量為 65536 MB (64GB)。
2. 台灣杉(Taiwania): 中央處理器為 Intel Xeon Gold 6148 2.40GHz CPU 20 核心 CPU，一個計算節點具備 192 GB 記憶體(加速計算節點記憶體 192GB、大型記憶體結點具備 384GB 記憶體)，2 顆 CPU，單顆 20 核心，總共 40 核心，加速結點同時具備 4 張 Nvidia Tesla P100 GPU 加速卡，全機總共 27,760 計算核心，總記憶體量為 157 TB。

	Intel Xeon Gold 6148 2.40GHz CPU		Intel(R) Core(TM) i7-8700K CPU @ 4.50GHz	
CPU 核心數	Real Time	平行化效能	Real Time	平行化效能
1	1944	—	436	—
4	439	4.4	131	3.3
6	316	6.2	109	4.0
12	179	10.8	—	—
20	116	16.7	—	—
40	78	25	—	—

以 1 核心來說 Intel i7 8700K 的 CPU 計算速度約是 Intel Xeon Gold 6148 CPU 的 3.5 倍，以使用 4 核心計算來看，Intel i7 8700K 的 CPU 約快 Intel Xeon Gold 6148 CPU 3.4 倍，6 核心來看 Intel i7 8700K 的 CPU 計算速度約快 2.9 倍。平行化效率比較(相較於 1 核心)，Intel Xeon Gold 6148 CPU 使用 4 核心加快 4.4 倍，12 核心加快 10.8 倍，使用一個計算節點全部核心約加快 25 倍，Intel i7 8700K CPU 4 核心加快 3.3 倍，使用 6 核心加快 4.0 倍，可以看到 Gaussian 09 的平行化效率並沒有特別好，使用很高的核心數並不會說效能就一定很好。

V. 初始分子結構對稱性的影響

從事單點以及結構最佳化計算時，起始分子結構的對稱性亦會影響計算時間，[這裡將討論分子結構對稱性對於計算時間的影響](#)，以下以吡嗪分子做結構最佳化為例，使用 6 核心搭配 32768 MB 記憶體體的計算時間為：

(1)中央處理器為 Intel(R) Core(TM) i7-6800K CPU @ 3.40GHz

總記憶體量為 65536 MB (64GB)

對稱性 (point group)	Real Time (s)
高 (D_{2h})	119
低 (C_1)	421

(2) 中央處理器為 Intel(R) Core(TM) i7-8700K CPU @ 4.50GHz

總記憶體量為 65536 MB (64GB)

對稱性 (point group)	Real Time (s)
高 (D_{2h})	105
低 (C_1)	387

可以看到通常起始結構對稱性越高，計算越有效率，以這個例子來看時間快了 3.5 ~ 3.7 倍，這對較大的分子或使用高階理論時格外重要。

VI. SCF 運算方式設定影響 (In-core、Direct 與 Conventional)

Gaussian 09 在進行 SCF 電子能量收斂過程中提供了幾種不同的暫存資料方式，分別為 **In-core**、**Direct** 與 **Conventional**。**In-core** 代表在收斂過程中，所需計算的雙電子積分會在經過第一次計算之後將其全部儲存在記憶體 (memory) 內。**Conventional** 代表在計算過程中，雙電子積分在經過第一次計算之後將其全部儲存在硬碟 (disk) 上，以供需要時讀取。而 **Direct** 則代表不將積分結果儲存起來，而是需要時直接計算。在 Gaussian 09 中，當記憶體足夠將自動使用 Incore，否則將切換為 Direct。以下我們以 B3LYP/aug-cc-pVTZ 方法計算 pyridine 分子的單點能量為例，比較 SCF 不同的設定對計算時間的影響。

測試所使用的電腦配備為

(1) 中央處理器：Intel(R) Core(TM) i7-6800K CPU @ 3.40GHz

總記憶體量為 65536 MB (64GB)

使用 6 核心搭配 41952 MB 記憶體 (Incore 的記憶體下限)

SCF 設定	Real Time (s)
In-core	33
Direct	68
Conventional	137

(2) 中央處理器為 Intel(R) Core(TM) i7-8700K CPU @ 4.50GHz

總記憶體量為 65536 MB (64GB)

SCF 設定	Real Time (s)
In-core	28
Direct	44
Conventional	87

所以在進行 Hartree-Fock 與 Density Functional Theory (DFT) 的計算時，如果記憶體的大小允許，則使用 In-core 的效率較高，若記憶體不足以使用 In-core，則程式會自動使用 Direct 方法，若強制設定 **SCF=Incore**，使用者必須設定足夠的記憶體才能夠進行計算 (這裡需要約 40 GB 記憶體)，不然計算會自動終止。

VII. Scratch file 位置的設定

在 Gaussian 計算過程中會產生暫存資料的 scratch file (rwf 檔)，檔案的大小有可能會高達數十 GB 以上，檔案的讀寫有可能成為計算速度的瓶頸，因此通常建議將 scratch file 的位置指定在與 input/output files 不同的高速硬碟分割 (partition)，最好配備 Raid 0 增加讀寫速度，千萬不要用網路磁碟。(Unix 中使用 setenv GAUSS_SCRDIR 指令來設定 scratch file 的位置，32 位元的 G09W 會有 2 GB 的檔案大小限制)

蔡承成，胡維平
國立中正大學
化學暨生物化學系
© Copyright 2018