# g16sub コマンドによる Gaussianジョブの投入

自然科学研究機構 岡崎共通研究施設

計算科学研究センター(RCCS)

### 更新履歴

- •2019/7/29 初稿作成
- ・2020/1/16 情報の更新

更新

- 2020/2/3 %mem, %nprocshared, %cpu についての注意
- 2021/2/18



この資料ではRCCSで独自に用意した g16sub コマンドを 用いて Gaussian ジョブを投入する方法を説明します。



- 事前に必要な準備
- サンプルGaussianインプットファイル
- •ファイルの転送
- ・ログイン
- •計算(ジョブ)の投入
- ・ジョブの状態確認
- ・ジョブが終了すると?
- formchkの実行
- ・ジョブ実行のヒント

### 事前に必要な準備

本資料の内容を実行するにあたって以下の準備が必要になります。

- ・RCCSのフロントエンド(ccfep)へログインできるようにする
- RCCSとのファイルのやりとりをできるようにする(scp, sftp)
  - (上記設定については<u>クイックスタートガイドページ</u>にも情報があります)
- <u>Gaussianのインプットファイル(.com, .gjf</u>)
  - g09sub, g16sub を使う場合は、メモリ量(%MEM)や並列数 (%CPU, %GPUCPU)の指定は必要ありません。

# サンプル Gaussian インプットファイル

今回はこのようなインプット(ch3cl.gjf)をサンプルとして使います。

```
%chk=ch3cl.chk
# HF/6-31G(d,p) Opt
```

methyl chloride

0,1			
С	-0.000004	1.127470	0.000000
Н	-0.511417	1.468491	0.885898
Н	-0.511417	1.468491	-0.885898
Н	1.022922	1.468527	0.000000
Cl	-0.000004	-0.657078	0.000000

通常、%mem, %nprocshared, %cpu の指定は g09sub や g16sub によって 上書きされます。コア数については –np (コア数) オプションで指定してください。 メモリは利用可能な上限値が自動的に設定されるため、指定は通常不要です。

(標準的な Gaussian インプットファイルの拡張子は .gjf もしくは .com です)

<u>ファイルの転送(1)</u>

#### この例では WinSCP を用いて RCCS に接続、転送します。

@ccfep.ims.ac.jp - WinSCP				<u>17</u> 35		×
ローカル(L) マーク(M) ファイル(F) コマンド(C) セッション(S) オプション(O) リモート(R) ヘルプ(H)						
🕀 🔁 同期 🔳 🥐 💽 🚳 📦 キュー 🔹 転送設定 デフォルト 🔹	- 🧭					
🪽 @ccfep.ims.ac.jp × 🕎 新しいセッション						
■デスクトップ ・ 🚰 ・ 🛐 ・ 💼 🕟 🏠 🔁 🔚 🖛 - → -		-	🪰 • 🝸 •  🔁 🌈 🏠 ファイルの検索	₽_  >	÷ +	
📓 アップロード 🗸 📝 編集 🔹 🗶 📝 🕞 プロパティ  音 新規 🔹 💽 🗹		ダウンロード・	② 編集 → 🗙 💟 🔂 プロパティ   🗳 新規 → 📗	+ - 4		
C:¥Users¥ ¥Desktop¥		/lustre/home/users	5/ /			
A前     サイズ     種類     更新日時	^	名前 ▶	サイズ 更新日時 ~ 2019/05/14 11:52:37	パーミッション rwxr-xr-x	所有者 root	^
ch3cl.aif 1 KB GJF ファイル 2019/05/24 9:05:36						
Second Contract Contract Contract	~		and the second sec		0	~
0 B (全 149 MB 中)/ 0 個目(全 28 ファイル中)	1 非表示	0 B (全 1.87 GB 中	ョ)/ 0 個目(全 61 ファイル中)		54	非表示

ファイルの転送(2)

#### RCCS側のホームディレクトリ(/home/users/ユーザ名)以下にCH3CIという ディレクトリを作成し、そこに ch3cl.gjf ファイルを送ってみます。

								,
@ccfep.ims	s.ac.jp - WinSCP					_		$\times$
- ローカル(L) マーク(M) ファ	ァイル(F) コマンド(C) セッション(S) オ	プション(O) リモート(R) ヘルプ(H)						
🖶 🛃 🛜 同期 📃 (	🥜 🛃 🚳 🗊 キュー・ 転送	設定 デフォルト 🔹	<i>ଟ</i> -					
@ccfep.ims.ac.jp	→ × 📑 新しいセッション							
デスクトップ 🔹 🖡	🗂 • 🝸 • 🗈 🖬 🏠 🎜 🔚		1	• 🚰 • 🝸 • 🗈	🔁 🎧 🍠 📘 🗋 ファイルの検索	s 📴 🔶>	Ŧ	
🗐 📄 アップロード 👻 🕯	編集 🔹 🗙 📝 🕞 プロパティ 📑	新規 - 🛃 🖃 💟	ダウン	コード - 📝 編集 - 🗙	🖉 🔂 プロパティ 📔 新規・	+ - 4		
C:¥Users¥	¥Desktop¥		/lustre/hor	ne/users/ / 🔭 .	ムディレクトリ	(RCCS)	)	
名前	サイズ 種類	更新日時	▲ 名前	שיר ש	イズ更新日時	パーミッション	所有者	^
					2019/05/14 11:52:37	rwxr-xr-x	root	
Ch3cl aif	1 KB GIE ファイル	2019/05/24 9:05:36	CH3CI		2019/05/27 10:09:46	rwxr-x		
				送信先ア	イレクトリ(R	CCS側)		
Gauss	Sianインノット(コ	ューサー側)						
( ( ( ( ( ( ( ( ( ( ( ( ( ( ( ( ( ( (								
the second second								
			✓					~
0 B (全 149 MB 中) /	0個目(全28ファイル中)	1	非表示 0B (全 1.	87 GB 中)/ 0 個目(全)	52 ファイル中)		54	非表示
						SFTP-3	0:06	5:03

("/home/users/ユーザ名"と "/lustre/home/users/ユーザ名" は同一の場所です)



#### 以下のように、ホームディレクトリ下の CH3CI ディレクトリにファイルを 置きます。このファイルを使ってRCCSでGaussianを実行します。

@ccfep.ims	s.ac.jp - WinSCP					_		$\times$
ローカル(L) マーク(M) ファイル(	F) コマンド(C) セッション(S) オプ	ション(O) リモート(R) ヘルプ(H)						
🕀 😂 🗟 同期  🖉	🔁 🚳 🎒 キュー 🔹 転送設	定 デフォルト・	<i>S</i> -					
🧧 @ccfep.ims.ac.jp 🗙 🕻	🚰 新しいセッション							
🗖 デスクトップ 🔹 🚰 🗸	🝸 • 🗈 🖬 🏠 🎜 📘	→		📕 CH3Cl 🛛 🝷 🚰	🔹 🝸 🔹 🔁 🎧 🎧 🌮 🗟 ファイルの検索		· •	
📑 アップロード 🔹 📝 編集	<ul> <li>X 📝 🕞 プロパティ 📑 新</li> </ul>	所規 - ■ 🖌 🗕 🗸		■ 💀 ダウンロード 👻 🗋	🏾 編集 🔸 🗶 📝 🔂 プロパティ 📔 新規・	+ - \		
C:¥Users¥ De	esktop¥			/lustre/home/users/	/CH3Cl/			
名前	サイズ 種類	更新日時	^	名前	サイズ 更新日時	パーミッション	所有者	
				ch3cl.gjf	1 KB 2019/05/24 9:05:36	rw-r		
Ch3cl.gjf	1 KB GJF ファイル	2019/05/24 9:05:36		Gai	ussianインプット(PC	(く何)		
				Gat		(1)		
for an and the second								
and the second se								
(m.m.)								
and the second s								
Contraction (Section 1)								
A second second second								
Barrow and the second								
and the second second								
			<					
∪ B (全 149 MB 中)/ 0 個	日 (全 28 ノア1ル中)		1 非衣示	∪ B (全 248 B 中) /	01回日(全1ノア1ル中)	SETD-3	0.1	1.13
					•	JEIF-5	0.1	1.15



(今回は PuTTY を利用しています)

実際に計算を投入するためにフロントエンド(ccfep)にログインし、 cd CH3CI/ と入力の後 Enter を押し、データ設置場所(CH3CI)に移動します。



先ほど作成したディレクトリに cd コマンドで移動します。 Is コマンドでファイルの存在も確認できます。



#### CH3CI ディレクトリにて g16sub コマンドでジョブを投入します。 コマンドは g16sub」(インプットファイル名) です(」は半角スペース)。

Using username "\*\*\*". Authenticating with public key "\*\*\*\*\*\*\*\*\*\*\*\*\*" Passphrase for key "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*": Last login: Thu May 23 17:26:40 2019 from \*\*\*\*\*.\*\*\*.\*\*\* [user@ccfep8 ~]\$ ls CH3Cl/ ch3cl.gjf [user@ccfep8 ~]\$ cd CH3Cl/ [user@ccfep8 CH3Cl]\$ ls ch3cl.gjf [user@ccfep8 CH3Cl]\$ ls

オプションを指定しない場合は "core" ジョブタイプで 6 コアを用います。 計算の制限時間は 72 時間となっています。

g16subの詳細な使い方は利用の手引き、g16subをオプション無しで 実行した場合に表示されるヘルプ等でご確認ください。

利用の手引き: <u>https://ccportal.ims.ac.jp/QuickStart</u>

本資料の最後にも簡単な説明を載せています。



#### 実際に計算を投入すると、以下のように使用するコアの数、 メモリの最大量、制限時間等の情報が表示されます。(下図青色部分)

[user@ccfep8 QUEUE detai	3 CH3C1]\$ 	g16sub ch	3cl.gjf				
QUEUE (MACH)	Jobtype	MaxMem	DefMem	TimLim	DefCPUs(Mi	in-Max)	
PN( 1x)	core	4.8GB	4.0GB	72:00:00	6(1-36)	(キュー)	情報の表示
Job detail							
MOL name(s) INP file(s) OUT file(s) Current din SCRATCH din QUEUE Memory Time limit lob script	) : ch ) : ch ) : ch r : /1 r : /w : PN : 24 : 72	3cl.gjf.lx 3cl.out ustre/home, ork/users/ .0GB :00:00	/users/***/( \${USER}/\${P  /users/***/(	CH3C1 BS_JOBID}	24. sh	(ジョブ)	の情報)
Input modi	fied : y	=========	===========	===========	===================		
/usr/local/H	oin/jsub	-q PN /lus	tre/home/us	ers/***/CH3C	1/PN_28254.s	sh	
4529602.cccr [user@ccfep8	ns1 ◀ 3 CH3C1]\$		<b>—</b> ジ∃	ゴD			

最後の "数字.cccms1" の数字部分が"ジョブID"と呼ばれる固有番号です。 ジョブ投入に失敗すると、このIDの代わりにエラー内容が表示されます。

### <u>ジョブの状態確認</u>

jobinfo コマンドを実行すると投入したジョブの状態が確認できます。 (投入直後だと反映されない場合があります。その場合少しお待ちください) jobinfo\_-q\_PN\_-c\_-I が標準的な使い方になります。(\_」は半角スペース)

[user@ccfep8 CH3C1]\$ jobinfo -q PN -c -1							
Queue	Job ID Name	Status	CPUs	User/Grp	Elaps	Node/(Reason)	
PN	4529602 PN_28254.sh	Run (2)	6	***/	00:00:00	cccc120 <u>4</u>	
[user@							

終了していないジョブがあれば、ジョブIDとともに情報が表示されます。

- 1. ジョブの固有 ID です(全頁も参照)。ジョブを途中で止める時等に必要です。
- ジョブのステータスです。"Run" ならば実行中、"Queue"ならば他のジョブの 終了を待っている状況です。
- 3. 実行中(Run)の場合は現在までの実行時間です。待ち(Queue)の場合、 これまで待った時間が表示されます。
- 実行中の場合は実行している演算ノード名が表示され、実行されていない場合 は実行されていない理由が表示されます。例えば、(cpu)の場合は空きCPUが ないため実行できないことを示します。 (投入直後は (other) と表示される場合があります)



ジョブの終了後、ディレクトリには以下のようなファイルが残っています。 PN\_で始まるファイルと ch3cl.gjf.lx は g16sub に作られたファイルで、 計算が正常に終了した場合は気にしなくても大丈夫です。

[user@ccfep8 CH3C1]\$	ls		
PN_28254.sh	PN_28254.sh.o4529602	ch3cl.gjf	ch3cl.out
PN_28254.sh.e4529602	ch3cl.chk	ch3cl.gjf.lx	
[user@ccfep8 CH3C1]\$			

出力ファイル ch3cl.out についてはジョブの実行中にも中身が確認できます。 less や tail 等のコマンドで途中経過を確認することができます。

今回のインプットのようにチェックポイントファイル(.chk)を指定すれば、 ファイル(ch3cl.chk)もこの場所に作られます。

### <u>formchk の実行</u>

計算で得られたチェックポイントファイル(.chk)をformchkコマンドで テキスト形式(.fchk)に変換する場合、RCCSではログインシェルに 応じて事前に以下のどちらかのコマンドを実行する必要があります。

<u>csh の場合 (/bin/csh)</u>

[user@ccfep8 CH3Cl]\$ source /local/apl/lx/g16/g16/bsd/g16.login

<u>bash/zsh の場合 (/bin/bash or /bin/zsh)</u>

[user@ccfep8 CH3Cl]\$ source /local/apl/lx/g16/g16/bsd/g16.profile

上記コマンドを実行しても何も起こっていないように見えるかも しれませんが、実際には内部設定が更新されています。

これで晴れて formchk が 実行できます。 [user@ccfep8 CH3C1]\$ formchk ch3cl.chk ch3cl.fchk Read checkpoint file ch3cl.chk Write formatted file ch3cl.fchk FChkPn: Coordinates translated and rotated FChkPn: Coordintes match /B/ after translation and rotation [user@ccfep8 CH3C1]\$

## <u>ジョブ実行のヒント(1): g16sub オプション</u>

- •g16sub デフォルト設定
  - "core" ジョブタイプ (-j core)
    - ・ 単一ノードでの実行です。同一ノードで他のジョブと共存する可能性があります。
  - 6 CPU を利用 (-np 6)
  - ・72 時間の実行時間 (-walltime 72:00:00)
  - ・デフォルトの Gaussian バージョンは年度によって異なります
    - 2019年度は Gaussian 16 Rev. B.01 を利用 (-rev g16b01), 2020-2021年度は Gaussian 16 Rev. C.01 を利用 (-rev g16c01)します
      - (-rev g16a03 を指定すると Gaussian 16 Rev. A.03 も利用できます)
  - Gaussianインプットファイルだけを指定し、オプションを全く指定しない場合 (g16sub (インプット))は以下と同じ動作になります。
    - 2020-2021年度の場合 g16sub -j core -rev g16c01 -np 6 -walltime 72:00:00 (インプット)
  - ・オプション値を変えれば利用 CPU 数などを変更できます

# <u>ジョブ実行のヒント(2): 別バージョンの利用</u>

- •現在、RCCSでは以下のバージョン、リビジョンの Gaussian が利用 可能です。g16sub, g09sub の -rev オプションで指定できます。
  - (カッコ内が -rev で指定する場合のキーワードです。)
  - Gaussian 16 Rev. C.01 (g16sub -rev g16c01; g16sub デフォルト)
  - Gaussian 16 Rev. B.01 (g16sub -rev g16b01)
  - Gaussian 16 Rev. A.03 (g16sub -rev g16a03)
  - Gaussian 09 Rev. E.01 (g09sub -rev g09e01; g09sub デフォルト)
  - Gaussian 09 Rev. D.01 (g09sub -rev g09d01)
  - Gaussian 09 Rev. C.01 (g09sub -rev g09c01)
  - Gaussian 09 Rev. B.01 (g09sub -rev g09b01)

### <u>ジョブ実行のヒント(3): コア数の指定</u>

- (以下、基本的に jobtype = core を想定しています。)
  - RCCS では Linda を導入していないため、jobtype = small や large を使っても Gaussian の多ノード並列計算はできません。
- Gaussian計算において、コア数は増やせば増やすほど速度が上がると は限りません。増やし過ぎると遅くなることもありえます。
- CPU点数あたりの計算効率という観点からもコア数は抑え目にした方が経済的です。
- ついでに、計算の投入されやすさという観点では利用するコア数が少ない方が確実に有利です。大きな計算は待たされやすくなります。
- ・さりとて、コア数を抑えすぎると今度は中々終わらなくなります...
- 「適切なコア数」はプログラムのベンチマーク的な数値だけで決まる
   ものではありません。状況に応じてご判断ください。
  - •例1) 査読への返答で急を要するため、多少CPU点数効率が悪くてもできるだけ速度が出る条件で実行する
  - 例2)3日後までに終わってくれればいいので、コア数を少なめにして効率よく、 確実に計算を実行する。

## <u>ジョブ実行のヒント(4): 作業ディレクトリ</u>

作業ディレクトリ(SCRATCH)は実際のジョブ投入時まで確定しません。 g16sub 実行時の表示はジョブ投入前の段階で判明している情報です。

OUT file(s)	: ch3cl.out
Current dir	: /lustre/home/users/***/CH3Cl
SCRATCH dir	: /work/users/\${USER}/\${PBS_JOBID}

実際には、\$USER はご自身のユーザID(3文字ログインID)に置き換えられ、 \$PBS\_JOBID はジョブ投入時に表示される文字列に置き換えられます (前ページの例では 4529602.cccms1)

ch3cl.out:

Entering Gaussian System, Link0=/local/apl/lx/g16b01/g16/.\_.g16
Initial command:
/local/apl/lx/g16b01/g16/l1.exe "/work/users/\*\*\*/4529602.cccms1/Gau-33628.inp"
-scrdir="/work/users/\*\*\*/4529602.cccms1/"
Default CPUs for threads: 6,7,8,9,10,11
Default is to use a totoal of 6 processors:

実際の作業ディレクトリは出力ファイル(ch3cl.out)冒頭でも確認できます。 g16sub, g09sub で投入したジョブが時間内に終了した場合、 終了時に作業ディレクトリは消去されますのでご注意ください。

### <u>ジョブ実行のヒント(5):g\*\*sub以外の方法</u>

- •g09sub,g16subを使わずに実行することも可能です。
- いくつか方法がありますが、/local/apl/lx/g16c01/samples 以下のサン プルを改変する方法が一番手軽です。 (g16c01の部分は適宜他の revision に置き換えて下さい)
  - CPU数やGPU数は GAUSS\_CDEF, GAUSS\_GDEF で指定してください。
  - メモリ量については -m コマンドラインオプション、GAUSS\_MDEF でユーザ がご自身で指定してください。(%Mem で指定する場合には ---noedit で自動置 き換えを抑える必要があります)
- •g16sub, g09sub に -P オプションをつけた場合、ジョブの投入直前の 段階で止まります。ここで作られたファイルをベースに改変すること も可能です。
  - ジョブスクリプト(PN\_\*.sh ファイル) は上記 samples/以下にあるものと同じ ような構造になっています。