FOCUSスパコンでの Gaussian 09 利用講習会

計算科学振興財団 2015年4月 23日 (木) 13:00~15:00

プログラム

0. FOCUSスパコンでのGaussian 09の環境設定と実行

1. ノード内並列の効率的実行方法

1-1 FOCUSスパコンでのGaussian 09の環境設定と実行

1-2 ジョブ投入サンプルスクリプト

1-3 並列数(%Nprocshared=)とメモリ指定(%mem=)

1-4 構造最適化からの振動数計算の手続き

2. Lindaを用いたノード間並列の効率的実行方法

2-1 Linda 並列実行サンプルスクリプトの解説

2-2 Model・property による Linda 並列の有効性

2-3 構造最適化からの振動数計算の手続き

2-4 %NProcSharedを入力ファイルに直接指定するLinda 並列実行サンプル スクリプトの解説

3. SCF が収束しない場合の対処法

4. 構造最適化が収束しない場合の対処法

5. 質疑応答

応用編にて解説 (基礎編では略します)

0. FOCUSスパコンでの Gaussian 09 の 環境設定と実行

0.1 FOCUSスパコンシステム構成



0.2 環境変数GAUSS_SCRDIRの指定 /work の使用を推奨

- 環境変数「GAUSS_SCRDIR」ではGaussianの作業ファイル(Gau-プロ セスID.拡張子)が作成されるスクラッチディレクトリを指定します。 ジョブが終われば自動的に消されます。
- /workは計算ノードのローカルディスクであり、ホームディレクトリ/ home1に比べて高速に書込みが可能です。
- 一方、作業中のカレントディレクトリとしてホームディレクトリ/ home1上のディレクトリを使用すると書込みが遅く、同時に多数の プロセスで/home1に書き込むと、この書き込み速度をそれらのプ ロセスで分け合うことになり、性能が劣化する可能性があります。 このことから/workの使用を推奨します。
- /workの容量については別途、表にまとめます。

0.3 ファイルシステムの容量と性能

システム	キュー名	書き込み速度	容量
A B C	a024h, a096h b024h, b096h c024h, c096h	75MB/s前後	400GB未満
D	d006h, d012h	200MB/s前後	2TB未満
D	d024h, d072h	300MB/s前後	6TB未満
E	e024h, e072h	200MB/s前後	2TB未満

システム	ファイル システム	書き込み速度	容量
А, В, С	/home1 /home2	75MB/s前後 1GB/s/ノード前後	400GB未満
D, E	/home1 /home2	100MB/s前後 1GB/s/ノード前後	200GB未満 契約容量未満

0.4 ジョブ管理コマンド (1/2)

実行可能なキューの確認 sinfo -s

[uleg0001@f	F002 ~]S	\$ sinfo -s			
PARTITION	AVAIL	TIMELIMIT	NODES(Activ/Ido	le/Other/Ttotal)	NODELIST
a024h	up	1-00:00:00	44/66/6/116	a[001-116]	
a096h	up	4-00:00:00	44/50/6/100	a[001-100]	
b024h	up	1-00:00:00	0/2/0/2	b[001-002]	
b096h	up	4-00:00:00	0/1/0/1	b001	
c024h	up	1-00:00:00	3/11/0/14	c[005-018]	
c096h	up	4-00:00:00	3/2/0/5	c[005-009]	
d006h	up	6:00:00	23/78/11/112	d[001-080],e[001	-032]
d012h	up	12:00:00	23/78/11/112	d[001-080],e[001	-032]
d024h	up	1-00:00:00	23/54/3/80	d[001-080]	
d072h	up	3-00:00:00	19/19/2/40	d[001-040]	
e024h	up	1-00:00:00	0/24/0/24	e[009-032]	
e072h	up	3-00:00:00	0/8/0/8	e[009-016]	
debug6m*	up	6:00	0/4/0/4	g[001-004]	

上記以外のキューは期間占有、短期期間占有(日単位予約)による キューで自分が所属する課題以外のものは利用できません。

0.4 ジョブ管理コマンド (2/2)

・ジョブ投入コマンド sbatch

[uleg0001@f002 ~]\$ sbatch スクリプトファイル

実行中のジョブの確認 squeue

[uleg0001@ff02 ~]\$ squeue
JOBID PARTITION NAME USER ST TIME NODES NODELIST(REASON)
xxxx d024h testrun uleg0001 R 4:58 60 d[007-066]
yyyy debug6m testrun uleg0001 CG 0:39 1 g001

・ジョブのキャンセル scancel

[uleg0001@ff02 ~]\$ \$ scancel ジョブID ジョブID ジョブID [uleg0001@ff02 ~]\$ scancel xxxx yyyy

0.5 課金の確認(1/3)

ユーザー課金確認 uacct

[uleg0001@ff02 share]\$ uacct Charge information of uleg0001 in 201504 : Computational: 8610.00 (Rack rate: 8610)

Items

A:	9 nodehours,	1043.00 yen
B:	0 nodehours,	0 yen
C:	90 nodehours,	7567.00 yen
D:	0 nodehours,	0 yen
E :	0 nodehours,	0 yen (0 yen)

Total charge : 11695.00

ユーザー単位の情報 引数(yyyymm)追加で過去の情報閲覧可能

0.5 課金の確認(2/3)

ユーザー課金確認 uacct_apl

[uleg0001@ff02 share]\$ uacct_apl Charge information of uleg0001 in 201504 : Computational: 0 (Rack rate: 0) Items Gaussian 09: 48 nodehours, 5766 yen MIZUHO/BioStation: 0 nodehours, 0 yen Parallel CONFLEX: 0 nodehours, 0 yen

Total charge : 5766.00

ユーザー単位の情報 引数(yyyymm)追加で過去の情報閲覧可能

0.5 課金の確認(3/3)

グループ当月課金確認 thismonth

[uleg00	01@ff02 ~	~]\$ thism	onth						
Charge	informati	ion of le	g in this	month.					
system	njob	avg_et	avg_nodes	avg_procs	et_nodes	charge	et_max	et_np_max	np_max
A	7	2.2	6.9	75.4	622.2	46667.0	15.6	7466.8	480
В	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
С	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
D	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
Е	1	0.0	10.0	200.0	0.0	30.0	0.0	0.2	200

TOTAL: 46697 yen

グループ単位の情報 引数(yyyymm)追加で過去の情報閲覧可能

0.6 Gaussian09利用料金@FOCUS

1)ソフトウェア利用料金(ノード時間あたり120円)

= 全CPU時間/ノード内コア数 * 120円 (ノード時間単価(税抜))

=120円 * 全CPU時間/12 (A,Cシステム)
=120円 * 全CPU時間/16 (Bシステム)
=120円 * 全CPU時間/20 (D,Eシステム)

2) 計算資源利用料金

- Aシステム (~100円(並列割引制度あり))
- Bシステム (100円)
- Cシステム (80円)
- Dシステム (~300円(並列割引制度あり))

Eシステム (600円)

ノード内並列しないと損!

1. ノード内並列の効率的実行方法

A,C,DシステムでのGaussian09利用料の概算計算例

Gaussian09 出力ファイル					
[uleg0001@ff02 g09test]\$ tail input0.log					
.01\State=1-A\HF=-301.5468797\RMSD=5.008e-09\Dipole=-0.0002927,0.01423					
84,-0.0316404\Quadrupole=0.464593,1.7311805,-2.1957735,-5.870745,-1.78					
33127,2.6608152\PG=C01 [X(H4O4)]\\@					
Any fool can criticize, condemn, and complain and most do.					
Dale Carnegie					
Job cpu time: 0 days 7 hours 25 minutes 7.9 seconds.					
File lengths (MBytes): RWF= 19 Int= 0 D2E= 0 Chk= 2 Scr= 1					
Normal termination of Gaussian 09 at Fri Feb 7 15:46:00 2014.					

Job cpu time: 0 days 7 hours 25 minutes 7.9 seconds.の値をコア数で割ると、 A, C: (7+25/60+7.9/3600)/12=0.5835 [ノード時間] D : (7+25/60+7.9/3600)/20=0.3709 [ノード時間] Aシステム (100/10+120)*0.5835=128 [円] Cシステム(80/10+120)*0.5835=116[円] Dシステム (300/15+120)*0.3709=51[円] ノード内並列を実行するならばDシステムが低コストです。

1-1. FOCUSスパコンでのGaussian 09の環境設定と実行

フロントエンドサーバにログインし、SLURMを利用してバッチジョブとしてGaussian 09の計算を 実行する方法を紹介します。

利用したいGaussian 09の環境設定の選択

SLURMのバッチスクリプト内に記載してシェルの環境設定とは異なるGaussian 09の 実行モジュールを使い分けることもできます。

SLURMバッチスクリプト記載例

#export g09root=/home1/share/g09
#export g09root=/home1/share/g09c01
#export g09root=/home1/share/g09d01s
export g09root=/home1/share/g09d01s

デフォルトのGaussian(現在はg09d01s のリンク)

Gaussian 09 Rev. C01 PGIコンパイラビルド バイナリ配布版 # Gaussian 09 Rev. D01 PGIコンパイラビルド バイナリ配布版 # G09 Rev. D01 PGIコンパイラソースコードビルド版

異なる G09を利用できます

export g09root=/home1/share/g09d01s_intel # G09 Rev. D01 intel コンパイラソースコードビルド版

ここでは intel版を読み込んでいます。 (不必要な設定はコメントアウトしています)

1-2. ジョブ投入サンプルスクリプト

/home1/share/g09 にジョブ投入サンプルスクリプト(逐次計算、ノード内並列、ノード間 並列(Linda))を置いております。これらを参考にジョブ投入スクリプトを作成してください。 以下はノード内並列用サンプルスクリプトg09sample.sh です。

[uleg0001@ff01~]\$ cat g09sample.sh #!/bin/bash # キュー名 #SBATCH -p d024h # 最大プロセス数 #SBATCH -n 20 **#SBATCH** -J Gaussian SHARED # ジョブ名 # 標準出力、%J はジョブID に置換 #SBATCH -e Gaussian SHARED.e%J # 標準エラー出力、%J はジョブID に置換 #q09d01s Intel Compiler versionを利用するため #SBATCH -o Gaussian SHARED.0%J # Intelコンパイラの環境変数を設定します。 # Intel コンパイラ環境変数設定 module load PrgEnv-intel export g09root=/home1/share/g09d01s intel # g09root ディレクトリの指定 #g09d01s intel版で計算に不具合が出た場合はg09d01sに置き換えてください。 #module load PrgEnv-pgi-1210 #export g09root=/home1/share/g09d01s # Gaissian 環境変数設定 . \$g09root/g09/bsd/g09.profile # 以下では\$MOL.comの中でメモリとプロセス数を適切に指定している事を仮定 # 入力ファイル名の指定 MOL=test178 # 以下、スクラッチディレクトリを指定 export GAUSS SCRDIR=/work/ # Gaussian を実行と時間計測 time q09 <\$MOL.com > \$MOL.log

sbatch コマンドを使ってジョブ投入スクリプト(例:g09sample.sh)を実行します。 \$ sbatch g09sample.sh

1-3 並列数(%Nprocshared=)とメモリ 指定(%mem=)

•ノード単位課金のため、1つのGaussian 09ジョブではノード 内コア数全てを利用した方が得です。

ノード内コア数を%NprocSharedに指定した並列計算を実 行してください。

・全てのコアを利用するよう適切なメモリサイズを指定しなければなりません。1コアの計算に必要なメモリ量をノード内コア数倍した値を%memに指定してください。

1-3 並列数(%Nprocshared=)とメモリ 指定(%mem=)(2)

• %NprocShared=12と%mem=72MWと指定して

PrsmSu: requested number of processors reduced to: 9 ShMem 1 Linda と出力ファイルに記録されているなら、12/9*72MW=96MWを指定します。

%memで指定するメモリ量は多過ぎても少なくても性能が劣化することがあります。以下の図はCCSD(T)一点計算で%NprocShare=12を指定した時、様々なメモリ割り当ての場合の性能例です。

