FAQ (一般・分子科学)

(最終更新: 2025/7/10)

センターの計算機利用に関するよくある質問集です。

FAQ

- パスワード初期化がうまくいかない/3文字のユーザーアカウント？
- 利用中にメールアドレスを変更したい
- ログインできない/海外出張中にログインしたい
- CPU点数やディスク使用量の制限値を超えると何が起きるのか
- ジョブのCPU点数の計算方法について
- ジョブで利用可能なメモリ量について
- メンバーの追加、削除をしたい
- プログラムのビルドやテスト、デバッグはどこで行えばいいのか
- 利用できるCPU数やGPU数の制限値を知りたい
- 資源(CPU点数、ディスク容量)追加を申請したい
- ログインシェルを変更するには
- ジョブがなかなか実行されない
- Cgroup mem limit exceeded メッセージと対応について
- No space left on device エラーについて
- 「障害発生と影響を受けたジョブに関するお知らせ」メールについて
- Received message too long と表示されて sftp 接続ができない(WinSCP 等含む)
- Gaussianのchkファイルを可視化のために変換したいがformchkが無い
- formchk実行時にメモリ不足と思われるエラーで失敗する
- Python環境(本体バージョン、ライブラリ等)構築について
- インテルコンパイラ(ifx, ifort, icx, icpx)を使いたいが存在しない
- 利用申請/事前準備が必要なソフト(ORCA, GRRM23, Open OnDemand, Crystal, AlphaFold3)
- ジョブスクリプトのヘッダ部分のサンプルが欲しい
- ソフトウェア導入の要望

パスワード初期化がうまくいかない/3文字のユーザーアカウント？

このウェブサイトで初期パスワード設定や公開鍵設定をするためには申請時に登録されたメールアドレスのみが必須です。3文字のユーザーアカウントは初期パスワード設定後の情報修正やスパコンシステムへのログイン時に必要になります。

このパスワード初期化時のメール送信がうまくいかない場合は、入力したメールアドレス、あるいは登録されているメールアドレスが間違っていることになります。メールがどうしても届かないようであればお問い合わせください。なお、申請時に登録されたメールアドレスは、課題代表者ならばNOUSにある申請書の「組織(Members)」ページ内にある「組織表ダウンロード」ボタンから確認が可能です。申請時にユーザーアカウントの欄を空欄のまま提出した場合にも採択後にメンバーのアカウント名が追記されるようになっています。

3文字のユーザーアカウントについては登録の最中にメールの内容にて確認することができます。あるいは、グループ内に既にログインできているユーザーがいれば、本ウェブサイト上のグループメンバーの一覧ページ(要ログイン)よりグループメンバーの 3 文字アカウント名を確認することもできます。ご自身が正しいグループに登録されているかどうかについてもこちらで確認が可能かと思います。

利用中にメールアドレスを変更したい

年度の途中でのメールアドレス変更はこのウェブページ上で行うことはできません。メンバー管理申請ページにある様式に新しいメールアドレスを入力の上、所定の方法でご提出ください。

申請時のメンバーのメールアドレスについては、課題代表者であればNOUSの申請書の「組織(Members)」ページから「組織表ダウンロード」ボタンより確認できます。

ログインできない/海外出張中にログインしたい

日本国外からのアクセスについてはこちらのページにある手続きが必要です。日本国内からのアクセスで問題がある場合はまずクイックスタートガイドの内容をご確認ください。

■ 接続先ホストとユーザー名

接続先ホストは ccfep.ims.ac.jp です。ユーザー名は RCCS から指定された３文字のアカウント(英字のみもしくは英字+数字)となります。

■ 公開鍵の登録

ログインの前に公開鍵を登録する必要があります。公開鍵の登録はこのウェブサイト上で行うことができます。登録方法についてはこちらのページをご確認ください。

■ ホームディレクトリ等のパーミッションに問題がある場合

ホームディレクトリや ~/.ssh, ~/.ssh/authorized_keys のパーミッションを誤って変更してしまった場合、ログインができなくなる場合があります。どうにもならなくなってしまった場合は管理者権限で変更を行いますのでお問い合わせください。

■ 接続できずにタイムアウトする/即座に接続が切断されてしまう

障害時にはこのような状況になる可能性があります。障害に関しては当ウェブサイトのトップページに情報が掲載されますのでまずはそちらをご確認ください。また、メンテナンス時もログインはできませんので、メンテナンスの終了までお待ちください。

Permission denied というメッセージが出る場合は接続できたものの公開鍵やユーザー名などに不備がある可能性が高いと思います。クイックスタートガイドなどを参考に設定の見直しをお願いします。

何のレスポンスもなくずっと待たされる場合、Connection timed out や Connection closed などのメッセージがすぐに表示される場合にはファイアウォール等で通信が遮断されている可能性があります。この場合はご利用されているネットワーク環境の担当者に一度お問い合わせください。

CPU点数やディスク使用量の制限値を超えると何が起きるのか

新規ジョブが投入できなくなります。
グループの CPU 点数を超過した場合、超過から 24 時間以上経過すると投入済みのジョブも削除されます。

使用量がシステム側に認識されるまで(showlim の表記に反映されるまで)には最大 10 分程度の時間がかかります。ディスク使用量の制限値を超過したためにデータを削除した場合、削除後 10 分程度待った上で新規ジョブの投入を行うようにしてください。

ジョブのCPU点数の計算方法について

ジョブのCPU点数は実際の実行時間から計算されます。例えばジョブの制限時間(walltime)が 24 時間で実際の実行時間が 1 時間だった場合、1 時間分の CPU 点数が消費されることになります。

ジョブで利用可能なメモリ量について

ジョブで利用可能なメモリ量は利用する CPU コア数に比例します。
largemem では 1 コアあたり約 8 GB (~7.45 GiB)を、largemem 以外では 1 コアあたり約 2 GB (~1.86 GiB)を使えます。64 コア(ncpus=64)を使った場合は largemem で 512 GB 程度, それ以外では 128 GB 程度が上限となります。CPU をあまり使わないものの大量のメモリを要する場合も、ncpus の値を大きくして対応してください。

メンバーの追加、削除をしたい

こちらのページに申請方法があります。

プログラムのビルドやテスト、デバッグはどこで行えばいいのか

ビルド作業やデバッグについてはログインサーバー(ccfep*)でそのまま行って下さい。GPUを使うプログラムの場合は ccgpu (ccfep から ssh ccgpu を実行すればログインできます)で行うことができます。並列 make (例: make -j 12 のようなコマンド)や MPI プログラムの実行(例: mpirun -np 16 ...)も問題ありません。

ただし、速度面についてはいくつか制約もありますので、その点はご注意下さい。また、ccfep* では CPU の利用状況に応じて CPU 点数も消費されます。(ただログインしているだけであれば事実上消費はありません。)

インタラクティブジョブについては、申請が認められれば利用可能となっております。利用を希望される場合はお問い合わせください。なお、インタラクティブジョブにはいくつか追加の制限(制限時間、CPU/GPU数)もありますのでご注意ください。

利用できるCPU数やGPU数の制限値を知りたい

制限値は原則として初期の割り当て点数から決められます。具体的な値はこちらの表をご覧下さい。

現在の制限値は jobinfo -s を実行すると上の方(User/Group Stat)に表示されます。以下の表記中、赤や青の **** で表示されている部分にあたります。

```
[user@ccfep* ~]$ jobinfo -s

User/Group Stat:
-----------------------------------------------------------------------------
 queue: H                    | user                 | group
-----------------------------------------------------------------------------
   NJob (Run/Queue/Hold/RunLim) |    x/    x/    x/-    |    x/    x/    x/****
   CPUs (Run/Queue/Hold/RunLim) |    x/    x/    x/**** |    x/    x/    x/****
   GPUs (Run/Queue/Hold/RunLim) |    x/    x/    x/**** |    x/    x/    x/****
   core (Run/Queue/Hold/RunLim) |    x/    x/    x/**** |    x/    x/    x/****
   lmem (Run/Queue/Hold/RunLim) |    x/    x/    x/-    |    x/    x/    x/***
 sncore (Run/Queue/Hold/RunLim) |    x/    x/    x/-    |    x/    x/    x/***
-----------------------------------------------------------------------------
note: "core" limit is for per-core assignment jobs (jobtype=core/gpu*)
note: "lem" limit is for jobtype=largemem
note: "sncore" limit is for 64 or 128 cores jobs

Queue Status (H):
(以下略)
```

制限値にはグループ全体のものと、個々のユーザのものがあります。個々のユーザーへの制限については、グループの代表者または権限を与えられたメンバーであればリソース制限のページより設定できます。また、jobtype=core, largemem については全ユーザー共通の制限値があり、それも表示されています。

資源(CPU点数、ディスク容量)追加を申請したい

こちらのページに申請方法があります。いくつか制限もありますので、ご注意ください。

ログインシェルを変更するには

現在はログインシェルを csh (tcsh), bash, zsh の中から選ぶことができますが、その変更はこのウェブページでしか行えません。ログインしている状態であれば、ページ上部に「アカウント情報」という項目が存在するので、それをクリックしてください。そこから、「編集」タブに移動すると、ログインシェルを選択する項目がありますので、csh, bash, zsh の中から好きなものを選んでください。ログインシェルを選択したあとは忘れずにページ下部にある「保存」をクリックしてください。なお、変更が実際に反映されるまでには最大で 20 分を要します。

ジョブがなかなか実行されない

ジョブが中々実行されない原因についてはいくつかのパターンが考えられます。以下に当てはまらない不明な点等ありましたお問い合わせ下さい。

■ 混雑していてリソース**(CPU, GPU)**が足りない

CPUやGPUの空き状況についてはこのウェブサイトにログインしてトップページを表示すると、右側のカラムに空き状況が表示されます。混雑していてジョブが入らない状況でも、他の jobtype では空きがある場合もあります。もし可能であれば、ジョブの振り替えもご検討下さい。

■ リソースに空きがあるのにジョブが流れない

jobinfo コマンドを実行し、右端のカラムに表示される理由をご確認下さい。主要な表示は以下のようなものです。

- (cpu), (gpu), (cpu/gpu): CPU や GPU の空きが無いことを示します
- (group): グループの制限値を超えるため、ジョブが投入できません
- (long): walltime が長すぎるため実行できません(ジョブは次回のメンテナンスまでに確実に終了する必要があります)
- (other): その他の状況です。投入直後のジョブにもこれが表示されますが、その場合は少しお待ちください。
    - -c をつけて (other) が表示される時、-c を外すことで正しい理由が表示される場合があります。

(group)の場合は、ご自身もしくはグループの他メンバーが多く利用しているため、制限されている状態です。代表者や権限を与えられたメンバーならばグループの個々のメンバーのリソース使用量に制限をかけることもできます(リソース制限)。

(long)の場合は、walltime を短くするか、さもなくばメンテナンス明けまで待つ必要があります。(通常の月例メンテナンスではジョブの消去は行いませんので、メンテナンス時にジョブが残っていても問題ありません。)

■ 空きコアの数は多いのにジョブが流れない(**jobtype=core**の場合)

多数のコアを使う jobtype=core のジョブは空きコアがあっても流れない可能性があります。jobtype=core では複数ノードにまたがったジョブは許されていないので、空きコアの総数が多くても、大きな空きがあるノードが存在しない場合があるためです。状況に応じて jobtype=vnode で 64 or 128 コアを利用することもご検討下さい。

Cgroup mem limit exceeded メッセージと対応について

制限を超えてメモリを使った場合、以下のようなメッセージが標準エラーに出力され、ジョブが異常終了します。

```
Cgroup mem limit exceeded: oom-kill:constraint=CONSTRAINT_MEMCG,nodemask=
(null),cpuset=*******.ccpbs1,mems_allowed=*,oom_memcg=/pbs_jobs.service/jobid/*******.ccpbs1,task_memcg=/pbs_jobs.service/jobid/*******.ccpbs1,task=molpro.exe,pid=******,uid=******

[Wed Feb  1 **:**:** 2023] memory: usage ********kB, limit *********kB, failcnt ********
[Wed Feb  1 **:**:** 2023] memory+swap: usage *********kB, limit *****************kB, failcnt 0
[Wed Feb  1 **:**:** 2023] kmem: usage ******kB, limit *****************kB, failcnt 0
[Wed Feb  1 **:**:** 2023] Memory cgroup stats for /pbs_jobs.service/jobid/*******.ccpbs1:
```

インプットを工夫して使用するメモリ量を減らす、jobtype=largemem を利用するなどして回避する必要があります。jobtype=core の場合は利用可能なメモリ量が ncpus の指定に比例することにご注意ください。使用するプログラムの性質によっては、ヘッダで ncpus=8 で 8 コア分のメモリを確保した上で mpirun -np 4 のように 4 コアだけ計算に使うようなことも考える必要があるかもしれません。

確保したメモリが超過していても、超過分が使われていない場合には Cgroup mem limit exceeded が表示されるもののジョブは正常に実行できる場合があります。この場合、Cgroup mem limits exceeded だけが表示され、それに続くプロセスの情報は表示されません。潜在的にはジョブがエラーになる可能性がありますので、確保するメモリ量を減らす等の対応を行っておくべきだと思われます。

No space left on device エラーについて

計算ノードのローカルディスク(/lwork/users/(ユーザー名)/(ジョブID))を使っていて、その容量制限を超過してしまった可能性が高い状況です。計算ノードのローカルディスクの利用可能な容量は 11.9 GB * ncpus (ncpus はそのノードで確保したコア数; ジョブ全体での ncpus ではありません)となるため、ncpus を増やすことでエラーを回避できる可能性があります。あるいは、ローカルディスク(/lwork)のかわりに速度は落ちるものの大容量なグローバル領域(/gwork)を使って回避することもできます。

g16sub/g09sub の場合、デフォルトではローカルディスク(/lwork)を使いますが、実行時に -N オプションを追加することでグローバル(/gwork)領域を使うように変更できます。

「障害発生と影響を受けたジョブに関するお知らせ」メールについて

計算ノードやネットワークスイッチ、ストレージのダウン等によりジョブが影響を受けた場合、ジョブを実行したユーザー宛にこのようなタイトルのメールが送られます。カッコでくくられた部分や***となっている部分はユーザー、ジョブ、発生日時、障害の内容に依存します。

```
(uid)様

2024年**月**日**時**分頃、****において
障害(****************)が発生しました。

影響を受けたジョブの一覧をお知らせします。
--------+--------+--------+--------------+-----+------------------
Job ID   Jobtype  User     Job Name       #Core Treatment
--------+--------+--------+--------------+-----+------------------
 (jobid) *****    (uid)    (jobname)       (num)XXX
--------+--------+--------+--------------+-----+------------------
```

このメールは表示されているジョブが障害の影響を受けたことを示しますが、上記では赤文字で示したジョブに対するアクションの値に注意する必要あります。

- 異常終了(Abort) => ジョブがエラー終了しました。リラン(Rerun)に失敗した場合も含みます。障害発生までに消費されたCPU点数は原則払い戻されます(払い戻し作業が少し遅れる場合はあります)。
- リラン(Rerun) => 最初からジョブの実行をやり直します。それまでに使われた CPU 点数は無視されます。

通常の設定であれば、異常終了時にはジョブは再実行(リラン, Rerun)されます。ただし、障害の影響のためにシステムが正確に判断できず、そのままジョブが終了する場合があります。その場合はお手数ではありますがジョブを再度投入下さい。また、ジョブの種類によっては強制終了された前のジョブの中間ファイル等の影響で再実行がうまくいかないケースもありえます。こちらについては、ジョブスクリプトを工夫して中間ファイルを適宜消去するなどの処理で再実行されても大丈夫な形にしたり、#PBS -r n をスクリプトに加えて再実行を抑制したりして対応するようお願いいたします。

Received message too long と表示されて sftp 接続ができない(WinSCP 等含む)

.bashrc が出力するデータが大きい場合に上記のようなエラーが出る場合があります。例えば、Intel oneAPI の setvars.sh を .bashrc で読み込んだ場合に発生することが確認されています。エラーを回避するためには出力を /dev/null に捨てる、$PS1 が存在する場合にのみ実行するようにする、あるいは問題となっている処理を

.bash_profile に移動させるような対応が考えられます。

Gaussianのchkファイルを可視化のために変換したいがformchkが無い

以下のコマンドを実行してからformchkコマンドを実行してください。ログインシェルの種類により実行するコマンドが異なることに注意してください。

ログインシェルが bash, zsh の場合:

```
$ source /apl/gaussian/16c02/g16/bsd/g16.profile
```

ログインシェルが csh (tcsh) の場合:

```
$ source /apl/gaussian/16c02/g16/bsd/g16.login
```

(PYTHONPATH: Undefined variable. というメッセージが表示された場合も設定は読み込まれています。そのまま formchk を実行できます)他のバージョンの Gaussian や違うキューを用いた計算等の場合には適宜ディレクトリの場所を置き換えてください。また、上記設定の代わりに gaussian の module を load しても formchk コマンドは利用可能です。

formchk実行時にメモリ不足と思われるエラーで失敗する

メモリが足りない場合、formchk が以下のようなメッセージでエラー終了します。

```
Out-of-memory error in routine WrCIDn-* (IEnd= ******** MxCore= ***********)
Use %mem=***MW to provide the minimum amount of memory required to complete this step.
Error termination via Lnk1e at (***date***).
```

formchk の使うメモリ量は GAUSS_MEMDEFで指定できますので、以下のように設定の上(値は適宜調整して下さい)、formchk を再度実行ください。

bash/zsh:

```
export GAUSS_MEMDEF=400MW
```

csh:

```
setenv GAUSS_MEMDEF 400MW
```

上記の例では 400MW (=3200MB) のメモリを指定しています。export GAUSS_MEMDEF=3200MB のようにバイト(Byte)単位でも指定できます。

Python環境(本体バージョン、ライブラリ等)構築について

ディストリビューション提供のパッケージが存在すれば管理者側でそれらを /usr 以下にインストールすることはできます。必要に応じてお問い合わせください。あるいは、以下の例のように pip3 に --user オプションをつけてインストールすれば、ご自身のホームディレクトリ以下にパッケージを導入することもできます。

```
$ pip3 install numpy --user
```

他では pyenv や miniforge を使った環境構築も推奨しています。ライセンス上問題がなければ、Anaconda も便利です。GPU の利用を想定している場合は、miniforge のような conda 環境の方が便利なことが多いです。(nvidia チャンネルに cuda のパッケージがあります)

センターで用意した Miniforge 環境もあります。/apl/conda/(導入日時)/conda_init.sh もしくは /apl/conda/(導入日時)/conda_init.csh を source することで読み込めます。

インテルコンパイラ(ifx, ifort, icx, icpx)を使いたいが存在しない

現在のシステムでは、ユーザーの皆様にインテルコンパイラを提供しておりません。必要な場合は以下のリンクより Intel oneAPI Base Toolkit と HPC Toolkit をホームディレクトリ以下に導入してください。
https://www.intel.com/content/www/us/en/developer/tools/oneapi/base-toolkit.html (Base Toolkit)
https://www.intel.com/content/www/us/en/developer/tools/oneapi/hpc-toolkit.html (HPC Toolkit)
(無償で導入できます。ただし、再配布には制限があります。)

利用申請/事前準備が必要なソフト(ORCA, GRRM23, Open OnDemand, Crystal, AlphaFold3)

ORCA, GRRM23, Open OnDemand, Crystal, AlphaFold3 は利用前に申請や準備が必要です。
以下のリンク先の内容をご確認ください。

- ORCA => 公式サイトへのユーザー登録と RCCS へのメール申請が必要です。
- GRRM23 => 利用申請書の作成が必要です。
- Open OnDemand => このウェブサイトでパスワードの登録が必要です。
- Crystal => 利用申請書の作成が必要です。(ページ下部の注意書きをご確認ください)
- AlphaFold3 => ご自身でパラメータを申請、ダウンロードする必要があります。

ジョブスクリプトのヘッダ部分のサンプルが欲しい

こちらのページにいくつかサンプルを用意しました。
利用の手引きの情報、各アプリケーション用のサンプル(/apl/(アプリ名)/(バージョン)/samples 以下に配置しています)も参考になると思います。

ソフトウェア導入の要望

下記の項目を全てご記入の上、rccs-admin[at]ims.ac.jp宛(迷惑メール対策のため、@を[at]に置換しています)に送信してください。有料ソフトウェアの場合、導入できないことがあります。

- 導入を希望するソフトウェアの名前、バージョン
- ソフトウェアの概要と特長
- 共同利用システムに導入を希望する必要性

- 開発元のURL

View PDF