

job自動実行システムjobautoのアイデア

- ローカルで管理する。計算サーバーを用意する(qsubシステムを想定)。
できるだけ単純なデザインにしたい。
- qsubファイルは大量に作ることになる。たとえばQSGWのイテレーションごとに作るとすると、物質数xイテレーション数、のqsubファイルができる。qsub.{id}
- ローカルディレクトリ LOCAL xxx/ecalj_remote/lfoobar/
リモートディレクトリ REMOTE work/ecalj_remote/lfoobar/
を(必要なファイルにだけ) rsyncさせるのを原則とする。
lfoobar/はマシン名情報などは入れてmic_gw1000などとする。ディレクトリは深く掘りすぎないほうがいいように思う。

リモートへの要求

ssh,scp可能. ecالجをインストールしておきパスを通しておく
Working directoryのディレクトリ名work/, qsubのテンプレート、リソース。

ローカルとリモート初期化

1. LOCAL/foobar/ecالج_remote/lfoobar/init/に初期ディレクトリ、初期ファイルを置いておく。

- 例えば物質数だけディレクトリを作って、POSCARを配置しておく。

2. 各ディレクトリに以下の2つのファイルを置く。

- qsubスクリプト: qsub.{id}
- qsub依存性ファイル: qsub.dependency.{id}

qsub.{id}はディレクトリに複数個あっても良い。

idは0,1,2,...でいい(なんでもいい)。

qsub.{id}においては正常終了したときのみqsub.finished.{id}というファイルを作って終了するようにしておく。異常終了のときは作らない。

qsub.dependency.{id}の中身には,qsub.{id}をスタートするのに必要なファイルを1行に1ファイル名で羅列しておく。

(単純な例: qsub.dependency.{id}の中身をqsub.finished.{id-1}としておく。qsub.dependency.0は空にしておく)

初期化のスクリプトは仕事による。たとえばPOSCARを配置しておいて、
LDAとQSGWを実行してバンドプロットする仕事など。

ジョブ自動管理のスクリプト jobmon.py

以下をローカルで10秒ごとに行う(time.sleep(10))。ジョブ管理スクリプトjobmon.pyをnohupで流す
(将来的にはデーモンにしてもよいが、nohupでやれるならそれでいい)。すなわち、

```
>nohup jobmon.py --ldir=LOCAL/foobar --rdir=REMOTE/foobar --binpath=binpath --  
pythonpath=pythonpath --remote=takao@ucgw --maxqsub=4 --maxcore=16
```

として起動する。

ここでbinpathにはbinaryの入るディレクトリ名、pythonpathには用いるpythonへのpathが入るとする。

LOCAL/foobar ローカルマシンの計算パス。

LOCAL/foobar/lfoobar/以下のディレクトリで計算は行う。

REMOTE/foobar リモートマシンの計算パス。

REMOTE/foobar/lfoobar/以下のディレクトリで計算は行う。

--remote=ユーザー名@リモートマシン名

--maxqsubは最大qsub数。

--maxcoreは最大mpi数です。

1. 初期シンク

LOCAL/foobar/lfoobar/initを

LOCAL/foobar/lfoobar/date にコピーしたのち、

REMOTE/foobar/lfoobar/date にrsyncする。

dateへは投入時点での日時を代入。

qsub.nowは空のファイルとして初期化。qsub.nowには現在実行中の{id}とリモートのqsubのJOBIDがペアで一行に書かれている。qsub.nowの各行は1 353143 !qsub.id JOBIDという形。

コピーした後,qsub.{id}という形のqsubファイルについては,そのファイル内の文字列__binpath__をbinpathに,__pythonpath__をpythonpathに,__maxcore__をmaxcoreで置き換える。

--initonlyで起動すると初期シンクのみで終了するとする。

2. rsyncをローカルで起動し、リモートのlfoobar/date以下のqub.*について更新がある場合のみローカルにコピーする。

3. ローカルでqsub.dependency.*{id}を見て実行可能なものをLOCAL/foobar/date/quelistの末尾に追加する。

4. qsub.nowに書かれている{ID}に関してリモートで>qstat {ID}して比較する。消失した{ID}については対応するidに関してfinished.{id}があるかどうかを確認。 finished.{id}があるときはquelistにfinishedと日付とともに同一行に追記する、ないときはfailedと追記する。

5. qsub数が指定した最大数より小さい時,quelistを上から見ていってリモートでqsubする。quelistの同一行にstartedと追記.dateも追記。 qsubしたときにqsub.{id}とリモートマシンの{ID}のペアはファイルqsub.logに記録する。

6. ローカルでrsyncを行い（予め指定した結果ファイルのみ）リモートからローカルへ一括でrsyncする。

ジョブ操作

- モニタする：date/quelistを見る。
- qstat: date/qstatを見る。
- kill : date/killを実行.一括でKILLするコマンドを用意しておく。jobmon.pyを止めたあと、qsub.nowのIDをキルすればよい。

- あるディレクトリで計算ストップ-->qsub.dependencyを消してrsync。そのディレクトリで実行中のqsubをkillする。quelistをリネームしてバックアップ。
- 再起動：quelistをキープしておけば、そのまま--ldir=LOCAL/foobar/date/ を起動するとよい。dateディレクトリが既存の場合、初期シンクはスキップすること。