生物情報解析システムの使い方

26, Feb. 2016

ゲノムインフォマティクストレーニングコース 準備編
基礎生物学研究所 情報管理解析室
西出 浩世　hiroyo@nibb.ac.jp

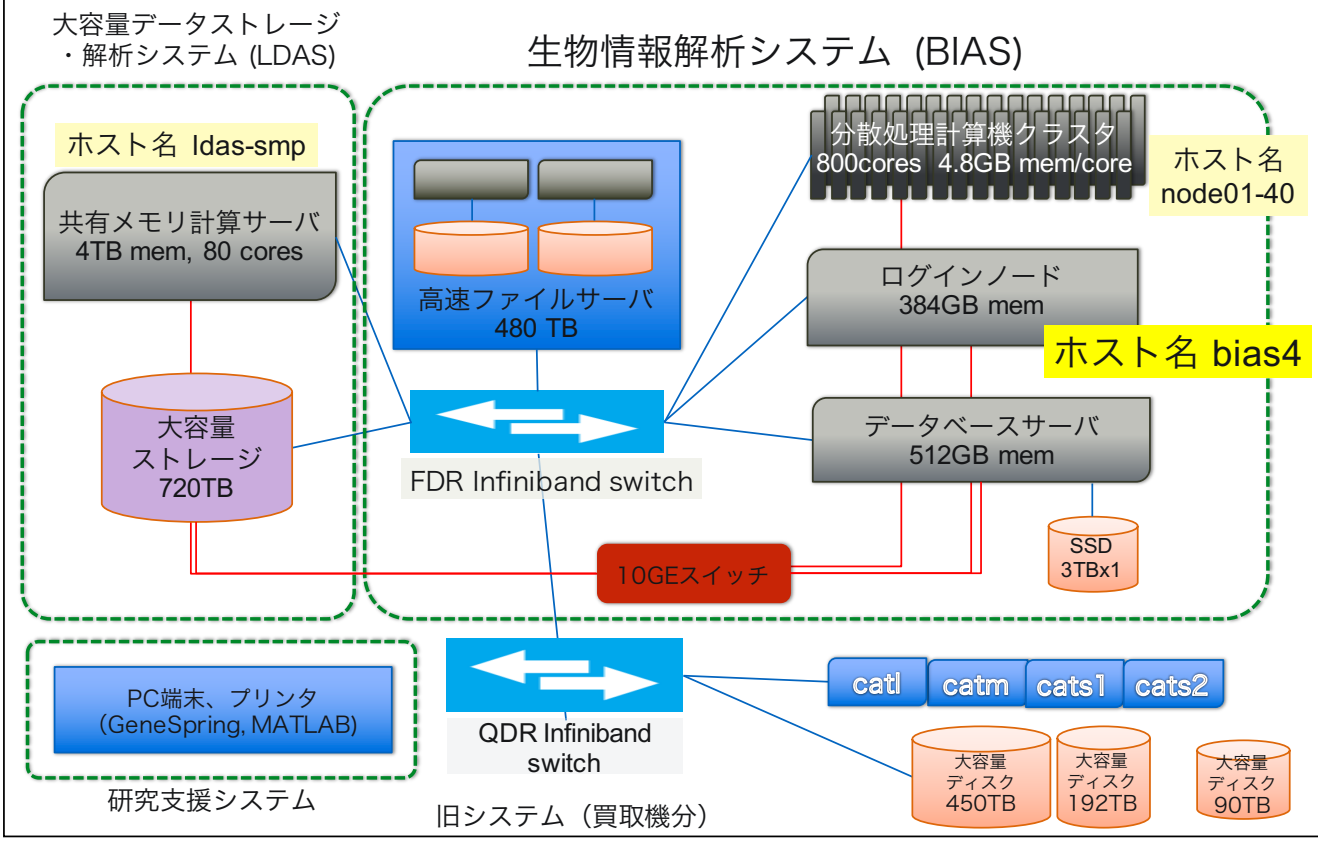
生物情報解析システムの紹介

Computer system for biological information analysis

分散処理計算機クラスタ
SGI Rackable server C2112-4RP
Intel Xeon (2.8GHz) 20core/node
96GB/node Memory,　40node, 800core

高速ファイルサーバ
DDN SFA7700
Lustre file system : 480TB

生物情報解析システムの構成



生物情報解析システム
基本的な使い方：ログイン先

bias4.nibb.ac.jp

- 基生研外の方は、ユーザーアカウントの申請が必要

  http://www.nibb.ac.jp/cproom/global/appli/index.html

- sshで接続する（telnetなどは利用できません）
- ログインノード上での作業は、プログラムの作成やファイル管理などの軽い処理にとどめ、大きな計算処理の実行はジョブ管理システム (SGE) を介して行う
- 正確なマシン名は、bias4-login.nibb.ac.jp だが、ログイン時には -login を省略可

ログイン方法：Macユーザ

- ターミナル 上で

  `$ ssh username@bias4.nibb.ac.jp`

- と入力してリターン

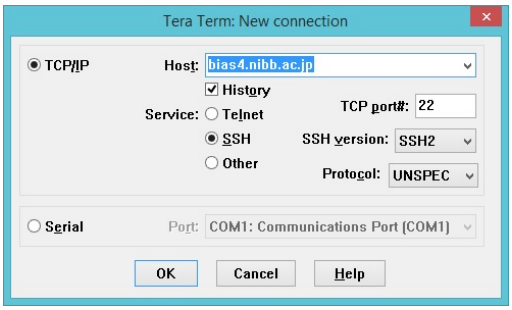
  `username@bias4.nibb.ac.jp's password:`

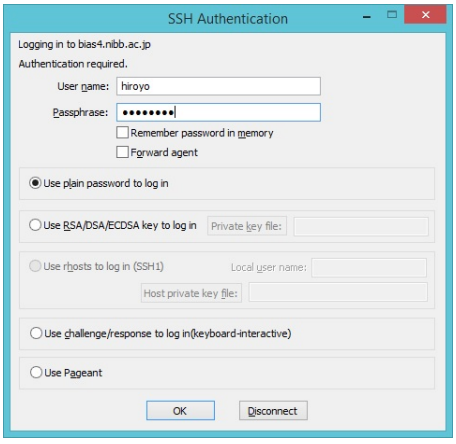
- と出たらパスワードを入力してリターン
  - 画面には何も出ません！ *******等もなし！

```
$ ssh username@bias4.nibb.ac.jp
The authenticity of host 'bias4.nibb.ac.jp (133.48.33.122)' can't be established.
RSA key fingerprint is 7b:94:9a:36:ac:60:ae:a0:14:2a:7c:0f:3c:bc:fe:24.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'bias4.nibb.ac.jp' (RSA) to the list of known hosts.
username@bias4.nibb.ac.jp's password:
Last login: Fri Jan 17 15:23:05 2014 from bigfox-01.nibb.ac.jp
[username@bias4-login ~]$
```

ログイン方法：Windowsユーザ

- TeraTermProをインストール→
  http://ttssh2.sourceforge.jp/
  - Host: bias4.nibb.ac.jp
  - Service: SSH,　TCP port#: 22,　SSH version: SSH2, Protocol: UNSPEC
  - User name: ユーザ名
  - Passphrase: パスワード

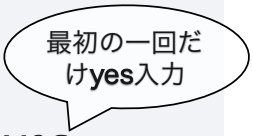




生物情報解析システム：ログインしてみましょう

```
$ ssh username@bias4.nibb.ac.jp

The authenticity of host 'bias4.nibb.ac.jp (133.48.33.122)' can't
be established.
RSA key fingerprint is
7b:94:9a:36:ac:60:ae:a0:14:2a:7c:0f:3c:bc:fe:24.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'bias4.nibb.ac.jp' (RSA) to the list of
known hosts.
username@bias4.nibb.ac.jp's password: #パスワード入力
Last login: Fri Jan 17 15:23:05 2014 from bigfox-01.nibb.ac.jp
[username@bias4-login ~]$ pwd
[username@bias4-login ~]$ ls
```



- pwd でログイン後のディレクトリを、
- ls でディレクトリの内容を確認

生物情報解析システムからログアウト

```
[username@bias4-login ~]$ exit
$
```

- exitコマンドで手元のマシンに戻る

生物情報解析システムにデータをコピー

- scp コマンドで離れたUNIXマシンにデータをコピーできる
- scp (secure copy)　暗号化して送信
- コピーが終わったらbias4にログインしてファイルを確認

```
$ cd ~/data
$ scp -r 6_sge bias4.nibb.ac.jp:
username@bias4.nibb.ac.jp's password:

$ ssh username@bias4.nibb.ac.jp
username@bias4.nibb.ac.jp's password:
[username@bias4-login ~]$ ls
```

scp コマンド

- 手元のMac（ローカル）から生物情報解析システム（リモート）へ

```
$ scp コピーしたいファイル　username@リモートホスト名:コピー先のパス
```

- リモートからローカルへ

```
$ scp username@リモートホスト名:コピーしたいファイルのパス ローカルのパス
```

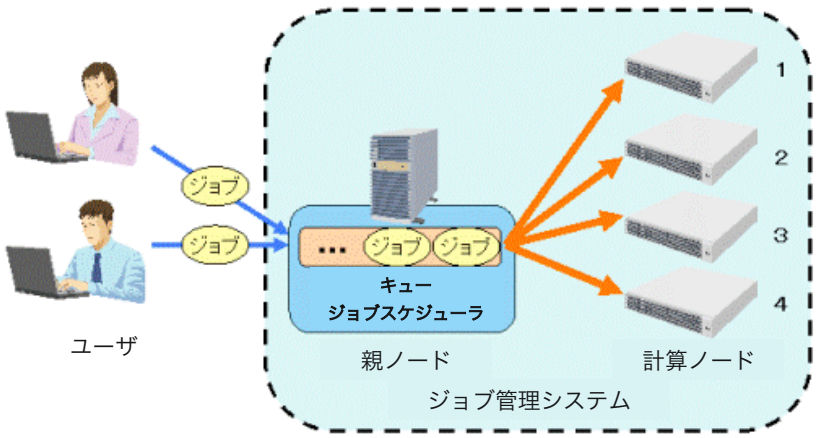
- コピー先で「:」以降を省略するとホームディレクトリになる

生物情報解析システム
SunGridEngine 利用方法

大型計算機を有効に使うには

複数の人間が同じ計算機群を使いたい...
どの計算機/CPUが空いてるか？
平等に使うには？

- ユーザのジョブを
  - 実行された順番に
  - 空いている計算機に割り振ってくれる

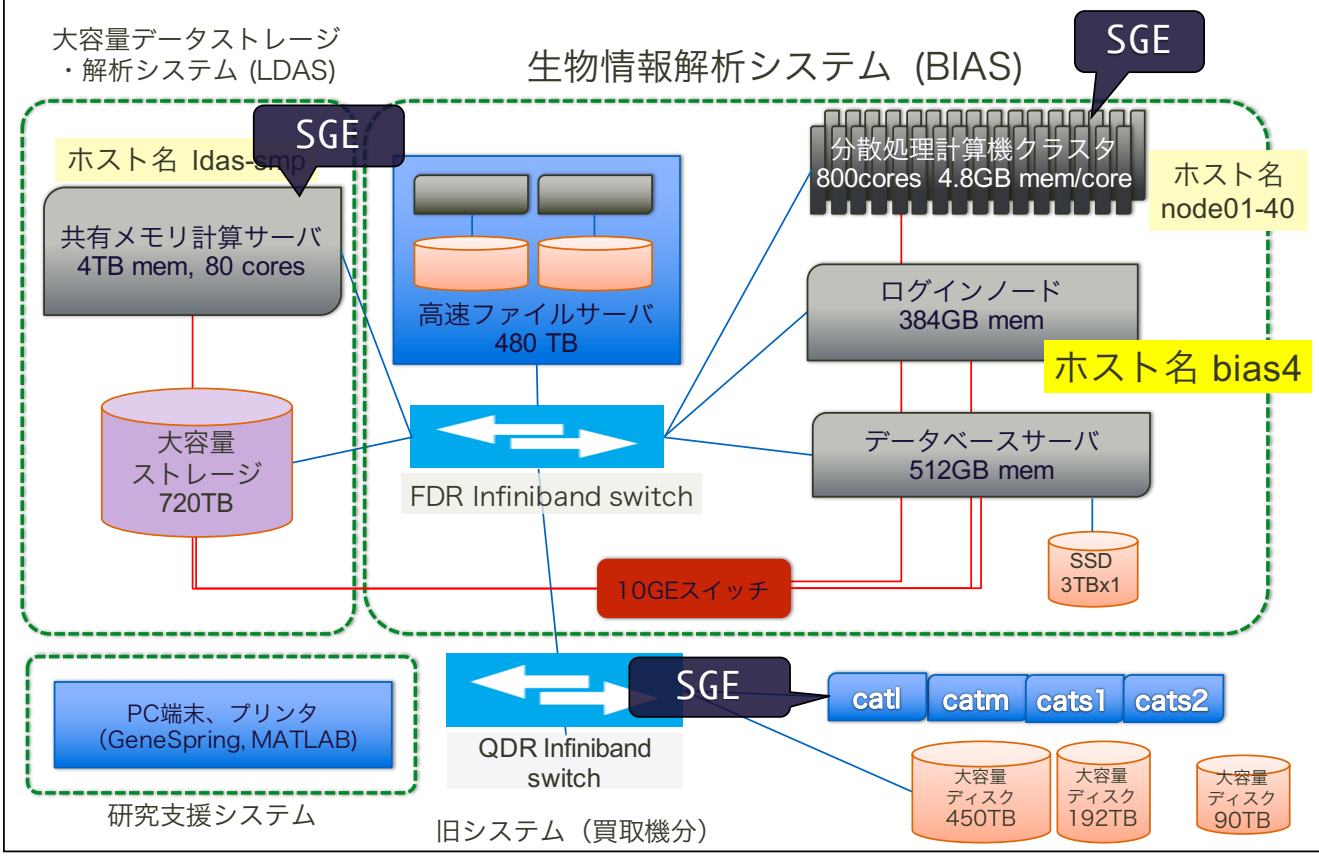


・ジョブ管理システム

ジョブ管理システム

- 親ノードが、複数ある計算機から資源の割り当てを自動で行い、効率を上げる
- ユーザは親ノードにジョブを投げるだけ（親ノードの名前すら知らなくてもよい）
- **生物情報解析システム上でのデータ解析は基本ジョブ管理システムを使うこと**
  - bias4.nibb.ac.jp はパワーがないので、皆がbias4上で解析を行うとすぐ倒れます
- **SunGridEngine (SGE)**

生物情報解析システムの構成



作業ディレクトリ

bias4.nibb.ac.jp 上での ~/6_sge

```
$ cd ~/6_sge
$ ls
```

ジョブ管理システム:SGEの利用

- 実行したいコマンドをシェルスクリプト内に記述し、**qsub**コマンド（後述）を用いてジョブ管理システムに実行させる
- シェルスクリプト ex.sh の中身を確認

```
$ less ex.sh
```

```
#!/bin/sh
#$ -cwd
#$ -q small

bowtie2 -x ecoli_genome -U ecoli.1.fastq -S ecoli.1.sam
```

qsub実行！

- シェルスクリプトをジョブ管理システム(SGE)に投入：qsub

```
$ qsub scriptfile
```

- ex.sh をqsubコマンドで実行

```
$ qsub ex.sh
Your job 814953 ("ex.sh") has been submitted
```

- 投入されたジョブはログインノードから分散処理計算機クラスタ内のノードに送られて実行される
- 標準出力と標準エラー出力のログがホームディレクトリにファイルとして作られる
- 投入したジョブの状況を見るには qstat コマンドを使う

```
$ qstat
```

qsub (bowtie2) 結果の確認

```
$ ls                                    結果ファイルの確認
$ ls ex.sh.*                            標準出力・標準エラー出力ファイル
ex.sh.e814953 ex.sh.o814953

$ less ex.sh.e814953                    標準エラー出力ファイルの中身を確認
330118 reads; of these:
  330118 (100.00%) were unpaired; of these:
    3364 (1.02%) aligned 0 times
    229054 (69.39%) aligned exactly 1 time
    97700 (29.60%) aligned >1 times
98.98% overall alignment rate
```

SGEの主なコマンド

qsub *script_file*	*script_file* ジョブを投入
qstat	自分のジョブの状態を表示
qstat -u '*'	全ユーザのジョブ状態を表示
qdel *job-ID*	*job-ID* のジョブを削除

```
$ qstat
job-ID  prior   name       user         state submit/start at     queue                          slots ja-task-ID
-----------------------------------------------------------------------------------------------------------------
814953 0.00000 ex.sh      hiroyo       r     01/08/2015 14:14:54 small@node04                   1
814954 0.00000 job.sh     hiroyo       qw    01/08/2015 14:14:54 small@node05                   1
814955 0.00000 job.sh     hiroyo       qw    01/08/2015 14:14:54 small@node05                   1

$ qdel 814953
hiroyo has deleted job 814953
```

qsub のオプション

- qsub には様々なオプションがあり、シェルスクリプト内で「#$」に続けて書いておくことで機能を加えることができる

```
$ less ex.sh
```

```
#!/bin/sh
#$ -q small          smallキューを指定
#$ -cwd              qsubしたディレクトリに移動してジョブを実行

bowtie2 -x ecoli_genome –U ecoli.1.fastq -S ecoli.1.sam
```

- -cwd を指定しているので、ファイルのパスを付けていない

qsub のオプション

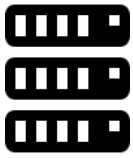
オプション	説明
#$ -o *filename*	標準出力の結果を指定したファイルに保存
#$ -e *filename*	標準エラー出力の結果を指定したファイルに保存
#$ -q *queue_name*	キューを指定してジョブを実行
#$ -cwd	qsubした時のディレクトリに移動してジョブを実行
#$ -v 環境変数=値	環境変数をジョブに渡す
#$ -N *job_name*	ジョブ名を指定する
#$ -s *shell_name*	ジョブスクリプトを指定したシェルで実行
#$ -a *MMDDhhmm*	ジョブの開始日時を指定
#$ -l *resource_name* 値	ジョブが使うリソース量を指定する
#$ -pe *PE_name* プロセス数	並列ジョブを実行する場合の環境と並列数の指定
#$ -t 開始番号-終了番号	アレイジョブを実行

キュー（ジョブの待ち行列）構成

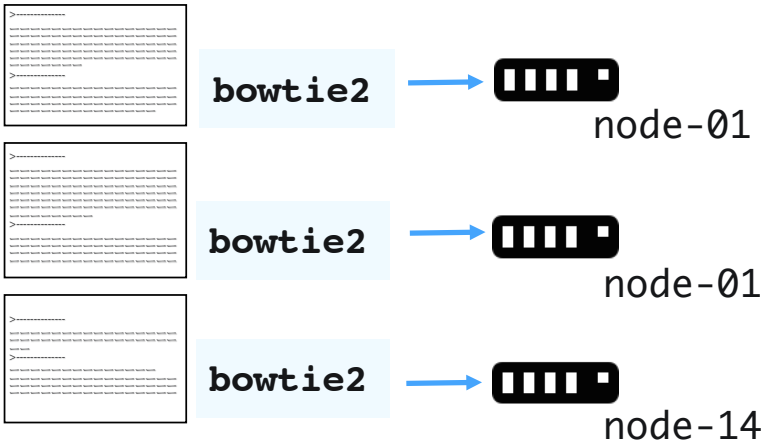
	分散並列処理型			共有メモリ型			
	分散処理計算機クラスタ			共有メモリ型計算サーバ			cat群
キュー名	small	medium	large	smps	smpm	smpl	cat
ジョブの特徴	短時間・並列多	中規模	長時間	中メモリ	大メモリ	最大メモリ	denovoアセンブリ用
利用ノード	node01-40	node01-40	node01-40	ldas-smp	ldas-smp	ldas-smp	catl, catm cat1, cat2
最大実行時間/job	6時間	72時間	no limit	no limit	no limit	no limit	no limit
最大ジョブ数/キュー	580	200	20	8	4	1	112
最大使用メモリ/ジョブ	4GB	4GB	4GB	500GB	1TB	4TB	512GB/1TB/96GB/96GB
利用できるPE	smp, mpi128, mpi256, make	smp, mpi128, mpi256, make	smp, mpi128, make	smp	smp	smp	smp, make

・キューを指定しない場合、デフォルトでは「small」で実行される
・ユーザあたりジョブ同時実行数は最大 400

for文を使って並列化



node-01~node40
分散処理計算機クラスタ



- 1台で順次実行していては時間がかかる計算も、分割して複数台、複数CPUに仕事をさせれば数倍の速度で終わる

for文を使って並列化：実際の例

```
$ less ex2.sh
```

```
#!/bin/sh
#$ -cwd

num=${1}
bowtie2 -x ecoli_genome -U ecoli.${num}.fastq
        -S ecoli.${num}.sam
```

- qsub に渡すシェルスクリプト

for文を使って並列化：例2

```
$ less ex3.sh
```

```
#!/bin/sh
for i in 1 2 3
do
 qsub ex2.sh ${i}
done
```

- qsubを実行するためのシェルスクリプト
- for文を使って以下のコマンドを順番にqsubしている

  ex2.sh 1

  ex2.sh 2

  ex2.sh 3

- 実行

```
$ ./ex3.sh
```

for文を使って並列化：例3

- 投入したジョブを確認
- 3つのbowtie2が異なるマシンで実行されている

```
$ qstat
```

```
7769444 0.50500 ex3.sh    hiroyo    r    02/16/2016 10:19:38 small@node26    1
7769445 0.50500 ex3.sh    hiroyo    r    02/16/2016 10:19:38 small@node11    1
7769446 0.50500 ex3.sh    hiroyo    r    02/16/2016 10:19:38 small@node19    1
```