2022年度
バイオユーザのための
京大化研スパコンシステム利用法

日本ヒューレット・パッカード合同会社
上原英也

spradm@scl.kyoto-u.ac.jp

0774-38-3265　(内線: 3265)
https://www.scl.kyoto-u.ac.jp/

共催
京都大学化学研究所附属バイオインフォマティクスセンター
NPOバイオインフォマティクス・ジャパン

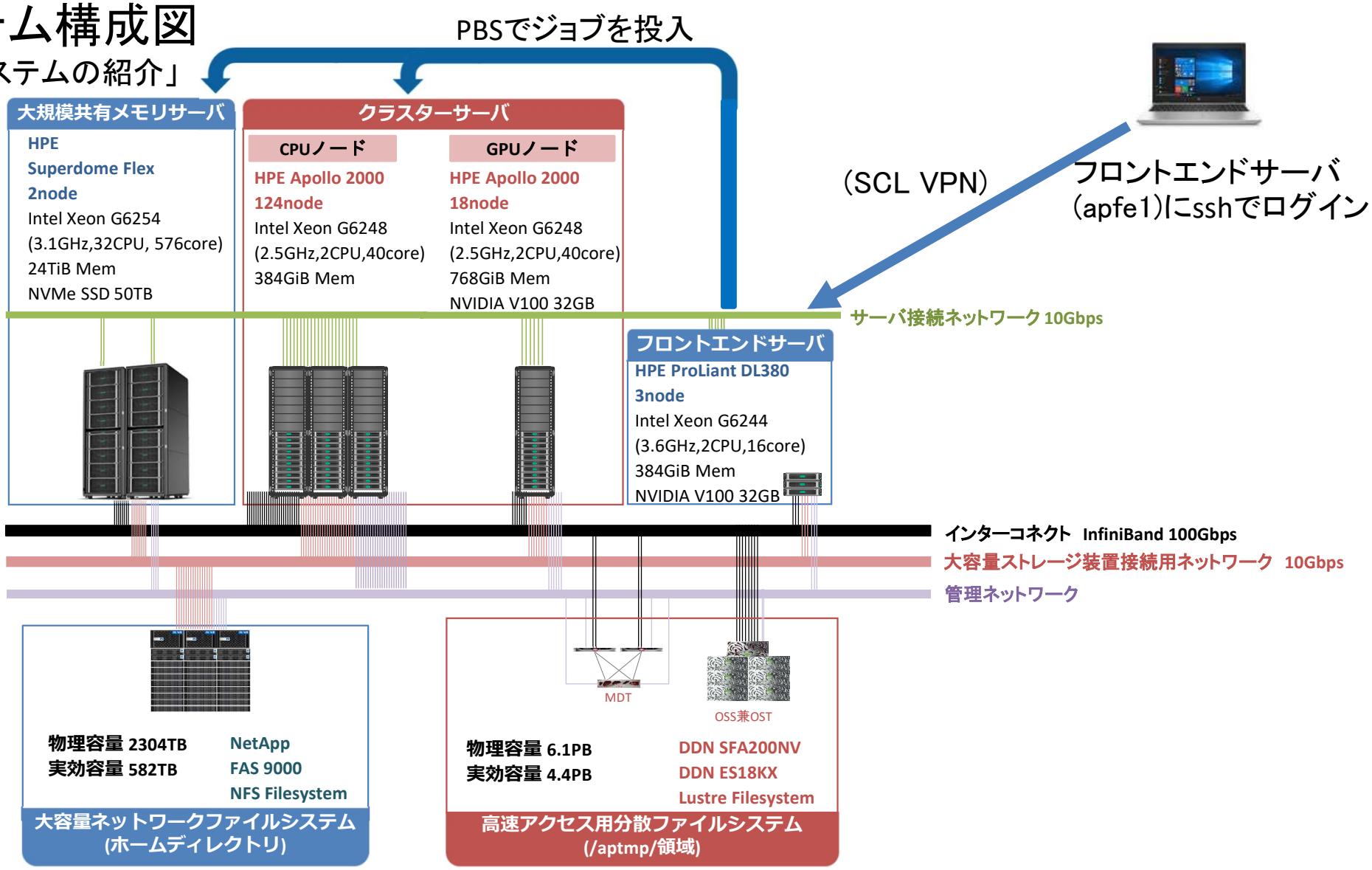
ver 1.0.1

内容

1 システム概要

1.1 システム構成図

（参）「システムの紹介」



1.2 サーバスペック

（参）「システムの紹介」→「大規模共有メモリシステム」
「システムの紹介」→「大規模計算クラスタ」

	大規模共有メモリシステム	大規模計算クラスタ (CPUノード)	大規模計算クラスタ (GPUノード)	フロントエンドサーバ
機種	HPE Superdome Flex	HPE Apollo2000	HPE Apollo2000	HPE ProLiant DL380
ノード数	2	124	18	3
ホスト名	sdf1, sdf2	ap001～ap124	apg01～apg18	apfe1
CPU	Intel Xeon G6254 3.1GHz x 32CPU (18コア/CPU)	Intel Xeon G6248 2.5GHz x 2CPU (20コア/CPU)		Intel Xeon G6244 3.6GHz x 2CPU (8コア/CPU)
コア数/ノード	576	40	40	16
メモリ/ノード	24TiB (42GiB/コア)	384GiB (9GiB/コア)	768GiB (18GiB/コア)	384GiB
GPU	-	-	NVIDIA V100 (5,120コア、32GB)	NVIDIA V100 (5,120コア、32GB)
SSD	50TB	-	-	-
OS	Red Hat Enterprise Linux(＊) 7.6			

(＊) CentOSの有償版。x86_64 (amd64) Linux用 binaryプログラムが動作します。

2 利用にあたって

2.1 新規利用申請 (アカウントの取得)

(参)「各種手続き」→「新規利用申請」

- 支払責任者が京都大学教職員であるか否かによって申請手順が異なります。
- スパコンシステム(計算サーバ)で計算される場合
  - 支払い責任者が京都大学教職員:
    - 「新規利用申請書を作成する」→「利用形態」で「計算サーバ使用」を選択
  - 支払い責任者が京都大学教職員以外
    - 「新規利用申請用紙(PDF)」の「計算サーバの利用」の「有」にチェック
- 【新規利用申請書作成時の注意点】をお読み下さい。

- 研究成果報告書(計算サーバの利用者)
  - 以下に該当する方はご提出は不要です
    - 所属が企業(営利目的でのご利用)
    - 新規利用申請が1月以降
    - 計算サーバもしくはアプリケーションを全く利用しなかった(※)
    - 特許出願中もしくは出願準備中などにより、研究内容を非公開としたい(※)

    (※) この場合には、その旨をスパコンシステムまでお知らせください。

2.2 利用負担金

（参）「各種手続き」→「利用負担金」

課金対象	計算ルール
基本料金	1,000円/月
計算サーバ(CPU時間)	0.00125円/秒 （化研所内利用者は 30,000円/月、所外利用者は 40,000円/月 の上限額あり）
ディスク基本料金 (ホーム領域)	無料 (20GBまで利用可能)
クラウドストレージサービス	無料 (20GBまで利用可能)
オプションサービス	
ディスク拡張サービス(ホーム領域)	+100GB 1,000円/年度 (最大300GBを追加可能)
クラウドストレージ拡張サービス	+100GB 1,000円/年度 (最大300GBを追加可能)

- 計算一時領域 (/aptmp/(ユーザ名)/) は課金されません(容量制限もなし)。
- 「各種手続き」→「利用負担金の参照」で利用金額等をご確認頂けます。

2.3 スパコンシステムログイン手順

(参)「使い方と注意事項」→「パソコンからの使い方」
「使い方と注意事項」→「計算サービス」→「ホスト名とログイン」

- スパコンシステムを利用するには フロントエンドサーバ apfe1 にログインして下さい。
  - (Linux, Mac) ターミナルで

```
$ ssh アカウント名@apfe1.scl.kyoto-u.ac.jp
```

(× login.scl.kyoto-u.ac.jp)

(Windows) TeraTerm, Putty, RLogin, 等ターミナルソフトをインストール

  - apfe1の GUIアプリケーション(LibreOffice, gnuplot, FigTree, IGV 等)を利用する場合
    - (Linux) ssh -Y(C) オプションを指定します。
    - (Mac) Xquartz をインストールして、ssh –Y(C)でログイン。
    - (Windows) Xming をインストール
      (参)「使い方と注意事項」→「パソコンからの使い方」→「WindowsでX11フォワーディング」
- ログインシェル(コマンドラインインターフェース)の変更
  - csh, tcsh (初期設定), bash, zsh を選択できます。

```
$ ldapchsh bash          # bashに変更する場合
```

2.3 スパコンシステムログイン手順 (続き)

（参）「使い方と注意事項」→「基本サービス」→「ホスト名とログイン」

- 京大ネットワーク（KUINS）以外の外部ネットワークからフロントエンドサーバにログインする場合は、まず最初にVPN でスパコンシステムのネットワークに接続して下さい。
  - (Windows, Linux, Mac) https://vpn.scl.kyoto-u.ac.jp/
  - (Mac) Mac App Store から F5Access をインストール

- 自PC⇔フロントエンドサーバ間でファイル転送するには以下のソフト・コマンドをご利用下さい。
  (Windows) WinSCP, FileZilla （参）「使い方と注意事項」→「パソコンからの使い方」→「Windowsでファイル転送」
  (Mac) Cyberduck, FileZilla （参）「使い方と注意事項」→「パソコンからの使い方」→「MacOSXでファイル転送」
  (Linux) scp, sftp, rsync
  - 改行コードに注意！
    Windowsで作成したテキストファイルをスパコンシステムにコピーして実行しようとすると、
    改行コードの問題でエラーになることがあります。
    （例）
    '¥r': コマンドが見つかりません
    ^M: コマンドが見つかりません
  - このような場合、dos2unix コマンドでファイルの改行コードをLinux用に変換して下さい。
    $ dos2unix (ファイル)

2.3 スパコンシステムログイン手順 (続き)

（参）「使い方と注意事項」→「基本サービス」→「ホスト名とログイン」

- フロントエンドサーバ（apfe1）では小規模なテスト計算以外の計算を実行しないで下さい
  （ジョブあたり30分・8GBのCPU時間・メモリサイズの制限あり）。

- 計算サーバで対話的に作業するにはPBSのインタラクティブバッチジョブ（§ 4.7)を利用して下さい
  - 原則として、計算サーバ（スパコン本体）には直接sshでログインしないで下さい
    （ジョブが正しく実行されているかtop 等で確認するくらいであれば可）。

2.4 ディスク領域

（参）「使い方と注意事項」→「計算サービス」→「ディスク領域」

	ホーム領域	計算一時領域	SSD
場所	~（ユーザ名）/	/aptmp/（ユーザ名）/	${TMPDIR}/
ファイルシステム	NFS	Lustre	xfs
課金	有	無	-
容量制限	有	無	-
Snapshot	有	無	-
サーバ間共有	○	○	✖
備考			SDFキューでのみ利用可

2.4.1 ホーム領域 (ホームディレクトリ)

（参）「使い方と注意事項」→「計算サービス」→「ディスク領域」

- Snapshot
  - ホームディレクトリ内の各ディレクトリに .snapshot/(日時)/ という隠しディレクトリがあり、その日時でのファイルを参照できます。

```
$ ls .snapshot/
daily.2022-05-09_0010/  weekly.2022-05-01_0015/  weekly.2022-05-08_0015
```

  - 誤って削除・修正したファイルを .snapshot/ から復元できます(普通に cp でコピー)。
  - .snapshot/ は ls -a で見えません！（タブ補完も効きません）
  - .snapshot/ の中のファイルは削除・変更できません。

- ディスク使用量の確認 (quota)

```
$ quota -s
Disk quotas for user lect-2 (uid 10152):
     Filesystem  space  quota  limit  grace  files  quota  limit  grace
fas9000-02_NFS:/HOME/user2/
                948M   18432M  20480M          373    1800k  2000k
```

現在のディスク使用量

ディスク使用量制限値

現在の総ファイル数

ファイル数制限値

2.4.2 計算一時領域 (/aptmp/(ユーザ名)/)

（参）「使い方と注意事項」→「計算サービス」→「ディスク領域」

- 基本的には計算時はこの領域を使用して下さい。
- 課金されません。
- 容量制限・保存期限はありません。
  - 空き容量が少なくなってきた場合はファイルの整理(圧縮・削除)をお願いする可能性があります。
  - ファイルサイズだけでなく、ファイル・ディレクトリ数にもご注意下さい！
    - ディスク使用量

```
$ du -ks (ディレクトリ)/          # (ディレクトリ)/の全サイズ [kb]
$ du -kS (ディレクトリ)/          # (ディレクトリ)/の中の各ディレクトリ直下の合計ファイルサイズ [kb]
$ du -hS (ディレクトリ)/ | sort -h    # ディレクトリのサイズでソート
```

    - ファイル数

```
$ find (ディレクトリ)/ | wc -l                    # (ディレクトリ)/の全ファイル・ディレクトリ数
$ du --inodes -S -t 10000 (ディレクトリ)/         # 直下にファイルが10,000個以上あるディレクトリだけ
                                                    出力
```

2.5 ウェブアプリケーション

（参）「使い方と注意事項」→「基本サービス」→「メールシステムの利用」
「使い方と注意事項」→「基本サービス」→「クラウドストレージ」

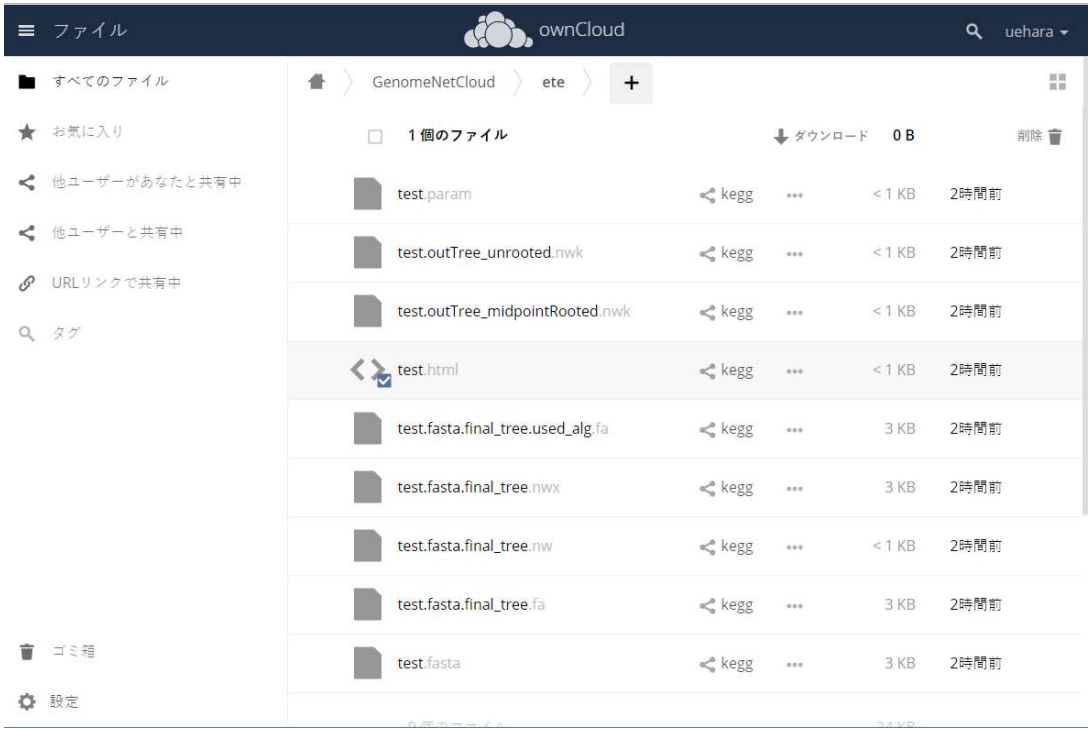
（注）ウェブメール、クラウドストレージページ内の
「ログアウト」ボタンは機能しません。

2.6 ownCloud計算サービス

（参）「アプリケーション一覧」→「ownCloud計算サービス」
「アプリケーション一覧」→「ownCloud計算サービス」 →「ETE」

- ownCloudを使用して計算を非同期的に実行します。
  - ownCloudフォルダ内の「GenomeNetCloud」→「ツール名」に入力ファイルを置くと計算が実行され、計算結果ファイルが作成されます。
  - ownCloud 計算サービスを利用をするには事前に準備が必要ですので、利用をご希望の際はスパコンシステムにお知らせ下さい。

- 現在対応している計算サービスは以下の通りです。
  - ETE (系統樹作成)



2.7 システム稼働状況

- スパコンシステムホームページのトップページ

← ノード毎の詳細情報
（要スパコンアカウント）

- リアルタイムでサーバの使用状況を確認するにはqstatmyjobs コマンド（§4.9）をご利用下さい

```
lect-1@apfe1 ~> qstatmyjobs
User: lect-1
                    jobs                ncpus                          mem(gb)              walltime(h)
Queue  vacant(use%) sum/sum_max  avail max  sum/sum_max    avail   max  sum/sum_max  default max
-----  ----------- ------------ ----------------------- ------------------------- ----------------
QUICK   1443( 78%)  0/2           4   4   0/unlimited      72    72   0/unlimited      1 01
SMALL    965( 82%)  0/unlimited  12  12   0/36            108   108   0/324            6 12
APC      857( 84%)  0/unlimited  35  40   0/unlimited      60   720   0/unlimited   2880 unlimited
APG       68( 89%)  0/4          16  40   0/unlimited       8   720   0/unlimited   2880 unlimited
SDF      398( 59%)  0/8           - 144   0/288            -  6144   0/8192        2880 unlimited
=======================================================================================================
TOTAL               0/unlimited             0/500                    0/12288
```

3 アプリケーションの利用

3.1 moduleコマンド

（参）「アプリケーション一覧」→「module コマンド」

- スパコンシステムのアプリケーションは module コマンドで管理されています。

```
$ module avail                          # 利用できるアプリケーションの module を表示。
$ module avail -L                       # 各アプリケーションの最新版だけを表示。
$ module avail bl                       # 名前が bl から始まる module だけを表示(注：case sensitive) 。

$ module load [-s] (module名)           # アプリケーションの module を読み込む(そのアプリケーションが
                                          依存する別のモジュールも併せてload されることがあります。
                                          -sオプションを付けると読み込み時のメッセージが表示されません。

$ module list                           # 現在読み込んでいる module の確認。

$ module switch [-f] (module名1) (module名2)   # 読み込んでいる module の切り替え(同一アプリの
                                                 場合はmodule名１は省略可)。エラーが出てswitch
                                                 できない場合は -f オプションを付けて下さい。

$ module unload (module名)              # 指定した module を破棄。
$ module purge                          # 全て破棄。
```

3.1 moduleコマンド (続き)

- シェルスクリプト（PBSジョブスクリプト）の中で module コマンドを使用する際は
  source /etc/profile.d/modules.sh　　（sh/bashスクリプト）
  source /etc/profile.d/modules.csh　　（csh/tcshスクリプト）
  を含めて下さい。
- 同時に読み込むモジュールの組み合わせによっては、アプリケーションが動作不良を起こすことがありますので、アプリケーションの実行後にその都度 module unload (or purge)するようにして下さい。

```
#!/bin/sh
source /etc/profile.d/modules.sh

module load prog1
prog1 xxxx
module purge

module load prog2
prog2 yyyy
module purge
```

3.2 バイオインフォマティクスアプリケーション

（参）「アプリケーション一覧」→「バイオインフォマティクス」

- インストールされているバイオインフォマティクスアプリケーション
  https://www.scl.kyoto-u.ac.jp/Appli/#biotool
- ゲノムネットサービスで利用しているバイオインフォマティクスアプリケーション

名称	概要	モジュール名
BLAST+	ホモロジー検索	blast+
Clustal Omega	マルチプルアラインメント	clustal-omega
ClustalW2	マルチプルアラインメント	clustalw2
DBGET	統合データベース検索システム	dbget
ETE Toolkit	系統樹作成	ete
FASTA	ホモロジー検索	fasta
FastTree	系統樹作成	FastTree
FastTreeMP	系統樹作成(FastTreeのOpenMP並列化版)	FastTreeMP
ghostx	ホモロジー検索	ghostx
ghostz	ホモロジー検索	ghostz
HMMER	モチーフ検索	hmmer
MAFFT	マルチプルアラインメント	mafft
MUSCLE	マルチプルアラインメント	muscle
PHYLIP	系統樹作成	phylip
PRRN	マルチプルアラインメント	prrn
raxml	系統樹作成	raxml
SIMCOMP	化合物構造検索	simcomp
ssearch	ホモロジー検索	ssearch
SUBCOMP	化合物部分構造検索	subcomp

3.3 バイオインフォマティクスデータベース

（参）「アプリケーション一覧」→「バイオインフォマティクスDB」

- バイオインフォマティクスデータベースは
  /db/(種別)/(データベース名)/
  に格納されています。
  - 種別: blast, bowtie, bowtie2, dbget, diamond, fasta, ghostx, hmmer, motif, rpsblast
  - データベース名: genbank, refseq, mgenes, ncbi, swissprot, trembl, pfam, 等

  (例) NCBI nr のBLASTファイル: /db/blast/ncbi/nr.*
- ~~毎週火曜に更新されます~~。
  - データベース更新情報: /db/dbinfo.txt
- /db/ の実体は /lustre/db/YYYYMMDD/ にあります。
  - 一ヶ月以上古いものは、毎月最初のもの以外は削除されます。
  - 毎月最初のものは .tpxz 形式で圧縮して保持しています。
    - ファイルの抽出手順は
      /lustre/db/00_How_to_extract_files.txt
      /lustre/db/extract.sh
      を参照下さい。

3.3 バイオインフォマティクスデータベース(続き)

- 利用可能なバイオインフォマティクスデータベース（抜粋）

データベース名	概要	ディレクトリ名
GenfBank	塩基配列データベース	genbank
GenBank-upd	塩基配列データベース	genbank-upd
GenPept	塩基配列データベース	genpept
GenPept-upd	塩基配列データベース	genpept-upd
RefSeq	塩基配列・アミノ酸配列データベース	refseq
MGENES	メタゲノム情報	mgenes
NR-NT	重複を取り除いた塩基配列データベース	nr-nt
NR-AA	重複を取り除いたアミノ酸配列データベース	nr-aa
NCBI BLASTデータベース	NCBIで公開されているBLASTデータベース (nr, nt, swissprot, refseq_protein, taxdb等)	ncbi
UniProt/Swiss-Prot	アミノ酸配列データベース	swissprot
UniProt/TrEMBL	アミノ酸配列データベース	trembl
UniRef	アミノ酸配列データベース	uniref
dbEST	EST(Expressed Sequence Tags)配列	dbest
dbGSS	GSS(Genome Survey Sequences)配列	dbgss
dbSTS	STS(Sequence Taged Sites)配列	dbsts
Silva (*)	リボソームRNA配列データベース	silva
RDP	リボソームRNA配列データベース	rdp
PR2	リボソームRNA配列データベース	pr2
PDBSTR	PDBのアミノ酸配列	pdbstr
Pfam	タンパク質ドメインファミリー	pfam
NCBI CDD	NCBI Conserved Domain Database	ncbi-cdd

(*) academic use only

4. バッチシステム PBS

4.1 バッチシステム (ジョブスケジューラー)とは

（参）「使い方と注意事項」→「計算サービス」→「ジョブ投入システム」

- 共有の計算機システムにおいて、複数のユーザができるだけ公平に計算機リソースを利用できる様にジョブのスケジューリングを行います。

- 当スパコンシステムではバッチシステムとして Altair社の PBS Professional を使用しています。

- 実行したいジョブ（計算処理）をファイル（ジョブスクリプトファイル）に書いて、それをフロントエンドサーバ apfe1 から qsub コマンドで投入します。

- ユーザの使用可能コア数・メモリ

同時実行ジョブの合計コア数 (ソフトリミット＊)	500 (300)
同時実行ジョブの合計メモリ (ソフトリミット＊)	12TB (3TB)

＊ ソフトリミットを越えたコア数分のジョブはシステム混雑時(キュー待ちが発生時)にスケジューリングの優先順位が低くなります

4.2 ジョブの投入

（参）「使い方と注意事項」→「計算サービス」→「ジョブ投入システム」→「ジョブ投入オプション」

$ qsub　[オプション]　（ジョブスクリプトファイル）

- 主なオプション
  - -q xxxx　　キューの指定 (QUICK, SMALL, APC, APG, SDF)
    ＊キューによって使用できるマシン、リソース、ジョブの優先度が異なります
  - -l xxxx　　ジョブの実行に必要なリソースを指定
    - 主なリソース
      - » select=1:ncpus=(使用コア数):mem=(使用メモリ)
        (ncpus, mem を指定しなかった場合はキューのデフォルト値になります)
      - » GPUを使う場合は ngpus=1 も追加して下さい。
      - » ジョブの最大経過時間を指定する場合は -l walltime=(時間):(分):(秒) を追加して下さい
  - -N xxxx　　ジョブ名の指定
  - -o xxxx　　標準出力ファイルのパス
  - -e xxxx　　標準エラー出力ファイルのパス
  - -j oe　　標準出力と標準エラー出力をまとめて出力
- qsubコマンドでジョブを投入すると Job ID が発行されます
  - ジョブがうまく実行できない等の問い合わせの際は Job ID をご連絡下さい

4.3 キュー一覧

（参）「使い方と注意事項」→「計算サービス」→「ジョブ投入システム」

キュー名	QUICK	SMALL	APC	APG	SDF
計算サーバ:台数	クラスタ(CPU): 124 クラスタ(GPU):18 大規模共有メモリ:2	クラスタ(CPU): 124 クラスタ(GPU):16	クラスタ(CPU):121 クラスタ(GPU):16	クラスタ(GPU):16	大規模共有メモリ:2
ジョブの特徴	テスト	小規模	中～大規模	GPU	大規模メモリ
キューの実行順位	100	70	50	50	90
ジョブあたりの制限					
最大コア数 (デフォルト)	4 (1)	12 (1)	40 (※1) (1)	40 (1)	144 (18)
最大メモリ (デフォルト)	72 GB (9 GB)	108 GB (9 GB)	720 GB (※1,2) (9 GB)	720 GB (18 GB)	6 TB (768 GB)
最大経過時間 (デフォルト)	1 h	12 h (6 h)	制限なし (2880 h)	制限なし (2880 h)	制限なし (2880 h)
ユーザあたりの制限					
最大ジョブ数 (ソフトリミット)	2 (1)	-	-	4 (1)	8 (4)
合計コア数 (ソフトリミット)	-	36	500	-	288 (144)
合計メモリ(ソフトリミット)	-	324 GB	4.5 TB	-	8 TB (6 TB)

(※1) MPIで複数ノードを使用したジョブの場合、1ジョブで500コア・4.5TBまで使用可

(※2) APCキューの全137ノードのうち、360GBより大きいメモリを使用できるのは16ノード(GPU搭載ノード)だけとなります

4.3 キューー一覧(続き)

キュー名	cdb
計算サーバ:台数	dl560: 1
ジョブの特徴	GenomeNet開発
キューの実行順位	50
ジョブあたりの制限	
最大コア数 (デフォルト)	32 (1)
最大メモリ (デフォルト)	1.3TB (20 GB)
最大経過時間 (デフォルト)	制限なし (720 h)
ユーザあたりの制限	
最大ジョブ数	32
合計コア数	32
合計メモリ	1.3TB

- dl560のホームディレクトリはapfe1のホームディレクトリとは異なります(/aptmp/ は共通)。
- PBS(cdbキュー)を使用しないでdl560で直接実行されたジョブは30分のCPU時間制限があります。
- dl560の64コア・1.5TBメモリのうち、cdbキューは合計58コア・1.3TBまで使用できます。
- 1時間以上計算時間がかかる大規模ジョブを流す必要がある時は、必ず、他の利用者に案内し了解をとって下さい。
- jupiterなどを走らせるときはインタラクティブバッチジョブ(“qsub -I -q cdb …”)を使用して下さい。コアを占有しないために、夜間など利用しない時間帯はジョブを終了して下さい(or -l walltime=… を指定しておく)。

4.4 ジョブスクリプトの例

(参)「使い方と注意事項」→「計算サービス」→「ジョブスクリプトの例」

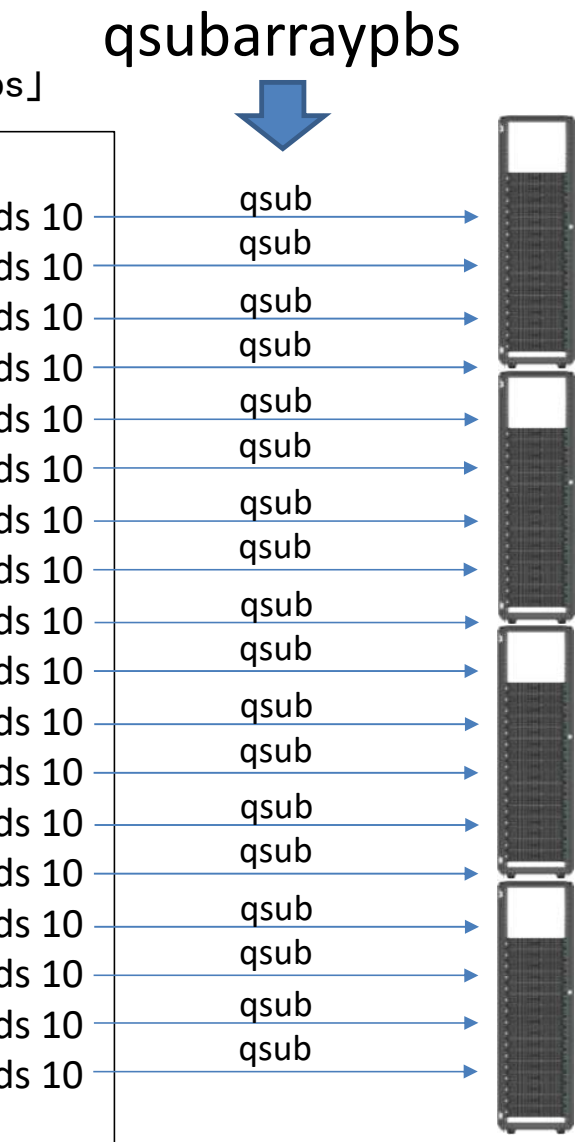
```
#!/bin/sh
#PBS -q APC                                  ←キューを指定
#PBS -l select=1:ncpus=10:mem=40gb           ←使用コア数、メモリサイズを指定
#PBS -N test                                 ←ジョブ名を指定
#PBS -o test.out                             ←標準出力ファイル
#PBS -e test.err                             ←標準エラー出力ファイル

source /etc/profile.d/modules.sh             ← moduleコマンドを使えるようにする
module load blast+/2.14.0                    ← blast+ を読み込み
cd $PBS_O_WORKDIR/                           ← qsubを実行したディレクトリに移動

blastp -db db/nr -query query.fa -out result.out -outfmt 7 -num_threads 10
```

- qsubコマンドのオプションはジョブスクリプト中で
  #PBS xxxx
  で指定することができます。
- 並列化されているプログラムを実行する時は、1コアしか使用しない場合でも
  プログラムのオプションで使用コア数を明示的に指定して下さい。

4.5 大規模入力ファイルの処理

（参）「アプリケーション一覧」→「バイオインフォマティクス」→「qsubarraypbs」

```
$ cat blastp.com
blastp -db db/nr -query query01.fa -out result01.out -outfmt 7 -num_threads 10
blastp -db db/nr -query query02.fa -out result02.out -outfmt 7 -num_threads 10
blastp -db db/nr -query query03.fa -out result03.out -outfmt 7 -num_threads 10
blastp -db db/nr -query query04.fa -out result04.out -outfmt 7 -num_threads 10
blastp -db db/nr -query query05.fa -out result05.out -outfmt 7 -num_threads 10
blastp -db db/nr -query query06.fa -out result06.out -outfmt 7 -num_threads 10
blastp -db db/nr -query query07.fa -out result07.out -outfmt 7 -num_threads 10
blastp -db db/nr -query query08.fa -out result08.out -outfmt 7 -num_threads 10
blastp -db db/nr -query query09.fa -out result09.out -outfmt 7 -num_threads 10
blastp -db db/nr -query query10.fa -out result10.out -outfmt 7 -num_threads 10
blastp -db db/nr -query query11.fa -out result11.out -outfmt 7 -num_threads 10
blastp -db db/nr -query query12.fa -out result12.out -outfmt 7 -num_threads 10
blastp -db db/nr -query query13.fa -out result13.out -outfmt 7 -num_threads 10
blastp -db db/nr -query query14.fa -out result14.out -outfmt 7 -num_threads 10
blastp -db db/nr -query query15.fa -out result15.out -outfmt 7 -num_threads 10
blastp -db db/nr -query query16.fa -out result16.out -outfmt 7 -num_threads 10
blastp -db db/nr -query query17.fa -out result17.out -outfmt 7 -num_threads 10
blastp -db db/nr -query query18.fa -out result18.out -outfmt 7 -num_threads 10
…
```



4.5 大規模入力ファイルの処理(続き)

（＊）PBSのコマンドではなく、当スパコンシステム独自のコマンドです
（参）「アプリケーション一覧」→「バイオインフォマティクス」→「qsubarraypbs 」

$ qsubarraypbs(*) [オプション] （ジョブを列挙したファイル）

- PBSのアレイジョブの機能を使って、ファイルに書かれた多数のジョブを分散実行します。
- オプション（基本的には qsubコマンドと同じ）
  - -q（queue名）　　… QUICK, SMALL, APC, APG, SDF（必須）
  - -l select=1:ncpus=(1行分のジョブ当たりのコア数):mem=(1行分のジョブ当たりの使用メモリ)
  - その他のqsub オプション（-N, -r, -v, -l, -p, -P, -W）
- ジョブを列挙したファイル
  - 1行につき1ジョブのコマンドを書いて下さい（; や && で区切って1行に複数のコマンドを列挙してもOKです）。
  - コマンドの出力をファイルにリダイレクトする場合は sh/bash 形式で書いて下さい。
    （コマンド） 1>xxx.out 2>xxx.err
  - # から始まる行はコメント行で無視されます（#PBS は機能しません）。
- 補足
  - qsubarraypbsコマンドの 実行前に、実行するアプリケーションを module load しておいて下さい。
  - qsubarraypbsの内部で cd $PBS_O_WORKDIR/ されます。
  - 最大500コア分(もしくはキューの上限値)のジョブが同時実行されます。
  - ディレクトリ pbslog/ にエラーメッセージ、エラーになったジョブ等が出力されます。
  - PBSのメール通知機能(-M, -m オプション)は使用できません。
  - （ジョブを列挙したファイル）は10,000行以内に収まるようにして下さい（PBSのアレイジョブの制限）

4.6.1 分割ジョブの作成: 入力ファイルの分割

- 指定した行数でファイルを分割

  $ wc -l（入力ファイル）　　# 全行数をカウント

  $ split -l（分割行数）-d -a 4（入力ファイル）（出力ファイル名）

  　　-d -a 4 … 出力ファイル名に4ケタ数字の連番を付加

- FASTAファイルの分割

  $ fasta_split（FASTAファイル）（分割数）

  $ module load fasta-splitter

  $ fasta-splilter --n-parts（分割数）（FASTAファイル）　# 分割数(--n-parts) or 分割サイズ(--part-size)を指定

- FASTQファイルの分割

  $ module load fastq-splitter

  $ fastq-splilter --n-parts（分割数）（FASTQファイル）　# 分割数(--n-parts) or 分割サイズ(--part-size)を指定

4.6.2 分割ジョブの作成: 連番が付いたジョブの作成

- seqコマンド : 連番の数字を出力するコマンド
- xargs: 標準入力から読み込んだ文字列を {} に代入して
  コマンドを実行(-i オプション)

```
$ seq -w 1 10 | xargs -i echo “command sample{}.pep“ >com.txt
$ cat com.txt
command sample01.pep
command sample02.pep
command sample03.pep
command sample04.pep
command sample05.pep
command sample06.pep
command sample07.pep
command sample08.pep
command sample09.pep
command sample10.pep
```

4.6.3 分割ジョブの作成: 入力ファイルのリストからジョブを作成

- parallel: 標準入力から読み込んだ文字列を代入してコマンドを実行 (xargsより高機能)

```
$ find /db/fasta/mgenes/T*.pep | parallel -k --dry-run "{} {.} {/} {//} {/.} {#}"
/db/fasta/mgenes/T30001.pep /db/fasta/mgenes/T30001 T30001.pep /db/fasta/mgenes T30001 1
/db/fasta/mgenes/T30002.pep /db/fasta/mgenes/T30002 T30002.pep /db/fasta/mgenes T30002 2
/db/fasta/mgenes/T30003.pep /db/fasta/mgenes/T30003 T30003.pep /db/fasta/mgenes T30003 3
/db/fasta/mgenes/T30004.pep /db/fasta/mgenes/T30004 T30004.pep /db/fasta/mgenes T30004 4
/db/fasta/mgenes/T30005.pep /db/fasta/mgenes/T30005 T30005.pep /db/fasta/mgenes T30005 5
...
```

{}	{.}	{/}	{//}	{/.}	{#}
そのまま表示	拡張子を取り除いて表示	ファイル名	ディレクトリ名	ファイル名（拡張子なし）	通し番号

以下のコマンドも同じ結果を与えます。

```
$ parallel --dry-run "{} {.} {/} {//} {/.} {#}" ::: /db/fasta/mgenes/T*.pep
```

例)

```
$ find /db/fasta/mgenes/T*.pep | parallel -k --dry-run "command {} 1>{/.}.out 2>{/.}.err" >com.txt
$ cat com.txt
command /db/fasta/mgenes/T30001.pep 1>T30001.out 2>T30001.err
command /db/fasta/mgenes/T30002.pep 1>T30002.out 2>T30002.err
command /db/fasta/mgenes/T30003.pep 1>T30003.out 2>T30003.err
…
```

4.7 インタラクティブバッチジョブ

（参）「使い方と注意事項」→「計算サービス」→「ジョブスクリプトの例」

- PBSのインタラクティブバッチジョブの機能を利用すると、計算ノードにログインして対話的に作業を行うことができます。
  - qsub に –I オプション(「I」ntaractive)を付けて下さい。
  - インタラクティブバッチジョブの利用は基本的にはSMALL キュー だけにして下さい。
  - ログインしっぱなしにならないようにご注意下さい。

```
[appadm@apfe1]$ qsub -I -q SMALL -l select=1:ncpus=10:mem=30gb -l walltime=12:00:00
qsub: waiting for job 205274.apfe3 to start
qsub: job 205274.apfe3 ready

cd /scratch/pbs_jobdir/pbs.205274.apfe3.x8z
[appadm@ap118]$ cd /scratch/pbs_jobdir/pbs.205274.apfe3.x8z
[appadm@ap118]$ cd $PBS_O_WORKDIR
```

4.8 ジョブの確認・削除

(参)「使い方と注意事項」→「計算サービス」→「投入ジョブ・キュー情報の参照」
「使い方と注意事項」→「計算サービス」→「ジョブ制御コマンド」

- ジョブ・キューのステータスの確認
  $ qstat [オプション] (Job ID or キュー名)
  - 主なオプション
    - -x # 終了したジョブも表示
    - -f [Job ID] # full format
    - -t [Job ID] # アレイジョブのサブジョブ毎に表示
    - -n1 [Job ID] # ジョブが実行されているホスト名を表示
    - -r # 実行中のジョブだけを表示
    - -q, -Q # 全queueのステータスを表示

＊ ジョブのステータス
Q: 実行待ち
R: 実行中
E: 終了処理中
F: 終了したジョブ
H: 保留状態
S: 中断中
B: 実行中のアレイジョブ
X: 終了したアレイジョブのサブジョブ

- ジョブの削除
  $ qdel [Job ID] [Job ID …]

- ジョブの実行の保留・保留解除 (qsubarraypbsで投入したジョブを一時的に止めたい場合など)
  $ qhold [Job ID] # ジョブの実行を保留(ステータスが Q のジョブのみ)
  $ qrls [Job ID] # ジョブの保留を解除

4.9 qstatmyjobsコマンド(＊)

(参)「使い方と注意事項」→「計算サービス」→「投入ジョブ・キュー情報の参照」

- ユーザの使用コア数・メモリ量、システムの利用状況の確認

```
$ qstatmyjobs
User: user
                      jobs                ncpus                              mem(gb)             walltime(h)
Queue   vacant(use%) sum/sum_max    avail max   sum/sum_max     avail   max   sum/sum_max    default max
-----   ------------ ------------- ----------------------- ---------------------------- ----------------
QUICK     2798( 52%)   0/2             4   4    0/unlimited      72    72    0/unlimited        1 01
SMALL     2656( 50%)   0/unlimited    12  12    0/36            108   108    0/324              6 12
APC       2536( 51%)  10/unlimited    32  40  110/unlimited     360   720  640/unlimited     2880 unlimited
APG        356( 44%)   0/4            40  40    0/unlimited     720   720    0/unlimited     2880 unlimited
SDF         62( 87%)   1/8             - 144   20/288           -   6144 6000/8192          2880 unlimited
=============================================================================================================
TOTAL                 11/unlimited           130/500                    6640/12288
```

(＊)PBSのコマンドではなく、当スパコンシステム独自のコマンドです

4.8 qstatmyjobsコマンド(続き)

項目	説明
Queue	キュー名
vacant	キューが利用可能な全コア数のうち、未使用（ジョブが実行されていない）のコア数
(use%)	キューが利用可能な全コア数のうち、使用中（ジョブが実行中）のコア数の割合(%)
jobs sum	ユーザが実行中のジョブの総数
jobs sum_max	ユーザが最大実行可能なジョブ数
avail ncpus	ジョブがすぐに実行開始できる最大コア数
max ncpus	そのキューで指定可能な最大コア数
sum ncpus	ユーザが実行中のジョブの合計コア数
sum_max ncpus	ユーザあたり、実行可能なジョブの最大合計コア数
avail mem(gb)	すぐに実行開始できる、ジョブの最大コア数指定時に、指定可能な最大メモリサイズ。 なお、もしジョブの最大コア数よりも少ないコア数を指定すると、より多くのメモリを指定しても、そのジョブがすぐに実行開始されることがあります。
max mem(gb)	そのキューで指定可能な最大メモリサイズ
sum mem(gb)	ユーザが実行中のジョブの合計メモリサイズ
sum_max mem(gb)	ユーザあたり、実行可能なジョブの最大合計メモリサイズ
default walltime(h)	何も指定しない場合に設定される経過時間
max walltime(h)	指定可能な最大経過時間

4.10 終了したジョブの実行情報の確認

(参)「使い方と注意事項」→「計算サービス」→「投入ジョブ・キュー情報の参照」

$ tracejob [オプション] [Job ID]

- 主なオプション
  -n（日数） … 何日遡ってログファイルを確認するか

```
$tracejob -n 2 191784.apfe3

Job: 191784.apfe3
…
04/19/2020 00:46:35  S    Exit_status=0 resources_used.cpupercent=149 resources_used.cput=00:08:18
                          resources_used.mem=39432kb resources_used.ncpus=4 resources_used.vmem=416888kb
                          resources_used.walltime=00:05:38
```

Exit_status	ジョブの終了ステータス（0でない場合はエラーが発生しています）
resources_used.cpupercent	平均CPU使用率
resources_used.mem	使用メモリ量
resources_used.ncpus	qsub時に指定したncpus
resources_used.walltime	ジョブの実行時間

(注) アレイジョブの場合は、後述の PbsExitStatus コマンドを使用して下さい。

4.10 終了したジョブの実行情報の確認(続き)

(参)「アプリケーション一覧」→「バイオインフォマティクス」→「PbsExitStatus」

- (アレイ)ジョブの実行情報の確認
  $ PbsExitStatus（*） [オプション] [Job ID]
  - -e ： Exit_status=0 でないもの(エラーが発生しているもの)だけを表示
  - -m ：使用メモリ(resource_used.mem) が最大のものを表示
  - -c ：出力行数をカウント
  - -t ：walltimeの合計を出力

例)

```
$ PbsExitStatus -e 213483[].apfe3
…
20200502:05/02/2020 13:22:44;0010;Server@apfe3;Job;213483[3583].apfe3;Exit_status=271
resources_used.cpupercent=99 resources_used.cput=00:17:02 resources_used.mem=33733660kb
resources_used.ncpus=2 resources_used.vmem=34207856kb resources_used.walltime=00:17:17
```

- Exit_status=0 でない場合は resources_used.mem や resources_used.walltime が
  指定した値(もしくはqueueのデフォルト値)を超えていないか確認

（＊）PBSのコマンドではなく、当スパコンシステム独自のコマンドです

4.11 SSDの利用 (SDFキュー限定)

- PBSでジョブを投入すると、ジョブの開始時に一時ディレクトリ
  /scratch/pbs_tmpdir/(JobID)/
  が自動的に作成されます。
  このディレクトリは ${TMPDIR}/ で利用することができます。

/scratch/ … SSD (sdf1, sdf2)
… Lustreファイルシステム
(PCクラスター)

- ジョブが終了すると ${TMPDIR}/ は自動的に削除されます。

- デノボアセンブリングツールなどIO負荷の高いアプリケーションを SDFキューで実行する際は、
  一時ディレクトリやファイルの出力先に ${TMPDIR} を指定して下さい。

- 使用例1) コマンドの一時ディレクトリを指定するオプションで ${TMPDIR}/を指定
  - spades --tmp-dir=${TMPDIR} …
  - megahit --tmp-dir ${TMPDIR} …
- 使用例2) コマンドの出力先に ${TMPDIR}/ を指定
  - mkdir ${TMPDIR}/output/
    flye -o ${TMPDIR}/output/ …
    mv ${TMPDIR}/output/ /aptmp/xxx/ # 計算結果を /aptmp/ に移動