# 2024年度 バイオユーザのための <mark>京大化研スパコンシステム</mark>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 1.2 サーバスペック 2利用にあたって 2.1 新規利用申請 2.2 利用負担金 2.3 スパコンシステムログイン手順 2.4 ディスク領域 2.4.1 ホーム領域 (ホームディレクトリ) 2.4.2 計算一時領域 (/aptmp/(ユーザ名)/) 2.4.3 ディレクトリサイズ・ファイル数の確認 2.5 ウェブアプリケーション 2.6 ownCloud計算サービス 2.7 システム稼働状況 3 アプリケーションの利用 3.1 moduleコマンド 3.2 バイオインフォマティクスアプリケーション 3.3 バイオインフォマティクスデータベース

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 4.2 ジョブの投入 4.3 キュー一覧 4.4 ジョブスクリプトの例 4.5 大規模入力ファイルの処理 4.6.1 入力ファイルの分割 4.6.2 連番が付いたジョブの作成 4.6.3 入力ファイルのリストからジョブを作成 4.7 インタラクティブバッチジョブ 4.8 ジョブの確認・削除 4.9 gstatmyjobsコマンド 4.10 終了したジョブの実行情報の確認 4.11 SSDの利用



# 1システム概要



## 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	大規模計算クラスタ ↓ (GPUノード)	フロントエンド ໜ サーバ
機種	HPE Superdome Flex	HPE Apollo2000 G10+	HPE DL380 G11	HPE DL380 G10+
ノード数	2	104	9	3
ホスト名	sdf1, sdf2	csXX, cmXX, clXX	ch1~ch9	fe1
CPU	Intel Xeon G6254 3.1GHz x 32CPU (18コア/CPU)	Intel Xeon G6348 2.6GHz x 2CPU (28コア/CPU)	Intel Xeon P8462Y+ 2.8GHz x 2CPU (32コア/CPU)	Intel Xeon G6346 3.1GHz x 2CPU (16コア/CPU)
コア数/ノード	576	56	64	32
メモリ/ノード	24TiB	256GiB (cs01∼cs44) 512GiB (cm01∼cm40) 1TiB (cl01∼cl20)	1TiB	512GiB
GPU	-	-	NVIDIA H100 x 2 (80GB)	NVIDIA A100 (80GB)
SSD	50TB	-	-	-
OS		Red Hat Enterpris	se Linux <sup>( * )</sup> 8.7	

(\*) 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
- 支払責任者が申請者となります。
- ■【新規利用申請書作成時の注意点】をお読み下さい。
  - ▶ お問い合わせ tel: 0774-38-3265 (内線: 3265) email: spradm@scl.kyoto-u.ac.jp



2.1 新規利用申請 (アカウントの取得) (続き)

(参)「各種手続き」→「新規利用申請」

- 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) <sup>注1,2)</sup>	0.00220円/秒		
CPU時間に対する課金の上限額	支払責任者の所属 京都大学化学研究所:40,000円/月 京都大学(化学研究所は除く):60,000円/月 学術機関(京都大学は除く):80,000円/月 民間機関:200,000円/月		* CPUノード 約10,000円/台/日
ディスク基本料金 (ホーム領域)	無料 (100GBまで利用可能) 🛛 🗮		
クラウドストレージサービス	無料 (100GBまで利用可能) 🛛 嬔		
オプションサービス			
ディスク拡張サービス(ホーム領域)	+300GB 1,000円/年度 (最大900GBを追加可能)	NEW	
クラウドストレージ拡張サービス	+300GB 1,000円/年度 (最大900GBを追加可能)	NEW	

■ 注1) 並列プログラムで同時に複数のCPUを利用した場合には、各CPUでのCPU時間の合計を課金の対象とします。

- 注2) GPUを利用した計算の場合、計算ジョブの経過時間(秒)をCPU時間と換算して課金します。
- 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 fe1 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@fe1. scl. kyoto-u. ac. jp

( × login.scl.kyoto-u.ac.jp)

(Windows) ターミナルソフト(MobaXterm, TeraTerm, Putty, Rlogin, 等) コマンドプロンプト

- fe1の GUIアプリケーション(seaview, LibreOffice, gnuplot, FigTree, IGV 等)を利用する場合
  - (Linux) ssh -Y (もしくは ssh -Y -C)でログイン。
  - (Mac) Xquartz をインストールして、ssh −Y (もしくは ssh −Y −C)でログイン。
  - (Windows) MobaXterm or 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」

#### 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」

■ 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。

\$ Idapchsh bash # bashに変更する場合

 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

#### 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」→「パソコンからの使い方」→「Windowsでファイル転送」 (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」

- 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, MobaXterm (Mac) Cyberduck, FileZilla
   (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

テキストファイルの改行コードをLinux用に変換
 \$ 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ではジョブ当たり、30分・8GBのCPU時間制限・メモリ制限があります。
- 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
~(ユーザ名)/	/aptmp/(ユーザ名)/	\${TMPDIR}/
NFS	Lustre	xfs
0	0	*
有	無	-
有	無	-
有	無	-
		SDFキューでのみ利用可 (§4.11)
	ホーム領域 ~(ユーザ名)/ NFS の 有 有 有	ホーム領域計算一時領域~(ユーザ名)/ NFS/aptmp/(ユーザ名)/OO有魚有無有無有無

## 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。

```
$ Is .snapshot/
daily.2024-05-09_0010 weekly.2024-04-28_0015 weekly.2024-05-05_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)



# 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・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)をお願いする場合があります
   ディスク使用量だけでなく、ファイル数にもご注意下さい!
- ディスク使用量の確認<sup>(注)</sup>



(注) Ifs-quota コマンドは /lustre/ ファイルシステムでの全使用量を出力するため、
 /aptmp/の使用量とずれが生じることがあります(/scratch/を使用している場合)

Snapshotの機能はありません(バックアップはご自身で取得して下さい)

#### 2.4.3ディレクトリサイズ・ファイル数の確認

\$ du -hs (ディレクトリ)/ # (ディレクトリ)/の全サイズ
\$ du -hS (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
\$ du -hS (ディレクトリ)/ | sort -h # ディレクトリのサイズでソート

\$ find (ディレクトリ)/ | wc -| # (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」

	京都大学化学研究所	i スーパーコンピュータ	システム
	システムの紹介 使い方と注意事項	各種手続き アプリケーショ	ン一覧 講習会
	新規利用者募集		
🖉 IceWall	ウェブアプリケーション		
	システム紹介		
Login	沿革		
スーパーコンピュータシステムで提供している以下のサービスを利用するための共通 認証画面になります。	共有メモリシステム	6 H 5 C 6 H	
・ ウェブメール カードフィール	計算クラスタ		
<ul> <li>フリッドストレーン</li> <li>スーパーコンビュータシステムユーザ限定ページ</li> <li>ビース・ビスコンビュータシステムユーザ限定ページ</li> </ul>	アプリケーション 重要な:	お知らせ:	
ユーザーIDとバスリートを入力して「ログイシ」ボダンを押してくたさい。 ユーザーID usemame			
パスワード password	京都大学化学研究所	スーパーコンピュータ	ンステム
ログイン	ウェブアプリケーション		
	ウェブメール	クラウドストレージ	ダウンロードセンター
ウェブメール、クラウドストレージページ内(			
ブアウト」ボタンは機能しません。			ログアウトパスワー

#### 2.6 ownCloud計算サービス

(参)「アプリケーションー覧」→「ownCloud計算サービス」 「アプリケーションー覧」→「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。
  - wmCloud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。
     エファイル
     「コークローク」
     「「ロール名」に入力ファイルを置くと
     「」
     「「ロール名」に入力ファイルを置くと
     「」
     「「ロール名」に入力ファイルを置くと
     「」
     「「ロール名」に入力ファイルを置くと
     「」
     「「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「」
     「
     「
     「」
     「」
     「
     「
     「
     「
     「
  - ➢ 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)

≡ ファイル	ownCloud			🔍 uehara 🗸
■ すべてのファイル	▲ GenomeNetCloud ≥ ete ≥ +			
★ お気に入り	□ 1個のファイル	+	ダウンロード 0B	削除 盲
< 他ユーザーがあなたと共有中	test.param	< kegg 🛛 ••	• < 1 KB	2時間前
< 他ユーザーと共有中	test.outTree_unrooted.nwk	< kegg 🚥	• < 1 KB	2時間前
	test.outTree_midpointRooted.nwk	< kegg 🚥	• < 1 KB	2時間前
Q 37	test.html	< kegg …	• < 1 KB	2時間前
	test.fasta.final_tree.used_alg.fa	< kegg 🛛 🚥	• 3 KB	2時間前
	test.fasta.final_tree.nwx	< kegg 🚥	• 3 KB	2時間前
	test.fasta.final_tree.nw	< kegg 🛛 ••	• < 1 KB	2時間前
	test.fasta.final_tree.fa	< kegg 🚥	• 3 KB	2時間前
	test.fasta	< kegg 🚥	• 3 KB	2時間前
✿ 設定				

### 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



■ リアルタイムでサーバの使用状況を確認: qstatmyjobs コマンド(§4.9)

User:	ei{lect-1}1001: qstatmyjobs -q ser: lect-1														
			JOBS		CP	PUS		MEM(	gb)		GP	US	WALLTIME(h)		
Queue	ıe avail(use%)		mysum/max	avail max mysum/max		avail	max	max mysum/max		max	mysum/max	default max			
SMALL	870(	85%)	0/UNLTD	12	12	0/96	48	48	0/UNLTD	-	-	-/-	6 12		
APC	702(	87%)	0/UNLTD	21	56	0/UNLTD	12	980	0/UNLTD			-/-	2880 UNLTD		
APG	480(	16%)	0/UNLTD	64	64	0/UNLTD	980	980	0/UNLTD	2	2	0/UNLT	2880 UNLTD		
SDF	608(	39%)	0/8	-	144	0/288	-	12288	0/UNLTD	-	-	-/-	2880 UNLTD		
TOTAL:		JOE	3S) 0/UNLTD		CPUS	5) 0/500		MEM)	0/18432		GPU	S) 0/4			

# 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。 # 利用できるアプリケーションの module を表示。 \$ module avail # 各アプリケーションの最新版だけを表示。 \$ module avail -L # 名前が bl から始まる module だけを表示(注: case sensitive)。 \$ module avail bl \$ 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: /db/(種別)/(データベース名)/
  - ▶ 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.\*
  - ▶ データベース更新情報: /db/dbinfo.txt
- /db/の実体は /lustre/db/YYYMMDD/ にあります。
  - ▶ 一ヶ月以上古いものは、.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。

NEW

 NCBI, EBI, PDBj のFTPサイト(一部)のミラー /db/ftp.ncbi.nih.gov/ /db/ftp.ebi.ac.uk/ /db/ftp.pdbj.org/

# 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

(\*) academic use only

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

# 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 (CPU, GPU, メモリ)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 fe1 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*)	18TB (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(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
  - - | xxxx ジョブの実行に必要なリソースを指定
    - 主なリソース
      - » −l 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: -I select=1:ncpus=(使用コア数):mem=(使用メモリ):ngpus=1
      - » ジョブの最大経過時間を指定:-I walltime=(時間):(分):(秒)
  - -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(※1)	SMALL	APC	<b>APG</b> <sup>(×3)</sup>	SDF
計算サーバ:台数	クラスタ(CPU): 104	クラスタ(CPU):101	クラスタ(GPU):9	大規模共有メモリ:2
ジョブの特徴	小規模	中~大規模	GPU	大規模メモリ
キューの実行順位	70	50	80	90
ジョブあたりの制限				
最大コア数 (デフォルト)	12 (1)	56 <sup>(※2)</sup> (1)	64 (1)	144 (18)
最大メモリ (デフォルト)	48GB (4 GB)	980 GB <sup>(※2)</sup> (4 GB)	980 GB (4 GB)	12 TB (768 GB)
最大経過時間 (デフォルト)	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)	-	-	4 (1)	8 (4)
合計コア数 (ソフトリミット)	96	500	-	288 (144)
合計メモリ(ソフトリミット)	_	9.8 TB	_	14 TB (6 TB)

(※1) QUICK キューは廃止されました
 (※2) 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・9.8TBまで使用可
 (※3) GPUは1ジョブ当たり最大2枚,1ユーザ当たり最大4枚までという制限があります。

#### 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
<pre>#PBS -1 select=1:ncpus=10:mem=40gb</pre>	←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 load blast+/2.15.0 cd \$PBS\_0\_WORKDIR/ ← moduleコマンドを使えるようにする

← blast+ を読み込み

← 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。 (要注意!) diamond, megahit, TraitRELAX

#### 4.5 大規模入力ファイルの処理 qsubarraypbs (参)「アプリケーション一覧」→「バイオインフォマティクス」→「gsubarravpbs」 \$ cat blastp.com qsub blastp -db db/nr -query query01.fa -out result01.out -outfmt 7 -num threads 10 qsub blastp -db db/nr -query query02.fa -out result02.out -outfmt 7 -num threads 10 qsub blastp -db db/nr -query query03.fa -out result03.out -outfmt 7 -num threads 10qsub blastp -db db/nr -query query04.fa -out result04.out -outfmt 7 -num threads 10qsub blastp -db db/nr -query query05.fa -out result05.out -outfmt 7 -num threads 10 qsub blastp -db db/nr -query query06.fa -out result06.out -outfmt 7 -num threads 10 qsub blastp -db db/nr -query query07.fa -out result07.out -outfmt 7 -num threads 10 qsub blastp -db db/nr -query query08.fa -out result08.out -outfmt 7 -num threads 10qsub blastp -db db/nr -query query09.fa -out result09.out -outfmt 7 -num threads 10 qsub blastp -db db/nr -query query10.fa -out result10.out -outfmt 7 -num threads 10 qsub blastp -db db/nr -query query11.fa -out result11.out -outfmt 7 -num threads 10 qsub blastp -db db/nr -query query12.fa -out result12.out -outfmt 7 -num threads 10qsub blastp -db db/nr -query query13.fa -out result13.out -outfmt 7 -num threads 10qsub blastp -db db/nr -query query14.fa -out result14.out -outfmt 7 -num threads 10 asub blastp -db db/nr -query query15.fa -out result15.out -outfmt 7 -num threads 10 qsub blastp -db db/nr -query query16.fa -out result16.out -outfmt 7 -num threads 10qsub blastp -db db/nr -query query17.fa -out result17.out -outfmt 7 -num threads 10 qsub blastp -db db/nr -query query18.fa -out result18.out -outfmt 7 -num\_threads 10-• • •

(\*)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」

#### 4.5 大規模入力ファイルの処理(続き)

\$ qsubarraypbs<sup>(\*)</sup> [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) ···· SMALL, APC, APG, SDF (必須)
- -I 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)
- 🔤 --max (同時実行数)
  - 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)。

#### 4.5 大規模入力ファイルの処理(続き)

#### ≻ 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。
- 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)

#### 🔤 🕨 同時実行数の変更

\$ qalter -Wmax\_run\_subjobs=(同時実行本数)(JobID)

(例)

\$ qalter -Wmax\_run\_subjobs=10 2225228[].fe3-adm

#### 4.6.1 分割ジョブの作成: 入力ファイルの分割

- FASTAファイルの分割
   \$ fasta\_split(FASTAファイル)(分割数) # 配列数が均等になるように分割

\$ module load fasta-splitter

\$ fasta-spliltter --n-parts(分割数)(FASTAファイル) # ファイルサイズが均等になる様に分割
 \$ fasta-spliltter --part-size(分割数サイズ)(FASTAファイル)

\$ module load seqkit
\$ seqkit split -O ./ -p(分割数)(FASTAファイル) # 配列数が均等になるように分割

■ FASTQファイルの分割

\$ module load fastq-splitter \$ fastq-spliltter --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 output{}.txt" >com.txt \$ cat com.txt

command sample01.pep output01.txt command sample02.pep output02.txt command sample03.pep output03.txt command sample04.pep output04.txt command sample05.pep output05.txt command sample06.pep output06.txt command sample07.pep output07.txt command sample08.pep output08.txt command sample09.pep output09.txt command sample10.pep output10.txt

## 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
```



#### 以下のコマンドも同じ結果を与えます。

```
$ parallel -k --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るのを防ぐため)。

[appadm@fe1]\$ qsub -I -q SMALL -I select=1:ncpus=10:mem=30gb -I walltime=12:00:00 qsub: waiting for job 205274.fe3-adm to start qsub: job 205274. fe3-adm ready

cd /scratch/pbs\_jobdir/pbs.205274.fe3.x8z [appadm@cs18]\$ cd /scratch/pbs\_jobdir/pbs.205274. fe3-adm.x8z [appadm@cs18]\$ 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(7日前まで)
  - -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
- ジョブの削除
   \$ gdel [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

\* 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

## 4.9 qstatmyjobsコマンド<sup>(\*)</sup>

(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: appadm

JOBS				BS	CPUS					MEM(gb)							GPUS	WALLTIME(h)			
Queue	avail(ι	use%)	mysum,	/max	avail	max	my	sum/max	ava	ail	max m	iysum/	max		avai	l ma	ax my	ysu	m/max	default	max
SMALL	1193 (	76%)	0,	/UNLTD	12	12		0/96		48	48	0/	ÚNL	TD		_	_		_/_	6	5 12
APC	1025 (	78%)	19,	/UNLTD	36	56		380/UNLT	D	780	980	1800/	ÚNL	TD		_	_		-/-	2880	) UNLTD
APG	56(	89%)	0,	/UNLTD	24	64		0/UNLT	D	765	980	0/	ÚNL	TD		2	2		0/UNLT	D 2880	) UNLTD
SDF	305 (	69%)	1,	/8	-	144		72/288		_	12288	2600/	UNL	TD		-	-		-/-	2880	) UNLTD
TOTAL:		JOB	s) 20,	/UNLTD		CPUS	s)	452/500			 Mem)	4400/	 ′184	.32			GPUS)	)	0/4		
fe3-adm	1:																				
	-										Req'd	Req'	d	ΕI	ap						
Job ID		Use	rname	Queue	Jol	bname	e	SessID	NDS	TSK	Memory	/Time	s	Ti	me						
14421. f	<sup>-</sup> e3-adm	app	adm	APC	MS	_FRJZ	ZR	15345*	1	30	120gb	2880	): R	59	:15	c 11	-adr	n/0	*30		
14425. f	e3-adm	app	adm	APC	MS	_FSI7	7Q	18073*	1	30	120gb	2880	): R	59	:13	c112	2-adr	n/0	*30		
15084. f 	€e3-adm	app	adm	APC	MS <u>.</u>	_H5QF	88	32758*	1	30	120gb	2880	): R	46	:13	cs10	)-adr	n/0	*30		

(\*)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

## 4.9 qstatmyjobsコマンド(続き)

\$ qstatmyjobs -m

#### 最もメモリが空いているノードを探す(-m オプション)

User:	appadm			00			10			MEN (~	· <b>b</b> . )				ODI	10		W/AT 1	ттмг	- (b)
Queue	avail(u	use%)	mysum/	max	avail	max m	iysum/max	ava	ail	max m	,b) Iysum/m	ax	ava	il r	nax n	nysu	m/max	defa		max
SMALL APC APG SDF	1193 ( 1025 ( 56 ( 305 (	76%) 78%) 89%) 69%)	0/ 19/ 0/ 1/	ÚNLTD ÚNLTD ÚNLTD 8	12 12 24 -	12 56 64 144	0/96 380/UNLT 0/UNLT 72/288	D D	48 880 765 –	48 980 980 12288	0/U 1800/U 0/U 2600/U	INL INL INL	TD TD TD TD	- - 2 -	- - 2 -		-/- -/- 0/UNLT -/-	2 D 2 2	6 880 880 880	12 UNLTD UNLTD UNLTD
TOTAL:		JO	BS) 20/	ÚNLTD		CPUS)	452/500			MEM)	4400/1	843	32		GPUS	S)	0/4			
fe3-ad	m:									<b>D</b> , , ,	<b>.</b> , ,									
Job ID	1	Us	ername	Queue	Job	oname	SessID	NDS	TSK	Req <sup>°</sup> d Memory	Req'd Time	S	Elap Time							
14421. 14425. 15084. 	fe3-adm fe3-adm fe3-adm	ap ap ap	padm padm padm	APC APC APC	MS_ MS_ MS_	_FRJZR _FSI7Q _H5QR8	15345* 18073* 32758*	1 1 1	30 30 30	120gb 120gb 120gb	2880: 2880: 2880:	R R R	59:15 59:13 46:13	cl cl cs	11-ac 12-ac 10-ac	dm/0 dm/0 dm/0	)*30 )*30 )*30			

# 4.8 qstatmyjobsコマンド(続き)

項目	説明
Queue	キュー名
avail	キューが利用可能な全コア数のうち、利用可能(ジョブが実行されていない)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: mysum	ユーザが実行中のジョブの総数
JOBS: max	ユーザが実行可能な最大ジョブ数
CPUS: avail	ジョブがすぐに実行開始できる最大コア数
CPUS: max	そのキューで指定可能な最大コア数
CPUS: mysum	ユーザが実行中のジョブの合計コア数
CPUS: max	ユーザあたり、実行可能なジョブの最大合計コア数
MEM(gb): avail	ジョブの最大コア数指定時に、すぐに実行開始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。
MEM(gb): max	そのキューで指定可能な最大メモリサイズ
MEM(gb): mysum	ユーザが実行中のジョブの合計メモリサイズ
MEM(gb): max	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
GPUS: avail	ジョブがすぐに実行開始できる最大GPU数
GPUS: max	そのキューで指定可能な最大GPU数
GPUS: mysum	ユーザが実行中のジョブの合計GPU数
GPUS: max	ユーザあたり、実行可能なジョブの最大合計GPU数
WALLTIME(h): default	何も指定しない場合に設定される経過時間
WALLTIME(h): max	指定可能な最大経過時間
fe3-adm: 以下	実行中のユーザジョブ情報(qstat -fnrt1) 左からジョブID、ユーザ名、バッチキュー、ジョブ名、セッションID、chunk数、コア数、メモリ、最大経過時間、ステータス、経過時間、実行ノード

#### 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.fe3-adm

Job: 191784.fe3-adm

•••

04/19/2024 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。
	Exit_status=-27 はメモリ超過でPBSにkillされたことを表します。)
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)「アプリケーションー覧」→「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[].fe3-adm

•••

20240502:05/02/2024 13:22:44;0010;Server@fe3-adm;Job;213483[3583].fe3-adm;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

(\*)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

# ご清聴ありがとうございました。

- お問い合わせ先 spradm@scl.kyoto-u.ac.jp 0774-38-3265 (内線: 3265) 京大化研スパコンシステム