スパコンの使い方

大阪大学 D3センター
大規模計算機システム担当

スーパーコンピュータ SQUID

- 3 種類の計算ノード と 21 PB のストレージで構成される
- 総理論演算性能は 16.591 PFLOPS

	汎用CPUノード	GPUノード	ベクトルノード
1ノードあたりのコア数	76	76	VH:24 VE:80
1ノードあたりの演算性能	5.837 TFLOPS	161.837 TFLOPS	25.61 TFLOPS
1ノードあたりのメモリ	256 GB	512 GB	VH:128 GB VE:384 GB
ノード数 （サーバ数）	1520ノード	42ノード （8GPU /ノード）	36ノード （8VE / ノード）

スーパーコンピュータ OCTOPUS

- 汎用CPUノード群 と 3.58 PB のストレージで構成される
- 総理論演算性能は 2.293 PFLOPS
- SQUID 汎用CPUノード群と比較し、1ノードあたりの性能が3倍以上に

	OCTOPUS 汎用CPUノード	SQUID 汎用CPUノード
1ノードあたりの コア数	256	76
1ノードあたりの 演算性能	16.384 TFLOPS	5.837 TFLOPS
1ノードあたりの メモリ	768 GB	256 GB
ノード数 （サーバ数）	140 ノード	1520ノード

スーパーコンピュータ利用の流れ

ユーザー端末

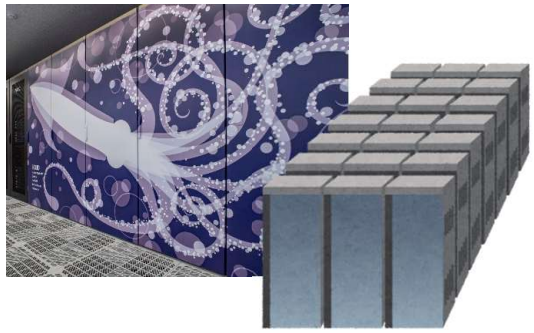
フロントエンド
ノードへの接続

スパコンの
フロントエンドノード
（作業用サーバ）

プログラム準備

ジョブスクリプト
作成

スパコンの
計算ノード
（計算用サーバ）



ジョブスクリプト投入

プログラム実行

フロントエンドノードへの接続

SSH (Secure Shell)接続

- ターミナル(Mac/Linux)やコマンドプロンプト(Win)を使用
- ターミナルソフトを使用(TeraTerm, Putty等)

接続先

[SQUID]　squidhpc.hpc.cmc.osaka-u.ac.jp
[OCTOPUS]　octopus.hpc.osaka-u.ac.jp

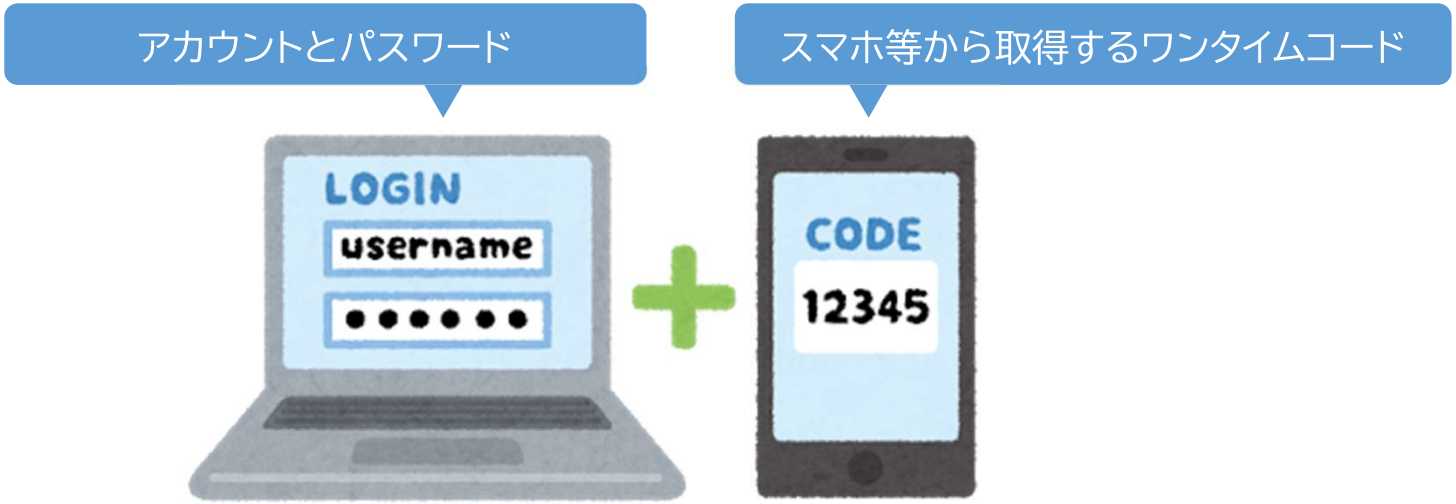
接続コマンド例

ssh　ユーザアカウント@squidhpc.hpc.cmc.osaka-u.ac.jp

学内/外、国内/外どこからでも接続可能

フロントエンドノードへの接続

SQUID と OCTOPUS は多要素認証でのログインとなります
多要素認証用の端末が必要です



※公開鍵認証には対応していません

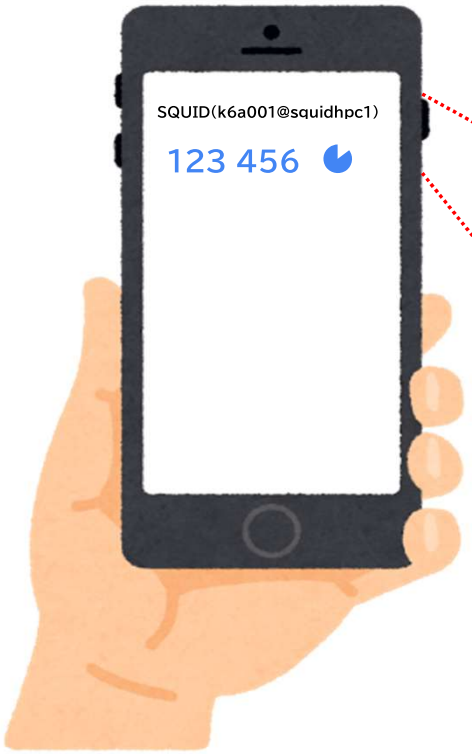
多要素認証用の端末

ご自身のスマートフォンやパソコンを多要素認証用の端末としてお使いください
以下いずれかのアプリケーションをインストールしてください

OS	アプリケーション	配布元
Android	Google Authenticator Microsoft Authenticator	Google Play Store
iOS		Apple App Store
Windows	WinAuth	Github
macOS	Step Two	Apple App Store

フロントエンドノードへの接続

SQUID と OCTOPUS に初めてログインすると QRコード が表示されます。
QRコード をアプリで読み込むことで 多要素認証 の登録が完了します



Initialize google-authenticator

Warning: pasting the following URL info your browser exposes the OTP secret to Google:

https://www.google.com/chart?chs=200*200&chld=M|0&cht=qr&otpauth://totp/user1@squidhpc.hpc.cmc.osaka-u.ac.jp%3Fsecret%3DDXXXXXXXXCLI%26issuer%3Dsquidhpc.hpc.cmc.osaka-u.ac.jp

「-1」を入力して
登録完了

Your new secret key is: 1A2B3C4D5E6F7G8

Enter code from app (-1 to skip): -1
Code confirmation skipped
Your emergency scratch codes are:

スーパーコンピュータ利用の流れ

ユーザー端末

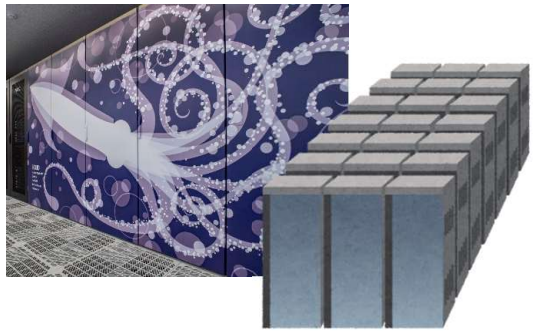
フロントエンド
ノードへの接続

スパコン
フロントエンドノード
（作業用サーバ）

プログラム準備

ジョブスクリプト
作成

スパコンの
計算ノード
（計算用サーバ）



ジョブスクリプト投入

プログラム実行

プログラムの準備

スパコンを利用するために
プログラムやアプリケーションを準備する必要があります

当センターの計算機で使用可能な主なプログラム言語
Fortran言語、C言語、C++言語、Python、R、Julia

当センターの計算機で使用可能な主なアプリケーション
OpenFOAM、LAMMPS、Gaussian、GROMACS
PyTorch、QuantumESPRESSO、etc…

必要なアプリケーションをご自身でインストールすることも可能です！

利用環境の設定

利用するプログラムやアプリケーションに応じて環境の設定が必要

Environment modulesというツールを使用

	Intelコンパイラ	NVIDIA HPC SDK	ベクトルコンパイラ	GNUコンパイラ
モジュール	BaseCPU	BaseGPU	BaseVEC	BaseGCC

などなど…

コマンド例

```
module load BaseCPU
  →Intelコンパイラのコマンド「ifx」が使用可能になる

module load BaseApp
module load gromacs
  →アプリケーション GROMACSが使用可能になる
```

プログラムの準備：まとめ

スパコンを利用するために、プログラムやアプリケーションを準備する必要があります

① 開発したC言語やFORTRAN言語のプログラムをお持ちの方

→スパコンにプログラムを持ってきて、コンパイルしましょう

② Pythonで機械学習をしている方

→スパコンで機械学習のフレームワーク等、Pythonパッケージを準備しましょう

③ オープンソースのアプリケーションで計算されている方

→スパコンに入力ファイル等必要なデータを持ってきましょう

→スパコンにアプリケーションをインストールしましょう

スーパーコンピュータ利用の流れ

ユーザー端末

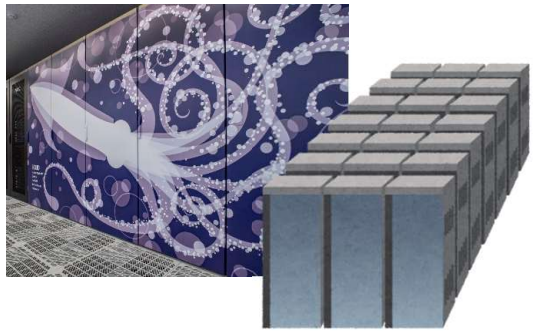
フロントエンド
ノードへの接続

スパコン
フロントエンドノード
（作業用サーバ）

プログラム準備

**ジョブスクリプト
作成**

スパコンの
計算ノード
（計算用サーバ）



ジョブスクリプト投入

プログラム実行

コンピュータの利用方法

インタラクティブ利用

コマンド等を通してコンピュータに直接命令し、リアルタイムで処理を実行

操作として手軽

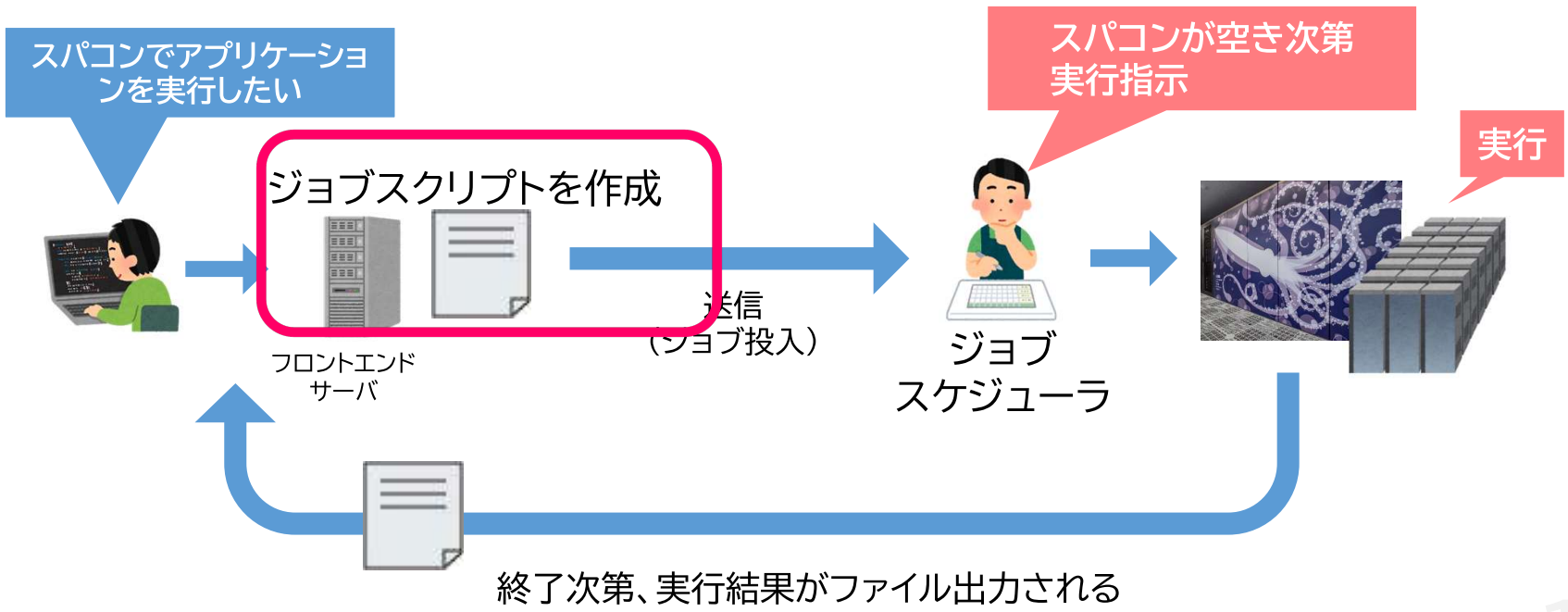
バッチ利用

コンピュータにまとめて処理を命令し実行

処理の命令が終われば、ログアウトしてもOK

バッチ利用

処理を「ジョブスクリプト」に記述し送信（→ジョブ）
ジョブスクリプトに基づき計算機が処理を実行



ジョブスクリプト

```
#!/bin/bash

#PBS -q SQUID
#PBS -l elapstim_req=1:00:00
#PBS -group=G12345

module load BaseCPU
cd $PBS_O_WORKDIR
./a.out
```

スパコンのリソースや環境設定

実行したい処理を記載したシェルスクリプト

ジョブスクリプト

```
#!/bin/bash
#PBS -q SQUID
#PBS -l elapstim_req=1:00:00
#PBS -group=G12345
module load BaseCPU
cd $PBS_O_WORKDIR
./a.out
```

使用する
リソースや環境

NQSオプション(#PBS～)でリソースや環境の設定を行う

オプション	説明
#PBS -q	**ジョブクラスを指定し、計算に使用する優先度等を指定する**
#PBS -l	**使用する資源値**
	elapstim_req ： ジョブの経過時間
	gpunum_job ： 1ノードあたりのGPU数
	cpunum_job ： 1ノード当たりのCPU数
#PBS --group	**グループ名の指定**
#PBS -v	環境変数の指定(setenvではなくこちらを使うことを推奨する)
#PBS -T	MPI 実行時に指定(IntelMPIの場合、#PBS -T intmpi と指定)

必須！

ジョブスクリプト

```
#!/bin/bash
#PBS -q SQUID
#PBS -l elapstim_req=1:00:00
#PBS -group=G12345
module load BaseCPU
cd $PBS_O_WORKDIR
./a.out
```

使用する
リソースや環境

ジョブクラス	利用可能経過時間	利用可能最大コア数	同時利用可能ノード数	備考
SQUID	120時間	38,912 Core (76Core × 512ノード)	512 ノード	
SQUID-H	120時間	38,912 Core (76Core × 512ノード)	512 ノード	高優先度
SQUID-S	120時間	38 Core (76Core × 0.5ノード)	0.5 ノード	ノード共有
DBG	10 分	152 Core (76Core × 2ノード)	2 ノード	デバッグ用

ジョブスクリプト

```
#!/bin/bash

#PBS -q SQUID
#PBS -l elapstim_req=1:00:00
#PBS --group=G12345

module load BaseCPU
cd $PBS_O_WORKDIR
./a.out
```

スパコンで
実行する処理

ファイルやディレクトリの実行・操作を記述(シェルスクリプト)

- 利用するプログラムやアプリケーションに応じて環境設定が必要
  →module loadを実施してください
- $PBS_O_WORKDIR :ジョブ投入時のディレクトリが設定される

ジョブスクリプト

```
#!/bin/bash

#PBS -q SQUID
#PBS -l elapstim_req=1:00:00
#PBS -group=G12345

module load BaseCPU

cd $PBS_O_WORKDIR

./a.out
```

ジョブクラスの指定

リソースの指定

環境設定

ジョブ投入時のディレクトリへ移動

a.outを実行する

スーパーコンピュータ利用の流れ

ユーザー端末

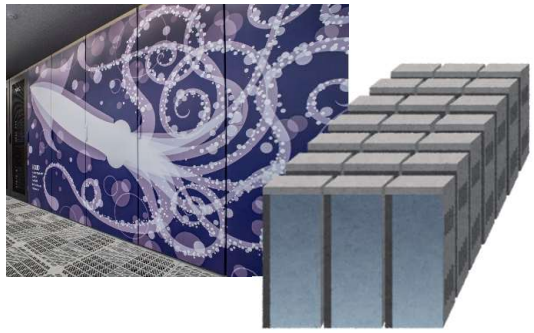
フロントエンド
ノードへの接続

スパコン
フロントエンドノード
（作業用サーバ）

プログラム準備

ジョブスクリプト
作成

スパコンの
計算ノード
（計算用サーバ）

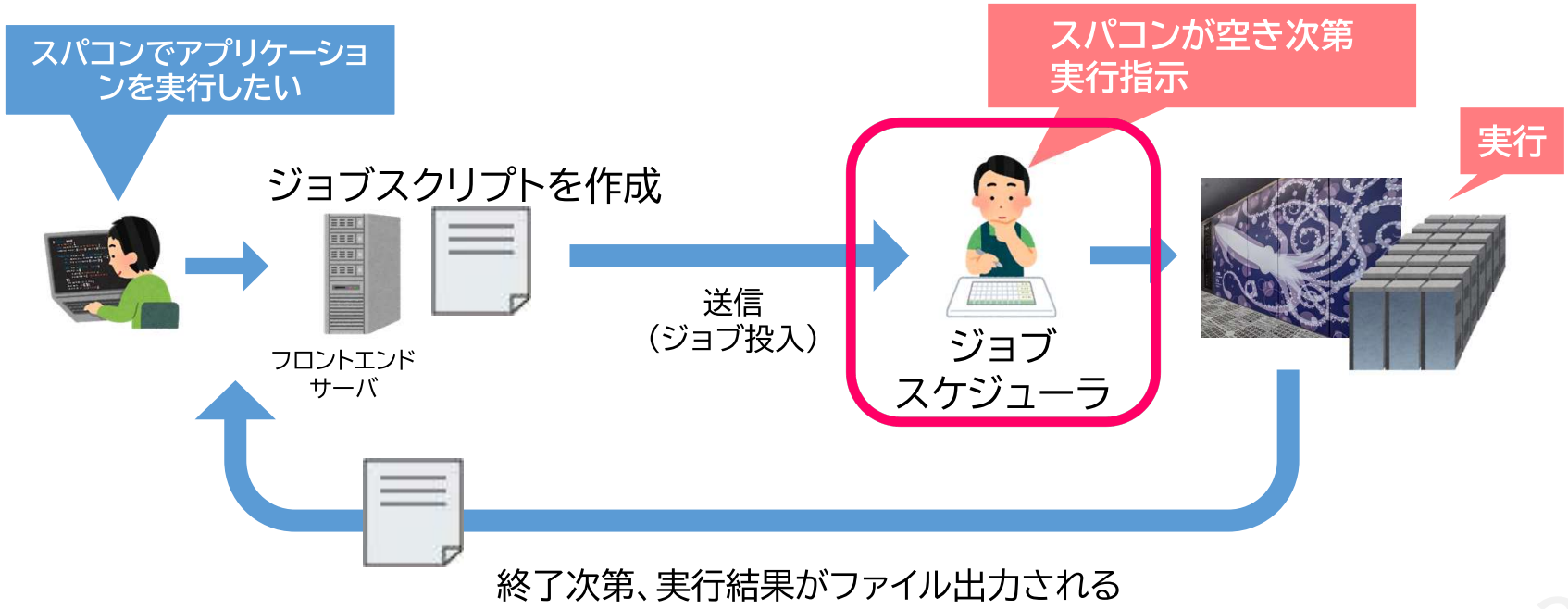


ジョブスクリプト投入

プログラム実行

バッチ利用

処理を「ジョブスクリプト」に記述し送信（→ジョブ）
ジョブスクリプトに基づき計算機が処理を実行



ジョブスケジューラとは

あらかじめ管理者によって設定された割当ポリシーに従い、ジョブを計算ノードに割り当てるソフトウェア

主な役割

スパコンの各ノードのストレージ容量、メモリ容量、性能、使用率を定期的に監視、管理

ユーザより実行したいジョブ要求を受信し、適切なノードを選定

ジョブ実行に伴う入出力データのファイル転送

ジョブスケジューラとは

当センターではバックフィル型を採用

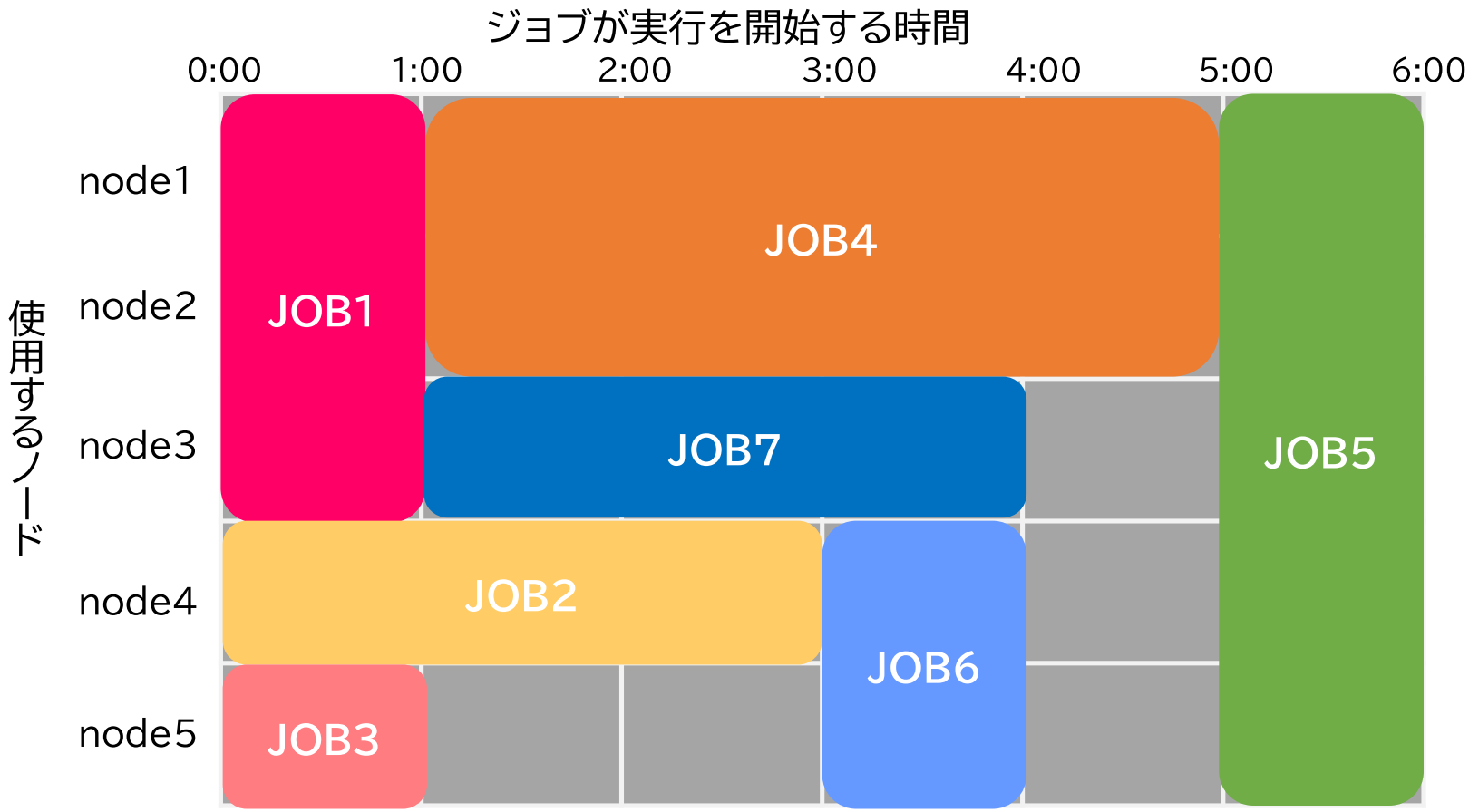
特徴

ジョブの実行開始時間のマップを作成する

マップに載れば、実行開始時間が保障される

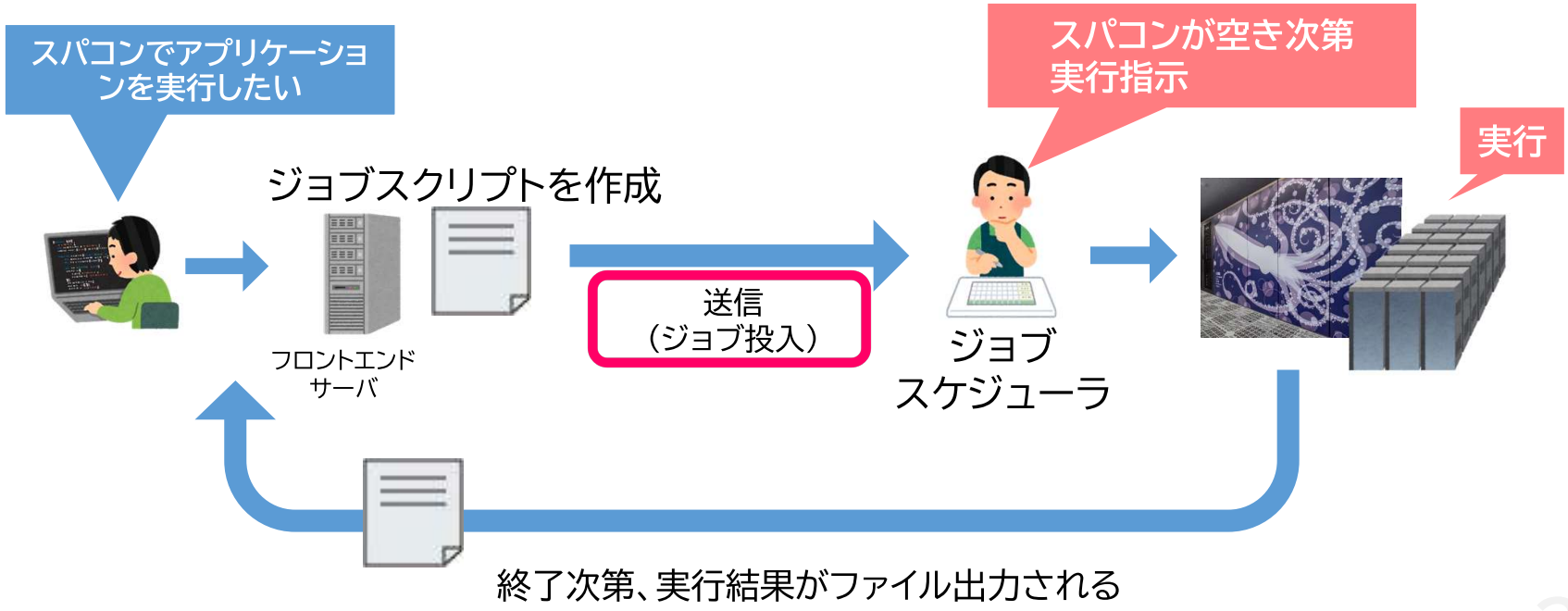
実行中は指定したリソースを占有して割り当てる

ジョブスケジューラのイメージ



バッチ利用

処理を「ジョブスクリプト」に記述
スクリプトに基づき計算機が処理を実行



ジョブの操作方法

ジョブの投入コマンド

$ qsub [ジョブスクリプトファイル]

投入に成功すると
“Request [リクエストID] submitted to queue: ジョブクラス名”
と表示され、ジョブごとにリクエストIDという通し番号が付与される
リクエストIDの例：123456.sqd

ジョブのキャンセルコマンド

$ qdel [リクエストID]

キャンセルに成功すると
“Request [リクエストID] was deleted”と表示される

投入済みジョブの確認方法

ジョブの状態確認コマンド

```
$ qstat
```

```
RequestID   ReqName    UserName   Queue   STT   Memory   CPU      Elapse
-------------------------------------------------------------------------
12345.sqd   nqs-test   k6a001     SQUID   RUN   8.72G    830.66   208
```

ジョブの状態

待ち状態では「**QUE**」
実行が始まると「**RUN**」となる。

実行時間

CPU : 実際にジョブが消費した時間
複数CPU指定の場合は、全CPUを累積表示

Elapse : ジョブが実行されてからの経過時間

ジョブのスケジューリング状況確認コマンド

```
$ sstat
```

```
RequestID   ReqName    UserName   Queue   Pri               STT   PlannedStartTime
-------------------------------------------------------------------------------------
12345.sqd   nqs-test   k6a001     SQUID   -1.5684/ -1.5684  QUE   2023-12-07 13:30:23
```

状態監視

混雑具合や優先度により、「実行時間の決定」までの待ち時間が異なるが、
一旦実行時間が決定されるとその時刻にジョブ実行が始まる。

実行開始時刻

システムメンテナンスやトラブル時は
再スケジュールされることをご了承ください。

実行結果の確認方法

実行結果や実行エラーは指定しない限り
実行結果：ジョブスクリプト名.oリクエストID
実行エラー：ジョブスクリプト名.eリクエストID
というファイル名で自動出力される

catやlessコマンドでファイルの内容を出力し確認

```
$ cat jobscript.nqs.o123456
```

意図通りの結果が表示されていれば計算は成功！

プログラムの実行：まとめ

スパコンでプログラムやアプリケーションを実行する際は**「バッチ利用」**

① **ジョブスクリプトを作成する**
→右のようなファイルを作成

```
#!/bin/bash

#PBS -q SQUID
#PBS -l elapstim_req=1:00:00
#PBS --group=G12345

module load BaseCPU
cd $PBS_O_WORKDIR
./a.out
```

② **ジョブスクリプトをSQUIDに送信する**
→qsubコマンドを使用

③ **定期的にジョブの状態を確認する**
→qstatやsstatコマンドを使用

④ **実行終了したら結果を確認する**
→サーバ上に保存されたファイルを開いて確認

さらなるスパコン利用に向けて

利用の参考になるWebページ

D3センター 大規模計算機システム Webページ
http://www.hpc.cmc.osaka-u.ac.jp

利用方法
http://www.hpc.cmc.osaka-u.ac.jp/system/manual/
FAQ
http://www.hpc.cmc.osaka-u.ac.jp/faq/
問い合わせフォーム
http://www.hpc.cmc.osaka-u.ac.jp/support/contact/auto form/
研究成果
http://www.hpc.cmc.osaka-u.ac.jp/researchlist/

さらなるスパコン利用に向けて

本日以降の講習会(全てオンライン)

開催日	講習会名	概要
6/1	初めてのスパコン	スーパーコンピュータの基礎的な知識と、その使い方 初心者向け【お試しアカウント付き】
6/3	スパコンに通じる並列プログラミングの基礎	並列プログラミングの手法や考え方の基礎
6/5	OpenMP入門	OpenMPによる一般的な並列プログラミングの基礎とその利用方法
6月～7月 予定	汎用CPUノード 高速化技法の基礎	Intelコンパイラの基礎的なチューニング手法の説明【お試しアカウント付き】
	GPUプログラミング入門(OpenACC)	GPUおよびGPUを用いた計算の特徴や概要を説明【お試しアカウント付き】
	スーパーコンピュータ バッチシステム入門 / 応用	計算機利用(バッチ利用)の概要 【お試しアカウント付き】
	GPUプログラミング実践(OpenACC)	OpenACCによる応用的なGPUプログラミングを説明【お試しアカウント付き】

その他講習会も実施予定(過去の講習会をご参考ください)
https://www.hpc.cmc.osaka-u.ac.jp/lecture_event/

スパコン利用方法のまとめ

- ご自身で開発したプログラム、オープンソースのアプリケーション等、柔軟に使用可能
- スパコンは「バッチ利用」
  - たくさんの人が同時に、計算規模に応じてスパコンを切り出して使う
  - ジョブスクリプトを使って、スパコンに計算を指示
- スパコンを使ってみたい方は講習会の参加もおすすめです！
- 疑問があれば system@cmc.osaka-u.ac.jp まで！

まずは試用制度をお試しください！

3カ月間 **無料**で以下の資源をご提供

SQUID

75 SQUIDポイント
＋ ストレージ 5 TB

- 汎用CPU ノード 294 ノード時間
- GPU ノード 48 ノード時間
- ベクトルノード 78 ノード時間

OCTOPUS
Osaka university Compute & sTOrage Platform Urging open Science

150 OCTOPUSポイント
＋ ストレージ 5 TB

- 汎用CPU ノード 294 ノード時間

- アプリケーション等 計算環境や技術サポートは**有償利用と同等に使用可能**
- 有償利用へアカウントの移行も可能