

お試しアカウント付き 並列プログラミング講習会 Oakforest-PACS利用の手引き

東京大学情報基盤センター



Oakforest-PACS (OFP)への公開鍵の登録

- Webブラウザで登録用ページにアクセスする

<https://ofp-www.jcahpc.jp/>

- ユーザ名とパスワードを聞かれるので、
センターから発行されたユーザ名とパスワードを入れる。

ポータル画面（ログイン前）

Oakforest-PACS 利用支援ポータル

[English/Japanese]

ログイン

ログイン

ユーザ名とパスワードを入力して【ログイン】ボタンをクリックしてください。

ユーザ名: パスワード:

ログイン リセット

JavaScript、Cookie を有効にしてお使いください。

- 動作確認済みブラウザ -

- ・ Internet Explorer バージョン 8 以上
- ・ Safari バージョン 5 以上
- ・ Firefox バージョン 9 以上
- ・ Google Chrome バージョン 18 以上

センターから配られた
利用者番号 と パスワード
を入力する

Copyright 2016 FUJITSU LIMITED

パスワード変更

- 最初のログイン時にパスワード変更を求められるので、新しいパスワードを入力してください。

ログアウト

パスワード変更

本機能で変更可能なパスワードは、Oakforest-PACSシステムの利用支援ポータル用パスワードです。

現在のパスワード	<input type="password"/>
新しいパスワード	<input type="password"/>
新しいパスワード(再入力)	<input type="password"/>

変更

新規パスワード規約

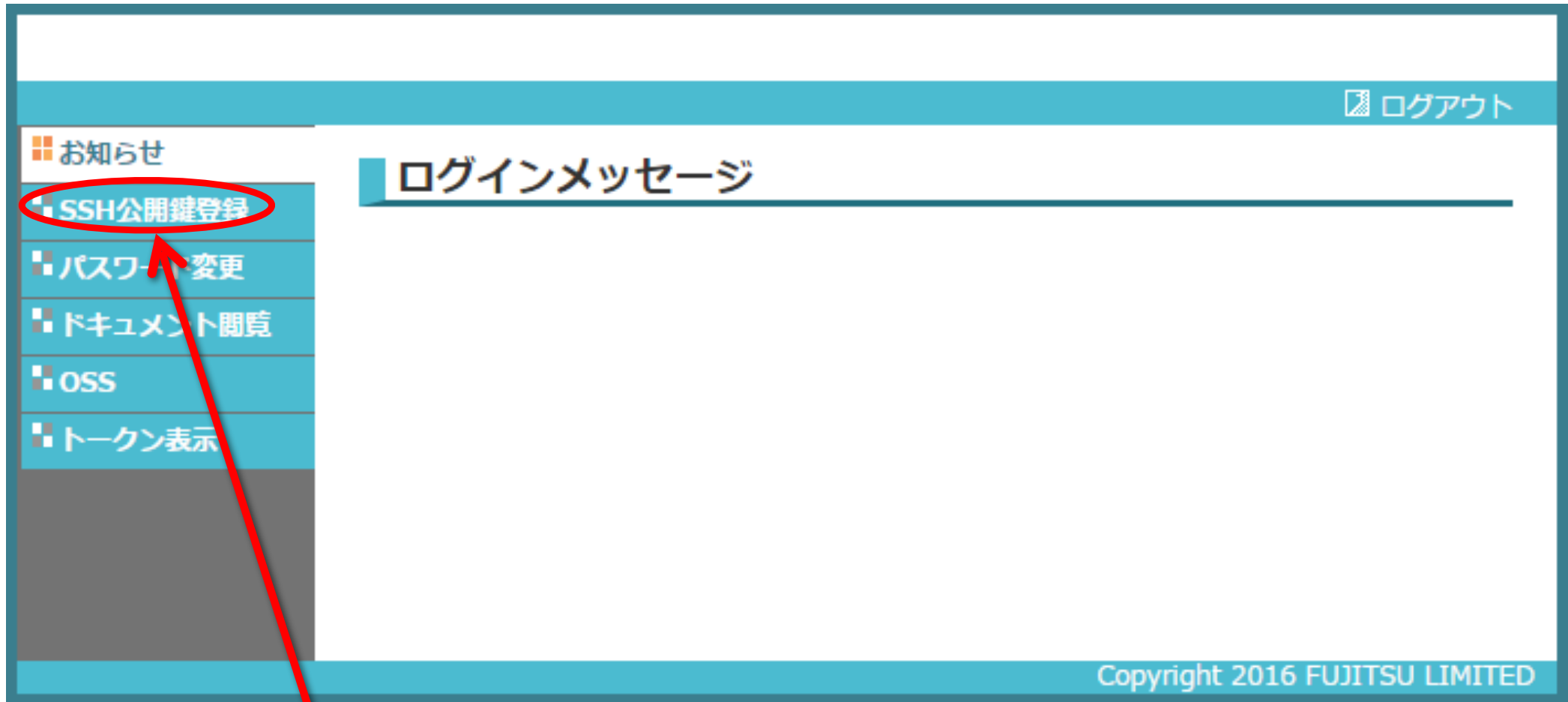
- 8文字以上の文字列を指定してください。
- 現在のパスワードと3文字以上異なる文字列を指定してください。
- 1つ以上の英字小文字、1つ以上の英字大文字、1つ以上の数字、1つ以上の特殊文字を指定してください。
- 使用可能な特殊文字は以下の通りです。
>> 空白、!、"、#、\$、%、&、'、(、)、*、+、,、-、.、/、:、;、<、=、>、?、@、[、\、]、^、_、`、{、|、}、~
- Linuxの辞書に登録されている文字は使用できません。
- 全角文字は使用できません。

Copyright 2016 FUJITSU LIMITED

鍵の登録

1. 左側メニューの「SSH公開鍵登録」をクリックする
2. 「公開鍵を追加」をクリックし、画面に公開鍵をコピーアンドペーストする
3. 「作成」ボタンを押す

ポータル画面



ここをクリック

ポータル画面（公開鍵登録）

The screenshot shows the 'SSH公開鍵登録' (SSH Public Key Registration) page. The left sidebar contains navigation links: お知らせ, SSH公開鍵登録, パスワード変更, ドキュメント閲覧, OSS, and トークン表示. The main content area has a 'ログアウト' button in the top right. The registration form includes:

- 登録方式** (Registration Method): Radio buttons for '直接入力' (Direct Input) and 'ファイルアップロード' (File Upload). 'ファイルアップロード' is selected and circled in red.
- SSH公開鍵** (SSH Public Key): A file selection field showing 'id_rsa.pub' and radio buttons for key types: RSA, DSA, ECDSA256, ECDSA384, ECDSA521, and Ed25519. 'RSA' is selected and circled in red.
- 登録** (Register): A button circled in red.

Below the form, there is a warning section: '公開鍵登録の際、以下の点にご注意ください。' (Please pay attention to the following points when registering the public key.)

- 登録する公開鍵の暗号化方式(RSA、DSA、ECDSA、Ed25519)を正しく選択していること。
- RSA公開鍵の場合、2048bit 以上で公開鍵を作成していること。
- DSA公開鍵の場合、1024bit 以上で公開鍵を作成していること。
- ECDSA公開鍵の場合、256bit、384bitもしくは521bit で公開鍵を作成していること。
- Ed25519公開鍵の場合、256bit で公開鍵を作成していること。
- 全角文字などの不正文字が含まれないこと。
- 複数の鍵登録はできません。(複数追加してください)

Red annotations include:

- A red circle around the 'ファイルアップロード' radio button.
- A red circle around the 'RSA' radio button.
- A red circle around the '登録' button.
- A red arrow pointing from the '登録' button to a text box containing '指定後クリック' (Click after specification).
- A red arrow pointing from the 'RSA' radio button to a text box containing '“RSA”であることを確認' (Confirm it is "RSA").
- A red arrow pointing from the file selection field to a speech bubble containing '~/.ssh/id_rsa.pub'.

ポータル画面（公開鍵登録：別のやり方）

The screenshot shows the 'SSH公開鍵登録' (SSH Public Key Registration) page. On the left is a navigation menu with items like 'お知らせ', 'SSH公開鍵登録', 'パスワード変更', 'ドキュメント閲覧', 'OSS', and 'トークン表示'. The main content area has a '登録方式' (Registration Method) section with radio buttons for '直接入力' (Direct Input) and 'ファイルアップロード' (File Upload). Below this is a text area for the public key, with a red box around the text '公開鍵をペースト' (Paste public key) and a speech bubble containing the terminal command 'cat ~/.ssh/id_rsa.pub'. At the bottom of the form is a '登録' (Register) button, which is circled in red with an arrow pointing to a red box containing the text 'ペースト後クリック' (Click after pasting). At the bottom of the page, there is a note: '公開鍵登録の際、以下の点にご注意ください。' (Please pay attention to the following points when registering the public key.) followed by a list of items: '・改行文字が含まれていないこと。(特に末尾に改行が含まれていないことに注意してください)' (No line breaks are included. (Please pay attention to the fact that no line breaks are included at the end.)) and '・ヘッダ(ssh-rsa, ssh-dss, ecdsa-sha2-nistp256, ecdsa-sha2-nistp384, ecdsa-sha2-nis'.

Oakforest-PACSへログイン

- ターミナルから、以下を入力する

```
$ ssh txxxxx@ofp.jcahpc.jp
```

「txxxxx」は利用者番号

- 接続するかと尋ねられるので、 `yes` を入力する
- 鍵の設定時に入れた
自分が決めたパスフレーズ
を入力する
- 成功すると、ログインができる

Oakforest-PACSのデータをPCに取り込む

- ターミナルでscpコマンドを使う
- `$ scp txxxxx@ofp.jcahpc.jp:~/a.f90 ./`
「txxxxx」は利用者番号
 - OFP上のホームディレクトリにある `a.f90` をPCのカレントディレクトリに取ってくる
 - ディレクトリごと取ってくる場合は `-r` を指定する
- `$ scp -r txxxxx@ofp.jcahpc.jp:~/SAMP ./`
 - OFP上のホームディレクトリにあるSAMPフォルダを、その中身ごと、PCのカレントディレクトリに取ってくる

PCのファイルをOakforest-PACSに置く

- 同様にターミナルでscpコマンドを使う
- `$ scp ./a.f90 txxxxx@ofp.jcahpc.jp:`
「txxxxx」は利用者番号
 - PCのカレントディレクトリにある `a.f90` を、OFP上のホームディレクトリに置く
 - ディレクトリごと置くには、`-r` を指定する
- `$ scp -r ./SAMP txxxxx@ofp.jcahpc.jp:`
 - PCのカレントディレクトリにあるSAMPフォルダを、その中身ごと、OFP上のホームディレクトリに置く

Oakforest-PACSにおける注意

- `/home` ファイルシステムは容量が小さく、ログインに必要なファイルだけを置くための場所です。
 - `/home` に置いたファイルは計算ノードから参照できません。ジョブの実行もできません。
- 転送が終わったら、`/work`ファイルシステムに移動(`mv`)してください。
- または、直接 `/work`ファイルシステムを指定して転送してください。
- ホームディレクトリ: `/home/txxxxx`
 - `cd` コマンドで移動できます。
- Workディレクトリ: `/work/gt00/txxxxx`

Oakforest-PACSスーパーコンピュータシステムでのジョブ実行形態

- 以下の2通りがあります
- **インタラクティブジョブ実行**
 - PCでの実行のように、コマンドを入力して実行する方法
 - スパコン環境では、あまり一般的でない
 - デバック用、大規模実行はできない
 - OFPでは、以下に限定
 - 1ノード（68コア）：2時間まで
 - 16ノード（1,088コア）：10分まで
- **バッチジョブ実行**
 - バッチジョブシステムに処理を依頼して実行する方法
 - 実行させたい処理をファイル（ジョブスクリプト）で指示する
 - スパコン環境で一般的
 - 大規模実行用
 - OFPでは、最大2048ノード（139,264コア）（24時間）

※講習会アカウントでは
バッチジョブ実行のみ、
最大16ノード15分まで

Oakforest-PACSスーパーコンピュータシステムでのジョブ実行形態(2)

- 2つの異なるメモリモードを用意
 - Flatモード
 - MCDRAMとDDR4メモリを個別にアクセス可能
 - Cacheモード
 - MCDRAMはDDR4メモリのキャッシュとして働く
- 各ジョブキューには、**-flat**, **-cache** をそれぞれ用意

インタラクティブ実行のやり方

- コマンドラインで以下を入力

- 1ノード実行用

- `$ pjsub --interact -g gt00 -L rg=interactive-{flat,cache},elapse=01:00`

※コマンドは改行せず1行で入力すること

- 16ノード実行用

- `$ pjsub --interact -g gt00 -L rg=interactive-{flat,cache},node=16,elapse=01:00`

※インタラクティブ用のノードがすべて使われている場合、資源が空くまで、ログインできません。
※講習会アカウントでは使えません。

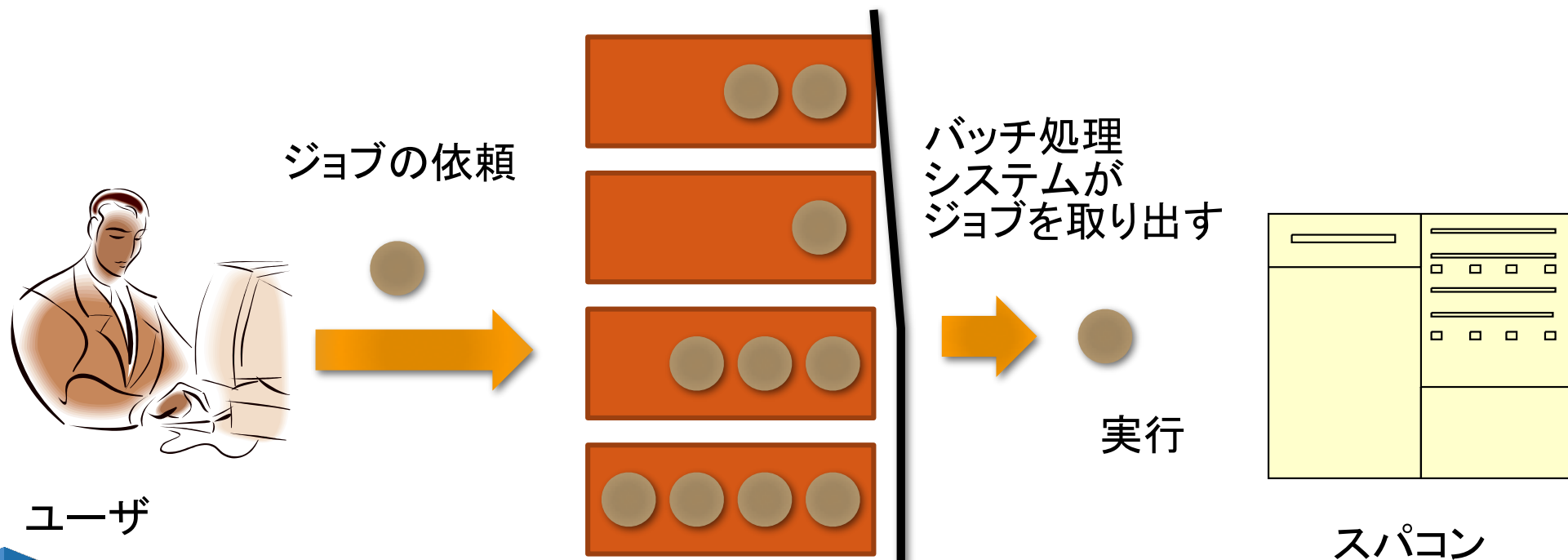
コンパイラの種類とインタラクティブ実行およびバッチ実行

- OFPでは、コンパイラはバッチ実行、インタラクティブ実行で共通に使えます。
- 例) Intelコンパイラ
 - Cコンパイラ: `icc`, `mpiicc` (Intel MPIを使う場合)
 - Fortran90コンパイラ: `ifort`, `mpiifort` (Intel MPIを使う場合)
 - KNL向け最適化: `-xMIC-AVX512`
 - ログインノードやプレポスト用ノードで実行する可能性もある場合:
`-axMIC-AVX512`

バッチ処理とは

- スパコン環境では、通常は、インタラクティブ実行（コマンドラインで実行すること）はできません。
- ジョブは**バッチ処理**で実行します。

バッチキュー



バッチキューの設定のしかた

- OFPでのバッチ処理は、富士通のバッチシステムで管理されています。
- 以下、主要コマンドを説明します。
 - ジョブの投入：`pjsub <ジョブスクリプトファイル名>`
 - 自分が投入したジョブの状況確認：`pjstat`
 - 投入ジョブの削除：`pjdel <ジョブID>`
 - バッチキューの状態を見る：`pjstat --rsc`
 - バッチキューの詳細構成を見る：`pjstat --rsc -x`
 - 投げられているジョブ数を見る：`pjstat --rsc -b`
 - 過去の投入履歴を見る：`pjstat -H`
 - 同時に投入できる数／実行できる数を見る：`pjstat --limit`

ジョブスクリプトの例

※実行させたい処理によって各項目の内容は異なります

```
#!/bin/bash
#PJM -L rscgrp=lecture-flat
#PJM -L node=2
#PJM --mpi proc=4
#PJM --omp thread=16
#PJM -L elapse=0:01:00
#PJM -g gt00
mpiexec.hydra -n ${PJM_MPI_PROC} ./a.out
```

リソースグループ名

利用ノード数

MPIプロセス数

プロセスあたりのスレッド数

経過時間制限値 (注)

利用グループ名

プログラムを実行

注) 空いている隙間リソースに割りあてるバックフィルスケジューリング機能により実行開始が早まる可能性が高くなるので、できるだけ事前に経過時間を見積って、適切な経過時間制限値を設定したほうが良い

本お試し講習会でのキュー・グループ名

- 本演習中のキュー名：
 - tutorial-flat (演習によっては tutorial-cache も使える場合がある)
 - 最大15分まで
 - 最大ノード数は16ノード(1088コア) まで
- 本演習時間以外 (24時間) のキュー名：
 - lecture-flat (演習によっては lecture-cache も使える場合がある)
 - 利用条件は演習中のキュー名と同様
- グループ名：gt00

pjstat --rsc の実行画面例

```
$ pjstat --rsc
RSCGRP                STATUS                NODE
regular-cache
|---- small-cache    [ENABLE,START]       3846
`---- medium-cache   [ENABLE,START]       3846
regular-flat
|---- small-flat     [ENABLE,START]       3846
`---- medium-flat    [ENABLE,START]       3846
interactive-cache
|---- interactive_n1-cache [ENABLE,START]       100
`---- interactive_n16-cache [ENABLE,START]       100
interactive-flat
|---- interactive_n1-flat [ENABLE,START]       100
`---- interactive_n16-flat [ENABLE,START]       100
debug-cache           [ENABLE,START]       234
debug-flat            [ENABLE,START]       234
prepost               [ENABLE,START]       12
```

使える
キュー名
(リソース
グループ)

現在
使えるか

ノードの
利用可能数

pjstat --rsc -x の実行画面例

```

$ pjstat --rsc -x
RSCGRP                STATUS                MIN_NODE  MAX_NODE  MAX_ELAPSE  REMAIN_ELAPSE  MEM(GB)  PROJECT
regular-cache
|---- small-cache    [ENABLE,START]        1         128      48:00:00      48:00:00      82  pz0105
`---- medium-cache   [ENABLE,START]       129        512      48:00:00      48:00:00      82  pz0105
regular-flat
|---- small-flat     [ENABLE,START]        1         128      48:00:00      48:00:00      96  pz0105
`---- medium-flat    [ENABLE,START]       129        512      48:00:00      48:00:00      96  pz0105
interactive-cache
|---- interactive_n1-cache [ENABLE,START]        1           1       02:00:00      02:00:00      82  pz0105
`---- interactive_n16-cache [ENABLE,START]        2          16       00:10:00      00:10:00      82  pz0105
interactive-flat
|---- interactive_n1-flat [ENABLE,START]        1           1       02:00:00      02:00:00      96  pz0105
`---- interactive_n16-flat [ENABLE,START]        2          16       00:10:00      00:10:00      96  pz0105
debug-cache          [ENABLE,START]        1         128       00:30:00      00:30:00      82  pz0105
debug-flat           [ENABLE,START]        1         128       00:30:00      00:30:00      96  pz0105
prepost              [ENABLE,START]        1           1       06:00:00      06:00:00     222  pz0105

```

↑
 使える
 キュー名
 (リソース
 グループ)

↑
 現在
 使えるか

↑
 ノードの
 実行情報

↑
 課金情報(財布)
 実習では1つのみ

pjstat --rsc -b の実行画面例

```

$ pjstat --rsc -b
RSCGRP                STATUS                TOTAL RUNNING QUEUED  HOLD OTHER  NODE
regular-cache
|---- small-cache    [ENABLE,START]       45      40      5      0      0    3846
`---- medium-cache   [ENABLE,START]        1       1       0      0      0    3846
regular-flat
|---- small-flat     [ENABLE,START]     150     120     30      0      0    3846
`---- medium-flat    [ENABLE,START]       7        3       4      0      0    3846
interactive-cache
|---- interactive_n1-cache [ENABLE,START]      0        0       0      0      0    100
`---- interactive_n16-cache [ENABLE,START]      0        0       0      0      0    100
interactive-flat
|---- interactive_n1-flat [ENABLE,START]      1         1       0      0      0    100
`---- interactive_n16-flat [ENABLE,START]      0         0       0      0      0    100
debug-cache          [ENABLE,START]       7         4       3      0      0    234
debug-flat           [ENABLE,START]       0         0       0      0      0    234
prepost              [ENABLE,START]       0         0       0      0      0    12

```

使える
キュー名
(リソース
グループ)

現在
使えるか

ジョブ
の総数

実行して
いるジョブ
の数

待たされて
いるジョブ
の数

ノードの利
用可能数