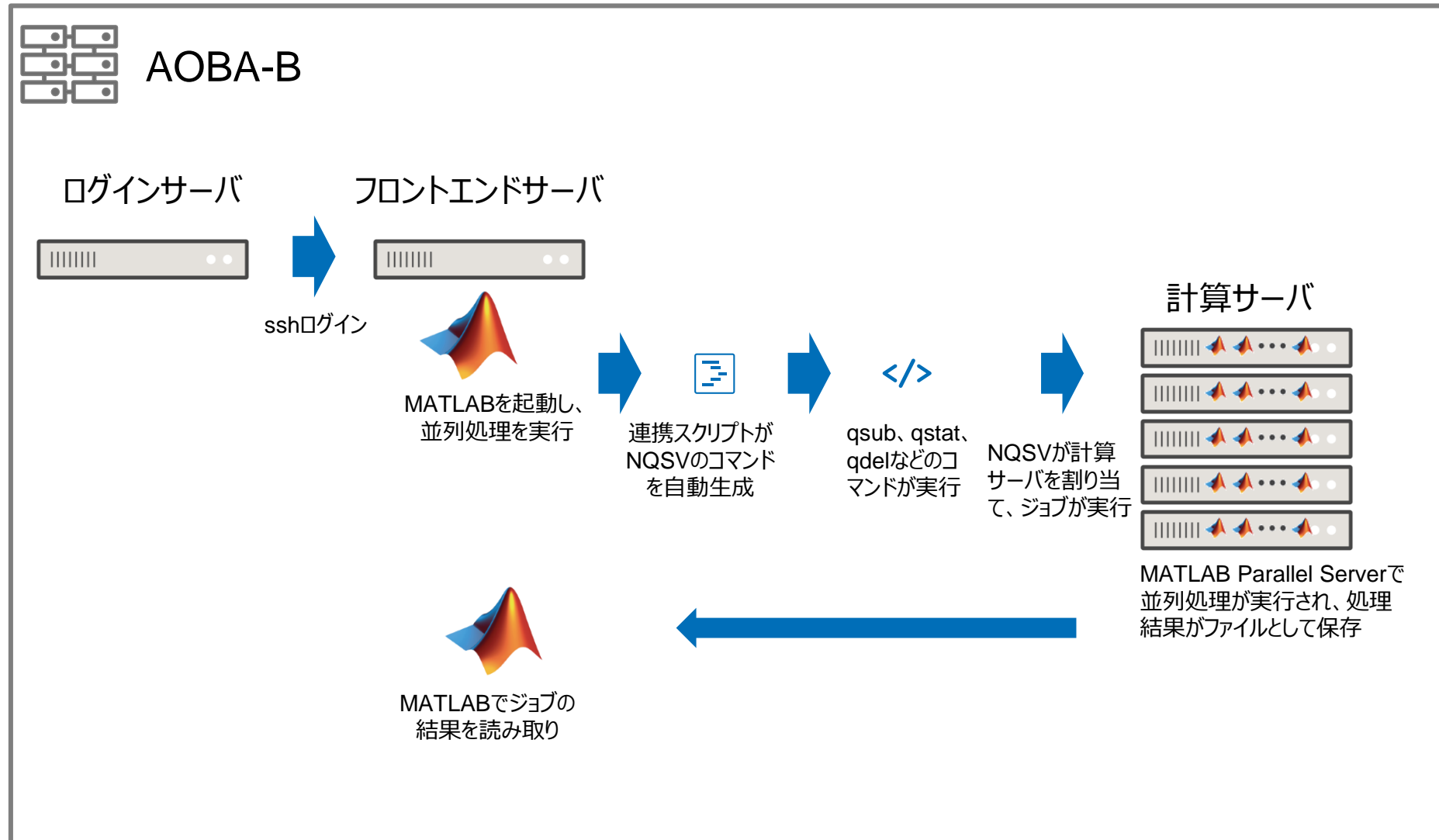


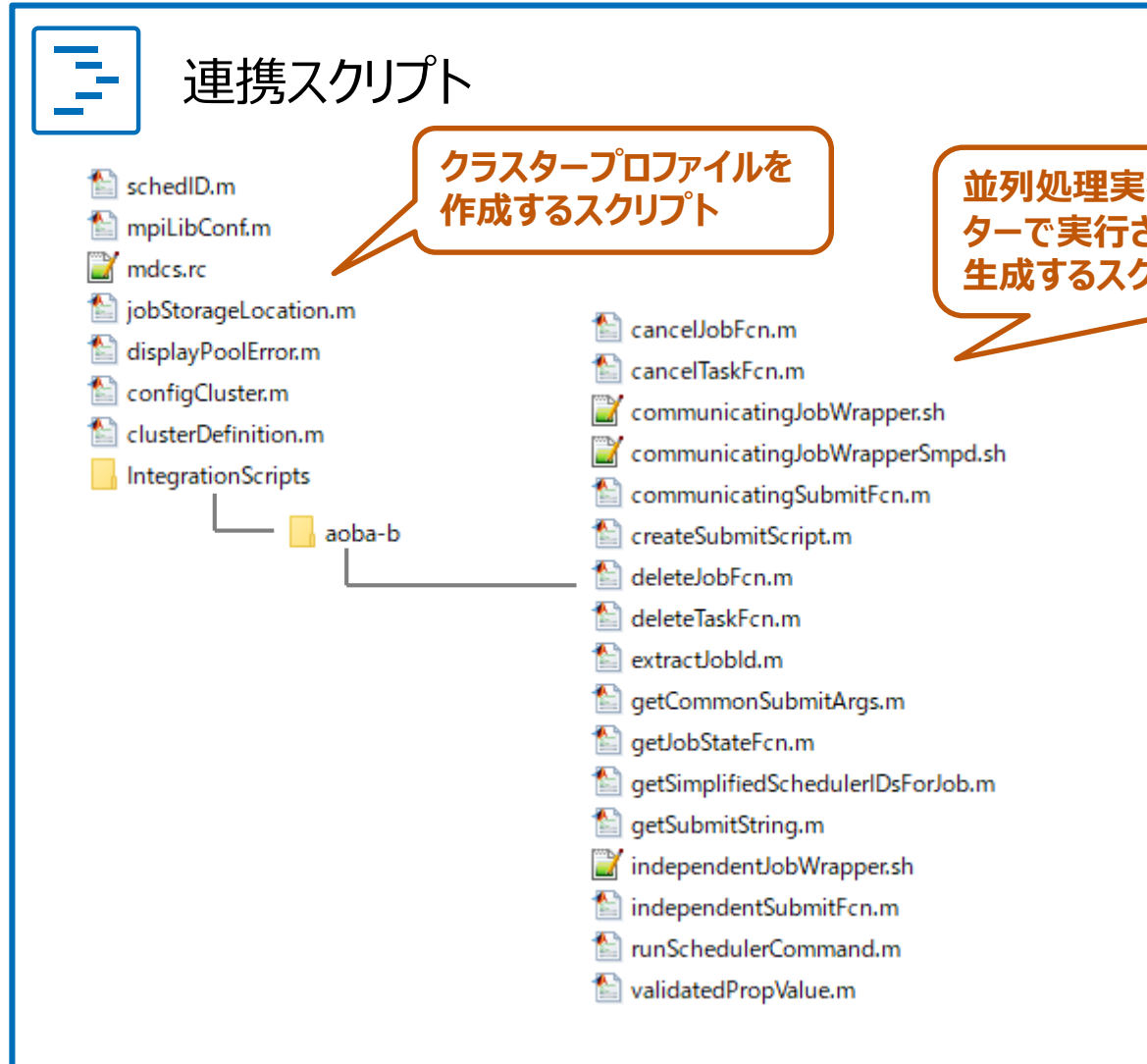
東北大学 サイバーサイエンスセンター御中 AOBA-BでのMATLAB Parallel Serverの使い方

MathWorks Japan
アプリケーションエンジニアリング部
プロダクトスペシャリストチーム

AOBA-BでのMATLABの並列処理のワークフロー



連携スクリプトの中身



用語	意味
クラスタープロファイル	MATLABの並列処理をどの環境で実行させるかの設定
連携スクリプト	MATLABの並列処理のコマンド実行時に、クラスタープロファイルの設定や要求するタスクに応じて、NQSVのqsubコマンド、qstat、qdelコマンドなどを動的に生成するスクリプト

連携スクリプトの使い方

mdcs.rcファイルの変更

(1) テキストエディタでmdcs.rcを編集し、必要に応じて値を変更します。

項目	デフォルト値	備考
Type	local	local: MATLABのマシンから直接NQSのコマンドを実行できる場合
NumWorkers	128	使用するMATLAB Parallel Serverの最大ワーカー数
ClusterMatlabRoot	R2020b:/mnt/stfs/ap/MATLAB, R2020a:/mnt/stfs/ap/MATLAB.R2020a	MATLAB Parallel Serverがインストールされたディレクトリのパス。 複数バージョンがある場合は、 「<バージョン名>:インストールディレクトリ」をカンマでつなげます。 例：R2020b:/mnt/stfs/ap/MATLAB, R2020a:/mnt/stfs/ap/MATLAB.R2020a
ClusterHost	無し	AOBA-Bの場合、使用しません。
LocalJobStorageLocation	無し	MATLABクライアントのジョブを格納するパスです。デフォルトの空のままでもOKです。
RemoteJobStorageLocation	/uhome/	MATLABから投げられたジョブやデータのファイルを格納するクラスター側のパスです。 ユーザーuseraがクライアントホストfront1からジョブを実行した場合、 RemoteJobStorageLocation = /uhome/と設定すると、 /uhome/usera/MdcsDataLocation/aoba-b/front1/R2020b/local にジョブファイルなどが作成されます。
JobStorageLocationOnPC	無し	AOBA-Bの場合、使用しません。

連携スクリプトの使い方

MATLABでの設定

(2) 連携スクリプトをMATLABのカレントフォルダまたはパスが通っているところに置きます。



連携スクリプト

連携スクリプトをカレントフォルダに置くか、
またはMATLABから `addpath('フォルダ名')`
でパスに追加します。

(3) MATLAB上で`configCluster.m`を実行します。

```
コマンドウィンドウ
>> configCluster
fx >> |
```

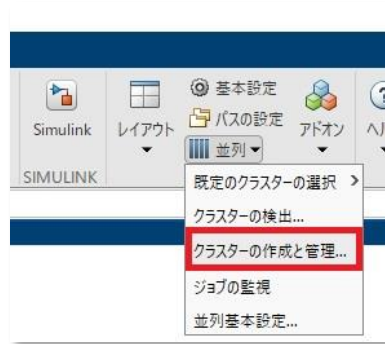
(4) AOBA-B用のクラスタープロファイルが作成されます。

連携スクリプトの使い方

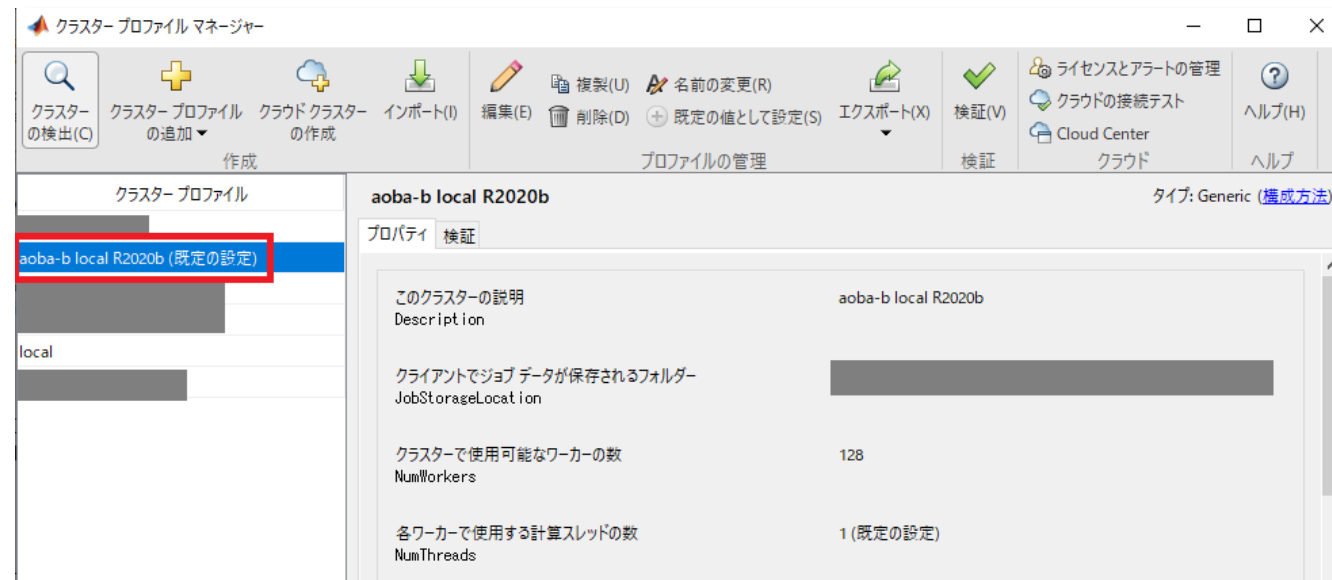
MATLABでの設定

(5) クラスタプロフィールを確認します。

「並列」→「クラスタプロフィールの作成と管理」をクリックします。



AOBA-B用のクラスタプロフィールが既定として設定されていることを確認します。



連携スクリプトの使い方

MATLABでの設定

(6) 必要に応じてクラスタープロファイルを変更します。

クライアントおよびクラスター ノードからジョブ保存場所にアクセス可能
HasSharedFilesystem false
既定値は true です

スケジューラ プラグイン
スケジューラ プラグインのスクリプトを格納するフォルダー
PluginScriptsLocation ired\R2020b\aoaba-b\matlab\IntegrationScripts\aoaba-b 参照...

プラグイン スクリプト用の追加プロパティ
AdditionalProperties

名前	値	
NodeCount	1	
ProcsPerNode	64	数値 ▼
ProjectCode		string ▼
QueueName	lx	string ▼
RemoteJobStorageLocat...	/tmp/ksaito/MdcsDataL...	string ▼
RequestName		string ▼

追加 削除

完了 キャンセル

プラグインスクリプト用の追加プロパティを必要に応じて変更します。

- メールアドレス (EmailAddress)
- 最大経過時間 (MaxElapseTime)
- ノード数 (NodeCount)
- ノードあたりのプロセス数 (ProcsPerNode)
- プロジェクトコード (ProjectCode)
- キュー名 (QueueName)
- リクエスト名 (RequestName)
- リクエストのリランの有無 (RequestRerun)
- ユーザー名 (UserNameOnCluster)

詳細については次以降のページをご覧ください。

連携スクリプトの使い方

MATLABでの設定

プロパティ名	説明	対応するNQSVのオプション	デフォルト値
JobStorageLocation	クライアントでジョブデータを保存するディレクトリ	—	/uhome/ユーザー名 /MdcsDataLocation/aoba-b/ホスト名 /R2020b/local
NumWorkers	クラスターで使用可能なワーカーの数	—	128
NumThreads	各ワーカーで使用する計算スレッドの数	—	1 (推奨は1ワーカーあたり1スレッドです)
ClusterMatlabRoot	MATLAB Parallel Serverがインストールされているディレクトリ	—	R2020bの場合 : /mnt/stfs/ap/MATLAB R2020aの場合 : /mnt/stfs/ap/MATLAB.R2020a
LicenceNumber	MATLAB Parallel Serverオンラインライセンスを使用する場合のライセンス番号	—	無し (オンラインライセンスを使用しないため)
RequiresOnlineLicensing	MATLAB Parallel Serverのオンラインライセンスを使用するかどうか	—	false (オンラインライセンスを使用しないため)
OperatingSystem	クラスター計算ノードのOS	—	unix
HasSharedFilesystem	MATLABクライアントのマシンとクラスターの計算ノードとでファイル共有サーバがあるかどうか	—	true
PluginScriptsLocation	連携スクリプトを格納しているディレクトリ	—	configCluster.mがあるIntegratonScriptsのディレクトリ

連携スクリプトの使い方

MATLABでの設定

プロパティ名	説明	対応するNQSVMのオプション	デフォルト値
AdditionalProperties	クラスターに渡す追加のプロパティ	—	—
AdditionalSubmitArgs	追加で渡すqsubの引数	—	
DebugMessagesTurnedOn	デバッグ用フラグ	—	false
EmailAddress	ジョブ投入時、終了時にメール通知をする場合のメール受信先	-M <メールアドレス> -m b -m e	無し
IdentityFile	SSHをパスワードではなく鍵ファイルでログインする場合はSSH鍵ファイルのパスを指定し、UseIdentityFileをtrueにします。	—	無し
MaxElapseTime	最大経過時間	-l elapstim_req	24:00:00
NodeCount	ノード数	-b	1
ProcsPerNode	1ノードあたりのプロセス数	(-bのノード数算出に使用)	128
QueueName	キュー名	-q	lx
RequestName	リクエスト名	-N	無し
RequestRerun	リクエストのリランの有無	-r y / -r n	false
UseIdentityFile	SSHをパスワードではなく鍵ファイルでログインする場合はtrueにし、IdentityFileでファイルパスを指定します。	—	false

連携スクリプトの使い方

MATLABでの設定

プロパティ名	説明	対応するNQSVMのオプション	デフォルト値
AutoAttachFiles	MATLAB Parallel Serverのクラスターに依存関係のあるファイルを自動的に送信するかの設定	—	true
AttachedFiles	MATLAB Parallel Serverのクラスターに送信するファイルまたはディレクトリ	—	無し
AdditionalPaths	ワーカーの検索パスに追加するフォルダ	—	無し
NumWorkerRange	ジョブを実行するワーカー数の範囲	—	[1 inf]
CaptureDiary	コマンドウィンドウへ出力を返すかどうか	—	false
EnvironmentVariables	MATLABクライアントからMATLAB Parallel Serverのワーカーにコピーする環境変数	—	無し

連携スクリプトの使い方

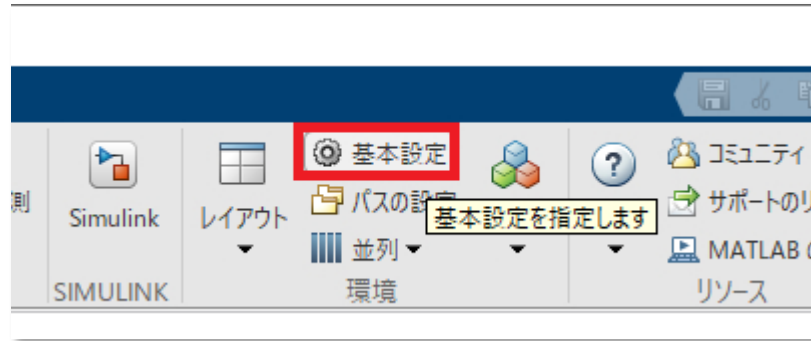
MATLABでの設定

MATLAB Parallel ServerをAOBA-Bで実行する際の、その他のNQSVオプションは以下のとおりです。

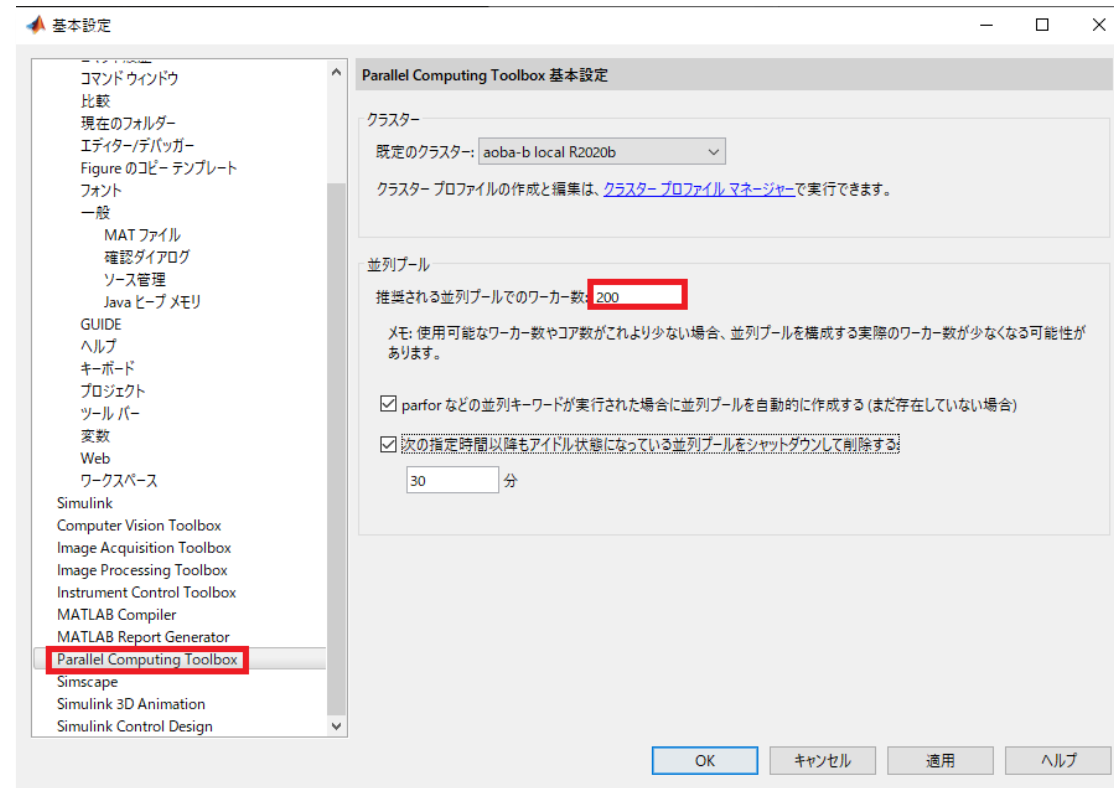
NQSVのオプション	デフォルト値
-T intmpi	複数ノードをまたぐ処理の場合、MPIの実行環境をIntel MPIに指定しています。

連携スクリプトの使い方 MATLABでの設定

(7) 並列処理数の設定変更を行います。



MATLABの「基本設定」をクリックします。

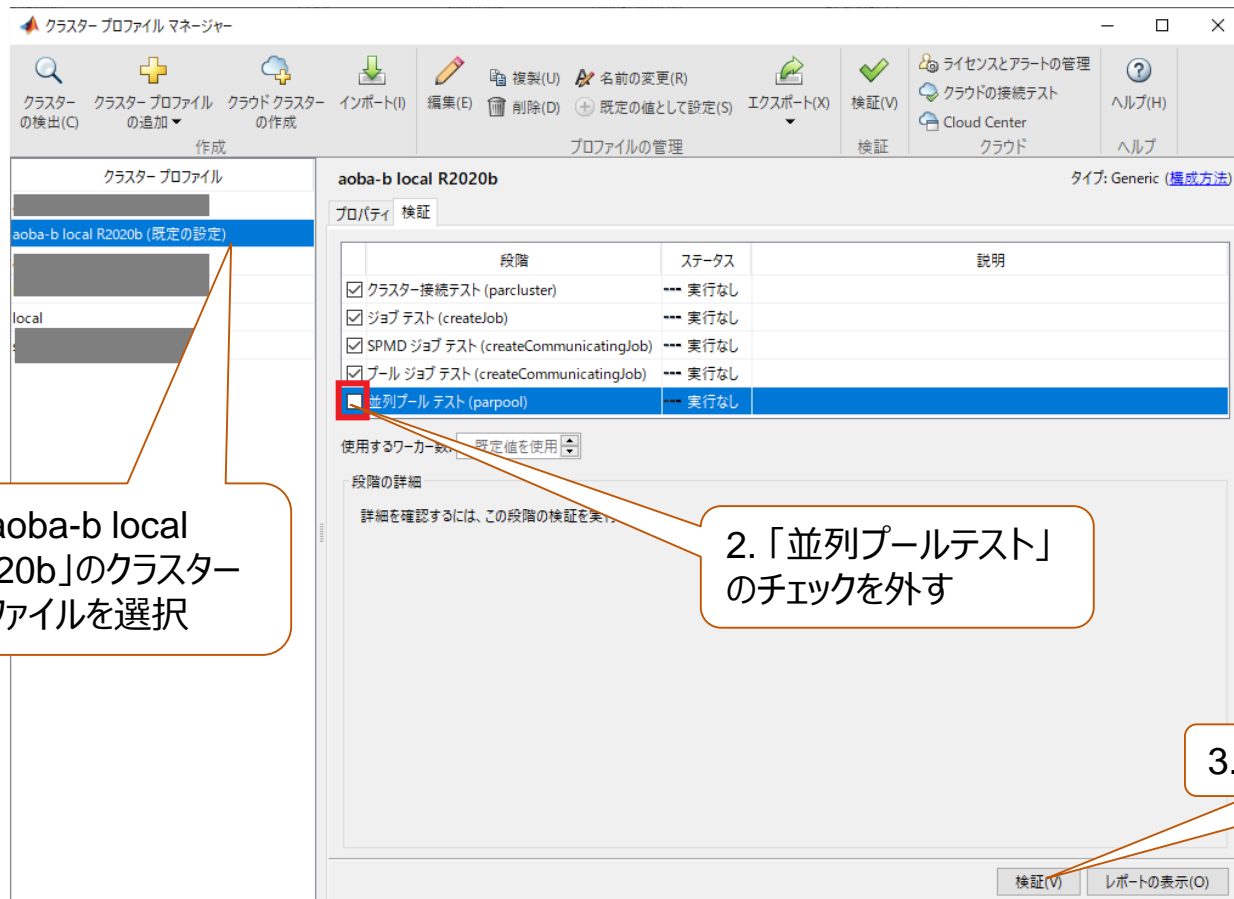


「Parallel Computing Toolbox」をクリックし、推奨される並列プールでのワーカー数を、デフォルトの12から、200などの値に変更します。(複数ノードで並列処理できるようにするため)

連携スクリプトの使い方

MATLABからのジョブ投入

(8) クラスタプロフィールの検証を行います。



1. 「aoba-b local R2020b」のクラスタプロフィールを選択

2. 「並列プールテスト」のチェックを外す

3. 「検証」をクリック

「プールジョブテスト」までの4つの検証をパスすれば問題ありません。

連携スクリプトの使い方

MATLABからのジョブ投入

(9) 最後にbatchでのオフロード処理をテストします。

parallelServerSample.mというファイルを作り、以下のコードを記載します。

parallelServerSample.m

```
n = 200;  
A = 500;  
a = zeros(n);  
parfor i = 1:n  
    a(i) = max(abs(eig(rand(A)))));  
end
```

MATLABのコマンドウィンドウから、batchコマンドを使用してジョブを実行します。

```
job1 = batch('parallelServerSample', 'Pool', 3, 'AutoAddClientPath', false);
```

Poolオプションに使用するワーカー数から1を引いた値を入れます。

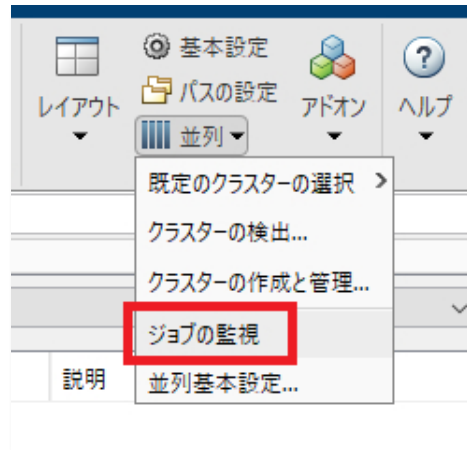
参考 : [バッチ並列ジョブの実行](#)

連携スクリプトの使い方

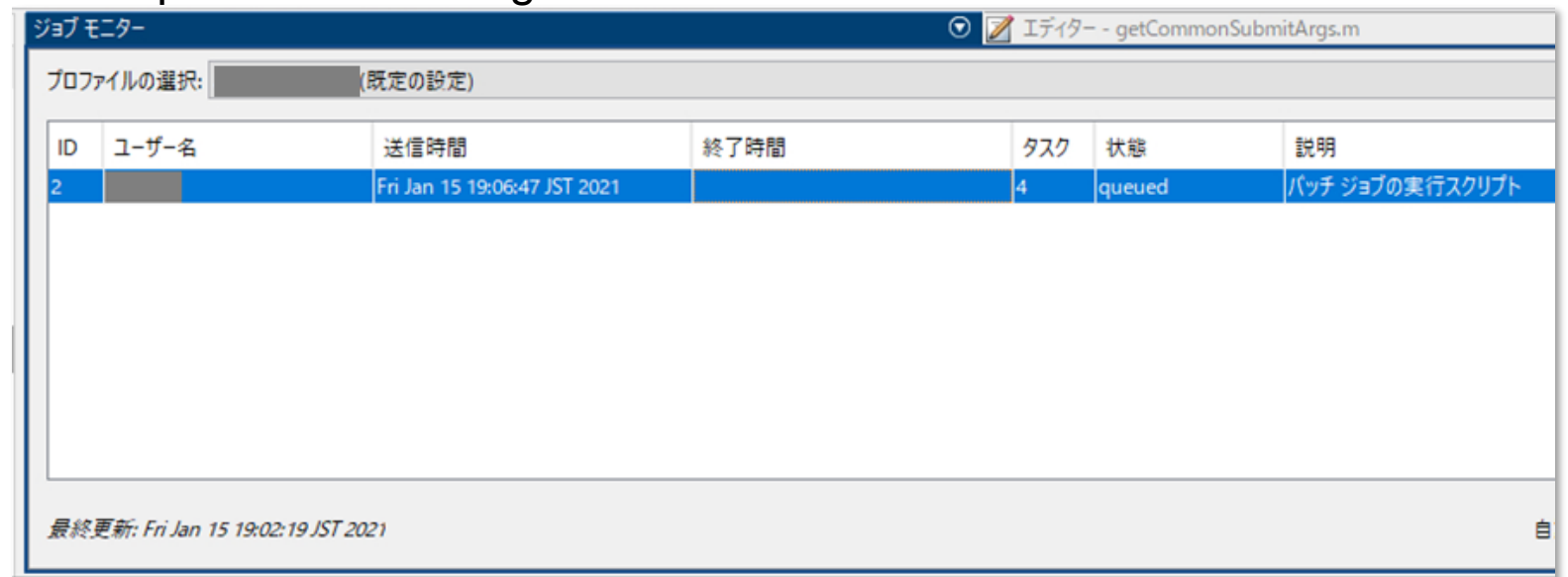
MATLABからのジョブ投入

(9) 最後にbatchでのオフロード処理をテストします。(続き)

ジョブ投入後、MATLABの「並列」メニューから「ジョブの監視」をクリックします。



クラスターサーバに投入したジョブの一覧が表示されます。処理が進むに連れ「状態」欄が「queued」、「running」、「finished」と遷移します。



連携スクリプトの使い方

MATLABからのジョブ投入

(9) 最後にbatchでのオフロード処理をテストします。(続き)

ステータスがfinishedになったら、ジョブモニターで右クリック「変数の読み込み」をクリックすることでクラスターサーバに投げた処理結果を取得できます。
または、以下のコマンドで処理結果を取得することもできます。

```
wait(job1);  
fetchOutputs(job1);
```

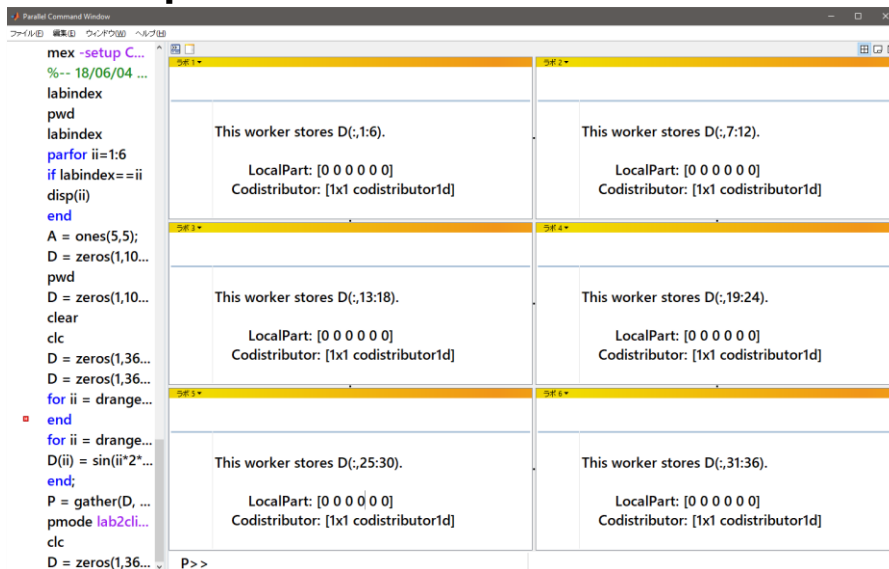

参考

並列処理

1. インタラクティブジョブ

- parfor: 汎用的な並列処理
- spmd: データを分割して同じ処理を実行 (single process multiple data)
- pmode: 対話的GUIを使った並列処理
- parsim: 並列シミュレーション(R2017a以降)

pmodeによる対話的な並列処理



```

mex -setup C:\...
%-- 18/06/04 ...
labindex
pwd
labindex
parfor ii=1:6
if labindex==ii
disp(ii)
end
A = ones(5,5);
D = zeros(1,10...);
pwd
D = zeros(1,10...);
clear
clc
D = zeros(1,36...);
D = zeros(1,36...);
for ii = drange...
end
for ii = drange...
D(ii) = sin(ii*2*...);
end;
P = gather(D, ...);
pmode lab2cli...;
clc
D = zeros(1,36...);
P >>
  
```

parforの例

```

n = 200;
A = 500;
a = zeros(n);
parfor i = 1:n
    a(i) = max(abs(eig(rand(A)))));
end
  
```

spmdの例 (円周率の計算)

```

spmd
a = (labindex - 1)/numlabs;
b = labindex/numlabs;
fprintf('Subinterval: [%-4g, %-4g]¥n', a, b);
end

spmd
myIntegral = integral(@pctdemo_aux_quadpi, a, b);
fprintf('Subinterval: [%-4g, %-4g] Integral: %4g¥n', ...
a, b, myIntegral);
end

% 結果の合計
spmd
piApprox = gplus(myIntegral);
end
  
```

並列処理

2. オフロードジョブ

- バッチ:
 クラスターでMATLABスクリプトを実行
- 独立ジョブ:
 クラスターの1ワーカーで実行するジョブ
- 通信ジョブ:
 クラスターの複数ワーカーで実行するジョブ

バッチの例

```
% クラスターの作成
c = parcluster();
% バッチ処理の実行
j = batch(c,@rand,1,{10,10}, 'CaptureDiary', ...
    true, 'CurrentFolder', '.');
% ジョブの終了待ち
wait(j)
% ジョブ結果の回収
out = fetchOutputs(j);
% ジョブの消去
delete(j)
```

通信ジョブの例

```
% クラスターの作成
c = parcluster;
% ジョブの作成
j = createCommunicatingJob(c,'Type', 'pool');
% タスクの作成
createTask(j, @myFunction, 1, {100});
% ジョブの投入
submit(j);
% ジョブの終了待ち
wait(j)
% ジョブ結果の回収
out = fetchOutputs(j)
% ジョブの消去
delete(j)

%% カスタム関数
function result = myFunction(N)
    result = 0;
    parfor ii=1:N
        result = result + max(eig(rand(ii)));
    end
end
```

独立ジョブの例

```
% クラスターの作成
c = parcluster
% ジョブの作成
j = createJob(c);
% タスクの作成
for ii = 1:10
    createTask(j,@rand,1,{10});
end
% ジョブの投入
submit(j);
% ジョブの終了待ち
wait(j);
% ジョブ結果の回収
out = fetchOutputs(j);
% ジョブの消去
delete(j)
```

用語集

用語	意味
ワーカー	MATLAB Parallel Serverの並列処理を実行するMATLABプロセスのこと。
インタラクティブジョブ	MATLABの並列処理のうち、MATLABクライアントとワーカーとの間に通信を張って処理を行う並列処理のこと。処理中はMATLABクライアントを起動したままにする必要があります。
オフロードジョブ	MATLABの並列処理のうち、MATLABクライアントとワーカーとの間で通信を張らずに処理を行う並列処理のこと。ジョブ投入後はMATLABクライアントを停止させてもワーカー内で処理が継続されます。

参考情報

- MATLAB Parallel Serverの製品について
<https://jp.mathworks.com/products/matlab-parallel-server.html>
- MATLABの並列処理について
<https://jp.mathworks.com/help/parallel-computing/getting-started-with-parallel-computing-toolbox.html>
- MATLABのバッチ処理のサンプル
<https://jp.mathworks.com/help/parallel-computing/batch-processing.html>

問い合わせ窓口

- MathWorksサポート窓口
https://jp.mathworks.com/support/contact_us.html
- コミュニティQ&Aサイト
<https://jp.mathworks.com/matlabcentral/answers/>



© 2021 The MathWorks, Inc. MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See www.mathworks.com/trademarks for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.