

Parallel Computing with MATLAB on the RCC HPC Cluster Q&A Summary

Q: Where can I download the support package?

A: The RCC MATLAB support package and additional information about remote submission can be found in the *Getting Started with Parallel Computing using MATLAB on the RCC HPC Cluster* document which can be found on this [RWTH ITC page](#).

Q: Are RCC users required to install the support package when submitting jobs directly from the cluster?

A: The support package is already installed on the cluster for all versions of MATLAB; however, you will need to call `configCluster`.

Q: When we submit the job from MATLAB installed on the cluster, do we submit the job in the same way?

A: Yes. See sections labeled "MATLAB client on the cluster" in the document *Getting Started with Parallel Computing using MATLAB on the RCC HPC Cluster* referenced above.

Q: I accidentally clicked "use an identity" when configuring my local client for remote submission. Now I can't enter a password anymore, as it's always directing me to pick an ID-file.

A: Rerun `configCluster`.

Q: Where is the cluster profile saved that we created with the configCluster.m script?

A: It is stored in the MATLAB preferences directory.

Q: What does the command `c.AdditionalProperties.WallTime = '05:00:00'` mean?

A: This sets the job's wall time to 5 hours. The wall time places a limit on the length of time for the job. If the job takes longer than five hours, it will be cancelled.

Q: Do I need to set WallTime?

A: Yes, you must set the wall time property of your jobs. For example, to set the wall time to 15 minutes:

```
>> c.AdditionalProperties.WallTime = '00:15:00';
```

Q: How can I access different login nodes that are specific for GPUs like g1, g2 and g?

A: To clarify, the GPUs are connected to compute nodes, not login nodes. You request GPUs with the `GpusPerNode` property. For example:

```
>> c.AdditionalProperties.GpusPerNode = 2;
```

To disable it, set the value to 0.

```
>> c.AdditionalProperties.GpusPerNode = 0;
```

Q: Do I have to save my job variable when closing my MATLAB desktop client while a job is running?

A: No, you can access your jobs later. For example:

```
>> c = parcluster;
```

```
>> c.Jobs % will list all of your jobs
```

Or by using the [Job Monitor](#).

Q: If we quit MATLAB how can we get the diary as our cluster object variable c is lost?

A: Call `parcluster` and then query for the job's diary.

```
>> c = parcluster;  
>> c.Jobs           % will list all of your jobs  
>> c.Jobs(3).diary  % will recall the diary of job #3
```

Or use the Job Monitor.

Q: How do I get the MATLAB ID of my job?

A: `job.ID`

Or use the Job Monitor.

Q: How can I get the task scheduler (Slurm) ID?

A: `job.getTaskSchedulerIDs`

Q: I am using batch to run a function with multiple output arguments in parallel. Does the specified number of expected outputs N mean we get all the first N outputs, or can we specify to retrieve the N'th output argument of the function?

A: N specifies the [number of outputs expected from the submitted function](#).

You can skip over output arguments, as such:

```
[~, ~, output] = job.fetchOutputs{:};
```

Q: Is there a way to wait for a specific job if we submitted multiple jobs?

A: Yes, for the job you want to wait for, call the `wait` method and it will wait for the job to complete.

For example, the below code will wait for `job2` to finish before unblocking the command prompt:

```
>> job1 = c.batch(@pause,0,{120});  
>> job2 = c.batch(@pause,0,{120});  
>> job3 = c.batch(@pause,0,{120});  
>> job2.wait
```

Q: Can Simulink also be used for parallel computing?

A: Yes, Simulink can be used for parallel computing and all Simulink specific tools are installed on the RCC. Note that [parsim](#) and [batchsim](#) are parallel constructs specifically developed for Simulink.

Q: Does job.load always bring the workspace to my local machine (even when calling functions)?

A: To clarify, `job.load` cannot be called on functions, only scripts. For scripts, you can specify which variables to load in your local MATLAB. For example:

```
>> job.load('a','b')
```

will only load variables `a` and `b` into the workspace.