

Getting Started with Parallel Computing using MATLAB on the Sunya HPC Cluster

This document provides the steps to configure MATLAB to submit jobs to a cluster, retrieve results, and debug errors.

CONFIGURATION

After logging into the cluster, start MATLAB. Configure MATLAB to run parallel jobs on your cluster by calling `configCluster`. `configCluster` only needs to be called once per version of MATLAB.

```
>> configCluster
```

Jobs will now default to the cluster rather than submit to the local machine.

NOTE: If you would like to submit to the local machine then run the following command:

```
>> % Get a handle to the local resources
>> c = parcluster('local');
```

CONFIGURING JOBS

Prior to submitting the job, we can specify various parameters to pass to our jobs, such as queue, e-mail, walltime, etc.

```
>> % Get a handle to the cluster
>> c = parcluster;

>> % Specify an account to use for MATLAB jobs
>> c.AdditionalProperties.AccountName = 'account-name';

>> % Request to run on a GPU
>> c.AdditionalProperties.GpusPerNode = 1;

>> % Specify e-mail address to receive notifications about your job
>> c.AdditionalProperties.EmailAddress = 'user-id@rajagiritech.edu.in';

>> % Specify memory to use for MATLAB jobs, per core (MB)
>> c.AdditionalProperties.MemUsage = '4000';

>> % Specify to run on a particular node
>> c.AdditionalProperties.NodeList = 'sny02';

>> % Specify a queue to use for MATLAB jobs
>> c.AdditionalProperties.QueueName = 'queue-name';

>> % Specify the walltime (e.g. 5 hours)
>> c.AdditionalProperties.WallTime = '05:00:00';
```

Save changes after modifying AdditionalProperties for the above changes to persist between MATLAB sessions.

```
>> c.saveProfile
```

To see the values of the current configuration options, display AdditionalProperties.

```
>> % To view current properties
>> c.AdditionalProperties
```

Unset a value when no longer needed.

```
>> % Turn off email notifications
>> c.AdditionalProperties.EmailAddress = '';
>> c.saveProfile
```

INTERACTIVE JOBS

To run an interactive pool job on the cluster, continue to use parpool as you've done before.

```
>> % Get a handle to the cluster
>> c = parcluster;

>> % Open a pool of 64 workers on the cluster
>> p = c.parpool(64);
```

Rather than running local on the local machine, the pool can now run across multiple nodes on the cluster.

```
>> % Run a parfor over 1000 iterations
>> parfor idx = 1:1000
    a(idx) = ...
end
```

Once we're done with the pool, delete it.

```
>> % Delete the pool
>> p.delete
```

TO LEARN MORE

To learn more about the MATLAB Parallel Computing Toolbox, check out these resources:

- [Parallel Computing Coding Examples](#)
- [Parallel Computing Documentation](#)
- [Parallel Computing Overview](#)
- [Parallel Computing Tutorials](#)
- [Parallel Computing Videos](#)
- [Parallel Computing Webinars](#)