

MATLAB Distributed Computing Server at University of Maryland, College Park

Application Description

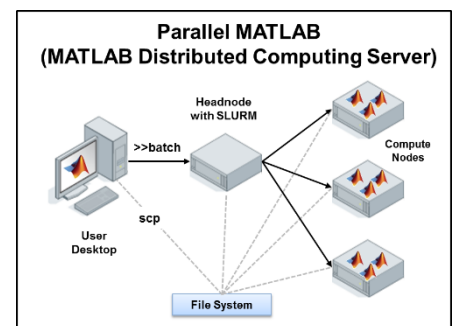
MATLAB Distributed Computing Server™ uses the syntax of [Parallel Computing Toolbox](#)™ and extends MATLAB® desktop workflows to cluster hardware by providing access to workers (MATLAB computational engines) that run on the computer cluster.

Two unique features of MATLAB Distributed Computing Server are:

- Access to MATLAB worker processes on the cluster from within the MATLAB desktop
- Built-in communication infrastructure between workers: built-in support for distributed arrays and message passing

Deployment of MATLAB Distributed Computing Server on Deepthought2

- MATLAB client available on users' local machine
- MATLAB job submission handled via [batch](#) within MATLAB client
 - Parallel Computing Toolbox on the MATLAB client provides all the tools necessary to create, monitor, and retrieve jobs
 - No need to learn scheduler commands
- Data and file transfer is facilitated through MATLAB infrastructure and `scp`
 - Use `scp` to transfer data from your desktop to the cluster filesystem
 - Use 'AdditionalPaths' in `batch` to specify cluster-specific directories to add to worker search path
 - Results can be saved to the cluster filesystem or retrieved through MATLAB job infrastructure
- Server license for 600 workers in total



User Access

- MATLAB Distributed Computing Server uses Parallel Computing Toolbox syntax; only the location of the workers changes
 - Develop and test with the desktop environment first. Consider differences in the desktop and cluster file systems
- Support:
 - Deepthought2: <http://www.glue.umd.edu/hpcc/dt2.html>
 - MATLAB at University of Maryland: <http://www.glue.umd.edu/hpcc/help/software/matlab.html>
 - Technical contact: hpcc-help@umd.edu



Best Practice for Developing Workflows

1. Develop your (serial or parallel) workflow on a desktop computer with MATLAB and Parallel Computing Toolbox.
 - Use MATLAB R2014a
 - Online tutorials are available: <http://www.mathworks.com/products/parallel-computing/tutorials.html>
2. Test your workflow with the local scheduler on your desktop computer
3. Submit your tested workflow to Deepthought2 resources
 - Users are encouraged to use only as many workers as are necessary
 - Benchmark workflows with different numbers of workers to determine an appropriate number of workers