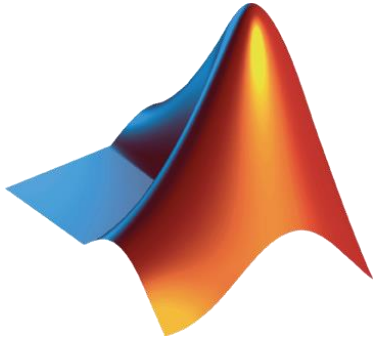
WORKSHOP: Parallel Computing with MATLAB (Part II)

Raymond Norris
Application Engineer
November 2023



Agenda

- Part I – Parallel Computing with MATLAB on the Desktop
  - Parallel Computing Toolbox
  - MATLAB Online
- Part II – Scaling MATLAB to JUWELS
  - MATLAB Parallel Server
  - Jupyter

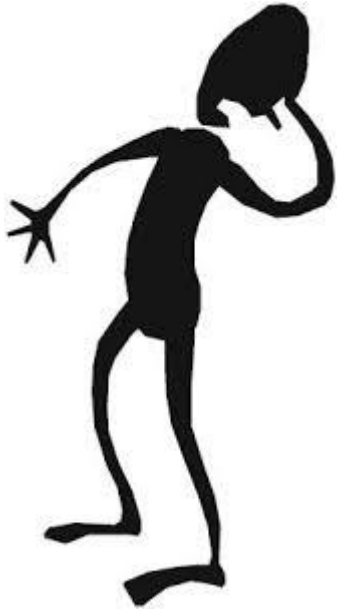
Agenda

- Part I – Parallel Computing with MATLAB on the Desktop
  - Parallel Computing Toolbox
  - MATLAB Online
- Part II – Scaling MATLAB to JUWELS
  - MATLAB Parallel Server
  - Jupyter

Overview

- How to configure MATLAB to submit multi-node jobs to the HPC cluster
- The job submission workflow
- Ways to tune job submissions to the HPC cluster
- How to optimize job submissions
- Troubleshooting job submission techniques
- Submitting interactive and batch jobs

A few notes about today's workshop…



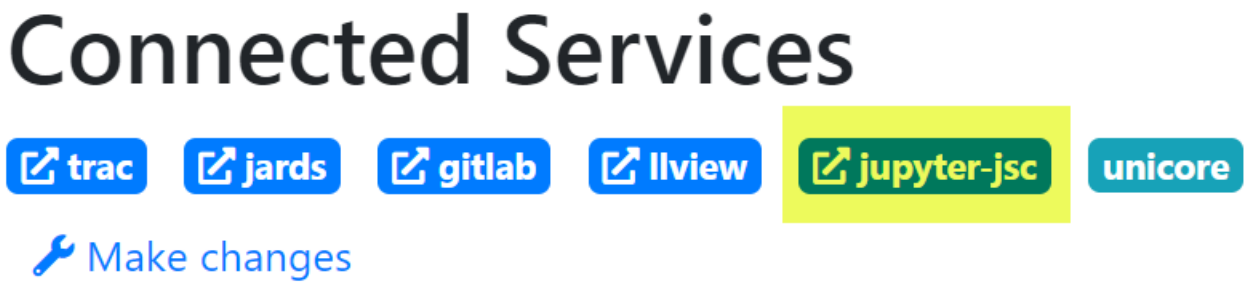
- The workflow and examples are about process, not performance
- Cluster documentation
  - https://www.fz-juelich.de/en/ias/jsc/systems/supercomputers
- MATLAB User Guide
  - https://www.fz-juelich.de/en/ias/jsc/services/user-support/software-tools/matlab
- Requirements
  - ❑ Account on JUWELS cluster

Accessing and running MATLAB on HPC compute nodes

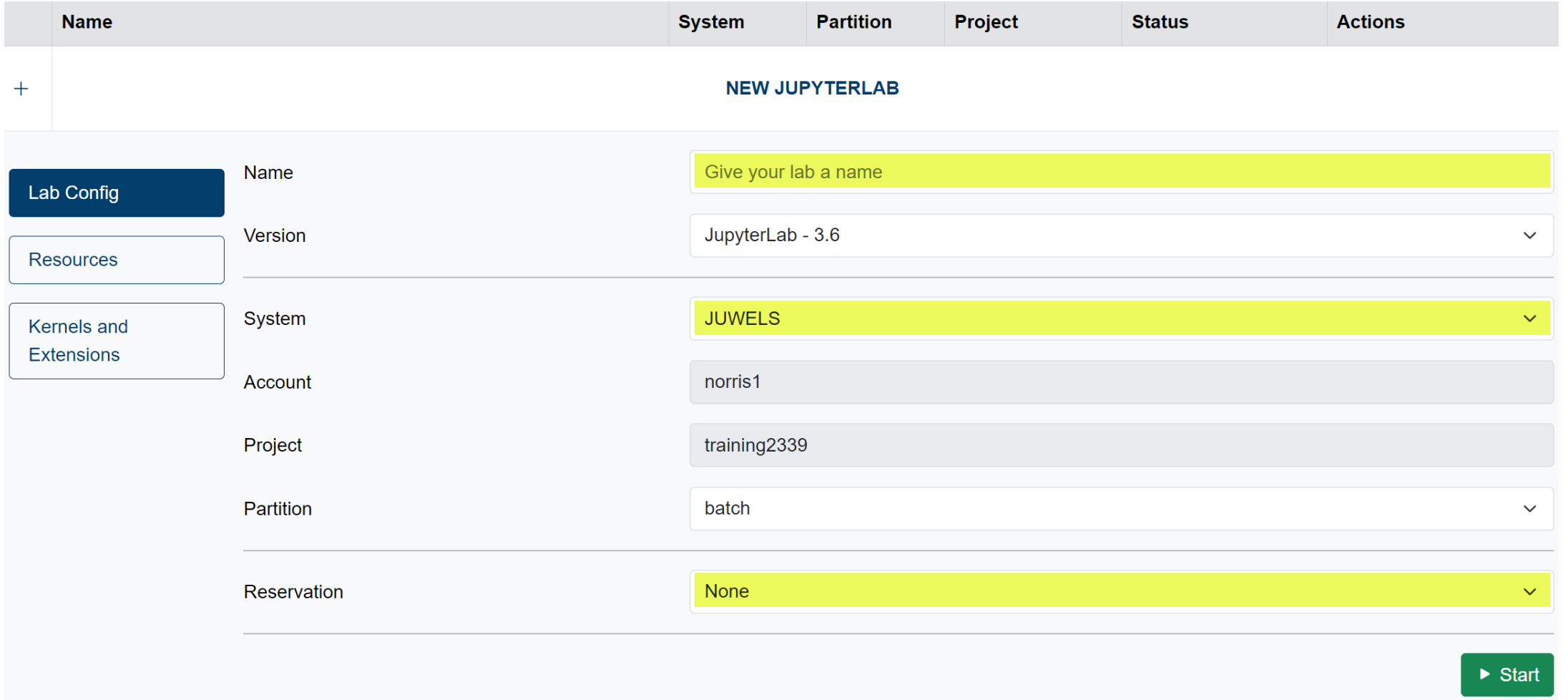
- Two options
  - SSH
    - Command line interface
    - Useful for either low-bandwidth or automation
  - Jupyter
    - Graphical interface

User Portal: **JuDoor**

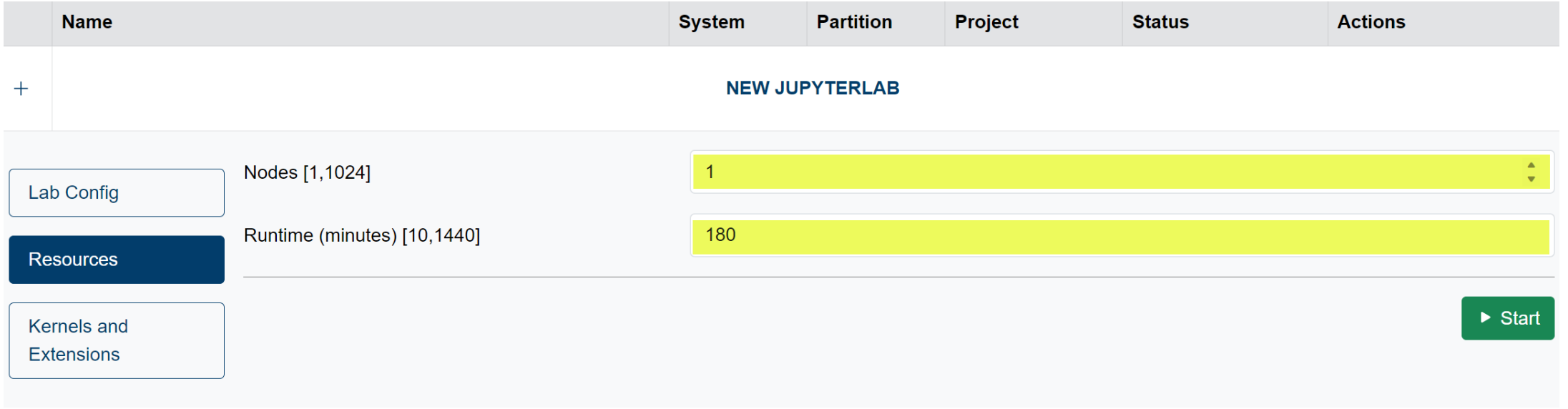
- https://judoor.fz-juelich.de/login



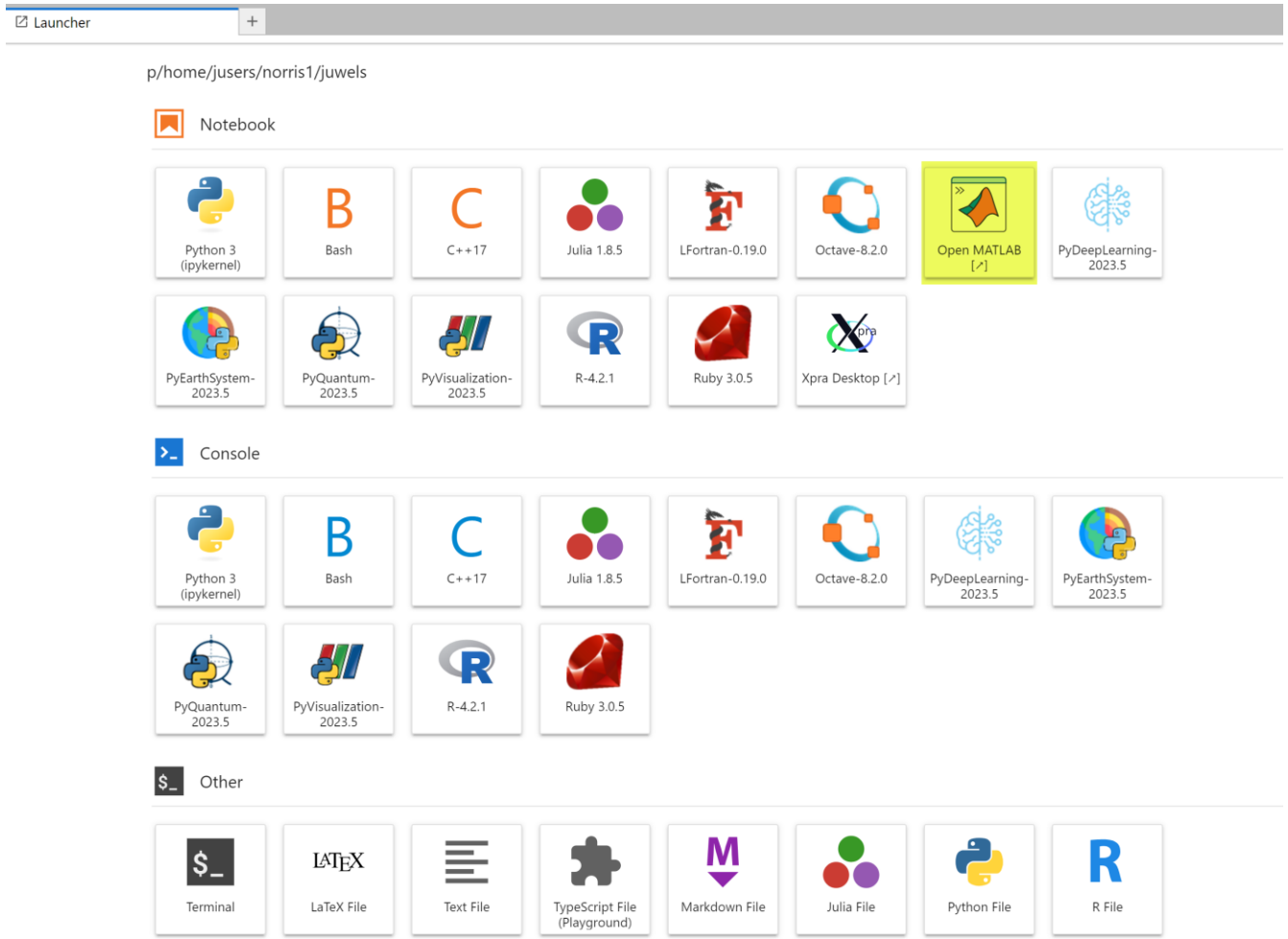
Launching a JupyterLab (1)



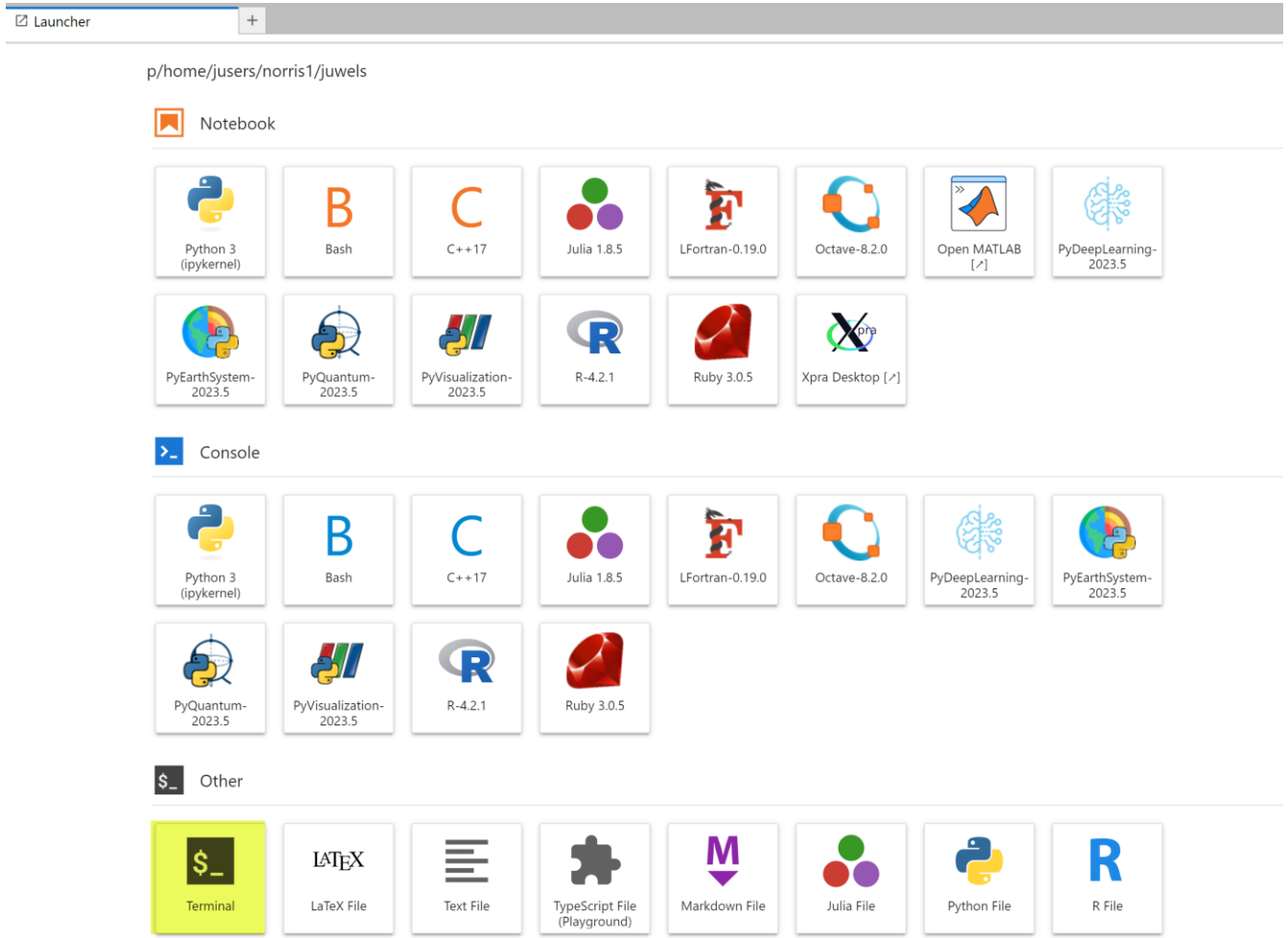
Launching a JupyterLab (2)



JupyterLab – MATLAB



JupyterLab – Shell terminal



Ways to run MATLAB

- Interactively
  - run serial code
  - with a parallel pool (`parpool`)
  - with batch jobs (`batch`)
- Noninteractive
  - write a Slurm job script (`sbatch`)

MATLAB job submitters

- `parpool`
  - Single session
  - Synchronous execution
  - Seamlessly runs `parfor`, `parfeval`, and `spmd`

https://www.mathworks.com/help/parallel-computing/parpool.html

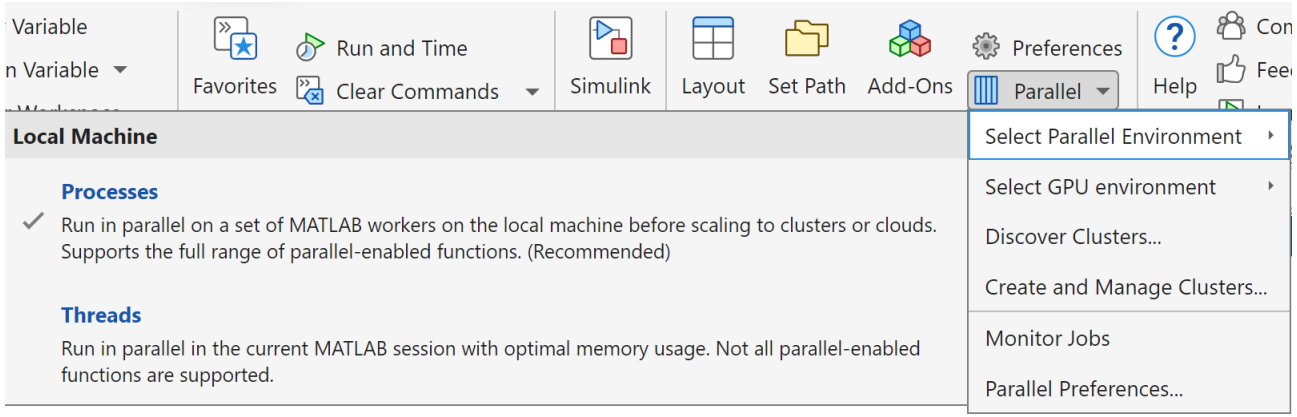
- `batch`
  - Multiple submissions
  - Non-blocking
  - Calls top-level function or script
  - Requires API to extract results

https://www.mathworks.com/help/parallel-computing/batch.html

Interactively: with a parallel pool

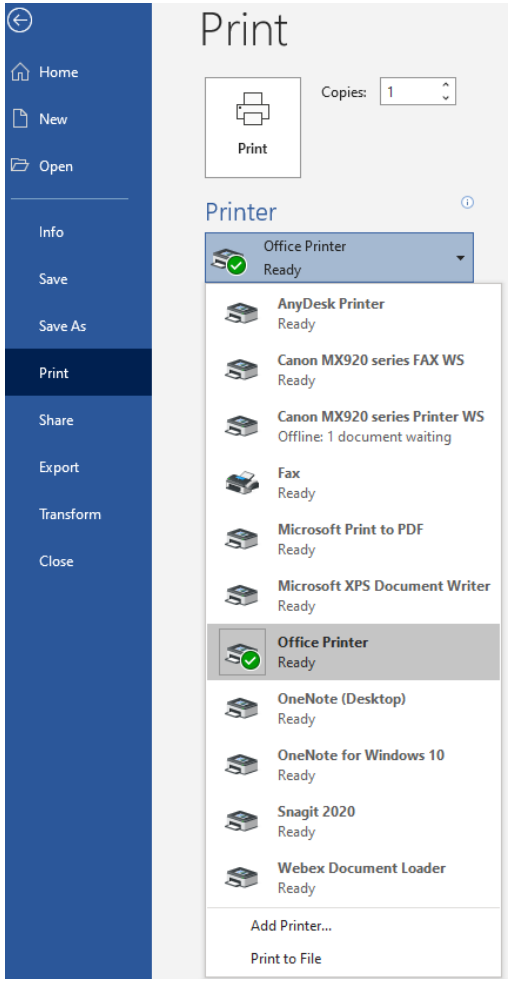
`parpool`

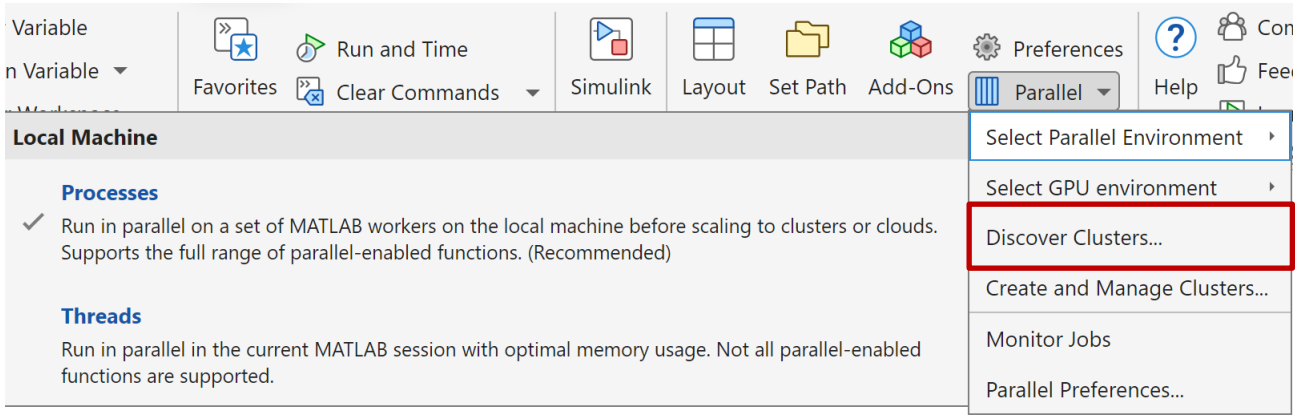
Parallel MATLAB – Single Node



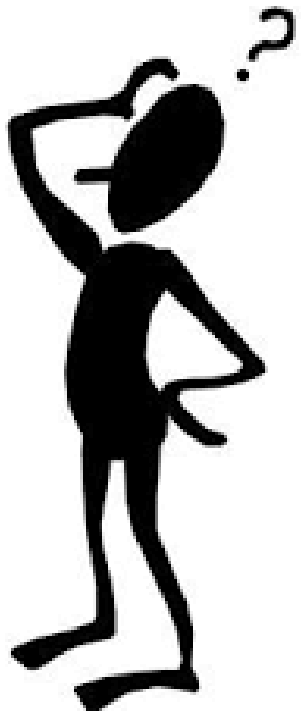

```
>> parpool("local",4);
Starting parallel pool (parpool) using the 'local' profile ...
Connected to parallel pool with 4 workers.
>>
>> tic, parfor idx = 1:80, pause(3), end, toc
Elapsed time is 61.088382 seconds.
```

Profiles

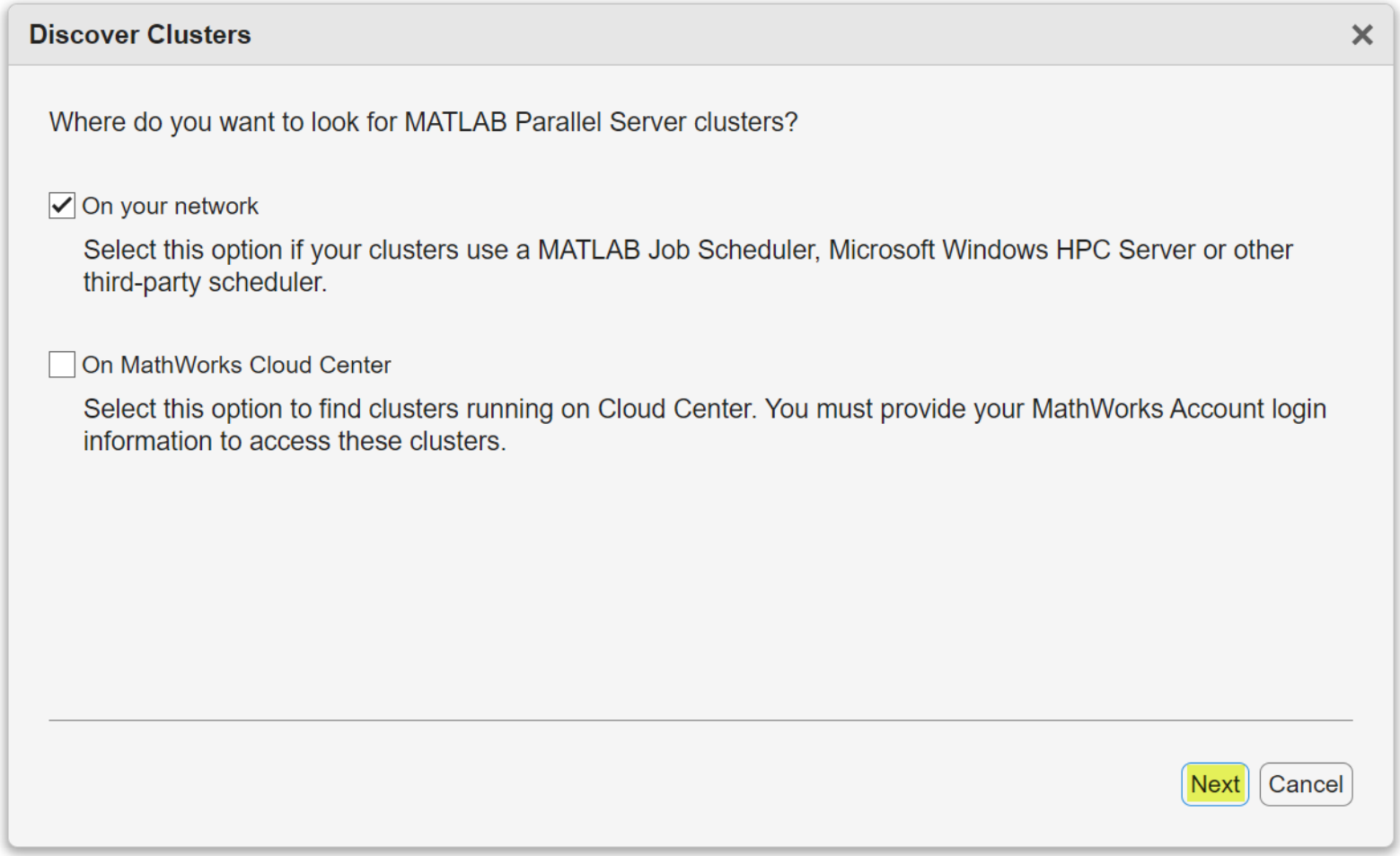




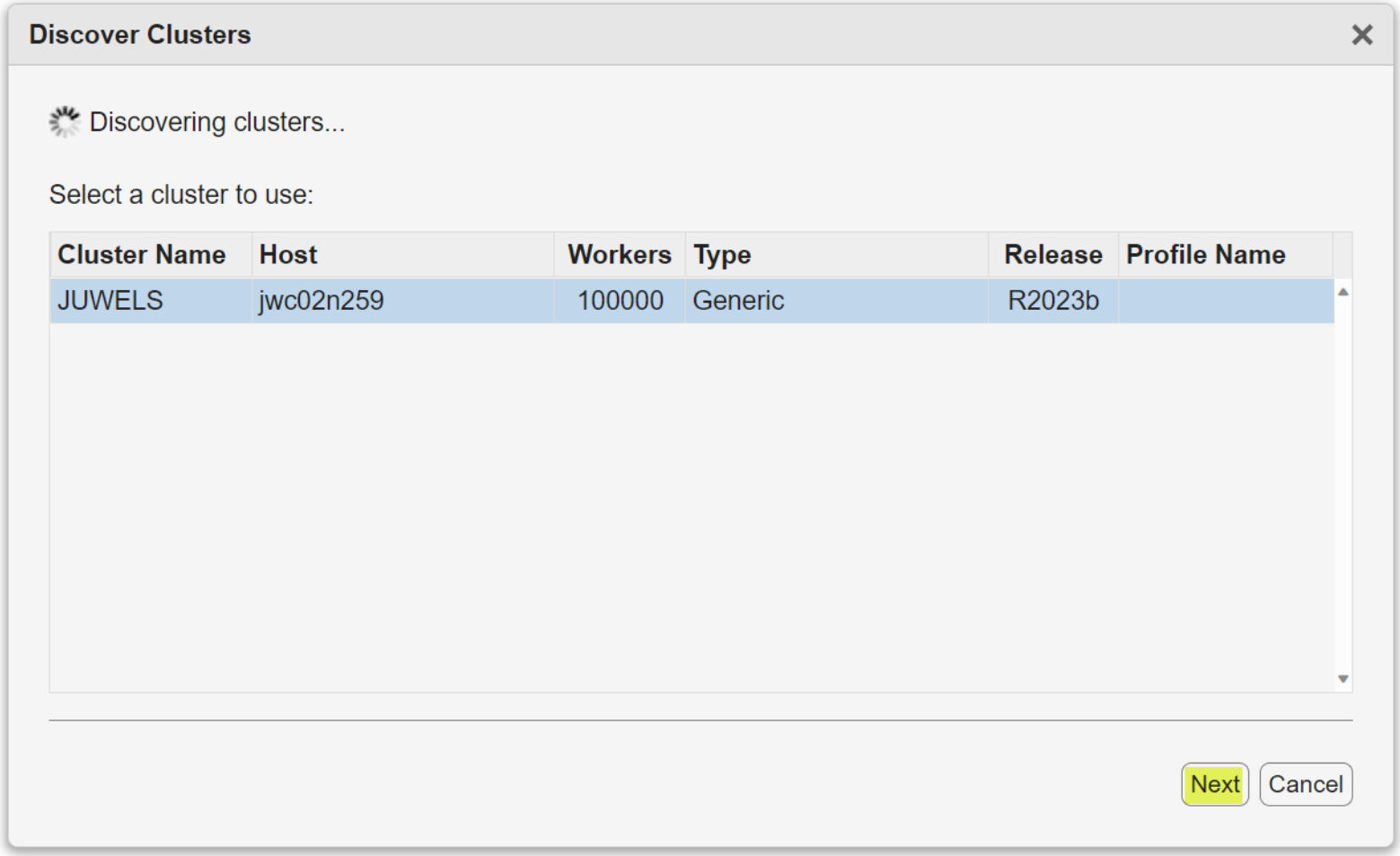
"How does MATLAB know about JUWELS?"



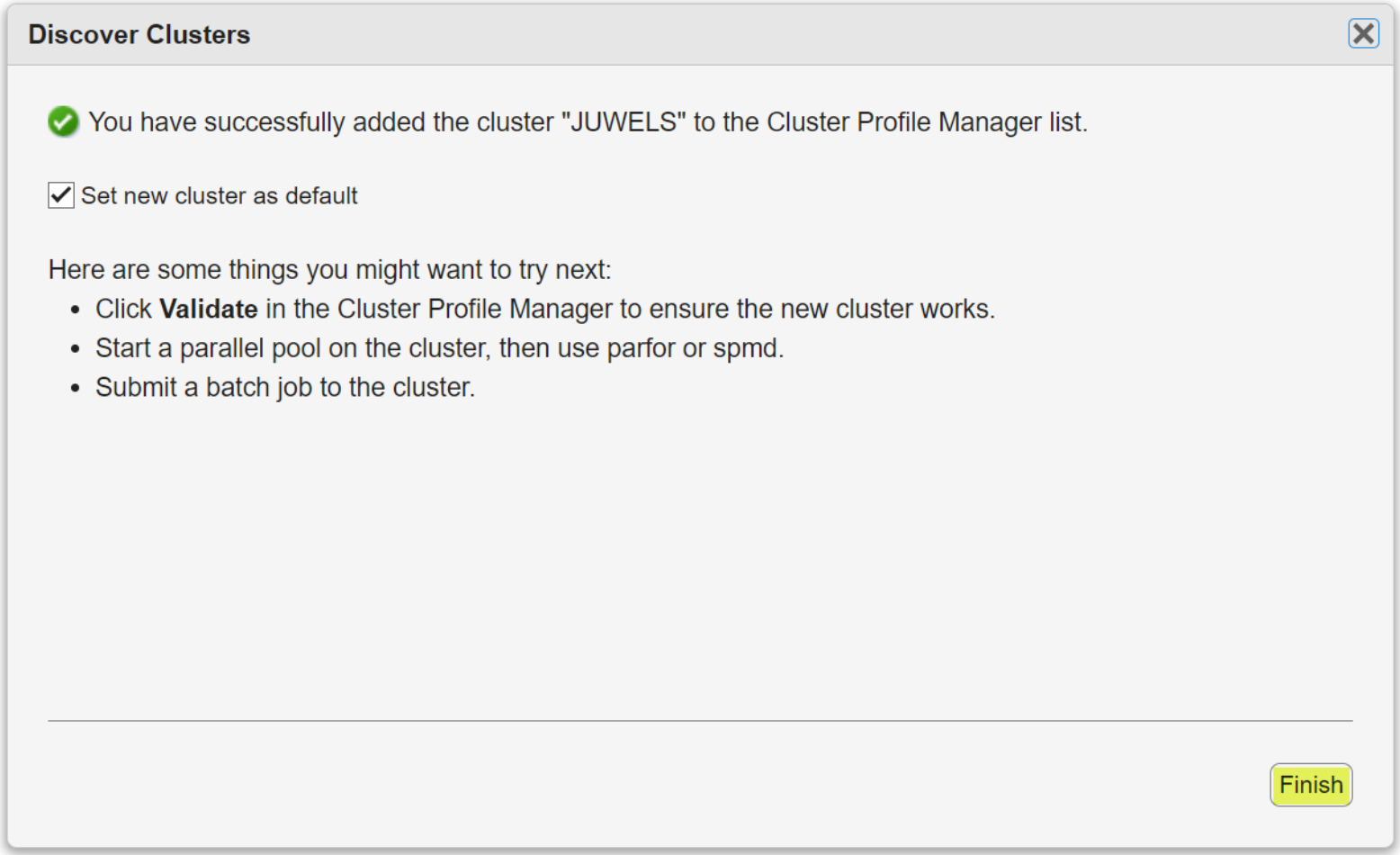
Discover Cluster (1)



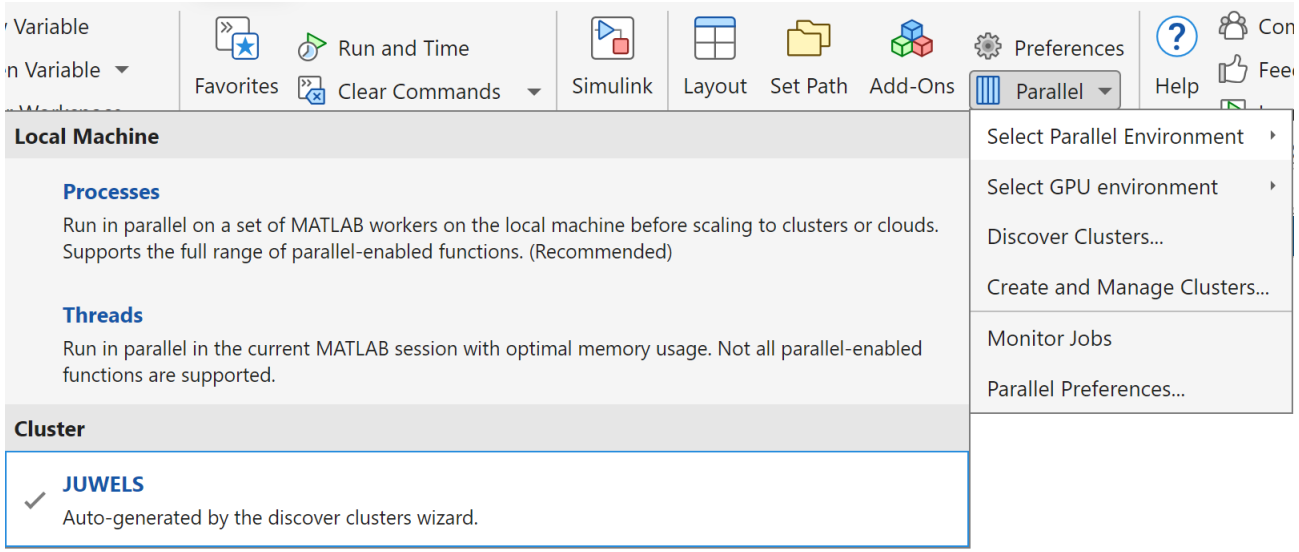
Discover Cluster (2)



Discover Cluster (3)



New JUWELS profile



JUWELS cluster

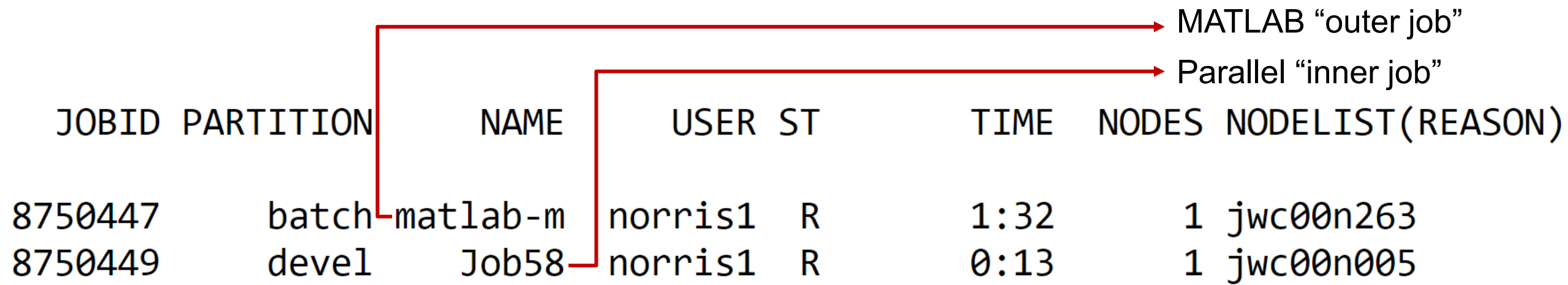
```
>> c = parcluster("JUWELS");

        Must set BudgetName before submitting jobs to JUWELS.  E.g.

        >> c = parcluster;
        >> c.AdditionalProperties.BudgetName = 'budget-name';
        >> c.saveProfile

        Must set WallTime before submitting jobs to JUWELS.  E.g.

        >> c = parcluster;
        >> % 5 hour, 30 minute walltime
        >> c.AdditionalProperties.WallTime = '05:30:00';
        >> c.saveProfile
```

Minimum required

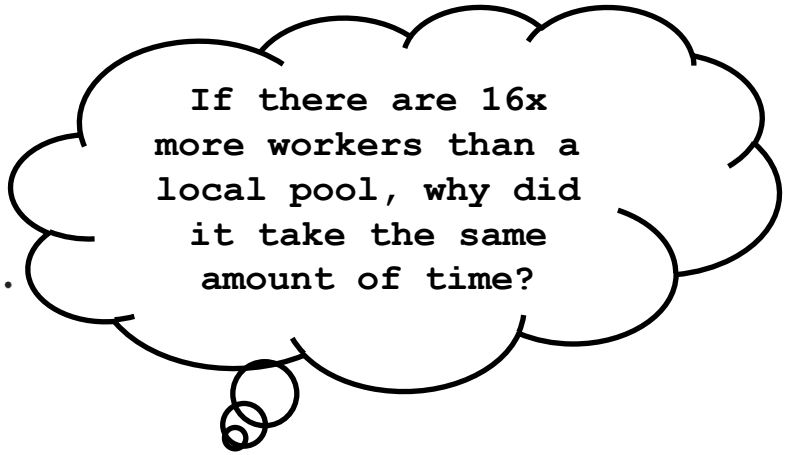
Parallel MATLAB – Multi-node

- To run a multi-node MATLAB job, MATLAB will generate and submit a new Slurm job
  - Executed during any “job launcher”
    - `parpool`, `batch`, `createJob`
  - True regardless if we’re running MATLAB desktop or via an Slurm job script

- Need to use the new JUWELS profile



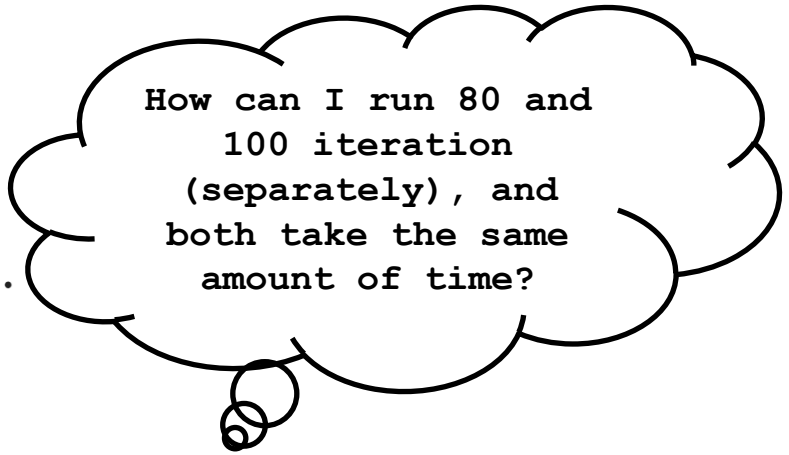
Where is the scaling (1)?




```
>> parpool("local",4);
Starting parallel pool (parpool) using the 'local' profile ...
Connected to parallel pool with 4 workers.
>>
>> tic, parfor idx = 1:80, pause(3), end, toc
Elapsed time is 61.088382 seconds.
>>
>> delete(gcp)
Parallel pool using the 'Processes' profile is shutting down.
```

```
>> c = parcluster("JUWELS");
>> pool = c.parpool(64);
Starting parallel pool (parpool) using the 'JUWELS' profile ...
additionalSubmitArgs =
    '--ntasks=64 --cpus-per-task=1 --ntasks-per-core=1
Connected to parallel pool with 64 workers.
>>
>> tic, parfor idx = 1:1280, pause(3), end, toc
Elapsed time is 60.416029 seconds.
```

Where is the scaling (2)?



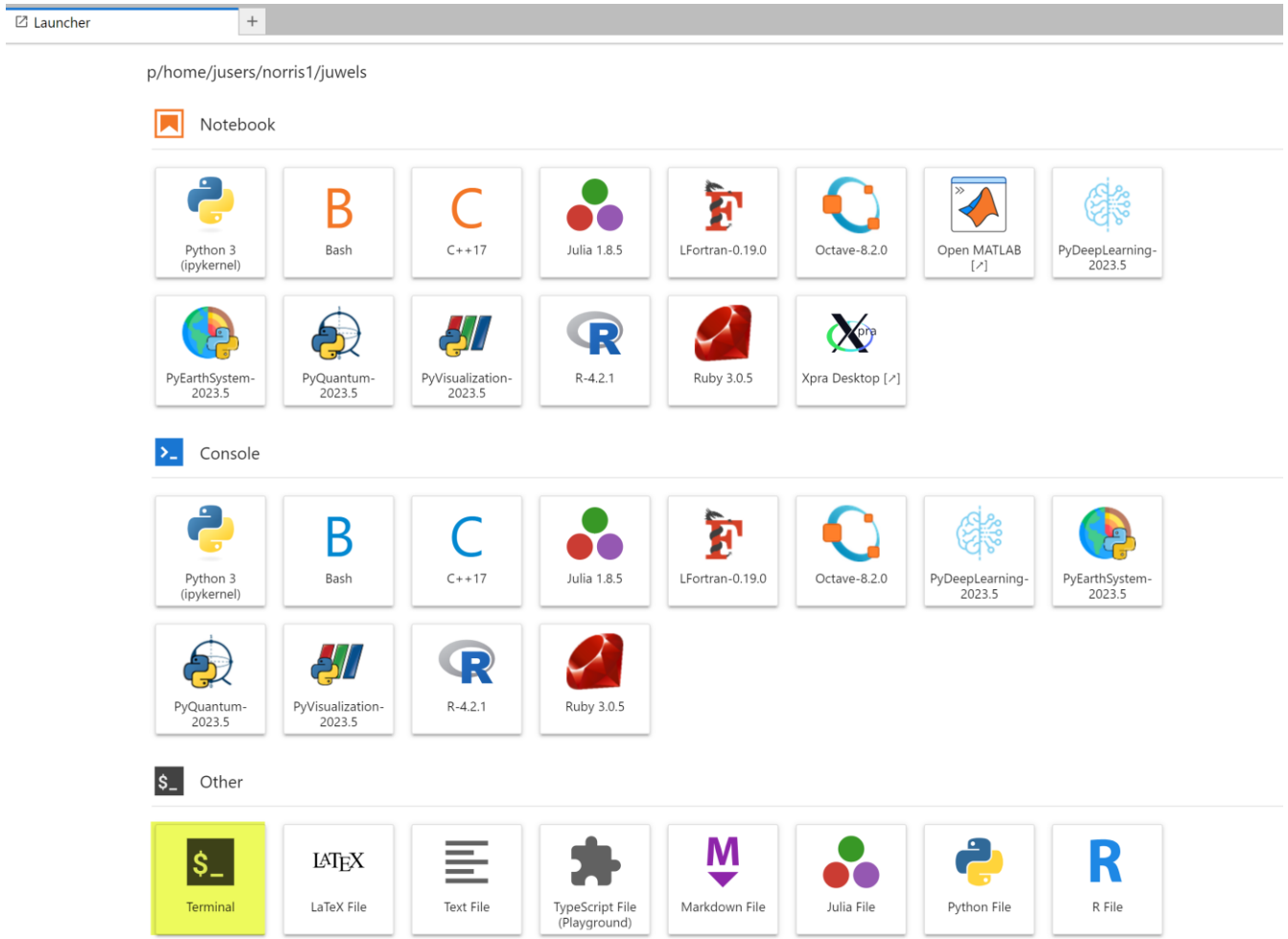

```
>> parpool("local",4);
Starting parallel pool (parpool) using the 'local' profile ...
Connected to parallel pool with 4 workers.
>>
>> tic, parfor idx = 1:80, pause(3), end, toc
Elapsed time is 61.088382 seconds.
>>
>> delete(gcp)
Parallel pool using the 'Processes' profile is shutting down.
```

```
>> c = parcluster("JUWELS");
>> pool = c.parpool(64);
Starting parallel pool (parpool) using the 'JUWELS' profile ...
additionalSubmitArgs =
    '--ntasks=64 --cpus-per-task=1 --ntasks-per-core=1
Connected to parallel pool with 64 workers.
>>
>> tic, parfor idx = 1:80, pause(3), end, toc
Elapsed time is 6.265180 seconds.
>>
>> tic, parfor idx = 1:100, pause(3), end, toc
Elapsed time is 6.061210 seconds.
```

How big of a Pool can or should we run? . . .

```
>> % Pool of 240 workers across 5 nodes
>> c = parcluster("JUWELS");
>> tic, pool = c.parpool(240); toc
Starting parallel pool (parpool) using the 'JUWELS' profile ...
additionalSubmitArgs =
    '--ntasks=240 --cpus-per-task=1 --ntasks-per-core=1
Connected to parallel pool with 240 workers.
Elapsed time is 159.348188 seconds.
>>
>> tic, parfor idx = 1:4800, pause(3), end, toc
Elapsed time is 60.473888 seconds.
>>
>> % Equivalent hours, if run serially
>> 4800 * 3 / 60 / 60
ans =
     4
>>
>> pool.delete
Parallel pool using the 'JUWELS' profile is shutting down.
```

JupyterLab – Shell terminal



Download workshop files

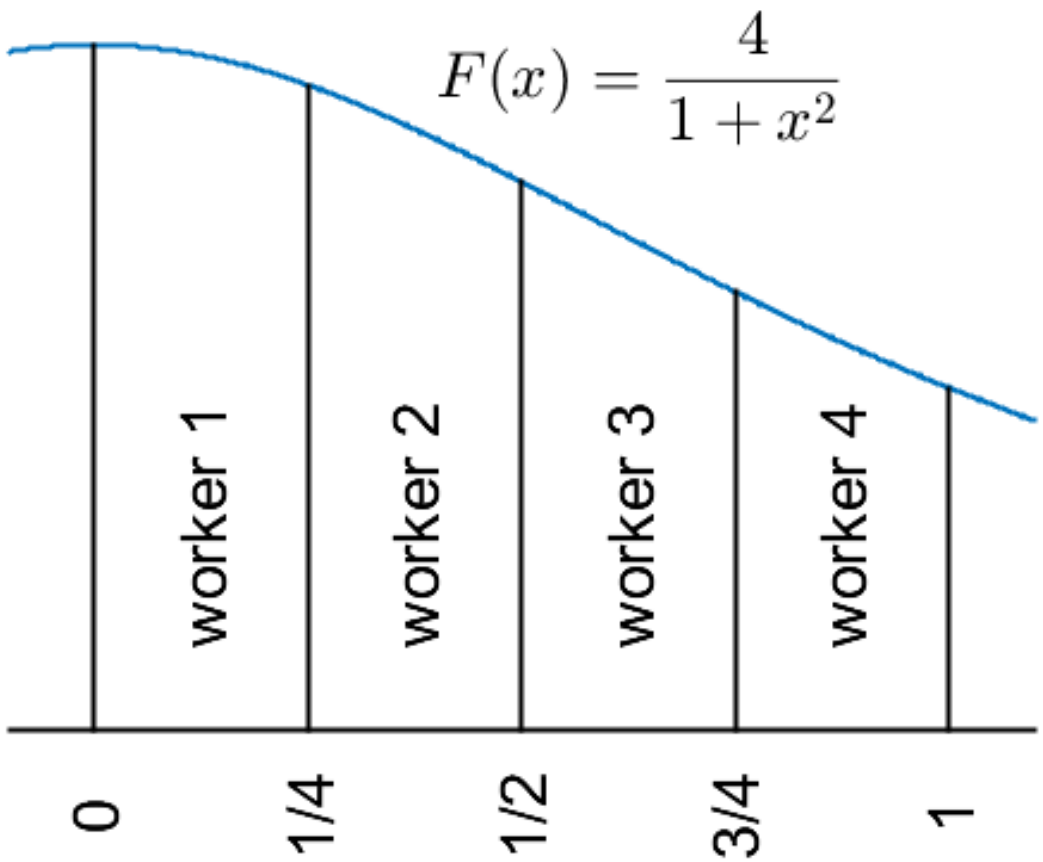
```
bash4.4$ # Copy Workshop files
bash4.4$ /p/project/training2339/copyWSfiles.sh
Copying the workshop files ...
Contents of ~/Documents/MATLAB/matlab-parallel-workshop:
Part-I
Part-II
Done.
bash4.4$
```

Change directories to workshop

```
>> cd(userpath)
>> cd("matlab-parallel-workshop/Part-II")
```

Exercise: Calculate π

$$\int_0^1 \frac{4}{1+x^2} dx = 4(atan(1) - atan(0)) = \pi$$



Calculate π

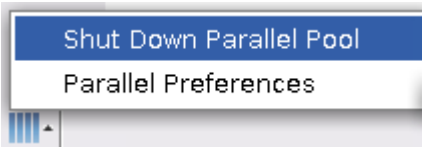
```
function calc_pi

p = gcp;
nsegments = p.NumWorkers;

% Range from 0 to 1, divided by number of workers
boundaries = linspace(0,1,nsegments+1);

parfor idx = 1:nsegments
    a = boundaries(idx)
    b = boundaries(idx+1);
    myIntegral(idx) = integral(@quadpi,a,b);
end

approx = sum(myIntegral);
fprintf('pi            : %.18f\n', pi)
fprintf('Approximation: %.18f\n', approx)
fprintf('Error         : %g\n',    abs(pi - approx))


function y = quadpi(x)
y = 4./(1 + x.^2);
```

Calculate π

```
function calc_pi_with_spmd

spmd
    a = (spmdIndex - 1)/spmdSize;
    b = spmdIndex/spmdSize;
    fprintf('Subinterval: [%-4g, %-4g]\n', a, b)

    myIntegral = integral(@quadpi, a, b);
    fprintf('Subinterval: [%-4g, %-4g]   Integral: %4g\n', ...
        a, b, myIntegral)

    piApprox = spmdPlus(myIntegral);
end

approx = piApprox{1}; % 1st element holds value on worker 1
fprintf('pi            : %.18f\n', pi)
fprintf('Approximation: %.18f\n', approx)
fprintf('Error         : %g\n',    abs(pi - approx))


function y = quadpi(x)
y = 4./(1 + x.^2);
```

Running on a local node

```
>> parpool("local",4);
Starting parallel pool (parpool) using the 'local' profile ...
Connected to parallel pool with 4 workers.
>>
>> calc_pi
pi            : 3.14159265358979311 6
Approximation: 3.14159265358979356 0
Error         : 4.44089e-16
```

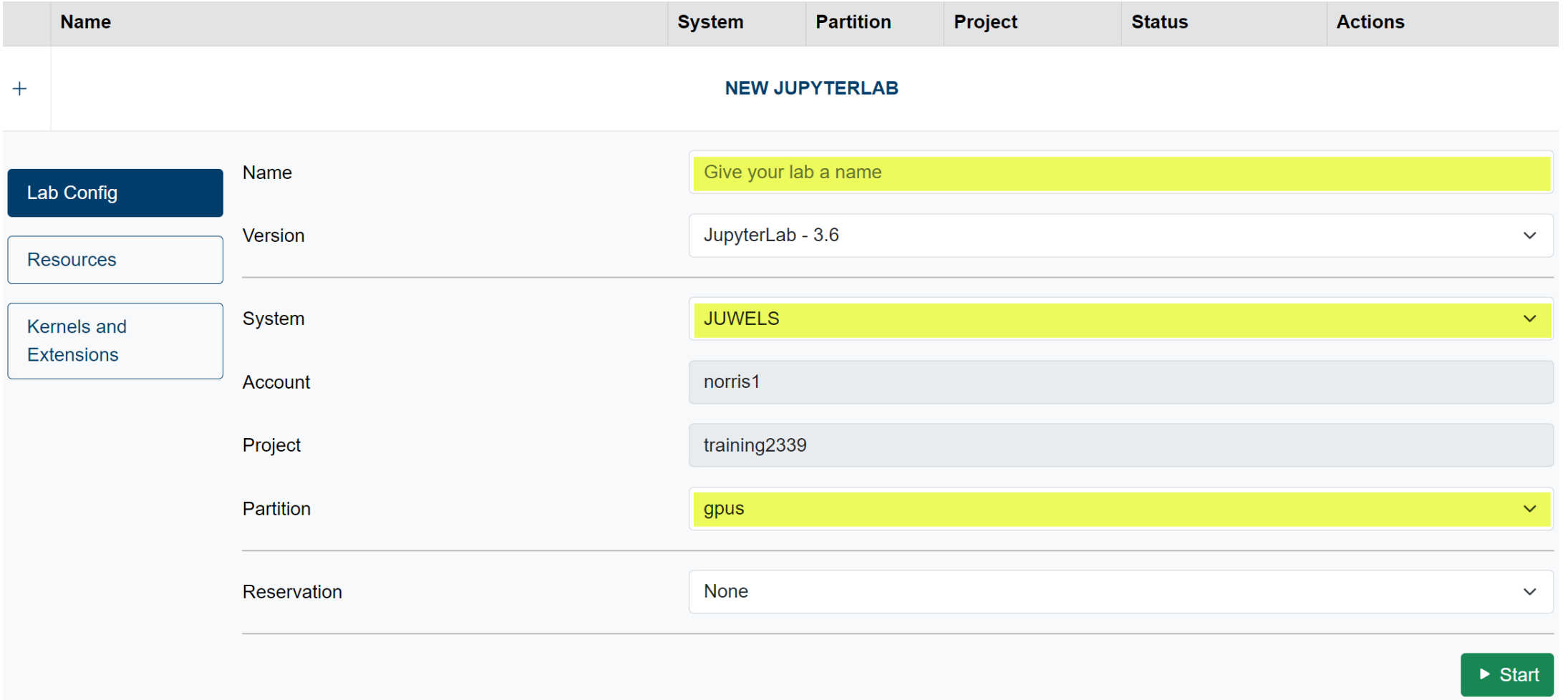


Running across multiple nodes

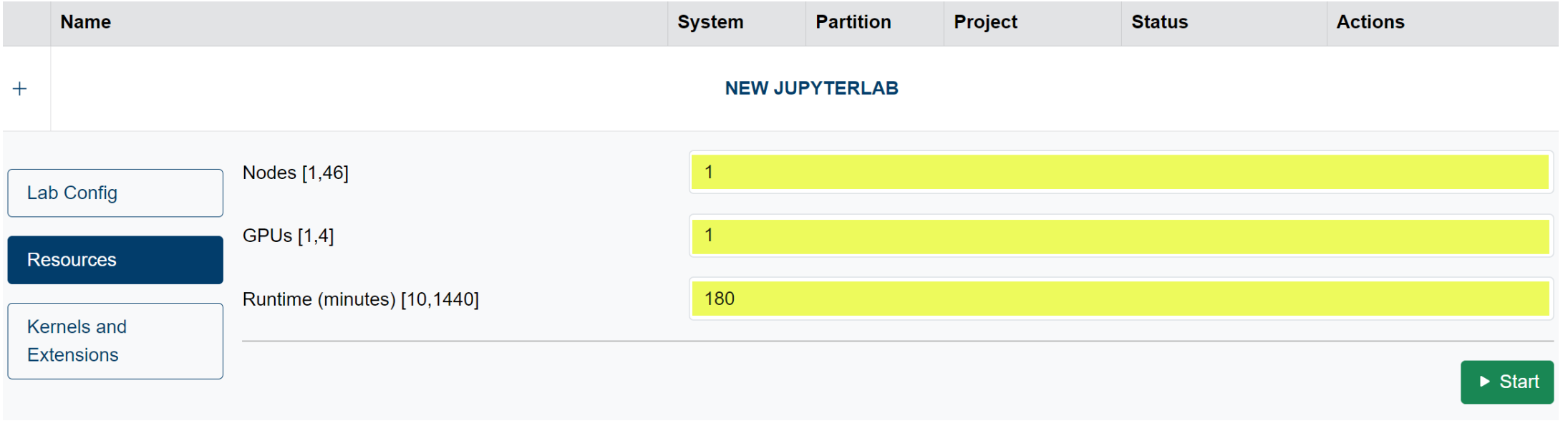
```
>> parpool("JUWELS",48);
Starting parallel pool (parpool) using the 'JUWELS' profile ...
additionalSubmitArgs =
    '--ntasks=48 --cpus-per-task=1 --ntasks-per-core=1
Connected to parallel pool with 48 workers.
>>
>> calc_pi
pi            : 3.14159265358979311 6
Approximation: 3.14159265358979311 6
Error         : 0
```

GPUs

Launching a JupyterLab on a GPU (1)



Launching a JupyterLab on a GPU (2)

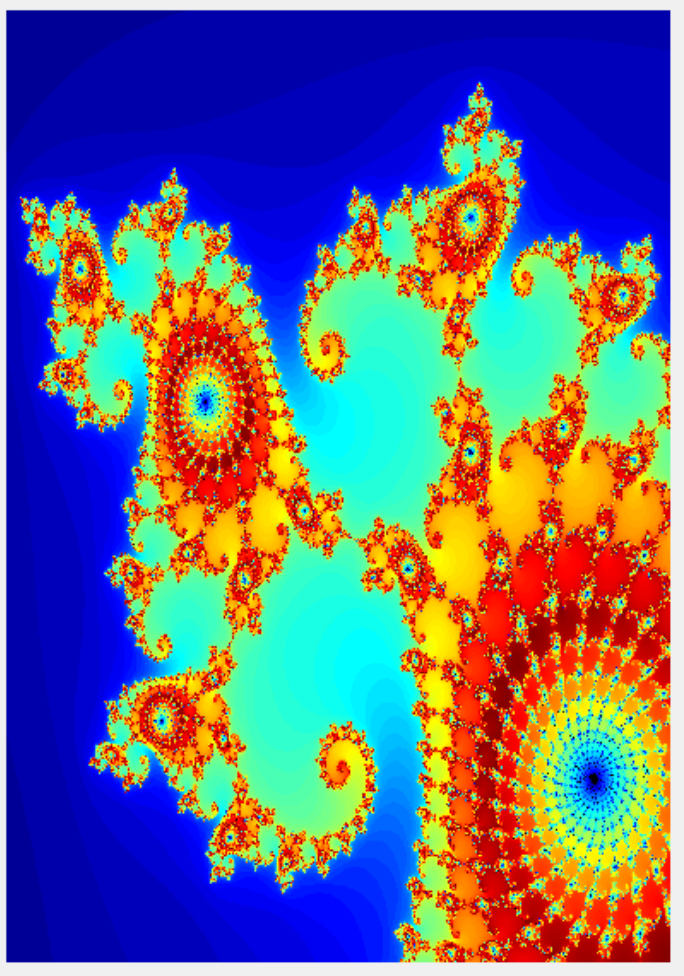


Tesla V100

```
>> gpuDevice
ans =
  CUDADevice with properties:

                      Name: 'Tesla V100-SXM2-16GB'
                     Index: 1
         ComputeCapability: '7.0'
            SupportsDouble: 1
     GraphicsDriverVersion: '535.54.03'
                DriverModel: 'N/A'
           ToolkitVersion: 11.8000
       MaxThreadsPerBlock: 1024
         MaxShmemPerBlock: 49152 (49.15 KB)
       MaxThreadBlockSize: [1024 1024 64]
              MaxGridSize: [2.1475e+09 65535 65535]
                SIMDWidth: 32
              TotalMemory: 16935682048 (16.94 GB)
```

Example: calc_mandelbrot (1)

```
function [x,y,count,t] = calc_mandelbrot(type)

maxIterations = 1000;
gridSize = 4000;
xlim = [-0.748766713922161, -0.748766707771757];
ylim = [ 0.123640844894862,  0.123640851045266];

t0 = tic;
if strcmp(type,'GPU')
    x = gpuArray.linspace(xlim(1),xlim(2),gridSize);
    y = gpuArray.linspace(ylim(1),ylim(2),gridSize);
    cname = 'gpuArray';
else
    x = linspace(xlim(1),xlim(2),gridSize);
    y = linspace(ylim(1),ylim(2),gridSize);
    cname = 'double';
end

[xGrid,yGrid] = meshgrid(x,y);
z0 = complex(xGrid,yGrid);
count = ones(size(z0),cname);

z = z0;
for n = 0:maxIterations
    z = z.*z + z0;
    inside = abs(z) <= 2;
    count = count + inside;
end
count = log(count);
t = toc(t0);
```



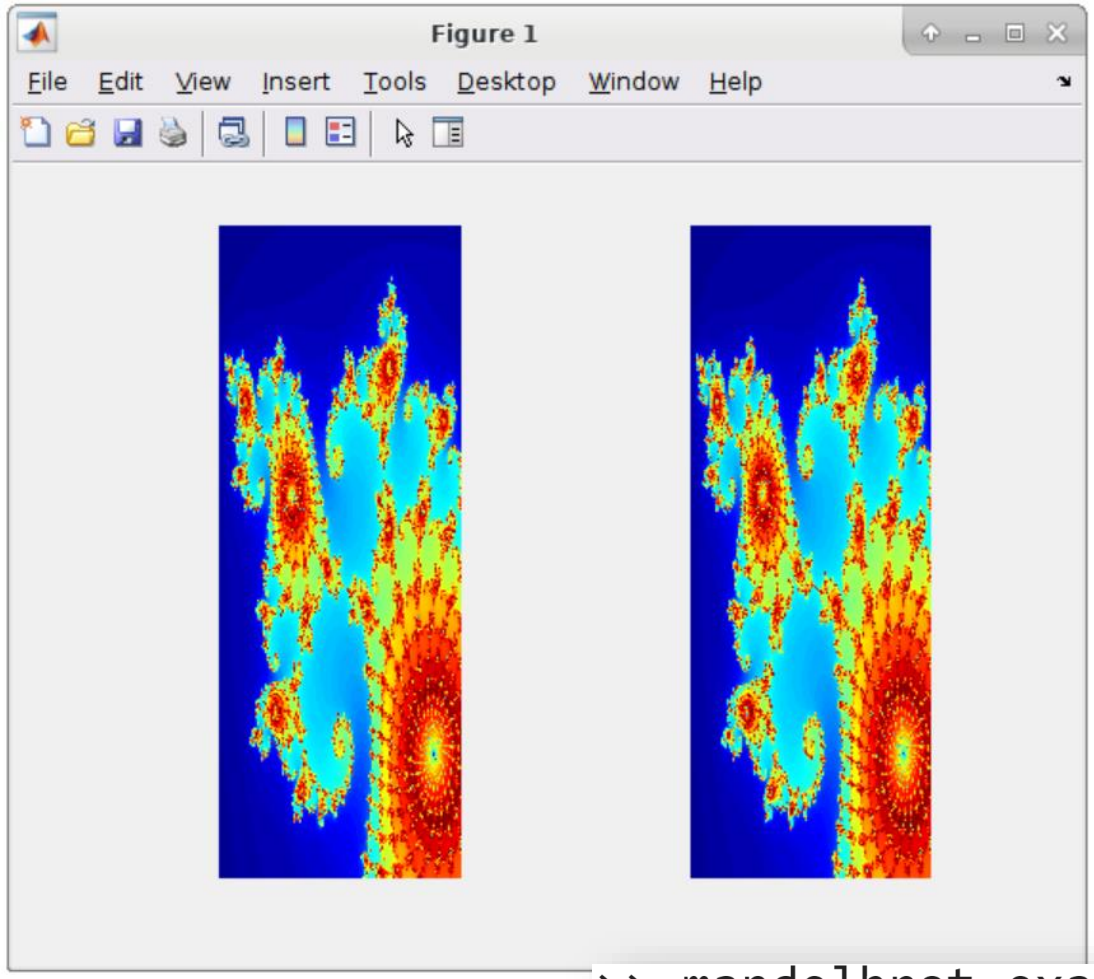
Example: calc_mandelbrot (2)

```
function mandelbrot_example

maxNumCompThreads(40);

% Run on CPU
[cpu_x,cpu_y,cpu_count,cpu_t] = calc_mandelbrot('CPU');

% Run on GPU
[gpu_x,gpu_y,gpu_count,gpu_t] = calc_mandelbrot('GPU');

figure
subplot(1,2,1)
imagesc(cpu_x,cpu_y,cpu_count)
colormap([jet; flipud(jet); 0 0 0]);
axis off

subplot(1,2,2)
imagesc(gpu_x,gpu_y,gpu_count)
colormap([jet; flipud(jet); 0 0 0]);
axis off

fprintf('CPU time: %0.2f\n',cpu_t)
fprintf('GPU time: %0.2f\n',gpu_t)
```

```
>> mandelbrot_example
```

Example: calc_mandelbrot (3)

```
>> mandelbrot_example
CPU time: 16.11
GPU time: 1.33
```




```
>> mandelbrot_example
CPU time: 373.91
GPU time: 0.51
```

FFT benchmarking

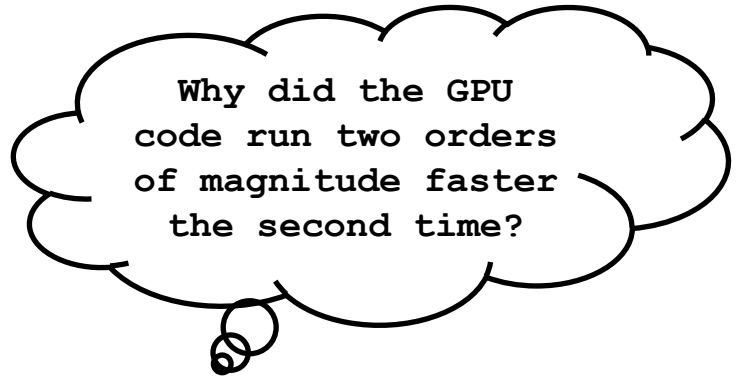
```
function [time_cpu, time_gpu] = calc_fft_cpu_gpu(N)

matrix_cpu = rand(N);

tic
out_cpu = fft(matrix_cpu);
time_cpu = toc;
disp(['Total time on CPU: ' num2str(time_cpu)])

t0 = tic;
% Transfer matrix to GPU device
matrix_gpu = gpuArray(matrix_cpu);

t1 = tic;
out_gpu = fft(matrix_gpu);
time_gfft = toc(t1);

% Gather back from GPU to CPU
gather_gpu = gather(out_gpu);

% Wait for transfer to complete
wait(gpuDevice)
time_gpu = toc(t0);

disp(['GPU FFT: ' num2str(time_gfft)])
disp(['Total time on GPU: ' num2str(time_gpu)])

disp(['FFT speed improvement: ' num2str(time_cpu/time_gfft)])
disp(['Total speed improvement: ' num2str(time_cpu/time_gpu)])
```

oom: Out of memory

```
>> % Clear GPU memory
>> reset(gpuDevice)
>>
>> % 8 GB matrix
>> [cpu_t, gpu_t] = calc_fft_cpu_gpu(2^15);
Total time on CPU: 0.65498
Error using fft
Out of memory on device. To view more detail about available memory on the GPU, use
'gpuDevice()'. If the problem persists, reset the GPU by calling 'gpuDevice(1)'.
Error in calc_fft_cpu_gpu (line 15)
out_gpu = fft(matrix_gpu);
```

2 GB matrix

```
>> % Clear GPU memory
>> reset(gpuDevice)
>>
>> % 2 GB matrix
>> [cpu_t, gpu_t] = calc_fft_cpu_gpu(2^14);
Total time on CPU: 0.12693
GPU FFT: 0.36451
Total time on GPU: 2.3109
FFT speed improvement: 0.34823
Total speed improvement: 0.054929
>>
>> [cpu_t, gpu_t] = calc_fft_cpu_gpu(2^14);
Total time on CPU: 0.12513
GPU FFT: 0.007035
Total time on GPU: 1.9625
FFT speed improvement: 17.7869
Total speed improvement: 0.06376
```



Interactively: with a batch job

`batch`

Exercise: “Hello, World!”


```
>> % Submit a job to the cluster to find out where MATLAB is running
>> % Get a handle to the HPC cluster
>> c = parcluster("JUWELS");
>> job = c.batch(@pwd,1,{});
```

#SBATCH …

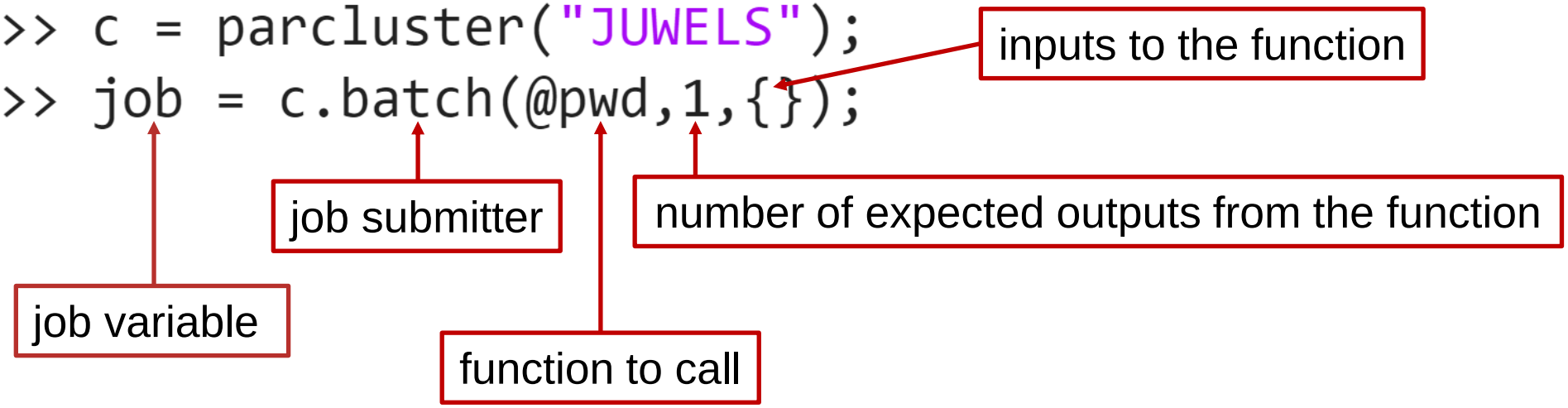
module load matlab

matlab …

If no profile is supplied to `parcluster`, use the default profile

Exercise: “Hello, World!”

```
>> % Submit a job to the cluster to find out where MATLAB is running
>> % Get a handle to the HPC cluster
>> c = parcluster("JUWELS");
>> job = c.batch(@pwd,1,{});
```



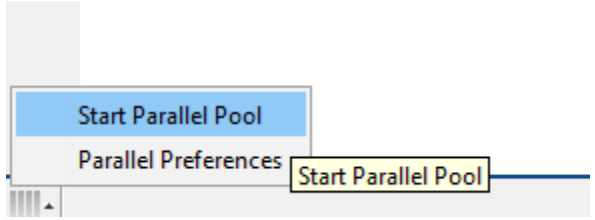
If no profile is supplied to `parcluster`, use the default profile

Fetching results

```
>> % Submit a job to the cluster to find out where MATLAB is running
>> % Get a handle to the HPC cluster
>> c = parcluster("JUWELS");
>> job = c.batch(@pwd,1,{});
additionalSubmitArgs =
    '--ntasks=1 --cpus-per-task=1 --ntasks-per-core=1
>>
>> % Check the state of the job
>> job.State
ans =
    'finished'
>>
>> % Fetch the results
>> job.fetchOutputs{:}
ans =
    '/p/home/jusers/norris1/juwels/Documents/MATLAB/matlab-parallel-workshop/Part-II'
```

How should you start a local parallel pool?

- Call `parpool` from the Command Window
- Have MATLAB automatically start a parallel pool if it hasn't already started
- From the lower lefthand corner



Where should you start a parallel pool?

```
function parallel_example

parfor idx = 1:8
    A(idx) = rand;
end
```

“What will happen the next time you run this code?”

```
Error using parpool
Found an interactive session. You cannot have multiple interactive sessions open simultaneously.
```

Then how do I tell the cluster my job needs a parallel pool?...

```
>> job = c.batch(.., 'Pool', pool_size);
```

Exercise: Submit `calc_pi` job

```
>> % Submit calc_pi job
>> c = parcluster("JUWELS");
>>
>> % Request 8 workers
>> job = c.batch(@calc_pi,0,{}, 'Pool',8);
additionalSubmitArgs =
    '--ntasks=9 --cpus-per-task=1 --ntasks-per-core=1
```

"If my Pool is size 8, why am I requesting 9 tasks?"

Fetch the results

```
>> % Submit calc_pi job
>> c = parcluster("JUWELS");
>>
>> % Request 8 workers
>> job = c.batch(@calc_pi,0,{}, 'Pool',8);
additionalSubmitArgs =
    '--ntasks=9 --cpus-per-task=1 --ntasks-per-core=1
>>
>> % Check the state of the job
>> job.State
ans =
    'finished'
>>
>> % Fetch the results
>> job.fetchOutputs{:}
>>
```

What can get returned from running a job?

- Function output
- Diary
- Saved files

Example

```
function [time, A] = test_fcn(sims)

disp('Start sim')

A = nan(sims,1);
t0 = tic;
parfor idx = 1:sims
    A(idx) = idx;
    pause(0.5)
    idx
end
time = toc(t0);

disp('Finished')

save RESULTS A
```

Job submission

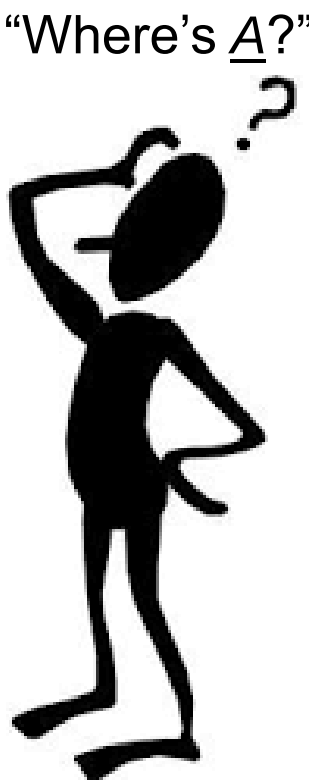
```
>> job2 = c.batch(@test_fcn,1,{300}, 'Pool',10);
additionalSubmitArgs =
    '--ntasks=11 --cpus-per-task=1 --ntasks-per-core=1
```

Fetch output

```
function [time, A] = test_fcn(sims)
```

```
c.batch(@test_fcn,1,{300},
```

```
>> % Fetch the results
>> job2.fetchOutputs{:}
ans =
   15.8432
>>
```



Diary

```
function [time, A] = test_fcn(sims)

disp('Start sim')

A = nan(sims,1);
t0 = tic;
parfor idx = 1:sims
    A(idx) = idx;
    pause(0.5)
    idx
end
time = toc(t0);

disp('Finished')

save RESULTS A
```

```
>> % View the diary
>> job2.diary
--- Start Diary ---
Start sim

ans =

    18



ans =

    14



ans =

    20
```

Fetch the diary from `calc_pi`

```
>> job.diary
--- Start Diary ---
pi            : 3.14159265358979311 6
Approximation: 3.14159265358979222 8
Error         : 8.88178e-16

--- End Diary ---
```

Save files

"Where does **RESULTS** get written to?"

```
function [time, A] = test_fcn(sims)

disp('Start sim')

A = nan(sims,1);
t0 = tic;
parfor idx = 1:sims
    A(idx) = idx;
    pause(0.5)
    idx
end
time = toc(t0);

disp('Finished')

save RESULTS A
```

“Who needs threads?...”

```
function t = threads_example(N)

matrix = rand(N);

t0 = tic;
fft(matrix);
t = toc(t0);

end
```

Single threaded vs multi-threaded

```
>> % Run on a single thread (core)
>> job_01 = c.batch(@threads_example,1,{2^14});
additionalSubmitArgs =
    '--ntasks=1 --cpus-per-task=1 --ntasks-per-core=1
>>
>> % Time to run on a single thread
>> t_01 = job_01.fetchOutputs{:}
t_01 =
    2.5239
>>
>> % Allocate 5 threads
>> c.NumThreads = 5;
>> job_05 = c.batch(@threads_example,1,{2^14});
additionalSubmitArgs =
    '--ntasks=1 --cpus-per-task=5 --ntasks-per-core=1
>>
>> % Time to run with 5 threads
>> t_05 = job_05.fetchOutputs{:}
t_05 =
    0.4825
```

Tuning jobs . . .

```
Failed to submit job to Slurm using command:
    sh
    '/p/home/jusers/norris1/juwels/.matlab/generic_cluster_jobs/juwels/Job84/tpa7405aa5_c7b0_4cf4_819f_383c7f0051
                Reason: sbatch: error: Batch job submission failed:
    Requested node configuration is not available
```

Other settable job properties (1)

```
>> c.AdditionalProperties
ans =
  AdditionalProperties with properties:

    AdditionalSubmitArgs: ''
              BudgetName: ''
              Constraint: ''
            EmailAddress: ''
             EnableDebug: 0
                 GPUCard: ''
             GPUsPerNode: 0
               Partition: ''
            ProcsPerNode: 0
             Reservation: ''
                WallTime: ''
```

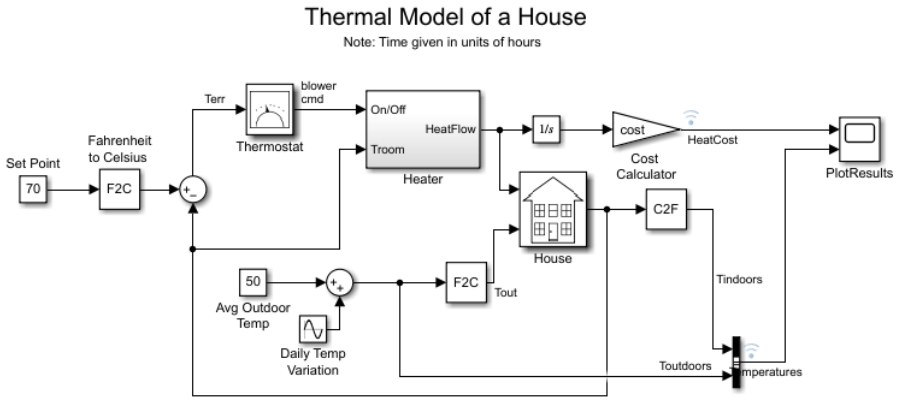
Other settable job properties (2)

- **BudgetName**
- Constraint
- EmailAddress
- GPUCard
- GPUsPerNode
- Partition
- ProcsPerNode
- Reservation
- **WallTime**

batchsim: Can I only run MATLAB? What about Simulink?

```
function simJob = batchsim_example(~)

c = parcluster;
c.AdditionalProperties.WallTime = '01:00:00';

mdl = 'ex_sldemo_househeat';

% Open and load model
openExample('simulink/OpenTheModelExample')
open_system(mdl)
load_system(mdl)

% Define temperatures
temps = 62:4:84;
tlen = length(temps);

% Initialize Simulation Inputs
in(1:tlen) = Simulink.SimulationInput(mdl);
for tidx = 1:tlen
    in(tidx) = in(tidx).setBlockParameter( ...
        [mdl '/Set Point'],'Value',num2str(temps(tidx)));
end

% Submit job
simJob = batchsim(c, in, 'ShowProgress','on', ...
    'StopOnError','on', 'Pool',tlen);
```

```
>> job = batchsim_example;
additionalSubmitArgs =
    '--ntasks=7 --cpus-per-task=1 --ntasks-per-core=1
>> job.diary
--- Start Diary ---
[18-Sep-2023 22:07:25] Checking for availability of parallel pool...
[18-Sep-2023 22:07:26] Starting Simulink on parallel workers...
[18-Sep-2023 22:07:50] Configuring simulation cache folder on parallel workers
[18-Sep-2023 22:07:50] Loading model on parallel workers...
[18-Sep-2023 22:08:14] Running simulations...
[18-Sep-2023 22:08:28] Completed 1 of 6 simulation runs
[18-Sep-2023 22:08:28] Completed 2 of 6 simulation runs
[18-Sep-2023 22:08:28] Completed 3 of 6 simulation runs
[18-Sep-2023 22:08:28] Completed 4 of 6 simulation runs
[18-Sep-2023 22:08:28] Completed 5 of 6 simulation runs
[18-Sep-2023 22:08:28] Completed 6 of 6 simulation runs
[18-Sep-2023 22:08:29] Cleaning up parallel workers...

--- End Diary ---
```



Submitting scripts, instead of functions

```
>> x = 4;
>> z = rand(3);
>>
>> % Submit a script (instead of a function)
>> job = c.batch('temp = rand(10); y = x, who');
additionalSubmitArgs =
    '--ntasks=1 --cpus-per-task=1 --ntasks-per-core=1
>> clear z
>>
>> who

Your variables are:

c    job  x

>> % Check the state of the job
>> job.State
ans =
    'finished'
```

Loading variables to local workspace

"If we cleared `z`, then why does `who` display it? And I didn't need `temp`!

```
>> % Load variables
>> job.load
>> who

Your variables are:

ans    c      job    temp   x      y      z

>> job.diary
--- Start Diary ---

y =

     4


Your variables are:

temp   x      y      z

--- End Diary ---
```

"I'll pass all the variables in your local workspace to all of the workers. Then I'll pass everything the workers generate and pass it back to your local workspace."

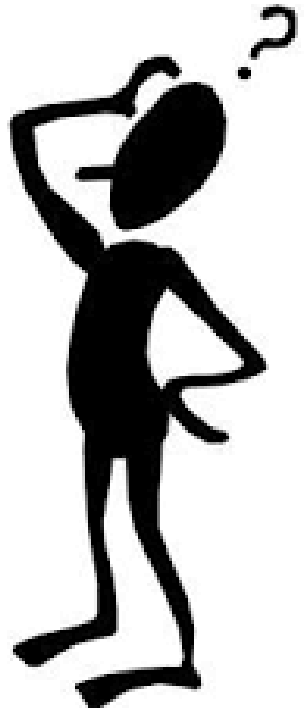
```
>> z = rand(3);
>>
>> % Submit a sc
>> job = c.batch
```

The cost of transferring unnecessary data

```
>> tic, job = c.batch('temp = rand(10); y = x, who'); toc
Elapsed time is 12.985297 seconds.
>>
>> z = rand(5000);      % 191 MB
>> tic, job = c.batch('temp = rand(10); y = x, who'); toc
additionalSubmitArgs =
    '--ntasks=1 --cpus-per-task=1 --ntasks-per-core=1
Elapsed time is 19.362395 seconds.
```

`willRun`: When will my job run?

```
>> % Run a job across multiple nodes
>> job = c.batch(@test_fcn,1,{630}, 'Pool',63);
additionalSubmitArgs =
    '--ntasks=64 --cpus-per-task=1 --ntasks-per-core=1
>>
>> % Get the job's state
>> job.State
ans =
    'queued'
>>
>> % Why is my job queued?
>> willRun(job)
ans =
    'Priority'
```

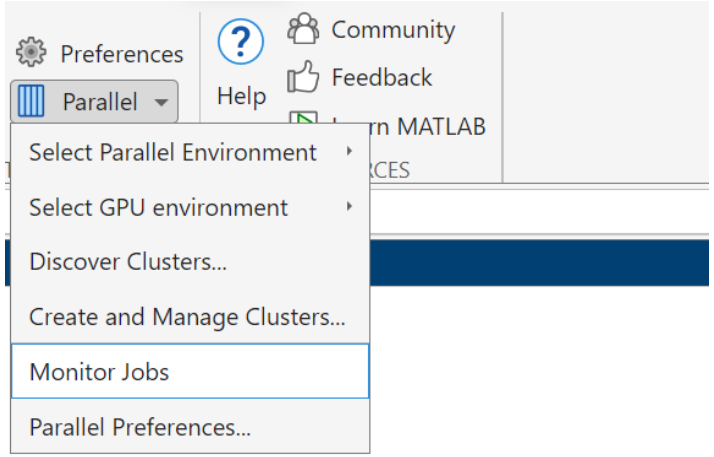


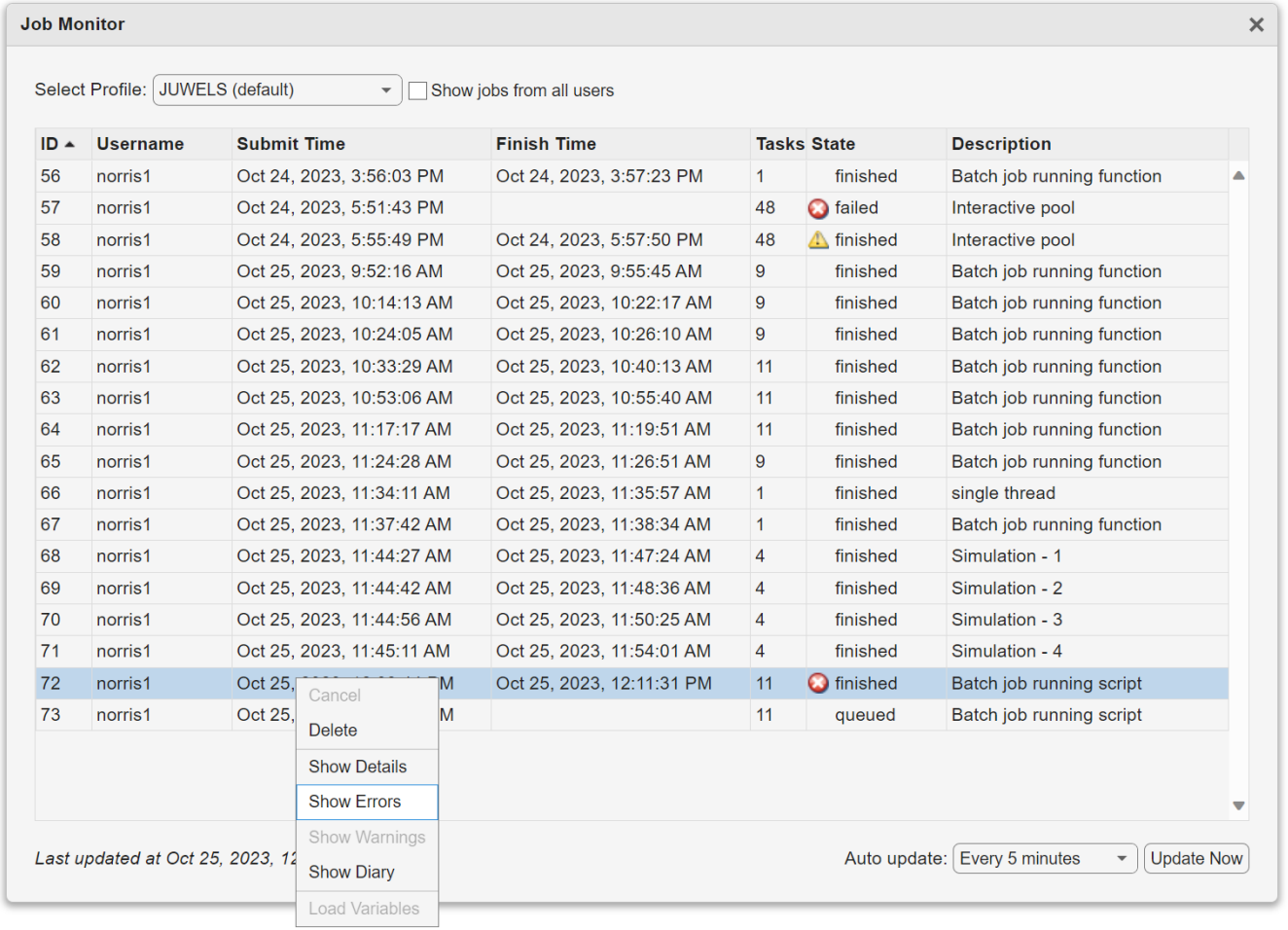
When has my job run and finished?

```
>> % Get email notification when the job has finished
>> c.AdditionalProperties.EmailAddress = 'user-id@fz-juelich.de';
>>
>> job = c.batch(@test_fcn,1,{300}, 'Pool',10);
additionalSubmitArgs =
    '--ntasks=11 --cpus-per-task=1 --ntasks-per-core=1 --mem-per-cpu
```

```
--mail-type=ALL --mail-user=user-id@fz-juelich.de
```

Retrieving past jobs





Keep cluster files minimal: delete jobs

- As a good practice, delete jobs you no longer need

```
>> % Delete jobs that are no longer needed
>> job.delete
```

Debugging and Troubleshooting

Example: Errored jobs (1)

```
>> % Undefined function
>> job = c.batch(@invalid_fcn,0,{});
additionalSubmitArgs =
    '--ntasks=1 --cpus-per-task=1 --ntasks-per-core=1
>>
>> % Reference to undefined variable or function
>> job2 = c.batch('y = a');
additionalSubmitArgs =
    '--ntasks=1 --cpus-per-task=1 --ntasks-per-core=1
```

Example: Errored jobs (2)

```
>> % Undefined function
>> job.State
>>
>> job.fetchOutputs{:}
Error using parallel.Job/fetchOutputs
An error occurred during execution of Task with ID 1.
Caused by:
    Unrecognized function or variable 'invalid_fcn'.
```

Fetch outputs, even with no outputs returned, to get the error message

Example: Errored submissions (3)

```
>> % Undefined variable in a script
>> job2.State
ans =
    'finished'
>>
>> job2.load
Error using parallel.Job/load
Error encountered while running the batch job. The error was:

Unrecognized function or variable 'a'.

If 'a' is a file that is required by 'y = a', add the full path name for 'a' to
the batch job's AttachedFiles property. For more information, see batch.
```

Logfile: Single core job

```
>> job = c.batch(@test_fcn,1,{300}, 'Pool',0);
additionalSubmitArgs =
    '--ntasks=1 --cpus-per-task=1 --ntasks-per-core=1
>>
>> % Retrieve log file for single worker job
>> c.getDebugLog(job.Tasks)
LOG FILE OUTPUT:
The scheduler has allocated the following node to this job: jwc00n011.juwels
Executing: "/p/software/juwels/stages/2023/software/MATLAB/2023b-GCCcore-11.3.0/bin/worker"

        Sponsored License -- for use in support of a program or activity sponsored by MathWorks.
        Not for government, commercial or other non-sponsored organizational use.

Exiting with code: 0
```

Logfile: Multi-core job

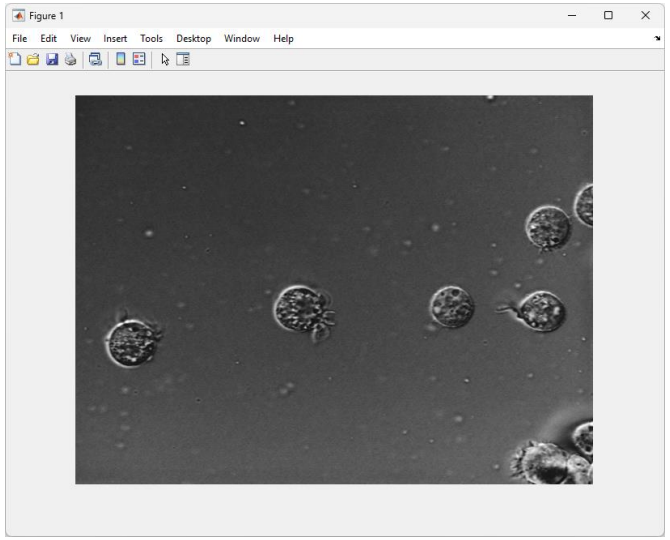
```
>> job = c.batch(@test_fcn,1,{300}, 'Pool',2);
additionalSubmitArgs =
    '--ntasks=3 --cpus-per-task=1 --ntasks-per-core=1
>>
>> % Retrieve log file for multi-worker job
>> c.getDebugLog(job)
LOG FILE OUTPUT:
The scheduler has allocated the following nodes to this job:
jwc00n024
"/p/software/juwels/stages/2023/software/MATLAB/2023b-GCCcore-11.3.0/bin/mw_mpiexec" -bind-to core:1 -l -n 3
[0]
[0]     Sponsored License -- for use in support of a program or activity sponsored by MathWorks.
[0]     Not for government, commercial or other non-sponsored organizational use.
[0]
[0] Parallel pool is shutting down.
Exiting with code: 0
```

Scheduler ID

```
>> job = c.batch(@test_fcn,1,{300}, 'Pool',2);
additionalSubmitArgs =
    '--ntasks=3 --cpus-per-task=1 --ntasks-per-core=1
>>
>> % Job ID vs Scheduler ID
>> job.ID
ans =
    74
>>
>> job.getTaskSchedulerIDs{:}
ans =
    '8756352'
```

Designing Robust Code

From Coding to Cluster




```
% Notes - From Coding to Cluster
% 1. Using a script, not a function
% 2. Paths are hardcoded
% 3. File separator is hard coded
% 4. Assumes TIF file exists
% 5. TIF files must be on the MATLAB path
% 6. Assumes output folder already exists where ever MATLAB is running
% 7. Results MAT-File will be overwritten next time it's run
```

```
filelist = dir('tif\*.tif');
fileNames = {filelist.name}';

segmentedCellSequence = batchProcessFiles(@detectCells,fileNames);
cd output
save SCS segmentedCellSequence
```

```
function [ofile, segmentedCellSequence] = process_files_v2(rootd,outd)
if nargin==0
    rootd = fullfile(pwd,'tif');
    outd = fullfile(pwd,'output');
end

filelist = dir(fullfile(rootd,'*.tif'));
if isempty(filelist)
    error('Failed to find image files: %s',rootd)
end
fileNames = {filelist.name}';

addpath(rootd)
segmentedCellSequence = batchProcessFiles(@detectCells,fileNames);

% Ensure output directory exists
if exist(outd,'dir')==false
    [passed,emsg,eid] = mkdir(outd);
    if passed==false
        error(eid,emsg)
    end
end

% Add timestamp for file uniqueness
ts = strrep(strrep(datestr(now),' ','_'),':','-');
ofile = fullfile(outd,['SCS_' ts]);
save(ofile,'segmentedCellSequence')
```

Run it locally

```
>> parpool("local",4);
Starting parallel pool (parpool) using the 'local' profile ...
Connected to parallel pool with 4 workers.
>>
>> ofile = process_files_v2
ofile =
    '/p/home/jusers/norris1/juwels/Documents/MATLAB/matlab-parallel-workshop/Part-II/output/SCS_25-Oct-2023_18-21-40'
>>
```

Run it on the cluster

```
>> c = parcluster("JUWELS");
>> job = c.batch(@process_files_v2,1,{'~/Documents','~/output'}, 'Pool',3);
additionalSubmitArgs =
    '--ntasks=4 --cpus-per-task=1 --ntasks-per-core=1
>>
>> % Wait for the job to finish
>> job.wait
>>
>> % Fetch the results
>> ofile = job.fetchOutputs{:}
Error using parallel.Job/fetchOutputs
An error occurred during execution of Task with ID 1.
Caused by:
    Error using process_files_v2
    Failed to find image files: ~/Documents
```

Run it on the cluster

```
>> c = parcluster("JUWELS");
>> job = c.batch(@process_files_v2,1,{'~/Documents/MATLAB/matlab-parallel-workshop/Part-II/tif','~/output'}, 'Pool',3);
additionalSubmitArgs =
    '--ntasks=4 --cpus-per-task=1 --ntasks-per-core=1
>>
>> % Wait for the job to finish
>> job.wait
>>
>> % Fetch the results
>> ofile = job.fetchOutputs{:}
ofile =
    '~/output/SCS_25-Oct-2023_18-28-56'
```

From Coding to Cluster (2)

```
% Notes - "From Coding to Cluster"
% 1. Using a script, not a function
%       return status or output directory
% 2. Paths are hardcoded
%       pass in root directory
% 3. File separator is hardcoded
%       use fullfile
% 4. Assumes TIF files exist
%       check results when touching file system
% 5. TIF files must be on the MATLAB path
%       add tif folder to the MATLAB path
% 6. Assumes output folder already exists wherever MATLAB is running
%       supply output folder to write to.  check if folder exists;
%       if not, create it
% 7. Results MAT-File will be overwritten next time job is run
%       add timestamp to filename
```

Running bulk jobs

```
function jobs = submit_jobs

c = parcluster;
c.AdditionalProperties.EmailAddress = 'my-email@work';

sims = [54 162 324 648];

for sidx = 1:length(sims)
    % Run code with different number of iterations
    jobs(sidx) = c.batch(@parallel_example,1,{sims(sidx)}, 'Pool',3);
    % Tag the job so that it's easier to find in the Job Monitor
    jobs(sidx).Tag = sprintf("Simulation - %d", sidx);
end

% Wait for the 2nd job to finish
jobs(2).wait

% Get the time for the 2nd job
t = jobs(2).fetchOutputs{:}
```

66	norris1	Oct 25, 2023, 11:34:11 AM	Oct 25, 2023, 11:35:57 AM	1	finished	single thread
67	norris1	Oct 25, 2023, 11:37:42 AM	Oct 25, 2023, 11:38:34 AM	1	finished	Batch job running function
68	norris1	Oct 25, 2023, 11:44:27 AM	Oct 25, 2023, 11:47:24 AM	4	finished	Simulation - 1
69	norris1	Oct 25, 2023, 11:44:42 AM	Oct 25, 2023, 11:48:36 AM	4	finished	Simulation - 2
70	norris1	Oct 25, 2023, 11:44:56 AM	Oct 25, 2023, 11:50:25 AM	4	finished	Simulation - 3
71	norris1	Oct 25, 2023, 11:45:11 AM	Oct 25, 2023, 11:54:01 AM	4	finished	Simulation - 4

Noninteractively: write a Slurm job script

`sbatch`

Submitting “locally” vs “multi-node”

```
function calc_pi_single_node

p = gcp("nocreate");
if isempty(p)
    % Query for available cores
    sz = str2num(getenv('SLURM_TASKS_PER_NODE'));
    if isempty(sz), sz = maxNumCompThreads; end
    p = parpool("local",sz);
end

nsegments = p.NumWorkers;

% Range from 0 to 1, divided by number of workers
boundaries = linspace(0,1,nsegments+1);

parfor idx = 1:nsegments
    a = boundaries(idx)
    b = boundaries(idx+1);
    myIntegral(idx) = integral(@quadpi,a,b);
end

approx = sum(myIntegral);
fprintf('pi            : %.18f\n', pi)
fprintf('Approximation: %.18f\n', approx)
fprintf('Error         : %g\n',    abs(pi - approx))

end

function y = quadpi(x)
y = 4./(1 + x.^2);
```

```
function calc_pi_multi_node

p = gcp("nocreate");
if isempty(p)
    p = parpool("JUWELS",48);
end

nsegments = p.NumWorkers;

% Range from 0 to 1, divided by number of workers
boundaries = linspace(0,1,nsegments+1);

parfor idx = 1:nsegments
    a = boundaries(idx)
    b = boundaries(idx+1);
    myIntegral(idx) = integral(@quadpi,a,b);
end

approx = sum(myIntegral);
fprintf('pi            : %.18f\n', pi)
fprintf('Approximation: %.18f\n', approx)
fprintf('Error         : %g\n',    abs(pi - approx))

end

function y = quadpi(x)
y = 4./(1 + x.^2);
```

Submit single-node job

```
#!/bin/sh

#SBATCH --tasks-per-node=8          # 8 local workers
#SBATCH --mem-per-cpu=4gb           # 4 GB RAM per core
#SBATCH --time=00:10:00             # 10 minutes

# Add MATLAB to system path
module load MATLAB

# Run code
matlab -batch calc_pi_single_node
```

Remember: Add MATLAB to the system path

matlab-single-node.slurm

Single-node job (1)

```
bash4.4$ sbatch -A training2339 matlab-single-node.slurm
Submitted batch job 8750373
bash4.4$
bash4.4$ squeue -j 8750373
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
           8750373     batch matlab-s  norris1  R       0:37      1 jwc00n263
bash4.4$
```

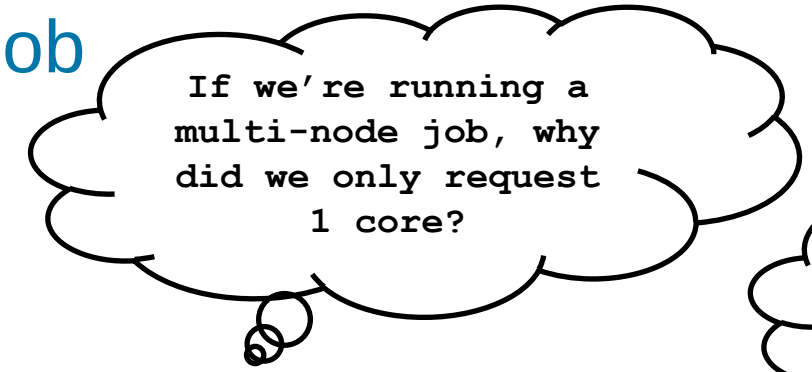
Single-node job (2)

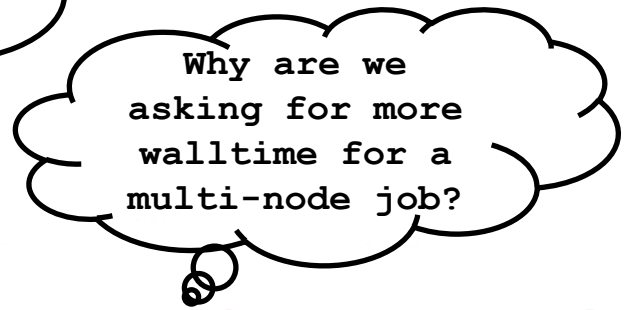
```
bash4.4$ cat slurm-8750373.out
Attention: This software is RESTRICTED to ACADEMIC users who are members of the group matlab.

        Sponsored License -- for use in support of a program or activity sponsored by MathWorks.
        Not for government, commercial or other non-sponsored organizational use.

Starting parallel pool (parpool) using the 'local' profile ...
Connected to parallel pool with 8 workers.
pi           : 3.141592653589793116
Approximation: 3.141592653589792228
Error        : 8.88178e-16
Parallel pool using the 'Processes' profile is shutting down.
bash4.4$
```

```
p = gcp("nocreate");
if isempty(p)
    % Query for available cores
    sz = str2num(getenv('SLURM_TASKS_PER_NODE'));
    if isempty(sz), sz = maxNumCompThreads; end
    p = parpool("local",sz);
end
```

```
#!/bin/sh

#SBATCH --tasks-per-node=8
#SBATCH --mem-per-cpu=4gb
#SBATCH --time=00:10:00
```

Submit multi-node job






```
#!/bin/sh

#SBATCH -n 1                    # 1 instance of MATLAB
#SBATCH --mem-per-cpu=4gb       # 4 GB RAM per core
#SBATCH --time=00:20:00         # 20 minutes

# Add MATLAB to system path
module load MATLAB

# Run code
matlab -batch calc_pi_multi_node
```

matlab-multi-node.slurm

Multi-node job (1)

```
bash4.4$ sbatch -A training2339 matlab-multi-node.slurm
Submitted batch job 8750447
bash4.4$
bash4.4$ squeue -j 8750447
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
           8750447     batch matlab-m  norris1  R       0:32      1 jwc00n263
bash4.4$
bash4.4$ squeue -u $USER
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)

           8750447     batch matlab-m  norris1  R       1:32      1 jwc00n263
           8750449     devel    Job58  norris1  R       0:13      1 jwc00n005
```

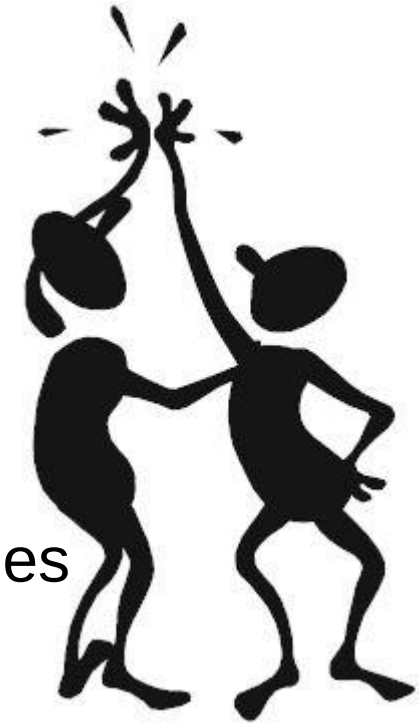
Multi-node pool of workers (2)

```
bash4.4$ cat slurm-8750447.out
Attention: This software is RESTRICTED to ACADEMIC users who are members of the group matlab.

        Sponsored License -- for use in support of a program or activity sponsored by MathWorks.
        Not for government, commercial or other non-sponsored organizational use.

Starting parallel pool (parpool) using the 'JUWELS' profile ...
additionalSubmitArgs =
    '--ntasks=48 --cpus-per-task=1 --ntasks-per-core=1 -A training2339 --mem-per-cpu=4gb'
Connected to parallel pool with 48 workers.
pi            : 3.1415926535897916
Approximation: 3.1415926535897916
Error         : 0
Parallel pool using the 'JUWELS' profile is shutting down.
bash4.4$
```

Wait! What about JURECA and JUSUF?

- Also supported, just need to “Discover” them as well . . .

Summary



- Use JupyterLabs to connect to JUWELS
- Prototype running on interactive compute node
- Create JUWELS profile by discovering cluster
- Toggle between local (single node) and JUWELS (multi-node) profiles
- Tune your job with AdditionalProperties
- Submit large scale from interactive session or Slurm job
- Tips, tricks and best practices for job submission and troubleshooting
- Contact IT Service Desk to get started
  - sc@fz-juelich.de
  - Who would be interested in 1-1 Coaching?