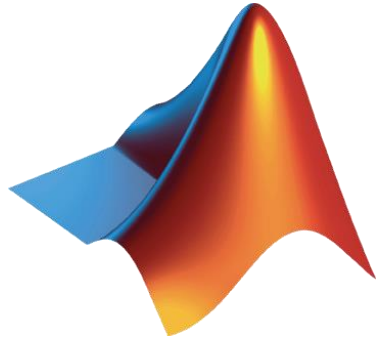
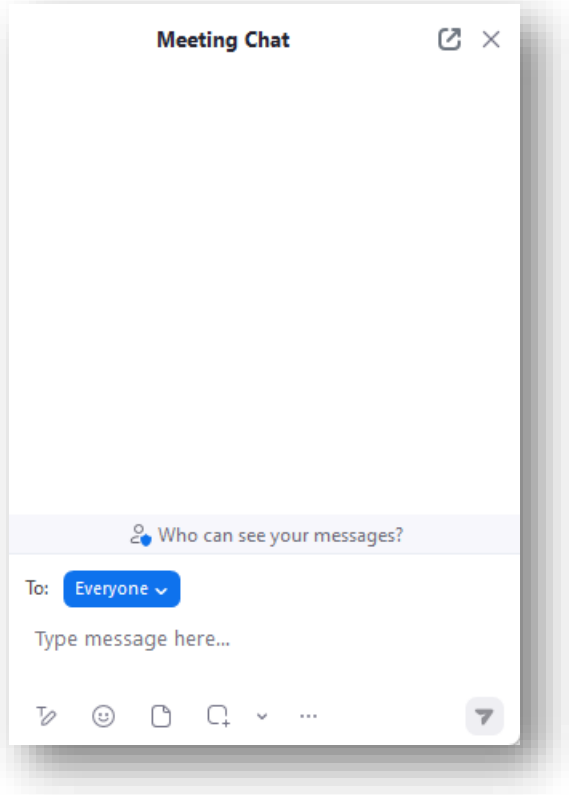
WORKSHOP: Parallel Computing With MATLAB (Part II)

Damian Pietrus
Application Engineer
May 7, 2024



Meeting Chat

- Please send chats to Everyone



Agenda

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

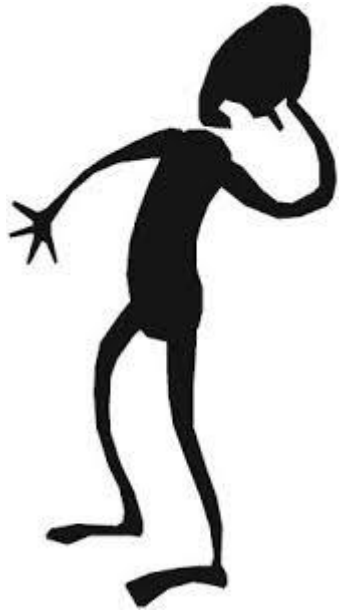
Agenda

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

Overview

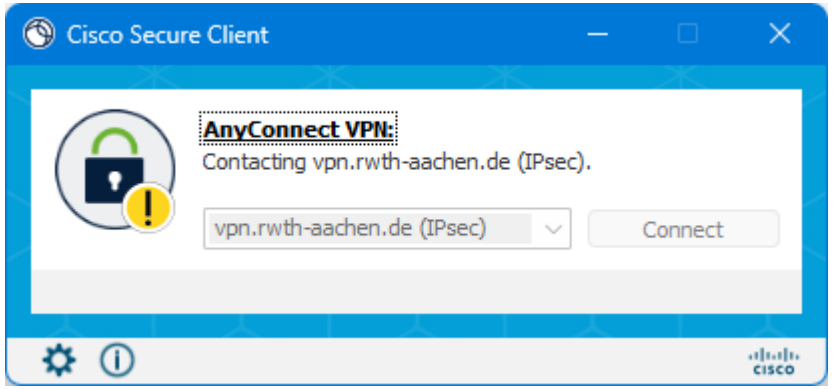
- How to configure MATLAB to submit remote jobs to the CLAIX Cluster
- The job submission workflow
- Ways to tune job submissions to the cluster
- How to optimize job submissions
- Troubleshooting job submission techniques
- Best practices for rehosting code onto the cluster

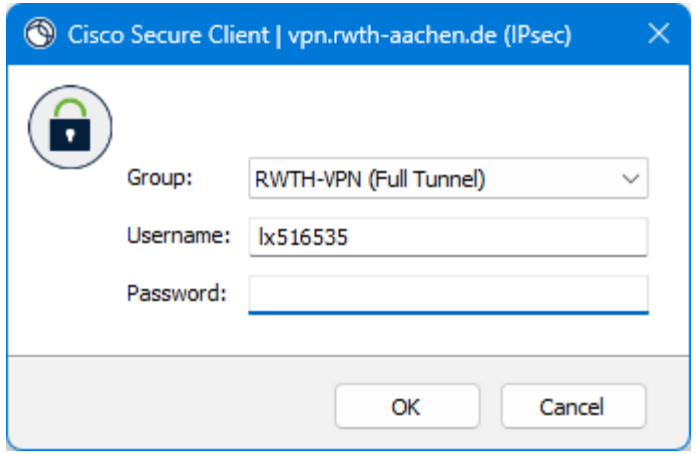
A note about today's workshop…

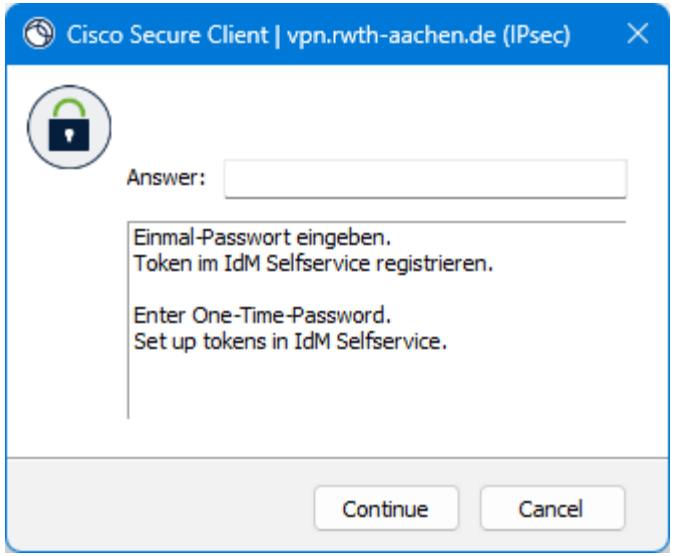


- The workflow and examples are about process, not performance
- MATLAB Getting Started Guide
  - https://help.itc.rwth-aachen.de/en/service/rhr4fjjutttf/article/271f660b911f4d9d9812cd7405785b77/
- Requirements
  - ❑ MATLAB and Parallel Computing Toolbox
    - ❑ (**Must match versions installed on the cluster**)
  - ❑ Account on CLAIX
  - ❑ [Optional] VPN for off-campus connectivity

VPN – Required for Off-Campus Cluster Access



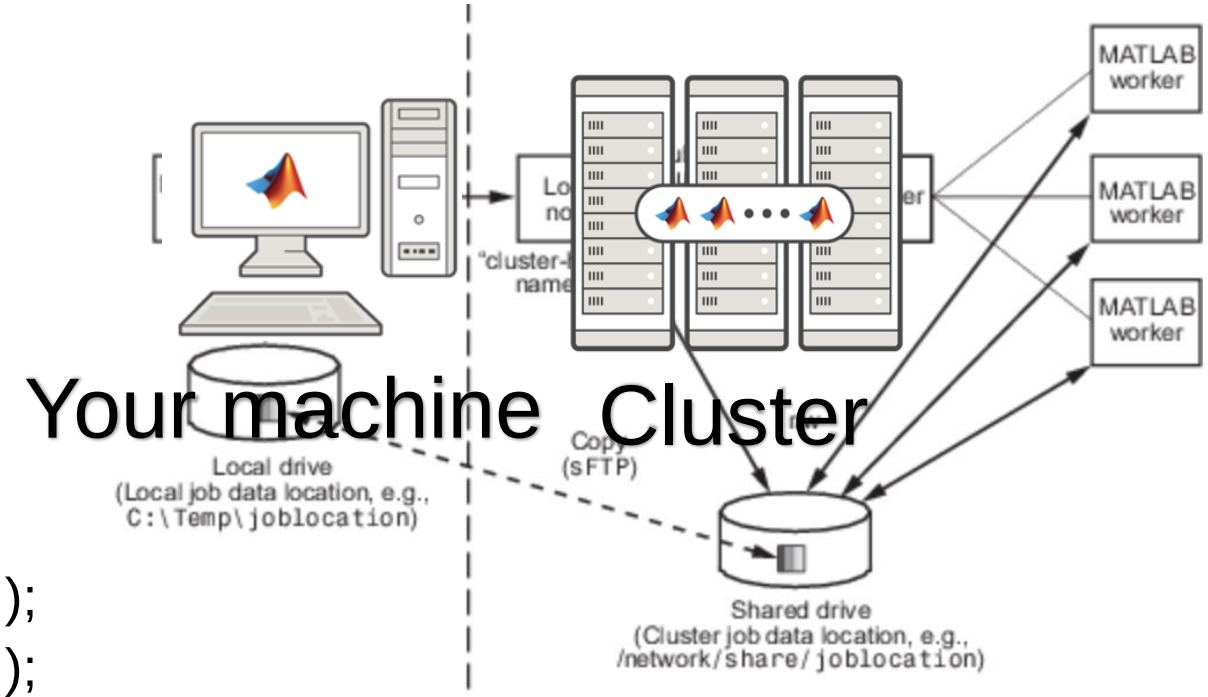




Desktop vs on-cluster submission

- Today’s workshop will focus on desktop submission
- For on-cluster submissions, remote desktops are available via Fast-X.

Scaling MATLAB to CLAIX



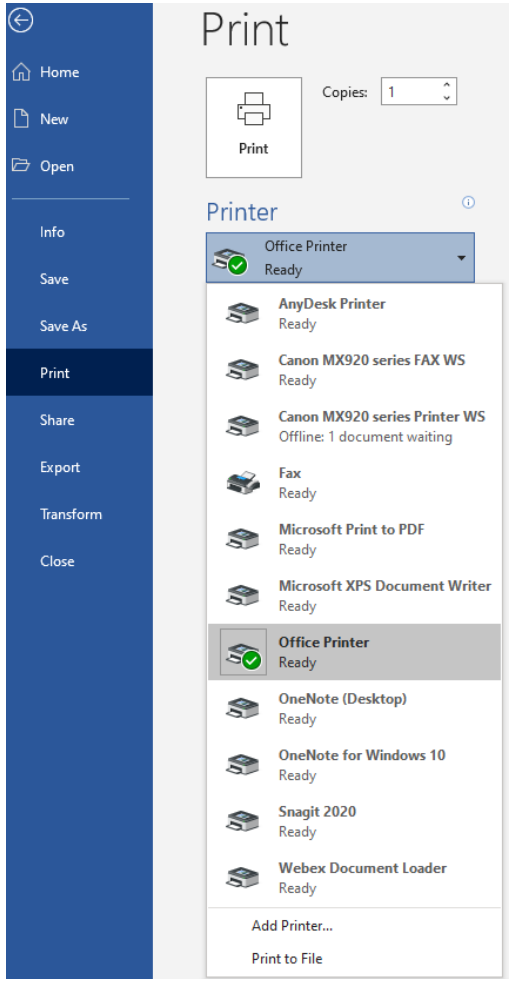

```
>> job1 = batch( );
>> job2 = batch( );
```

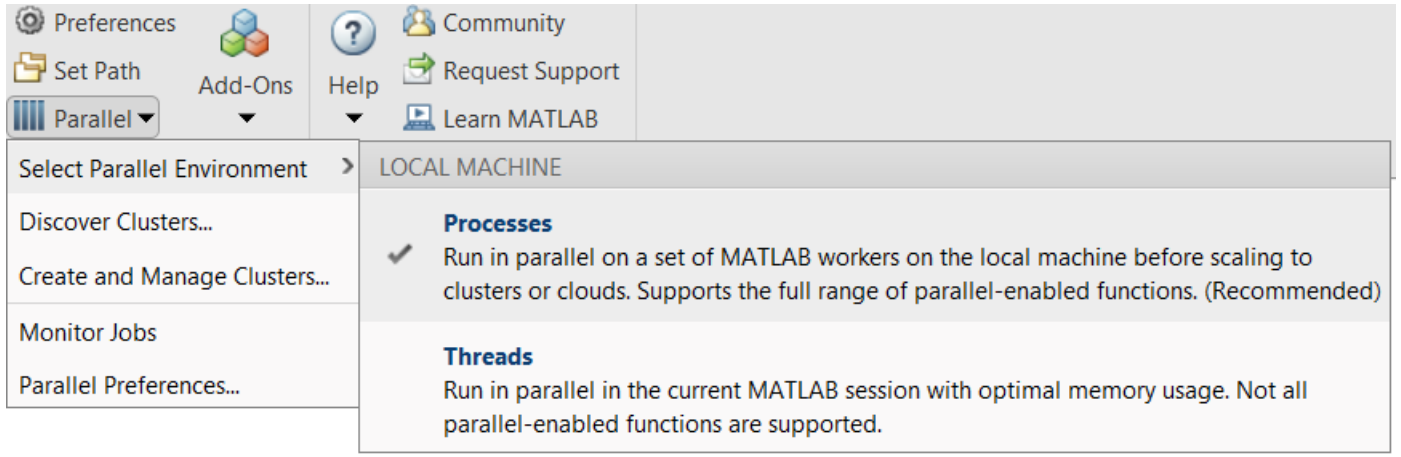
```
#SLURM …

module load matlab

matlab …
```

Known Issues

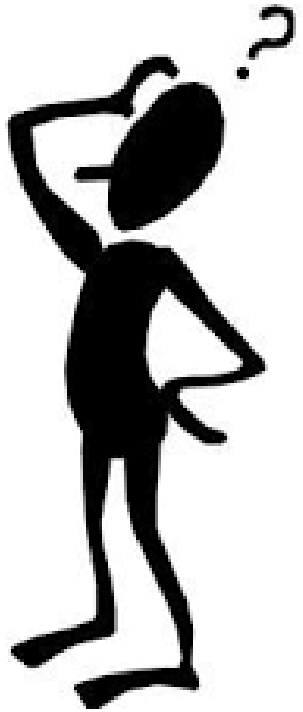
- Issue with large number of workers. Only exists in the following releases
  - R2021b: Workaround using Java file mirroring
  - R2022b: Fixed
- Issue with time stamp being out of sync with cluster.
  - R2022b: Fixed

Profiles



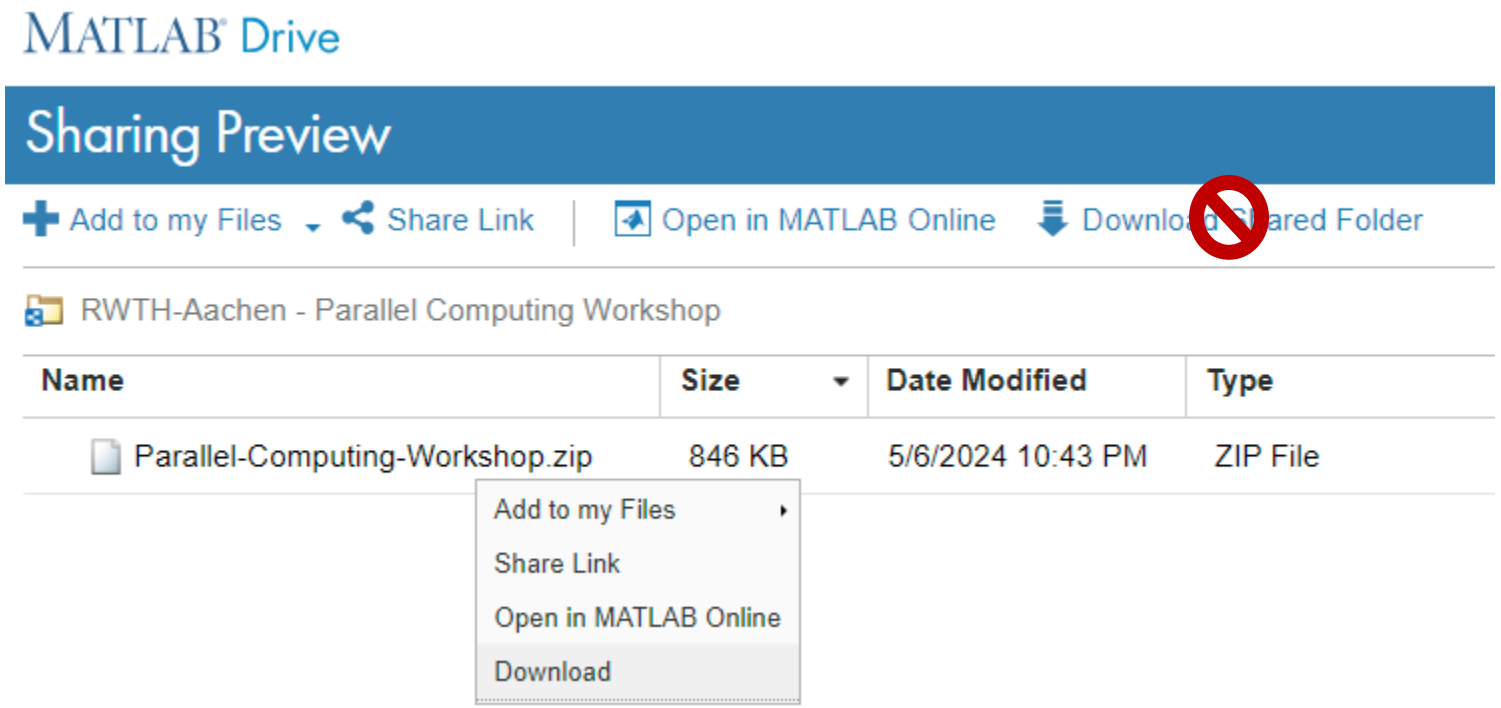


"How does MATLAB know about the CLAIX cluster?"

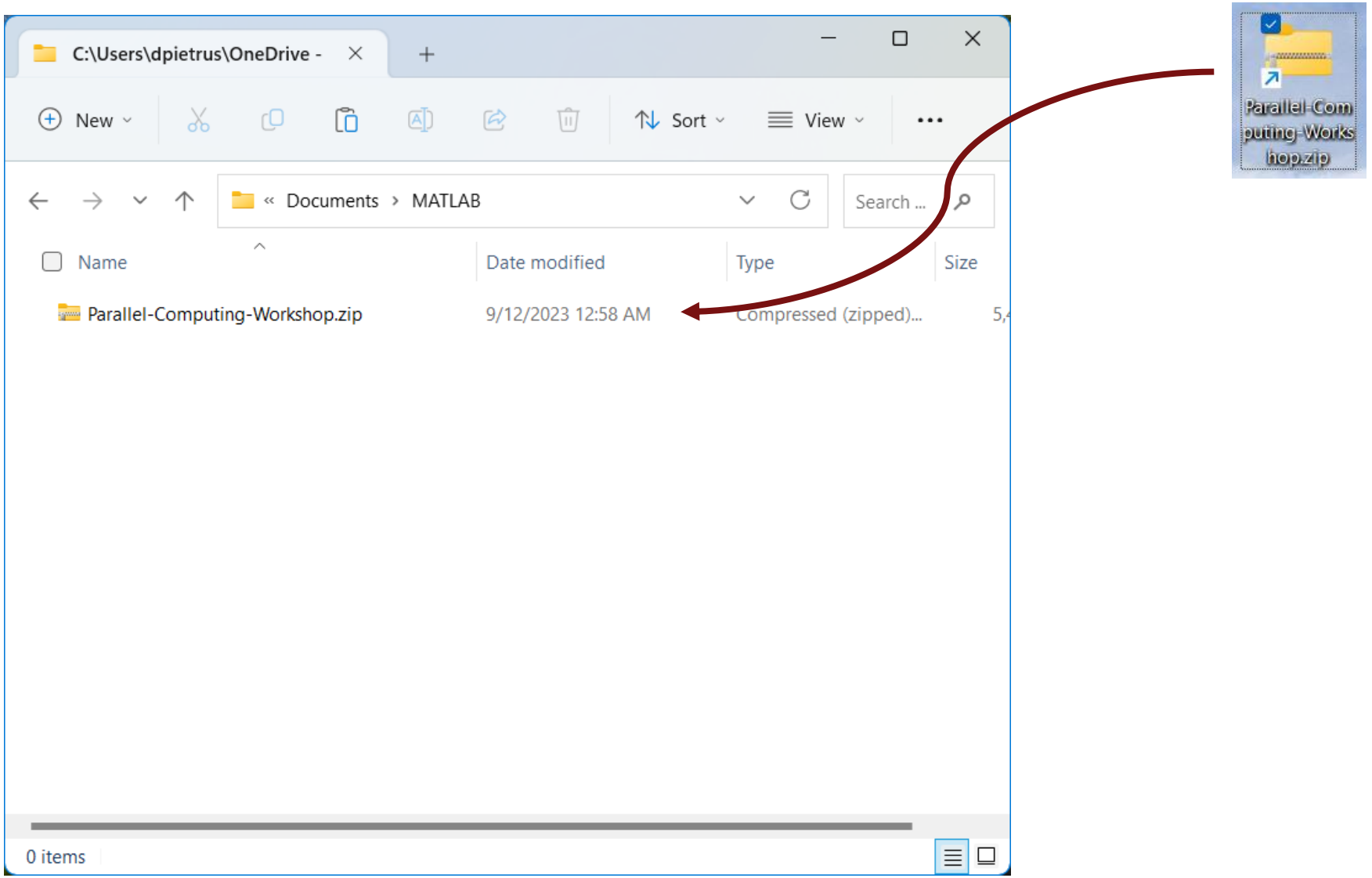


Download Instructions

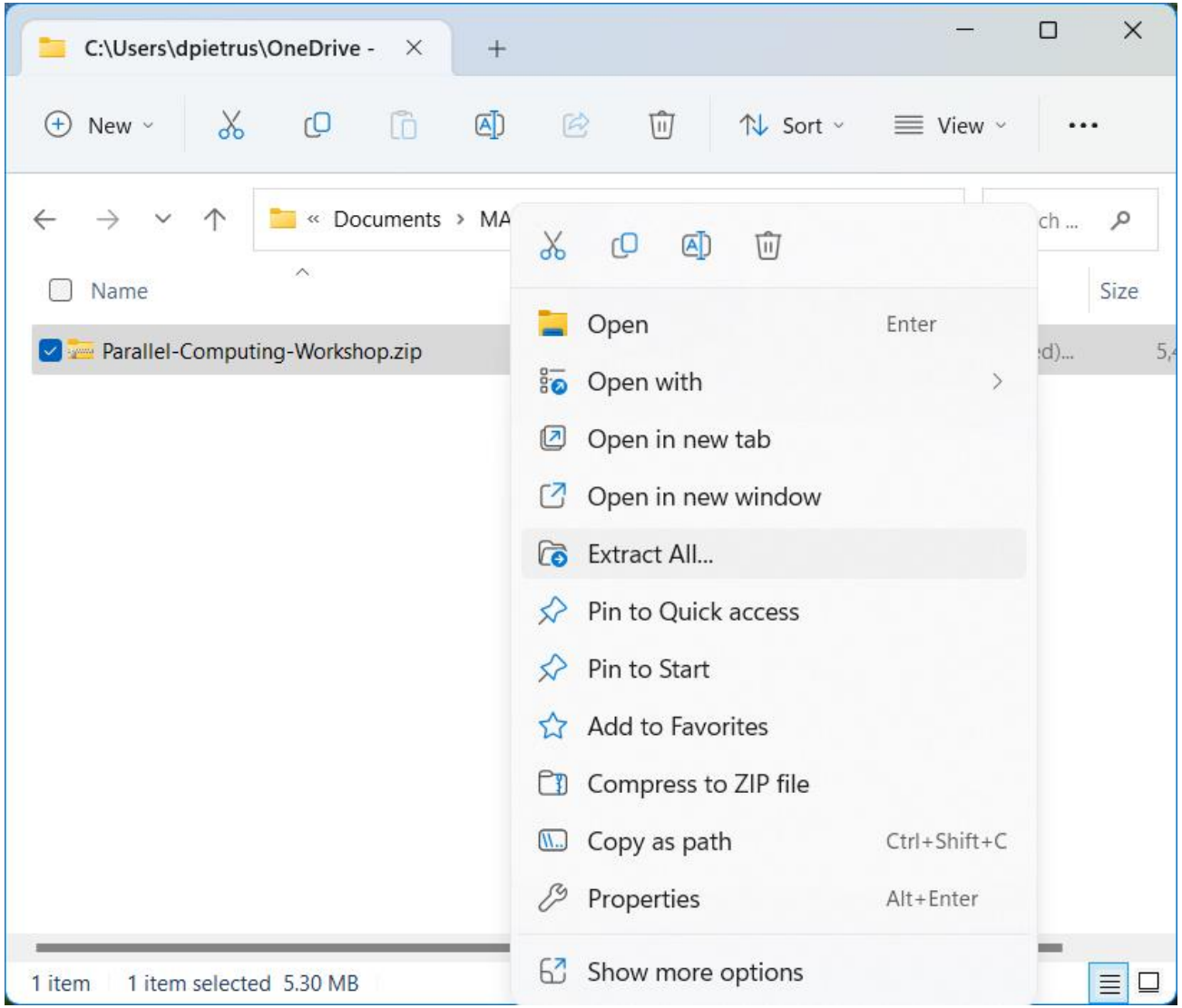
- https://tinyurl.com/RWTH-Aachen-PCT
- Right-click on file and select `Download`



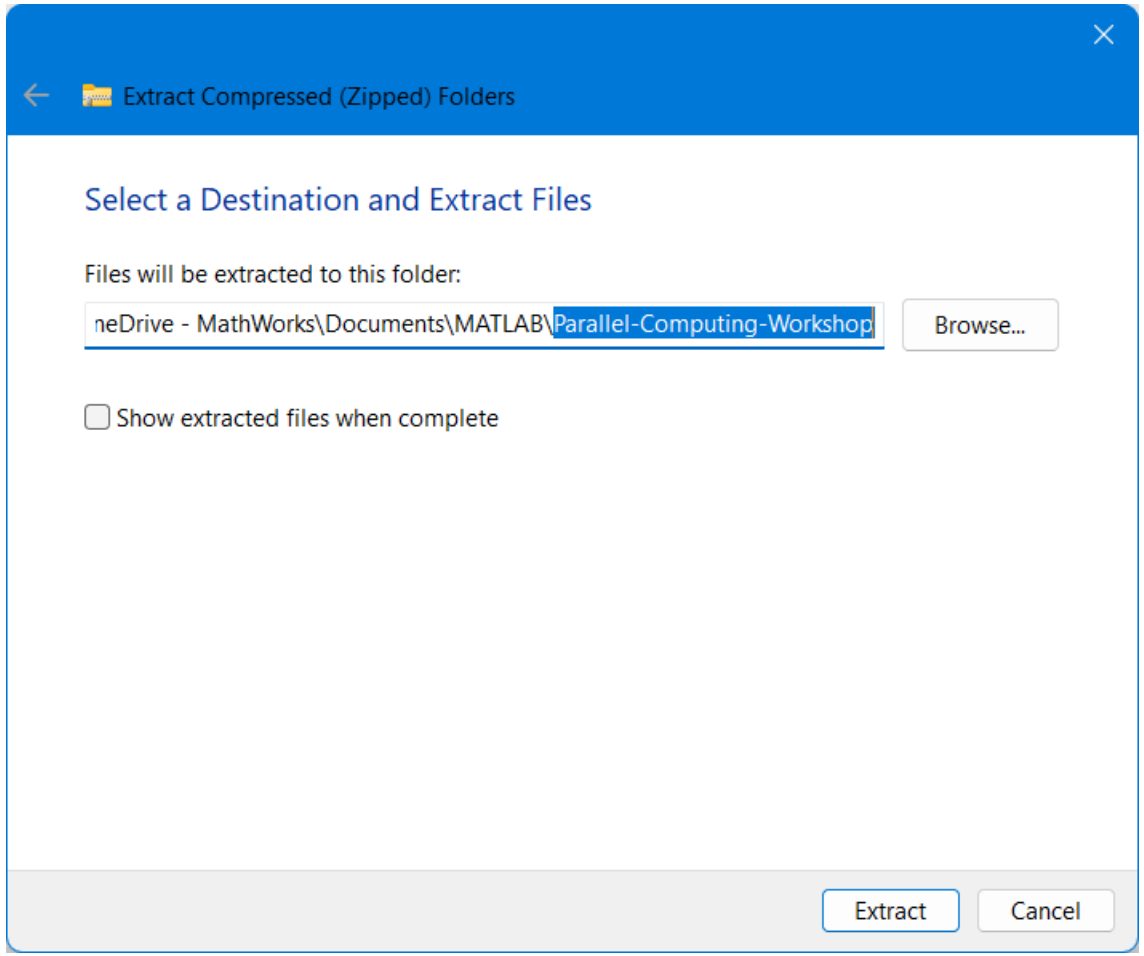
Install files – Windows

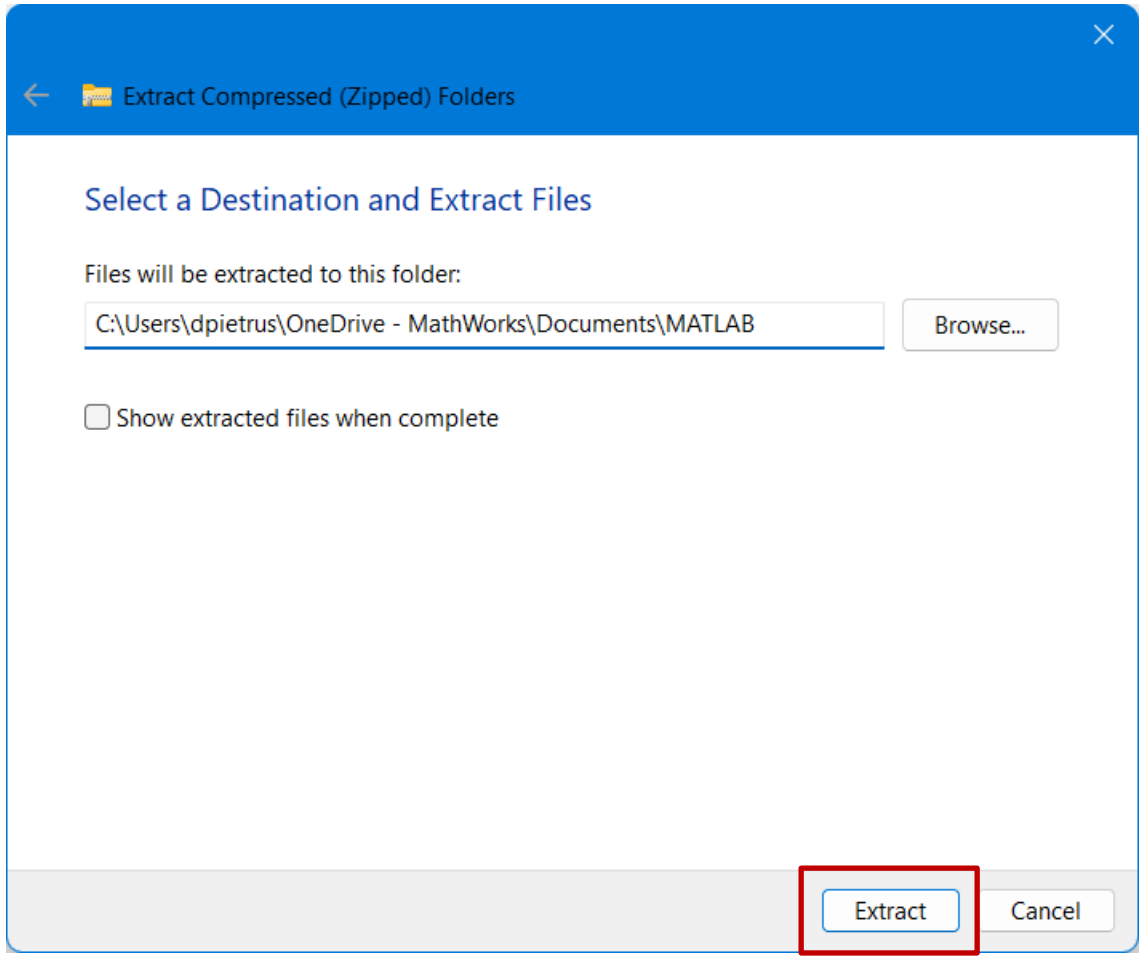


Install files – Windows



Install files – Windows



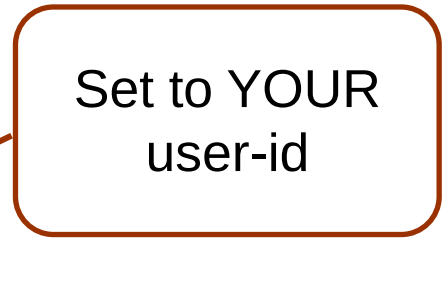


Install files – Linux/macOS

```
dpietrus@MATHWORKS-khjf3:~$ unzip -q ~/Parallel-Computing-Workshop.zip -d ~/Documents/MATLAB/
dpietrus@MATHWORKS-khjf3:~$ ls -1 ~/Documents/MATLAB/
+pctDebug
IntegrationScripts
Parallel-Computing-Workshop
claixDesktop.conf
cleanJobStorageLocation.m
clusterDefinition.m
configCluster.m
displayPoolError.m
jobStorageLocation.m
schedID.m
```

Configure MATLAB to create CLAIX profile

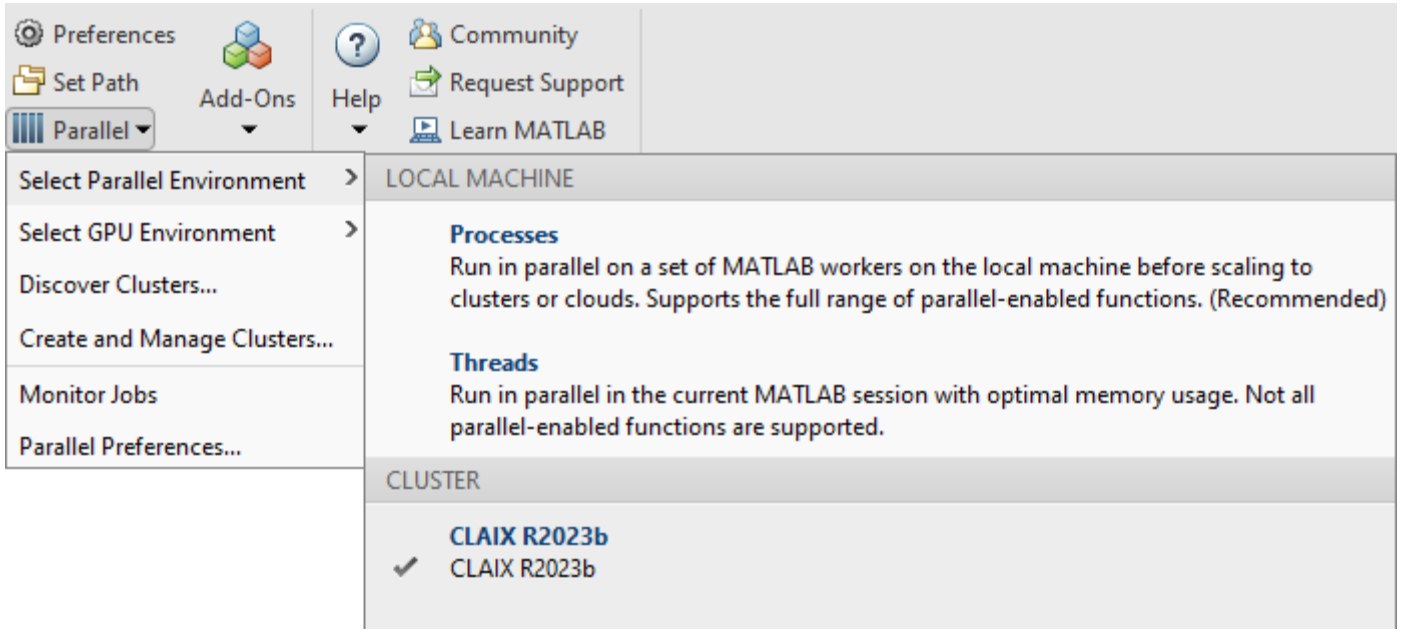
```
>> % Create a new CLAIX profile
>> configCluster
Username on CLAIX (e.g. jdoe): USER-ID
Complete.  Default cluster profile set to "CLAIX R2023b".
```



Set required job parameters

```
>> % Get handle to CLAIX cluster
>> c = parcluster;

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

    >> c = parcluster;
    >> % 5 hour, 30 minute walltime
    >> c.AdditionalProperties.WallTime = '05:30:00';
    >> c.saveProfile
>>
>> % Configure required job parameter
>> % 30 minute WallTime
>> c.AdditionalProperties.WallTime = '00:30:00';
>>
>> % Save changes to profile
>> c.saveProfile
```

New CLAIX profile



Only call `configCluster` once per version of MATLAB

MATLAB job submitters

- `parpool`
  - 1 single session at a time
  - Synchronous execution, therefore, MATLAB client continuously running
  - Directly runs `parfor`, `parfeval`, and `spmd`

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

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

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

Exercise: “Hello, World!”


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

```
#SBATCH …
module load matlab
matlab …
```

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

Exercise: “Hello, World!”

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

inputs to the function

additional job options

job submitter

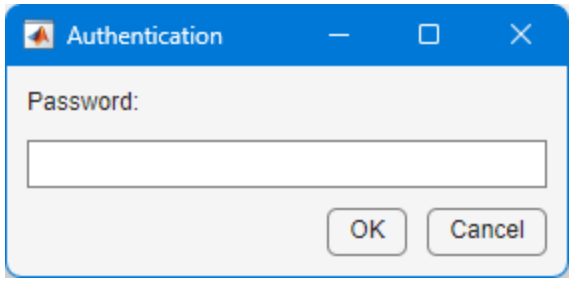
number of expected outputs from the function

job variable

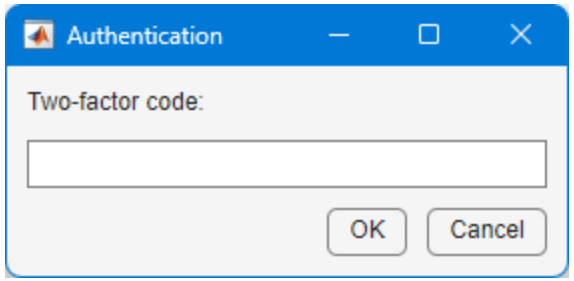
function to call

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

Multifactor Cluster Access



Enter password



Enter MFA code

Fetching results

```
>> % Submit job to cluster to find out where MATLAB is running
>> % Get handle to HPC cluster
>> c = parcluster;
>> job = c.batch(@pwd, 1, {}, 'CurrentFolder', '.');
Submit arguments: --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 =
    '/rwthfs/rz/cluster/home/lx516535'
```

Benign warning if *CurrentFolder* isn't set

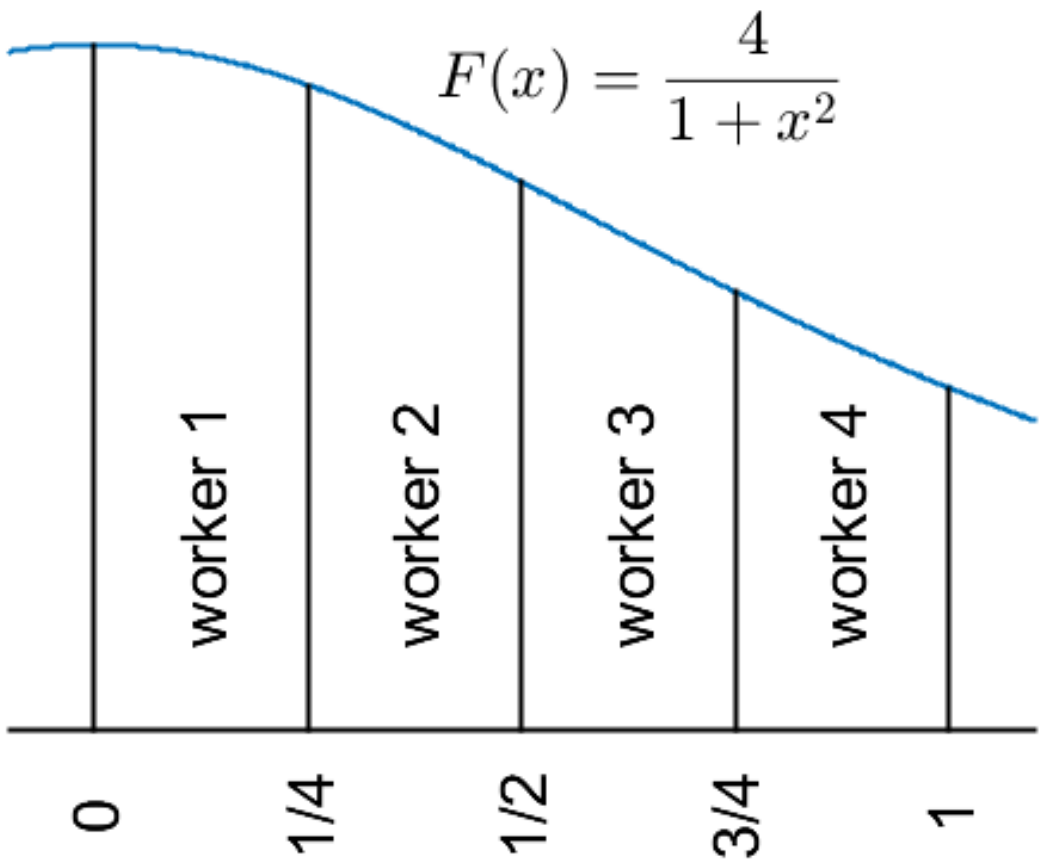
```
>> job.fetchOutputs{:}
Warning: The task with ID 1 issued the following warnings:
  Warning: Worker unable to change folder to 'C:\Users\dpietrus\OneDrive -
  MathWorks\Documents' at the start of the batch job. The job will be executed from
  '/rwthfs/rz/cluster/home/lx516535'. To execute from a different folder use the
  'CurrentFolder' parameter of batch. To suppress this warning, set 'CurrentFolder' to '.'.

ans =
    '/rwthfs/rz/cluster/home/lx516535'
```

Change directories to workshop

```
>> cd(fullfile(userpath,'Parallel-Computing-Workshop','Part-II'))
```

Exercise: Calculate π

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



Calculate π

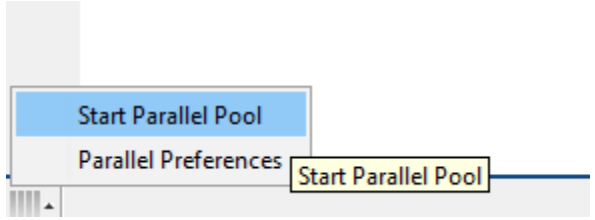
```
function calc_pi

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);
```

How should you start a local parallel pool?

- Call `parpool` from code or 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.
```

Interactive parallel pool is disabled for R2022b and older

```
>> pool = c.parpool(4);
Starting parallel pool (parpool) using the 'CLAIX R2023b' profile ...

Error using parallel.Cluster/parpool
Parallel pool failed to start with the following error. For more detailed information, validate the profile 'CLAIX R2023b' in the Cluster Profile
Manager.
Caused by:
    Error using parallel.internal.pool.AbstractInteractiveClient>iThrowWithCause
    Failed to start pool.
        Error using parallel.Job/submit
        Job submission failed because the plugin function 'communicatingSubmitFcn.m' errored.
            Error using displayPoolError


            ************************************************
            CLAIX R2023b does not support calling

                >> parpool('CLAIX R2023b',4)

            Instead, use batch()

                >> job = batch(...,'pool',4);

            Call

                >> doc batch

            for more help on using batch.
            ************************************************
```

Interactive parallel pool can error in R2023a and newer

```
>> pool = c.parpool(4);
Starting parallel pool (parpool) using the 'CLAIX R2023b' profile ...
Submit arguments: --ntasks=4 --cpus-per-task=1 --ntasks-per-core=1 --mem-per-cpu=3900mb -t 00:30:00
Parallel pool using the 'CLAIX R2023b' profile is shutting down.

Error using parallel.Cluster/parpool
Parallel pool failed to start with the following error. For more detailed information, validate the profile 'CLAIX
R2023b' in the Cluster Profile Manager.
Caused by:
    Error using parallel.internal.pool.AbstractInteractiveClient>iThrowWithCause
    Failed to initialize the interactive session.
        Error using parallel.internal.pool.SpfClientSession/addClientEndpointsAndConnect
        An unexpected error occurred accessing a parallel pool. The underlying error was: Failed to connect to
        endpoint after 5 attempts.
        Cause: Attempt 2
        Unable to connect to tcp://tcpnodelay=ncm0971:27370/protocol/catapult
        because: Unable to connect to: tcp://tcpnodelay=ncm0971:27370/protocol/catapult
         Error: resolve: No such host is known [system:11001].
```

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

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

Submit `calc_pi` job

```
>> % Submit calc_pi job
>> c = parcluster;
>>
>> % Request 16 workers
>> job = c.batch(@calc_pi, 0, {}, 'CurrentFolder', '.', 'Pool', 16);
Submit arguments: --ntasks=17 --cpus-per-task=1 --ntasks-per-core=1
```

"If my Pool is size 16, why am I requesting 17 tasks?"

Fetch the results

```
>> % Submit calc_pi job
>> c = parcluster;
>>
>> % Request 16 workers
>> job = c.batch(@calc_pi, 0, {}, 'CurrentFolder', '.', 'Pool', 16);
Submit arguments: --ntasks=17 --cpus-per-task=1 --ntasks-per-core=1
>>
>> % Check the state of the job
>> job.State
ans =
    'finished'
>>
>> % Fetch the results
>> job.fetchOutputs{:}
>>
```

"Where's the output?"

What gets “returned”?

- 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

```
>> job = c.batch(@test_fcn, 1, {300}, 'CurrentFolder', '.', 'Pool', 10);
Submit arguments: --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
>> job.fetchOutputs{:}
ans =
   15.7717
>>
```

"Where's *A*?"

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

ans =

     4


ans =

     5


ans =

     8
```

Sample Diary – calc_pi_with_spmd

```
>> job.diary
--- Start Diary ---
Worker  1:
  Subinterval: [0    , 0.0625]
Worker  3:
  Subinterval: [0.125, 0.1875]
Worker  4:
  Subinterval: [0.1875, 0.25]
Worker  7:
  Subinterval: [0.375, 0.4375]
Worker  9:
  Subinterval: [0.5 , 0.5625]
Worker 10:
  Subinterval: [0.5625, 0.625]
Worker 11:
  Subinterval: [0.625, 0.6875]
Worker 12:
  Subinterval: [0.6875, 0.75]
Worker 14:
  Subinterval: [0.8125, 0.875]
Worker  9:
  Subinterval: [0.5 , 0.5625]   Integral: 0.194967
Worker 10:
  Subinterval: [0.5625, 0.625]   Integral: 0.184839
Worker 11:
  Subinterval: [0.625, 0.6875]   Integral: 0.174752
Worker 12:
  Subinterval: [0.6875, 0.75]   Integral: 0.164855
Worker 13:
  Subinterval: [0.75, 0.8125]   Integral: 0.155262
Worker 14:
  Subinterval: [0.8125, 0.875]   Integral: 0.146054
Worker 15:
  Subinterval: [0.875, 0.9375]   Integral: 0.137285
Worker 16:
  Subinterval: [0.9375, 1   ]   Integral: 0.128988
pi            : 3.141592653589793116
Approximation: 3.141592653589793116
Error         : 0

--- End Diary ---
>>
```

Save files

"Where does RESULTS get written to?"

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

disp('Start sim')

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

disp('Finished')

save RESULTS A
```

"Is there a way we could pull the file back to our machine?"

FileStore (1)

```
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
store = getCurrentFileStore;
key = 'RESULTS';
copyFileToStore(store, 'RESULTS.mat', key)
```

```
function bring_back_file_from_cluster(store, key)

% Filename (to write serialize data to)
destination = strcat(key,'.mat');

% Create local file
copyFileFromStore(store, key, destination)
```

Same file

"look-up key"

FileStore (2)

```
>> job = c.batch(@test_fcn, 1, {300}, 'CurrentFolder', '.', 'Pool', 10);
Submit arguments: --ntasks=11 --cpus-per-task=1 --ntasks-per-core=1
>>
>> % Create client-side file store
>> store = job.FileStore;
>>
>> % Assign callback when file is added to the store
>> store.KeyUpdatedFcn = @bring_back_file_from_cluster;
>>
>> % Wait for the job to finish
>> job.wait
>>
>> % Load data from copied over file
>> load RESULTS
>> whos A
  Name        Size              Bytes  Class     Attributes

  A         300x1               2400  double
```

“Who needs threads?...”

```
function t = threads_example(N)

matrix = rand(N);

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

end
```

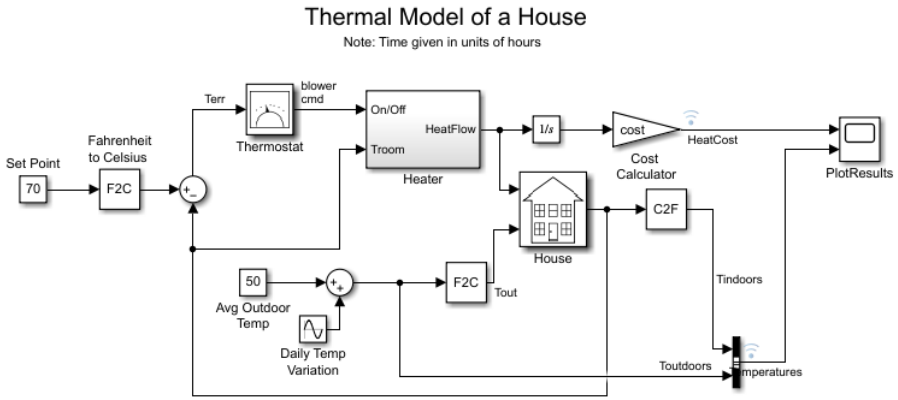
Single threaded vs multi-threaded

```
>> c.AdditionalProperties.MemPerCPU = '8GB';
>>
>> % Run on a single thread (core)
>> job_01 = c.batch(@threads_example, 1, {2^14}, 'CurrentFolder', '.');
Submit arguments: --ntasks=1 --cpus-per-task=1 --ntasks-per-core=1 --mem-per-cpu=8GB
>>
>> % Time to run on a single thread
>> t_01 = job_01.fetchOutputs{:}
t_01 =
    3.4860
>>
>> % Allocate 20 threads
>> c.NumThreads = 20;
>> job_20 = c.batch(@threads_example, 1, {2^14}, 'CurrentFolder', '.');
Submit arguments: --ntasks=1 --cpus-per-task=20 --ntasks-per-core=1 --mem-per-cpu=8GB
>>
>> % Time to run with 20 threads
>> t_20 = job_20.fetchOutputs{:}
t_20 =
    0.3015
```

`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;
Submit arguments: --ntasks=7 --cpus-per-task=1 --ntasks-per-core=1 --mem-per-cpu=4GB
>> job.diary
--- Start Diary ---
[14-Mar-2024 12:02:15] Checking for availability of parallel pool...
[14-Mar-2024 12:02:15] Starting Simulink on parallel workers...
[14-Mar-2024 12:02:34] Configuring simulation cache folder on parallel workers...
[14-Mar-2024 12:02:34] Loading model on parallel workers...
[14-Mar-2024 12:02:56] Running simulations...
[14-Mar-2024 12:03:02] Completed 1 of 6 simulation runs
[14-Mar-2024 12:03:02] Completed 2 of 6 simulation runs
[14-Mar-2024 12:03:02] Completed 3 of 6 simulation runs
[14-Mar-2024 12:03:02] Completed 4 of 6 simulation runs
[14-Mar-2024 12:03:02] Completed 5 of 6 simulation runs
[14-Mar-2024 12:03:02] Completed 6 of 6 simulation runs
[14-Mar-2024 12:03:02] Cleaning up parallel workers...
```



Other settable job properties

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

     AdditionalShellCommands: ''
        AdditionalSubmitArgs: ''
          AuthenticationMode: 'Multifactor'
                 ClusterHost: 'login18-1.hpc.itc.rwth-aachen.de'
                  Constraint: ''
           DisplaySubmitArgs: 1
                EmailAddress: ''
                 EnableDebug: 0
                     GPUCard: ''
                 GPUsPerNode: 0
                   MemPerCPU: '4GB'
                   Partition: ''
                ProcsPerNode: 0
                 ProjectName: ''
    RemoteJobStorageLocation: ''
        RequireExclusiveNode: 0
                 Reservation: ''
                    Username: ''
                    WallTime: ''
```

GPUs

gpuDevice

```
>> % Get info on GPU device
>> c.AdditionalProperties.GPUsPerNode = 1;
>> job = c.batch(@gpuDevice, 1, {}, 'CurrentFolder', '.');
Submit arguments: --ntasks=1 --cpus-per-task=1 --ntasks-per-core=1
>>
>> % Fetch results
>> device = job.fetchOutputs{:}
device =
  CUDADevice with properties:

                      Name: 'Tesla V100-SXM2-16GB'
                     Index: 1
         ComputeCapability: '7.0'
```

gpuDevice - macOS

```
>> % Get info on GPU device
>> c.AdditionalProperties.GPUsPerNode = 1;
>> job = c.batch(@gpuDevice, 1, {}, 'CurrentFolder', '.', 'AutoAddClientPath', false);
Submit arguments: --ntasks=1 --cpus-per-task=1 --ntasks-per-core=1 --mem-per-cpu=4gb --gres=gpu:1
>>
>> % Get results
>> device = job.fetchOutputs{:}
Warning: While loading an object of class 'CUDADevice':
GPU acceleration with Parallel Computing Toolbox is not supported on macOS.
```

gpuDeviceTable - macOS

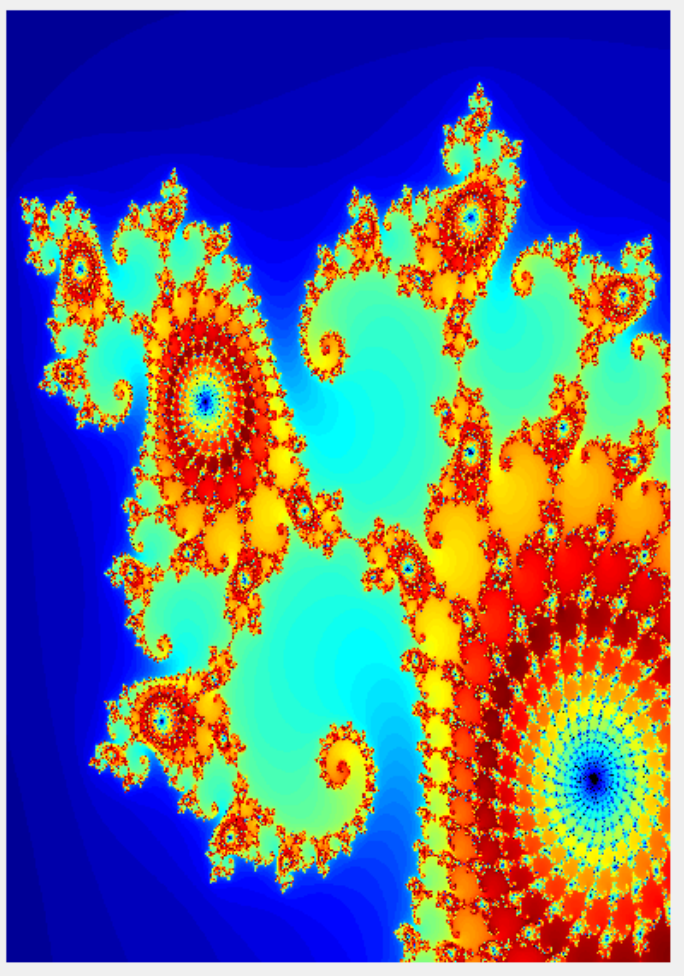
```
>> % Get GPU device table
>> c.AdditionalProperties.GPUsPerNode = 1;
>> job = c.batch(@gpuDeviceTable, 1, {}, 'CurrentFolder', '.');
Submit arguments: --ntasks=1 --cpus-per-task=1 --ntasks-per-core=1 --mem-per-cpu=3900mb
>>
>> % Get GPU device table
>> deviceTable = job.fetchOutputs{:}
deviceTable =
  1×5 table
    Index              Name              ComputeCapability    DeviceAvailable    DeviceSelected
    _____    ____________________________    _________________    _______________    ______________
      1      "Tesla V100-SXM2-16GB"              "7.0"                true              false
```

Tesla V100

```
>> % Get info on GPU device
>> c.AdditionalProperties.GPUsPerNode = 1;
>> job = c.batch(@gpuDevice, 1, {}, 'CurrentFolder', '.');
Submit arguments: --ntasks=1 --cpus-per-task=1 --ntasks-per-core=1
>>
>> % Fetch results
>> device = job.fetchOutputs{:};
>>
>> % GPU card
>> device.Name
ans =
    'Tesla V100-SXM2-16GB'
>>
>> % Calculate total memory (GB) on GPU
>> round(device.TotalMemory/1024^3)
ans =
    16
```

Example: 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,'gpuArray')
    x = gpuArray.linspace(xlim(1),xlim(2),gridSize);
    y = gpuArray.linspace(ylim(1),ylim(2),gridSize);
else
    x = linspace(xlim(1),xlim(2),gridSize);
    y = linspace(ylim(1),ylim(2),gridSize);
end

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

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

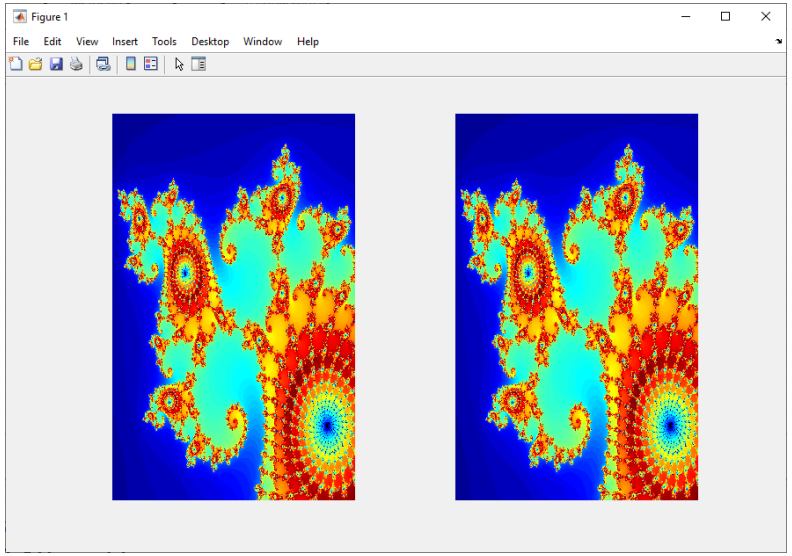


Example: mandelbrot (2)

```
function mandelbrot_example

% Get handle to cluster
c = parcluster;

% Run on CPU
c.NumThreads = 24;
cpu_j = c.batch(@calc_mandelbrot,4,{'double'},'AutoAddClientPath',false);

% Run on GPU
c.NumThreads = 1;
c.AdditionalProperties.GPUsPerNode = 1;
gpu_j = c.batch(@calc_mandelbrot,4,{'gpuArray'},'AutoAddClientPath',false);

% Plot results
% Wait for the job to finish running
cpu_j.wait
[cpu_x,cpu_y,cpu_count,cpu_t] = cpu_j.fetchOutputs{:};
figure
subplot(1,2,1)
imagesc(cpu_x,cpu_y,cpu_count)
colormap([jet; flipud(jet); 0 0 0]);
axis off

% Wait for the job to finish running
gpu_j.wait
[gpu_x,gpu_y,gpu_count,gpu_t] = gpu_j.fetchOutputs{:};
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)
```

```
>> calc_mandelbrot
```

Example: mandelbrot (3)




```
>> mandelbrot_example
Submit arguments: --ntasks=1 --cpus-per-task=24
Submit arguments: --ntasks=1 --cpus-per-task=1
CPU time: 20.46
GPU time: 1.56
```

```
CPU time: 370.51
GPU time: 13.69
```

```
>> mandelbrot_example
```

GPU benchmarking with FFT

```
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)])
```

Matrix sizes

```
>> % Size of matrix (e.g., rand(N)) in GB
>> sz = 2^16 * 2^16 * 8 / 1024^3
sz =
    32
>> sz = 2^15 * 2^15 * 8 / 1024^3
sz =
     8
>> sz = 2^14 * 2^14 * 8 / 1024^3
sz =
     2
>> sz = 2^13 * 2^13 * 8 / 1024^3
sz =
    0.5000
```

FFT 65536 x 65536

```
>> c.AdditionalProperties.MemPerCPU = '40GB';
>> c.AdditionalProperties.GPUsPerNode = 1;
>> job = c.batch(@calc_fft_cpu_gpu, 2, {2^16}, 'CurrentFolder', '.');
Submit arguments: --ntasks=1 --cpus-per-task=1 --ntasks-per-core=1 --mem-per-cpu=40GB -t 00:30:00 --gres=gpu:1
>>
>> % Wait for job to finish
>> job.wait
>>
>> % Fetch results
>> [cpu_t, gpu_t] = job.fetchOutputs{:}
Error using parallel.Job/fetchOutputs
Outputs can only be fetched if the job is in State 'finished'.
>>
>> % Check the state of the job
>> job.State
ans =
    'failed'
>>
>> job.diary
--- Start Diary ---
--- End Diary ---
>>
>> c.getDebugLog(job.Tasks)
LOG FILE OUTPUT:
The scheduler has allocated the following node to this job: nrg01.hpc.itc.rwth-aachen.de
Executing: "/cvmfs/software.hpc.rwth.de/Linux/RH8/x86_64/ISV/MATLAB/2023b/bin/worker"
Picked up JAVA_TOOL_OPTIONS: -Xmx2048m
/home/lx516535/.matlab/generic_cluster_jobs/claix/MATHWORKS-KHJF3/Job49/independentJobWrapper.sh
Exiting with code: 137
slurmstepd: error: Detected 1 oom-kill event(s) in StepId=45220226.batch.
Some of your processes may have been killed by the cgroup out-of-memory handler.
```

FFT 65536 x 65536

```
>> % Increase memory request
>> c.AdditionalProperties.MemPerCPU = '120GB';
>> job = c.batch(@calc_fft_cpu_gpu, 2, {2^16}, 'CurrentFolder', '.');
Submit arguments: --ntasks=1 --cpus-per-task=1 --ntasks-per-core=1 --mem-per-cpu=120GB -t 00:30:00 --gres=gpu:1
>>
>> % Wait for job to finish
>> job.wait
>>
>> % Fetch results
>> [cpu_t, gpu_t] = job.fetchOutputs{:}
Error using parallel.Job/fetchOutputs
An error occurred during execution of Task with ID 1.
Caused by:
    Error using calc_fft_cpu_gpu
    Maximum variable size allowed on the device is exceeded.
>>
>> % Check the state of the job
>> job.State
ans =
    'finished'
>>
>> job.diary
--- Start Diary ---
Total time on CPU: 57.2947

--- End Diary ---
>>
>> c.getDebugLog(job.Tasks)
LOG FILE OUTPUT:
The scheduler has allocated the following node to this job: ncg03.hpc.itc.rwth-aachen.de
Executing: "/cvmfs/software.hpc.rwth.de/Linux/RH8/x86_64/ISV/MATLAB/2023b/bin/worker"
Picked up JAVA_TOOL_OPTIONS: -Xmx2048m
Exiting with code: 0
```

FFT 16384 x 16384

```
>> c.AdditionalProperties.MemPerCPU = '120GB';
>> job = c.batch(@calc_fft_cpu_gpu, 2, {2^14}, 'CurrentFolder', '.');
Submit arguments: --ntasks=1 --cpus-per-task=1 --ntasks-per-core=1 --mem-per-cpu=120GB
>>
>> % Wait for job to finish
>> job.wait
>>
>> % Fetch results
>> [cpu_t, gpu_t] = job.fetchOutputs{:}
cpu_t =
    2.0382
gpu_t =
    3.3994
>>
>> % Check the job diary
>> job.diary
--- Start Diary ---
Total time on CPU: 2.0382
GPU FFT: 0.040656
Total time on GPU: 3.3994
FFT speed improvement: 50.1328
Total speed improvement: 0.59957
```

Unset GPU resources when done

```
>> % Unset GPU when no longer needed
>> c.AdditionalProperties.GPUsPerNode = 0;
```

Submitting scripts, instead of functions

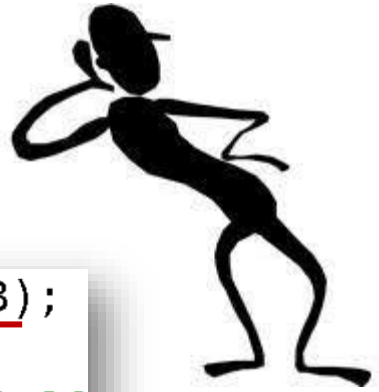
```
>> x = 4;
>> z = rand(3);
>>
>> % Submit a script (instead of a function)
>> job = c.batch('temp = rand(10); y = x; who', 'CurrentFolder', '.');
Submit arguments: --ntasks=1 --cpus-per-task=1 --ntasks-per-core=1
>> clear z
>>
>> who

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
```

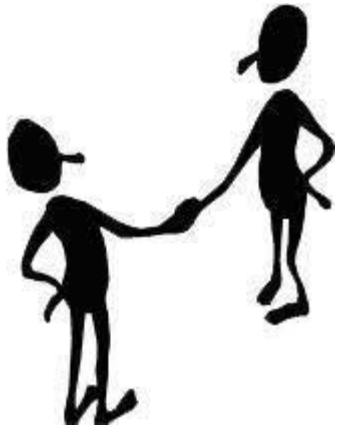
The cost of transferring unnecessary data

```
>> x = 4;
>> tic, job = c.batch('temp = rand(10); y = x; who', 'CurrentFolder', '.'); toc
Submit arguments: --ntasks=1 --cpus-per-task=1 --ntasks-per-core=1 --mem-per-cpu=3900mb
Elapsed time is 54.832436 seconds.
>>
>> z = rand(1000);      % 8 MB
>> tic, job = c.batch('temp = rand(10); y = x; who', 'CurrentFolder', '.'); toc
Submit arguments: --ntasks=1 --cpus-per-task=1 --ntasks-per-core=1 --mem-per-cpu=3900mb
Elapsed time is 105.693532 seconds.
>>
>> z = rand(1700);      % 22 MB
>> tic, job = c.batch('temp = rand(10); y = x; who', 'CurrentFolder', '.'); toc
Submit arguments: --ntasks=1 --cpus-per-task=1 --ntasks-per-core=1 --mem-per-cpu=3900mb
Elapsed time is 232.747192 seconds.
```

*Connected over VPN – submitting from Massachusetts to Aachen

Adding files to the job (1)

- AutoAddClientPath
  - Set to false if you have added your own local paths to the MATLAB client path
- AutoAttachFiles
  - Useful for small number, often changing files
- AttachedFiles
  - List files not automatically added to the job (e.g., binary files)
- AdditionalPaths
  - List absolute paths on the cluster (include subdirectories if needed)



Adding files to the job (2)

- AttachedFiles
  - Files or folders to be transferred from your client to the workers
  - For small files not already staged on the cluster

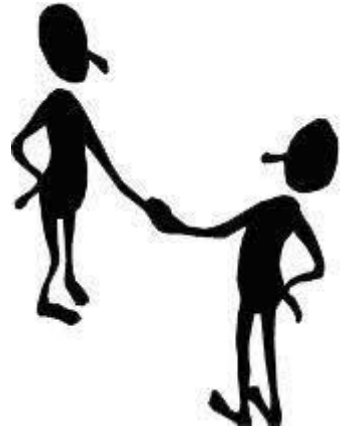
```
>> % Attach files
>> job = c.batch(@myJob, 4, {}, 'CurrentFolder', '.', 'Pool', 10, ...
         'AttachedFiles', 'C:\Work\MATLAB\ProjectHelperFiles');
```

Adding files to the job (3)

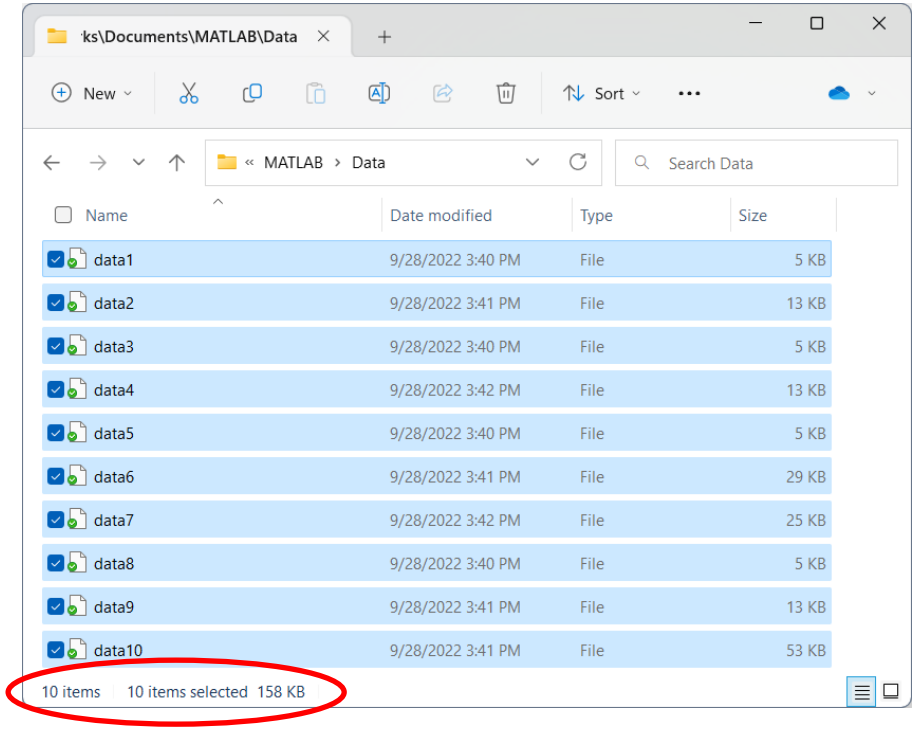
- AdditionalPaths
  - Paths added to the MATLAB search path of the workers before the script or function executes
  - Used to make data already staged on the cluster ‘visible’ to the workers

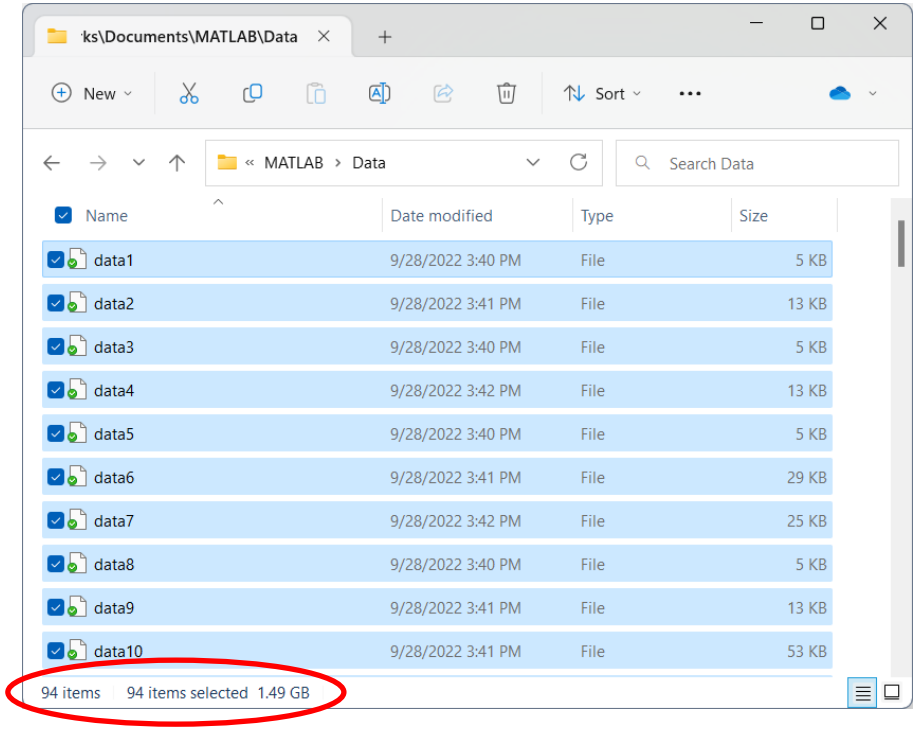
```
>> % Add remote folders
>> job = c.batch(@myJob, 4, {}, 'CurrentFolder', '.', 'Pool', 10, ...
        'AdditionalPaths', '/home/lx516535/matlab/project_files');
```

Should I send files with the job?



- By default, each job will copy all required files
  - How many jobs are you going to submit?
  - How large, in totality, are your files?
  - Do your files change a lot?



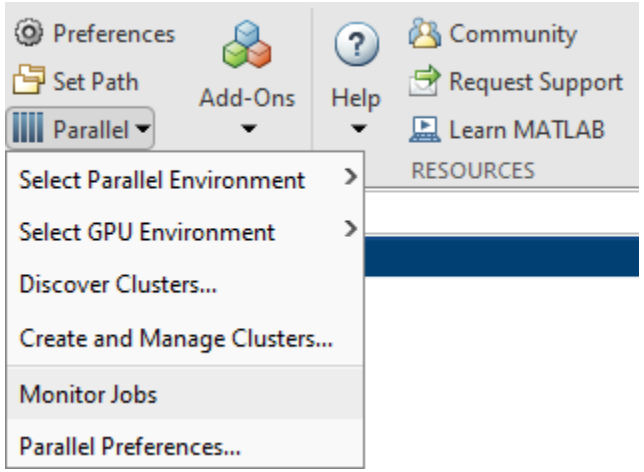


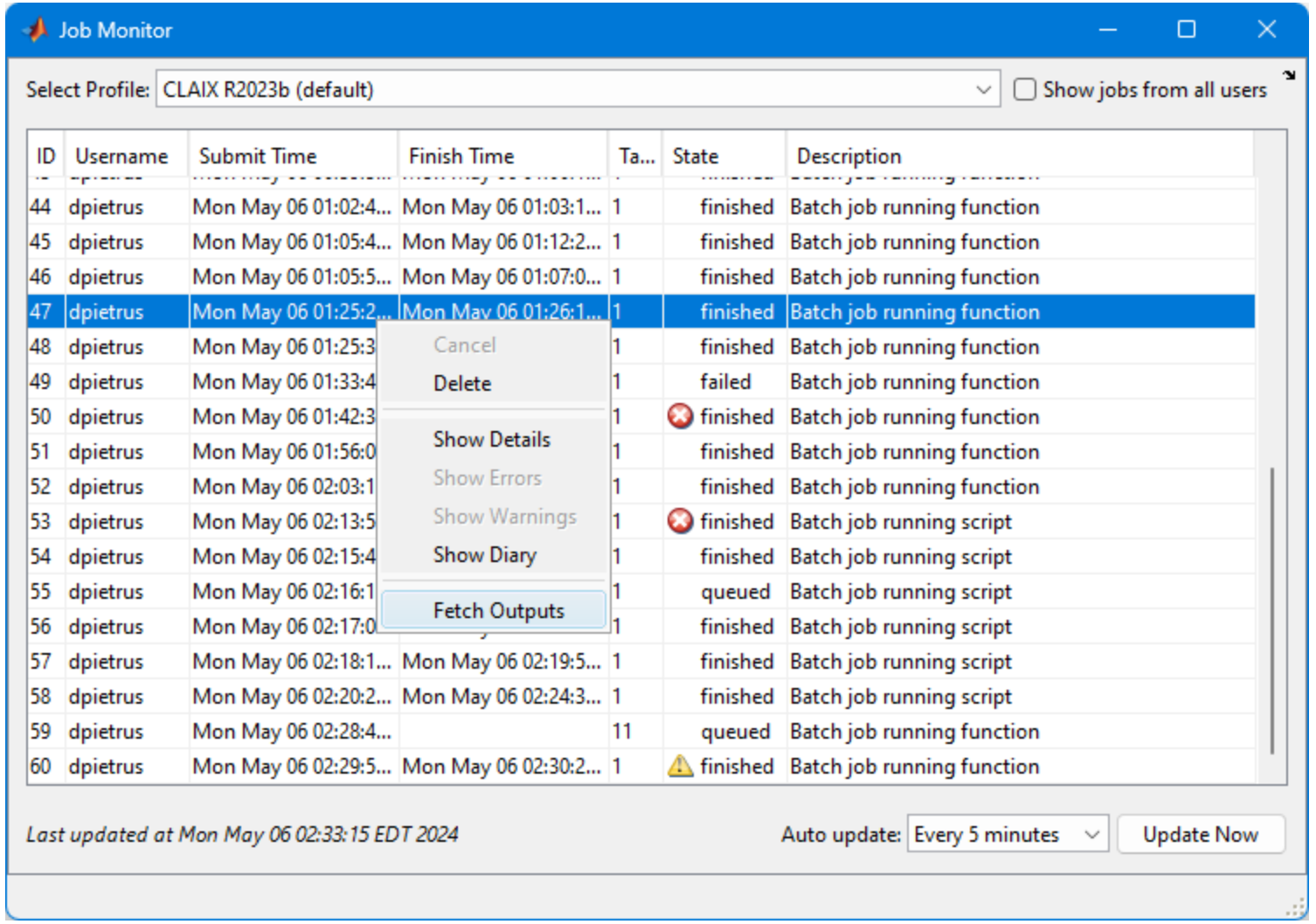
When has my job run and finished?

```
>> % Get email notification when my job has finished
>> c.AdditionalProperties.EmailAddress = 'user-id@rwth-aachen.de';
>> job = c.batch(@test_fcn, 1, {300}, 'CurrentFolder', '.', 'Pool', 10);
Submit arguments: --ntasks=11 --cpus-per-task=1 --ntasks-per-core=1
```

```
--mail-user=user-id@rwth-aachen.de
```

Retrieving past jobs





Keep cluster clean: delete jobs

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

```
>> % Finished with the job, delete it to cleanup jobs
>> job.delete
```

Debugging and Troubleshooting

Be aware of version support…

```
>> job = c.batch(@pwd, 1, {}, 'CurrentFolder', '.');
Submit arguments: --ntasks=1 --cpus-per-task=1 --ntasks-per-core=1 --mem-per-cpu=3900mb -t 00:30:00
>>
>> job.wait
>>
>> % 3 minutes later...
>>
>> job.State
ans =
    'failed'
>>
>> % Check log file
>> c.getDebugLog(job.Tasks)
LOG FILE OUTPUT:
The scheduler has allocated the following node to this job: nrm040.hpc.itc.rwth-aachen.de
Executing: "/cvmfs/software.hpc.rwth.de/Linux/RH8/x86_64/ISV/MATLAB/2020a/bin/worker"
/home/lx516535/.matlab/generic_cluster_jobs/claix/MATHWORKS-KHJF3/Job62/independentJobWrapper.sh line 36:
/cvmfs/software.hpc.rwth.de/Linux/RH8/x86_64/ISV/MATLAB/2020a/bin/worker: No such file or directory
Exiting with code: 127
```

Example: Errored jobs (1)

```
>> % Undefined function
>> job = c.batch(@invalid_fcn, 1, {}, 'CurrentFolder', '.');
Submit arguments: --ntasks=1 --cpus-per-task=1 --ntasks-per-core=1
>>
>> % Undefined variable in a script
>> job2 = c.batch('x = a', 'CurrentFolder', '.');
Submit arguments: --ntasks=1 --cpus-per-task=1 --ntasks-per-core=1
```

Example: Errored jobs (2)

```
>> % Undefined function
>> job.State
ans =
    'finished'
>>
>> 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'.
```

Example: Errored jobs (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 'x = 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}, 'CurrentFolder', '.', 'Pool', 0);
Submit arguments: --ntasks=1 --cpus-per-task=1 --ntasks-per-core=1 --mem-per-cpu=3900mb -t 00:30:00
>>
>> % Retreive log file for single worker job
>> c.getDebugLog(job.Tasks)
LOG FILE OUTPUT:
The scheduler has allocated the following node to this job: nrm102.hpc.itc.rwth-aachen.de
Executing: "/cvmfs/software.hpc.rwth.de/Linux/RH8/x86_64/ISV/MATLAB/2023b/bin/worker"
Picked up JAVA_TOOL_OPTIONS: -Xmx2048m
Exiting with code: 0
```

Logfile: Multi-core job

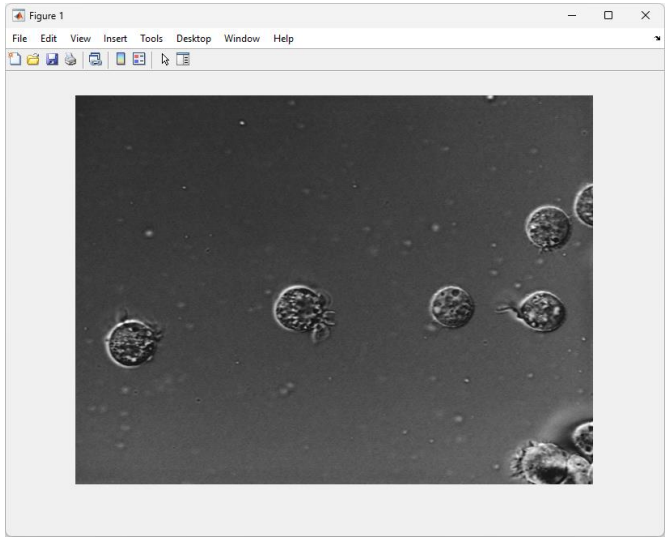
```
>> job = c.batch(@test_fcn, 1, {300}, 'CurrentFolder', '.', 'Pool', 10);
Submit arguments: --ntasks=11 --cpus-per-task=1 --ntasks-per-core=1 --mem-per-cpu=3900mb -t 00:30:00
>>
>> % Retreive log file for multiple worker job
>> c.getDebugLog(job)
LOG FILE OUTPUT:
The scheduler has allocated the following nodes to this job:
ncm[0306,1015]
"srun" -l -n 11 "/cvmfs/software.hpc.rwth.de/Linux/RH8/x86_64/ISV/MATLAB/2023b/bin/worker" -parallel
 0: Picked up JAVA_TOOL_OPTIONS: -Xmx2048m
 0: Parallel pool is shutting down.
Exiting with code: 0
```

Scheduler ID

```
>> job = c.batch(@test_fcn, 1, {300}, 'CurrentFolder', '.', 'Pool', 10);
Submit arguments: --ntasks=11 --cpus-per-task=1 --ntasks-per-core=1 --mem-per-cpu=3900mb
>>
>> % Job ID vs SLURM Scheduler ID
>> job.ID
ans =
    65
>>
>> job.getTaskSchedulerIDs{:}
ans =
    '45220330'
```

Designing Robust Code

From Coding to Cluster (1)




```
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

```
>> % Start a local Processes parallel pool
>> parpool('Processes', 4);
Starting parallel pool (parpool) using the 'Processes' profile ...
Connected to parallel pool with 4 workers.
>>
>> % Call the function locally
>> ofile = process_files_v2
ofile =
    'C:\Users\dpietrus\Documents\MATLAB\files\output\SCS_14-Mar-2024_15-27-00'
```

Run it on the cluster

```
>> % Submit job to cluster
>> job = c.batch(@process_files_v2, 1, {'/home/lx516535', '/home/lx516535/output_files'}, ...
        'Pool', 3, 'CurrentFolder', '.');
Submit arguments: --ntasks=4 --cpus-per-task=1 --ntasks-per-core=1 --mem-per-cpu=3900mb -t 00:30:00
>>
>> % 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: /home/lx516535
```

Run it on the cluster

```
>> % Submit job to cluster
>> job = c.batch(@process_files_v2, 1, {'/home/lx516535/input_files', '/home/lx516535/output_files'}, ...
         'Pool', 3, 'CurrentFolder', '.');
Submit arguments: --ntasks=4 --cpus-per-task=1 --ntasks-per-core=1 --mem-per-cpu=3900mb -t 00:30:00
>>
>> % Wait for the job to finish
>> job.wait
>>
>> % Fetch the results
>> ofile = job.fetchOutputs{:}
ofile =
    '/home/lx516535/output_files/SCS_2-May-2024_15-32-12'
```

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{:}
```

Summary

- Prototype running on your desktop machine
- Create CLAIX profile (`configCluster`)
- Toggle between local profile (desktop machine) and CLAIX profile (multi-node)
- Tune your job with AdditionalProperties
- Best practices for job submission and troubleshooting
- Contact HPC Service Desk with questions or issues
  - servicedesk@itc.rwth-aachen.de
- Who would be interested in 1-1 Coaching?