HIGH PERFORMANCE RESEARCH COMPUTING

Using Matlab on ACES

ACES Training

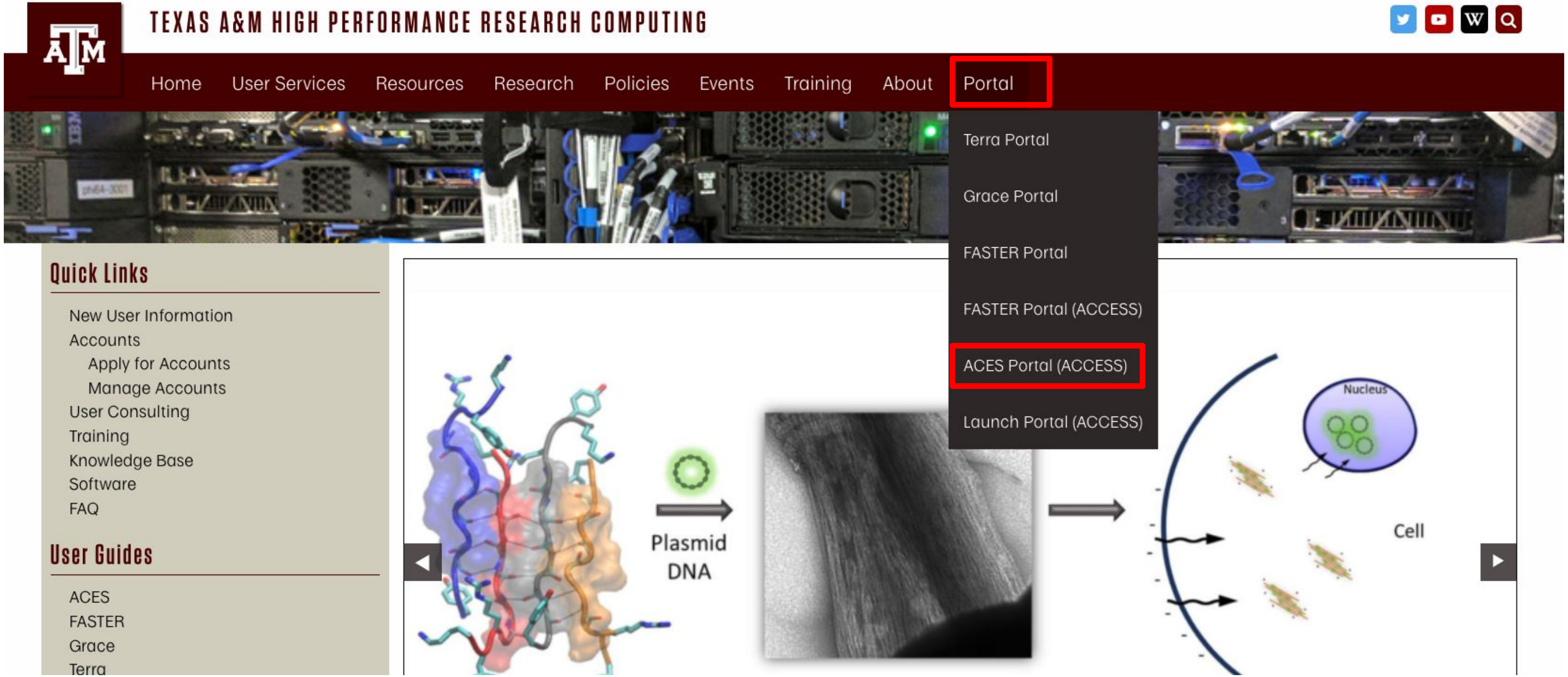
April 16, 2024

High Performance
Research Computing
DIVISION OF RESEARCH

Outline

- Running the Matlab GUI on the Portal
- Parallel Matlab: Multi Threading
- Parallel Matlab: Multi Processing
  - Cluster Profiles
  - Parallel Programming
  - GPUs
- Generating and submitting batch jobs using Drona Composer
- CASE Study: Monte Carlo Pi (time permitted)

Accessing the ACES Portal



TEXAS A&M HIGH PERFORMANCE RESEARCH COMPUTING

Home | User Services | Resources | Research | Policies | Events | Training | About | Portal

- Terra Portal
- Grace Portal
- FASTER Portal
- FASTER Portal (ACCESS)
- ACES Portal (ACCESS)
- Launch Portal (ACCESS)

Quick Links

- New User Information
- Accounts
  - Apply for Accounts
  - Manage Accounts
- User Consulting
- Training
- Knowledge Base
- Software
- FAQ

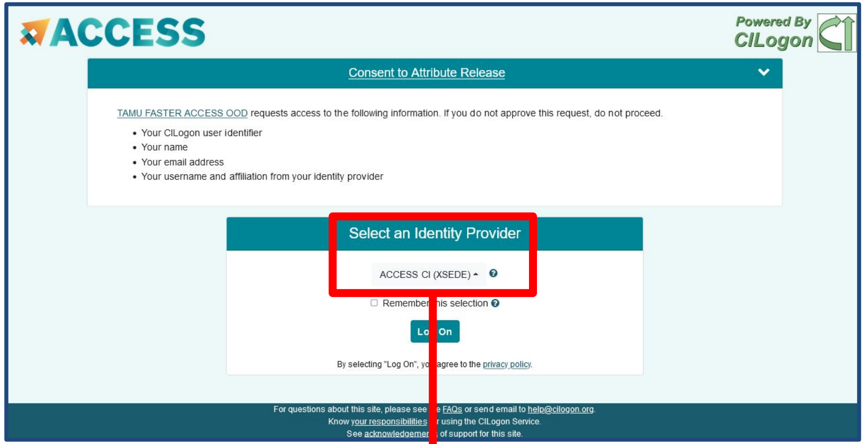
User Guides

- ACES
- FASTER
- Grace
- Terra

HPRC webpage: hprc.tamu.edu

Login to ACES through ACCESS

Log-in using your ACCESS credentials.



ACCESS

Powered By CILogon

Consent to Attribute Release

TAMU FASTER ACCESS OOD requests access to the following information. If you do not approve this request, do not proceed.

- Your CILogon user identifier
- Your name
- Your email address
- Your username and affiliation from your identity provider

Select an Identity Provider

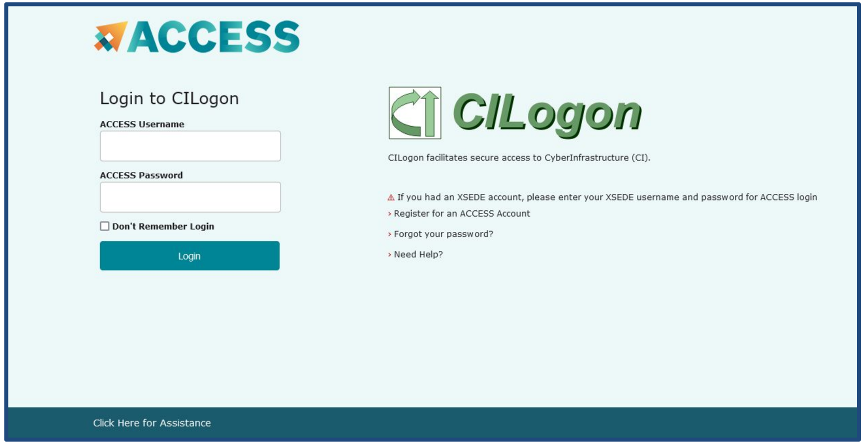
ACCESS CI (XSEDE)

□ Remember this selection

Log On

By selecting "Log On", you agree to the privacy policy.

For questions about this site, please see the FAQs or send email to help@cilogon.org.
Know your responsibilities for using the CILogon Service.
See acknowledgements of support for this site.



ACCESS

Login to CILogon

ACCESS Username

ACCESS Password

□ Don't Remember Login

Login

CILogon

CILogon facilitates secure access to CyberInfrastructure (CI).

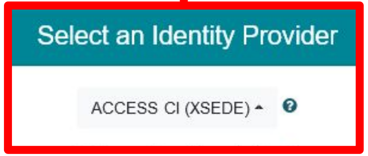
⚠ If you had an XSEDE account, please enter your XSEDE username and password for ACCESS login

› Register for an ACCESS Account

› Forgot your password?

› Need Help?

Click Here for Assistance



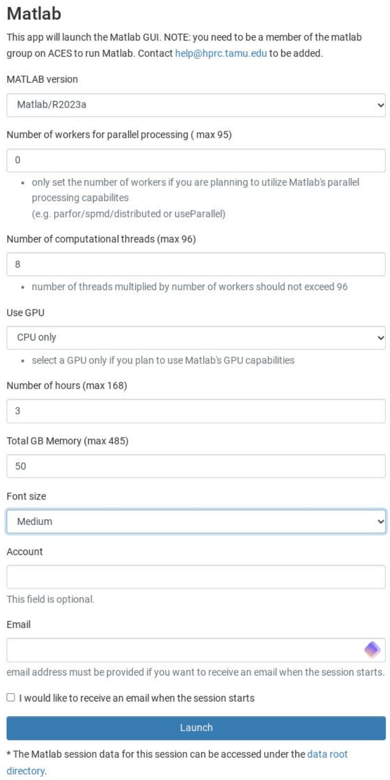
Select an Identity Provider

ACCESS CI (XSEDE)

Select the Identity Provider appropriate for your account.

Running Matlab on the Portal

1. Click on “Interactive Apps”
2. Select “Matlab_training”
3. Fill out the Form
   - Set time to 3 hours, threads to 8, memory 50GB
4. Click Launch
5. Once Matlab Job is running click on “Launch Matlab”



Matlab

This app will launch the Matlab GUI. NOTE: you need to be a member of the matlab group on ACES to run Matlab. Contact help@hprc.tamu.edu to be added.

MATLAB version

Matlab/R2023a

Number of workers for parallel processing (max 95)

0

- only set the number of workers if you are planning to utilize Matlab's parallel processing capabilites (e.g. parfor/spmd/distributed or useParallel)

Number of computational threads (max 96)

8

- number of threads multiplied by number of workers should not exceed 96

Use GPU

CPU only

- select a GPU only if you plan to use Matlab's GPU capabilities

Number of hours (max 168)

3

Total GB Memory (max 485)

50

Font size

Medium

Account

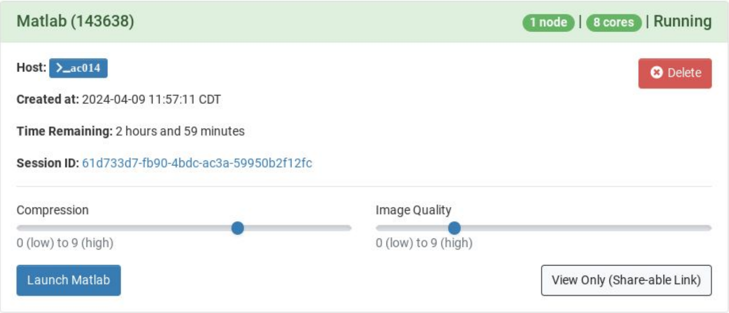
This field is optional.

Email

email address must be provided if you want to receive an email when the session starts.

□ I would like to receive an email when the session starts

* The Matlab session data for this session can be accessed under the data root directory.



Matlab (143638) 1 node | 8 cores | Running

Host: >_ac014

Created at: 2024-04-09 11:57:11 CDT

Time Remaining: 2 hours and 59 minutes

Session ID: 61d733d7-fb90-4bdc-ac3a-59950b2f12fc

Compression 0 (low) to 9 (high)

Image Quality 0 (low) to 9 (high)

Demo Time

(Let’s visit the portal and start the Matlab GUI)

Matlab Parallel processing on ACES

Now we know to run Matlab on ACES, let's do some parallel programming

1. Let's go back to the Interactive Matlab sesion we just created
   - if the session was closed for some reason, click "Launch Matlab" again
2. Open the live script "matlab_aces.mlx"
   - Upload file to ACES
   - Copy from directory /scratch/training/Matlab/matlab_aces_spring24.mlx
3. We will use the live script to explain and practice
   - parallel processing using multi threading
   - discover parallel profiles
     - local profile
     - cluster profile
   - parallel pools / parallel concepts
   - GPU

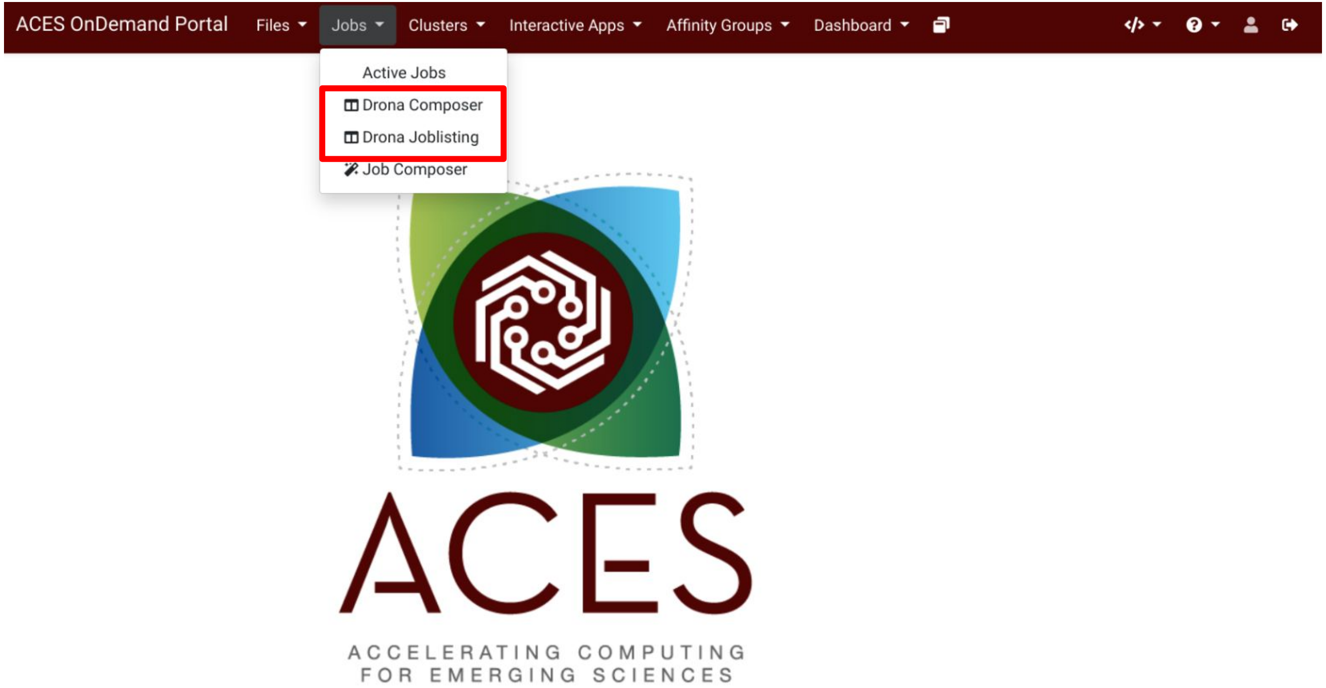
Back to the live script

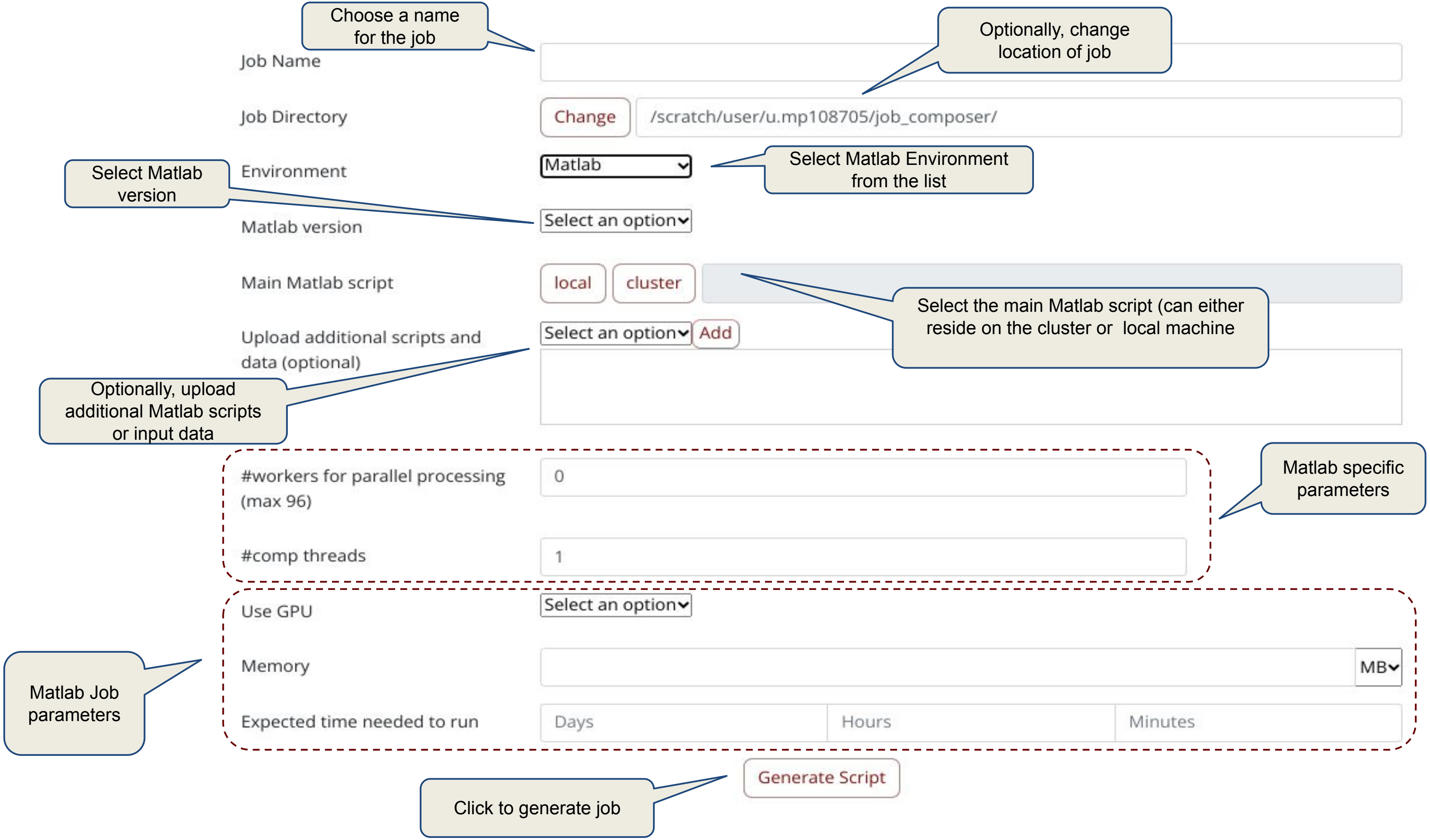
(time to do some parallel processing)

Drona Composer

For non-interactive jobs, user has to create a batch script specifying the resources and the commands to run. To simplify this task, HPRC developed Drona Composer: a framework to create any kind of workflows. Here, we will use the Matlab environment to generate and submit a Matlab job

Accessing the Drona Composer





Job Name	
Job Directory	Change /scratch/user/u.mp108705/job_composer/
Environment	Matlab
Matlab version	Select an option
Main Matlab script	local cluster
Upload additional scripts and data (optional)	Select an option Add
#workers for parallel processing (max 96)	0
#comp threads	1
Use GPU	Select an option
Memory	MB
Expected time needed to run	Days Hours Minutes

Generate Script

Choose a name for the job

Optionally, change location of job

Select Matlab version

Select Matlab Environment from the list

Select the main Matlab script (can either reside on the cluster or local machine

Optionally, upload additional Matlab scripts or input data

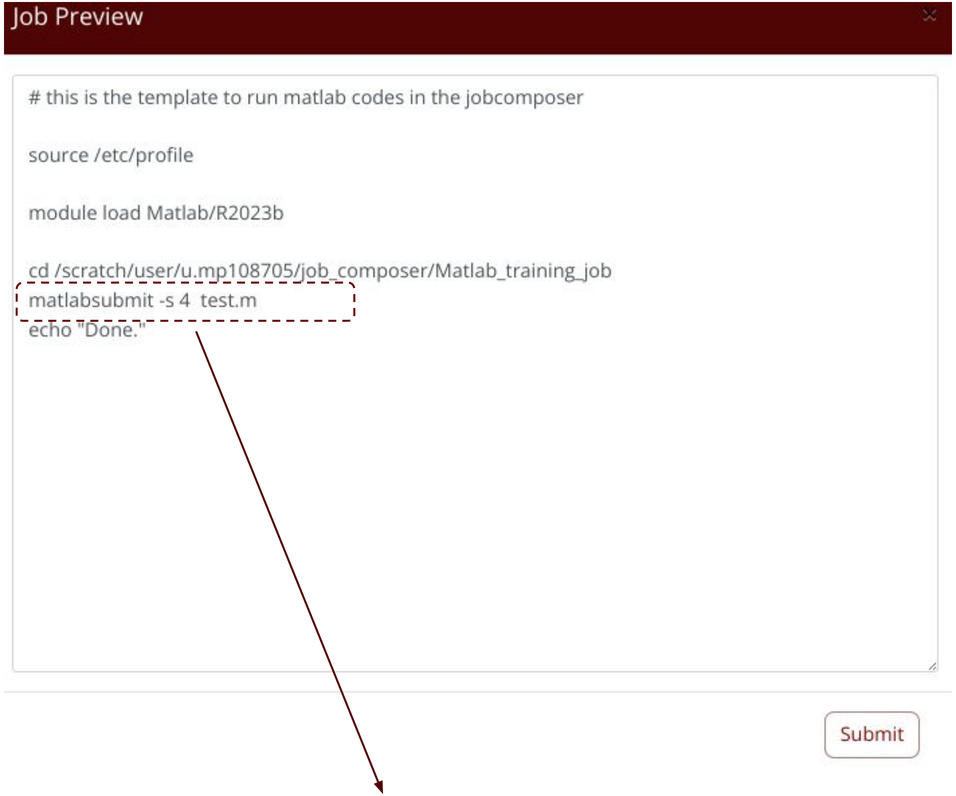
Matlab specific parameters

Matlab Job parameters

Click to generate job

Demo Time

(let’s create and submit a job using Drona Composer)



```
# this is the template to run matlab codes in the jobcomposer

source /etc/profile

module load Matlab/R2023b

cd /scratch/user/u.mp108705/job_composer/Matlab_training_job
matlabsubmit -s 4  test.m
echo "Done."
```

- batch_job_id-XXX
  - file containing the Slurm batch id (mostly for convenience)
- matlabsubmit_wrapper.m
  - boilerplate matlab code to set up environment and call to the main Matlab script provided by user
- slurm.out
  - redirected output from the script
- submission_script
  - the generated Slurm batch script

matlabsubmit will create a directory, named MatlabSubmitLOG<N> where all generated files and redirected output will be stored *(<N> is the matlabsubmit job ID)*

Case Study

(time permitted, you will compute Pi using Monte Carlo method. using parfor, spmd, distributed, and gpu)

HIGH PERFORMANCE RESEARCH COMPUTING
TEXAS A&M UNIVERSITY

Thank you.

Any questions?