README

Instructions for Download and Sharing

Download the Document as PDF

- Open the Google Doc version of the training.
- Ensure you only export the page titled “HTC Training” followed by its version number.
- Save/export as PDF.

Version Control

- The document begins at Version 1.0.
- Minor updates (e.g., grammar, formatting, small clarifications) → increment by 0.1 (e.g., 1.1 → 1.2).
- Major updates (e.g., new sections, substantive content changes) → increment by a full version (e.g., 1.0 → 2.0).

Update Workflow

- Before making any edits (major or minor):
- Duplicate the most recent version of the document.
- Make edits in the duplicate copy.
- Update the version number and the Last Updated date in the header of the duplicate.
- This ensures older versions remain accessible for reference.

Last Updated Date

- Always update the Last Updated date at the top of the document whenever changes are made.
- This signals to users that they are viewing the most current version.

Sharing

- Once exported, the PDF should be distributed to end users.
- Only the PDF version should be shared externally, never the editable Google Doc.

HTC Training (1.0)

JSU High-Throughput Computing Curriculum

Module 1: Introduction to High-Throughput Computing

Objectives: Define High-Throughput Computing, and understand where this form of computing is critical.

- **Introduction to HTC with HTCondor** [Reference video]
    - **High Throughput Computing (HTC):** Using computing resources so as to maximize the throughput of tasks over time toward solving a common problem. In HTC, many tasks (jobs) are run, often independently, to achieve large-scale goals. [path-cc.io/glossary]
    - **Distributed High Throughput Computing (dHTC):** An HTC setup that spans multiple independent administrative domains. These are pools of resources that may be geographically or administratively apart, joined by shared infrastructure so tasks can run across them. [path-cc.io/glossary]
    - **HTCondor / HTCSS:** Software that enables HTC workloads; first developed at UW–Madison in the 1990s to manage large numbers of jobs across distributed resources. [https://path-cc.io/glossary/]
    - **Other Relevant Terminology: [**https://path-cc.io/glossary/]
        i. **Node:** A computing server where jobs execute.
        ii. **Access Point (AP):** Where users submit workloads.
        iii. **Execution Point (EP):** Where jobs actually run.
        iv. **Central Manager (CM):** Tracks resources and manages scheduling.
        v. **Compute Entrypoint (CE):** Connects local resources to distributed HTC.
- **Introduction to the PATh project (NSF award #24-530)** [Reference video]
    - PATh: Partnership to Advance Throughput Computing, led by UW–Madison and OSG, expands access to dHTC. [http://path-cc.io/about]
    - *UW–Madison (via PATh) donated two nodes (execution + access) to support JSU's research computing efforts.² This is JSU's first step toward building a larger research computing capacity.*
- **Examples of HTC use**
    - **High Throughput Computing for Comparative Genomics on Large Public Datasets** [video]
        i. Conor Bendett shares about his work using HTC for comparative genomics for large public datasets. He further discusses CHTC and HTCondor and his experience, noting that there are lots of pipelining small tools; not always the easiest to manage; troubleshooting memory/disk space can be time-consuming (unpredictable outputs), and the great support from CHTC facilitators.
    - **Utilizing HTCondor, Pelican, and DAGman workflows for high-throughput phenotyping** [video]

    i. Ariana Negreiro disusses the Digital Livestock Lab at the UW-Madison Dept of Animal and Dairy Sciences which focuses on research applications of ML and computer vision for farm management and genetic selection. Negreiro disucssed the data processing pipeline and the CHTC Resources (HTCondor, Pelican and DAGMAN) that they rely upon for their work.
- **HTCondor and Interactive Use: A success story** [video]
    i. Oliver Freyermuth discusses his practical experience with an interactive-first approach to leverage HTC resources. He discusses this story and points out areas of success and effective use of these tools.

Module 2: Linux and Command Line Basics

Objectives: Navigate file systems, edit files, manage permissions. Execute and chain basic commands. (Navigate directories using cd, ls, pwd. Create move, copy, and delete files using touch,mkdir, cp, mv, rm. Edit files using nano or vim. Set file permissions using chmod. Use pipes and redirects.)

- **Linux** is the operating system used in most research computing. The **command line** (or terminal) is a text-based way to give instructions, like making folders, moving files, or running jobs. To edit scripts or batch files, you can use editors such as nano (best for beginners) or vim (more advanced).
- New users should start with the [**Ubuntu Command Line for Beginners tutorial**]
    - **Assessment (optional):**
        1. Create a directory called `htc_training`
        2. Create a subfolder called `module2`
        3. Within the `module2` subdirectory, create subfolders `data scripts results log`
        4. inside the `data` folder, create a file called `numbers.txt` that has the numbers 10, 20, 30, 40, and 50 (each on its own line).
        5. In the `scripts` folder, create a script named `count_numbers.sh`
            a. The script should:
                i. Record the date/time into a log file.
                ii. Count how many numbers are in `numbers.txt`
                iii. Add up all the numbers.
                iv. Save both results into `results/count.txt`
        6. Make the script executable and run it.
        7. Check that `results/count.txt` shows the correct line count and the sum.
        8. Check that `logs/run.log` contains the date/time of the run.
        9. Create an archive file called `module2_numbers_artifacts.tgz` that includes your `results` and `logs` folders.

Module 3: Job Scheduling and Execution

Objectives: Further understand what HTCondor does. Understand Basic Submit Files.

- **Hello World Tutorial: Submit HTC Jobs using HTCondor** [video]
- In HTC, a **job** is a unit of work submitted to the system. Each job must include a description of what to run (the executable or script), how much memory and CPU it requires, and what input or output files it will use. To run efficiently at scale, large research problems are broken into many smaller jobs that can be processed in parallel. Each job runs in its own sandbox, an isolated environment that contains only the files and resources defined for that task. On Linux and macOS, jobs are often run using shell scripts, while on Windows this role is played by batch files; both differ from compiled executables, which are standalone programs.
- To practice, learners should follow the [**OSG Quickstart tutorial for submitting an HTCondor job**], using the full manual steps to reinforce earlier Linux and command-line training.
    - **Assessment (optional):**
        1. Create a new folder: `HTC_training/module3`
        2. Inside it, write a script called `work.sh` that:
            a. Prints the start time
            b. Prints the hostname (machine name)
            c. Waits for 5 seconds
            d. Prints the end time
        3. Make your script executable.
        4. Create a submit file called `job.sub` that will run your script and save:
            a. Standard output to .out
            b. Errors to `.err`
            c. A log file `.log`
        5. Submit the job to HTCondor.
        6. Monitor your job in the queue.
        7. Cancel your job at least once while it is waiting or running.
        8. Re-submit the job so that it finishes successfully.
        9. Check the `.out` file to confirm it contains the start time, hostname, and end time.
        10. Create an archive file called `module3_condor_artifacts.tgz` that includes your script, submit file, and job outputs/logs.

Module 4: Multiple Jobs & Transfer Data

Objectives: Understand how to submit multiple jobs, transfer files, transfer output. Submit, monitor, and cancel jobs using HTCondor.

- **HTC Job Execution with HTCondor** [video]
- **Submitting Multiple Jobs with HTCondor** [video]
- **A User's Guide to Submitting HTCondor jobs** [video]
- Effective data management is essential in HTC. Researchers often need to make software portable by packaging it into containers or compressed archives (tarballs) so it can run consistently across different systems.
- For distributed jobs, clear input and output file strategies are required: jobs must know what data to pull in and where to place results. Data transfer can be a challenge at scale, so HTC systems use tools like Globus or built-in transfer mechanisms.
- In HTCondor, the `transfer_input_files` option within a submit file specifies exactly which files should be sent to each job's sandbox, ensuring that jobs have what they need to run successfully anywhere.
    - ***Overview: Data Staging and Transfer to Jobs*** provides context about placing files (in /home, OSDF, etc.), best practices for small vs large files and using systems such as the OSDF.
        - ***Transfer Smaller Job Files To and From /home using HTCondor*** on the OSG Portal shows how to use `transfer_input_files` and `transfer_output_files` in submit files to stage data via HTCondor.
    - **Assessment (optional):**
        1. Create a folder `HTC_training/module3_2` with `data` and `transfer` subfolders.
        2. Place a sample text file in `data/`
        3. Copy the file into the `transfer/` folder using a file transfer command.
        4. Compress the data/ folder into a `.tar.gz` file.
        5. Extract the archive into a new folder called `unpacked/`
        6. Verify the file contents.

Module 5: Software & Environments

Objectives: Load software with environment modules and create reproducible environments

- **HTCondor: Environment and services for running a job [documentation]**
- **Backpacking with Code: Software Portability for DHTC [video]**
- Having the same software setup across machines is critical in research computing. Understanding how to load software via environment modules and create reproducible environments enables users to switch between software versions cleanly on shared systems. Reproducible environments ensure that jobs will behave the same way wherever they run. Users will often need to specify environment variables, use wrappers, or package the environment so that their job remains reproducible.
    - **Assessment (optional):**
        1. Create a folder `HTC_training/module5`

2. Load or create a software environment (e.g., Conda).
3. Install a package (e.g., `numpy`).
4. Write a Python script that imports the package and prints a result.
5. Run the script inside your environment.
6. Save the script and environment info.
7. Archive as `module5_artifacts.tgz`

Supplemental Information

DAGMan: DAGMan is an HTCondor workflow management tool. Instead of submitting one job at a time, DAGMan allows you to define dependencies between jobs (*Job B can only start after Job A finishes successfully*). For example, if you have a 3-step process (*data preprocessing → model training → analysis*) DAGMan will ensure each stage runs in sequence automatically.

- **HTCondor DAGman Workflows tutorial** [video]

Helpful Case Studies/Examples of HTC usage

- Weather-Driven Insect Dispersal Simulations Using Workflows that Combine HTC and HPC Resources [video]
- Using high-throughput computing to develop precision mental health algorithms [video]
- HTC24: Jupyter Notebooks as a frontend for a HTC analysis facility [video]
- HTC24: Throughput Machine Learning in CHTC [video]

References

Canonical Ubuntu. (n.d.). *The Linux command line for beginners*. Ubuntu. Retrieved September 24, 2025, from https://ubuntu.com/tutorials/command-line-for-beginners#1-overview

Center for High Throughput Computing (Director). (2020a, June 17). *Submitting Multiple Jobs with HTCondor—YouTube* [Video recording]. https://www.youtube.com/watch?v=m7dQChJH5LU&list=PLO7gMRGDPNumCuo3pCdRk23GDLNKFVjHn&index=4

Center for High Throughput Computing (Director). (2020b, September 1). *Introduction to High Throughput Computing with HTCondor* [Video recording]. https://www.youtube.com/watch?v=oMAvxsFJaw4

Center for High Throughput Computing (Director). (2020c, September 4). *HTCondor DAGman Workflows tutorial—YouTube* [Video recording]. https://www.youtube.com/watch?v=1MvVHxRs7iU&list=PLO7gMRGDPNumCuo3pCdRk23GDLNKFVjHn&index=5

Center for High Throughput Computing (Director). (2020d, October 22). *A User's Guide to Submitting HTCondor jobs—YouTube* [Video recording]. https://www.youtube.com/watch?v=p2X6s_7e51k&list=PLO7gMRGDPNumCuo3pCdRk23GDLNKFVjHn&index=3

Center for High Throughput Computing (Director). (2022, May 27). *Weather-Driven Insect Dispersal Simulations Using Workflows that Combine HTC and HPC Resources* [Video recording]. https://www.youtube.com/watch?v=EFYMwxAnA44

Center for High Throughput Computing (Director). (2025, April 29). *Hello World Tutorial: Submit HTC Jobs using HTCondor* [Video recording]. https://www.youtube.com/watch?v=d5siupeu2kE

Center for High Throughput Computing, Computer Sciences Department, University of Wisconsin-Madison, & HTCondor. (n.d.). *Environment and services for a running job—HTCondor Manual 24.12.4 documentation*. Retrieved September 24, 2025, from https://htcondor.readthedocs.io/en/latest/users-manual/env-of-job.html

OSG (Open Science Grid). (n.d.-a). *Overview: Data Staging and Transfer to Jobs—OSPool Documentation*. Retrieved September 24, 2025, from https://portal.osg-htc.org/documentation/htc_workloads/managing_data/overview/

OSG (Open Science Grid). (n.d.-b). *Quickstart-Submit Example HTCondor Jobs—OSPool Documentation*. Retrieved September 24, 2025, from https://portal.osg-htc.org/documentation/htc_workloads/submitting_workloads/tutorial-quickstart/

OSG (Open Science Grid). (n.d.-c). *Transfer Smaller Job Files to and from /home—OSPool Documentation*. Retrieved September 24, 2025, from https://portal.osg-htc.org/documentation/htc_workloads/managing_data/file-transfer-via-htcondor/

Partnership to Advance Throughput Computing. (n.d.-a). *About PATh*. PATh. Retrieved September 24, 2025, from https://path-cc.io/about/

Partnership to Advance Throughput Computing. (n.d.-b). *Glossary and Concepts*. PATh. Retrieved September 24, 2025, from https://path-cc.io/glossary/

Partnership to Advance Throughput Computing. (2020, October 29). *Welcome and PATh Overview (2020 OSG Campus Workshop)—YouTube*. https://www.youtube.com/watch?v=RSQnJOky4jo

Partnership to Advance Throughput Computing (Director). (2021a, June 17). *Using high-throughput computing to develop precision mental health algorithms* [Video recording]. https://www.youtube.com/watch?v=qQaXtPkyXGE

Partnership to Advance Throughput Computing (Director). (2021b, August 4). *HTC Job Execution with HTCondor* [Video recording]. https://www.youtube.com/watch?v=9896xAhT4dY

Partnership to Advance Throughput Computing (Director). (2021c, August 5). *Backpacking with Code: Software Portability for DHTC* [Video recording]. https://www.youtube.com/watch?v=xUeIQbVXOMQ

Partnership to Advance Throughput Computing (Director). (2024a, July 25). *HTC24: Throughput Machine Learning in CHTC* [Video recording]. https://www.youtube.com/watch?v=PE93xEhzKwQ

Partnership to Advance Throughput Computing (Director). (2024b, July 29). *HTC24: HTCondor and Interactive Use: A success story* [Video recording]. https://www.youtube.com/watch?v=59w5UTRmyXo

Partnership to Advance Throughput Computing (Director). (2024c, July 29). *HTC24: Jupyter Notebooks as a frontend for a HTC analysis facility* [Video recording]. https://www.youtube.com/watch?v=iWVefhf-mUY

Partnership to Advance Throughput Computing (Director). (2025a, August 15). *HTC25: High Throughput Computing for Comparative Genomics on Large Public Datasets—YouTube* [Video recording]. https://www.youtube.com/watch?v=UFtBKB6pG1Y

Partnership to Advance Throughput Computing (Director). (2025b, August 15). *HTC25: Utilizing HTCondor, Pelican, and DAGman workflows for high throughput phenotyping* [Video recording]. https://www.youtube.com/watch?v=YT2mHKkCKwk

Training Videos

dHTC Facilitation Training - 5 videos by PaTH

- Introduction to dHTC and OSG
  - Partnership to Advance Throughput Computing
  - •228 views • 4 years ago
- How to execute computational work using HTCondor, and on the Open Science Grid
  - Partnership to Advance Throughput Computing
  - •284 views • 4 years ago
- Software and Data Support in OSG
  - Partnership to Advance Throughput Computing
  - •44 views • 4 years ago
- Facilitating researchers to advance their work through dHTC approaches and services
  - Partnership to Advance Throughput Computing
  - •31 views • 4 years ago
- dHTC and OSG Services to Advance Research on Campuses
  - Partnership to Advance Throughput Computing
  - •32 views • 4 years ago

OSG School 2021 Public Lectures - 5 videos by PaTH

- Introduction to the Virtual School, HTC, and OSG
  - Partnership to Advance Throughput Computing
  - •330 views • 3 years ago
- HTC Job Execution with HTCondor
  - Partnership to Advance Throughput Computing
  - •333 views • 3 years ago 1:01:2
- Introduction to OSG
  - Partnership to Advance Throughput Computing
  - •536 views • 3 years ago
- Backpacking with Code: Software Portability for DHTC
  - Partnership to Advance Throughput Computing
  - •104 views • 3 years ago
- Handling Data on OSG
  - Partnership to Advance Throughput Computing
  - •163 views • 3 years ago

HTC Condor User Tutorials - 6 videos by cHTC

- Introduction to High Throughput Computing with HTCondor
  - Center for High Throughput Computing
  - •4.8K views • 4 years ago
- A User's Guide to Submitting HTCondor jobs
  - Center for High Throughput Computing

    - •6.5K views • 4 years ago
- Submitting Multiple Jobs with HTCondor
    - Center for High Throughput Computing
    - •2.7K views • 4 years ago
- HTCondor DAGman Workflows tutorial
    - Center for High Throughput Computing
    - •1.8K views • 4 years ago
- Backpacking with code -- Software Portability for DHTC
    - Center for High Throughput Computing
    - •281 views • 4 years ago
- The HTCondor ClassAd Language Tutorial
    - Center for High Throughput Computing
    - •1K views • 4 years ago

Additional Videos

- An Introduction to Using HTCondor
    - Center for High Throughput Computing
    - •3.7K views • 2 years ago
- ACES: Containers for Scientific Workflows (Singularity / Apptainer)
    - Texas A&M HPRC
    - •154 views • 1 year ago
- HTCondor and GIS Working Together
    - Center for High Throughput Computing
    - •75 views • 2 years ago
- Hello World Tutorial: Submit HTC Jobs using HTCondor
    - Center for High Throughput Computing
    - •20 views • 1 day ago
- Two Factor Authentication demonstration - CHTC
    - Center for High Throughput Computing
    - •464 views • 2 years ago

Helpful Case Studies/Examples of HTC usage

- Weather-Driven Insect Dispersal Simulations Using Workflows that Combine HTC and HPC Resources
    - Center for High Throughput Computing
    - •60 views • 2 years ago
- Using high-throughput computing to develop precision mental health algorithms
    - Partnership to Advance Throughput Computing
    - 34 views 3 years ago
- HTC24: Jupyter Notebooks as a frontend for a HTC analysis facility
    - Partnership to Advance Throughput Computing
    - 17 views 9 months ago
- HTC24: Throughput Machine Learning in CHTC
    - Partnership to Advance Throughput Computing
    - 30 views 9 months ago

Training Workflow

HTC Training Workflow

1. The requester fills out the JSU Research Computing Request form.
2. JSU CI Facilitation staff will review the intake form
3. JSU Facilitation staff will follow up to offer times for an initial consultation.
4. Initial consultation occurs, researcher is routed to:
    a. HTC
    b. HPC
    c. AI
    d. Other
5. **OSG office hours with JSU RC team member → so they know who is using the system if they reach out for support**
    a. **Tuesdays (5:30 pm ET / 1-2:30 pm PT) or Thursdays (11:30 am-1 pm ET / 8:30-10 am PT)**
    b. **Sign in for Office Hours**
    c. **Meeting link: https://osg-htc.org/OfficeHoursZoom**
6. **The individual attends the JSU Research Computing Intake Training that occurs**
    a. **Thrice a semester (FA/SP)**
    b. **Overview 1-1 intake session to learn about their research. If it is a research group requesting resources, the entire group should join the intake.**
    c. **General HTC Session**
        i. **Review of HTC policies and guidelines → JSU policies and HTCondor documentation**
7. **Intro tutorial using OSG's Guest Jupyter notebook. Notebook is not to be used for actual data!**
8. **Individual receives their login credentials from the JSU Facilitation team and reads and signs the acknowledgment of having read the policy**
9. **Can also send them to the OSG Trainings they host each semester**
    a. *"All User Training sessions are offered on Tuesdays from 2:30-4 pm ET (11:30am - 1pm PT), **on the third Tuesday** of the month. The trainings are designed as stand alone subjects. You do not need to bring/have your dataset prepared before the training. The only prerequisites are some familiarities with using the command line inteface or shell. Having some familiarities with HTCondor job submissions are useful but not required."*

Training Working Doc

In-Person

- What is HTC?
    - Use path-cc.io/glossary definitions for both HTC and dHTC. **REFERENCE**
        i. Break down HTC and dHTC using graphs/figures
        ii. Describe how it differs from HPC
        iii. Showcase how it supports research (3 examples)
    - Then describe the PATh project (NSF award #24-530) and its relation to JSU (path-cc.io/about) **REFERENCE**
        i. Mention they donated 2 nodes to support JSUs Research Computing efforts (execution and access)
            - Describe what a node is
        ii. Mention that PATh brings together CHTC and the OSG Consortium. Define both in terms researchers will understand.
        iii. For CHTC, conclude by describing it as a larger and more advanced HTC system that is specific to the UW-Mad campus. Mention how JSU is looking to expand their Research Computing capabilities similarly – the acquisition of this node is the first step.
        iv. For OSG describe things such as:
            - OSDF
            - OSPool
            - Glide ins etc (as external researchers can glide in and use JSUs node)
- Linux / Command Line Basics
    - https://ubuntu.com/tutorials/command-line-for-beginners#3-opening-a-terminal **REFERENCE**
    - Be sure to also include things such as nano or vim for script/batch editing. Nano is best for beginners
    - **GOAL:** Navigate directories using cd, ls, pwd. Create move, copy, and delete files using touch,mkdir, cp, mv, rm. Edit files using nano or vim. Set file permissions using chomd. Use pipes and redirects.
    - **OPTIONAL TEST OUT:** Allows experienced users to skip the module by demonstrating competency
        10. Create a directory called htc_training/
        11. Create a new file hello.sh inside it that contains
            echo “Hello, HTC World!”
        12. Make the script executable and run it, sending the output to a file called output.txt
        13. Copy output.txt into a subdirectory called results/
        14. Use one command to count how many lines are in output.txt and redirect that into a file called summary.log
        15. Display the contents of summary.log with cat

        i. Submission through a terminal output screenshot, zip and submit the entire folder, or running in a shared remote shell
- HTCondor Literacy and job execution
    - Provide a brief history of HTCondor (https://htcondor.org/) **REFERENCE**
    - Define Access Point (AP), Execution Point (EP), Central Manager (CM), Compute Entrypoint (CE) as defined by https://path-cc.io/glossary/
    - Define a Job, why workflows must be broken down via HTC, define sandbox (Attributes of jobs that must be defined include the executable or script to run, the amount of memory, CPU and other machine resources it needs, and descriptions of the file inputs it need), difference between a shell script (linux/mac)/batch file(windows) vs an executable
    - Follow the OSG Quickstart example for submitting an HTCondor job **REFERENCE** (https://portal.osg-htc.org/documentation/htc_workloads/submitting_workloads/tutorial-quickstart/) it is probably best to follow the full manual tutorial so new users can reinforce the earlier Linux/command line training information.
    - **GOAL:** Create a basic HTCondor submit file and run a test job on the node
- HTC Basics – Software Portability & Data Handling
    - Understand how to package software for portability (containers/tarballs/etc)
    - Describe input/output file strategies for distributed jobs
    - Identify data transfer challenges and how to use DT systems on HTC
    - Use of transfer_input_files in an HTCondor submit file

Asynchronous