# Introduction to High Throughput Computing

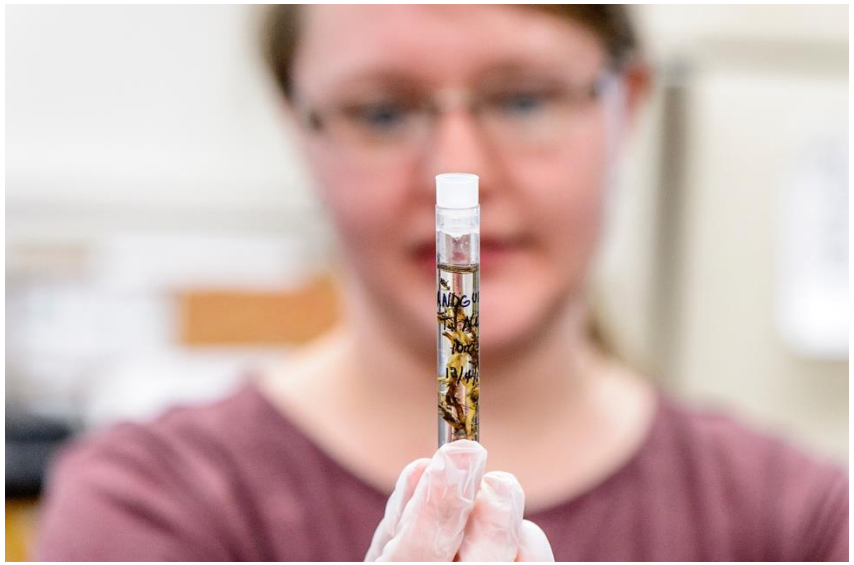
Christina Koch

OSG School 2026

July 13, 2026

# Researcher Problems

# Researcher #1



- New research student
- Working in plant pathology lab, studying plant genomes
- Can run first step of pipeline on one sample
- Now has 50 samples to run

# Researcher #2

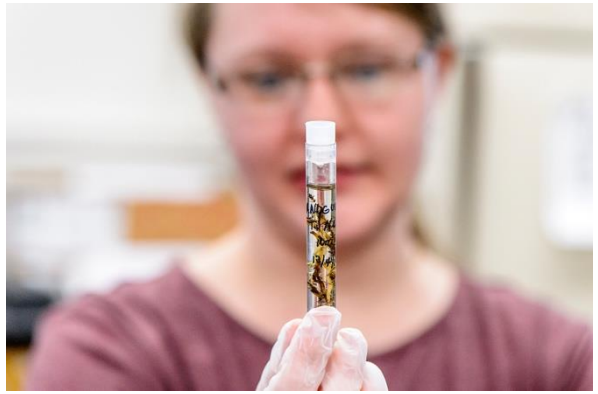
- Starting master's project
- Using self-written model which predicts accuracy of a medical trial design
- Model takes 3-4 hours to run
- Want to test many designs (each design is expressed as a combination of parameters)

# Researcher #3


- Member of a large physics collaboration
- Want to predict (with probability) behavior of particle in detector
- Collaboration has particle simulation code already
- Probability estimate comes from running millions of particle simulations

# What do they have in common?

Each researcher has a (non-ordered) list of **tasks** that would take too long to run sequentially on their local computer.



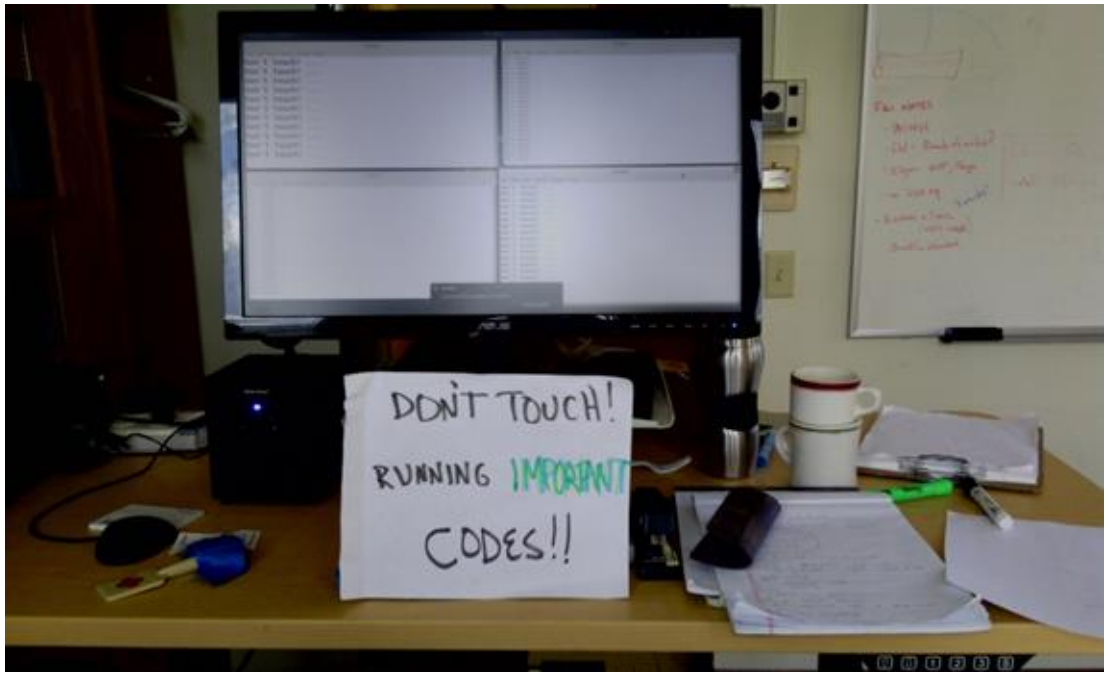
Running **analysis pipeline** for each sample.

Running a **simulation** for each parameter combination


Running millions of **simulations**.

# Why do we care?

Each researcher has a (non-ordered) list of tasks that would take **too long to run sequentially on their local computer**.



# Scaling Up

Each researcher has a (non-ordered) list of tasks that would take **too long to run sequentially on their local computer**.

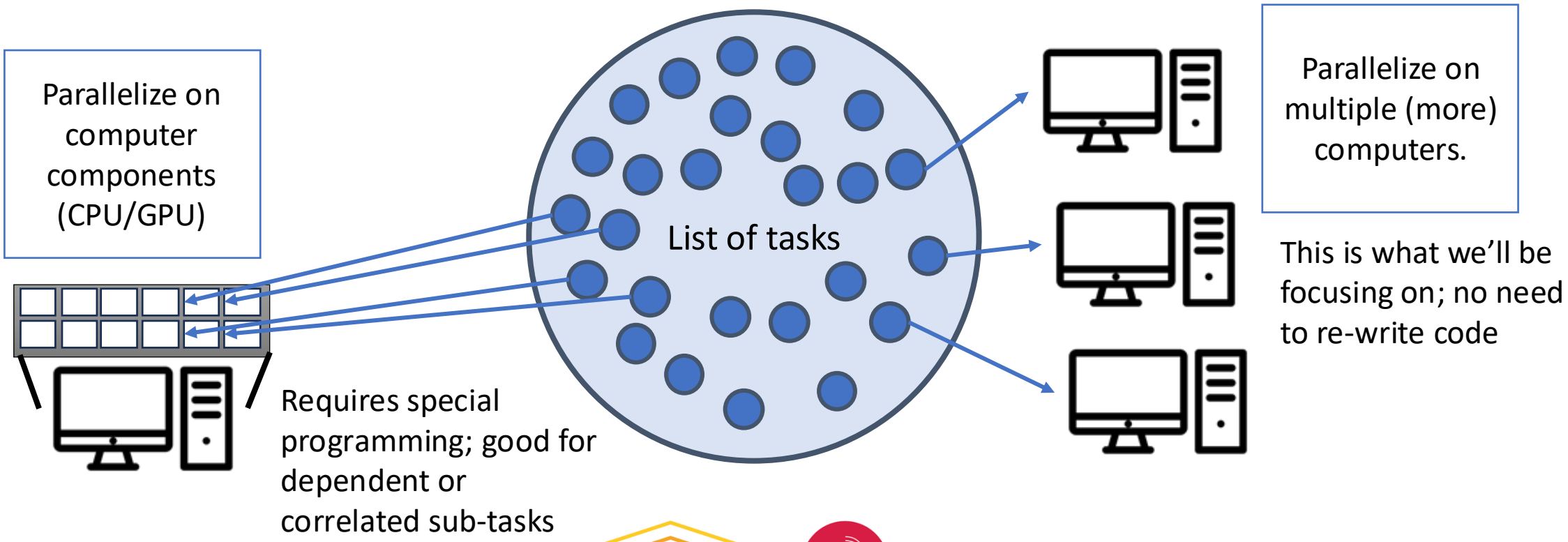
Don’t let computing be a barrier to your research!! We want to be able to tackle big problems.

The strategy to speed things up with computers is ALWAYS to run in parallel.

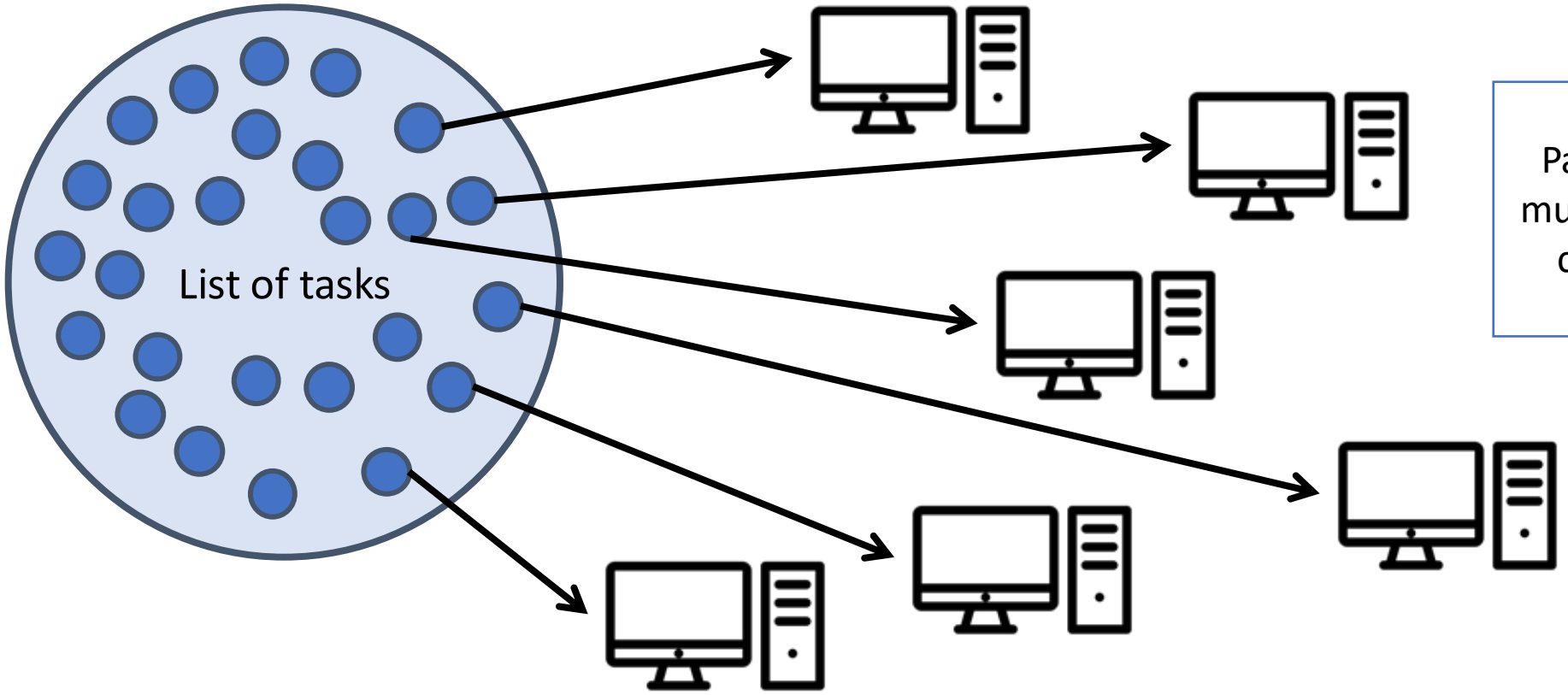
Then you can go from using a small number of resources (one computer) to a LOT of resources (for example, many computers).

# Running in Parallel

There are different ways to parallelize in computing:

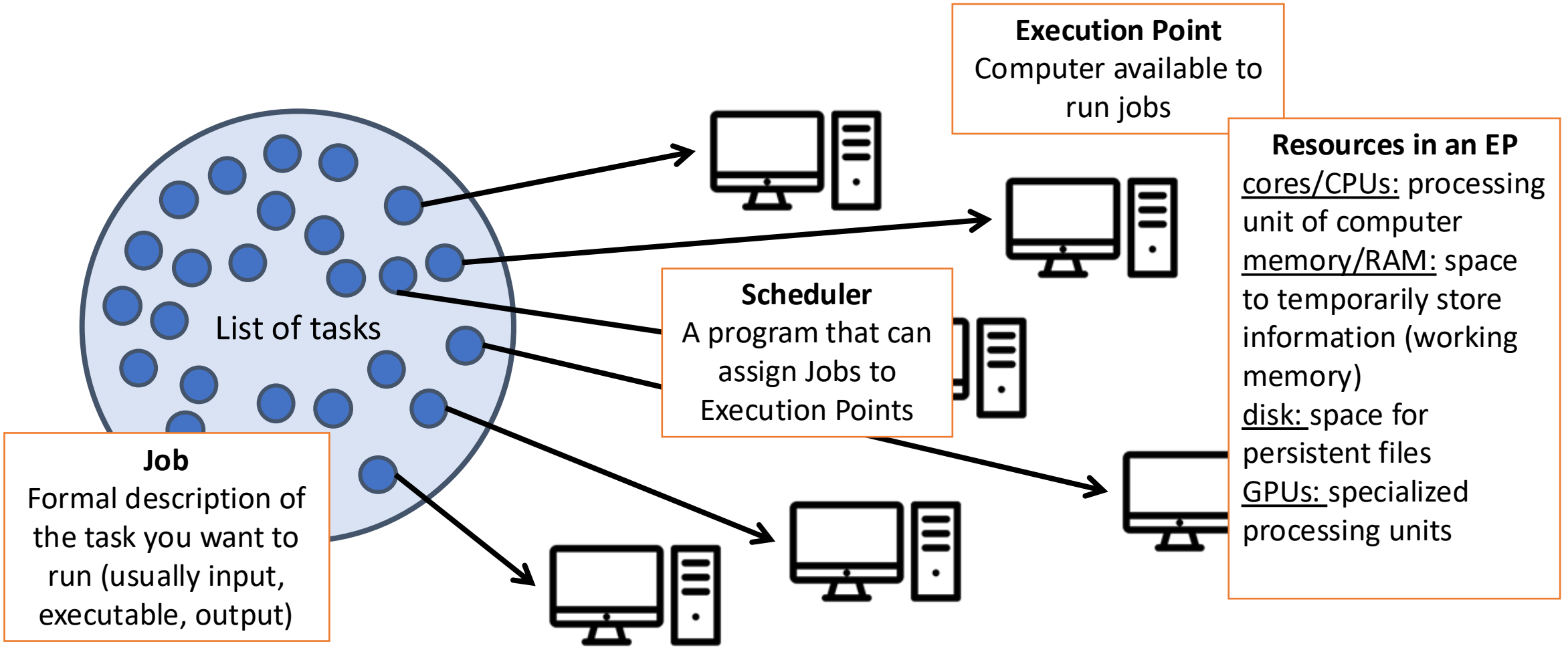


# High Throughput Computing (HTC)




Parallelize on multiple (more) computers.

# Some Terms



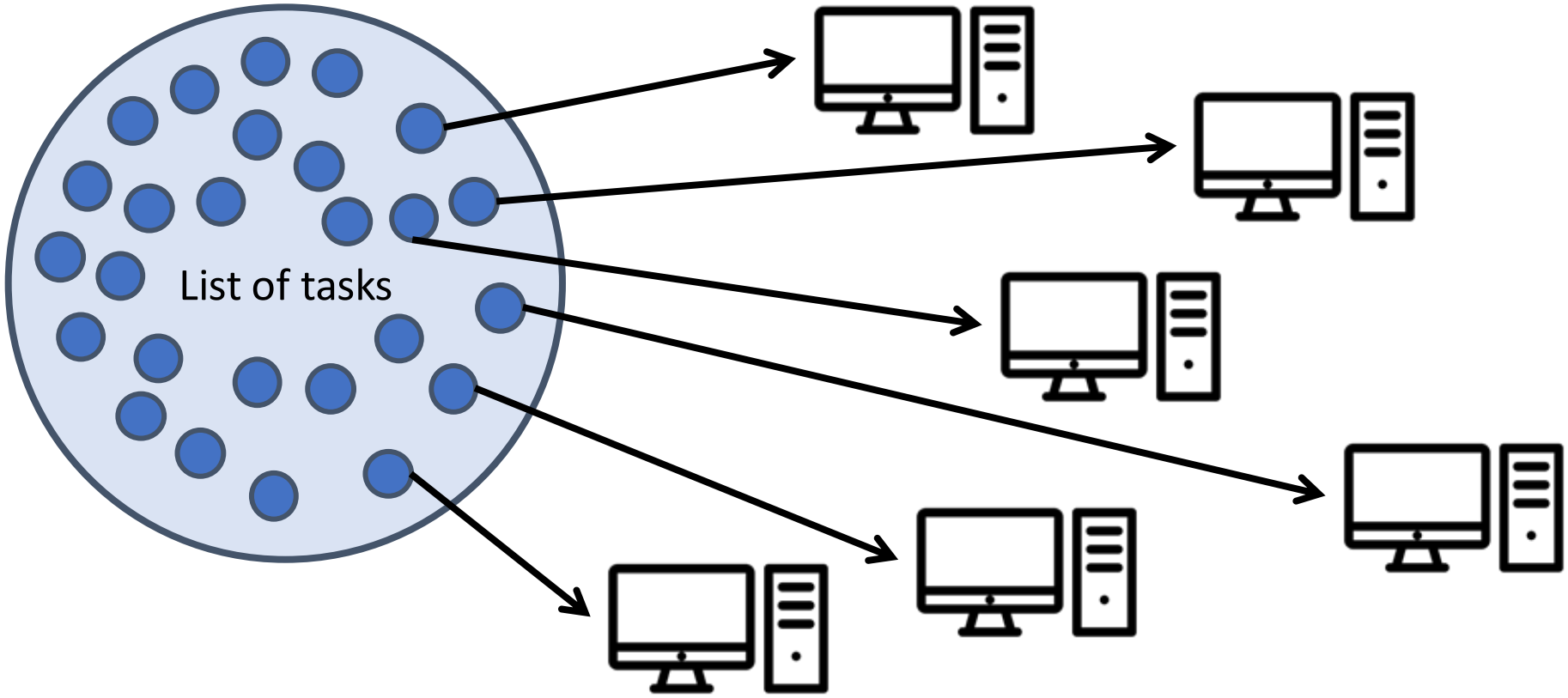

**Execution Point**
Computer available to run jobs

**Resources in an EP**
cores/CPUs: processing unit of computer
memory/RAM: space to temporarily store information (working memory)
disk: space for persistent files
GPUs: specialized processing units

**Scheduler**
A program that can assign Jobs to Execution Points

**Job**
Formal description of the task you want to run (usually input, executable, output)

# High Throughput Computing (HTC)



# What you need to do HTC

- A “home” to organize and start the computation
- Access to more computing capacity (an HTC system)
- Tools to manage and run our list of tasks (the scheduler)
- Components needed to run our tasks: software, scripts, data

**This is exactly what we’re going to cover this week.**

# A “Home” for HTC Workflows

**Access Point account**

- ❑ ap40.uw.osg-htc.org
- ❑ some of you have have access to additional Access Points

- OSG Online Guides
  - Main Page > Get Started on the OSPool > Account Setup
  - Account and guide portal: https://portal.osg-htc.org/

# Access to HTC Systems

**Open Science Pool** – Mon – Fri

- OSG School materials
  - OSPool Introduction (Tuesday)
- OSG Online Guides
  - Main Page > Get Started on the OSPool > Welcome
  - Account and guide portal: https://portal.osg-htc.org/

# Managing and Running Jobs

**HTCondor** (for most cases) and **DAGMan** (for workflows)

- OSG School materials
  - HTCondor Introduction (Mon)
  - Troubleshooting (Thurs)
  - Workflows with DAGMan (Thurs)
- OSG Online Guides
  - Main Page > Submit HTC Workloads > HTC Workload Planning, Testing and Scaling Up
  - Main Page > Submit HTC Workloads > Monitor, Review and Troubleshoot Jobs

# Job Components: Software, Data, Scripts

- OSG School materials:
  - Software (Tuesday)
  - Data (Wednesday)
- OSG Online Guides
  - Main Page > Submit HTC Workloads > Managing Data for Jobs
  - Main Page > Submit HTC Workloads > Using Software

# Big picture goals for the week

| Mon | Tues | Wed | Thurs | Fri |
|---|---|---|---|---|
| Identify your job "task", your list of inputs, and be able to express them in an HTCondor submit file. | Follow instructions to create a container and use it in a job; adapt for different software.<br><br>Describe the components of the OSPool | Enumerate the files needed for your job(s) and stage them in the appropriate place. | Identify where jobs and workflows might run into issues when starting out.<br><br>Run a job that does *your* work, based on what you've learned in the previous three days. | Continue to refine your job/workflow prototype.<br><br>Identify next steps for your learning and workflow implementation. |

# A note about LLMs/Agentic AI

- LLMs and “AI” agents are **powerful tools**, well suited to many of the tasks we are doing this week:
  - Writing scripts or job submit files
  - Writing container build files
  - Troubleshooting errors or issues
- AND, **they are most effective** when their user has the **background** and **context** to provide meaningful prompts and interpret the results.
- AND, we brought you here so **you could learn new things**! 

# Our suggestions

- Don’t use any LLMs / AI agents for at least the first 2 days of the School.
  - Ask School staff/neighbors for help if you get stuck.
  - Write or type out information yourself
- By day 4, use LLMs judiciously for tasks that you could probably do yourself or easily look up.
  - “Modify this definition file to install this software.”
  - “Create a template submit file for me, based on the School materials.”
  - “Where should I look for errors?”

# Getting Started

- We’re here to help you do the following:
  - Think about your work as a list of jobs
  - Get it running on the OSPool
- Lots of resources available:
  - OSG School materials: slides, exercises
  - OSG guides and training materials
  - Other technical lessons (unix, git, naming things, docker…let’s crowd-source other materials as needed)

# Connecting with Each Other

- Connect with people who are doing similar work to you!
  - Today: lunch with people in similar domains
  - Tomorrow: lunch with people using similar tools
- Do you want to share your contact info? Or a cool resource you know about for computing? Share here:
  - **https://go.wisc.edu/4b3k15**
- Staff are also a resource! Talk to us and sign up for consultations on the whiteboard.

# Acknowledgements

This work was supported by the National Science Foundation under Cooperative Agreement OAC-2030508 – Partnership to Advance Throughput Computing (PATh)