



Diamond Harnessing GPU Resources for Scientific Deep Learning

Max Burnette, NCSA



**National Center for
Supercomputing Applications**

UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN



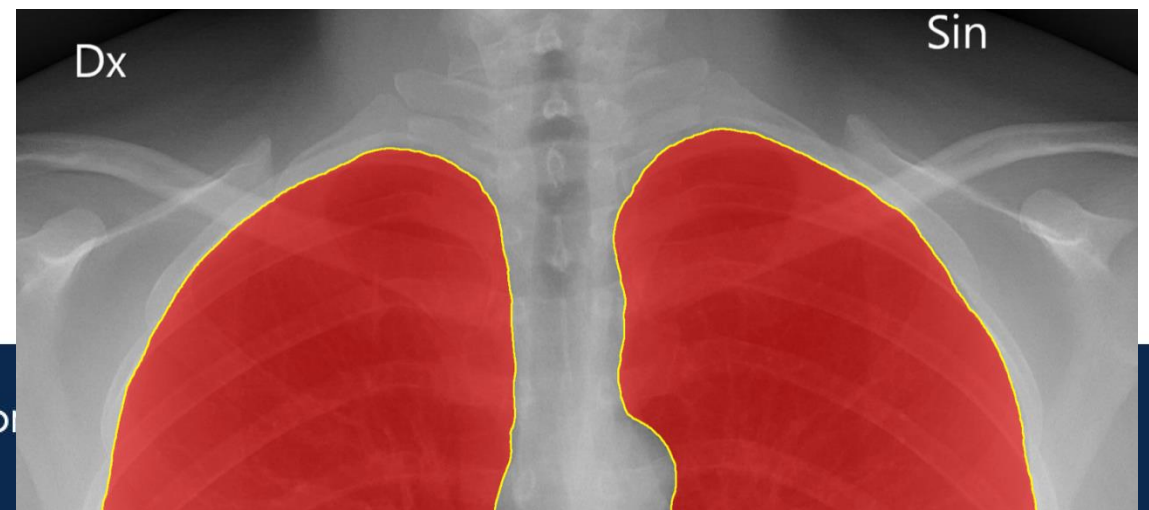
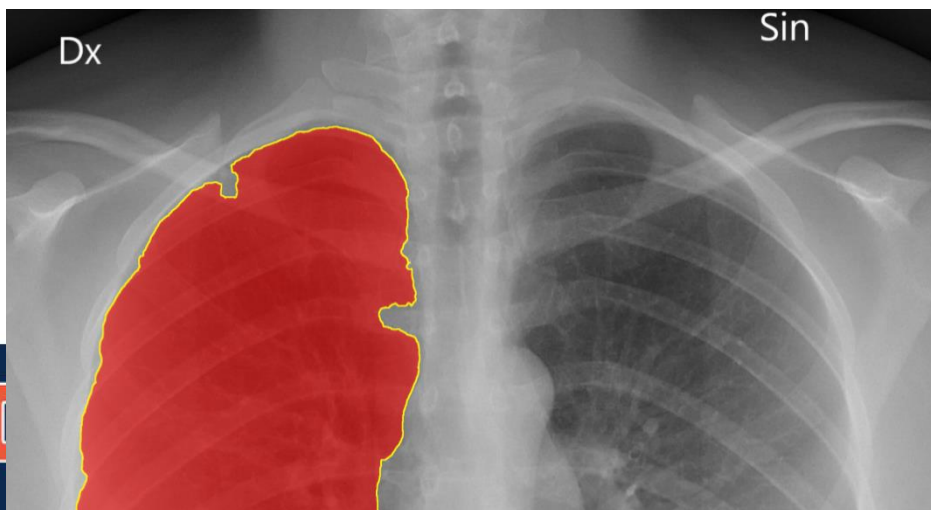
Challenges of Deep Learning

- Researchers with diverse backgrounds and levels of expertise seek to adopt DL methods to advance their research
- Modern DL methods demand significant GPU resources, commonly provided by HPC clusters
- Regardless of their HPC expertise, researchers encounter usability, performance, portability, and reproducibility obstacles
- Today's clunky interface to HPC resources present a major barrier to entry



Example Scenarios

- AlphaFold2 - protein shape prediction
- OpenProteinSet – large dataset of multiple sequence alignments (MSAs)
- Want to fine-tune on some specific proteins
- Meta's SAM3 – image segmentation
- Want to train for specific domains (before/after)



Example Scenarios

- AlphaFold2 - protein shape prediction
- OpenProteinSet – large dataset of multiple sequence alignments (MSAs)
- Want to fine-tune on some specific proteins

Software installation

Run across many GPUs

Data management

Job management

Provenance management



Example Scenario - Points of Failure

Software installation

Launcher, scheduler, drivers, environment (e.g. CUDA, Conda)

Run across many GPUs

Data management

Large data volumes, filesystems, consistency

Job management

Provenance management

Maximum walltimes, checkpointing and restarting, monitoring of progress



Diamond Project Goals

- A web-based platform for training and fine-tuning, across GPU clusters (e.g. NSF ACCESS HPCs)
- Portable and reproducible environments across heterogeneous systems (build once, run anywhere)
- Seamless environment configuration, job running, and monitoring
- Leverage Globus Compute for account management and provisioning

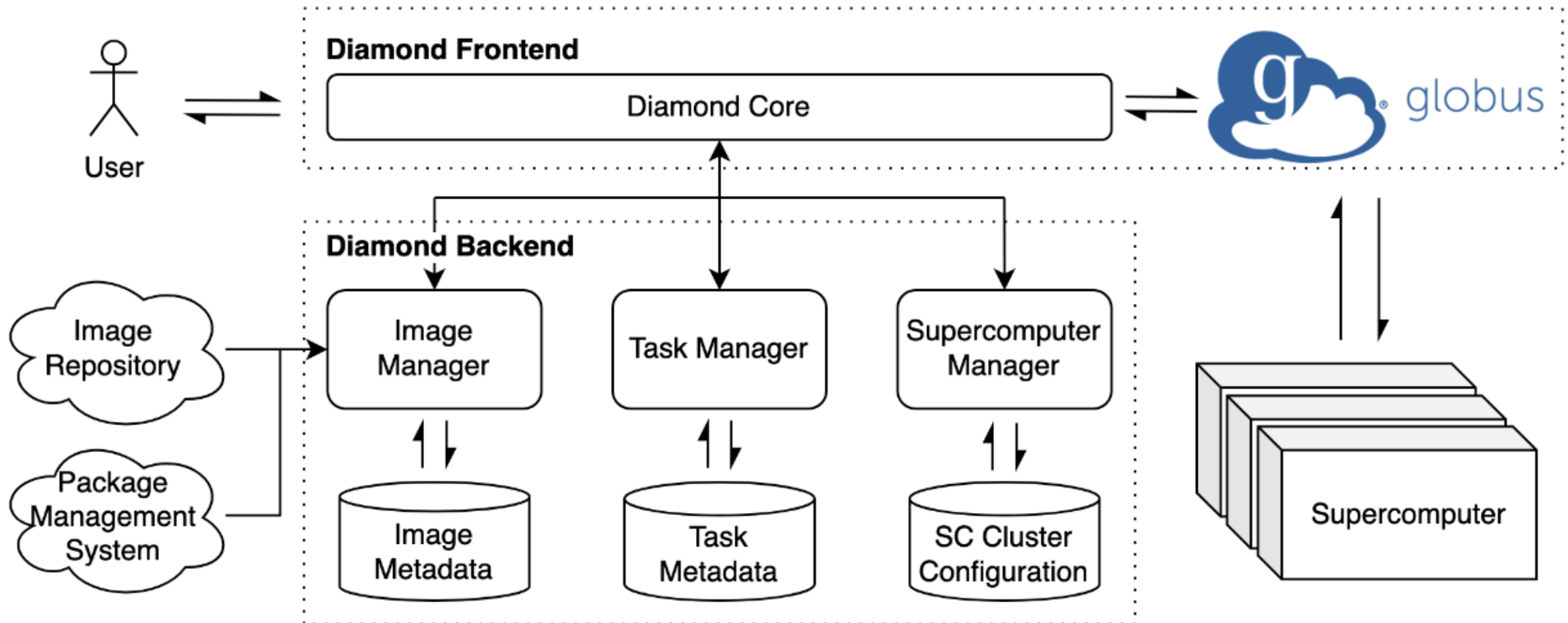


Key Features

- Endpoint Manager
 - view & configure Compute endpoints
 - reusable environments across clusters
 - cross-cluster portability
 - (scheduler, launcher, affinity)
- Task Templates
 - preconfigured jobs
- Fault-Tolerant Task Manager
 - checkpoint recovery, email alerts
- Integrated Data Management
 - Globus Transfer & Search
- Real-Time GPU Monitoring
 - NVML/DCGM metrics, WEB dashboard



Architecture Diagram

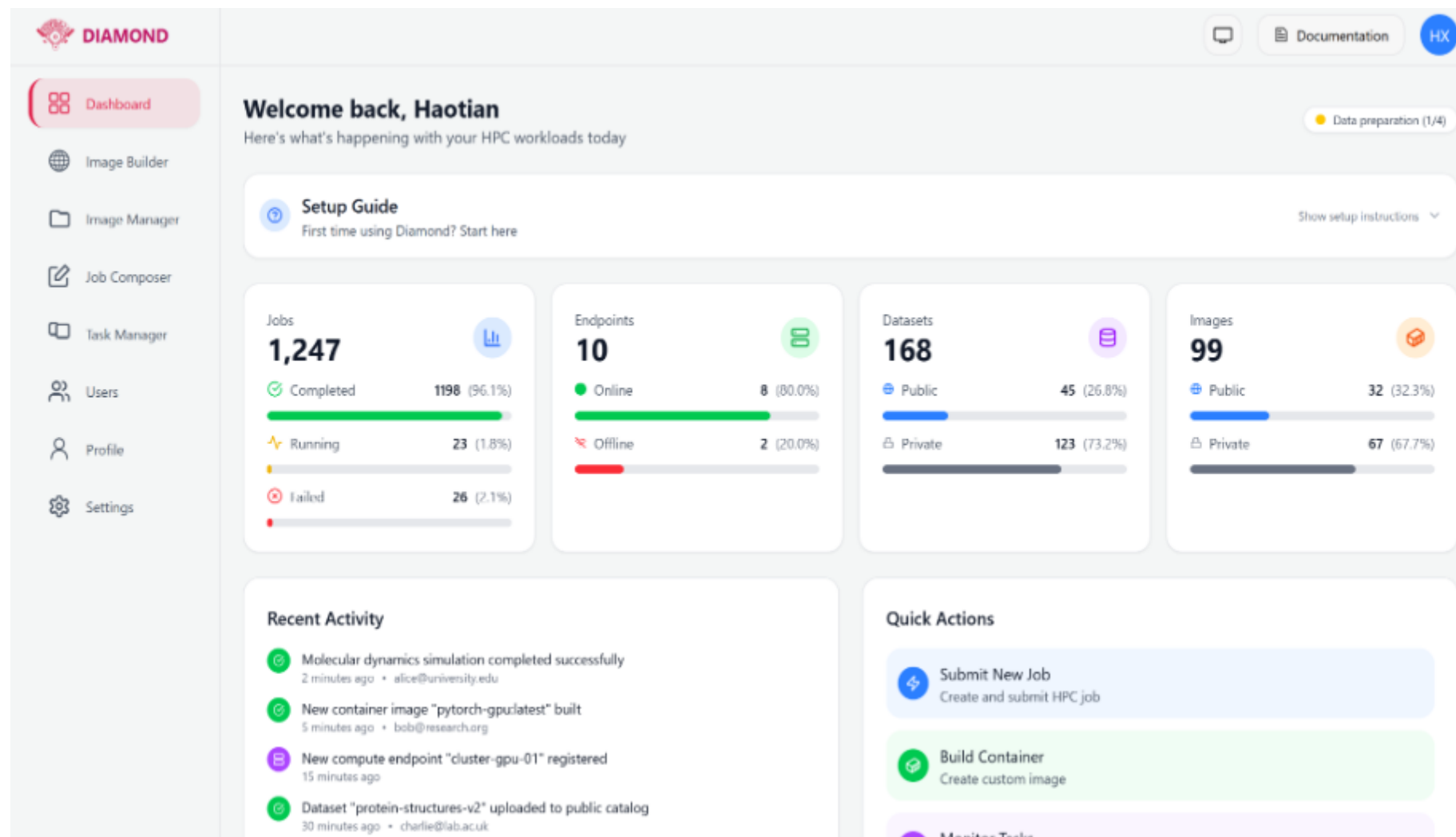


Supported HPCs

- Container built on Frontera reused seamlessly on Perlmutter & Lonestar6
- GH200-based clusters (DeltaAI, Vista) require rebuild due to CPU arch
 - **Supported:**
 - TACC: Frontera, LS6
 - NCSA: Delta (MEP)
 - NERSC: Perlmutter
 - RCAC: Anvil (MEP)
 - **In Progress:**
 - TACC: Vista
 - NCSA: DeltaAI
 - ANL: Polaris, Sophia



Dashboard



Container Builder

Image Builder

Select the HPC endpoint and configure compute resource allocation settings

Step 1 of 4 25%

HPC Endpoint

Select your compute endpoint

frontera-test (online)

Partition

Choose compute partition

rtx

HPC Account

Your compute account

A-ccsc

Or enter manually

A-ccsc

Reservation (Optional)

Reserved compute resources

Reservation Name

Enter reservation name (optional)

Leave empty if no reservation is required

[< Previous](#)[Next >](#)

Endpoint Configuration

Image Builder

Configure your container name, build location, and base image

Step 2 of 4 50%

Container Name

my-container

Choose a unique name for your container image

Build Location (Optional)

/home/user/builds

Directory where the container will be built on the HPC system

Base Image

ubuntu:22.04

Base container image to build from (e.g., ubuntu:22.04, centos:8, python:3.9)

[< Previous](#)[Next >](#)

Container Metadata



Container Builder

Image Builder

Configure build dependencies, environment variables, and commands

Step 3 of 4 75%

Dependencies (Optional)

```
# Example dependencies
apt-get update && apt-get install -y \
  python3 \
  python3-pip \
  gcc \
  make
pip3 install numpy scipy matplotlib
```

System packages and dependencies to install (shell commands)

Environment Variables (Optional)

```
# Example environment variables
export PATH="/opt/myapp/bin:$PATH"
export PYTHONPATH="/opt/myapp/lib:$PYTHONPATH"
export OMP_NUM_THREADS=4
```

< Previous

Next >

Dependencies/ENVs

Image Builder

Review your configuration before starting the build process

Step 4 of 4 100%

Build Process Information

The build process may take several minutes to complete depending on your configuration. You'll be able to monitor the build logs in real-time once the process starts.

Computing Resources

Endpoint:	ebf62610-98e3-4733-87e3-bec3c72dd7fd
Partition:	rtx
Account:	A-ccsc
Reservation:	None

Container Configuration

< Previous

Start Build

Review and Build



Container Builder

The screenshot displays the DIAMOND Container Builder web interface. On the left is a sidebar with navigation links: Dashboard, Images (highlighted), Datasets, Tasks, Profile, and Settings. The top header includes the DIAMOND logo, a settings icon, a Documentation link, and a user profile icon labeled 'ZZ'. The main content area is titled 'Container Images' with a subtitle 'Build, manage, and deploy your container images across HPC resources'. A red button '+ Build New Image' is in the top right. Below, a card for 'deepseed-img-3' (status: Unknown) shows details: BASE IMAGE 'haotianxie/deepspeed-base:0.7', TASK ID 'fb308e6c-56d9-488a-9f9c-0bcf30a7aeb7', and LOCATION '/work2/00946/zzhang/frontera/diamond'. A red 'Delete' button is on the right.

Image Name	Status	Base Image	Task ID	Location	Actions
deepseed-img-3	Unknown	haotianxie/deepspeed-base:0.7	fb308e6c-56d9-488a-9f9c-0bcf30a7aeb7	/work2/00946/zzhang/frontera/diamond	Delete

New image can be used for tasks.



Job Composition

The screenshot shows a 'Submit New Task' form with the following fields and values:

- Task Name ***: task_ft-opt_oct11_01
- Endpoint ***: tacc-frontera (online)
- Partition ***: rtx-dev
- Account ***: deep-learning-at-sca
- Container ***: c_oct10_01
- Account Manual Input**: deep-learning-at-sca
- Dataset (Optional)**: No dataset
- Reservation (Optional)**: Enter reservation
- Number of Nodes**: 1
- Time Duration (HH:MM:SS) ***: 00:30:00
- Task Command (Optional)**: `deepspeed --num_gpus 4 /workspace/DeepSpeedExamples/applications/DeepSpeed-Chat/training/step1_supervised_finetuning/main.py --model_name_or_path /home1/00946/zzhang/.cache/huggingface/hub/models--facebook--opt-125m/`

Buttons at the bottom: Cancel, Submit Task

- Resource Selection
 - endpoint, partition, account
- Execution Context
 - no. of nodes, container environment, optional log path
- Command Details
 - task command line, e.g.
`python train.py --epochs 10`



Job Monitoring

Tasks

Submit and manage computational tasks across HPC resources

+ Submit New Task

Total Tasks
1

Running
0

Completed
0

Pending
0

Failed
0

All Status

t_oct11_01 COMPLETING

Delete

ENDPOINT
89f65975-029d-4fa7-9312-faed27506150

TASK ID
69ecc907-f802-4526-b4fb-154f77fba7fa

CREATED
10/13/2025, 4:32:37 PM

STDOUT LOG
[\\$WORK/diamond/logs/t_oct11_01.stdout](#)

STDERR LOG
[\\$WORK/diamond/logs/t_oct11_01.stderr](#)



Upcoming Features

- Generalized Artifacts Pass job outputs into subsequent jobs
(fine-tuned model image -> vLLM server)
- Dataset transfer & management
- diamondhpc.ai

