

Proposal



Project 5: Dashboard for tracking WESTPA simulation progress

Name: Edison Law

Email: edisonlaw16@gmail.com

GitHub: edison16a

LinkedIn: <https://www.linkedin.com/in/edison-law-04a3ab290/>

School: UC Berkeley

Primary Language: English

Synopsis & Benefits to Community.....	2
Technical Proposal.....	3
Schedule / Deliverables.....	7
Related Work.....	9
Biographical Information.....	9
After GSoC 2026.....	11

Synopsis

This project creates a progress dashboard for WESTPA simulations that would report simulation progress and time estimates.

The dashboard will use WESTPA's work managers to monitor simulation progress. The first version will focus on a CLI tool for WESTPA progress tracking.

My goal is to implement a dashboard that is useful and maintainable. The main focus will be on a WESTPA CLI monitoring tool that can report the following:

- Current iteration progress
- Completed iteration summaries
- Time & ETA information
- Progress information derived from WESTPA run data

I want to keep the first version simple. My first goal will be a CLI monitoring tool that could later support a graphical user interface.

Benefits to Community

This project would make WESTPA simulations easier to monitor.

WESTPA already stores some simulation information in `west.t.h5`, and the codebase already has a simulation manager, work managers, and a data manager that can help track simulations.

The dashboard would allow users to directly see:

- Current iteration number
- Last completed iteration
- Simple progress statistics
- Recent iteration duration
- Approximate completion time

This would make WESTPA runs easier to monitor. This project also creates a good starting point for future work on a more complex GUI and MDAnalysis integration.

Technical Proposal

1. Project Scope

The project could include a WESTPA CLI tool and an MDAnalysis module. I plan to prioritize the WESTPA CLI monitoring tool first.

The first goal is:

- A new CLI monitoring tool in WESTPA

The CLI tool will have:

- Stable progress reporting
- Completed iteration information
- ETA / timing estimates

Further goals:

- More detailed iteration progress
- A more complex user interface

2. Main Idea

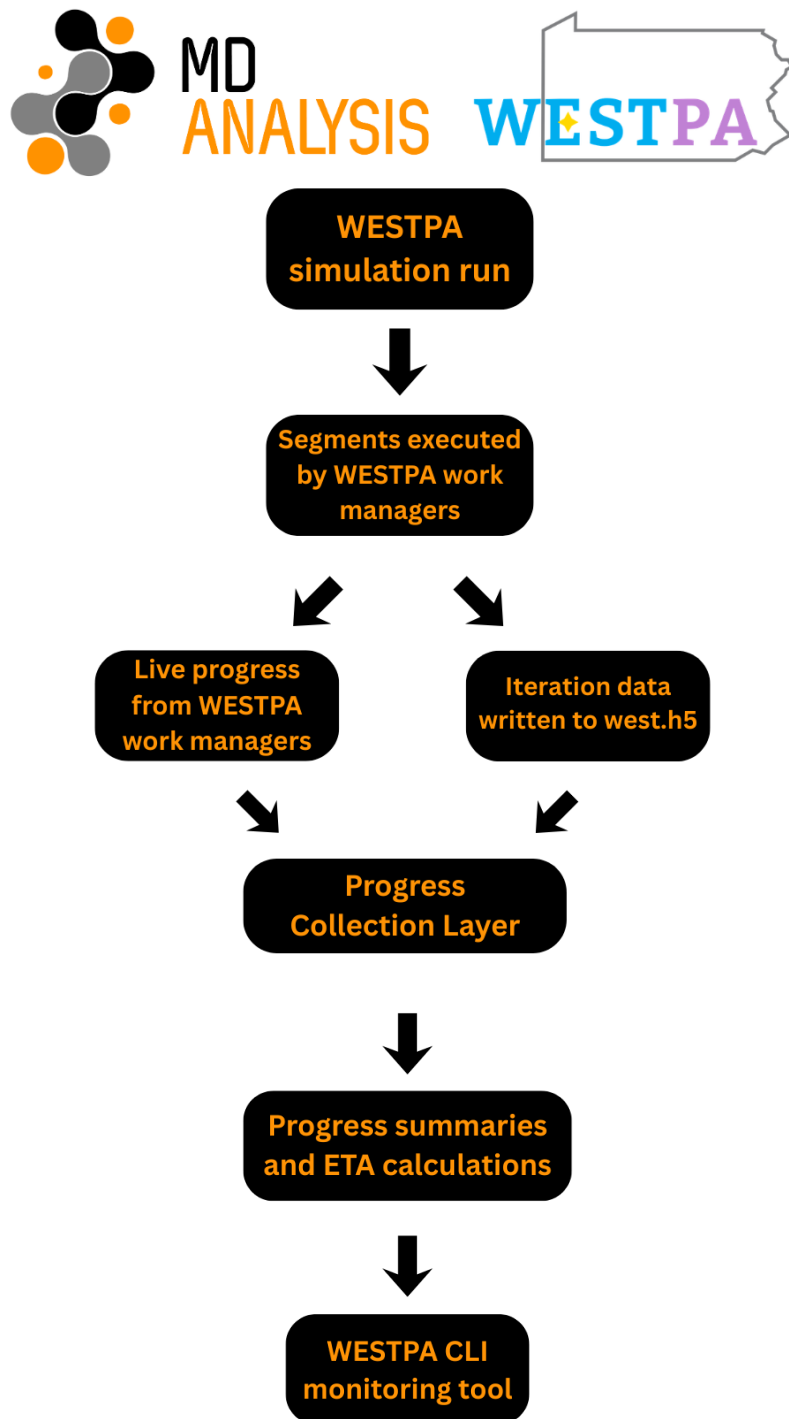
Progress information in WESTPA includes:

1. Completed iteration data from west.h5
2. Live execution data from the simulation manager and work managers

The first version will be designed around:

- Reading west.h5

- Summarizing completed iterations
- Reporting recent progress
- Estimating timing from recorded iteration behavior



3. Proposed CLI tool

I propose a WESTPA CLI command for progress tracking.

The new command would provide a monitoring view of a running or recently updated WESTPA simulation.

The CLI tool would report:

- Current / latest completed iteration
- Total requested iterations if available
- Elapsed time for recent iterations
- Simple ETA
- Basic run summary information from summary
- Iteration data such as segment counts or progress-coordinate summaries



\$ w_progress

Status: RUNNING

Current iteration: 18 / 100
Last completed iteration: 17
Segments in current iter: 200
Segments completed: 126
Segments remaining: 74

Recent iteration times:
Iteration 15: 00:02:11
Iteration 16: 00:02:08
Iteration 17: 00:02:14

Estimated time for current
iteration: 00:01:17
Estimated remaining run time:
02:56:40

Last completed iteration summary:
Populated bins: 12
Total segment weight: 1.0
Mean progress coordinate: 1.84
Max progress coordinate: 3.12

4. HDF5-based monitoring

The core monitoring logic will utilize west.h5.

WESTPA stores run information in west.h5, including iteration groups and summary information.

The west.h5 reader will extract:

- Latest completed iteration
- Requested run length
- Recent iteration durations
- Summary data
- Selected fields such as pcoord and segment metadata

I chose west.h5 as a data source as it is:

- Reliable
- Already part of WESTPA's core architecture
- Easier to test than deep live instrumentation
- Enough to provide useful iteration-level progress and timing summaries

5. ETA estimation

The CLI tool should provide a basic ETA.

The first ETA calculation would be based on:

- Recent completed iteration times
- Average iteration duration
- Current iteration number and requested total iterations

I plan to keep the ETA logic conservative and easy to understand.

6. Live-progress extension (Further Goal)

WESTPA work managers, such as ZMQ and Python multiprocessing, can also be utilized.

I want the following to be a further goal due to the 90-hour scope of the project:

- Explore whether live iteration progress can be pulled from the simulation / work-manager layer
- Use that to supplement the HDF5-based monitoring

7. MDAnalysis-side contribution (Further Goal)

If there is time after the main CLI tool is working, I will add a small helper for aggregating or tracking multiple simulations.

8. Testing

I will add tests for:

- HDF5 parsing
- Progress calculations
- ETA calculations
- CLI outputs

9. Documentation

I will document:

- What fields the tool reports
- How ETA is calculated
- Limitations of the first version
- Examples of expected output

Schedule / Deliverables

Community Bonding Period

Review relevant WESTPA code paths, such as `w_run`, the simulation manager, and the data manager

Confirm with mentors the preferred CLI interface and scope

Confirm expected MDAnalysis-side aggregation

Build small local code samples, reading `west.h5` and summarizing iterations

Finalize the first version output format and ETA scope

Weeks 1 to 2

Build HDF5 parsing to read `west.h5`

Extract latest completed iteration, summary fields, and iteration timing data

Add unit tests for parsing and summary extraction

Checkpoint: The tool can read WESTPA run state from `west.h5`

Weeks 3 to 4

Build the CLI command

Add CLI output and interface

Add basic ETA logic

Test the command on local WESTPA runs

Checkpoint: Working MVP CLI tool which shows completed progress and ETA

Weeks 5 to 6

Improve formatting and edge-case handling

Add tests for the CLI tool to check behavior when there is incomplete or partial data

Try to see if live-progress data can be pulled from the runtime state

Start a small MDAnalysis side helper if the CLI tool is working well

Checkpoint: Stable first-version CLI tool with tests

Weeks 7 to 8

Finish documentation

Polish code and CLI behavior

Address mentor feedback

Complete further goals if possible

Weeks 9 to 10

Fix bugs found through testing or mentor feedback

Finish any remaining cleanup in code or documentation

Final polish of code

Buffer time in case earlier tasks take longer than expected

Deliverables

New CLI tool for WESTPA progress tracking

HDF5-based progress and timing summaries & ETA reporting

Basic documentation

Tests for parsing, progress, and ETA logic

MDAnalysis aggregation helper if time allows

Related Work

WESTPA already has the main infrastructure that this project would build on. It already manages work through its work managers and stores run data in west.h5, including iteration data and summary information.

The project would build a dashboard tool focused on progress tracking and time estimates that uses WESTPA's existing infrastructure.

Biographical Information

I'm Edison from the U.S. (San Francisco Bay Area).

I'm a student interested in Computer Science & Biomedical Engineering. I'm excited to start contributing to MDAnalysis.

Background and Programming Languages

- Python (NumPy), JavaScript/TypeScript (Next.js), HTML/CSS, Swift, and some ML/AI experience.
- This is my first year applying to GSoC, and I am excited about open source and want to be more involved.
- I have experience writing tests in hackathon/internship projects, and I'm comfortable debugging and iterating based on feedback.

Projects:

FlameSense - Winner of Los Altos Hacks 2025

- Competed against 350+ competitors. Built a wildfire-risk prediction system using a sequential neural network trained on historical burn data and real-time weather inputs (temperature, humidity, wind) to forecast spread for emergency planning. Built in HTML/CSS, Python, and utilized a .keras model

SensePlan - Winner of Luma A2A - Agents Hackathon 2025

- Competed against 130+ professional engineers. Built an autonomous AI agent that discovers relevant medical clinics or health services via web data, evaluates options based on user preferences, and places appointments end-to-end using programmatic voice calls (Bright Data + VAPI).

CallSense - Winner of Milpitas Hacks 2025

- Competed against 150+ competitors. Built an AI assistant for 911 call centers that transcribes live emergency calls, extracts critical details, prioritizes incidents by urgency, and suggests targeted follow-up questions to support faster, more accurate dispatch decisions.

CS Experience:

Software Engineer Intern at Tanius Technology (Trading Firm): Worked on data analysis and systems (e.g., analyzing relationships in market data; built an iOS app + a C# web server)

Why computational physics/chemistry/materials mean a lot to me

I've had severe food allergies since I was a child. I've been interested in biology/chemistry and medical science because I've wanted to learn more about molecular interactions. I'm

especially interested in software projects that help analyze simulations to eventually benefit the scientific community and research.

Scientific experience

- **UCLA Applications of Nanoscience:** Gained wet lab experience across biopolymers, microfluidics, nanotoxicity, water filtration, and prototyping a nanotech wound-care gel (“NanoKnit”). My project involved simulation/data analysis work, and I’m interested in contributing to MDAnalysis.
- **Stanford Inspiring Future Scientists through Shadowing:** Collaborated with Stanford Ph.D. candidate researchers in chemistry, biology, and genetics, assisting with experimental setup and observing advanced instrumentation and simulations.

After the end of GSoC 2026

After the end of GSoC 2026, I would like to continue contributing to MDAnalysis and WESTPA. I want to keep contributing through fixes, cleanup, or improvements related to this project.

I would also love to keep learning the WESTPA and MDAnalysis codebases and contribute to other issues or small features.