

2026-27 HIVE Summer Internship Project

Visualising How Artificial Intelligence Schedules Jobs in Cloud Computing

Primary Academic Supervisor

A/Prof. Elham Mardaneh

Project Background

Cloud computing means running software and storing data on shared remote servers rather than on a single local computer. When many computing jobs compete for those servers, the system must decide which job runs on which machine and when. Each job may need different amounts of processor capacity and memory, while each machine has limits. Poor decisions can leave machines underused, overload a resource, delay jobs, or require more machines and energy than necessary.

Our current research with Ty Riekert, a 3rd Year Student in Adv Science (Physics) (Hons), asks whether reinforcement learning - a form of AI that learns by receiving rewards or penalties for its decisions - can make these scheduling decisions well. Ty has built an initial model in a controlled setting where all jobs are known before scheduling starts. He has also implemented two established learning methods and measures their performance using the number of machines used, missed deadlines, and the highest level of resource use.

The next challenge is understanding why a learned decision rule makes good or poor choices. A final score can tell us which method performs better, but not where its behaviour starts to go wrong. For example, a method may wait even when a useful machine is available, spread jobs in a way that wastes remaining capacity, or show how an earlier choice about where to place a job reduces the capacity available for later jobs, eventually causing a delay or missed deadline.

This project will make that decision process visible. Interactive views will show jobs, machines, available capacity, possible choices, and the consequences of each decision over time. We will use these views to

identify repeated failure patterns, improve the learning model, and test whether the changes produce better schedules. The internship is intended to support a research publication and prepare a later Honours thesis on the more realistic case where jobs arrive over time and future demand is unknown.

Project Description, Expected Outputs, Possible Stretch Goals

The aim is to build an interactive visual analysis tool that helps researchers understand how artificial-intelligence scheduling methods make decisions, not simply to display their results.

The main output will use the HIVE Tiled Display, a large high-resolution screen wall. It will show, at the same time: jobs waiting or running; which machine each job uses; how much processor and memory capacity remains; approaching deadlines; the choices available to the learning method; and the choice it makes. A shared time control will let users play, pause, step forward or backward, and compare two learning methods with a conventional non-learning scheduling method on the same set of jobs.

The visualisation will be used to locate where undesirable behaviour begins - for example, unnecessary waiting, opening an extra machine, creating a resource bottleneck, or leaving small unusable pieces of capacity across several machines. These observations will be converted into a specific change to the learning model. We will then rerun the same experiments to test whether the change improves performance.

A second prototype will use the HIVE Wedge, a stereoscopic three-dimensional display, to test whether a machine-by-time-by-resource view reveals patterns that are hard to see in two dimensions.

Expected outputs are an interactive HIVE application, reproducible experiment data, a comparison of scheduling methods, one tested model improvement, documented software, a final report and presentation, and material suitable for developing a research paper. The internship will stay within the controlled setting. The harder real-time case, where new jobs arrive and the future is unknown, is reserved as a clear Honours thesis direction.

Links to background reading and any relevant recent work in the field

- Load balancing and task scheduling methods in cloud computing – a review: <https://ieeexplore.ieee.org/document/10829119>
- A systematic literature review of machine learning-based resource allocation techniques in cloud computing:
<https://link.springer.com/article/10.1007/s00607-025-01526-8>
- Learning Scheduling Algorithms for Data Processing Clusters:
<https://arxiv.org/abs/1810.01963>
- A Hierarchical Framework of Cloud Resource Allocation and Power Management Using Deep Reinforcement Learning:
<https://arxiv.org/abs/1703.04221>
- Proximal Policy Optimization Algorithms:
<https://arxiv.org/abs/1707.06347>

What type of visualisation will the student develop or produce?

The student will build an interactive visual-analytics tool that turns a cloud schedule into an animated scene. Each computing job will be shown moving from waiting to a physical machine, while processor and memory use change over time. Users can play, pause, rewind and inspect decisions, and compare the learning-based scheduler with conventional methods side-by-side. A 3D HIVE view will show how machine use and spare capacity evolve through time.

How will the visualisation contribute to your research outcomes?

Numerical scores tell us which scheduler is better, but not why. The HIVE visualisation will let us follow a poor result back through time and see how earlier placements created later problems, for example scattered spare capacity that cannot fit an urgent job. These visual patterns will generate specific hypotheses about the learning method. We will then change the model and rerun the same workloads to test whether the identified problem is reduced.

If the project is successful, where would you consider publishing the results?

If the findings are sufficiently strong and novel, we will prepare a peer-reviewed journal or conference paper. The central research question for that paper will be: can interactive visual analysis reveal recurring reasons an artificial-intelligence scheduler performs poorly, and can those insights lead to a measurable improvement? Potential journals include Future Generation Computer Systems, IEEE Transactions on Cloud Computing, and Journal of Cloud Computing.

The internship publication would focus on the controlled setting in which all jobs are known at the start. A later Honours thesis would be a separate research stage, extending the problem to real-time arrivals where future jobs are unknown. That extension could support a second publication if it produces sufficiently new results.

Draft Project Timeline:

Week 1

Nov 16 - Full day HIVE induction

Nov 17 - Area and Project Induction with Primary supervisor

Develop project plan with HIVE and academic team

Literature and relevant project review

Project Familiarisation

Week 2

Finalise the research questions and visualisation requirements with the HIVE team. Prepare the existing scheduling experiments so that decisions, machine use, job status, processor and memory use, and performance measures are recorded consistently. Verify a conventional scheduling method for comparison with the learning-based methods. Develop the initial layout for the HIVE Tiled Display.

Week 3

Develop the first interactive visualisation for the HIVE Tiled Display.

Visualise jobs waiting, being processed and completed, together with their allocation to physical machines. Display how processor and memory capacity are being used on each machine and identify jobs approaching or missing deadlines. Test whether the complete system state can be clearly understood at any selected point in time.

Week 4

Extend the visualisation to show the scheduling decision itself. For each decision, show which job-to-machine choices were possible, which were not possible, which choice was made, and how strongly the learning-based method preferred the available choices. Add the ability to select a job or machine and follow it through the scheduling process.

Week 5

Add interactive time controls, including play, pause, rewind and step forward/backward. Develop synchronised side-by-side comparison of the two learning-based methods and the conventional scheduling method using the same workload. Test and refine the complete visualisation on the HIVE Tiled Display with HIVE staff.

Week 6

Develop a complementary three-dimensional visualisation for the HIVE Wedge showing how machine use and available computing resources

change over time. Compare the three-dimensional and two-dimensional representations to determine whether the 3D view reveals useful patterns that are difficult to identify in the Tiled Display visualisation. Retain and refine the 3D component only where it provides additional analytical value.

Week 7

Run controlled experiments using identical workloads across the different scheduling methods. Use the HIVE visualisation to investigate why particular schedules perform well or poorly and identify repeated patterns associated with unnecessary waiting, inefficient use of machine capacity, activation of additional machines, or missed deadlines. Formulate a small number of specific explanations that can be tested experimentally.

Week 8

Select one important behaviour identified through the visual analysis and modify the learning-based scheduling method to address it. Repeat the experiments and compare the original and revised approaches using both the visualisation and numerical performance measures. Over the next two weeks (9 and 10), finalise the HIVE demonstration, report, presentation, software documentation and publication-quality figures. Assess the potential for a research publication and define the subsequent Honours research direction: online cloud resource management, where new computing jobs arrive over time and future jobs are not known in advance.

Week 9

Focus on report writing and presentation (including demo) preparation

Week 10

Focus on report writing and presentation (including demo) preparation

5th Feb Final Showcase Day and final report due

Student Experience and Supervision:

How often will you meet with the student over the 10-week period?

Once per week. There might be periods where I am travelling, but Tony Mathew (research associate) will be a co-supervisor and able to cover any week that I am travelling.

Your work desk location and the location of student desk

The student will be located in Building 314 (we can only request space once the internship has been approved and are not allowed to “reserve” space prior).

Student Attributes:**Please indicate any preference for student's academic discipline or field of study**

Computer Science, Applied Science, Operations Research, or a related quantitative discipline.

What competencies are required to start this project

Beginner - Machine learning/optimisation. Some familiarity is useful, but the student should be able to develop these skills during the internship. We should not require an advanced background.

Intermediate - Data structures, analytics, statistical modelling

Intermediate - Python programming. This is the most important technical starting competency because the current scheduling and learning code is implemented computationally and the visualisation will also be Python-based.

Do you have any other student attributes you think are important to the project?

The student should enjoy problem-solving and be curious about how computational methods behave. They should be willing to learn new visualisation tools, work systematically with computational experiments, and investigate why an algorithm produces particular results rather than only reporting numerical scores. Good communication and willingness to work collaboratively with the academic and HIVE teams are also important.