

Alan Zamani

GPU Platform & Orchestration Engineer · Python · Slurm · Kubernetes · Ansible · MCP

Perth, Australia | alan.zamani63@gmail.com | alanzamani.dev | github.com/azamani84
orcid.org/0000-0002-1444-2817

Summary

Nine years building production systems in Australia, eight at a Tier-1 national supercomputing centre. Designs, builds and operates multi-tenant GPU scheduling platforms: allocation and fair-share policies, quota and walltime enforcement, telemetry and self-service tooling across heterogeneous clusters. Built vScheduler, an open-source GPU session orchestrator in production since 2022 across 51 GPU nodes, with the accompanying paper published at ISC High Performance 2026. Now moving it to Kubernetes with a gated CI/CD pipeline, and building MCP servers that let LLM agents operate production infrastructure safely.

Experience

HPC Software Engineer, Visualisation Specialist, Pawsey Supercomputing Research Centre (CSIRO) – Perth, Australia Mar 2018 – present

One of Australia's two national supercomputing centres. Designs and operates the interactive GPU platform behind the centre's visualisation services: scheduling, remote access and the infrastructure underneath it.

- Designed, built and shipped vScheduler (open source, github.com/PawseySC/vScheduler), a Python scheduling and orchestration service allocating GPU-backed interactive sessions across two production clusters (a Linux HPC partition and a Windows GPU cluster). In production since 2022 across 51 GPU nodes, serving 150-200 users and about 500 sessions (5,000 session-hours) a month.
- Implemented two allocation models in one system: on-demand scheduling via a custom load balancer that ranks nodes on live utilisation and health, plus reservation-based allocation validated against a booking service. No open-source alternative combines both; the capability previously required commercial products such as AWS NICE DCV and NICE EnginFrame (an estimated USD 80,000 in licences at 200 users).
- Built the control plane end to end: socket-based telemetry agents streaming session and health events, a MySQL state store as the single source of truth, systemd services, and quota, walltime and fair-share enforcement.
- Shipped a Slurm-style operator CLI (vmanage, valloc, vkill, vquota, vinfo, vreport) with HTML reports.
- Built automated hybrid HPC pipelines for Ansys Fluent and Ansys Mechanical: users pre- and post-process interactively on the Windows GPU cluster while solves run on the supercomputer, with Slurm jobs submitted on their behalf from the GUI and results returned automatically over key-based SSH or Ansys Remote Solve Manager (RSM), fully scripted and transparent to users.
- Architected for failure isolation and portability (separated control and data planes, per-site deployable modules); owns releases and operations, shipping periodically and recovering monthly post-maintenance breakages (OS patches, Python upgrades, database connectivity) the same day.
- Provisions the estate with Ansible across bare metal, KVM/Proxmox VMs as well as an in-house OpenStack cloud.
- Engineered zero-touch, image-based provisioning for the 19-node Windows GPU cluster on open-source tooling (iPXE, wimboot): nodes network-boot, apply WIM/FFU images with unattended configuration and Group Policy, and return to service fully configured, every build defined as code in Git, saving an estimated USD 20,000.
- Operates the clientless remote-access gateway: Apache Guacamole brokering persistent RDP, VNC and SSH sessions in the browser, behind a reverse proxy and institutional SSO.
- Operates the centre's render farm and NVIDIA Omniverse environment.
- Mentored two interns building data-science and digital rock physics applications on heavy numerical compute, and wrote the training material and public Confluence documentation that onboard users to the platform.

Software Engineer, Stake – Sydney, Australia Mar 2017 – Feb 2018

Built the customer-facing trading platform in a regulated financial-services environment.

- Built account opening, order placement and portfolio monitoring in Angular on REST APIs and market-data feeds.

Publications

A Scheduler System for Interactive Visualization on HPC Resources – ISC High Performance 2026, Hamburg June 2026

First author. System design, implementation and evaluation.

- Paper: doi.org/10.23919/ISC.2026.11520479 | Video: youtube.com/watch?v=ED_c447rv3Y

Projects

vScheduler

2022 – present

An open-source HPC scheduler for interactive GPU sessions, developed in-house and in production at Pawsey Supercomputing Research Centre since 2022 across 51 GPU nodes. Unifies a Linux HPC partition and a Windows GPU cluster under one centralised scheduling model, with both on-demand and reservation-based allocation, integrating Apache Guacamole and LibreBooking through their REST APIs. Source: github.com/PawseySC/vScheduler.

- Stack: Python, On-Demand/Booking-Based Scheduler, MySQL, Linux, ETL, Load Balance, Socket Communication, Orchestration, Remote Desktop Protocol, REST API, Session Management, Resource Allocation, Multi-tenant, HPC

vScheduler on Kubernetes

In development

The full vScheduler stack moved from Ansible-provisioned VMs to Kubernetes and deployed declaratively from one Kustomize base, with dev and prod overlays. Credentials live in Secrets and settings in ConfigMaps, the shared database persists on a PersistentVolumeClaim, and schema setup runs as an idempotent Job. A Service per session node lets the scheduler reach each one through cluster DNS by the same naming pattern it uses in production.

- Stack: Kubernetes, Kustomize, Docker, MariaDB, Apache Guacamole, LibreBooking

vScheduler Delivery Pipeline

In development

A self-hosted CI/CD pipeline for vScheduler on Kubernetes, built on Gitea Actions with GitHub Actions-compatible workflows. Required checks gate every merge; each image is built once, tagged by commit and promoted unchanged from dev through staging to a protected production environment, with a smoke test gating each promotion and automatic rollback on failure. Deploys need only kubectl, so it targets any Kubernetes cluster, on premises or cloud.

- Stack: CI/CD, GitHub Actions, Gitea, k3s, Kustomize, Container registry

vScheduler MCP Server

In development

A Model Context Protocol server that lets LLM agents such as Claude Code operate vScheduler in natural language. Built on the official Python MCP SDK over streamable HTTP, it generates one typed tool per command from a YAML allowlist that validates every argument, and runs each call over key-based SSH from its own container. State-changing tools are flagged destructive so the client confirms first; results are checked, not trusted.

- Stack: Model Context Protocol, MCP server, AI agent tool, LLM integration

ParaView MCP for Visualisation

Proof of concept, Nov 2025

Applying LLNL's open-source ParaView-MCP to interactive visualisation at Pawsey Supercomputing Research Centre: an MCP server that drives the ParaView application from natural language, where the agent reads the viewport back as an image and refines the scene from what it sees. It runs against a multi-client pvserver with the ParaView GUI attached, so a researcher watches the render being built as they describe it. Trialled to find where natural-language control helps people who are not ParaView experts, and where it still needs one.

- Stack: Model Context Protocol, ParaView, Scientific visualisation, Multimodal LLM, Python

Skills

HPC operations: Slurm, GPU scheduling, Interactive HPC, Bare-metal provisioning (iPXE), openDCIM

Automation & IaC: Ansible, Infrastructure as Code (IaC), GitHub Actions/Jenkins

Engineering practice: Modern software development lifecycle (SDLC), Agile/Scrum (Jira, Confluence), Git workflows (GitHub, GitLab, Gitea), CI/CD with gated promotion and automatic rollback, Containerised delivery, Versioned releases, Documentation

Containers, orchestration & virtualisation: Kubernetes, Docker/Helm, Kustomize, KVM/Proxmox

Platforms & observability: Linux, OpenStack private cloud, S3 object storage, Prometheus/Grafana

Languages & frameworks: Python, C++ , Bash, TypeScript/Next.js, React/Angular

AI tooling: Claude Code, GitHub Copilot, OpenAI Codex, Ollama

Education

UNSW Sydney, MSc in Petroleum Engineering

2015 – 2017

- Thesis: [Pore network modelling for coal seams](#)
- Computational research on 3D numerical flow simulation over networks modelled from large 3D datasets (micro-CT imaging), validated and benchmarked against an expensive reference (Lattice Boltzmann).

Certifications

Certified Kubernetes Administrator: Cloud Native Computing Foundation (in progress)