VIDHUN VIJAYAKUMAR SUJA

ML Systems Engineer · AI Infrastructure · LLM Inference · Systems Research

Edinburgh, UK | vsvidhun06@gmail.com | linkedin.com/in/vidhun-suja

github.com/vsvidhun06-blip |

Right to Work: UK Graduate Route visa (valid through Oct 2028)

SUMMARY

ML Systems and Research Software Engineer with an MSc in Software Engineering from Heriot-Watt University, focused on retrieval, RAG evaluation and LLM inference. As a research software engineering intern in a five-person team, built hybrid retrieval with reranking, citation-grounded answering with abstention, and recoverable ingestion, measuring ranking quality, grounding, latency and failure recovery. Personal projects: **Mini-vLLM/CARL**, an educational inference engine with an adaptive runtime controller, and **WEV-SMT**, an SMT-based verifier for weak-memory dependency analysis.

PUBLICATIONS

CARL: Online Contextual Bandit Control for Adaptive LLM Serving. V. Vijayakumar Suja. *Manuscript submitted to MLSys 2027; under review.* Adaptive runtime scheduling for experimental LLM inference, evaluated on Mini-vLLM.

WEV-SMT - SMT Verifier for C/C++11 Thin-Air Weak-Memory Semantics. V. Vijayakumar Suja, M. Doko. *Manuscript submitted to TACAS 2027; under review.* Dependency analysis with SMT; proofs in Isabelle/HOL.

TECHNICAL SKILLS

ML Systems & LLM Inference: PyTorch, Triton, CUDA Graphs, LLM serving, inference optimisation, continuous batching, paged KV cache, chunked prefill, prefix caching, FlashAttention, speculative decoding, W8A8 quantisation, multi-LoRA, contextual bandits (LinUCB), CUDA-event benchmarking

Retrieval & Applied AI: BM25, dense embeddings, hybrid retrieval, cross-encoder reranking, citation grounding

Languages: Python, C/C++, Java, TypeScript, SQL

Backend & Observability: FastAPI, Spring Boot, Kafka, Redis, PostgreSQL, Docker, Prometheus, Grafana, Locust, GitHub Actions CI/CD, Zipkin

Formal Methods: Z3, SMT/OMT, Isabelle/HOL, weak memory models (C/C++11, ARM, POWER)

SELECTED PROJECTS

Mini-vLLM + CARL - Single-GPU LLM Inference Engine & Adaptive Runtime Controller Jan 2026 - Present

Stack: Python · PyTorch · Triton · CUDA Graphs · LinUCB · FastAPI · Prometheus · Grafana · Docker | GitHub

- **Inference engine:** built an experimental single-GPU LLM serving engine with continuous batching, paged KV-cache management, chunked prefill and radix-tree prefix caching. A four-request TinyLlama benchmark on an RTX 4060 Laptop GPU recorded **44.6 vs. 24.4 tok/s** for batched vs. sequential generation (1.83× throughput).
- **Kernels & instrumentation:** fused Triton RoPE kernel with numerical parity tests against PyTorch; CUDA-event kernel benchmarks and a per-step wall-clock profiler separating prefill, decode, KV allocation and scheduler overhead.
- **Observability:** Prometheus metrics and a Grafana dashboard via Docker Compose, tracking TTFT/TPOT percentiles, queue depth, KV-cache utilisation and CUDA Graph hit rate.
- **CARL - adaptive runtime control:** contextual **LinUCB** controller for batch and chunk-size selection. On a Tesla T4 under a saturated burst workload (10 paired seeds), it improved throughput over a non-learning rule by +3.76 **tok/s** (95% CI +3.29 to +4.23) but stayed below objective-matched static configurations; at a sub-critical Poisson load it trailed both on end-of-run utility.
- **Experiment reliability:** executed reproducible GPU experiments on Google Colab with incremental checkpointing, interruption recovery, and SHA-256-verified source and artifact backups.

WEV-SMT - SMT Verifier for C/C++11 Thin-Air Weak-Memory Semantics Jan 2026 - May 2026

Stack: Java · Z3 (SMT / OMT) · Isabelle/HOL · C/C++11 models · herd7 · Dat3M | GitHub | *Supervised by Dr. Marko Doko*

- Investigated information loss in instruction-local dependency summaries under composition; mechanised formal results in Isabelle/HOL for a delimited straight-line, 32-bit program fragment.
- Evaluated dependency propagation using an exact per-instruction SMT oracle: identified all 66 single-instruction fake dependencies but missed all 72 chain-form cancellations, while retaining all 171 real dependencies; the misses reflect a limitation of the prototype's dependency closure, not of SMT solving.
- Built SMT-based experimental tooling using Java and Z3 for selected weak-memory models, with witness extraction, regression tests and reproducible comparisons between original and corrected implementations.

EXPERIENCE

Research Software Engineer Intern (Retrieval & RAG), Kompetenzen Technologies Dec 2023 - Aug 2024

Document Intelligence & Retrieval Systems | Five-person research team

- Developed a retrieval pipeline combining BM25, dense embeddings and cross-encoder reranking across thousands of documents, improving **nDCG@10 from 0.58 to 0.68** against a BM25 baseline on 500 held-out queries.
- Implemented context-selection controls, sentence-level citation mapping and evidence-based abstention, reducing unsupported claims **from 18% to 9%** while maintaining 80% answer coverage on a manually reviewed evaluation set.
- Profiled cross-encoder bottlenecks with Locust and introduced bounded candidate sets, batched reranking and asynchronous request handling, reducing p95 retrieval and reranking latency **from 1.2 s to 0.8 s** under a fixed 20-user workload.
- Implemented persistent ingestion checkpoints, transactional writes and content-hash deduplication, enabling recovery from interruptions with **zero duplicate committed records** across 100 injected-failure and retry tests.

EDUCATION

MSc Software Engineering, Heriot-Watt University, Edinburgh, UK Jun 2026

Dissertation 78/100, graded *Exceptional*; modules: Machine Learning, Concurrent Systems, Formal Methods, Distributed Computing.

BTech Computer Science, College of Engineering Muttathara, Kerala, India Jul 2023