Khalil Ben Maaoui

Software Infrastructure Engineer - Distributed Systems & HPC Scheduling

khalil.benmaaoui@gmail.com | +49 176 8085 7907 | Dresden, Germany | LinkedIn | GitHub

Skills

Programming & Backend
C/C++ · Python · Bash · FastAPI · REST APIs · SQL · PostgreSQL · slurmdb · PowerShell

Systems & Infrastructure
Linux · RHEL · Slurm · SPANK · GRES / Generic Licenses · Docker · QEMU · TCP/UDP · DPDK · Networking · Wireshark · systemd / journalctl

Software Engineering
Algorithms & Data Structures · Graph Algorithms · Concurrency & Locking · System Design · OOP · CI/CD · Testing · Git · Pytest · Playwright · Claude Code / Codex

Languages
Arabic (Native), French (Fluent), English (Fluent), German (A1)

Professional Experience

Software Engineer · C/C++/Python · Backend Systems & Infrastructure 2025/04 - 2026/08 | Hybrid, Germany
SpiNNcloud

- **Massively Parallel Systems Architecture:** Contributed to runtime and infrastructure software for a neuromorphic supercomputer spanning approximately 5 million cores and 720 interconnected boards in a 2D hexagonal torus topology.
- **Allocation Systems & Technical Ownership:** Drove architecture and implementation across three separate hardware allocators for internal company use, customer deployments, and a Slurm-managed supercomputer. Coordinated evolving contracts between workload scheduling, topology, resource state, and physical machine configuration.
- **Initial Torus Allocation Algorithm:** Implemented a two-pointer, sliding-window greedy placement algorithm over the 2D torus; incorporated authoritative topology state and request-level tolerance for unavailable boards or links.
- **Slurm-Integrated Best-Fit Packing:** Developed deterministic topology-aware placement using recursive 2D free-region partitioning, a tree representation, DFS candidate traversal, region splitting, wrap-around coordinates, hardware triads, and separate handling of single-board residual capacity.
- **Fragmentation-Aware Placement Strategies:** Designed candidate scoring and compact near-square placement shapes to preserve future allocation possibilities, locality, and capacity balance. Evaluated success, utilization, fragmentation, and locality using simulations and observability; favored fast predictable heuristics over more expensive optimization approaches.
- **Hardware Request Abstraction:** Translated simple user requests such as single- or multi-board counts into physical coordinates and torus shapes. Across allocator variants, supported or evaluated exact-machine reservations, explicit coordinates and shapes, and down-board/down-link tolerance.
- **Slurm Integration & Scheduling Contracts:** Proposed and drove the migration from custom scheduling toward Slurm. Helped define an integration that preserved native select/cons_tres scheduling authority, priorities, backfilling, and fair-share, while a prolog-stage placement engine selected actual boards and injected job environment information.
- **Resource Accounting Tradeoffs:** Investigated Slurm GRES, Generic Licenses, and SPANK for a single physical topology spanning multiple Slurm nodes; analyzed node-bound resource limits, cluster-level license tradeoffs, and Linux cgroup enforcement before the final team decision.

Professional Experience (continued)

SpiNNcloud - Software Engineer (continued)

- **Inventory, Concurrency & State Consistency:** Worked through a progression from YAML-backed hardware state guarded by POSIX file locks to database-backed inventory, board availability, and allocation mappings. Used queues, thread synchronization, and transactional updates to avoid conflicting hardware assignments and maintain a single authoritative topology view.
- **Physical Machine Partitioning:** Contributed to computing board-to-board (B2B) connectivity after placement and to socket-based bitmask commands sent to board controllers to enable or disable links. Isolated single- and multi-board regions before discovery to prevent communication collisions across allocations.
- **Graph Discovery, Reachability & Routing:** Implemented C/C++ contributions to chip/board discovery, nearest-neighbor exchanges, clock-cycle-constrained flood-fill traversal, visited-state tracking, graph reachability, and routing algorithms. Helped establish coherent machine dimensions and coordinates so multiple boards could operate as one routable machine.
- **Machine Bring-Up Design & Validation:** Participated in designing the full chip-to-chip and board-to-board lifecycle, from link establishment and discovery convergence through route initialization and core-to-core message validation. Investigated and benchmarked host-driven versus in-machine discovery approaches with the team.
- **Core Communication Stack & Hardware I/O:** Developed and tested layered C++ host-to-device components for the UDP-based SpiNNaker2 Datagram Protocol (SDP2), covering transport and protocol handling, packet processing, command abstractions, request-response workflows, and configurable multi-severity diagnostic logging.
- **Job Management Service & Client APIs:** Developed Python components for job lifecycle management and resource acquisition; designed and implemented a C++ API for configurable hardware requests, board queries, and resource reservations.
- **Fault Handling, Release & Reconciliation:** Owned placement-engine interactions for transactional allocation and release. Worked with infrastructure colleagues on epilog cleanup, automatic periodic health checks, hardware blacklisting, availability changes, and recovery from inconsistent state.
- **Cross-Team Engineering & Research:** Drove design reviews, technical discussions, specifications, and live demonstrations with runtime and infrastructure leads. Collaborated with researchers at Sandia National Laboratories and the University of Manchester on packet processing, routing, and HPC workload management.

Working Student · Software Engineering 2024/07 - 2025/03 | Dresden, Germany
Rohde & Schwarz

- **Performance Engineering:** Optimized a company-wide signal-transformation library through algorithmic and data-structure improvements, achieving 2.5x faster execution.
- **Linux & QEMU Debugging:** Configured and ran QEMU virtual machines, debugged applications inside guests, and inspected QEMU source, socket calls, and termination/control behavior to emulate app-facing responses. Found application defects during systematic testing.
- **PTP / O-RAN Root-Cause Analysis:** Debugged recurring PTP synchronization failures in an O-RAN system; traced clock drift to MTU-induced transmission timing shifts and mixed-traffic interference and delivered fixes restoring master-slave stability.
- **DPDK Timing Benchmarks:** Measured sender warm-up behavior and packet jitter across configurations to identify an effective pre-execution strategy ahead of O-RAN transmission.
- **Linux Diagnostics & Test Automation:** Used systemd and journalctl to investigate service failures and developed Python/Pytest/Playwright CI workflows spanning unit, integration, and end-to-end testing.

Project

Job Scheduling & Topology-Aware Resource Management - Simulated HPC/GPU Clusters 2026/03 - Present
Personal open-source project | Python · FastAPI · PostgreSQL · SQLAlchemy · React · CI/CD

- **System Architecture & Component Isolation:** Designed a modular workload management platform separating Client, Server, Scheduler, and Placer responsibilities, with configurable scheduling and placement interfaces and explicit contracts between queue management, placement, resource allocation, and persistence.
- **Job Lifecycle & Orchestration:** Implemented persistent submission, queuing, running, completion, cancellation, and release state transitions with event history; integrated capacity-aware multi-cluster job routing.
- **Scheduling Algorithms & Policies:** Implemented configurable FIFO and priority scheduling through priority queues, with independent cluster schedulers, deterministic ordering, availability checks, and queue processing after resource release.
- **Topology-Aware Placement:** Modeled Ring, Mesh, and 2D/3D Torus topologies with neighbor discovery and wrap-around distance calculations; implemented deterministic Pack and Spread placement choices for spatially compact or distributed resource allocations.
- **GPU Network Topology Modeling:** Extended the domain with a two-tier leaf-spine fabric representing switches, ports, network interfaces, and physical links; developed graph-based pathfinding, bandwidth/latency constraints, connectivity checks, and read-only GPU candidate discovery.
- **Infrastructure & Resource Modeling:** Designed relational cluster, node, resource-unit, job, and multi-node allocation models; implemented institute-level resource quotas, time-bound reservations, and node decommissioning semantics while preserving running allocations.
- **Backend & Concurrency Control:** Used PostgreSQL row-level locking and transactions to protect hardware availability and coordinate resource assignment, release, and job-state updates consistently.
- **Testing & CI/CD:** Built unit and integration tests for scheduling, placement, and lifecycle operations with GitHub Actions validation of backend, frontend, migrations, linting, and automated tests.
- **Cluster Monitoring & Visualization:** Created an interactive React interface for cluster topologies, resource availability, job histories, utilization, queues, and recent activity with periodically refreshed operational metrics.
- **Human-Directed Agentic Development:** Used reusable Claude/Codex workflows for code exploration, research, architecture assessment, and implementation support. Retained design decisions and acceptance criteria through two-human review, repeated iteration, and automated test suites.

Education

Exchange Student in Computer Science 2024 - 2025 | Germany
Ilmenau University of Technology

MSc in Software Engineering 2021 - 2024 | Tunisia
Tek-Up University

Bachelor in Physics 2017 - 2020 | Tunisia
Faculty of Sciences of Tunis

Additional Experience & Achievements

Freelance C++/JavaScript Developer · Upwork · 2022/03 - 2022/06
Designed and implemented more than 50 levels of an educational game teaching C++, translating programming concepts into interactive challenges.

Competitive Programming | C++ & Algorithms & Data Structures
IEEEXtreme 17.0 (2023): ranked 21st of 1,048 teams in Tunisia and 849th of 8,263 worldwide. Three years of competitive programming and more than 600 algorithmic problems solved.