Abhirag Awasthi

web abhirag.com
email hey@abhirag.com
mobile +91 8619188322

Systems engineer building high-performance distributed systems at massive scale with extreme reliability. Rust has been my primary medium since 2020.

LANGUAGES

Expert	Rust
Intermediate	Python, Go, C, SQL, Zig, C#, Lisp (*various dialects*)
Basic	Lua, Elixir, JavaScript, HTML, CSS, Typst

TOOLS

Backend	Tokio, Hyper, Axum, Actix, Pingora, Go stdlib, Flask, Django
Database	RocksDB, LMDB, DuckDB, PostgreSQL, Redis, SQLite, Bolt
Message Queue	NATS, Kafka
Orchestration	Kubernetes, Keda, Docker, Nomad
Infrastructure	AWS, GCP, Consul, Vault, Packer, Terraform, OpenResty
Observability	Tracing, OpenTelemetry, Prometheus, Grafana, Tracy, Parca, Vector
Data Engineering	Polars, DBSP, Spark, NumPy, Matplotlib, Parquet, Protobuf
Build System	Cargo, Make, Meson
Other	PyO3, jemalloc

EXPERIENCE

2022–2025 **Principal Engineer (Rust Platform Team)** — VerSe Innovation

① Led the design and implementation of a distributed notification engine:
- Delivering 3.3 billion push notifications and 70 million pull notifications daily
- Across android, iOS and web channels
- Throughput of 300k notifications per second while ensuring 99.9999% dispatch reliability

② Built a feature store enabling real-time ML inference for a short videos platform:
- Zero downtime over 18 months while serving 60 million daily requests at p99 < 5ms
- 400k+ hot database handle reloads (once every 20 mins) with zero faults for real-time feature updates

③ Reduced infrastructure costs by $1 million annually through performance engineering:
- Replaced managed load balancers with a reverse proxy built in Rust
- Built KEDA scalers to autoscale downstream services based on notification load
- Utilized continuous profiling to optimize hot paths and inefficient memory management patterns

④ Achieved sub-200ms ML inference for video recommendations by porting hot path Python components to native Rust extensions

2019–2021 Site Reliability Engineer — Shuttl

① Developed and maintained core service infrastructure supporting ≈ 100 production services
② Built tooling to improve the developer experience of interacting with the service infrastructure
③ Built batch pipelines that derived vehicle velocity and acceleration from GPS data to detect deviations from designated stops
④ Wrote the core service infrastructure for running cloud-native Rust services with Consul and Vault integration
⑤ Reduced data serialization/deserialization overhead in Python services by a factor of 3.8:
- Rewrote the layer in Rust
- Preserved binary compatibility using property based testing

2016-2018 Founder — Mockint

Designed and developed a platform for conducting technical mock interviews

2012-2016 Systems Software Engineer — Hewlett-Packard

① Maintained core firmware services: Power Management, Firmware Installation Management, Reset, Backup/Restore, Time
② Improved firmware upgrade fault tolerance by implementing fine-grained locking, minimizing the time window where power failures could corrupt the boot partition
③ Exposed system calls through safe APIs and SNMP OIDs
④ Owned primary triage for crash dumps
⑤ Resolved critical defects for customers with multi-million dollar contracts

PROJECTS

Notification Engine — Rust, Axum, RocksDB, NATS JetStream

- User segmentation via nested boolean queries with comparison operators on typed fields
- Roaring bitmap based inverted indexing for efficient query evaluation
- RocksDB storage layer for protobuf-encoded user profiles and bitmap indexes
- Primary-Standby replication through NATS streaming of RocksDB WAL for fast failover
- Decoupled platform-specific dispatchers processing notifications over NATS stream

Real-time Stream Analytics — Rust, DBSP, Kafka

- Incremental view maintenance over Kafka notification event streams using DBSP circuits
- Stream processing throughput of 3.3 million events/sec, ingesting and analyzing 1 billion notification events in 5 minutes on single 4-core/16GB node
- Computing real-time unique user counts, success metrics, and error type distributions through partitioned rolling aggregates

Smol Recommendation Engine — Rust, LMDB, ONNX Runtime

- Exact vector similarity search with p99 ≈ 6.3ms for generating 20 candidates from a corpus of 6.4 million text chunks on single 10-core/32GB node
- 64x embedding compression (4096 bytes→64 bytes) while retaining 90.76% retrieval accuracy via Matryoshka Representation Learning and binary quantization
- LMDB-persisted binary embeddings with in-memory SIMD-optimized hamming distance computation, eliminating vector database dependency
- Optional re-ranking recovering retrieval accuracy lost to quantization

Converge Rust, Pingora

- Layer 7 reverse proxy with path-based routing and active health checks
- Weighted backend selection algorithms: round-robin, random, and consistent hashing (Ketama/FNV) with header-based session affinity
- Graceful reload support enabling zero-downtime binary upgrades without dropping active connections

Feature Store Rust, RocksDB, Hyper

- Feature retrieval API (hyper/protobuf) with per-request entity and timeline selection
- RocksDB SST files as data interchange format between batch pipelines and serving layer
- Lock-free atomic DB handle swapping: periodic bulk SST ingestion from object storage into fresh RocksDB instances, hot-swapped under concurrent reads

EDUCATION

2008-2012 Dhirubhai Ambani Institute of Information and Communication Technology

- Bachelor of Technology in Information and Communication Technology
- GPA - 7.5/10