High-Scale URL Engineering: Beyond Redirection

A deep dive into the architecture of a high-performance URL shortener.

ÁLVARO OLIVEIRA
SOFTWARE ENGINEER

THE PROBLEM: SCALING THE UNSEEN

THE ANALYTICS BOTTLENECK

- **Redirection is easy**
  resolving a slug takes < 10ms.

- **Analytics is hard**
  collecting IP, Geo, Device, and Browser data for millions of clicks can kill your database.

- **The Goal**
  track everything without adding a single millisecond of latency to the user.

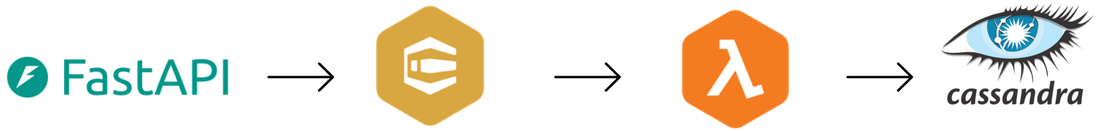
ARCHITECTURE: DECOUPLED & RESILIENT

EVENT-DRIVEN MICROSERVICES FLOW

The "Core" handles the user; the "Worker" handles the data. Isolation is the key to availability.

CORE PATH

ASYNC PATH



ENGINEERING UNIQUE IDS

DISTRIBUTED ID GENERATION (SNOWFLAKE-LIKE)

How to generate short, unique slugs without a central bottleneck?

- **Strategy:** Custom Epoch + Atomic Redis Counter + Bitwise Operations.
- **Result:** Short, Base62-encoded slugs that are mathematically guaranteed to be unique.

THE BITWISE LOGIC

ANATOMY OF A 32-BIT+ UNIQUE ID

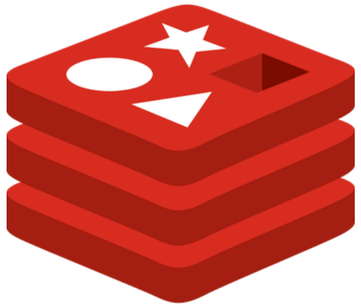
ID = (timestamp_seconds << 10) | redis_sequence

- **Custom Epoch (Jan 2025):** Keeps the timestamp bits small.
- **Redis INCR (10 bits):** Supports 1,024 unique URLs per second.
- **Bitwise Shift:** Separates time and sequence into a single unique integer.

POLYGLOT PERSISTENCE

CHOOSING THE RIGHT TOOL FOR THE JOB

Relational integrity for User/URL mapping.



Ultra-fast caching and atomic sequence counting.

High-velocity write-heavy storage for clickstream logs.

WHY APACHE CASSANDRA?

SCALING WRITE-HEAVY WORKLOADS

Relational DBs struggle with index updates at billions of rows.

Cassandra uses LSM-trees for near-instant writes.

Perfect for Time-Series Analytics like click tracking.

INGESTION VIA SQS & LAMBDA

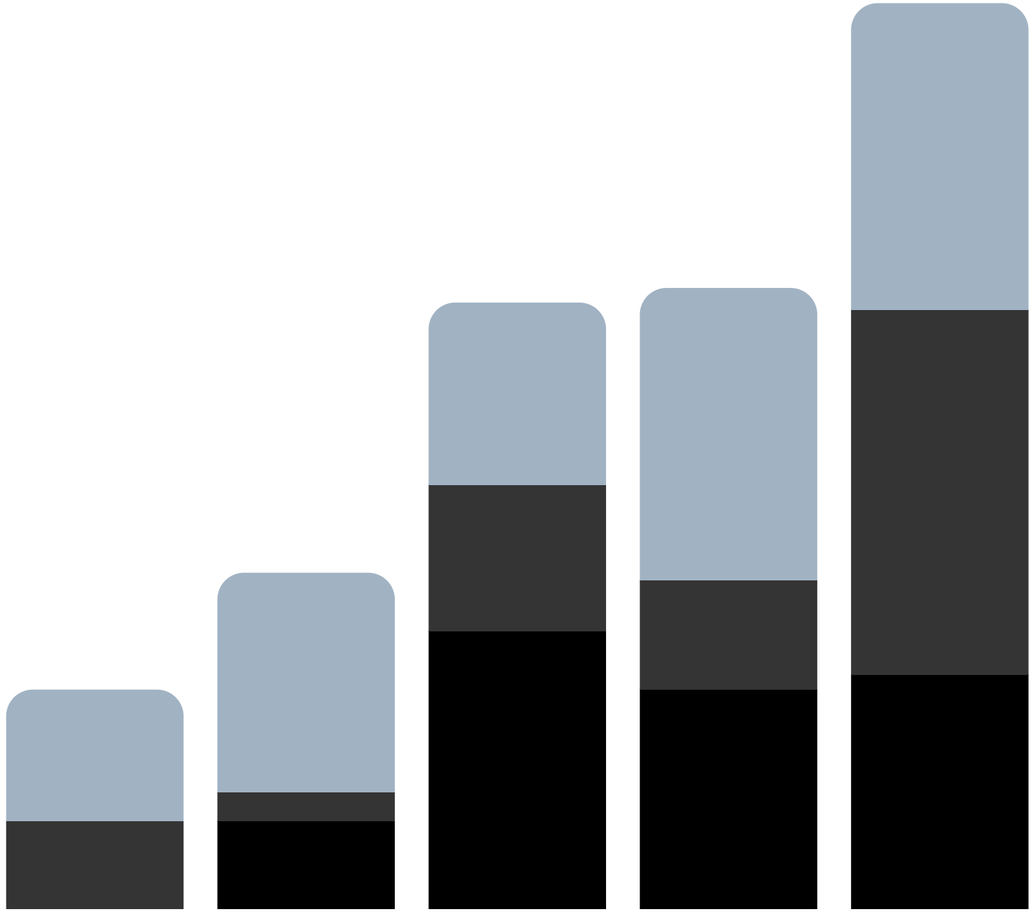
HANDLING BACKPRESSURE

- **SQS:** Acts as a buffer during traffic spikes.
- **Lambda:** Scales horizontally to process the queue.
- **Resilience:** If the database lags, messages stay safe in the queue. No data is lost.

TURNING DATA INTO INSIGHTS

THE ANALYTICS DASHBOARD

Features: Real-time growth rates, and Device/Browser distribution powered by the Cassandra engine.



PERFORMANCE AUDIT: THE "REAL TALK"

BENCHMARKING & BOTTLENECKS

- **Specs:** 4 vCPUs / 16GB RAM.
- **Status:** ~50 requests/second.
- **The Findings:** Hardware is 90% idle. The bottleneck isn't the CPU—it's the Synchronous I/O blocking the Python Event Loop.

THE 10X SCALING ROADMAP

PATH TO 2,000+ REQUESTS/SECOND

STRATEGY

- **Async Everything:** Migrate to SQLAlchemy-async and aioboto3.
- **SQS Batching:** Sending messages in batches of 10 to reduce network overhead.
- **Worker Tuning:** Optimizing Uvicorn to leverage all 4 CPU cores.

CONCLUSION

LESSONS FROM THE BUILD

System design is about trade-offs. Knowing your bottlenecks is as important as knowing your stack.

QUESTIONS?

Message Me!

alvarohmoliveira@gmail.com