

Benjamin Tobler Resumé

ben@tobler.nz | +1 206 370-2990 | tobler.nz

ABOUT

Software engineer working on distributed database & storage systems, networking, and virtualization.

AWS [EC2](#) team member from founding, through beta, GA and to \$1Bn ARR service. Owned the hypervisor side of the EC2 control plane through GA. Led design & implementation of [AWS's VPC](#) control plane. Led design & implementation of [OCI's Virtual Machine Service](#) offering.

Designed & implemented zero copy RPC & network messaging for [Amazon Aurora](#), with thread per core architecture (M:N task scheduling) and replica partitioning. Designed and [patented novel optimistic concurrency primitives](#) on specific value types to eliminate transaction aborts caused by hot key contention. Implemented [10k connection problem](#) solution for Java pre non-blocking IO support, and [EMV \(Chip & PIN\)](#) verification and HSM vendor abstraction APIs for financial transaction processing.

EDUCATION

M.Sc. Computer Science (Distinction), [Network Security](#), University of Cape Town, 2005

Awarded [NRF](#) Prestige Scholarship for Masters study

[Published](#) in proceedings [IFIP SEC](#) 2004 Toulouse & [Infosec SA](#) 2004 Johannesburg

PATENTS

Selected database & virtualization management patents

[US9223843](#), [US9552242](#), [US20170083565](#), [US20210342311](#), [US9055117](#), [US8504691](#), [US8190682](#)

EMPLOYMENT

Oracle Cloud Infrastructure 2016 - 2021, Consulting MTS

Virtual Machine Service

Led the OCI VM offering definition, design & implementation for [OCI](#). Delivered with block storage and virtual network integration (via both VIF and virtio) in ~4 months.

Defined product requirements and implemented proof of concept for Bring Your Own Image ([BYOI](#)) support for VMware, VirtualBox, QEMU and other virtualization products.

Strictly serializable database for control planes

Designed novel ([patent 20210342311](#)) optimistic concurrency primitives (e.g. opaque test-and-increment counters) to prevent hot key contention from aborting transaction commits. Proof of concept implementation & evaluation of new backend storage for improved scaling and availability.

Collaborated on design, implementation and deployment of transition to cell architecture to mitigate noisy neighbors and limit operational blast radius. Advised and collaborated with client teams on effective adoption and schema for varied use cases from standard micro-service backend to source of truth for metadata distributed data planes at scale.

Amazon Web Services 2005 - 2014, Senior SDE

Amazon Aurora distributed storage

Designed and implemented Aurora's zero-copy, vectored IO custom RPC and

network messaging that achieved the ambitious latency & throughput goals for fanning out log writes from the database engine to distributed storage.

Mentored remote Bay Area team through integration of database engine with distributed storage via custom RPC.

Identified numerous performance bottlenecks in the networking and storage I/O paths via profiling, and implemented solutions: e.g. task scheduler per core to minimize task queue lock contention, eliminating heap allocations on copy of C++ function object callbacks for IO/event completions, and message buffer vectors to avoid copying user data payloads.

Virtual Private Cloud

Led design and implementation the VPC virtual network management control plane. The system manages VPC customers' network metadata, including network topology maps to physical resources, metadata ACLs, and provides a stream source for distribution to hypervisors.

Collaborated with AWS Linux kernel team on successful effort to improve packet throughput on hypervisors. Achieved by moving source identification logic from control plane agent in user space, to a kernel module.

Extended and maintained the distributed firewall system that implements EC2 instance security groups across all the hypervisors in a data center.

Optimized in-memory representation of VPC network topology and security policy on hypervisors to support massive VPC & EC2 scaling while making efficient use of limited dom0 hypervisor memory.

EC2 from founding, through beta and GA

Designed and implemented the EC2 control plane hypervisor agent. Ran on > 100K physical hosts (and now > 1M hypervisors) to manage the customer virtual machine (EC2 instance) lifecycle. Extended the agent to collaborate with neighboring hypervisors to download and cache boot images from object store service (S3). The improvement allowed fast EC2 instance launches despite limited edge bandwidth (circa private beta launch).

Implemented many internal and public utilities, micro-services and tools for EC2: e.g. command-line AMI packager and up-loader, internal dynamic DNS vendor for EC2 instances and hypervisors, and network rate limiting at hypervisor to prevent noisy neighbor problems.

Helped grow the EC2 team in Seattle by ramping up new team members tasked with developing new EC2 networking and security functionality.