VOLT ACTIVE DATA

Volt on ARM

Same Platform. Same Guarantees. Lower Cost Per Decision.

Cloud bills have made total cost of ownership impossible to ignore. ARM-based instances such as AWS Graviton, Azure Cobalt and Google Axion are priced below comparable x86 instances. But a mission-critical data platform is only as portable as its vendor makes it, and many are x86-only, or run on ARM as an afterthought.

When the platform dictates the hardware, the consequences are immediate:

- x86-Only Lock-In
- Cost Per Transaction Climbs
- Latency Spikes Under Load
- Energy Budgets Overrun
- Vendor Picks Your Instances

Volt's ARM support eliminates that trade-off. Native 64-bit ARM builds of the Volt server and Kubernetes operator deliver the full enterprise feature set, and applications run unmodified. In Volt's own benchmark, AWS Graviton delivered roughly 40% more work per core than a comparable Intel instance at the same 10 ms 99th-percentile latency target.

How it works

Volt is built on AArch64 as a first-class target, not translated or emulated. The performance-sensitive parts of the engine that were originally written in x86 assembly were rewritten for ARM, and every release ships native builds for both architectures: server, operator and container images, from the same release train, with the same features.

Volt's architecture suits ARM unusually well. The engine pins each partition to a single core and runs it to completion, so it gains nothing from hardware multithreading. On hyperthreaded x86 instances, two hardware threads share one physical core, and once utilization passes roughly 50% they contend for it; in Volt's benchmark, 99th-percentile latency deteriorated sharply at exactly that point. On Graviton, every vCPU is a physical core, and no such deterioration was observed as load rose.

Nothing changes above the platform. The same SQL, stored procedures, client libraries, Helm charts and operational procedures apply. Choosing ARM is an instance-family decision at deployment time, not a migration.

What this means in practice

Lower cost per decision. More work per core means fewer or smaller instances for the same workload. In the benchmark, the cost of supporting a 400,000-transactions-per-second charging workload was around a third lower on Graviton than on the equivalent Intel instances at the AWS pricing in force.

Predictable latency at high utilization. Without a multithreading cliff, clusters can run hotter before tail latency moves. For charging, policy and fraud systems with a hard 99th-percentile budget, that is usable capacity you would otherwise leave idle as headroom.

No application change. Applications, schemas and procedures are architecture-neutral. Teams move existing workloads by changing the instance type in the deployment, and can mix architectures across environments while they do.

The same workload on x86 and on ARM

	Hyperthreaded x86	Volt on ARM (Graviton)
Work per core at p99 ≤ 10 ms	Baseline	Roughly 40% more[1]
Tail latency as CPU passes 50%	Deteriorates as hardware threads contend	Stable
Cost of 400K TPS charging workload	Baseline	Around a third lower[1]
Energy per instance	Baseline	Lower (AWS: up to 60% less)
Feature set	Full	Full
Application changes	None	None

1 Source: Volt benchmark, July 2024, based on AWS spot pricing in the Dublin region. Full results: voltactivedata.com/blog/2024/07/volt-significantly-faster-and-cheaper-than-intel-on-aws-arm

Architectural properties

- Native AArch64 builds of the Volt server and Kubernetes operator; multi-architecture container images; generally available since V13.1 (January 2024), following a beta in V13.0, and carried in every release and LTS line since
- Full enterprise feature set on ARM: K-safety, command logging, snapshots, cross-data-center replication (Active(N)), in-service software upgrade (ISSU), export and import
- Benchmarked on AWS Graviton (c7g) against Intel (c7i) using Volt's charging workload: 3- to 9-node clusters, K=1, 99th-percentile latency cutoff of 10 ms, linear scaling to the limit of the instance networking
- Runs on any AArch64 Linux platform, including AWS Graviton, Azure Cobalt, Google Axion and Ampere Altra-based servers
- Operator and Helm charts are architecture-aware; deployment procedures are unchanged

Key outcomes

- Cut infrastructure cost for the same workload, with roughly 40% more work per core
- Keep 99th-percentile latency stable at high utilization, and use the headroom you used to hold back
- Choose hardware on price and availability, not on what the data platform supports
- Move with no application changes: an instance-type decision, not a migration

AVAILABILITY

ARM support is included with the Volt Enterprise license. The AArch64 build is a separate download and container image, available for every current release.

Book a technical call voltactivedata.com/contact

Volt Active Data is the real-time decisioning layer for mission-critical systems.