Testing Methodology and Configurations:

How Arm Neoverse CSS-based Microsoft Cobalt 200 virtual machines (VMs) accelerate cloud and agentic AI performance

Results based on Arm internal testing. All benchmark results were obtained under controlled and repeatable test conditions using the latest available software and platform configurations. Tests were conducted on Ubuntu 26.04 with consistent kernel, compiler, and runtime environments. Results may vary.

Platforms and Azure VM instances

The benchmarks compare Azure virtual machines powered by Microsoft Cobalt 200, Microsoft Cobalt 100, and AMD EPYC Turin processors.

Benchmark	Category	Cobalt 200	Cobalt 100	AMD EPYC Turin
NGINX	Cloud Workloads	Standard_D4ps_v7	Standard_D4ps_v6	Standard_D4as_v7
MySQL	Cloud Workloads	Standard_E16ps_v7	Standard_E16ps_v6	Standard_E16as_v7
x265	Cloud Workloads	Standard_D64ps_v7	Standard_D64ps_v6	Standard_D128as_v7 (64 physical cores)
Redis	Cloud Workloads	Standard_D4ps_v7	Standard_D4ps_v6	Standard_D4as_v7
fastText: Skip-gram	Agentic Workloads	Standard_D8ps_v7	Standard_8Dps_v6	Standard_D8ads_v7
fastText: CBOW	Agentic Workloads	Standard_D8ps_v7	Standard_8Dps_v6	Standard_D8ads_v7
Milvus	Agentic Workloads	Standard_D8ps_v7	Standard_8Dps_v6	Standard_D8ads_v7
SWE-bench	Agentic Workloads	Standard_D16ps_v7	Standard_D16ps_v6	Standard_D16ads_v7

Cloud Workloads

General testing principles

All systems used the default OS image, compiler version and library versions. NGINX HTTPS load-balancer throughput was measured using wrk with 300 connections and 64 threads. Each test ran for 60

seconds, with three tests conducted. Tests were measured on AGI Gateway, load balancing across six upstream servers. Testing was conducted on Ubuntu 26.04; the Cobalt 200 instance was Standard_D4ps_v7.

MySQL benchmarks

MySQL throughput and throughput-per-dollar were measured using HammerDB. Testing used MySQL 9.7.2 on Ubuntu 26.04 with 256 warehouses (16 times the number of vCPUs). Each virtual-user test included a 10-minute ramp-up followed by a 20-minute test run. The environment used GCC 15.2.0, OpenSSL 3.5.5, glibc 2.43, and Azure kernel 7.0.0-1017-azure.

x265 benchmarks

The x265 benchmarks measured 4K video encoding performance using the Bosphorus Y4M video, consisting of 600 uncompressed frames. The x265 version 4.3 (commit 0900b28e7057ec0f4492237963aa92b0f6469a76) was compiled with clang-22 as release mode. The reported results are the mean runtime of 10 runs on Ubuntu 26.04. Instance sizes were selected to have at least 64 cores; the AMD EPYC Turin instance was Standard_D128as_v7 (64 physical cores). The Intel instance d128sv7 is dual socket, so only one of the sockets was use for the evaluation. For the other instances no particular cores were selected or pinned. The benchmarks report the compression using 8, 10 and 12 bit encoding. The results report from 1 to 64 threads performance.

Redis benchmarks

Redis throughput and throughput-per-dollar were measured using memtier_benchmark 2.4.1 with Redis 8.10.2. Testing used a single Redis server instance configured with io-threads=1, no configured memory limit (maxmemory=0), and the noeviction policy. The client used four threads with 50 connections per thread, a sweep of pipeline depths, 128-byte values, a 1:10 SET-to-GET ratio, a key range maximum of 10 million, and 600-second runs . The test environment used Ubuntu 26.04, kernel 7.0.0-1017-azure, and GCC 15.2.0.

Agentic Workloads

fastText benchmarks

The fastText benchmarks cover two training models: Skip-gram and Continuous Bag-of-Words (CBOW). The number of threads is configured to match the number of vCPUs available on the machine. The software is built from the latest repository commit available at the time of benchmarking (1142dc4).

Milvus benchmarks

The Milvus benchmarks are conducted using VectorDBBench, a benchmarking tool for vector databases, with the HNSW index type. All benchmark runs use Milvus v2.6.15, and performance is measured in queries per second (QPS), as reported by VectorDBBench.

SWE Bench benchmarks

SWE Bench benchmarks used SWEBench Verified tasks in Docker environments with mini-swe-agent. A separate replay server supplied prerecorded LLM responses with recorded tool calls, allowing comparison of CPU-side agent orchestration, tool execution, and container overhead. Throughput is reported as completed tasks per minute during the steady-state interval.