基准测试 · ARM64 对比 X86_64

Imaxe Monitor Solution - Zabbix 7.0 1.2

我们发布的 Graviton 镜像与同源 x86_64 构建在同级实例类型上的实测对比。

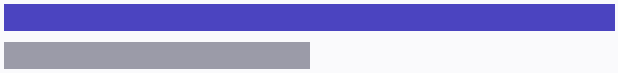
编号 PERF-zabbix70-1.2-20261107　　生成于 2026-11-07　　us-east-1

结论摘要

在已测试项中，Graviton 的中位性能为 +14,1 %。

全文颜色约定：**ARM64 (Graviton)** · **x86_64**. 差值另附符号（▲ / ▼），便于黑白阅读。

1 测试结果

测试项	ARM64 · C7G.LARGE	X86_64 · C6I.LARGE	Δ ARM64	对比
sysbench — CPU (1 hilo)	**1113 events/s**	1074 events/s	▲ +3,6 %	
sysbench — CPU (todos los hilos)	**2221 events/s**	1113 events/s	▲ +99,5 %	
sysbench — memoria	**40914 MiB/s**	19077 MiB/s	▲ +114,5 %	
openssl — AES-256-GCM	**4147 MB/s**	8772 MB/s	▼ -52,7 %	
openssl — SHA-256	**1600 MB/s**	1284 MB/s	▲ +24,6 %	
gzip — compresión de un corpus fijo	**129,6 MB/s**	155,6 MB/s	▼ -16,7 %	
fio — lectura aleatoria 4k	**3043 IOPS**	3020 IOPS	▲ +0,8 %	
fio — escritura aleatoria 4k	**3047 IOPS**	3047 IOPS	▼ 0,0 %	
sysbench — CPU, latencia p95	**0,90 ms**	1,82 ms	▲ +102,2 %	
fio — lectura 4k, latencia p99	**5800 µs**	29230 µs	▲ +404,0 %	

在可比测试项中，Graviton 的中位优势为 +14,1 %。每组柱状图按其自身一对中的较大值缩放，因为各测试项单位不同。

2 多线程扩展性

参数	ARM64	X86_64
CPU（1 线程）	**1113 events/s**	1074 events/s
CPU（全部线程）	**2221 events/s**	1113 events/s
扩展倍数	**×2,00 (2 vCPU)**	×1,04 (2 vCPU)
相对理想值的效率	**99,8 %**	51,8 %



相对理想值的效率

从一个线程扩展到全部 vCPU，Graviton 的处理量提升 2,00 倍，x86_64 实例提升 1,04 倍。

> vCPU 在两个系列中并不是同一回事：在 Graviton 上它是一个物理核心，在 Intel 上它是与孪生线程共享执行单元的 SMT 线程。因此单线程的成绩可能相近，差距只有在全部线程都忙起来时才显现——而这正是产品的工作方式：nginx 的 worker、PHP-FPM 的进程池和数据库的连接会同时用满所有 vCPU，而不是一个。

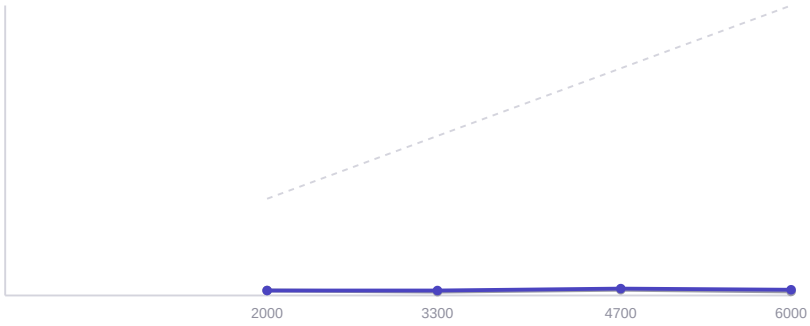
3 方法

▸ 测试环境

参数	ARM64	X86_64
区域	us-east-1	us-east-1
实例类型	c7g.large	c6i.large
CPU 与内存	2 vCPU · 3799 MiB	2 vCPU · 3811 MiB
处理器	Neoverse-V1	Intel(R) Xeon(R) Platinum 8375C CPU @ 2.90GHz
内核	7.0.0-1017-aws	7.0.0-1017-aws

> 每种架构都在各自的测试实例上测量，且仅在该实例通过产品测试之后进行。测量期间产品处于运行状态，这与买家机器的真实状态一致。

4 负载曲线



实际交付请求数与请求速率的对比。虚线对角线为请求速率：曲线偏离之处即为该机器饱和点。

请求速率	ARM64	X86_64	ARM64 P99	X86_64 P99
2000	**103,9** ▲	104,2 ▲	**67057 ms**	56975 ms
3300	**101,7** ▲	78,8 ▲	**49891 ms**	41113 ms
4700	**142,4** ▲	119,5 ▲	**43587 ms**	53015 ms
6000	**119,7** ▲	84,0 ▲	**44825 ms**	62663 ms

> ▲ 表示该档位机器已无法维持请求速率。超过该点后，p99 衡量的是排队而非延迟。

> 负载由一台独立的机器产生：两种架构使用相同的实例类型、位于同一可用区，以恒定到达速率通过私有地址访问被测机器。两次测量的客户端完全相同，唯一变化的是负责响应的机器。

▸ 承载能力与工作成本

项目	ARM64	X86_64	Δ ARM64
可持续的最大速率	**142,4 req/s**	119,5 req/s	**▲ +19,1 %**

> 压力爬坡的起点高于其中一台机器的承载上限，因此这台没有任何干净的台阶，展示的是观测到的最大吞吐——那是一个上限，而不是可持续速率。

> 可持续的最大速率是机器完整服务下来的最后一个台阶，而不是图上的峰值：饱和的台阶吞吐看着高但响应很差，把它算作承载能力等于承诺一个产品已经吃力的负载。每百万次请求的成本按该速率计算，当两台机器承载能力不同时，这才是站得住的比较：每小时价格说的是开着一台机器的成本，不是它所完成工作的成本。

5 结论

Imaxe Monitor Solution - Zabbix 7.0 发布于 Graviton（arm64），因为它实测更好且成本更低：相对同级 x86_64 实例有 +14,1 % 的中位优势。

> 本报告的全部数据均来自对所发布镜像的实际运行。未能在两种架构上测量的测试项均已标注，并排除在中位数之外。

A 附录 — 执行的命令

> 所执行命令的完整记录，以便复现或质疑这些数据。

测试项	命令
sysbench — CPU (1 hilo)	`sysbench cpu --cpu-max-prime=20000 --threads=1 --time=60 run`
sysbench — CPU (todos los hilos)	`sysbench cpu --cpu-max-prime=20000 --threads=$(nproc) --time=60 run`
sysbench — memoria	`sysbench memory --memory-block-size=1M --memory-total-size=100G --threads=2 run`
openssl — AES-256-GCM	`openssl speed -elapsed -evp aes-256-gcm -seconds 3 2>&1`
openssl — SHA-256	`openssl speed -elapsed -evp sha256 -seconds 3 2>&1`
gzip — compresión de un corpus fijo	`f=/mnt/bench/corpus.txt; s=$(stat -c%s $f); t0=$(date +%s%N); i=0; while [ $i -lt 3 ]; do gzip -6 -c $f >/dev/null; i=$((i+1)); done; t1=$(date +%s%N); awk -v s=$s -v a=$t0 -v b=$t1 -v n=3 'BEGIN{printf "throughput: %.2f MB/s\n", n*s/1000000/((b-a)/1000000000)}'`
fio — lectura aleatoria 4k	`fio --name=randread --ioengine=libaio --direct=1 --bs=4k --iodepth=16 --rw=randread --size=4G --runtime=60 --time_based --filename=/mnt/bench/testfile`
fio — escritura aleatoria 4k	`fio --name=randwrite --ioengine=libaio --direct=1 --bs=4k --iodepth=16 --rw=randwrite --size=4G --runtime=60 --time_based --filename=/mnt/bench/testfile`
sysbench — CPU, latencia p95	`sysbench cpu --cpu-max-prime=20000 --threads=$(nproc) --time=60 run`
fio — lectura 4k, latencia p99	`fio --name=randread --ioengine=libaio --direct=1 --bs=4k --iodepth=16 --rw=randread --size=4G --runtime=60 --time_based --filename=/mnt/bench/testfile`