БЕНЧМАРК · ARM64 ПРОТИВ X86_64

Imaxe Monitor Solution - Zabbix 7.0 1.2

Измеренное сравнение публикуемого образа Graviton и идентичной сборки x86_64 на эквивалентных типах инстансов.

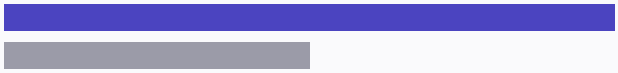
Рег. № PERF-zabbix70-1.2-20261107 Создан 2026-11-07 us-east-1

ГЛАВНОЕ

Graviton даёт +14,1 % производительности по медиане в измеренных тестах.

Цветовая схема по всему документу: **ARM64 (Graviton)** · **x86_64**. Дельты дополнительно помечены символом (▲ / ▼) для чтения в чёрно-белом виде.

1 Результаты

ТЕСТ	ARM64 · C7G.LARGE	X86_64 · C6I.LARGE	Δ ARM64	СРАВНЕНИЕ
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 %	

Медианное преимущество Graviton по сопоставимым тестам: +14,1 %. Каждая полоса масштабируется к большему в своей паре, поскольку у каждого теста своя единица измерения.

2 Масштабирование по потокам

ПАРАМЕТР	ARM64	X86_64
CPU в 1 поток	**1113 events/s**	1074 events/s
CPU во все потоки	**2221 events/s**	1113 events/s
Коэффициент масштабирования	**×2,00 (2 vCPU)**	×1,04 (2 vCPU)
Эффективность относительно идеала	**99,8 %**	51,8 %



Эффективность относительно идеала

При переходе от одного потока ко всем vCPU Graviton увеличивает объём работы в 2,00 раза, а инстанс x86_64 — в 1,04.

> vCPU в этих двух семействах — не одно и то же: у Graviton это физическое ядро, у Intel — поток SMT, делящий исполнительные блоки со своим близнецом. Поэтому в один поток цифры могут быть похожими, а разница проявляется, когда заняты все, — именно так и работает продукт: nginx со своими воркерами, PHP-FPM со своим пулом и база данных со своими соединениями используют все vCPU сразу, а не одну.

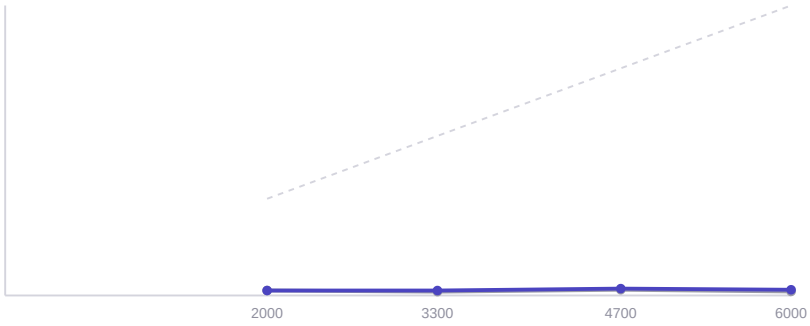
3 Методика

▸ Испытательная среда

ПАРАМЕТР	ARM64	X86_64
Регион	us-east-1	us-east-1
Тип инстанса	c7g.large	c6i.large
CPU и память	2 vCPU · 3799 MiB	2 vCPU · 3811 MiB
Процессор	Neoverse-V1	Intel(R) Xeon(R) Platinum 8375C CPU @ 2.90GHz
Ядро	7.0.0-1017-aws	7.0.0-1017-aws

> Каждая архитектура измеряется на своём тестовом инстансе и только после того, как этот инстанс прошёл тесты продукта. Продукт работает во время измерения — именно так выглядит машина покупателя.

4 Кривая нагрузки



Обслуженные запросы против запрошенных. Пунктирная диагональ — запрошенная частота: там, где линия от неё отходит, машина насыщается.

ЗАПРОШЕННАЯ ЧАСТОТА	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

▲ отмечает ступень, на которой машина уже не держит запрошенную частоту. Дальше p99 измеряет очередь, а не задержку.

Нагрузку создаёт отдельная машина — того же типа для обеих архитектур и в той же зоне доступности, — обращаясь к измеряемой машине по приватному адресу с постоянной частотой поступления запросов. Клиент в обоих прогонах одинаков: меняется только машина, которая отвечает.

▸ Ёмкость и стоимость работы

СТАТЬЯ	ARM64	X86_64	Δ ARM64
Максимальная устойчивая нагрузка	**142,4 req/s**	119,5 req/s	**▲ +19,1 %**

Рампа началась выше того, что выдерживала одна из машин, поэтому для неё нет ни одной чистой ступени и показан максимальный наблюдавшийся поток — это потолок, а не устойчивая нагрузка.

Максимальная устойчивая нагрузка — это последняя ступень, которую машина отработала полностью, а не пик графика: насыщенная ступень выдаёт много и отвечает плохо, и считать её ёмкостью значило бы обещать нагрузку, на которой продукту уже тяжело. Стоимость миллиона запросов считается при этой нагрузке, и это единственное честное сравнение, когда машины держат разное: цена за час говорит, сколько стоит держать её включённой, а не сколько стоит выполненная работа.

5 Вывод

Imaxe Monitor Solution - Zabbix 7.0 публикуется на Graviton (arm64), потому что показывает лучшие результаты и стоит меньше: +14,1 % медианного преимущества над эквивалентным инстансом x86_64.

> Все цифры получены на публикуемом образе. Тесты, которые не удалось измерить на обеих архитектурах, помечены и исключены из медианы.

A Приложение — выполненные команды

> Ровно то, что было выполнено, чтобы цифры можно было воспроизвести или оспорить.

ТЕСТ	КОМАНДА
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`