BENCHMARK · ARM64 FRENTE A X86_64

Imaxe Monitor Solution - Zabbix 7.0 1.2

Comparación medida entre la imagen Graviton que publicamos y una compilación x86_64 idéntica, sobre tipos de instancia equivalentes.

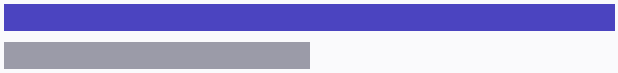
Ref. PERF-zabbix70-1.2-20261107 Generado 2026-11-07 us-east-1

EN UNA LÍNEA

Graviton rinde un +14,1 % de mediana en los tests medidos.

Convención de color en todo el documento: **ARM64 (Graviton)** · **x86_64**. Los deltas llevan además símbolo (▲ / ▼) para lectura en blanco y negro.

1 Resultados

TEST	ARM64 · C7G.LARGE	X86_64 · C6I.LARGE	Δ ARM64	COMPARACIÓN
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 %	

Ventaja mediana de Graviton en los tests comparables: +14,1 %. Cada barra se escala al mayor de su propia pareja, porque cada test tiene su unidad.

2 Escalado con los hilos

PARÁMETRO	ARM64	X86_64
CPU con 1 hilo	**1113 events/s**	1074 events/s
CPU con todos los hilos	**2221 events/s**	1113 events/s
Factor de escalado	**×2,00 (2 vCPU)**	×1,04 (2 vCPU)
Eficiencia sobre el ideal	**99,8 %**	51,8 %



Eficiencia sobre el ideal

Al pasar de un hilo a todos los vCPU, Graviton multiplica su trabajo por 2,00 y la instancia x86_64 por 1,04.

> Un vCPU no es lo mismo en las dos familias: en las Graviton es un núcleo físico y en las Intel un hilo SMT que comparte las unidades de ejecución con su gemelo. Por eso un solo hilo puede ir parecido en las dos y la diferencia solo aparece al usarlos todos, que es como trabaja el producto: nginx con sus workers, PHP-FPM con su pool y la base de datos con sus conexiones usan todos los vCPU a la vez, no uno.

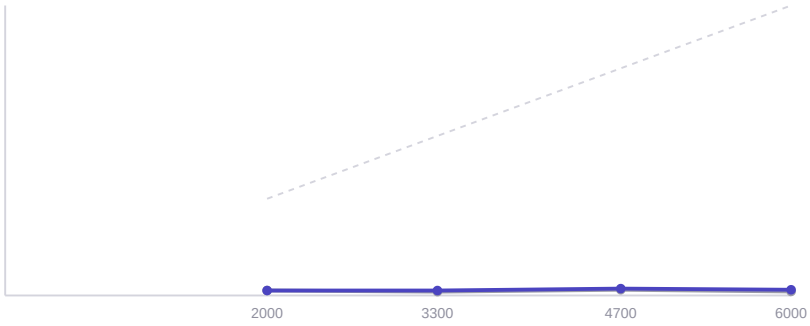
3 Metodología

▸ Entorno de pruebas

PARÁMETRO	ARM64	X86_64
Región	us-east-1	us-east-1
Tipo de instancia	c7g.large	c6i.large
CPU y memoria	2 vCPU · 3799 MiB	2 vCPU · 3811 MiB
Procesador	Neoverse-V1	Intel(R) Xeon(R) Platinum 8375C CPU @ 2.90GHz
Kernel	7.0.0-1017-aws	7.0.0-1017-aws

> Cada arquitectura se mide en su propia instancia de test, y solo después de que esa instancia haya pasado los tests del producto. El producto está encendido mientras se mide, que es como está la máquina de un comprador.

4 Curva de carga



Peticiones entregadas frente a peticiones pedidas. La diagonal discontinua es la tasa pedida: donde una línea se despega, esa máquina satura.

TASA PEDIDA	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

> ▲ marca un escalón donde la máquina ya no sostiene la tasa pedida. A partir de ahí el p99 mide cola de espera, no latencia.

> La carga la genera una máquina aparte, del mismo tipo en las dos arquitecturas y en la misma zona, que golpea a la máquina medida por su dirección privada a tasa de llegada constante. El cliente es idéntico en las dos pasadas: lo único que cambia es la máquina que responde.

▸ Capacidad y coste del trabajo

CONCEPTO	ARM64	X86_64	Δ ARM64
Tasa máxima sostenida	**142,4 req/s**	119,5 req/s	**▲ +19,1 %**

> La rampa empezó por encima de lo que aguantaba alguna de las dos, así que para esa no hay ningún escalón limpio y lo que se enseña es el caudal máximo observado: un techo, no una tasa sostenida.

> La tasa máxima sostenida es el último escalón que la máquina sirvió entero, no el pico de la gráfica: un escalón saturado entrega mucho y responde mal, y contarlo como capacidad sería prometer una carga en la que el producto ya va mal. El coste por millón de peticiones se calcula a esa tasa, y es la comparación que vale cuando las dos máquinas no aguantan lo mismo: el precio por hora dice lo que cuesta tenerla encendida, no lo que cuesta el trabajo que hace.

5 Conclusión

Imaxe Monitor Solution - Zabbix 7.0 se publica en Graviton (arm64) porque mide mejor y cuesta menos: un +14,1 % de ventaja mediana sobre la instancia x86_64 equivalente.

> Todas las cifras de este informe salen de una pasada sobre la imagen que se publica. Los tests que no se pudieron medir en las dos arquitecturas aparecen como tales y quedan fuera de la mediana.

A Apéndice — comandos ejecutados

> Exactamente lo que se ejecutó, para que los números se puedan reproducir o rebatir.

TEST	COMANDO
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`