قياس الأداء · ARM64 مقابل X86_64

Imaxe Monitor Solution - Zabbix 7.0 1.2

مقارنة مقيسة بين صورة Graviton التي ننشرها وبناء x86_64 مطابق، على أنواع مثيلات مكافئة.

مرجع PERF-zabbix70-1.2-20261107 أُنشئ في 07-11-2026 us-east-1

الخلاصة

يقدّم Graviton أداءً وسيطًا بنسبة +14,1 % في الاختبارات المقيسة.

اصطلاح الألوان في المستند كله: **x86_64** · **ARM64 (Graviton)**. تحمل الفروق رمزًا إضافيًا (▲ / ▼) لتُقرأ بالأبيض والأسود.

1 النتائج

الاختبار	ARM64 · C7G.LARGE	X86_64 · C6I.LARGE	Δ ARM64	مقا
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	**ms 0,90**	ms 1,82	% 102,2+ ▲	
fio — lectura 4k, latencia p99	**µs 5800**	µs 29230	% 404,0+ ▲	

الأفضلية الوسيطة لـ Graviton في الاختبارات القابلة للمقارنة: +14,1 %. كل شريط مقيس إلى الأكبر في زوجه، لأن لكل اختبار وحدته.

2 التوسّع عبر الخيوط

المعامل	ARM64	X86_64
المعالج بخيط واحد	**1113 events/s**	1074 events/s
المعالج بكل الخيوط	**2221 events/s**	1113 events/s
معامل التوسّع	**2,00× (vCPU 2)**	1,04× (vCPU 2)
الكفاءة مقارنة بالمثالي	**99,8 %**	51,8 %



الكفاءة مقارنة بالمثالي

عند الانتقال من خيط واحد إلى كل الأنوية الافتراضية، يضاعف Graviton عمله 2,00 مرة والمثيل x86_64 1,04 مرة.

> النواة الافتراضية ليست الشيء نفسه في العائلتين: في Graviton هي نواة فيزيائية، وفي Intel خيط SMT يتقاسم وحدات التنفيذ مع توأمه. لذلك قد يبدو الخيط الواحد متقاربًا في الاثنين، ولا يظهر الفارق إلا عند تشغيلها كلها — وهذه هي طريقة عمل المنتج فعليًا: nginx بعمّاله، وPHP-FPM بمجمّعه، وقاعدة البيانات باتصالاتها، تستخدم كل الأنوية الافتراضية معًا لا واحدة.

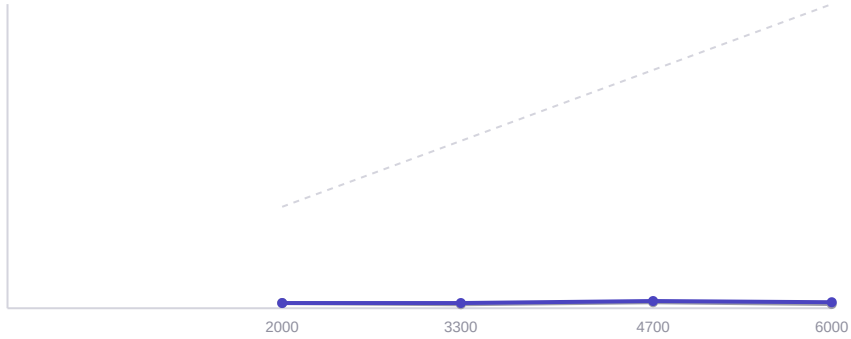
3 المنهجية

▸ بيئة الاختبار

المعامل	ARM64	X86_64
المنطقة	us-east-1	us-east-1
نوع المثيل	c7g.large	c6i.large
المعالج والذاكرة	2 vCPU · 3799 MiB	2 vCPU · 3811 MiB
المعالج	Neoverse-V1	Intel(R) Xeon(R) Platinum 8375C CPU @ 2.90GHz
النواة	aws-7.0.0-1017	aws-7.0.0-1017

> تُقاس كل معمارية على مثيل الاختبار الخاص بها، وبعد أن يجتاز ذلك المثيل اختبارات المنتج فقط. المنتج يعمل أثناء القياس، وهو ما عليه جهاز المشتري فعليًا.

4 منحنى الحِمل



الطلبات المُخدّمة مقابل الطلبات المطلوبة. القطر المتقطع هو المعدل المطلوب: حيث ينفصل الخط عنه، يكون الجهاز مُشبعًا.

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

▲ يشير إلى درجة لم يعد الجهاز يحافظ فيها على المعدل المطلوب. بعدها يقيس p99 الانتظار في الطابور لا زمن الاستجابة.

يُولَّد الحِمل من جهاز منفصل — النوع نفسه من المثيلات لكلتا المعماريتين وفي منطقة التوافر نفسها — يطرق الجهاز المقيس عبر عنوانه الخاص بمعدل وصول ثابت. العميل واحد في المرورين: الشيء الوحيد الذي يتغير هو الجهاز الذي يستجيب.

▸ السعة وتكلفة العمل

البند	ARM64	X86_64	Δ ARM64
أقصى معدّل مستدام	**req/s 142,4**	req/s 119,5	**▲ +19,1 %**

بدأ التدرّج فوق ما تتحمّله إحدى الآلتين، فلا توجد لها درجة نظيفة وما يُعرض هو أقصى تدفّق مرصود: سقف، لا معدّل مستدام.

أقصى معدّل مستدام هو آخر درجة خدمتها الآلة كاملة، وليس ذروة الرسم البياني: الدرجة المشبعة تُسلّم كثيرًا وتستجيب بسوء، واعتبارها سعةً يعني الوعد بحمل يتعثّر عنده المنتج أصلًا. تُحسب التكلفة لكل مليون طلب عند ذلك المعدّل، وهي المقارنة الصامدة حين لا تتحمّل الآلتان الحمل نفسه: السعر بالساعة يقول كم يكلّف إبقاؤها مشتغلة، لا كم يكلّف العمل الذي تؤديه.

5 الخلاصة

يُنشر Imaxe Monitor Solution - Zabbix 7.0 على Graviton (arm64) لأنه يقيس أفضل ويكلّف أقل: أفضلية وسيطة قدرها +14,1 % على مثيل x86_64 المكافئ.

كل الأرقام في هذا التقرير من تشغيل فعلي على الصورة المنشورة. الاختبارات التي تعذّر قياسها على المعماريتين مُبيّنة كذلك ومستبعدة من الوسيط.

A ملحق — الأوامر المنفّذة

ما نُفِّذ بالضبط، حتى يمكن إعادة إنتاج الأرقام أو الاعتراض عليها.

الاختبار	الأمر
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`