بینچ مارک · ARM64 بمقابلہ X86_64

Imaxe Monitor Solution - Zabbix 7.0 1.2

جو Graviton امیج ہم شائع کرتے ہیں اور اسی ماخذ سے بنی x86_64 بلڈ کا، مساوی انسٹینس اقسام پر ماپا گیا موازنہ۔

حوالہ PERF-zabbix70-1.2-20261107 تیار شدہ 07-11-2026 us-east-1

خلاصہ

ماپے گئے ٹیسٹوں میں Graviton +14,1 % وسطی کارکردگی دیتا ہے۔

پورے دستاویز میں رنگ کا اصول: **x86_64** · **ARM64 (Graviton)**. فرق کے ساتھ علامت (▲ / ▼) بھی ہے، تاکہ سیاہ و سفید میں پڑھا جا سکے۔

1 نتائج

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

قابلِ موازنہ ٹیسٹوں میں Graviton کی وسطی برتری: +14,1 %۔ ہر بار اپنے ہی جوڑے کے بڑے کے تناسب سے ہے، کیونکہ ہر ٹیسٹ کی اکائی مختلف ہے۔

2 تھریڈز پر اسکیلنگ

پیرامیٹر	ARM64	X86_64
CPU، ایک تھریڈ	**events/s 1113**	events/s 1074
CPU، تمام تھریڈز	**events/s 2221**	events/s 1113
اسکیلنگ کا عدد	**(vCPU 2) 2,00×**	(vCPU 2) 1,04×
مثالی کے مقابلے میں کارکردگی	**% 99,8**	% 51,8



مثالی کے مقابلے میں کارکردگی

ایک تھریڈ سے تمام vCPU تک جاتے ہوئے Graviton اپنا کام 2,00 گنا بڑھاتا ہے اور x86_64 انسٹینس 1,04 گنا۔

> دونوں خاندانوں میں vCPU ایک ہی چیز نہیں: Graviton میں یہ ایک طبعی کور ہے، Intel میں یہ ایک SMT تھریڈ ہے جو اپنے جڑواں کے ساتھ عمل درآمد کی اکائیاں بانٹتا ہے۔ اسی لیے ایک تھریڈ پر اعداد ملتے جلتے لگ سکتے ہیں اور فرق تب ظاہر ہوتا ہے جب سب مصروف ہوں — اور مصنوعہ بالکل اسی طرح کام کرتا ہے: nginx اپنے ورکرز کے ساتھ، PHP-FPM اپنے پول کے ساتھ اور ڈیٹابیس اپنے کنکشنز کے ساتھ تمام vCPU بیک وقت استعمال کرتے ہیں، ایک نہیں۔

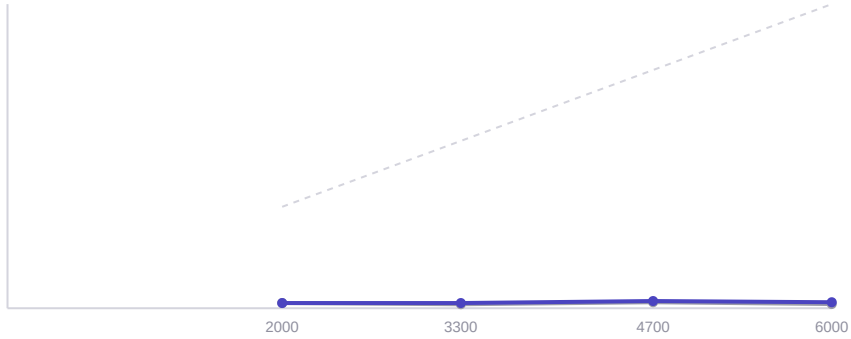
3 طریقۂ کار

▸ ٹیسٹ ماحول

پیرامیٹر	ARM64	X86_64
خطہ	us-east-1	us-east-1
انسٹینس قسم	c7g.large	c6i.large
CPU اور میموری	vCPU · 3799 MiB 2	vCPU · 3811 MiB 2
پروسیسر	Neoverse-V1	Intel(R) Xeon(R) Platinum 8375C CPU @ 2.90GHz
کرنل	aws-7.0.0-1017	aws-7.0.0-1017

> ہر آرکیٹیکچر اپنے ہی ٹیسٹ انسٹینس پر ماپا جاتا ہے، اور صرف اُس وقت جب وہ انسٹینس مصنوعات کے ٹیسٹ پاس کر چکا ہو۔ پیمائش کے دوران مصنوعہ چل رہا ہوتا ہے — خریدار کی مشین دراصل ایسی ہی ہوتی ہے۔

4 لوڈ منحنی



مطلوبہ شرح کے مقابلے میں پوری کی گئی درخواستیں۔ نقطہ دار قطر مطلوبہ شرح ہے: جہاں لکیر اس سے ہٹتی ہے، وہاں مشین سیر ہو چکی ہے۔

مطلوبہ شرح	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

▲ اُس درجے کو ظاہر کرتا ہے جہاں مشین مطلوبہ شرح برقرار نہیں رکھ پاتی۔ اس کے بعد p99 قطار کا انتظار ناپتا ہے، تاخیر نہیں۔

لوڈ ایک الگ مشین سے آتا ہے — دونوں آرکیٹیکچرز کے لیے ایک ہی انسٹینس قسم، اسی ایویلیبلٹی زون میں — جو ماپی جانے والی مشین کو اس کے نجی پتے پر مستقل آمد کی شرح سے درخواستیں بھیجتی ہے۔ دونوں مرتبہ کلائنٹ ایک ہی ہے: صرف جواب دینے والی مشین بدلتی ہے۔

▸ گنجائش اور کام کی لاگت

مد	ARM64	X86_64	Δ ARM64
زیادہ سے زیادہ برقرار شرح	req/s 142,4	req/s 119,5	▲ +19,1 %

ریمپ کا آغاز اس سے اوپر ہوا جو دونوں میں سے ایک برداشت کر سکتی تھی، اس لیے اس کے لیے کوئی صاف زینہ نہیں اور جو دکھایا گیا ہے وہ مشاہدہ شدہ زیادہ سے زیادہ تھروپٹ ہے: ایک حد، برقرار شرح نہیں۔

زیادہ سے زیادہ برقرار شرح وہ آخری زینہ ہے جسے مشین نے مکمل سرو کیا، گراف کی چوٹی نہیں: سیر شدہ زینہ بہت کچھ دیتا ہے مگر جواب برا دیتا ہے، اور اسے گنجائش شمار کرنا ایسے بوجھ کا وعدہ ہوگا جہاں مصنوعہ پہلے ہی لڑکھڑا رہا ہے۔ فی دس لاکھ درخواستیں لاگت اسی شرح پر نکالی جاتی ہے، اور جب دونوں مشینیں ایک جیسا بوجھ نہ اٹھائیں تو یہی موازنہ قائم رہتا ہے: فی گھنٹہ قیمت بتاتی ہے مشین چالو رکھنے کی لاگت، اس کے کیے گئے کام کی نہیں۔

5 نتیجہ

Imaxe Monitor Solution - Zabbix 7.0 کو Graviton (arm64) پر شائع کیا جاتا ہے کیونکہ یہ بہتر ماپتا ہے اور کم لاگت رکھتا ہے: مساوی x86_64 انسٹینس پر +14,1 % وسطی برتری۔

> اس رپورٹ کے تمام اعداد شائع ہونے والی امیج پر چلائے گئے ٹیسٹ سے ہیں۔ جو ٹیسٹ دونوں آرکیٹیکچرز پر نہیں ماپے جا سکے وہ اسی طرح دکھائے گئے ہیں اور وسط سے باہر ہیں۔

A ضمیمہ — چلائے گئے کمانڈز

بالکل وہی جو چلایا گیا، تاکہ اعداد دہرائے یا چیلنج کیے جا سکیں۔

ٹیسٹ	کما نڈ
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`