बेंचमार्क · ARM64 बनाम X86_64

Imaxe Monitor Solution - Zabbix 7.0 1.2

हम जो Graviton इमेज प्रकाशित करते हैं और उसी स्रोत से बनी x86_64 बिल्ड की, समकक्ष इंस्टेंस प्रकारों पर मापी गई तुलना।

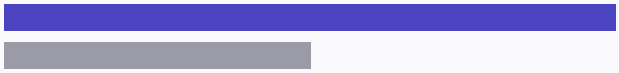
संदर्भ PERF-zabbix70-1.2-20261107 निर्मित 2026-11-07 us-east-1

सार

मापे गए परीक्षणों में Graviton +14,1 % माध्यिका प्रदर्शन देता है।

पूरे दस्तावेज़ में रंग परिपाटी: **ARM64 (Graviton)** · **x86_64**. अंतर के साथ चिह्न (▲ / ▼) भी है, ताकि श्वेत-श्याम में भी पढ़ा जा सके।

1 परिणाम

परीक्षण	ARM64 · C7G.LARGE	X86_64 · C6I.LARGE	Δ ARM64	तुलना
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 %	

tुलनीय परीक्षणों में Graviton का माध्यिका लाभ: +14,1 %। प्रत्येक बार अपनी ही जोड़ी के बड़े मान के अनुपात में है, क्योंकि हर परीक्षण की इकाई अलग है।

2 थ्रेड्स पर स्केलिंग

पैरामीटर	ARM64	X86_64
CPU, 1 थ्रेड	**1113 events/s**	1074 events/s
CPU, सभी थ्रेड	**2221 events/s**	1113 events/s
स्केलिंग गुणक	**×2,00 (2 vCPU)**	×1,04 (2 vCPU)
आदर्श की तुलना में दक्षता	**99,8 %**	51,8 %



आदर्श की तुलना में दक्षता

एक थ्रेड से सभी vCPU तक जाने पर Graviton का काम 2,00 गुना बढ़ता है और x86_64 इंस्टेंस का 1,04 गुना।

> दोनों परिवारों में vCPU एक ही चीज़ नहीं है: Graviton में यह एक भौतिक कोर है, Intel में यह एक SMT थ्रेड है जो अपने जुड़वाँ के साथ निष्पादन इकाइयाँ साझा करता है। इसीलिए एक थ्रेड पर आंकड़े मिलते-जुलते दिख सकते हैं और अंतर तभी उभरता है जब सभी व्यस्त हों — और उत्पाद ठीक ऐसे ही काम करता है: nginx अपने वर्कर्स के साथ, PHP-FPM अपने पूल के साथ और डेटाबेस अपने कनेक्शनों के साथ सभी vCPU एक साथ इस्तेमाल करते हैं, एक नहीं।

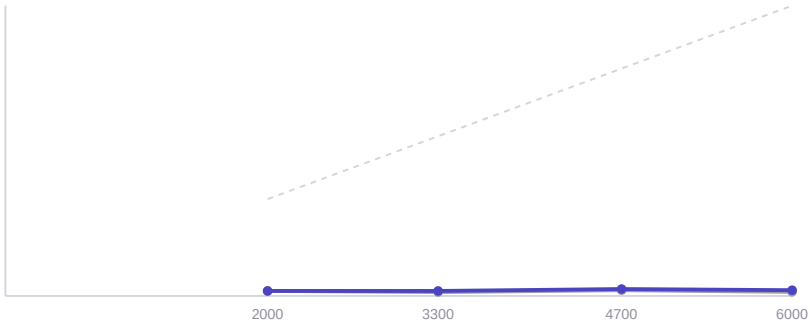
3 कार्यप्रणाली

▸ परीक्षण वातावरण

पैरामीटर	ARM64	X86_64
क्षेत्र	`us-east-1`	`us-east-1`
इंस्टेंस प्रकार	`c7g.large`	`c6i.large`
CPU और मेमोरी	`2 vCPU · 3799 MiB`	`2 vCPU · 3811 MiB`
प्रोसेसर	`Neoverse-V1`	`Intel(R) Xeon(R) Platinum 8375C CPU @ 2.90GHz`
कर्नेल	`7.0.0-1017-aws`	`7.0.0-1017-aws`

> प्रत्येक आर्किटेक्चर अपने ही टेस्ट इंस्टेंस पर मापा जाता है, और तभी जब वह इंस्टेंस उत्पाद के परीक्षण पास कर चुका हो। माप के दौरान उत्पाद चालू रहता है — खरीदार की मशीन वास्तव में ऐसी ही होती है।

4 लोड वक्र



अनुरोधित दर की तुलना में पूरी की गई अनुरोध संख्या। बिंदुदार विकर्ण अनुरोधित दर है: जहाँ रेखा उससे अलग होती है, वहाँ मशीन संतृप्त है।

अनुरोधित दर	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

▲ उस चरण को दर्शाता है जहाँ मशीन अनुरोधित दर नहीं बनाए रख पाती। उसके आगे p99 प्रतीक्षा-कतार मापता है, विलंब नहीं।

लोड एक अलग मशीन से आता है — दोनों आर्किटेक्चर के लिए एक ही इंस्टेंस प्रकार, एक ही उपलब्धता क्षेत्र में — जो मापी जा रही मशीन को उसके निजी पते पर स्थिर आगमन दर से अनुरोध भेजती है। दोनों बार क्लाइंट एक ही है: बदलती है तो सिर्फ़ जवाब देने वाली मशीन।

▸ क्षमता और काम की लागत

मद	ARM64	X86_64	Δ ARM64
अधिकतम टिकाऊ दर	**142,4 req/s**	119,5 req/s	**▲ +19,1 %**

रैंप वहाँ से शुरू हुआ जो दोनों में से एक की क्षमता से ऊपर था, इसलिए उसके लिए कोई साफ़ चरण नहीं है और जो दिखाया गया है वह देखा गया अधिकतम थ्रूपुट है: एक सीमा, टिकाऊ दर नहीं।

अधिकतम टिकाऊ दर वह अंतिम चरण है जिसे मशीन ने पूरा सर्व किया, ग्राफ़ का शिखर नहीं: संतृप्त चरण बहुत कुछ देता है पर जवाब खराब देता है, और उसे क्षमता मानना ऐसा भार वादा करना होगा जिस पर उत्पाद पहले ही लड़खड़ा रहा है। प्रति दस लाख अनुरोध लागत इसी दर पर निकाली जाती है, और जब दोनों मशीनें एक जैसा भार नहीं झेलतीं तो यही तुलना टिकती है: प्रति घंटा मूल्य बताता है मशीन चालू रखने की लागत, उसके किए काम की नहीं।

5 निष्कर्ष

Imaxe Monitor Solution - Zabbix 7.0 को Graviton (arm64) पर प्रकाशित किया जाता है क्योंकि यह बेहतर मापता है और कम लागत का है: समकक्ष x86_64 इंस्टेंस पर +14,1 % का माध्यिका लाभ।

इस रिपोर्ट के सभी आँकड़े प्रकाशित की जा रही इमेज पर चलाए गए परीक्षण से हैं। जो परीक्षण दोनों आर्किटेक्चर पर नहीं मापे जा सके, वे ऐसे ही दर्शाए गए हैं और माध्यिका से बाहर हैं।

A परिशिष्ट — निष्पादित कमांड

> ठीक वही जो चलाया गया, ताकि आँकड़े दोहराए या चुनौती दिए जा सकें।

परीक्षण	कमांड
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`