বেঞ্চমার্ক · ARM64 বনাম X86_64

Imaxe Monitor Solution - Zabbix 7.0 1.2

আমরা যে Graviton ইমেজ প্রকাশ করি এবং একই উৎস থেকে তৈরি x86_64 বিল্ডের, সমতুল্য ইনস্ট্যান্স ধরনে পরিমাপ করা তুলনা।

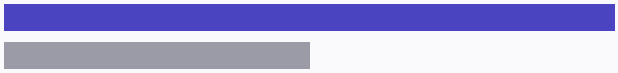
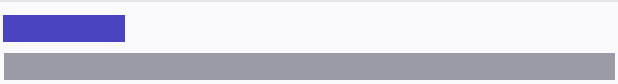
সূত্র PERF-zabbix70-1.2-20261107 তৈরি 2026-11-07 us-east-1

সারসংক্ষেপ

পরিমাপ করা পরীক্ষাগুলিতে Graviton +14,1 % মধ্যমা কর্মক্ষমতা দেয়।

সম্পূর্ণ নথিতে রঙের রীতি: **ARM64 (Graviton)** · **x86_64**. পার্থক্যের সঙ্গে চিহ্নও (▲ / ▼) আছে, যাতে সাদা-কালোয় পড়া যায়।

1 ফলাফল

পরীক্ষা	ARM64 · C7G.LARGE	X86_64 · C6I.LARGE	Δ ARM64	তুলনা
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 %	

তুলনাযোগ্য পরীক্ষাগুলিতে Graviton-এর মধ্যমা সুবিধা: +14,1 %। প্রতিটি বার তার নিজের জোড়ার বড়টির সাপেক্ষে মাপা, কারণ প্রতিটি পরীক্ষার একক আলাদা।

2 থ্রেডজুড়ে স্কেলিং

প্যারামিটার	ARM64	X86_64
CPU, ১ থ্রেড	**1113 events/s**	1074 events/s
CPU, সব থ্রেড	**2221 events/s**	1113 events/s
স্কেলিং গুণক	**×2,00 (2 vCPU)**	×1,04 (2 vCPU)
আদর্শের তুলনায় দক্ষতা	**99,8 %**	51,8 %



আদর্শের তুলনায় দক্ষতা

এক থ্রেড থেকে সব vCPU-তে গেলে Graviton-এর কাজ 2,00 গুণ বাড়ে, x86_64 ইনস্ট্যান্সের 1,04 গুণ।

> দুই পরিবারে vCPU এক জিনিস নয়: Graviton-এ এটি একটি ভৌত কোর, Intel-এ এটি একটি SMT থ্রেড যা যমজের সঙ্গে এক্সিকিউশন ইউনিট ভাগ করে। তাই এক থ্রেডে সংখ্যা কাছাকাছি দেখাতে পারে, আর পার্থক্য দেখা দেয় তখনই যখন সবগুলো ব্যস্ত — পণ্যটি ঠিক এভাবেই কাজ করে: nginx তার worker নিয়ে, PHP-FPM তার pool নিয়ে আর ডেটাবেস তার সংযোগ নিয়ে একসঙ্গে সব vCPU ব্যবহার করে, একটি নয়।

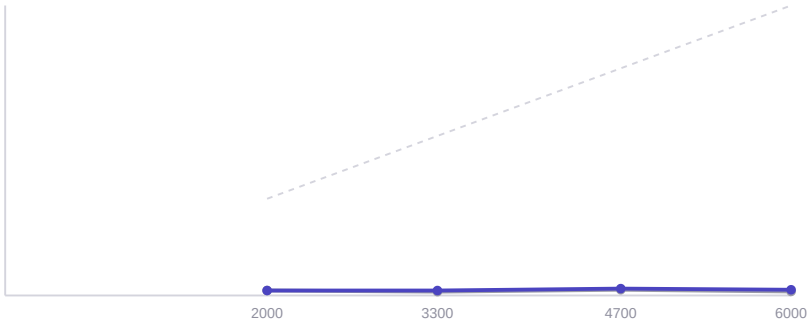
3 পদ্ধতি

▸ পরীক্ষার পরিবেশ

প্যারামিটার	ARM64	X86_64
অঞ্চল	us-east-1	us-east-1
ইনস্ট্যান্স ধরন	c7g.large	c6i.large
CPU ও মেমরি	2 vCPU · 3799 MiB	2 vCPU · 3811 MiB
প্রসেসর	Neoverse-V1	Intel(R) Xeon(R) Platinum 8375C CPU @ 2.90GHz
কার্নেল	7.0.0-1017-aws	7.0.0-1017-aws

> প্রতিটি আর্কিটেকচার তার নিজস্ব টেস্ট ইনস্ট্যান্সে পরিমাপ করা হয়, এবং কেবল সেই ইনস্ট্যান্স পণ্যের পরীক্ষায় উত্তীর্ণ হওয়ার পরেই। পরিমাপের সময় পণ্যটি চালু থাকে — ক্রেতার মেশিন আসলে এমনই থাকে।

4 লোড বক্ররেখা



অনুরোধকৃত হারের বিপরীতে সরবরাহ করা অনুরোধ। ছেদচিহ্নিত কর্ণটি অনুরোধকৃত হার: যেখানে রেখা তা থেকে সরে যায়, সেখানেই মেশিন সম্পৃক্ত।

অনুরোধকৃত হার	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

> ▲ সেই ধাপ চিহ্নিত করে যেখানে মেশিন আর অনুরোধকৃত হার ধরে রাখতে পারে না। তার পরে p99 সারিতে অপেক্ষা মাপে, বিলম্ব নয়।

> লোড তৈরি করে আলাদা একটি মেশিন — দুই আর্কিটেকচারের জন্য একই ইনস্ট্যান্স টাইপ, একই অ্যাভেইলেবিলিটি জোনে — যা মাপা মেশিনটিকে তার প্রাইভেট ঠিকানায় ধ্রুব আগমন-হারে আঘাত করে। দুই দফাতেই ক্লায়েন্ট অভিন্ন: বদলায় শুধু যে মেশিনটি উত্তর দেয়।

▸ ধারণক্ষমতা ও কাজের খরচ

খাত	ARM64	X86_64	Δ ARM64
সর্বোচ্চ টেকসই হার	**142,4 req/s**	119,5 req/s	**▲ +19,1 %**

> র‍্যাম্প শুরু হয়েছে দুইয়ের একটির সহনক্ষমতার উপরে, তাই সেটির জন্য কোনো পরিচ্ছন্ন ধাপ নেই এবং দেখানো হয়েছে পর্যবেক্ষিত সর্বোচ্চ থ্রুপুট: একটি ছাদ, টেকসই হার নয়।

> সর্বোচ্চ টেকসই হার হলো শেষ যে ধাপটি মেশিন পুরোপুরি সার্ভ করেছে, গ্রাফের চূড়া নয়: সম্পৃক্ত ধাপ অনেক কিছু দেয় কিন্তু সাড়া দেয় খারাপ, আর সেটিকে ধারণক্ষমতা ধরা মানে এমন লোডের প্রতিশ্রুতি দেওয়া যেখানে পণ্যটি আগেই হিমশিম খাচ্ছে। প্রতি দশ লক্ষ অনুরোধের খরচ ওই হারেই হিসাব করা হয়, আর দুই মেশিন সমান লোড না নিলে এই তুলনাটিই টেকে: ঘণ্টাপ্রতি দাম বলে মেশিন চালু রাখার খরচ, তার করা কাজের খরচ নয়।

5 উপসংহার

Imaxe Monitor Solution - Zabbix 7.0 Graviton (arm64)-এ প্রকাশিত হয় কারণ এটি ভালো মাপে এবং কম খরচ করে: সমতুল্য x86_64 ইনস্ট্যান্সের তুলনায় +14,1 % মধ্যমা সুবিধা।

> এই প্রতিবেদনের সব সংখ্যা প্রকাশিত ইমেজে চালানো পরীক্ষা থেকে। যে পরীক্ষাগুলি দুই আর্কিটেকচারে মাপা যায়নি সেগুলি তেমনভাবেই দেখানো এবং মধ্যমার বাইরে।

A পরিশিষ্ট — চালানো কমান্ড

> ঠিক যা চালানো হয়েছে, যাতে সংখ্যাগুলি পুনরুৎপাদন বা প্রশ্ন করা যায়।

পরীক্ষা	কমান্ড
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`