Enable up to 40% better price-performance with AWS Graviton2 based Amazon EC2 instances

Jeff Underhill — AWS – Amazon EC2 Principal Business Development Manager
Arthur Petitpierre — AWS – Amazon EC2 Specialist Solutions Architect
Sudhir Raman — AWS – Amazon EC2 Principal Product Manager

Broadest and deepest platform choice

CATEGORIES

General purpose

Burstable

Compute intensive

Memory intensive

Storage (High I/O)

Dense storage

GPU compute

Graphics intensive

+

CAPABILITIES

Choice of processor
(AWS, Intel, AMD)

Fast processors
(up to 4.0 GHz)

High memory footprint
(up to 12 TiB)

Instance storage
(HDD and SSD)

Accelerated computing
(GPUs and FPGA)

Networking
(up to 100 Gbps)

Bare Metal

Size
(Nano to 32xlarge)

+

OPTIONS

Amazon EBS

Amazon Elastic Inference

=

MORE THAN

275

INSTANCE TYPES

for virtually every workload and business need

aws

Broadest choice of processors

Intel® Xeon Scalable processors

AMD EPYC processors

Graviton processors

aws

AWS Graviton processors

Custom AWS silicon with 64-bit Arm Neoverse cores

Targeted optimizations for cloud-native workloads

Rapidly innovate, build, and iterate on behalf of customers

aws

First instance powered by AWS Graviton processor

Announced at re:Invent 2018

Amazon EC2 A1

Optimized cost and performance for scale-out applications

Significant cost savings

AWS Graviton Processor with 64-bit Arm Neoverse cores and custom AWS silicon

Applications

Scale-out workloads

Web tier

Containerized microservices

Arm-based software development

Configurations

6 instance sizes

Up to 16 vCPUs, 32GiB memory

Up to 10 Gbps NW, 3.5 Gbps EBS

Bare metal options

Availability

9 Regions

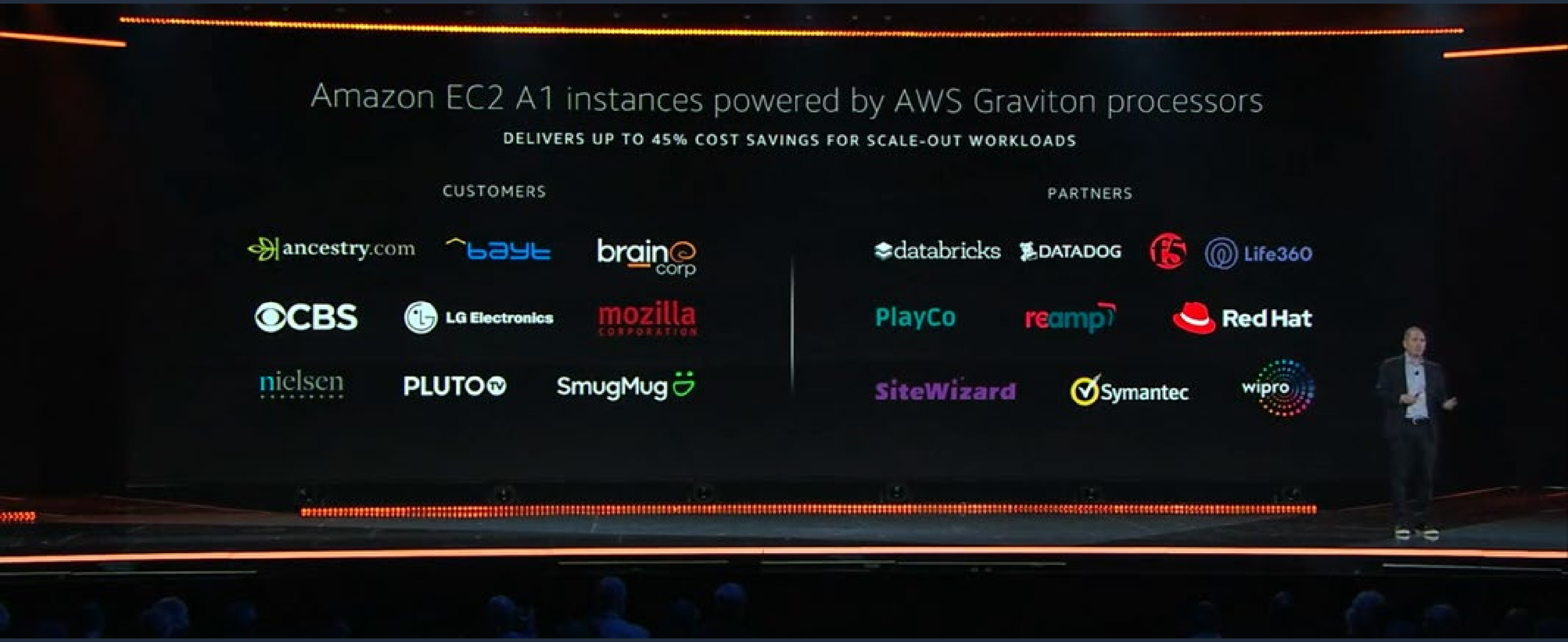
US (N. Virginia, Oregon, Ohio)

EU (Ireland, Frankfurt)

APAC (Mumbai, Sydney, Tokyo, Singapore)

aws

Amazon EC2 A1 instances customer success stories



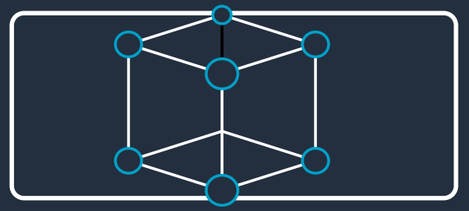
aws

Broadening workloads and target applications

Web and gaming servers

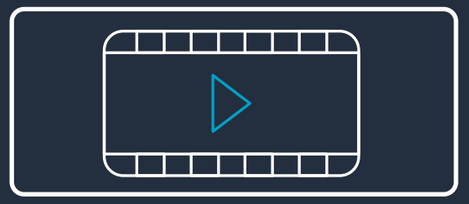
Open-source databases

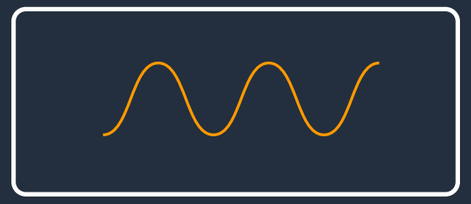
High performance computing



In-memory caches

Media encoding





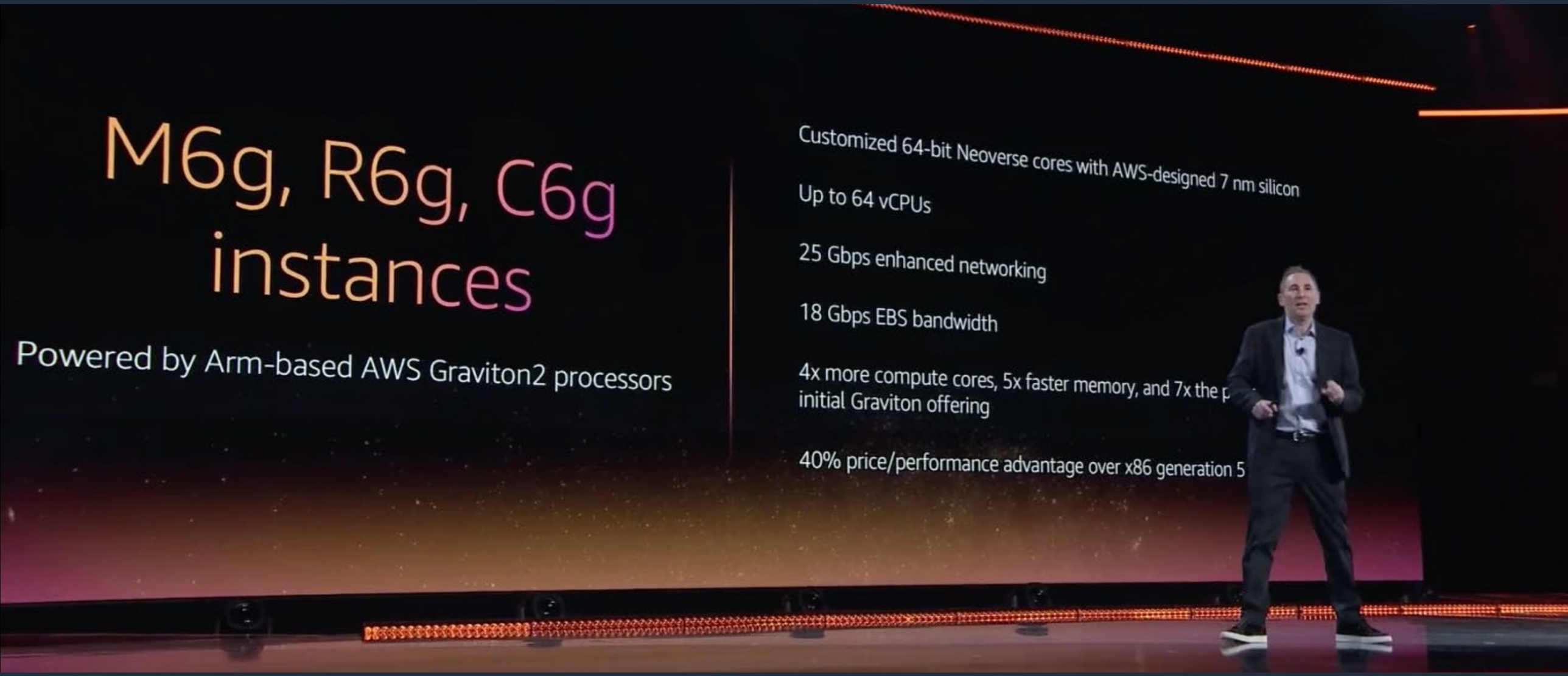
EDA

Analytics

Microservices

aws

re:Invent 2019 – Graviton2-based C6g, M6g, R6g instances



aws

AWS Graviton2 processor vs first generation AWS Graviton

AWS Graviton2 processor

4X	5X	7X
compute cores	faster memory	performance

aws

AWS Graviton2 based instances

Up to 40% better price-performance for general purpose, compute intensive, and memory intensive workloads.

M6g

Built for: General-purpose workloads such as application servers, mid-size data stores, and microservices.

C6g

Built for: Compute intensive applications such as HPC, video encoding, gaming, and simulation workloads.

R6g

Built for: Memory intensive workloads such as open-source databases, or in-memory caches.

Available Now

Coming Soon

Local NVMe-based SSD storage options will also be available in general purpose (M6gd), compute-optimized (C6gd), and memory-optimized (R6gd) instances

aws

Amazon EC2 M6g: sizes and specifications

20% lower cost vs. M5

Instance	vCPUs	Memory (GB)	Network Bandwidth (Gbps)	EBS Optimized	EBS Bandwidth (Mbps)	EBS Optimized Burst Bandwidth (Mbps)
m6g.medium	1	4	Up to 10	Yes	315	4,750
m6g.large	2	8	Up to 10	Yes	630	4,750
m6g.xlarge	4	16	Up to 10	Yes	1,188	4,750
m6g.2xlarge	8	32	Up to 10	Yes	2,375	4,750
m6g.4xlarge	16	64	Up to 10	Yes	4,750	4,750
m6g.8xlarge	32	128	12Gbps	Yes	9,000	9,000
m6g.12xlarge	48	192	20Gbps	Yes	13,500	13,500
m6g.16xlarge	64	256	25Gbps	Yes	19,000	19,000
m6g.metal	64	256	25Gbps	Yes	19,000	19,000

aws

Lower TCO with Graviton2 powered instances

Highest performance in their instance families

M6g offers 20% lower cost vs M5

Up to 40% better price/performance vs comparable instances

Best price performance within their instance families

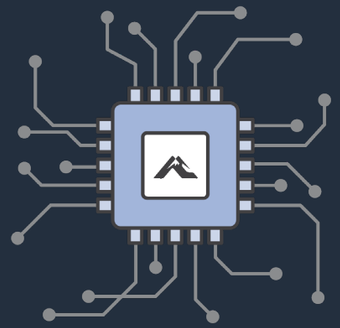
aws

AWS Graviton2 processor deep dive

aws

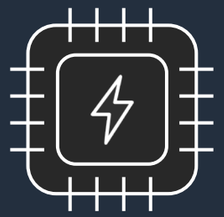
AWS Graviton2 powered instances

Graviton2



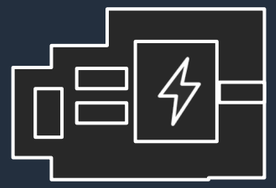
Industry leading performance

Nitro Security Chip



Integrated into motherboard
Protects hardware resources

Nitro Card



Amazon Elastic Block Store,
Elastic Network Adapter
Monitoring, and security

Nitro Hypervisor

Lightweight hypervisor
Memory and CPU allocation
Bare Metal-like performance

Exclusive purpose built & Modular building blocks

aws

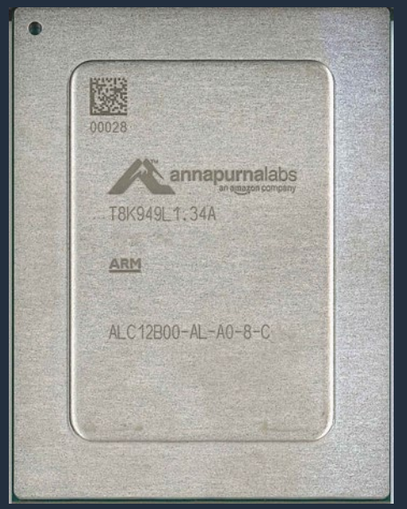
Leap to AWS Graviton2

AWS Graviton processor

- First Arm processor in AWS
- First-class citizen in EC2
- 16nm
- ~5 Billion transistors

AWS Graviton2 processor

- 4x the vCPUs
- 7x CPU performance
- ~2x performance/vCPU
- 7nm
- ~30 Billion transistors



aws

AWS Graviton2 - Cores

Arm® Neoverse™ N1 cores

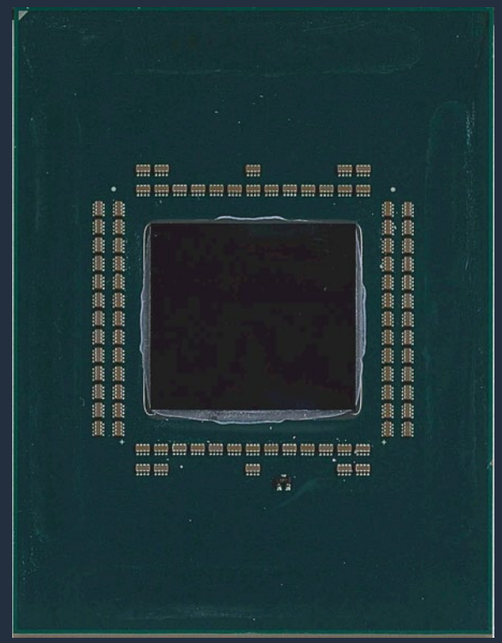
Arm v8.2 compliant

Worked closely with Arm on creation of N1

- Large 64KB L1 caches and 1MB L2 cache/vCPU
- Coherent Instruction cache
- Lower overheads of interrupts, virtualization, and context switching
- 4-wide front-end, with 8-wide dispatch/issue
- Dual-SIMD units
- Instructions to accelerate ML inference: int8, fp16

Every vCPU is a physical core

- No simultaneous multithreading (SMT)



aws

AWS Graviton2 - Interconnect

64 cores connected together with a mesh

~2TB/s bisection bandwidth

32MB LLC

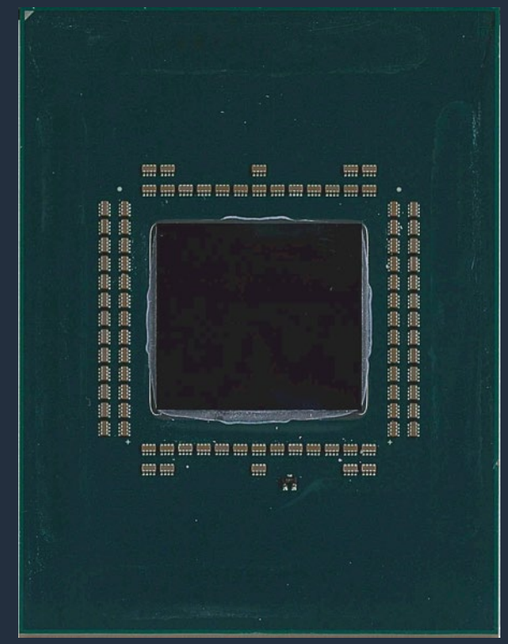
- With the private caches over 100MB of user-accessible caches

No NUMA concerns

- Every core sees the same path to memory and to other cores

64 lanes of PCIe gen4

- Provide flexibility for different instance configurations



aws

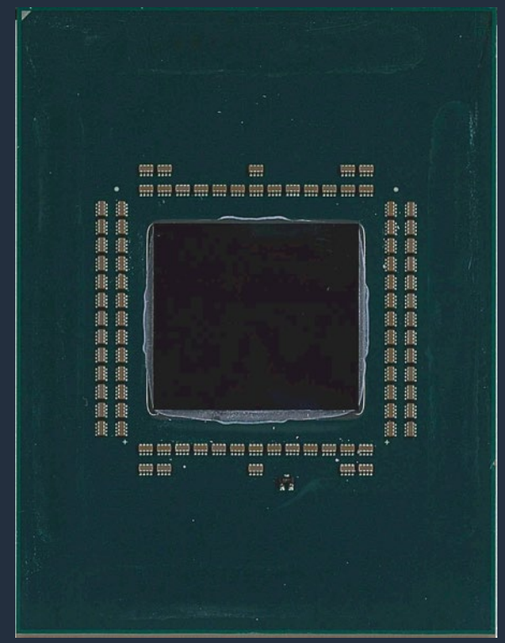
AWS Graviton2 - System

8x DDR4-3200 channels → over 200GB/s

- Always AES-256 encrypted DRAM with ephemeral key
- Uniform memory latency from all CPU cores

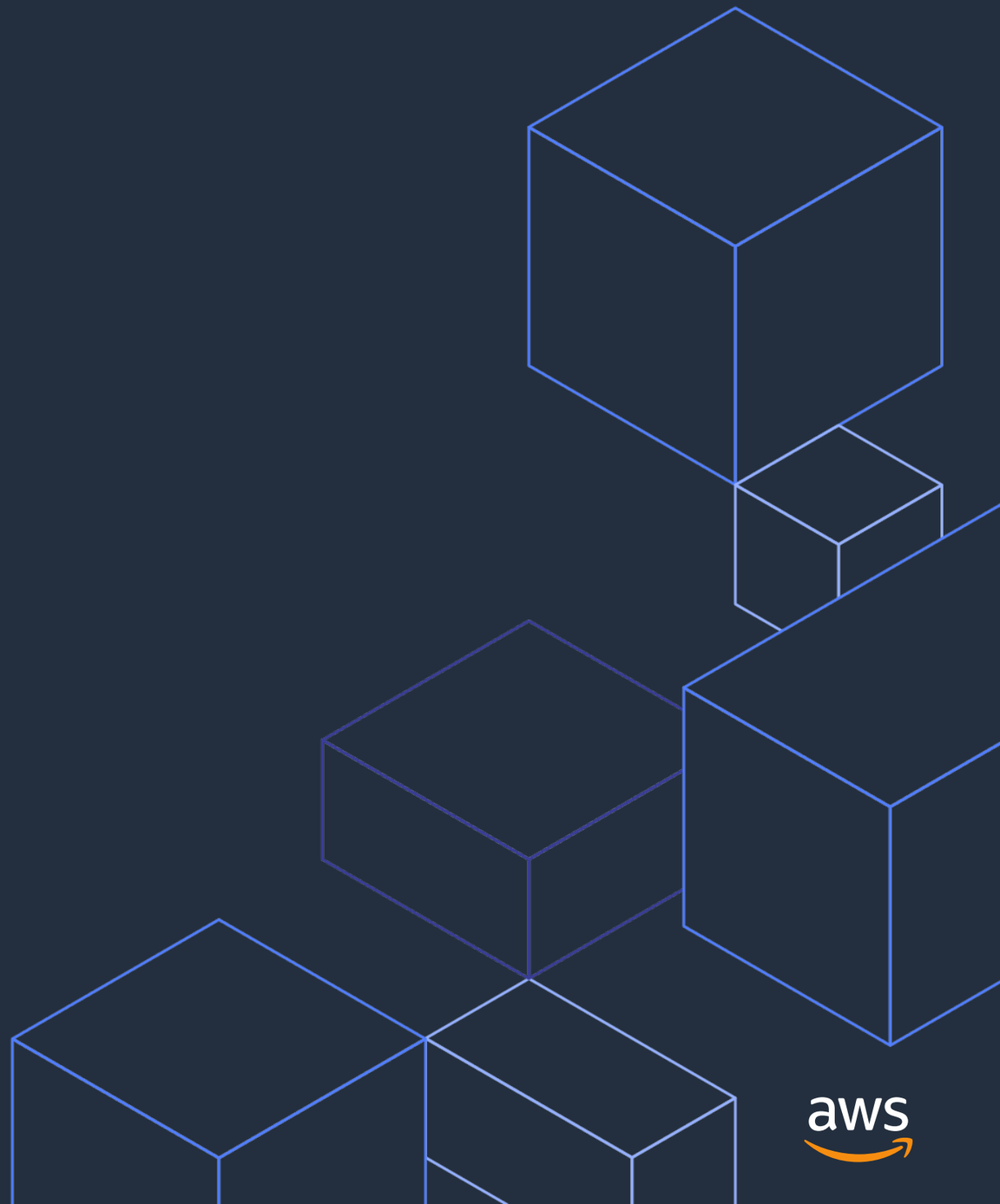
1Tbit/s of compression accelerators

- 2xlarge and larger instances will have a compression device
- DPDK and Linux kernel drivers will be available ahead of GA
- Data compression at up to 15GB/s and decompression at up to 11GB/s



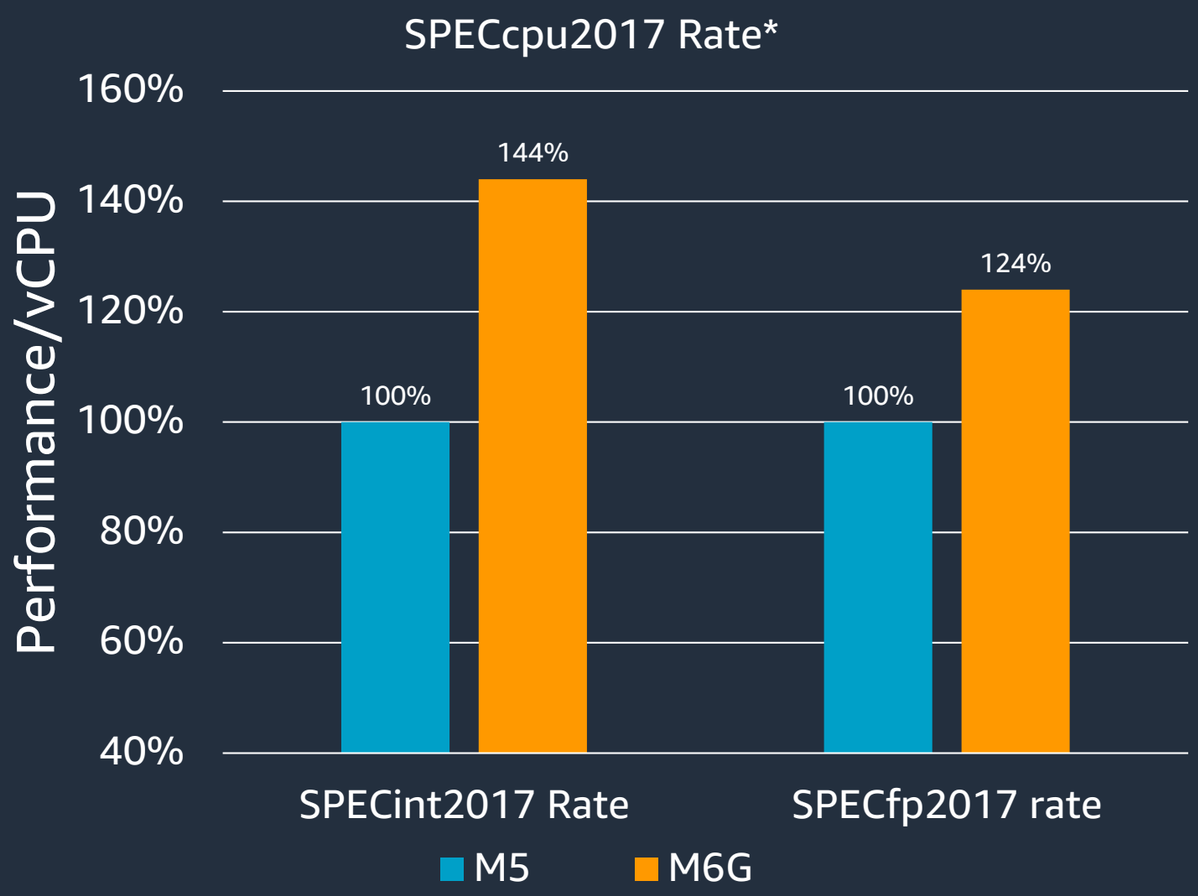
aws

M6g performance



SPEC cpu2017

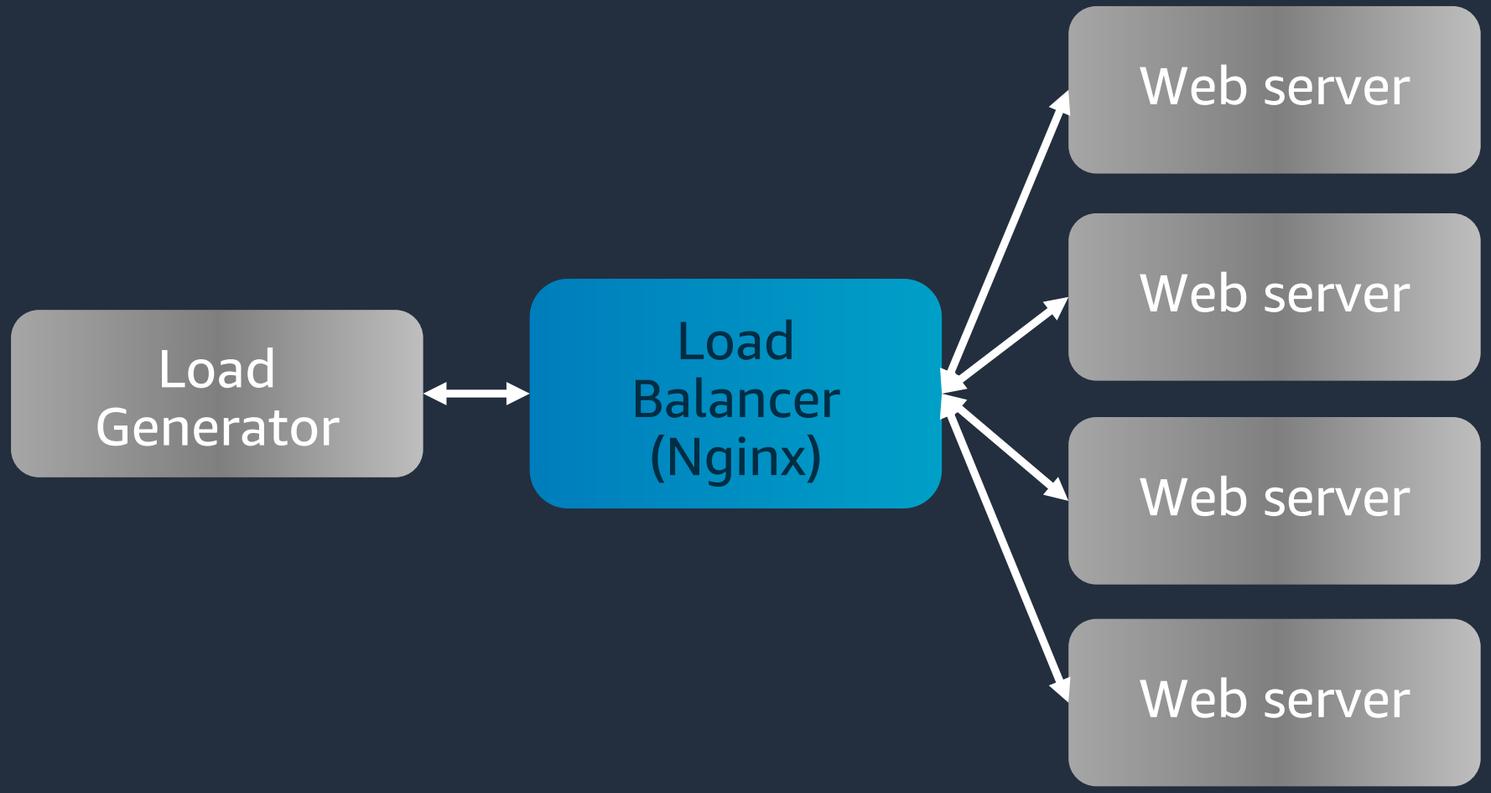
- Industry standard CPU intensive benchmark
- Run on all vCPUs concurrently
- Comparing performance/vCPU

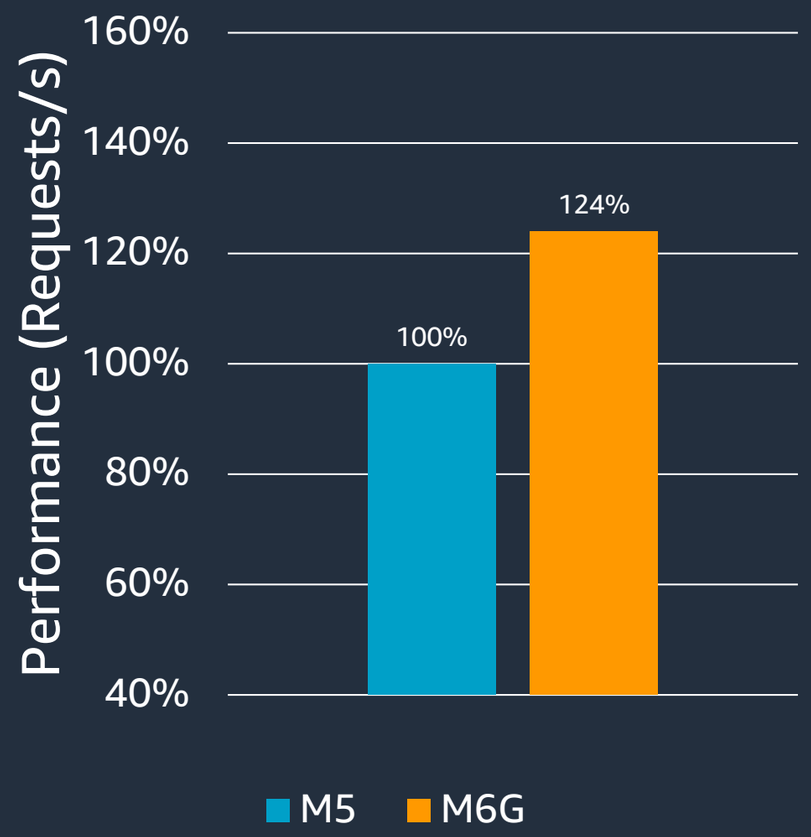


* All SPEC scores estimates, compiled with GCC9 -O3 -march=native, run on largest single socket size for each instance type tested.

aws

Load Balancing with Nginx

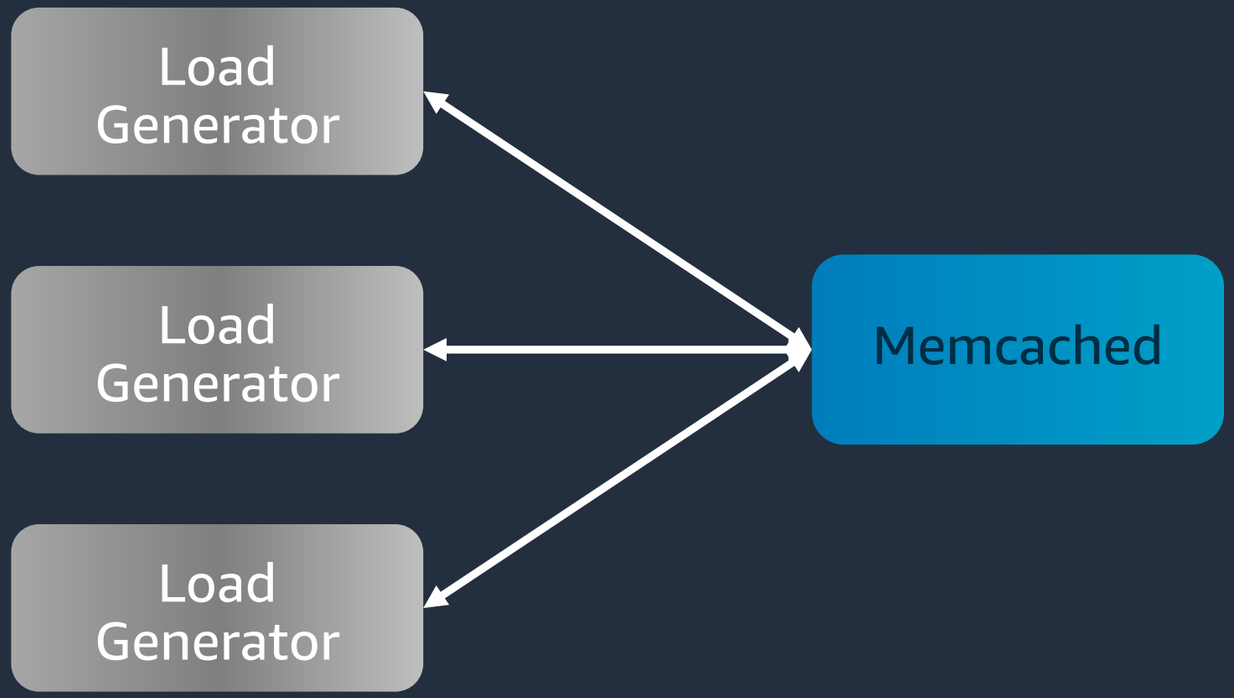


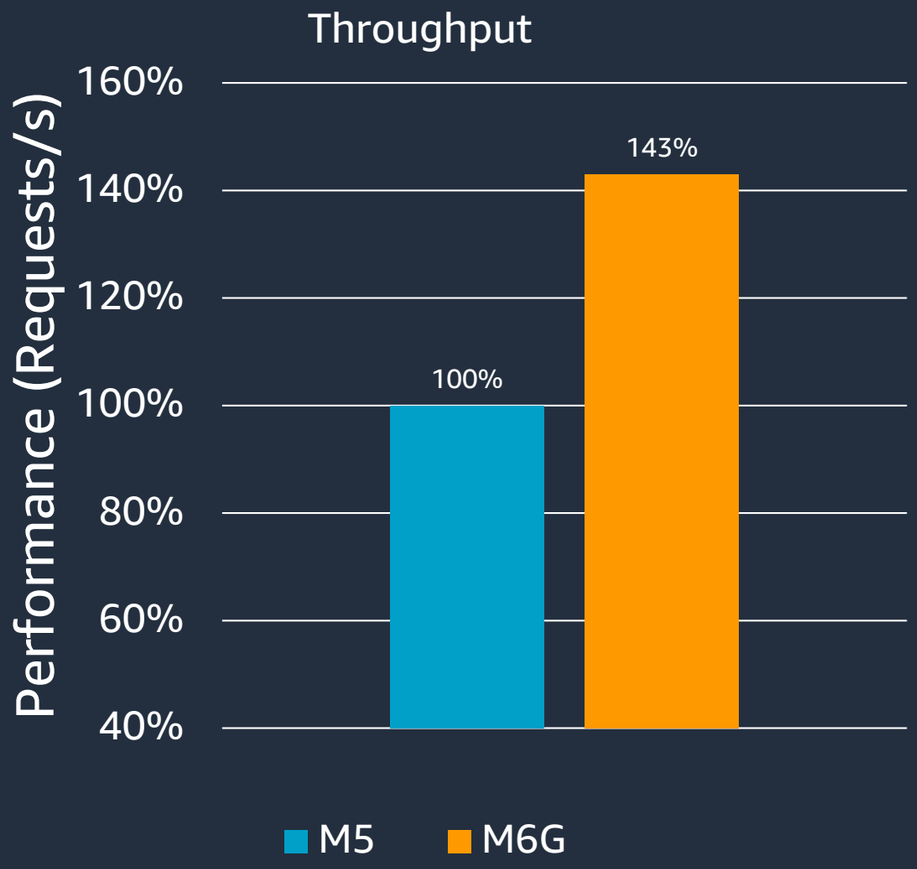


NGINX v1.15.9, 512 clients, 128 GET/POST payloads, all HTTPS connections, AES128-GCM-SHA256, OpenSSL 1.1.1, 4 target machines, all tests run on 4xl size; load generator c5.9xl; web servers c5.4xls; All servers run in a cluster placement group

aws

Memcached

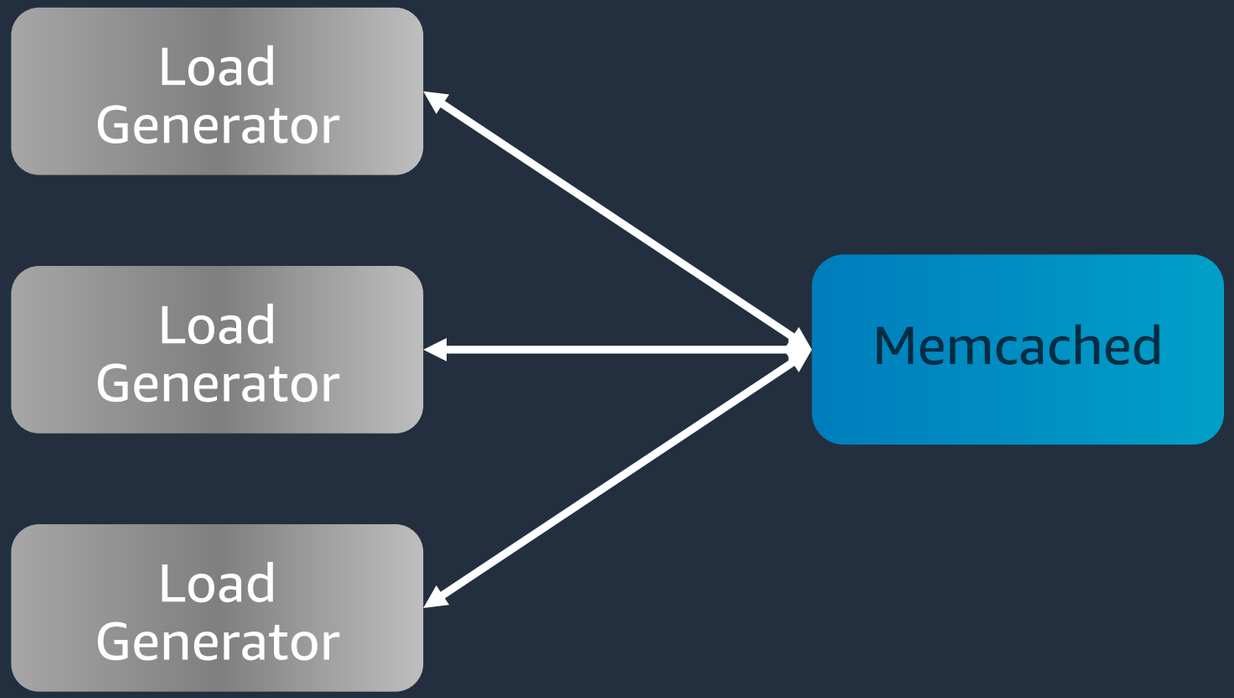


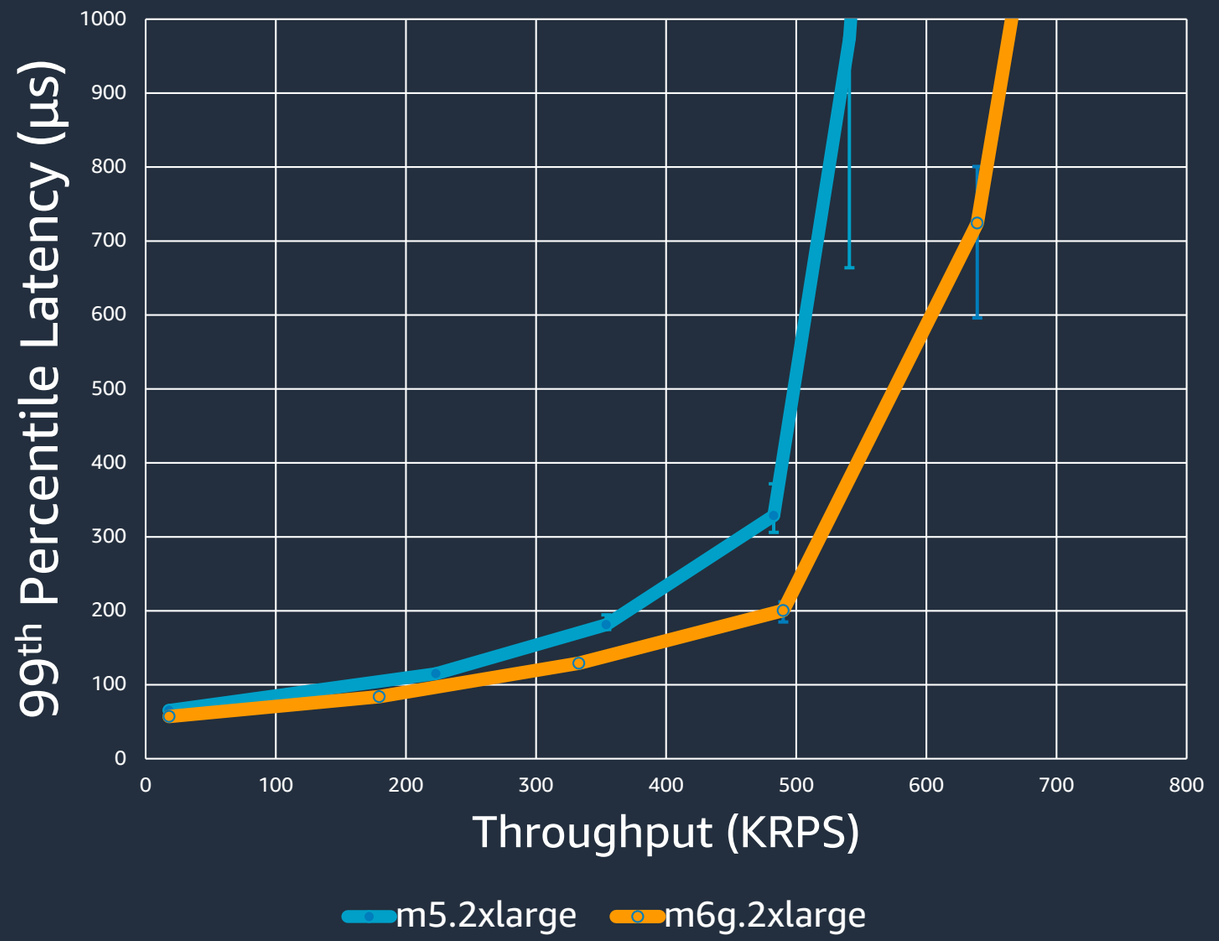


Memcached v1.5.16, 16B keys, 128B values, 7.8M KV-pairs, 576 connections for load generation from 2x c5.9xlarge instances, 16 additional connections used to measure latency from 1 additional c5.9xlarge,;each connection maintains 4096 outstanding requests; All servers in a cluster placement group

aws

Memcached

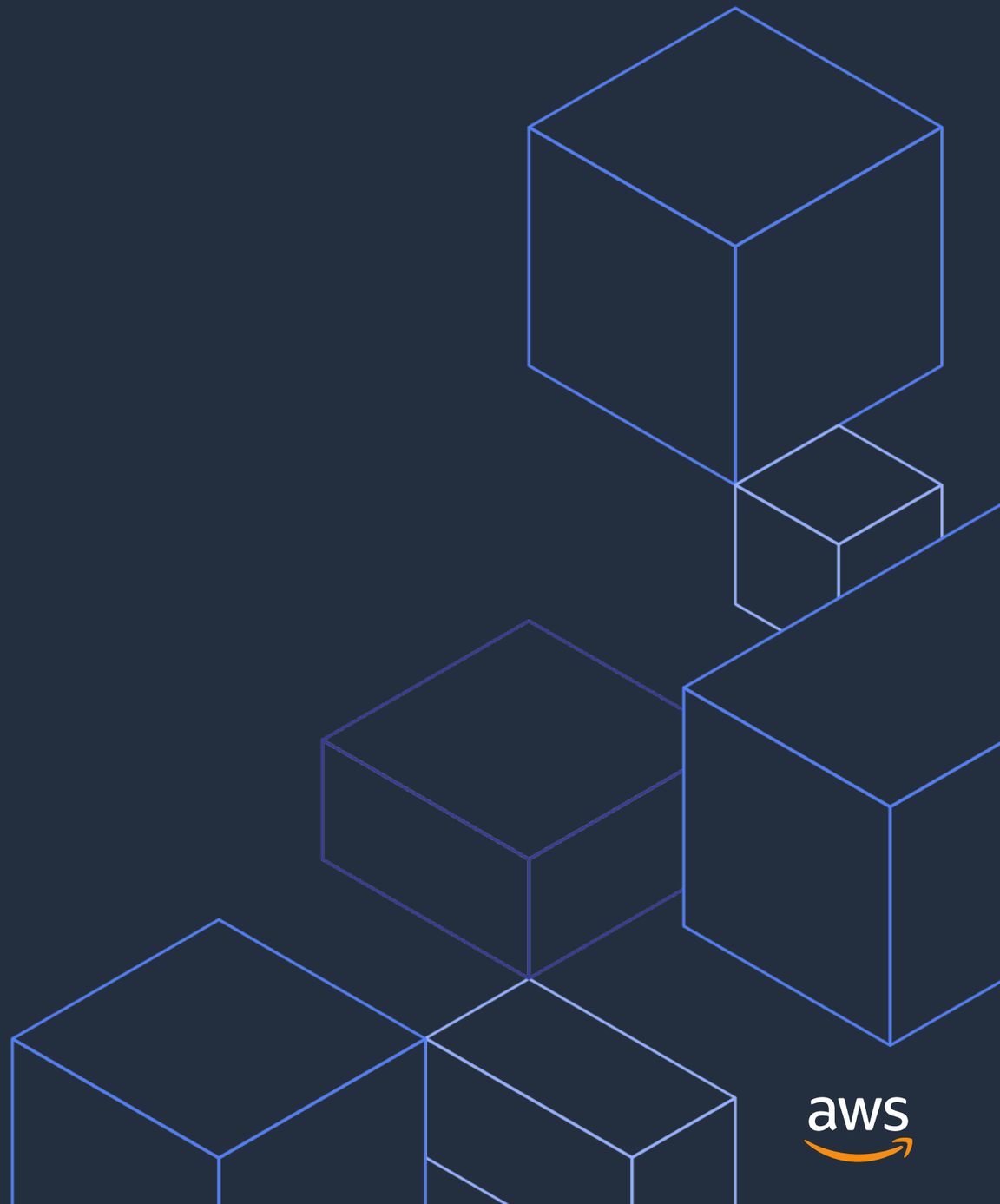




Memcached v1.5.16, 16B keys, 128B values, 7.8M KV-pairs, 576 connections for load generation from 2x c5.9xlarge instances, 16 additional connections used to measure latency from 1 additional c5.9xlarge, each connection maintains 4 outstanding requests; all servers in a cluster placement group

aws

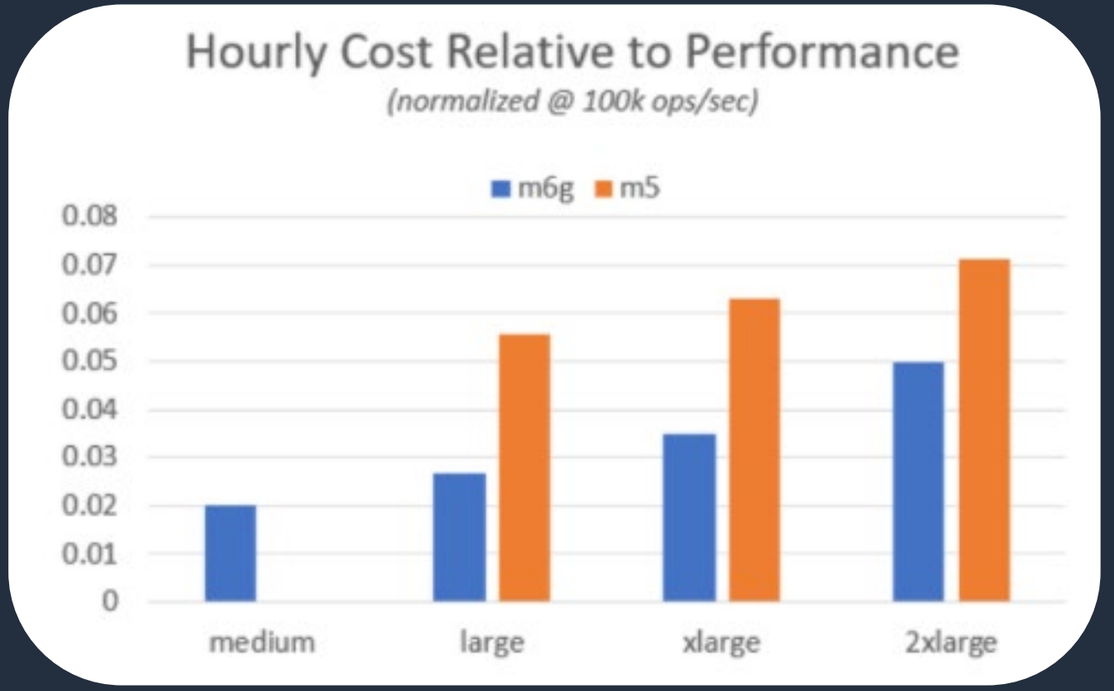
M6g feedback



KeyDB – M6g up to 65% faster than M5

When it comes to the cost of work, some M6g instances can be over 2X cheaper when looking at computing cost / performance. The m6g.medium provides the best bang for your buck with the m6g.large and m6g.xlarge also with major benefits.

Looking at straight up performance, the m6g.large is 1.65X faster than m5.large and 1.45X faster comparing the ‘xlarge’ instances.



https://docs.keydb.dev/blog/2020/03/02/blog-post/

Honeycomb.io – Observations on M6g Instances

As of today, we've shifted 100% of our dogfood shepherd workload to run on M6g, using 30% fewer instances than we used with C5.

Liz Fong-Jones
Principal Developer Advocate,
at Honeycomb.io

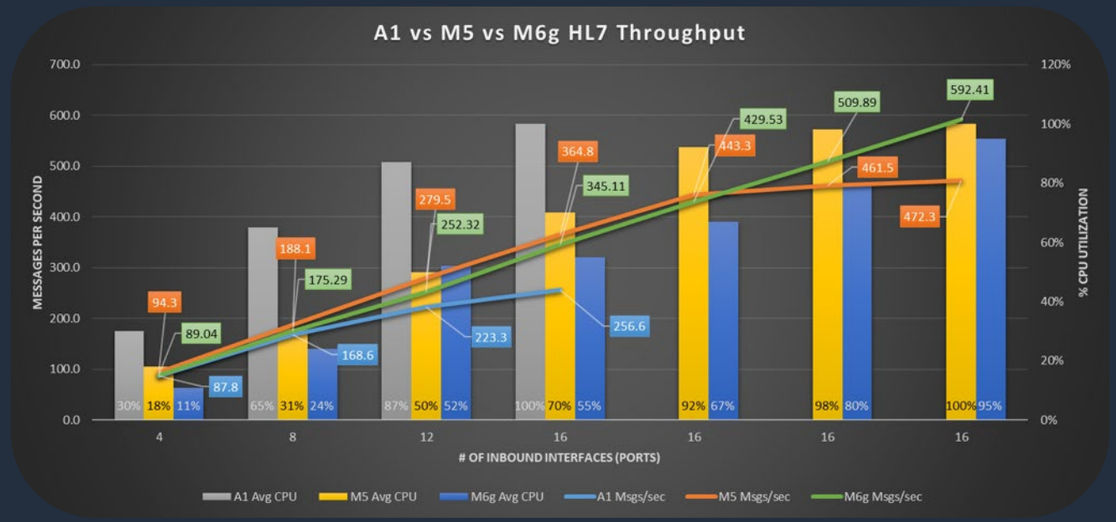
https://www.honeycomb.io/blog/observations-on-arm64-awss-amazon-ec2-m6g-instances/

aws

InterSystems IRIS on AWS Graviton2 Processors

We at InterSystems are very excited to see the performance gains and cost savings that AWS Graviton2 processors will provide to InterSystems IRIS customers. We anticipate that these combined benefits will drive significant adoption of Arm-based platforms among IRIS customers, and we look forward to providing support in 2020!

InterSystems™
IRIS Data Platform



https://community.intersystems.com/post/intersystems-iris-arm-based-aws-graviton2-processors

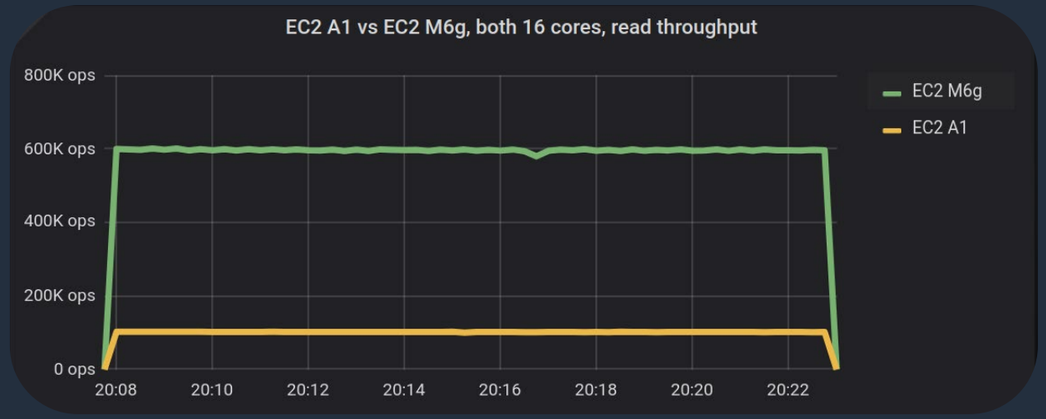
aws

How about a database workload?

- Scylla is a high-throughput, low latency Big Data database
  - Thread-per-core architecture guarantees full CPU utilization
  - I/O Scheduler guarantees peak I/O throughput
  - Can easily reach AWS I3's 15GB/s bandwidth limit

SCYLLA.

- Can you use Scylla on A1 instance?
  - Yes. It works, and it is supported
  - But doesn't reach peak performance



- M6g instances change the game
  - All CPU and memory-bound workloads are supported
  - 4xlarge → 37.5k reads/s/core; 5x improvement over A1!
  - 64 vCPU theoretical limit around 2.4M reads/s
  - Amazon EBS is still instance storage

aws

AnandTech review

We've been hearing about Arm in the server space for many years now, with many people claiming "it's coming"; "it'll be great", only for the hype to fizzle out into relative disappointment once the performance of the chips was put under the microscope. Thankfully, this is not the case for the Graviton2: not only were Amazon and Arm able to deliver on all of their promises, but they've also hit it out of the park in terms of value against the incumbent x86 players.

https://www.anandtech.com/show/15578/cloud-clash-amazon-graviton2-arm-against-intel-and-amd

aws

M6g customer feedback

hotelbeds

We were excited to test out the new AWS Graviton2 processors for one of our workloads developed using Java11 + SpringBoot2. Our initial testing shows that the Amazon EC2 M6g instances based on AWS Graviton2 deliver up to 43% better price performance vs. the current generation M5 instances.

Mobiuspace recently tested its Java-based containerized backend services on the new AWS Graviton2 based Amazon EC2 M6g instances and observed a performance improvement of 40% compared to the M5 instances. Due to this performance improvement and 20% lower price, Mobiuspace is looking forward to adopting them.

For our compute centric .NET Core based workloads that we have tested on the new Graviton2 based M6g instances, we have been excited to see a 30% performance gain over the existing 5th generation instances we are currently using in production today.

For more visit: https://aws.amazon.com/ec2/graviton/

aws

AWS Graviton software ecosystem momentum

AWS Graviton software ecosystem momentum

Operating Systems

Amazon Linux 2

ubuntu

Ubuntu 18.04LTS, 20.04LTS

Red Hat Enterprise Linux 7.6 and 8.x

SUSE
We adapt. You succeed.

SUSE Linux Enterprise Server 15

Debian

fedora

FreeBSD

NetBSD

Containers

docker

Docker Desktop Community and Docker Enterprise Engine

Amazon Elastic Container Service (Amazon ECS)

Amazon Elastic Kubernetes Service (Amazon EKS)

Amazon Elastic Container Registry (Amazom ECR)

Firecracker

Micro VMs

Tools and software

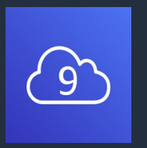
AWS Marketplace

AWS Systems Manager

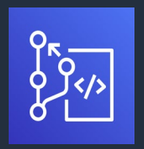
Amazon CloudWatch

Amazon Inspector

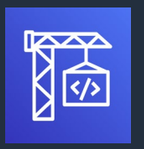
AWS Batch



AWS Cloud9



AWS CodeCommit



AWS CodeBuild

AWS CodePipeline

Amazon Corretto OpenJDK

aws

AWS Graviton software ecosystem momentum


Jenkins


GitLab


Drone.io


GitHub


GitHub Actions


Travis CI


Chef


Nginx+


Honeycomb


AWS CodeDeploy


CrowdStrike


DataDog

Rapid7


Qualys


Tenable

aws

Transitioning to AWS Graviton2 based instances

aws

Languages, toolkits and runtimes

- AWS Tools are available: *awscli, cloudwatch agent, SSM agent*
- AWS SDKs are available: *C/C++, node.js, Python, Go, Java*

- Interpreted and compiled-bytecode languages: *Python, Java, Ruby, PHP, node.js* – Can run without modification
    - .Net Core: Runs on Linux and Arm64

- Compiled applications: Need to be recompiled for Arm64

- AMI's: Most Linux distributions, and NetBSD/FreeBSD have Arm64 versions

- Containers: Need Arm64 versions of container images
    - Majority of Docker official images are already available for Arm64
    - Can be generated with CodeBuild and/or Docker Buildx

aws

A quick look at containers

- Amazon ECS is fully supported on AWS Graviton

- Amazon EKS is in preview:
- https://github.com/aws/containers-roadmap/tree/master/preview-programs/eks-arm-preview

- Amazon ECR supports multi-architecture manifest lists:
- https://aws.amazon.com/about-aws/whats-new/2020/05/ecr-now-supports-manifest-lists-for-multi-architecture-images/

- AWS CodeBuild support Arm64 instances:
- https://aws.amazon.com/blogs/devops/build-arm-based-applications-using-codebuild/

- Docker has support for building multi-arch containers with buildx :
  https://community.arm.com/developer/tools-software/tools/b/tools-software-ides-blog/posts/getting-started-with-docker-for-arm-on-linux

aws

Transition strategy – identify dependencies

- Identify all third party libraries and dependencies
  - standard libraries
  - open-sources libraries
  - paid-for proprietary libraries

- Check if they support Arm64 and check the providers roadmap
  - If they don’t, and they are critical to your application, let us know via the AWS EC2 Forum: https://forums.aws.amazon.com/forum.jspa?forumID=30

aws

Transition strategy – test, infrastructure, and deployment

- Test! All tests must be ported and it may be necessary to write new tests. This is especially pertinent if you had to recompile your application.
  - Perform the usual range of unit testing, acceptance testing, and pre-prod testing.

- Update your infrastructure as code resources, such as AWS CloudFormation templates, to provision your application to arm64 instances.
  - This will likely be a simple change, modifying the instance type, AMI, and user data to reflect the change to the Arm instance.

- Perform a Green/Blue Deployment. Create your new Arm64 based stack alongside your existing stack and leverage Route 53 weighted routing to route 10% of requests to the new stack.
  - Monitor error rates, user behavior, load, and other critical factors in order to determine the health of the ported application in production.

aws

How to go further – AWS Graviton2 getting started guide

https://github.com/aws/aws-graviton-gettting-started

This guide has been assembled by our Graviton team and will help you transition and optimize your applications.

When compiling code, using the right compiler options can make a huge difference.

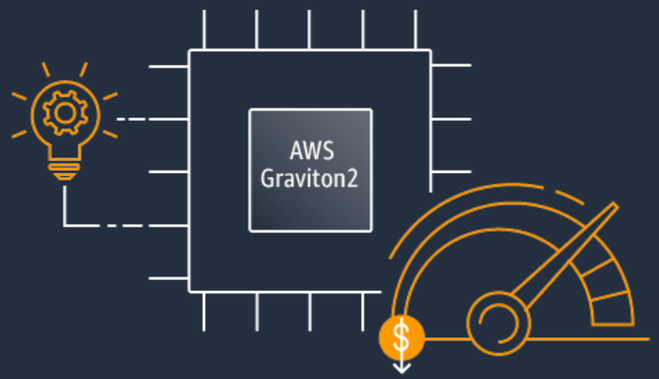
Using latest versions of operating systems, compilers and language runtimes will provide access to latest Arm64 optimizations.

aws

Summary

Amazon EC2 M6g, C6g, and R6g instances deliver up to 40% better price-performance compared to current generation M5, C5, and R5 instances

Most workload transitions are seamless, check out AWS Graviton2 today and let us know how you get on



aws

Q&A

Thank you

Jeff Underhill	AWS – Amazon EC2 Principal Business Development Manager
Arthur Petitpierre	AWS – Amazon EC2 Specialist Solutions Architect
Sudhir Raman	AWS – Amazon EC2 Principal Product Manager