[AWS Black Belt Online Seminar]

Amazon EC2 Deep Dive:

AWS Graviton2 Arm CPU 搭載インスタンス

サービスカットシリーズ

Specialist Solutions Architect, HPC
Daisuke Miyamoto
2020/07/07

AWS 公式 Webinar
https://amzn.to/JPWebinar

過去資料
https://amzn.to/JPArchive

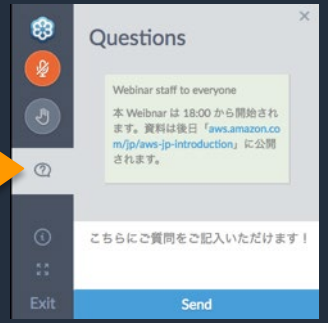
AWS Black Belt Online Seminar とは

「サービス別」「ソリューション別」「業種別」のそれぞれのテーマに分かれて、アマゾン ウェブ サービス ジャパン株式会社が主催するオンラインセミナーシリーズです。

質問を投げることができます！

- 書き込んだ質問は、主催者にしか見えません
- 今後のロードマップに関するご質問は
  お答えできませんのでご了承下さい

① 吹き出しをクリック
② 質問を入力
③ Sendをクリック



Twitter ハッシュタグは以下をご利用ください
#awsblackbelt

内容についての注意点

- 本資料では2020年07月07日時点のサービス内容および価格についてご説明しています。最新の情報はAWS公式ウェブサイト(http://aws.amazon.com)にてご確認ください。

- 資料作成には十分注意しておりますが、資料内の価格とAWS公式ウェブサイト記載の価格に相違があった場合、AWS公式ウェブサイトの価格を優先とさせていただきます。

- 価格は税抜表記となっています。日本居住者のお客様には別途消費税をご請求させていただきます。

- AWS does not offer binding price quotes. AWS pricing is publicly available and is subject to change in accordance with the AWS Customer Agreement available at http://aws.amazon.com/agreement/. Any pricing information included in this document is provided only as an estimate of usage charges for AWS services based on certain information that you have provided. Monthly charges will be based on your actual use of AWS services, and may vary from the estimates provided.

本セミナーの概要

❑ 本セミナーで学習できること

- ❖ Amazon EC2 の基礎
- ❖ AWS Graviton2 プロセッサとはなにか、どのようなメリットがあるのか
- ❖ AWS Graviton2 搭載インスタンスの詳細とベンチマーク、活用事例

❑ 対象者

- ❖ Amazon EC2 を利用しており、コスト最適化を進めたい方
- ❖ AWS Graviton2 プロセッサについて詳しく知りたい方
- ❖ 最新の Arm プロセッサに興味のある方
- ❖ 次の AWS のサービスの概要レベルの知識が前提になります

Amazon VPC / Amazon EC2 / Amazon S3 などのAWS基礎サービス

自己紹介

- 名前
  - 宮本 大輔 （みやもと だいすけ）
- 所属
  - アマゾン ウェブ サービス ジャパン 株式会社
  - 技術統括本部
  - Specialist Solutions Architect, HPC
- 好きな AWS サービス
  - AWS ParallelCluster
  - Amazon FSx for Lustre
  - AWS Snowball シリーズ

本日の流れ

- Amazon EC2とは
- AWS Graviton2 の概要
- AWS Graviton2 の詳細アーキテクチャ
- AWS Graviton2 の利用ガイド
- AWS Graviton2 の ベンチマーク
- AWS Graviton2 の活用事例

Amazon EC2 とは

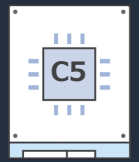
仮想サーバサービス Amazon EC2 (Elastic Compute Cloud)

- 必要なときに必要な計算リソースを確保可能な仮想サーバサービス
- ワークロードに応じて様々なインスタンスタイプを選択可能
- 数分で起動し、秒単位の従量課金（一部タイプについては1時間単位）
- インスタンスを停止するだけでマシンスペック変更が可能

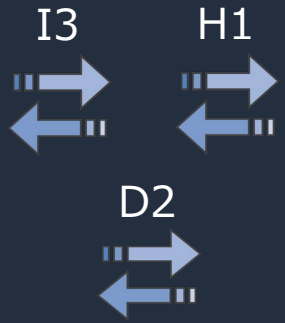
インスタンスタイプ一覧と分類

汎用

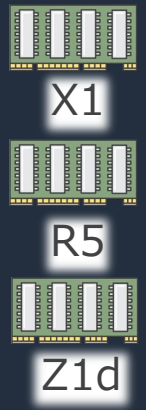
コンピューティング最適化



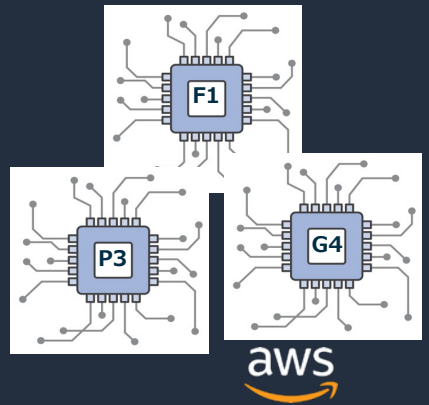
ストレージ・IO最適化



メモリ最適化



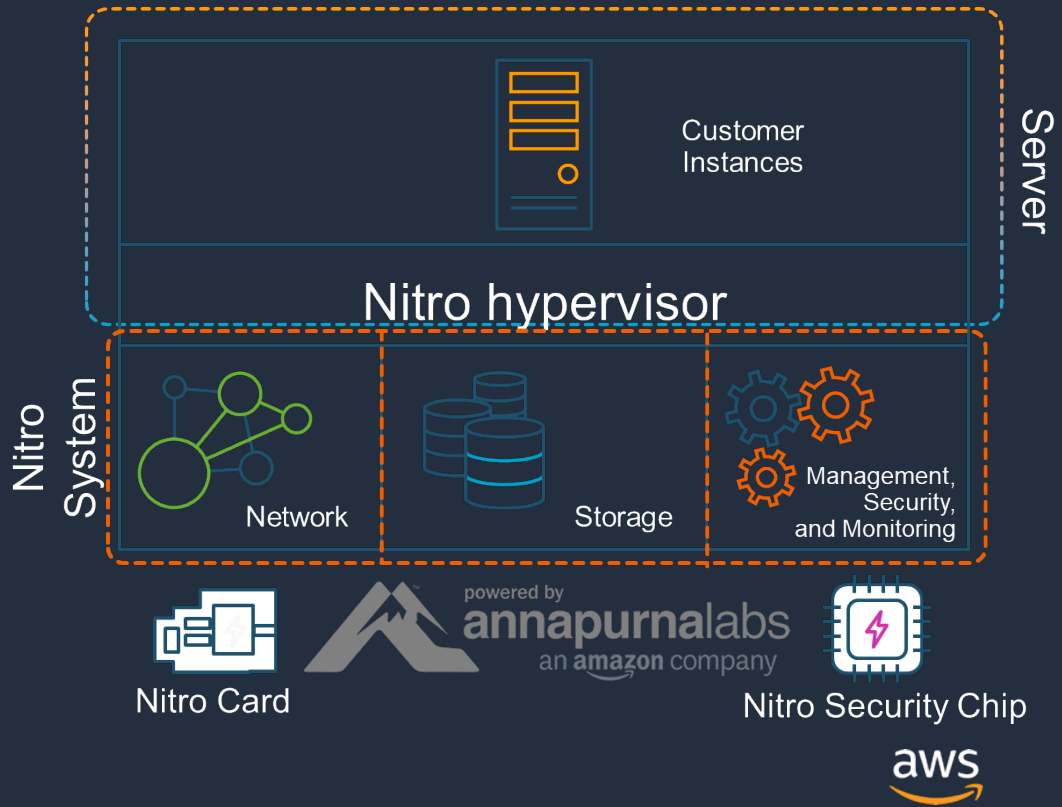
GPU ・FPGA アクセラレーテッド



EC2のシステム基盤 (AWS Nitro System)

独自のハードウェア/Hypervisorにより最適化された性能を提供

- C5,M5,R5など最新世代のインスタンスが対応
- 専用のASICやKVM ベースHypervisorにより、仮想化のオーバーヘッドを低減
- 幅広いインスタンスタイプでベアメタルタイプを提供



EC2インスタンスのネーミングポリシー

Amazon EC2 で選択できる高性能CPUの選択肢

Intel Xeon processor
(x86_64 arch)

最大3.9GHz駆動
Cascade Lakeコア搭載
C5インスタンス

AMD EPYC processor
(x86_64 arch)

最大3.3GHz駆動
Romeコア搭載
C5aインスタンス

AWS Graviton Processor
(64-bit Arm arch)

64bit Arm Neoverse N1ベース
Graviton2 CPU搭載
C6gインスタンス

アプリケーションとワークロードに応じて
最適なコンピューティング環境を選択

aws

EC2 購入オプション

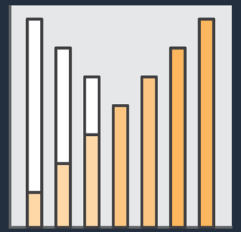
オンデマンドインスタンス

長期コミット無し、使用分への支払い(**秒単位/時間単位)**。Amazon EC2の定価

スパイクするようなワークロード

リザーブドインスタンス (Savings Plans)

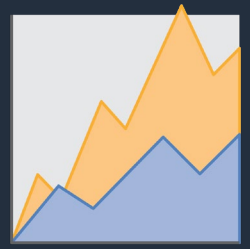
1年/3年の長期コミットをする代わりに**大幅なディスカウント**価格



一定の負荷の見通しがあるワークロード

スポットインスタンス

Amazon EC2の空きキャパシティを活用し、**最大90%値引き。中断が発生することがある**



中断に強く、かつ様々なインスタンスタイプを活用できるワークロード

ワークロードに合わせて購入方法を選択することで
コスト効率よくEC2を利用可能に

まとめ：Amazon EC2

- Amazon EC2 により、必要なときに必要な計算リソースを利用する事が可能
- 様々なインスタンスタイプが存在しており、ワークロードの特性に応じて活用することでコストパフォーマンスよく処理を行うことができる
- ワークロードの特性に応じて、購入オプション（オンデマンド、リザーブド、スポット）を選択

AWS Graviton2 とは

Arm アーキテクチャ とは？

ARM Ltd により設計・ライセンスされるプロセッサコアのアーキテクチャ
ARM Ltd は自社では製造を行わず、ライセンスを受けた他社が製造

- 高い消費電力性能
- 累計出荷数 1,500 億個以上（2018年時点）という実績
- 組み込みからスマートフォン、サーバ向けまで幅広く活用多くの採用実績による豊富なエコシステム

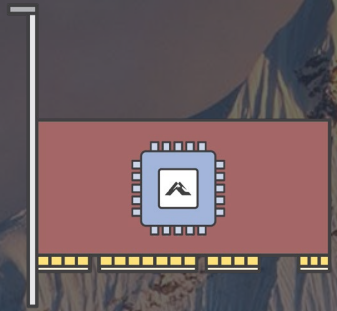
arm

Arm アーキテクチャの採用により
よりコストパフォーマンスの高いインスタンスを提供できるのではないか

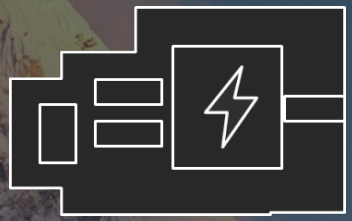
aws

Annapurna Labs によるチップ開発の歴史

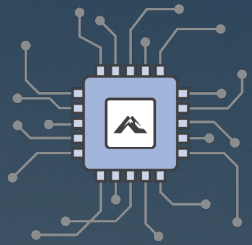
I/O Accelerator Card



Nitro Card



AWS Graviton, Inferentia

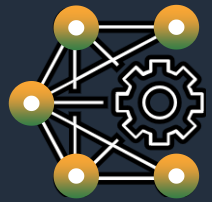


5+ years of innovation with Annapurna Labs

aws

第一世代 AWS Gravitonプロセッサ

64-bit Arm Neoverseコア採用
AWSカスタムチップ



クラウドネイティブワークロード
に最適化

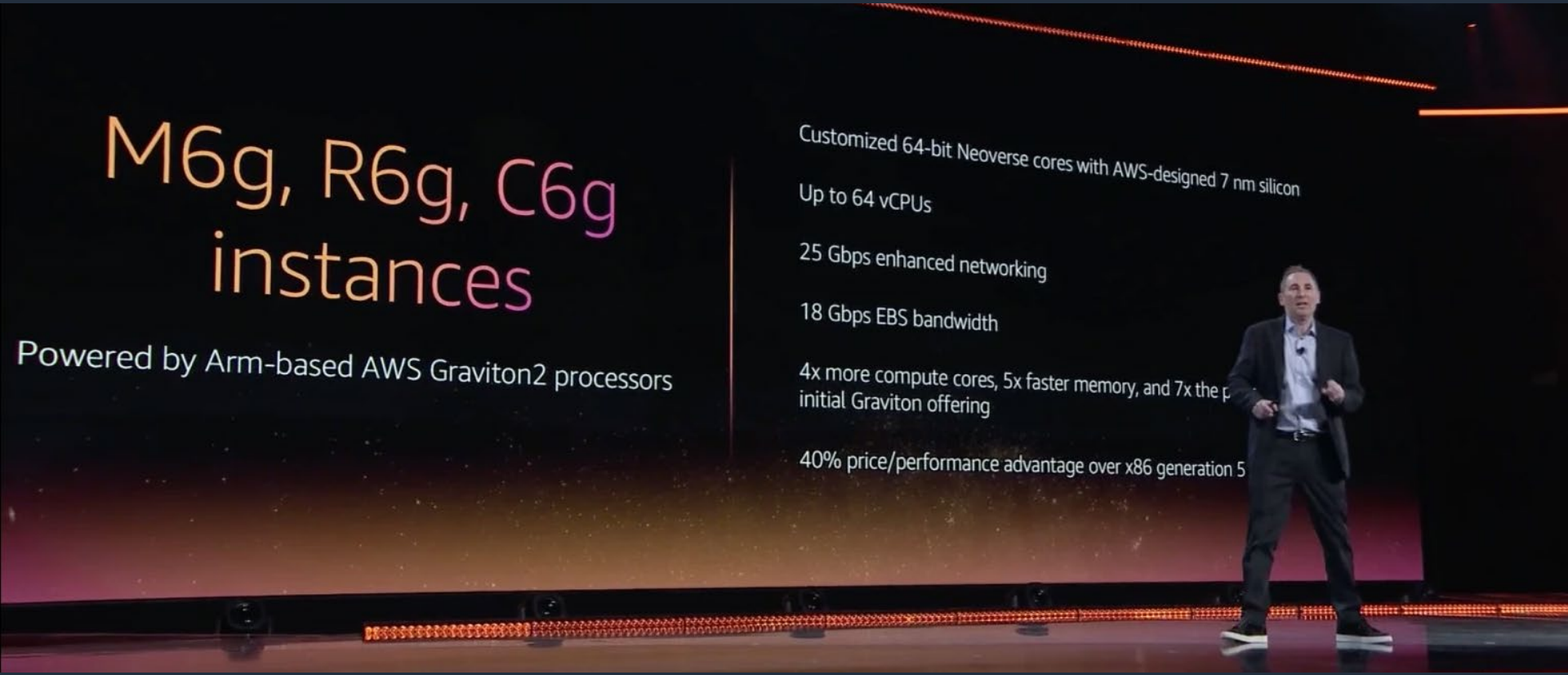
顧客からのフィードバックのもと
迅速に革新、構築、反復

A1 活用事例

nielsen

Nielsen processes **12B + jobs daily** and experiences significant **cost savings** with their scale out workloads on Amazon EC2 A1

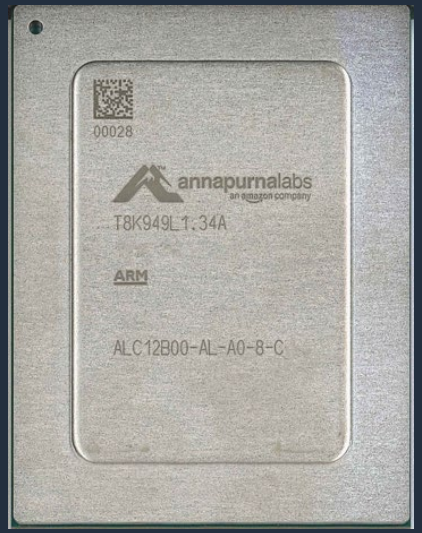
re:Invent 2019 – Graviton2 搭載 M6g, R6g, C6g 発表

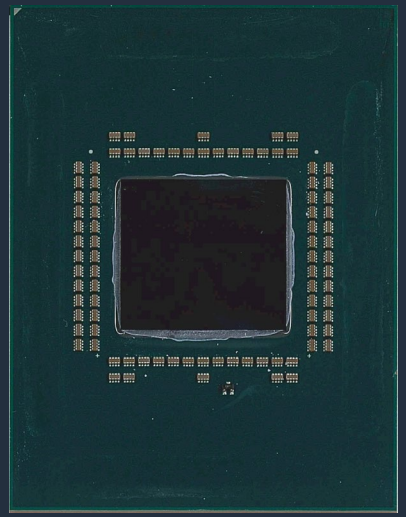


AWS Graviton2 プロセッサ

AWS が独自に設計した Arm Neoverse N1 コア採用のCPU

- 1 チップに 64 物理コアを搭載
- 7 nm プロセスルール、300億トランジスタ
- 前世代 AWS Graviton と比較して 4 倍の vCPU 数、7 倍の CPU 性能





６つのAWS Graviton2プロセッサ搭載インスタンスを発表

汎用	コンピューティング最適化	メモリ最適化
4GB DRAM/vCPU	2GB DRAM/vCPU	8GB DRAM/vCPU
M6g	**C6g**	**R6g**

M5 等の x86 系インスタンスと比較して
最大 40 % コストパフォーマンスが向上

東京リージョンを含む 6 のリージョンで利用可能

NVMe SSDをサポートするバリエーションも今後提供予定 (**M6gd**, **C6gd**, **R6gd**)

（2020年7月7日現在）

aws

参考：M6g インスタンスファミリー

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 Regions – US East (N.Virginia and Ohio), US West (Oregon), Europe (Ireland & Frankfurt), and Asia Pacific (Tokyo)

AWS Graviton2 搭載インスタンスの利点

コスト：

- x86 系インスタンスの同サイズと比較した場合、**約 20 % 安価**
    - 例：m6g.16xlarge: 3.168 USD/hour
      m5.16xlarge: 3.968 USD/hour

パフォーマンス：

- x86 系インスタンスでは **2 vCPU = 1 物理コア** だが、
  Arm 系インスタンスは **1 vCPU = 1 物理コア** であり、
  同インスタンスサイズでは 2 倍の物理コアが利用可能

複数のアプリケーションベンチマークで
既存の x86 系インスタンスと比較して約40 % のコストパフォーマンス向上

AWS Graviton2 搭載インスタンス利用時の注意点

x86 系インスタンスとはアーキテクチャが異なるため、OS やアプリケーションを Graviton2 のために用意する必要がある

- Arm 専用の AMI からインスタンスを起動する必要がある
- x86 系インスタンスからのインスタンスタイプの変更は不可能
- 自作のアプリケーションやパッケージマネージャで提供されていないアプリケーションについては、コンパイルを行う必要がある
- 未対応の商用アプリケーションは利用できない
- 1 物理コアの同士の比較では x86 系インスタンスより遅くなる可能性がある

AWS の提供する Arm エコシステム

Operating Systems

- Amazon Linux 2
- Ubuntu 18.04LTS, 20.04LTS
- Red Hat Enterprise Linux 7.6 and 8.x
- SUSE Linux Enterprise Server 15
- Debian
- fedora
- FreeBSD
- NetBSD

Containers

- Docker Desktop Community and Docker Enterprise Engine
- Amazon Elastic Container Service (Amazon ECS)
- Amazon Elastic Kubernetes Service (Amazon EKS Preview)
- 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

拡大し続ける Arm エコシステム


Jenkins


GitLab


Drone.io


GitHub


GitHub Actions


Travis CI


Chef


Nginx+


Honeycomb


AWS CodeDeploy


CrowdStrike


DataDog

Rapid7


Qualys


Tenable

aws

AWS Graviton2 搭載インスタンスに関する FAQ

- Q. Savings Plans（リザーブドインスタンス）や、スポットインスタンスなどの購入オプションは利用可能ですか？
    - A. 可能です
- Q. Systems Manager によるインスタンス管理は可能ですか？
    - A. 可能です、Session Managerによるマネージメントコンソールからのログインも利用できます
- Q. Storage にはどのようなサービスが利用可能ですか？
    - x86系インスタンスと同様、EBSやEFSをご利用いただけます
- Q. Windows を利用可能ですか？
    - A. 対応しておりません

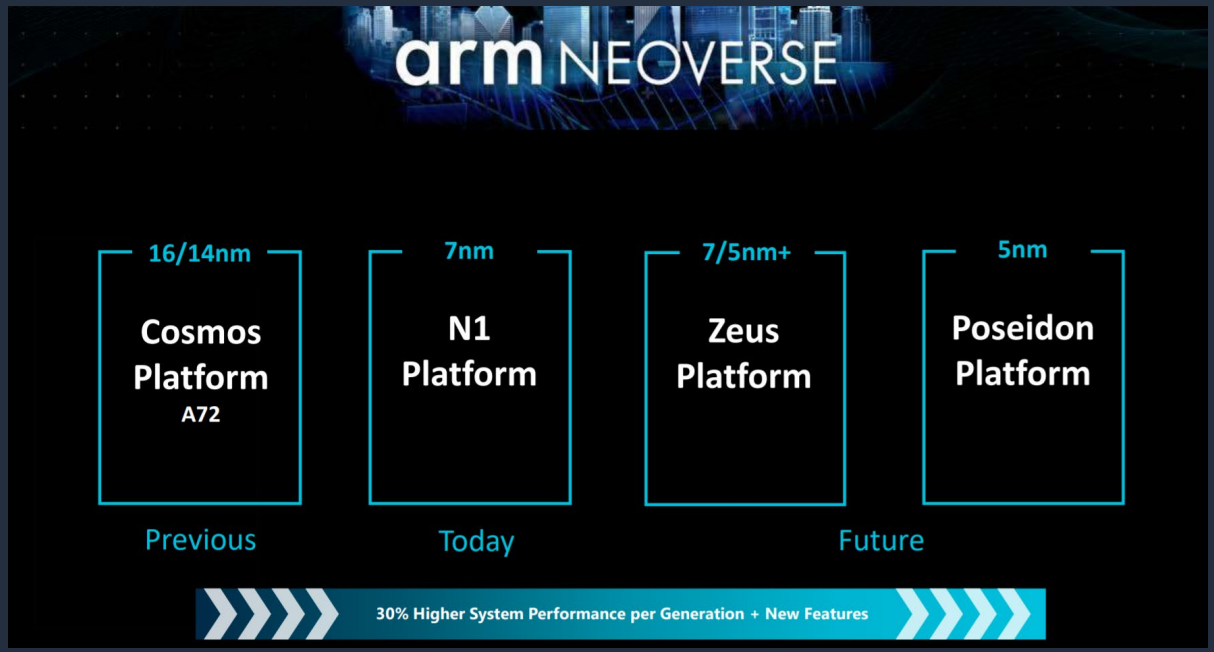
まとめ: AWS Graviton2

- AWS/Annapurna Labs が独自に設計を行った AWS Graviton2 プロセッサ搭載インスタンスが登場
- 同サイズの x86 系インスタンスと比較し、
  2 倍の物理コアが利用可能かつ、約 20 % 安価な価格設定により
  様々なワークロードで 約 40 % のコストパフォーマンス向上
- 利用にはアプリケーションの Arm 対応が必要となる点には注意が必要
- AWSだけでなく、3rd partyも含めた幅広い Arm エコシステムにより
  活用や移行をサポート

AWS Graviton2 の詳細アーキテクチャ

Arm Neoverse

Graviton2 ではコアアーキテクチャとして
Arm のサーバー向けブランドであるNeoverse N1 を採用



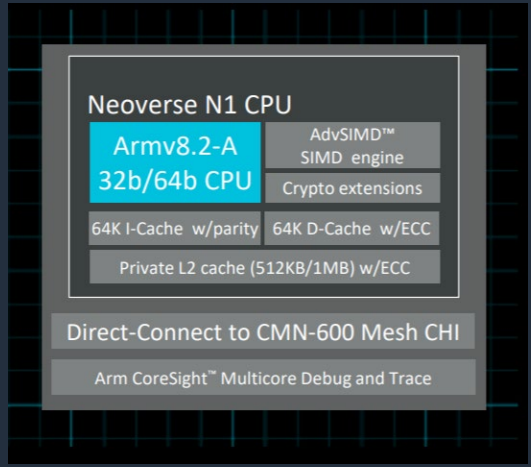
https://www.hotchips.org/hc31/HC31_1.2_20190816_Arm_Neoverse_N1_CPU.pdf

Graviton2 – CPUコア

Arm Neoverse N1 コア

Arm v8.2 準拠

- 各vCPU毎に64KB L1キャッシュ、1MB L2キャッシュを持つ
- コヒーレント命令キャッシュ
- 割り込み、仮想化、コンテキストスイッチにおいて少ないオーバーヘッド
- 4命令同時デコード、8命令同時発行
- ２つのSIMDユニット
- ML推論をアクセラレートする命令セット: int8, fp16対応



https://www.hotchips.org/hc31/HC31_1.2_20190816_Arm_Neoverse_N1_CPU.pdf

vCPUは物理コアと一対一で対応

- 同時マルチスレッディング(SMT)には非対応

aws

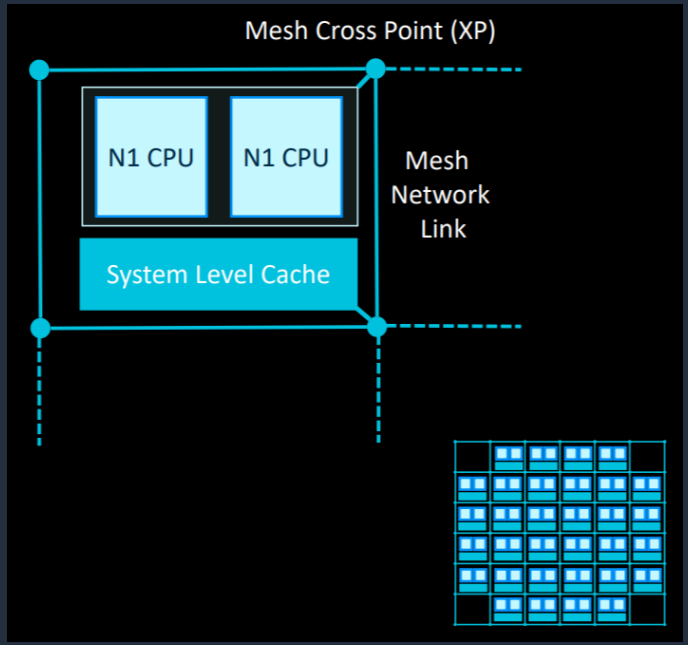
Graviton2 – インタコネクト

64コア間は~2TB/s メッシュ型インタコネクトで接続

32MB LLC
- プライベートキャッシュと合わせ約100MBのアクセス可能なキャッシュを搭載

NUMAの懸念の解消
- 全てのコアからメモリ及び他のコアへのパスが同様に見える内部構成

64レーン PCIe Gen4
- 異なるインスタンスの構成に対して柔軟に対応



メッシュインターコネクト例
https://www.hotchips.org/hc31/HC31_1.2_20190816_Arm_Neoverse_N1_CPU.pdf

Graviton2 – システム

8 x DDR-3200チャネル (> 200GB/s)

- インスタンス上のDRAMメモリの内容は常時AES-256で暗号化
- 全てのCPUコアから統一されたメモリレイテンシ

1Tbit/s伸張圧縮アクセラレータ搭載

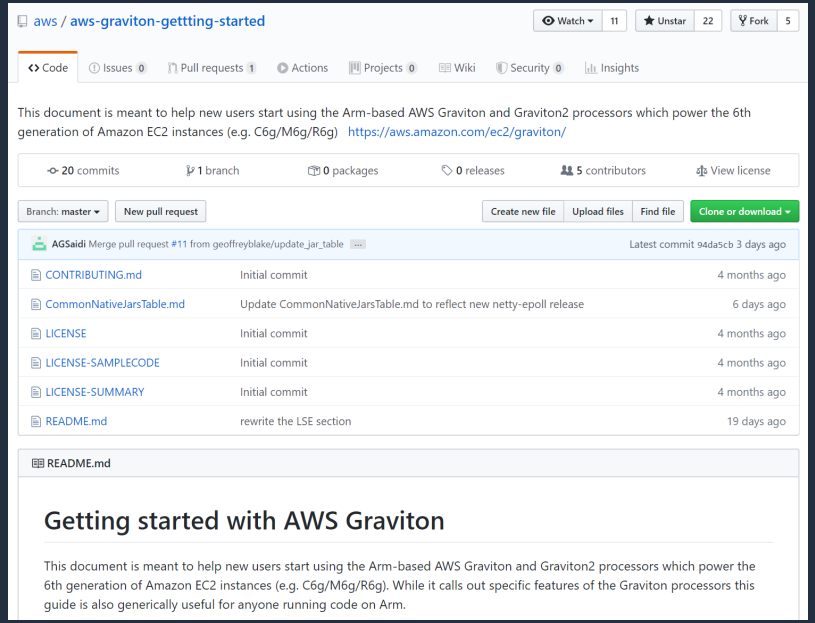
- 2xlarge以上のインスタンスでは圧縮デバイスを持つ
- DPDK、Linuxカーネルドライバは今後リリース予定
- 圧縮15GB/s, 伸張11GB/s

AWS Graviton2 の利用ガイド

AWS Graviton getting-started

Graviton/Graviton2利用時に最初にご確認いただきたいドキュメント

- 推奨コンパイルオプション
- SSE 命令の NEON 命令への変換
- アプリケーションでのパフォーマンス情報
- デバッグ・プロファイル
- Java Support
- etc..



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

AWS Graviton2 利用ガイド

AWS Graviton getting-started の内容を踏まえつつ、以下の内容についてご紹介

- Graviton 向けにコンパイルを 行う際の注意点
- Javaの利用
- コンテナの利用
- 機械学習アプリケーション利用時の注意点

C/C++ におけるコンパイル時の注意点

GCCのバージョン

- GCC 9以降を推奨

GCC における Graviton2 向け推奨コンパイルオプション:

- `-march=armv8.2-a+fp16+rcpc+dotprod+crypto`

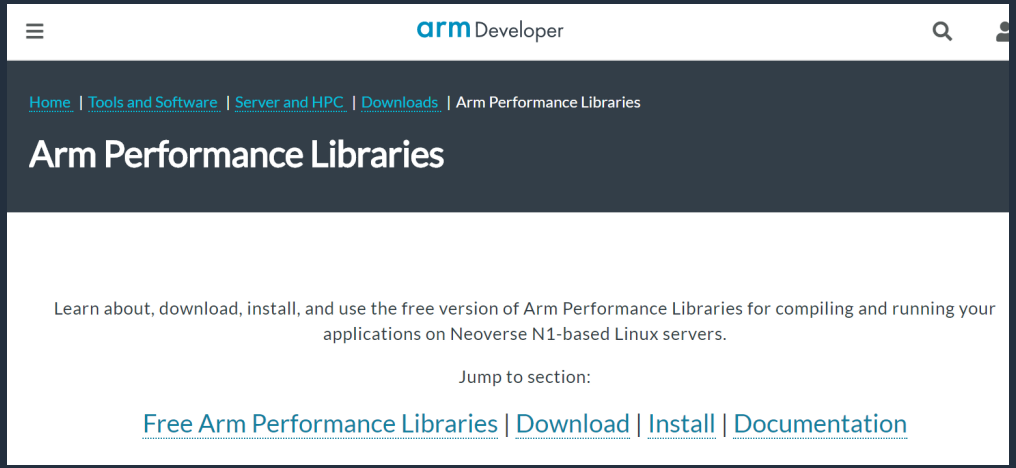
Large-System Extensions の活用:

- Graviton2 では、LSE (Large-System Extensions) をサポートしており、POSIX thread における高速なスレッド間同期などを提供
- 現在は Ubuntu 20.04 において LSE に対応した libc6-lse ライブラリが提供されており、上記コンパイルオプションを使用し適切にリンクを行うことで利用可能
- HPCアプリケーション等でのOpenMP利用時にも効果が大きい

Arm Performance Libraries の利用

Arm Neoverse N1 向けに最適化された数学系ライブラリを無償で提供

- BLAS, LAPACK, FFT, Sparse Linear Algebra, libmath
- RHEL, SUSE Linux Enterprise Server, Ubuntu をサポート
- GCC 7.1/8.2/9.3 に対応



https://developer.arm.com/tools-and-software/server-and-hpc/downloads/arm-performance-libraries

Java 利用時の注意点

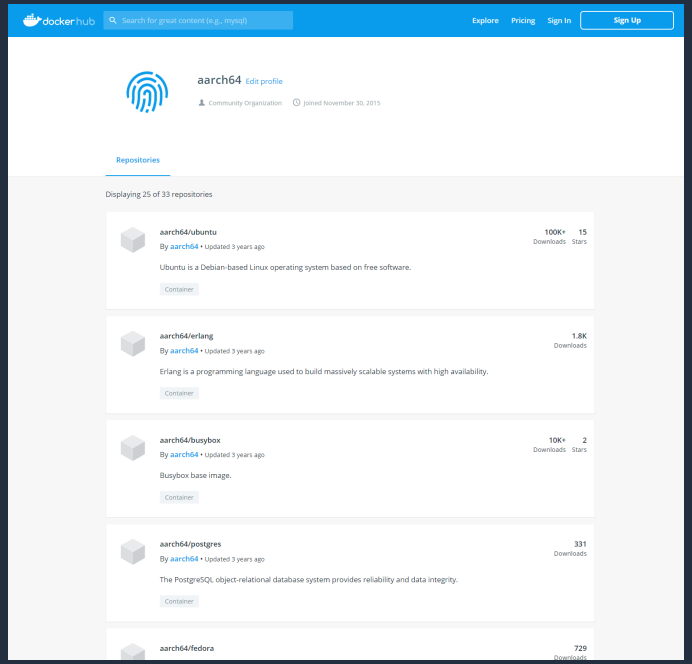
- 基本的には 同一の バイナリで動作するが、暗号化や圧縮のライブラリについては、高速化のためにネイテイブライブラリを使用している場合もある
  その際はマルチアーキテクチャ向けにビルドし直す必要がある
  例：https://github.com/aws/aws-graviton-gettting-started/blob/master/CommonNativeJarsTable.md

- JDKのバージョンやディストリビューションによっても性能に影響を与える
  OpenJDK 11 及び Corretto 11 でベンチマークを行うことを推奨

  参考：Arm Treasure Data 様による A1 での Presto ベンチマーク
  https://prestosql.io/blog/2019/12/23/Presto-Experiment-with-Graivton-Processor.html

Graviton2 でのコンテナの利用方法

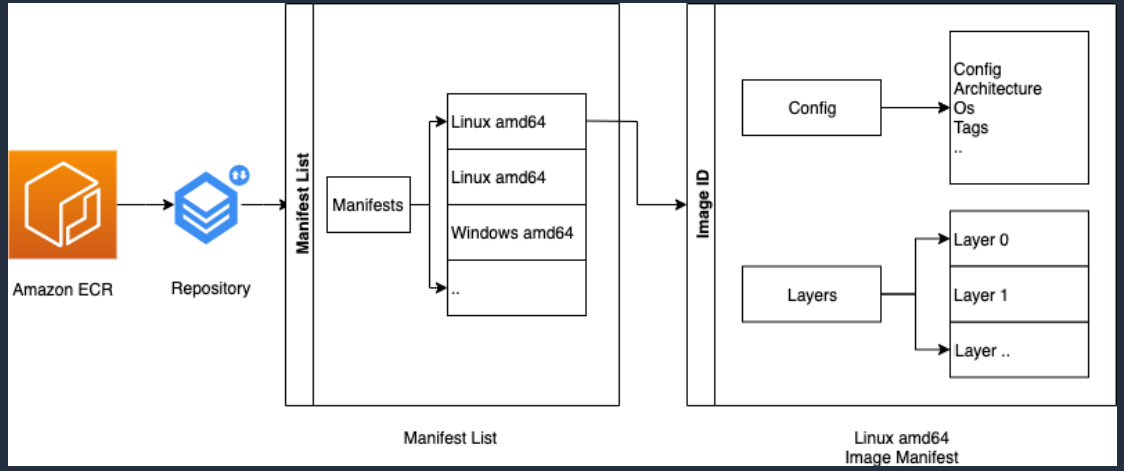
- Graviton2 上で Docker を利用することは可能（イメージの作成・実行等）
- ただし、Image内の実行ファイルは、arm64とamd64 (x86_64) で互換性が無く、アーキテクチャごとにImageを作成することとなる
- DockerHub では現在主要な Image については、aarch64 向け Image が公開済み
- Elastic なども Elasticsearch の aarch64 向け公式イメージを提供



https://hub.docker.com/u/aarch64/

マルチアーキテクチャ対応 Docker Image 作成方法

- Docker の Manifest 機能を使用することで、単一のレポジトリで、複数のアーキテクチャをホストする事が可能（ECR対応済み）
- Dockerの Experimental 機能である **buildx** サブコマンドにより、QEMU を使用して、amd64 環境でも aarch64 用イメージの作成が可能
- AWS CodeBuild では Arm タイプを選択可能であり、aarch64 ビルドが可能



https://aws.amazon.com/jp/blogs/containers/introducing-multi-architecture-container-images-for-amazon-ecr/

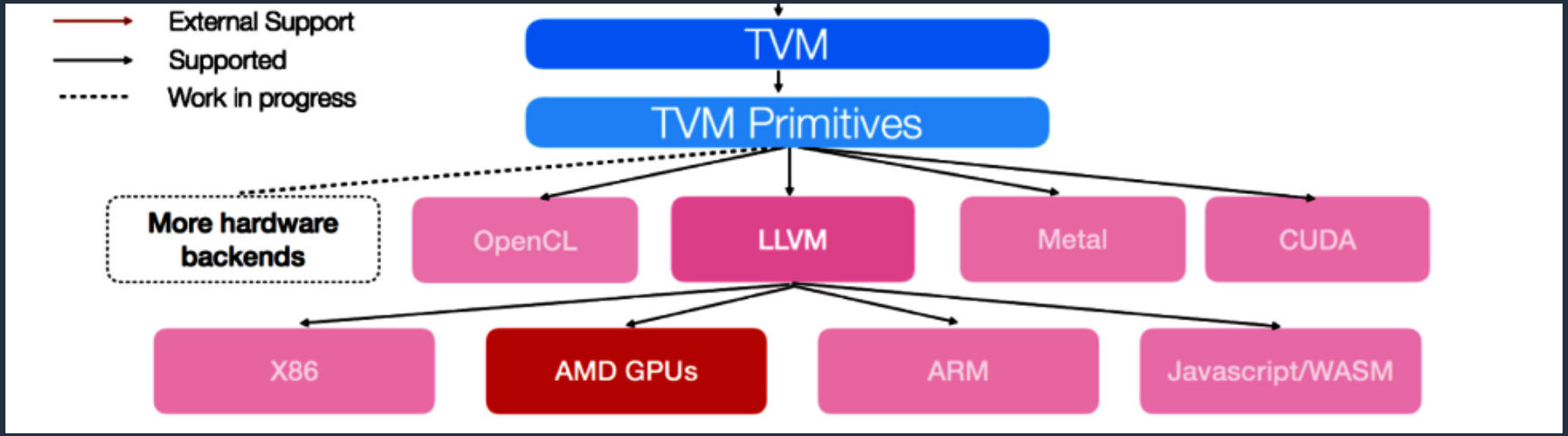
AWS のコンテナ系サービスにおける Graviton2 対応状況

- Amazon ECS: 対応済み
- Amazon EKS: 2020年7月7日現在パブリックプレビュー中
  https://docs.aws.amazon.com/eks/latest/userguide/arm-support.html
- Amazon ECR: マルチアーキテクチャレジストリに対応
- AWS Batch: 対応済み
- AWS CodeBuild: Arm イメージを選択可能

機械学習アプリケーション

機械学習ワークロードでは、TensorFlowなどの各種フレームワークが Arm に最適化されておらずパフォーマンスが得られない場合がある

→ オープンソースの Deep Learning コンパイラである Apache TVM を活用し、Arm に最適化されたコードを出力（NEON対応、FP16対応等）



https://tvm.apache.org/2017/10/06/nnvm-compiler-announcement

Graviton2 移行ガイド

現在利用しているアプリケーションによって、複数の移行パターンが存在

- **アプリケーションがyum/aptなどのパッケージマネージャから取得できる場合**
  - まずはパッケージマネージャからインストールを行い、評価を実施
  - 要件を満たすパフォーマンスが得られない場合は、ソースコードからのコンパイルを検討
- **PythonやRubyなどのスクリプト言語や Javaでアプリケーションが記述されている場合**
  - 基本的にはそのまま利用する事が可能だがパフォーマンステストは行う必要がある
  - 高速化のためにネイティブバイナリが使用されている場合もあり注意
- **C/C++ など、ソースコードからコンパイルを行う必要がある場合**
  - getting-started 推奨コンパイルオプションを使用 （特に16xlarge等の大きなインスタンス利用時にはLSEに注意）

AWS Graviton2 のベンチマーク

ワークロードとターゲットアプリケーションの広がり

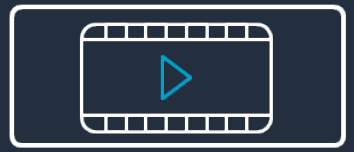
ウェブ・ゲーミングサーバー

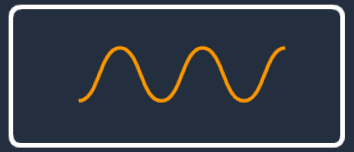
オープンソース
データベース

HPC

インメモリ キャッシュ

メディア エンコーディング





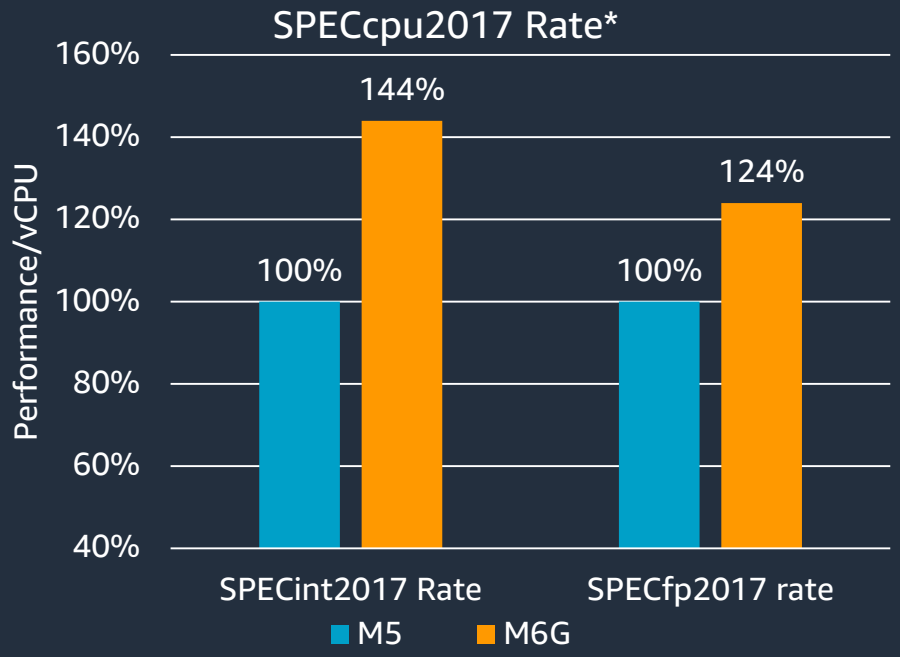
EDA

分析

マイクロサービス

SPECcpu2017

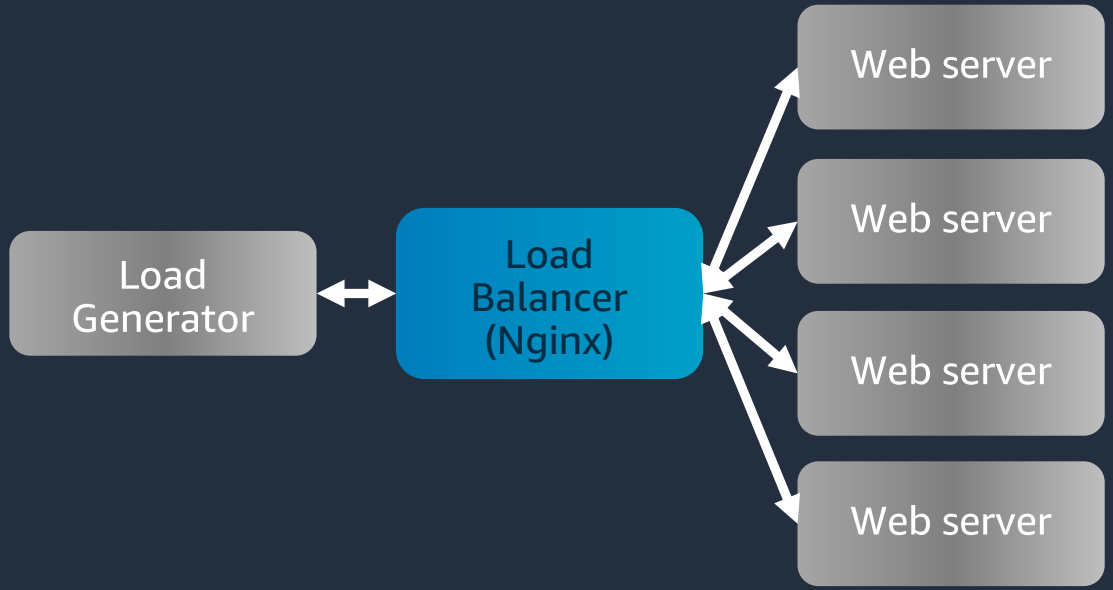
- 業界標準のCPUベンチマーク
- すべてのvCPU上で同時に実行
- M5とM6gでvCPUあたりの性能を比較

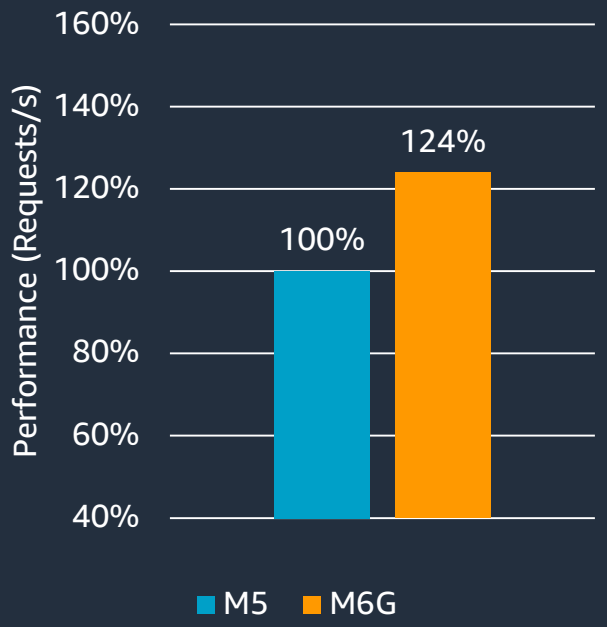


* All SPEC scores estimates, compiled with GCC9 -O3 -march=native, run on largest single socket size for each instance type tested.

aws

Nginx によるロードバランシング

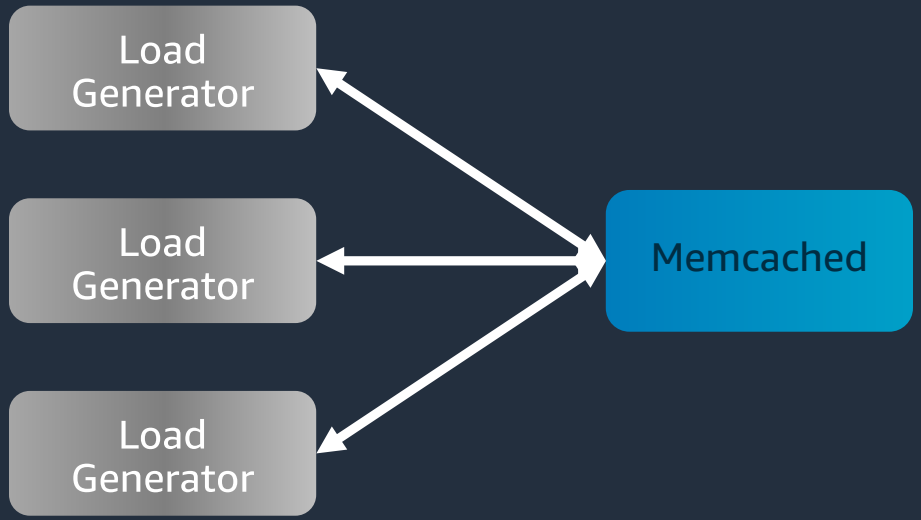


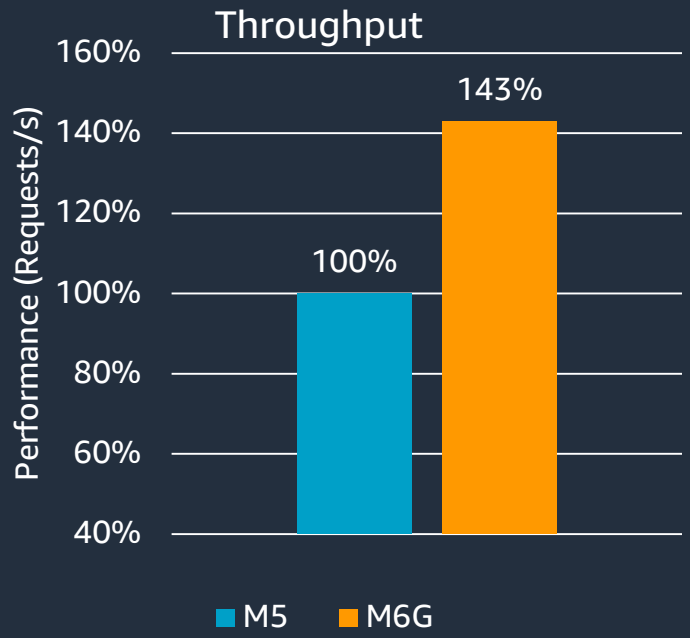


NGINX v1.15.9, 512 clients, 128 GET/POST payloads, all HTTPS connections, AES128-GCM-SHA256, OpenSSL 1.1.1, 4 target machines, all tests run on 4xl size; load generator c5.9xl; web servers c5.4xls; All servers run in a cluster placement group

aws

Memcached

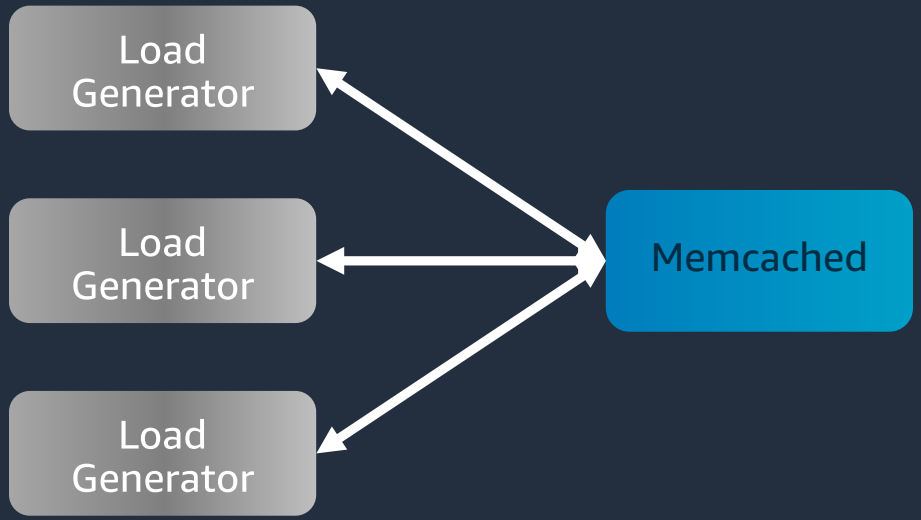


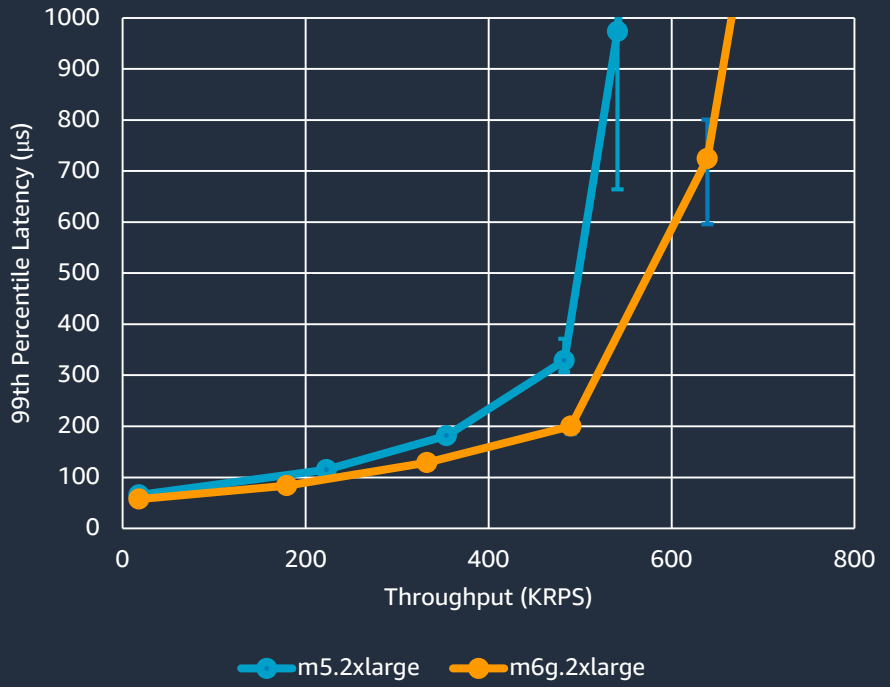


Memcached v1.5.16, 16B keys, 128B values, 7.8M KV-pairs, 576 connections for load generation from 2x c5.9xlarge instances, 16 additional connections used to measure latency from 1 additional c5.9xlarge,;each connection maintains 4096 outstanding requests; All servers in a cluster placement group

aws

Memcached



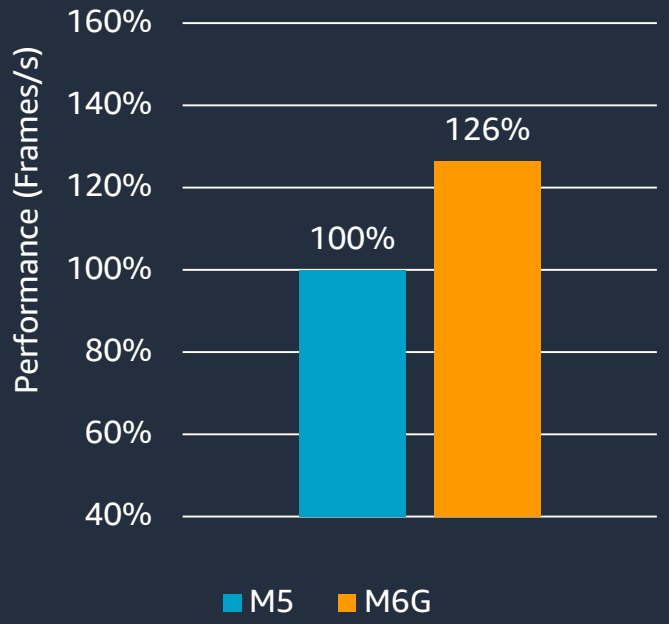


Memcached v1.5.16, 16B keys, 128B values, 7.8M KV-pairs, 576 connections for load generation from 2x c5.9xlarge instances, 16 additional connections used to measure latency from 1 additional c5.9xlarge, each connection maintains 4 outstanding requests; all servers in a cluster placement group

aws

Media Encoding with x264

- libx264エンコーダを利用し、1080pの非圧縮映像をH.264にエンコーディング
- M5、M6gいずれも 4xlarge を使用して速度を比較
- M6g が 約26 % 高い性能

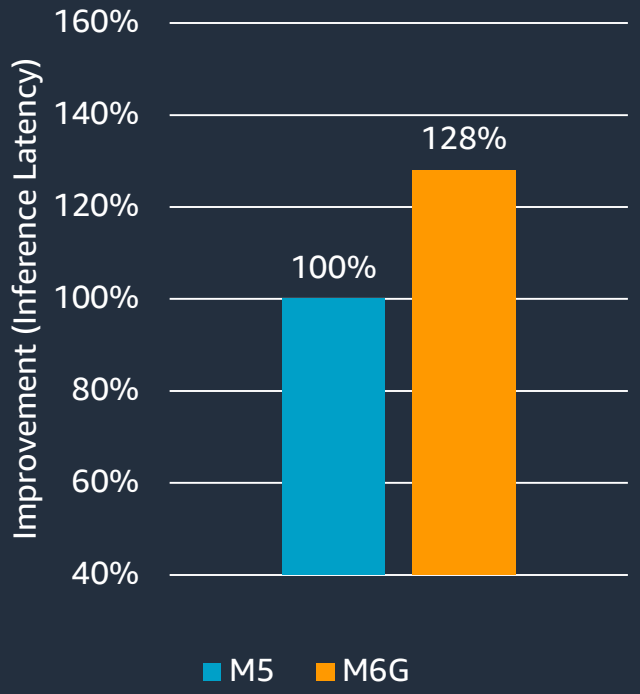


X264 (videolan.org version 3759fcb7), 4xl instance size, medium preset, input uncompressed 1080p50, output encoded h264 1080p50

aws

機械学習

- BERT: 自然言語処理モデル
- Graviton2は機械学習ワークロードをアクセラレートするための FP16 及び INT8をサポート
- モデルコンパイラであるTVM を使用
- M6gではM5を凌ぐ性能を達成

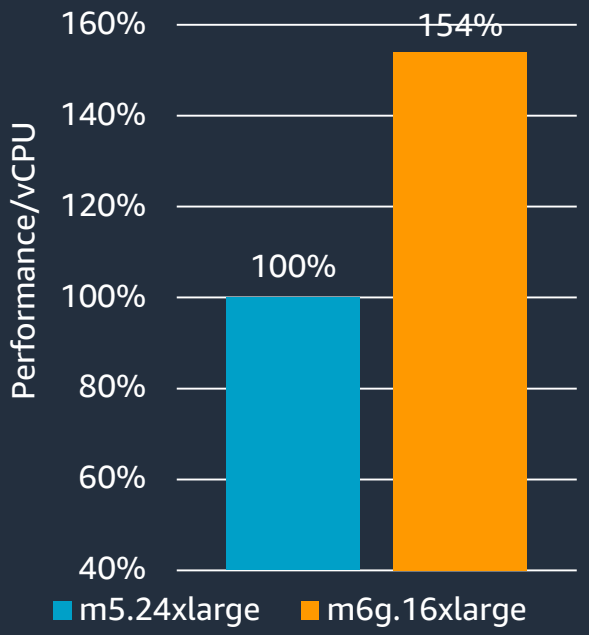


BERT classification using TVM and 64 length sequence on CPUs
Batch size of one; dedicated instances on xlarge size

aws

EDA Performance – Arm and Cadence Xcellium

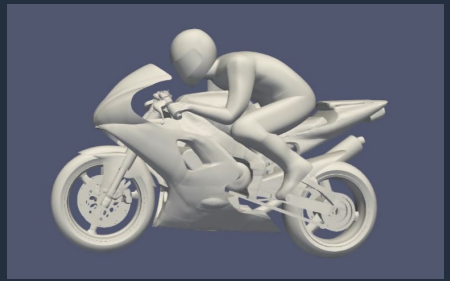
- プロセッサ設計時のシミュレーションに M6g を活用
- Arm社ではプロセッサ設計時のシミュレーションに膨大な時間を費やしていた
    - プロジェクトの時期に応じて変動性が高い
- ケイデンス社 Xcelliumを利用してCortex-A53をシミュレーション
    - プロセッサ上のDPU RTLに対して570の検証シミュレーションを実施

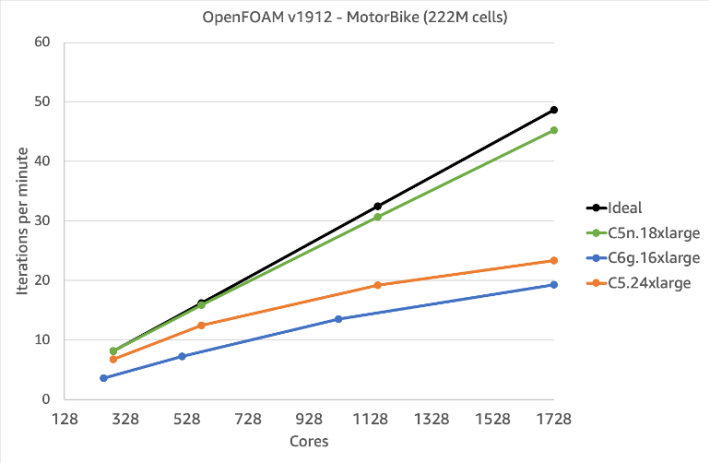


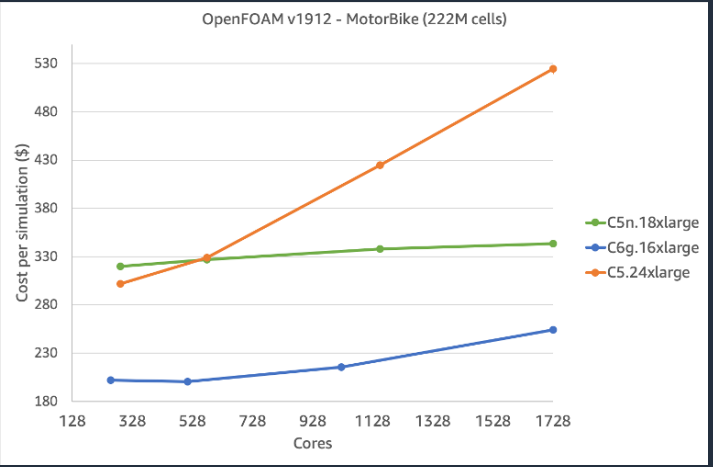
AWS Graviton2 の活用によりEDA領域でもvCPUあたり大幅な性能向上を実現

C6g による OpenFOAM Benchmarks

- 222M cell Motorbike モデルを用いて評価
- 計算性能としては、EFAを搭載したC5nが最も高い
- シミュレーションあたりのコストでは、C6g が最も安価







https://aws.amazon.com/blogs/compute/c6g-openfoam-better-price-performance/
https://gitlab.com/arm-hpc/packages/-/wikis/packages/openfoam

Arm 公式Blog での HPC 利用例

- Evaluation of the NEMO Ocean Model on Arm Neoverse-based AWS Graviton2
    - https://community.arm.com/developer/tools-software/hpc/b/hpc-blog/posts/evaluation-of-the-nemo-ocean-model-on-aws-graviton2
- Demonstration of low mach-number CFD modeling with Nalu on AWS Graviton2 M6g instances
    - https://community.arm.com/developer/tools-software/hpc/b/hpc-blog/posts/low-mach-number-cfd-modeling-with-nalu-on-graviton2-aws-m6g
- Seismic Modeling with Arm Neoverse N1 and AWS Graviton2
    - https://community.arm.com/developer/tools-software/hpc/b/hpc-blog/posts/seismic-modeling-with-arm-neoverse-n1-and-aws-graviton2

aws

AWS Graviton2 の活用事例

Arm Treasure Data: BigData 処理

Presto での TPC Benchmark H

- buildx を使用してマルチアーキテクチャのDocker Imageを作成
- OpenJDK11 を使用

“m6g.4xlarge is up to 30 percent faster than the current generation m5.4xlarge instance type. Considering m6g.4xlarge is also 20 percent lower cost than m5.4xlarge, **we can achieve up to 50 percent better ROI in total**. ”

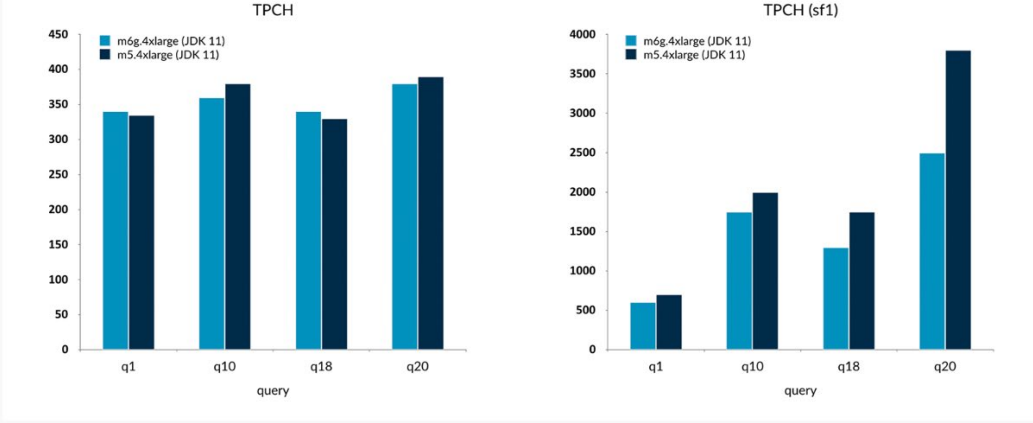


Figure 1

Figure 1 shows the notable results for m6g instances, because the benchmark we use is a CPU-intensive workload. The performance difference is significant in the larger data set (sf1), where we observe m6g.4xlarge is up to 30 percent faster than m5.4xlarge, with the same vCPU and memory configurations.

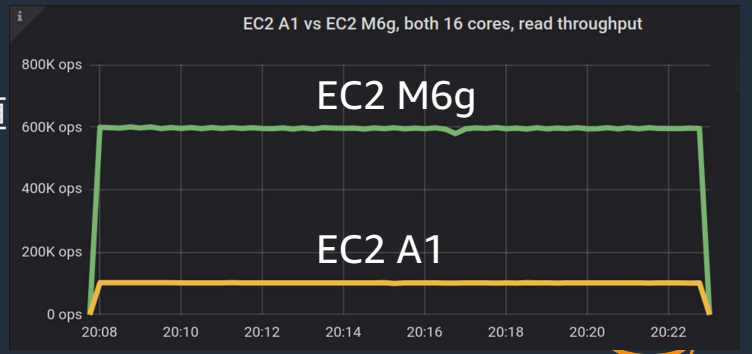
https://blog.treasuredata.com/blog/2020/03/27/high-performance-sql-aws-graviton2-benchmarks-with-presto-and-arm-treasure-data-cdp/

aws

Scylla

SCYLLA.

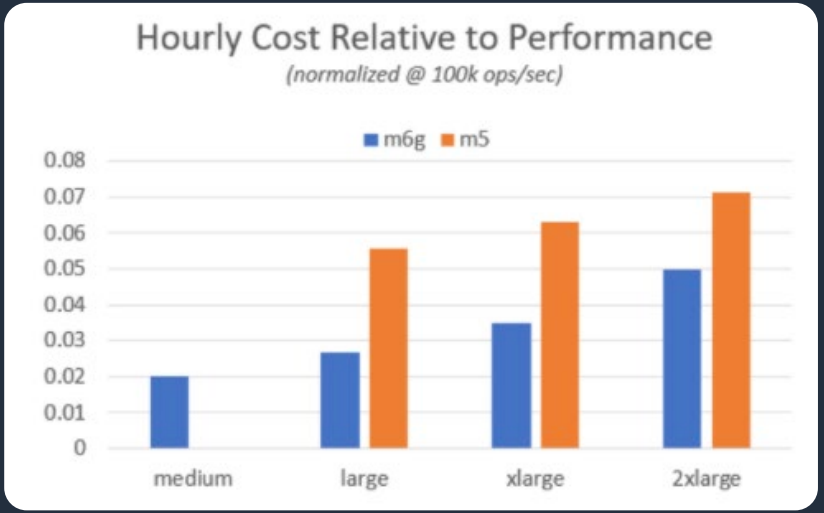
- Scyllaは高スループット、低遅延ビッグデータデータベース
  - CPUの高い利用効率を維持するThread-per-coreアーキテクチャ
  - I/Oスケジューラはピーク I/Oスループットを保証
  - AWS i3インスタンスの15GB/s バンド幅制限に到達
- M6gインスタンスで大幅な性能向上
  - 全てのCPU及びメモリを利用したワークロードをサポート
  - A1と比較し5倍の性能 (4xlarge → 37.5k reads/s/core)
  - 64vCPUを使用した場合の理論性能限界2.4M reads/s
  - Amazon EBSをストレージとして利用した場合の性能値
- M6gdインスタンスでさらなる性能向上
  - >5GB/s ディスク帯域, ～1M IOPS



KeyDB – M6g up to 65% faster than M5

KeyDB

- 高速なインメモリDBであるKeyDBでのベンチマーク
- M5と比較してM6gの利用により最大2倍のコストパフォーマンスが得られた



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

aws

M6g customer feedback

C5からM6gに移行することで、実行するインスタンスが30% 削減され、各インスタンスのコストが 10% 軽減された

Java11 + SpringBoot2 を使用したワークロードでM6gをテストし、M5と比較して最大43%優れたコストパフォーマンスが得られた

コンテナ化されたJavaベースのワークロードにおいて、M6gはM5と比較して40 %のパフォーマンス向上を確認できた

.NET Core ベースのワークロードにおいて、M6gは現在本番環境で使用している第5世代インスタンスよりも30 % 高いパフォーマンスが得られた

https://aws.amazon.com/ec2/graviton/

aws

AWS Graviton2 Roadmap

- AWS Graviton2 搭載インスタンスに対応予定のサービス
  - Amazon ElastiCache
  - Amazon EMR

- 内部的にAWS Graviton2 を利用予定
  - Elastic Load Balancing

その他にも様々なサービスが AWS Graviton2 に対応予定

まとめ

- Amazon EC2 では、ワークロードに合わせたインスタンスタイプの選択や適した購入オプションによりコスト効率よく処理を行う事が可能

- AWS 独自の AWS Graviton2 では、同サイズの M5 等と比較して 2 倍の物理コアが利用可能であり高いパフォーマンスが期待できる

- 既に多くのお客様で Graviton2 搭載インスタンスをご利用頂いており、様々なワークロードで約 40 % のコスト削減を達成

是非この機会に、AWS Graviton2 も含めて
インスタンスタイプの見直しをしてみませんか？

AWS の日本語資料の場所「AWS 資料」で検索

https://amzn.to/JPArchive

AWS Well-Architected 個別技術相談会

毎週”W-A個別技術相談会”を実施中

- AWSのソリューションアーキテクト(SA)に対策などを相談することも可能
- 申込みはイベント告知サイトから

(https://aws.amazon.com/jp/about-aws/events/)

AWS イベント で[検索]

ご視聴ありがとうございました

AWS 公式 Webinar
https://amzn.to/JPWebinar

過去資料
https://amzn.to/JPArchive

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 Regions – US East (N.Virginia and Ohio), US West (Oregon), Europe (Ireland & Frankfurt), and Asia Pacific (Tokyo)

Amazon EC2 C6g: sizes and specifications

Instance	vCPUs	Memory (GB)	Network Bandwidth (Gbps)	EBS Optimized	EBS Bandwidth (Mbps)	EBS Optimized Burst Bandwidth (Mbps)
c6g.medium	1	2	Up to 10	Yes	315	4,750
c6g.large	2	4	Up to 10	Yes	630	4,750
c6g.xlarge	4	8	Up to 10	Yes	1,188	4,750
c6g.2xlarge	8	16	Up to 10	Yes	2,375	4,750
c6g.4xlarge	16	32	Up to 10	Yes	4,750	4,750
c6g.8xlarge	32	64	12Gbps	Yes	9,000	9,000
c6g.12xlarge	48	96	20Gbps	Yes	13,500	13,500
c6g.16xlarge	64	128	25Gbps	Yes	19,000	19,000
c6g.metal	64	128	25Gbps	Yes	19,000	19,000

AWS Regions – US East (N.Virginia and Ohio), US West (Oregon), Europe (Ireland & Frankfurt), and Asia Pacific (Tokyo)

Amazon EC2 R6g: sizes and specifications

Instance	vCPUs	Memory (GB)	Network Bandwidth (Gbps)	EBS Optimized	EBS Bandwidth (Mbps)	EBS Optimized Burst Bandwidth (Mbps)
r6g.medium	1	8	Up to 10	Yes	315	4,750
r6g.large	2	16	Up to 10	Yes	630	4,750
r6g.xlarge	4	32	Up to 10	Yes	1,188	4,750
r6g.2xlarge	8	64	Up to 10	Yes	2,375	4,750
r6g.4xlarge	16	128	Up to 10	Yes	4,750	4,750
r6g.8xlarge	32	256	12Gbps	Yes	9,000	9,000
r6g.12xlarge	48	384	20Gbps	Yes	13,500	13,500
r6g.16xlarge	64	512	25Gbps	Yes	19,000	19,000
r6g.metal	64	512	25Gbps	Yes	19,000	19,000

AWS Regions – US East (N.Virginia and Ohio), US West (Oregon), Europe (Ireland & Frankfurt), and Asia Pacific (Tokyo)