The Value Proposition of Migrating AWS Workloads to ARM-Based Instances

Executive Summary

This report examines the value proposition of migrating Amazon Web Services (AWS) workloads from the traditional x86 architecture (utilized by AMD and Intel processors) to the ARM architecture, specifically focusing on AWS Graviton processors. The analysis highlights the significant benefits associated with this migration, primarily in terms of cost optimization and performance enhancement, while also addressing the potential challenges related to compatibility. The evidence strongly suggests that for a wide range of workloads, migrating to Graviton instances can lead to substantial reductions in operational expenses and improved performance metrics. Furthermore, the report underscores the growing need for automated assessment tools, such as a chatbot, to facilitate this migration process by identifying suitable workloads and potential roadblocks.

Introduction

In the dynamic landscape of cloud computing, organizations are continuously seeking strategies to optimize their infrastructure for both cost efficiency and performance. AWS Graviton processors, custom-designed by Amazon Web Services utilizing the ARM architecture, have emerged as a compelling alternative to traditional x86-based instances in the Amazon Elastic Compute Cloud (EC2) environment.[1] This report delves into the advantages and disadvantages of migrating AWS workloads from x86 to ARM, providing a comprehensive overview of the potential benefits and challenges involved in this transition. The increasing complexity of cloud environments and the diverse range of available instance types necessitate the adoption of intelligent tools to guide users in making informed decisions about their infrastructure choices.[1]

Cost Optimization

Migrating to AWS Graviton instances presents a significant opportunity for cost optimization across various workload types. AWS Graviton-based instances are designed to deliver superior price performance, often costing up to 20% less than comparable x86-based Amazon EC2 instances.[2] This cost advantage stems from the energy-efficient ARM architecture and AWS's focus on optimizing its custom-designed processors for cloud workloads.[4]

Numerous customer testimonials and case studies corroborate these cost savings. Zendesk, for example, reported a 42% reduction in monthly costs after migrating to

Graviton instances.[5] Instructure experienced a 20% improvement in cost savings alongside increased throughput by adopting AWS Graviton3 instances.[2] Zomato achieved a 25% improvement in price performance by moving their data platform to Graviton-based instances.[2] These real-world examples highlight the tangible financial benefits that organizations can realize by transitioning to Graviton.[6]

The cost efficiency of Graviton extends beyond the instance price itself. Graviton-based instances consume up to 60% less energy than comparable x86 instances for the same performance, contributing to lower operational expenses and aligning with sustainability goals.[2] This reduced power consumption not only lowers energy bills but also supports corporate environmental initiatives.[1]

AWS provides tools like the AWS Graviton Savings Dashboard to help users visualize and understand the potential cost savings of migrating to Graviton. This dashboard analyzes current and future Graviton usage, enabling organizations to evaluate the financial impact of this transition.[2] Furthermore, AWS offers a free trial for Graviton2-based T4g.small instances, allowing businesses to explore the performance and compatibility of Graviton with minimal financial risk.[8]

Performance Enhancement

Beyond cost savings, AWS Graviton instances often deliver equal or better performance than their x86 counterparts for a variety of workloads.[6] The ARM architecture, with its focus on efficiency and optimized instruction sets, allows Graviton processors to excel in compute-intensive tasks such as data analytics, machine learning, and video transcoding.[1]

AWS Graviton2 processors offer up to 40% better price performance than comparable 5th generation Intel or AMD processors, with significant improvements in compute cores, memory speed, and cache size.[4] The subsequent Graviton3 processors further enhance performance, offering up to 25% better compute performance and twice the floating-point performance compared to Graviton2.[13] The latest generation, Graviton4, continues this trend with even greater performance gains for various workloads.[14]

Benchmark comparisons between comparable x86 and Graviton instance families consistently demonstrate the performance advantages of Graviton. KeyDB, an in-memory database, showed up to 1.65X faster performance on smaller M6g instances (Graviton2) compared to M5 instances (x86).[15] C6g instances (Graviton2) deliver up to 40% better price performance over C5 instances (x86) for compute-intensive workloads.[16] Memory-optimized R6g instances (Graviton2) also offer superior price performance compared to R5 instances (x86) for

memory-intensive applications.[17]

Customer testimonials further validate these performance gains. Zendesk reported up to a 30% performance improvement after migrating to Graviton instances.[5] Informatica achieved a 25% reduction in search latency by using AWS Graviton3-based instances.[18] These examples illustrate the potential for significant performance enhancements across different application domains.[2]

Compatibility Considerations

While the benefits of migrating to AWS Graviton are compelling, organizations must also consider potential compatibility challenges. Applications originally developed for the x86 architecture may encounter issues when running on the ARM-based Graviton instances.[19] This is primarily due to differences in instruction sets, which may require recompilation of code and updates to application dependencies.[4]

One common challenge is ensuring application compatibility with the ARM architecture. Many applications rely on specific processor architectures, which can lead to performance degradation or failure on ARM.[19] Updating application dependencies, including libraries, tools, and frameworks, to ARM-compatible versions is crucial.[19] Performance testing is also essential to identify any bottlenecks or issues that need to be addressed in the ARM environment.[19]

Limited support for third-party tools and software can also pose a challenge. Not all third-party software, especially proprietary or legacy applications, may yet support the ARM architecture.[19] Early collaboration with software vendors to determine ARM compatibility or explore ARM-friendly alternatives is recommended.[19]

Migrating containerized applications requires careful consideration as well. Organizations using Docker need to create multi-architecture Docker images that can run on both Intel/AMD and ARM platforms.[19] Tools like Docker's buildx can assist in building these multi-architecture images.[19] Thorough testing across both architectures is crucial to ensure seamless migration without disruptions.[19]

AWS provides several tools and resources to assist with Graviton migration assessment and porting. The Porting Advisor for Graviton is an open-source command-line tool that analyzes source code and highlights potential compatibility issues, providing recommendations for alternatives.[24] The AWS Graviton Ready Program identifies partners with software products that support AWS Graviton-based instances, making it easier for customers to find compatible solutions.[26] For.NET applications, the AWS Porting Assistant for.NET helps assess and port.NET Framework

applications to.NET Core on Linux, which is a prerequisite for running on Graviton.[28]

Cost Comparison in Seoul Region

To illustrate the potential cost savings in a specific AWS region, this section compares the On-Demand hourly costs, vCPU count, and Memory (GiB) of selected x86 instance types with their commonly recommended Graviton equivalents in the Seoul region (ap-northeast-2). The pricing information is based on publicly available data and may vary.

Instance Type (x86)	vCPUs	Memory (GiB)	On-Demand Hourly Cost (USD)	Equivalent Instance Type (Graviton)	vCPUs	Memory (GiB)	On-Demand Hourly Cost (USD)
t3.large	2	8	0.0832	t4g.large	2	8	0.0832
m5.xlarge	4	16	0.236	m6g.xlarge	4	16	0.188
c5.xlarge	4	8	0.192	c6g.xlarge	4	8	0.154
r5.xlarge	4	32	0.304	r6g.xlarge	4	32	0.244

As the table indicates, migrating from x86 to Graviton in the Seoul region can result in direct cost reductions for comparable instance sizes. For instance, the m6g.xlarge instance offers a lower hourly cost than the m5.xlarge, while providing the same number of vCPUs and memory. Similar cost benefits can be observed for other instance families as well.

Savings Plans and Reserved Instances

AWS Savings Plans and Reserved Instances offer additional cost-saving opportunities for both x86 and Graviton instance families. Savings Plans provide discounts of up to 66% on Compute Savings Plans and up to 72% on EC2 Instance Savings Plans in exchange for a commitment to a consistent amount of usage over a one or three-year

term.[30] Compute Savings Plans offer flexibility across instance families, sizes, AZs, regions, OS, and tenancy, while EC2 Instance Savings Plans provide the lowest prices for committing to specific instance families within a region.[30]

Reserved Instances (RIs) also offer significant discounts (up to 72%) compared to On-Demand pricing for a one or three-year term, with options for upfront, partial upfront, or no upfront payments.[32] While RIs require a commitment to a specific instance type, they can provide capacity reservations and are suitable for steady-state workloads.[32]

Notably, Savings Plans can be applied to both x86 and Graviton instance usage, allowing organizations to benefit from discounted rates regardless of the underlying architecture.[34] The percentage discounts offered by Savings Plans and Reserved Instances are generally similar for comparable commitment terms and payment options across both x86 and Graviton instance families.[35] Therefore, the choice between x86 and Graviton should primarily be driven by the On-Demand cost differences and the performance characteristics of the workloads on each architecture, with Savings Plans and RIs providing additional layers of cost optimization for committed usage.[35]

Conclusion

The evidence presented in this report strongly supports the value proposition of migrating AWS workloads from x86 to ARM-based Graviton instances. The potential for significant cost savings, coupled with performance enhancements for a wide range of workloads, makes Graviton an attractive option for organizations looking to optimize their cloud infrastructure. While compatibility challenges exist, AWS provides a growing ecosystem of tools, resources, and partner support to facilitate a smooth transition. The need for automated assessment tools like a chatbot is evident in simplifying the migration process by guiding users through compatibility checks, cost-benefit analysis, and performance considerations, ultimately accelerating the adoption of this transformative technology.

참고 자료

1. The Cost-Performance Advantages of Graviton | Matilda Cloud, 4월 14, 2025에 액세스, https://www.matildacloud.com/blog/the-cost-performance-advantage-of-graviton-instances
2. AWS Graviton Processor - Amazon EC2, 4월 14, 2025에 액세스, https://aws.amazon.com/ec2/graviton/
3. Optimizing Cost with AWS Graviton Instances - CloudThat, 4월 14, 2025에 액세스,

https://www.cloudthat.com/resources/blog/optimizing-cost-with-aws-graviton-instances
4. What Is AWS Graviton? Here's When To Use It - CloudZero, 4월 14, 2025에 액세스, https://www.cloudzero.com/blog/aws-graviton/
5. Achieving 30% Performance Improvement and Reduced Carbon Emissions by Migrating to AWS Graviton Instances | Zendesk Case Study, 4월 14, 2025에 액세스, https://aws.amazon.com/solutions/case-studies/zendesk-graviton-case-study/
6. Optimize AWS Costs with Graviton-Based Services - Several Clouds, 4월 14, 2025에 액세스, https://www.severalclouds.com/post/optimize-aws-costs-with-graviton-based-services
7. Migrating to AWS Graviton: Unlocking Cost Savings, Performance, and Efficiency, 4월 14, 2025에 액세스, https://www.tothenew.com/blog/migrating-to-aws-graviton-unlocking-cost-savings-performance-and-efficiency/
8. AWS Graviton Processors Explained: Cut Costs by 40% with Better Efficiency and Security, 4월 14, 2025에 액세스, https://dev.to/devops_descent/aws-graviton-processors-explained-cut-costs-by-40-with-better-efficiency-and-security-3dcn
9. Sustainability with AWS Graviton, 4월 14, 2025에 액세스, https://aws.amazon.com/ec2/graviton/graviton-sustainability/
10. Reducing Your Carbon Footprint with AWS Graviton and Cloud Computing - CloudSphere, 4월 14, 2025에 액세스, https://cloudsphere.com/reducing-your-carbon-footprint-with-aws-graviton-and-cloud-computing/
11. AWS Graviton Savings Dashboard Step-by-Step Guide - Amazon.com, 4월 14, 2025에 액세스, https://aws.amazon.com/ec2/graviton/graviton-savings-dashboard/
12. What is AWS Graviton? - nOps, 4월 14, 2025에 액세스, https://www.nops.io/glossary/what-is-aws-graviton/
13. Use Graviton instances and containers - AWS Prescriptive Guidance, 4월 14, 2025에 액세스, https://docs.aws.amazon.com/prescriptive-guidance/latest/optimize-costs-microsoft-workloads/net-graviton.html
14. AWS Graviton4-based Amazon EC2 R8g instances: best price performance in Amazon EC2, 4월 14, 2025에 액세스, https://aws.amazon.com/blogs/aws/aws-graviton4-based-amazon-ec2-r8g-instances-best-price-performance-in-amazon-ec2/
15. Benchmarking the AWS Graviton2 with KeyDB – M6g up to 65% faster, 4월 14, 2025에 액세스, https://docs.keydb.dev/blog/2020/03/02/blog-post/
16. Amazon EC2 C6g Instances - Amazon Web Services, 4월 14, 2025에 액세스, https://aws.amazon.com/ec2/instance-types/c6g/
17. Amazon EC2 R6g Instances - Amazon Web Services, 4월 14, 2025에 액세스, https://aws.amazon.com/ec2/instance-types/r6g/

18. Accelerating the Ingestion of 33 TB/Day by 25% Using AWS Graviton Processor | Informatica Case Study, 4월 14, 2025에 액세스, https://aws.amazon.com/solutions/case-studies/informatica-graviton-case-study/
19. A Deep Dive into AWS Graviton Instances - CloudOptimo, 4월 14, 2025에 액세스, https://www.cloudoptimo.com/blog/a-deep-dive-into-aws-graviton-instances/
20. Transitioning your service or application - AWS Graviton2 for Independent Software Vendors, 4월 14, 2025에 액세스, https://docs.aws.amazon.com/whitepapers/latest/aws-graviton2-for-isv/transitioning-your-service-or-application.html
21. Other considerations - AWS Graviton Performance Testing: Tips for Independent Software Vendors, 4월 14, 2025에 액세스, https://docs.aws.amazon.com/whitepapers/latest/aws-graviton-performance-testing/other-considerations.html
22. Migrating from x86 to AWS Graviton on Amazon EKS using Karpenter | Containers, 4월 14, 2025에 액세스, https://aws.amazon.com/blogs/containers/migrating-from-x86-to-aws-graviton-on-amazon-eks-using-karpenter/
23. Migration to Graviton for container based workload - Sigma Technology, 4월 14, 2025에 액세스, https://sigmatechnology.com/case/migration-to-graviton-for-container-based-workload/
24. What's your experience with AWS Graviton processors? - Reddit, 4월 14, 2025에 액세스, https://www.reddit.com/r/aws/comments/1hiikoa/whats_your_experience_with_aws_graviton_processors/
25. Using Porting Advisor for Graviton | AWS Compute Blog - Amazon.com, 4월 14, 2025에 액세스, https://aws.amazon.com/blogs/compute/using-porting-advisor-for-graviton/
26. AWS Graviton Partners, 4월 14, 2025에 액세스, https://aws.amazon.com/ec2/graviton/partners/
27. Introducing the AWS Graviton Ready Program, 4월 14, 2025에 액세스, https://aws.amazon.com/about-aws/whats-new/2021/11/aws-graviton-ready-program/
28. What is Porting Assistant for .NET? - AWS Documentation, 4월 14, 2025에 액세스, https://docs.aws.amazon.com/portingassistant/latest/userguide/what-is-porting-assistant.html
29. Porting Assistant for .NET - Amazon Web Services, 4월 14, 2025에 액세스, https://aws.amazon.com/porting-assistant-dotnet/
30. Compute and EC2 Instance Savings Plans - AWS, 4월 14, 2025에 액세스, https://aws.amazon.com/savingsplans/compute-pricing/
31. What are Savings Plans? - Savings Plans - AWS Documentation, 4월 14, 2025에 액세스, https://docs.aws.amazon.com/savingsplans/latest/userguide/what-is-savings-plans.html
32. EC2 Reserved Instance Pricing – Amazon Web Services, 4월 14, 2025에 액세스,

https://aws.amazon.com/ec2/pricing/reserved-instances/pricing/
33. Amazon EC2 Reserved Instances - AWS, 4월 14, 2025에 액세스, https://aws.amazon.com/ec2/pricing/reserved-instances/
34. Reviewing your cost structure - AWS Graviton2 for Independent Software Vendors, 4월 14, 2025에 액세스, https://docs.aws.amazon.com/whitepapers/latest/aws-graviton2-for-isv/reviewing-your-cost-structure.html
35. Graviton processors and cost savings : r/aws - Reddit, 4월 14, 2025에 액세스, https://www.reddit.com/r/aws/comments/1fjr8yw/graviton_processors_and_cost_savings/