HIGH PERFORMING DATA PROCESSING ON CLOUD WITH BARE METAL INSTANCES

Sharada Lakshmanan

Databricks Consulting (Cloud, Data Engineering)

ABSTRACT

High performance data processing of both batch and streaming pipelines in cloud virtual instances poses challenges on performance and cost compared to the on prem bare metal instances.The cloud services can be expensive on infrastructure storage and maintenance. The long term cost effectiveness of cloud storage in comparison with bare metal instances has to be done in detail to ensure the balance of cost and data processing capabilities. The kubernetes architecture on cloud ensures resiliency but the high performance of on prem bare metal in comparison with cloud virtual instances with kubernetes poses challenge in data processing with a high operational cost.

KEYWORDS

High Performance processing in Cloud, Bare metal instances on cloud, Kubernetes with resiliency and cost effective features

1. INTRODUCTION

Baremetal instance On-Prem on SPARK-

- Provides No Virtualization Overhead with better CPU,memory and performance.
- Ideal for Large Scale ETL and ML(Machine Learning) workloads,streaming workloads
- Custom tuning of OS,JVM and Spark configs are covered.

The deployment options are

- Canonical MAAS(Metal as Service) - Automates provisioning of physical servers, integrates with Kubernetes or standalone Spark clusters
- Using Hadoop/Spark Standalone Cluster to install Spark directly on bare metal nodes, manage with Ansible or similar tools.
- Kubernetes on bare metal uses K8s SPARK operator for job submission and scaling.

AWS EC2 Bare Metal Instances

- Amazon EC2 Bare Metal instances provide direct access to the physical hardware of the underlying server, bypassing the virtualization layer. These instances are ideal for workloads requiring low-level hardware access, such as specialized applications, legacy systems, or licensing-restricted software.

Key Features

- Direct Hardware Access: Bare metal instances expose the physical CPU, memory, and storage resources directly to the operating system, enabling deep performance analysis and hardware-level optimizations.

- Custom Processors: Instances like C7i, M7i, and R7i are powered by 4th generation Intel Xeon Scalable processors, offering up to 15% better performance compared to other x86-based processors.
- Specialized Accelerators: Support for Intel technologies such as Data Streaming Accelerator (DSA), In-Memory Analytics Accelerator (IAA), and QuickAssist Technology (QAT) for enhanced performance in analytics and cryptographic workloads.
- High Scalability: Available in configurations like metal-24xl (96 vCPUs) and metal-48xl (192 vCPUs), with memory scaling up to hundreds of GiB.

2. BARE METAL INSTANCES IN AWS

AWS Bare Metal Instances provide direct access to the physical hardware of the underlying server without the virtualization layer. These instances are ideal for workloads requiring low-level hardware access, performance-intensive applications, and legacy systems.

Key Features includes-

Bare metal instances offer full access to the physical CPU, memory, and other hardware resources. They are different from virtualized instances and they do not run under a hypervisor, allowing users to install their own hypervisors or operating systems. AWS uses the term vCPU in its APIs and console for consistency, but these instances provide access to physical CPUs.

They are powered by advanced processors, such as the 4th generation Intel Xeon Scalable processors, which deliver up to 15% better performance compared to other x86-based processors. Additionally, they support specialized hardware accelerators like Intel Data Streaming Accelerator (DSA).

AWS Instance Types

- AWS offers various bare metal instance types tailored for different workloads:
- C7i, M7i, and R7i: These instances provide up to 192 vCPUs and are available in regions like US East (Ohio, N. Virginia), US West (Oregon), and Europe (Ireland, Spain, Stockholm).
- R7iz: Designed for high-frequency workloads, these instances offer up to 128 vCPUs with an all-core turbo frequency of 3.9 GHz.
- GPU-enabled Bare Metal Instances: AWS provides GPU-enabled options like G4dn.metal and G5g.metal for workloads requiring GPU acceleration.

Use Cases

Bare metal instances are suitable for:

- Applications requiring direct hardware access for performance analysis.
- Legacy workloads incompatible with virtualized environments.
- Licensing-restricted software that mandates physical hardware.
- Running custom hypervisors or containerized environments.

2.1. Bare Metal Instances In AWS- Why Do We Need?

The demand of high-performance workloads processing for applications such as High performance legacy batch processing running in On Prem Bare metal, SAP,SAP HANA and GPU-heavy AI applications has been steadily increasing. These workloads often require direct access to the underlying hardware to achieve optimal performance, which is where AWS EC2 Bare Metal Instances come into play. AWS EC2 Bare Metal Instances provide the benefits of bare metal performance while still offering the flexibility and scalability of the cloud.

Bare Metal Instances are ideal for workloads that require:

Direct access to hardware features, such as Intel VT-x or AMD-V.

- High-performance computing (HPC) workloads.
- Applications that require low-latency and high-throughput networking.
- Workloads that need to run in a non-virtualized environment, such as SAP HANA or GPU-heavy AI applications, legacy high performing batch applications.

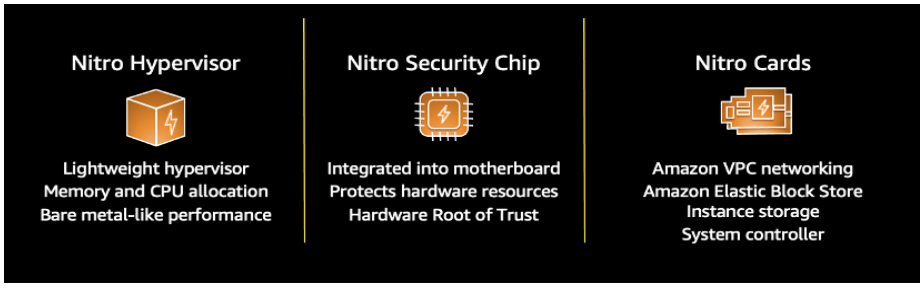
Bare Metal EC2 instances provide direct access to the physical server's resources without the overhead of a hypervisor. These instances are best suited for workloads that require complete control over the underlying hardware, such as legacy applications that are not virtualizable, or those needing very high performance.

Key Instance Families:

M5.metal, C5.metal, R5.metal: These instances provide dedicated hardware and are ideal for workloads that need high-performance computing and full control over the hardware environment.
Deployment-This can be done with AWS Management Console or the AWS CLI with the process of launching, configuring, and managing Bare Metal Instances is straightforward and well-documented

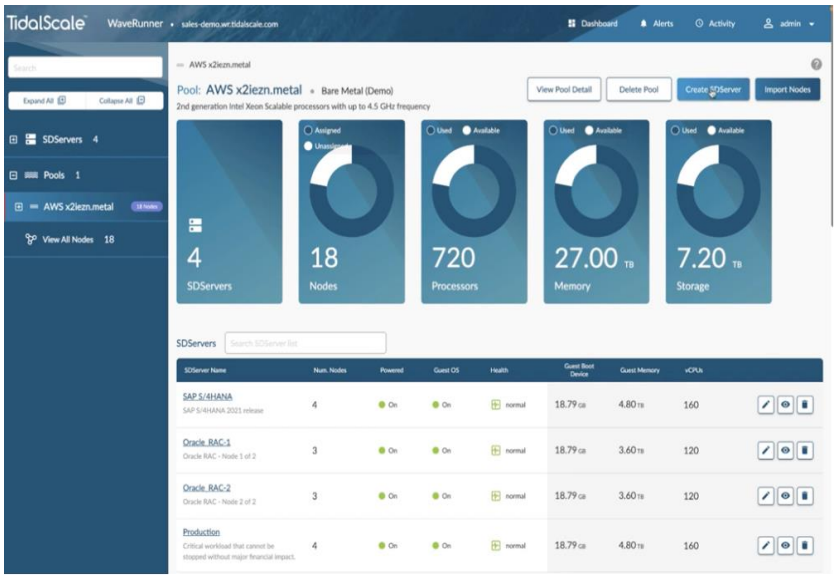
AWS Nitro and TidalScale mitigates virtualization overhead for high-volume workloads by offloading management tasks from the CPU. Nitro uses dedicated hardware to handle networking, storage, and security, allowing all host resources to be used for customer instances. TidalScale aggregates multiple AWS bare-metal instances into a single, high-performance system, ideal for resource-intensive applications.

This enables "hyper-convergence" by pooling multiple, high-performance physical servers into a single, massive software-defined virtual machine.



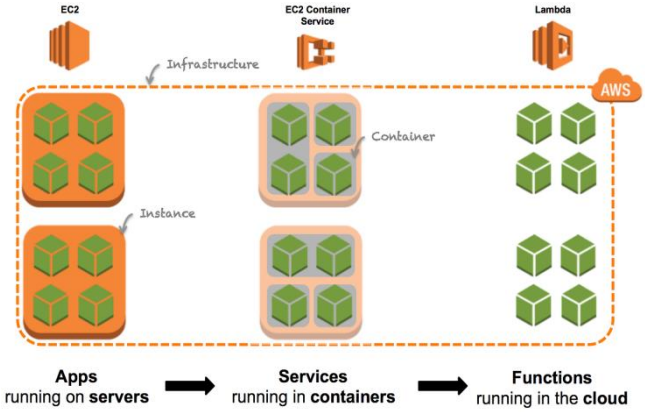
Key Performance & Technical Benefits

- Massive Scaling: Combines resources from several Nitro bare metal instances to create a single system image.
- Native Hardware Access: TidalScale's hyperkernel runs directly on AWS Nitro, leveraging Intel VT-x for near-native performance.
- Optimized Throughput: Utilizes high-speed, low-latency networking, such as Elastic Fabric Adapter (EFA), for node communication.
- Reduced Overhead: By eliminating traditional virtualization, it maximizes CPU and memory performance, making it ideal for memory-intensive workloads



2.2. Bare Metal Instances in AWS With Kubernetes

This illustrates the framework AWS Nitro with bare metal instance batch processing workloads.



A startup journey on AWS from bare metal monolith to serverless Bare-Metal Aws Infrastructure The new m5n, m5dn, r5n, and r5dn instances are already available and can utilize up to 100 gbps. The nitro system provides bare metal capabilities that eliminate virtualization overhead and support workloads that require full access to host. Aws recently announced new bare metal instances for amazon ec2. Bare metal instances allow ec2 customers to run applications that benefit from. Bare-Metal Aws Infrastructure.

Cost Effective Infrastructure Switching from Cloud to BareMetal Bare-Metal Aws Infrastructure Bare metal instances allow ec2 customers to run applications that benefit from deep performance analysis tools, specialized. The new m5n, m5dn, r5n, and r5dn instances are already available and can utilize up to 100 gbps. Aws recently announced new bare metal instances for amazon ec2. The nitro system provides bare metal capabilities that eliminate virtualization overhead and support workloads. Bare-Metal Aws Infrastructure.

Kubernetes Optimizations
To make EKS production-grade for High performance batch processing, apply the following factors:

- Taints/Tolerations: Isolated latency-sensitive workloads
- CPU pinning: Used static CPU manager policy and guaranteed QoS class
- Networking: Disabled kube-proxy and moved to Calico eBPF dataplane
- HugePages: Reduced TLB misses in memory-intensive processes
- Affinity Rules: Ensured collocation of tightly coupled workloads

AWS EKS-Anywhere is not the only tool to self-host Kubernetes. There are many, like below:
Rancher: Rancher simplifies Kubernetes cluster management with a user-friendly interface and supports multiple clusters across different environments.

Kubeadm: Kubeadm provides a simple way to set up and manage Kubernetes clusters by initializing and managing the control plane and worker nodes.

Metal3: Metal3 manages bare metal Kubernetes clusters, offering a way to provision and operate clusters on physical servers using Kubernetes-native tools.

K3s: K3s is a lightweight Kubernetes distribution designed for resource-constrained environments and ease of deployment, making it ideal for edge computing and development.

When do we use AWS EKS-Anywhere?

AWS EKS stands out because it integrates well with the AWS ecosystem and AWS services, such as Elastic Load Balancing, Identity and Access Management (IAM), and CloudTrail, making it easier to build comprehensive cloud native applications.

Real-World Performance Metrics

- Avg message latency (WebSocket -> Action): 650µs
- Max throughput (single pod): 47,200 msgs/sec
- Pod cold start (with image pre-warmed): ~2.1s
- Daily Financial transactions across 7 exchanges: ~54,000

AWS Nitro with well-tuned EKS, it forms the backbone of a truly modern, scalable trading platform.

2.3. Bare Metal Instances in AWS with Kubernetes for High Performance

System Architecture: Kubernetes + Nitro + Precision

1.Data Ingestion Pods
Handle WebSocket streams from Binance, Coinbase, OKX, etc.
Use asyncio-based Python consumers
Pinned to isolated cores via cpuManagerPolicy: static

2. Preprocessing Pods

Convert raw ticks to OHLC, VWAP and other custom technical indicators
Use shared memory mounts and Redis streams for speed

3. Execution Engine

Written in Python with C extensions
Uses aiohttp + uvloop
Direct access to ENA adapter for HTTP latency <500µs

4. Observability

Fluent Bit sidecars + AWS CloudWatch Logs
Prometheus Scraper + Amazon Managed Grafana
SLOs on message lag, CPU steal, and execution delay

We will compare c5n.18xlarge and the c5n.metal instance types in this post. The C5n instance family is commonly used for HPC applications and is based on the Intel Xeon Platinum 8000 series (Skylake-SP) processor with a sustained all core Turbo CPU clock speed of up to 3.5 GHz. Both of the c5n.18xlarge and the c5n.metal instances provide two CPUs with 18 physical cores each. Both these instances have 100Gb/s networking and Elastic Fabric Adapter (EFA) to run HPC applications at scale. Both these instances use the same underlying HW, only difference being the non-metal instance (c5n.18xlarge) using Nitro hypervisor

We have evaluated four different workloads: Weather Research and Forecasting (WRF) Model (weather forecasting), OpenFOAM (computational fluid dynamics), GROMACS (molecular dynamics), and High Performance Linpack (synthetic matrix solver). These were selected to show applications with different characteristics. In each case we run the application at a scale of 16 instances (576 cores) using AWS ParallelCluster and FSx for Lustre as the shared filesystem.

WRF (Weather Research and Forecasting Model): WRF is one of the most widely used numerical weather prediction (NWP) models with over 48,000 registered users spanning over 160 countries. The benchmark case used for this study is the CONUS 2.5km (version for WRF v4). As with other weather models, this workload will include I/O heavy portions of the workload to both read initial conditions as well as generate output. The performance metric used is the rate of simulation, based on the total wall-clock time that includes both I/O and compute time.

GROMACS: GROMACS is a molecular dynamics (MD) package designed for simulations of proteins, lipids, and nucleic acids. For this evaluation we run the benchRIB input set (Ribosome in water with 2M atoms) from Max Planck Institute for Biophysical Chemistry. The performance number used in comparisons is the ‘ns/day’ metric that GROMACS generates at the completion of a run.

OpenFOAM: OpenFOAM is an open-source computational fluid dynamics package. For this we use a scaled up (15 million cell) version of the 4 million cell motorbike case that is part of the standard OpenFOAM v2012 tutorial suite. The performance metric used for comparison is based on the total time for all iterations and converted to iterations per hour.

High Performance Linpack (HPL): HPL is a software package that solves a dense linear system in double-precision floating point. It forms the basis for the rankings of the Top 500 list, which ranks the “fastest supercomputers in the world.” Although this is not an end-application, we have included it for completeness. For these runs the optimized HPL implementation provided by Intel in the OpenAPI package was used. The performance metric used for comparison is the GigaFLOPs reported at completion of the run.

Dimensions	Kubernetes on VMs	Kubernetes on Bare Metal
Operating System	•Distinguish Host OS from Node OS. •VMs can provide various Node OS for applications.	•Users need to install an OS supported by the bare-metal server. •Host OS also serves as Node OS, meaning that only one OS can be deployed and shared among all applications.
Access of Resources	•A single physical server can host multiple Kubernetes nodes. •With hypervisor managing and allocating hardware resources, VMs on the same server share the entirety of the hardware resources.	•A single physical server hosts only one Kubernetes node •Applications have direct access to the hardware resources.

	Kubernetes on VMs	Kubernetes on Bare Metal
Suitable Scenarios	• Limited resource investment • Limited O&M personnel • Difficulty in estimating future resource usage • Existing production applications running on virtualized/hyperconverged infrastructure • Requirement for quick deployment and flexible scaling of Kubernetes clusters • Need for automation to handle large volumes of repetitive tasks • Need to provide separate Kubernetes environments for "multi-tenancy" • Need to support both virtualized and containerized applications within constrained resources.	• Ample resources and investment budget • Adequate O&M personnel • Absence of virtualized or hyperconverged infrastructure • Running resource-intensive applications that require dedicated hardware resources, including high-performance computing (HPC) applications, machine learning and deep learning, online gaming, virtual reality, etc. • Subject to strict data compliance regulations, with each application requiring a dedicated hardware environment.

2.4. Bare Metal instance in AWS vs On Prem Bare metal

When comparing bare metal instances in AWS to on-prem bare metal, consider the following key differences:

Control and Management: AWS provides a managed environment with AWS management, while on-prem bare metal servers require more hands-on management and maintenance.
Performance: AWS bare metal instances offer direct access to hardware resources, which can lead to better performance for demanding workloads.

Cost: AWS pricing is typically based on usage, while on-prem bare metal servers may have a higher upfront cost but can be more predictable for predictable workloads.

Scalability: AWS offers more flexibility in scaling resources, while on-prem setups may have limitations in managing the number of servers.

Security and Compliance: AWS provides a higher level of security and compliance features, while managing security on on-prem servers is the responsibility of the organization.

In summary, AWS bare metal instances provide a more scalable and managed environment, while on-prem bare metal offers greater control and flexibility for specific use cases.

Performance & Hardware Access

CPU/GPU Options:
AWS offers the latest hardware across multiple instance types:
i3.metal, i7ie.metal, c5n.metal, m5.metal, and GPU-powered p5.metal.

Throughput:
AWS bare metal supports up to 25 Gbps networking via C5n, I7ie, and C7gn.metal.
Virtualization Overhead:
Neither uses virtualization. You get full access to the underlying hardware.
AWS achieves this through the Nitro System.

Deployment & Availability

Region Coverage:
AWS has bare metal servers in every major region worldwide.
Self-Serve vs Contact-to-Deploy:
On AWS, you spin up a bare metal server just like any EC2 instance.
Speed to Provision:
AWS bare metal is ready in minutes via API or Console.

2.5. Bare Metal instance in AWS vs VM instances in AWS

Bare Metal instances in AWS vs VM instances in AWS:

Performance: Bare Metal instances provide direct access to physical hardware, resulting in faster disk I/O and lower latency compared to VM instances.

Resource Allocation: VM instances can run multiple applications on a single physical server, while Bare Metal instances are dedicated to a single user, allowing for more efficient resource allocation.

Control and Customization: Bare Metal instances offer complete control over the hardware and software environment, making them ideal for applications that require high performance and customization.
Scalability: VM instances can be scaled easily by adding more resources, while Bare Metal instances may require purchasing additional hardware for scaling.

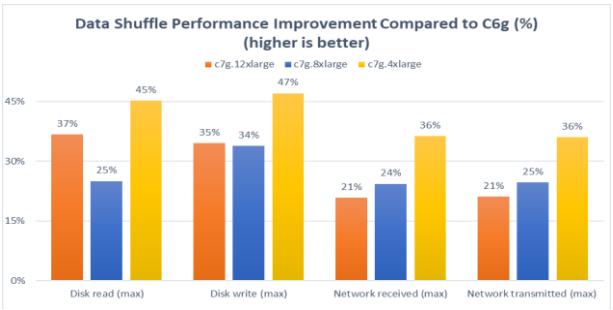
Cost: While Bare Metal instances have a higher upfront cost, they can offer long-term cost savings for high-demand applications due to their performance efficiency
EC2 is my go-to when in need of flexibility and auto-scaling across various AWS services.
ECS is perfect for simplifying containerized applications without the hassle of managing individual instances.

Bare metal wins hands down for high-performance needs, complete control, and security.

Bare Metal Fits Best:

HPC and applications that need every ounce of performance.
Long-term, stable projects that can benefit from predictable costs and flat-rate pricing.
Compliance-heavy industries that require dedicated hardware.

K8s cluster type	CPU		RAM latency (write/read)	Storage		Network	
	Speed (execution time)	Utilization		TPC (transactions per second)	Latency	Bandwidth	Latency
VM	47.07 sec	86.81%	174.53 / 173.75 ms	4,636	55.21 ms	6.52 MB/sec	145 us
Bare metal	21.46 sec	43.75%	62.02 / 47.33 ms	12,029	21.28 ms	31 MB/sec	24.5 us

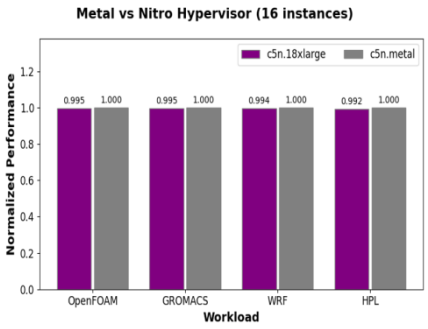


2.6. AWS Baremetal in Nitro Performance

Section headings are numbered 1. Xxx, 2. Yyy, etc. in 14 pt. bold “Small Caps” Times New Roman font with a 6 pt. line spacing following.
Subsection headings are numbered 1.1. Aaa, 1.2. Bbb, etc. in 12 pt. bold Times New Roman font with a 6pt line spacing following.

2.6.1.AWS Baremetal in Nitro Performance with cost

Bare metal performance comparison



WRF (Weather Research and Forecasting Model): WRF is one of the most widely used numerical weather prediction (NWP) models with over 48,000 registered users spanning over 160 countries.

GROMACS: GROMACS is a molecular dynamics (MD) package designed for simulations of proteins, lipids, and nucleic acids. For this evaluation we run the benchRIB input set (Ribosome in water with 2M atoms) from Max Planck Institute for Biophysical Chemistry.

OpenFOAM: OpenFOAM is an open-source computational fluid dynamics package. For this we use a scaled up (15 million cell) version of the 4 million cell motorbike case that is part of the standard OpenFOAM v2012 tutorial suite.

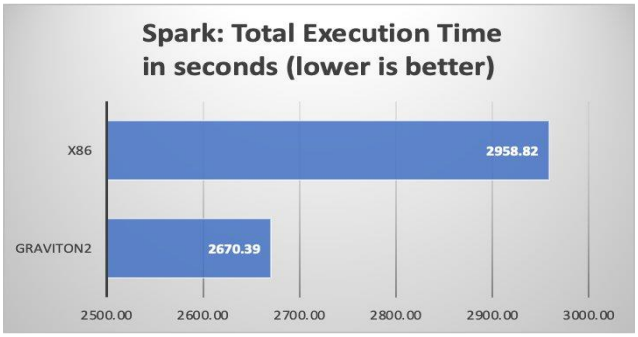
High Performance Linpack (HPL): HPL is a software package that solves a dense linear system in double-precision floating point. It forms the basis for the rankings of the Top 500 list, which ranks the "fastest supercomputers in the world."

How do bare metal and VPS compare across performance, cost, and security?

Dimension	VPS (Virtual Private Server)	Bare Metal Server
Performance consistency	Variable. Performance can fluctuate due to shared hardware and noisy neighbors.	Deterministic. Full access to physical resources with no contention.
Latency	Higher and less predictable due to virtualization overhead.	Lower and consistent, ideal for latency-sensitive workloads.
Resource isolation	Logical isolation via hypervisor. Hardware is shared.	Physical isolation. Single-tenant by design.
Cost model	Appears low at entry but scales unpredictably with usage.	Fixed, transparent monthly pricing with predictable spend.
Cost efficiency at scale	Degrades as workloads grow and require more instances or premium tiers.	Improves with sustained utilization and stable workloads.
Security posture	Shared infrastructure increases attack surface and audit complexity.	Reduced risk due to physical isolation and no shared tenants.
Compliance readiness	Requires additional controls and documentation to meet standards.	Simplifies SOC 2, PCI DSS, HIPAA, and data residency requirements.
Scalability speed	Fast to spin up new instances for short-term needs.	Scales more deliberately but with predictable performance outcomes.

Batch vs. Streaming Suitability

- Streaming pipelines: Bare metal is superior for low-latency, real-time data ingestion and processing where jitter or latency spikes are unacceptable.
- Batch Pipelines: For large, predictable, and sustained jobs, bare metal provides the highest performance. For smaller or unpredictable batch jobs, virtual instances (especially with AWS Batch) allow better cost management.



The cost breakdown for x86:

- Job runtime – 49.3 minutes = 0.82 hours
- Total vCPU cost – 404 vCPUs x 0.82 hours job runtime x 0.052624 USD per vCPU = 17.4333 USD

- Total GB cost – 1,616 memory-GBs x 0.82 hours job runtime x 0.0057785 USD per memory GB = 7.6572 USD
- Storage cost – 18,000 storage-GBs x 0.82 hours job runtime x 0.000111 USD per storage GB = 1.6386 USD
- Additional storage – 20,000 GB – 20 GB free tier * 100 workers = 18,000 additional storage GB
- EMR Serverless total cost (x86): 17.4333 USD + 7.6572 USD + 1.6386 USD = 26.7291 USD

Let's compare to the cost breakdown for Graviton 2:

- Job runtime – 44.5 minutes = 0.74 hours
- Total vCPU cost – 404 vCPUs x 0.74 hours job runtime x 0.042094 USD per vCPU = 12.5844 USD
- Total GB cost – 1,616 memory-GBs x 0.74 hours job runtime x 0.004628 USD per memory GB = 5.5343 USD
- Storage cost – 18,000 storage-GBs x 0.74 hours job runtime x 0.000111 USD per storage GB = 1.4785 USD
- Additional storage – 20,000 GB – 20 GB free tier * 100 workers = 18,000 additional storage GB
- EMR Serverless total cost (Graviton2): 12.5844 USD + 5.5343 USD + 1.4785 USD = 19.5972 USD

The tests indicate that for the benchmark run, AWS Graviton2 lead to an overall cost savings of 27%.

We measure the Graviton performance and cost improvements using two calculations: total query runtime and geometric mean of the total runtime. The following table shows the results for equivalent sized C6g and C7g instances and the same Spark configurations.

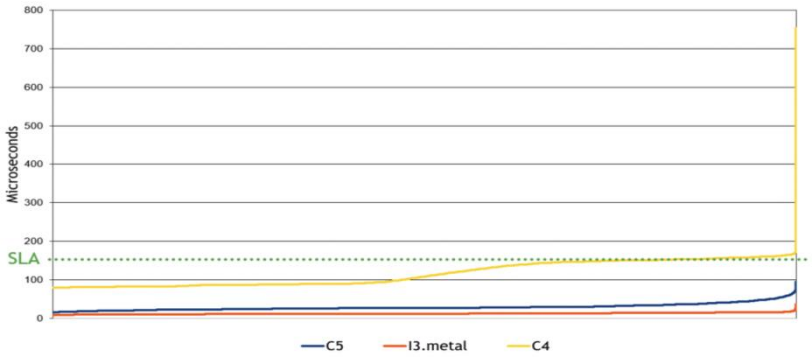
Benchmark Attributes	12 XL	8 XL	4 XL
Task parallelism (spark.executor.core*spark.executor.instances)	188 cores (4*47)	188 cores (4*47)	188 cores (4*47)
spark.executor.memory	6 GB	6 GB	6 GB
Number of EC2 instances	5	7	16
EBS volume	4 * 128 GB io1 disk	4 * 128 GB io1 disk	4 * 128 GB io1 disk
Provisioned IOPS per volume	6400	6400	6400
Total query runtime on C6g (sec)	2099	2098	2042
Total query runtime on C7g (sec)	1728	1738	1660
Total run time improvement with C7g	18%	17%	19%
Geometric mean query time on C6g (sec)	9.74	9.88	9.77
Geometric mean query time on C7g (sec)	8.40	8.32	8.08
Geometric mean improvement with C7g	13.8%	15.8%	17.3%
EMR on EKS memory usage cost on C6g (per run)	$0.28	$0.28	$0.28
EMR on EKS vCPU usage cost on C6g (per run)	$1.26	$1.25	$1.24
Total cost per benchmark run on C6g (EC2 + EKS cluster + EMR price)	$6.36	$6.02	$6.52

2.6.2. AWS Baremetal in Nitro Performance datapoints

BareMetal use cases-

There are several scenarios or application types that might require, or benefit from, AWS bare-metal instances over virtualized EC2 instances:

- Legacy applications that weren't built for and won't run properly in virtualized environments.
- Applications with archaic software licensing terms and enforcement mechanisms tied to hardware.
- Software that requires direct hardware access, such as Type 1 hypervisors. AWS bare-metal instances can run Hyper-V, which can be done with a virtual switch connected to a VPC and private Dynamic Host Configuration Protocol server that's ready to launch guest VMs just like a stock Windows Server.
- Kubernetes container clusters run without the overhead of VMs. Containers on EC2 or container instances on Fargate are already quite responsive and can be provisioned in seconds, but running cluster nodes on bare metal enables more container instances per node and further reduces latency.



2.7. AWS Baremetal instances for AIML workloads

AWS offers bare metal instances with GPU capabilities, such as the G4dn.metal and G5g.metal instances, providing high performance for GPU-intensive workloads like machine learning and graphics rendering.

GPU-Enabled Bare Metal Instances
G4dn.metal:

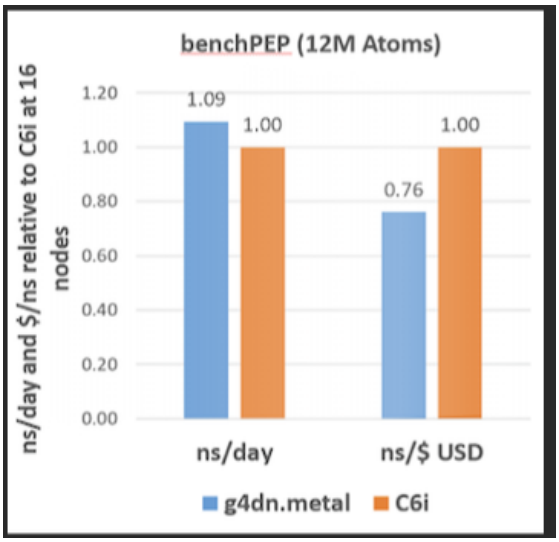
Specifications: Equipped with 8 NVIDIA T4 GPUs, 96 vCPUs, and 384 GiB of RAM.
Use Cases: Ideal for machine learning inference, small-scale training, graphics-intensive applications, and remote workstations. G4dn instances are optimized for cost-effectiveness in GPU workloads

Use Cases

- **Machine Learning**: Inference and small-scale training with TensorFlow, PyTorch, MXNet, etc.
- **Graphics Rendering**: 3D rendering, CAD, and video processing.
- **Game Streaming**: Cloud gaming platforms.
- **Remote Workstations**: High-performance virtual desktops for designers and engineers.
- **Video Transcoding**: GPU-accelerated media processing.

Advantages of metal Variant

- **Bare Metal Access:** Direct access to the CPU and GPU hardware for maximum performance.
- **No Virtualization Overhead:** Ideal for workloads requiring low latency and hardware-level tuning.
- **Full GPU Control:** Useful for custom drivers, CUDA, and GPU passthrough scenarios.



3. CONCLUSION

To conclude between AWS Bare metal instances and aws vm instances based on cost and performance for Legacy batch processing and streaming applications.

This is one of the conceptualisation models proposed for migration of legacy applications in bare metal instances in On prem to AWS bare metal instances for a simulated high performance system.

Performance: Bare metal instances provide direct access to hardware resources, offering up to 15% better performance over comparable x86-based Intel processors compared to virtual instances.

Scalability: Virtual instances allow for scalability and flexibility, making them suitable for dynamic workloads, while bare metal instances are better for high-performance applications that require dedicated resources.

To achieve scalability in AWS Bare metal instances- with TidalScale on AWS customers now aggregate the CPUs, memory, network, interrupts, and storage of multiple AWS bare metal instances into a single system image capable of running unmodified operating systems, middleware and applications. Each bare metal instance was designed by AWS with an optimal ratio of CPU performance to memory bandwidth, both of which are virtualized to form a software-defined server from several bare metal instances.

To enable vertical scale up from horizontally-scaled AWS metal instances, TidalScale virtualizes processors, IO, and memory, and creates Scalable Coherent Shared Memory which is an efficient, scalable, coherent distributed memory .

Control and Customization: Bare metal instances offer complete control over the hardware and software environment, which can be beneficial for resource-intensive applications.

Cost Efficiency: Virtual instances can be more cost-effective due to shared resources, but they may introduce overhead and performance bottlenecks for certain workloads.

Use of Reserved Instances and Savings Plans

- *Reserved Instances (RIs)*: Commit to a specific instance type, region, and tenancy for 1 or 3 years to benefit from significant discounts.
- *Savings Plans*: Flexible commitments offering discounts across multiple instance families, ideal for predictable workloads.
- For bare-metal, RIs and Savings Plans can significantly reduce ongoing costs; however, their applicability depends on workload stability.
- **Security:** Bare metal instances provide higher levels of data privacy since they are dedicated to a single user, reducing the risk of interference from other virtual machines.
- HIPAA Compliance: Nitro instances allow for strict, dedicated physical isolation, which is crucial for handling Protected Health Information (PHI). Organizations must configure services to encrypt data at rest and in transit, maintain strict IAM policies, and enable auditing via CloudTrail.
- GDPR Compliance: AWS provides a Data Processing Addendum (DPA) to meet GDPR requirements. The Nitro system, which offloads virtualization to hardware, ensures that AWS personnel cannot access customer data on the instance, supporting GDPR's data privacy obligations.
- Shared Responsibility Model: While AWS manages physical security, hardware, and the hypervisor, you are responsible for the OS, applications, data encryption, and network configuration on the bare metal server.
- Security Controls:
  - AWS Artifacts: Use this to download compliance reports (SOC2, ISO).
  - AWS Config: Monitor and enforce compliance rules automatically.
  - Encryption: Utilize AWS Key Management Service (KMS) for data protection

In summary, the choice between bare metal and virtual instances depends on the specific needs of your application, including performance requirements, scalability, control, and cost considerations.

ACKNOWLEDGEMENTS

The authors would like to thank everyone, just everyone!

REFERENCES

[1] Different Types of EC2 instances. Amazon EC2 (Elastic Compute Cloud) is… | by Amirthanivas | Medium
[2] Bare-Metal Workloads on AWS with EC2 Bare Metal Instances: Deploy SAP HANA or GPU-Heavy AI Workloads on Bare Metal Instances - DEV Community
[3] Turbocharging Kubernetes with AWS Nitro: Building a Low-Latency Trading Pipeline for High-Frequency Strategies | by Svetlozar Kondakov | Devmap | Medium
[4] Hyper Metal: Scaling AWS Instances Up with TidalScale | AWS HPC Blog

AUTHOR

Sharada Lakshmanan