The Virtualization Tax

Why bare metal servers deserve a second look in the age of cloud-everything

Vanni Torelli, Gluon Consulting Pte. Ltd.

April 2026 (1.0)

Contents

Executive summary

The cloud computing narrative of the last fifteen years has been remarkably effective: virtualization is presented as a near-costless abstraction that unlocks elasticity, global scale, and operational simplicity. For many workloads, this is true. For a large and growing class of production systems, those with stable, predictable resource requirements, though, the industry has over-rotated toward virtualization as a default, paying a significant performance and economic penalty for flexibility that is never exercised.

This paper examines the measurable costs of hypervisor-layer abstraction, drawing on published benchmarks, public financial filings, and real-world repatriation case studies.

The argument is not that cloud infrastructure is fundamentally flawed; rather, that the industry has conflated the value of managed infrastructure with the value of virtualization, and they are not the same thing. You can have the former without the latter, and for the right workloads, you should.

The hidden cost of virtualization

Every layer of abstraction has a cost. In virtualized environments, that cost manifests across CPU, memory, storage, and network, often in ways that are invisible to application-level monitoring.

CPU steal time

In multi-tenant cloud environments, virtual CPUs compete for physical core time. When the hypervisor schedules another tenant's workload on your physical core, your vCPU waits.

This accumulated wait time, visible as "steal time" in Linux tools like top and mpstat, directly reduces your effective compute capacity. A batch job on a VM experiencing 20% steal time takes proportionally longer: a 2-hour ETL job becomes a 2.5-hour job.

The noisy neighbor problem

Resource contention extends well beyond CPU. Hypervisors use memory balloon drivers to reclaim RAM when the host is under pressure, potentially triggering OS-level swapping without warning.

Physical NICs are shared across tenants, introducing jitter. A server with DDR5-4800 in a 4-channel configuration offers roughly 153 GB/s of theoretical memory bandwidth; split across four VMs, each gets approximately 38 GB/s under ideal conditions, and less under contention.

Network-attached storage overhead

Cloud block storage (AWS EBS, Google Persistent Disk, Azure Managed Disks) is fundamentally network-attached. Every I/O request traverses the data center's

internal network to a storage cluster, adding 0.5–2ms of latency per operation compared to local NVMe. For database workloads where I/O latency compounds across thousands of queries per second, this overhead is substantial.

On bare metal, steal time is zero by definition.

Performance benchmarks

Published benchmarks consistently show significant performance gaps between bare metal and virtualized environments when running identical workloads.

Gcore Kubernetes benchmark (2023)

Gcore conducted a controlled comparison of Kubernetes clusters deployed on bare metal worker nodes versus VM-based worker nodes. The results across CPU, memory, and storage were striking.

Metric	Bare Metal	Virtual Machine	BM Advantage
CPU (pi calc, seconds)	21.46s	47.07s	2.2× faster
RAM throughput	—	—	≈3× faster
PostgreSQL TPS (8GB)	14,087	7,359	1.9×
PostgreSQL TPS (75GB)	12,029	4,636	2.6×

Hypervisor overhead in isolation

Industry measurements indicate that the hypervisor layer alone, before any noisy-neighbor contention, consumes 5–10% of the host's physical resources. With multi-tenant contention factored in, effective performance degradation can reach 20–30%. For workloads that run at or near capacity, this is not a rounding error, it is the difference between meeting and missing SLAs.

Tail latency: where it hurts most

Cloud environments are generally optimized for average throughput, not worst-case latency. The p99 latency gap between bare metal and VMs is consistently

larger than the median gap, because the worst 1% of requests suffer from steal time spikes, VM live migrations, and storage network variability, sources of jitter that simply do not exist on dedicated hardware. For real-time systems, trading platforms, and game servers, p99 latency determines user experience.

The economics

The performance case is supported by an equally compelling economic argument. Several high-profile companies have publicly documented the financial impact of moving workloads off hyperscale cloud infrastructure.

Dropbox

Dropbox built “Magic Pocket,” a custom storage system in colocation facilities, migrating 90% of customer data off AWS. As documented in their S-1 filing, this initiative saved approximately $75 million over two years. Gross margins improved from 33% to 67% between 2015 and 2017, driven primarily by the infrastructure optimization.

37signals (Basecamp / HEY)

After spending over $3.2 million per year on AWS, 37signals invested approximately $600,000 in Dell servers in colocation. One year after completing the transition, they reported annual savings of $2 million, projecting $10 million in total savings over five years.

Dukaan

The Indian e-commerce platform moved from public cloud to managed bare metal and saw monthly infrastructure costs drop from $90,000 to $1,500, a 98% reduction—while also resolving persistent latency issues.

GEICO

After spending a decade migrating over 600 applications to public cloud, GEICO found itself spending more than $300 million annually on cloud providers. The insurer has since begun selectively repatriating storage-intensive workloads.

Ahrefs

Perhaps the most striking case of all. Ahrefs, the Singapore-based SEO platform, operates approximately 850 servers in a colocation data center with its own hardware. Their engineering team calculated that running the equivalent infrastructure on AWS would have cost roughly $448 million over 2.5 years, compared to the $39.5 million they actually spent in colocation. That is an 11.3× cost difference per server.

The existential dimension makes this case unique: Ahrefs' total revenue over those three years was approximately $257 million. The hypothetical AWS bill would have exceeded their entire revenue, meaning the company would not have been profitable, or even viable, as a cloud-only business. A follow-up analysis covering six years showed that their data centers cost $122 million in total, a figure that would have ballooned to approximately $1.1 billion on the cheapest AWS offering.

Notably, Ahrefs still uses AWS for frontend hosting across global regions, a pragmatic hybrid approach that puts heavy compute on bare metal and uses cloud only where its strengths (global edge distribution) genuinely apply. Their own NVMe drives also outperform AWS's EBS gp3 storage, so the move was not just cheaper but faster.

Company	Action	Reported Savings
Ahrefs	Own hardware in colocation (850 servers)	$400M+ avoided over 2.5 yr
Dropbox	Built custom infra in colocation	$75M over 2 years
37signals	Migrated to owned servers in colocation	$2M/year ($10M over 5 yr)
Dukaan	Moved to managed bare metal	$90K→$1.5K/month
GEICO	Selective repatriation	Was spending $300M+/yr

Andreessen Horowitz estimates that across 50 top public software companies, cloud infrastructure costs are suppressing approximately $100 billion in market value due to their impact on gross margins.

Bare metal hosting: OpEx without the hypervisor

A persistent misconception equates bare metal with capital expenditure: buying servers, racking them, and managing a cage in a colocation facility. This framing is outdated. Modern bare metal hosting providers offer dedicated physical servers on monthly or hourly billing, combining the performance of dedicated hardware with the operational model CFOs expect.

The false dichotomy

The cloud industry has bundled several distinct value propositions into a single offering: virtualization, managed services, elasticity, and a marketplace of integrated tools. The assumption that you must accept virtualization overhead to access the others is false. You can rent bare metal servers from providers like Hetzner, OVH, Equinix Metal, or Vultr on a pure OpEx basis, and still use managed databases, S3-compatible object storage, CDNs, and monitoring services alongside them.

The price gap

A dedicated server with a 64-core AMD EPYC processor, 128GB RAM, and dual NVMe drives can be rented from European bare metal providers for approximately €100–200 per month. An equivalently-specced cloud VM on a major hyperscaler can cost 5–10× that amount. The difference is the cloud provider's margin for the hypervisor, control plane, and the elasticity overhead that a stable workload never uses.

What you still get

Modern bare metal providers handle hardware failures, provide remote management (IPMI/KVM), offer DDoS protection, and expose provisioning APIs.

You are not managing physical infrastructure, you are renting a machine that someone else maintains, but that runs your OS, your kernel tuning, and your software without a hypervisor in between.

Capability	Cloud VM	Bare Metal Hosting
Billing model	OpEx (hourly/monthly)	OpEx (hourly/monthly)
Hypervisor overhead	5–10%+ baseline	None
Hardware failure handling	Provider-managed	Provider-managed
Auto-scaling	Built-in	Manual / API-provisioned
Kernel / OS control	Limited	Full
Noisy neighbor risk	Present	None
Cost at equivalent spec	5–10× higher	Baseline

The drawbacks

An intellectually honest case for bare metal must acknowledge the real challenges. The difficulty is, in fact, part of the competitive moat for teams that can execute on it.

Talent scarcity

Writing metal-sympathetic code requires developers versed in memory models, cache coherence, lock-free data structures, NUMA-aware allocation, and low-level concurrency primitives. These skills are rare and expensive. Choosing bare metal narrows your hiring pool significantly.

Concurrency complexity

Bare metal invites you to exploit every core and hardware thread, but concurrent programming remains one of the hardest disciplines in software engineering. Race conditions, deadlocks, and subtle memory-ordering bugs are difficult to reproduce and catastrophic in production.

Provisioning speed

While bare metal hosting providers have dramatically improved provisioning times, spinning up a new dedicated server still takes minutes to hours, versus seconds for a cloud VM. For workloads with genuinely unpredictable demand spikes, this difference matters.

Hardware portability

Code tuned for a specific CPU microarchitecture may behave differently on next-generation hardware. You are coupling software to silicon in ways that create migration friction.

Institutional knowledge risk

When your system depends on deep hardware-specific expertise held by a small number of engineers, you have created a bus-factor problem. Cloud-native patterns, being more mainstream, have a broader knowledge base.

Diminishing returns for most workloads

Most applications are I/O-bound, waiting on databases, network calls, or user input. The microsecond-level gains from bare metal optimization do not matter for a typical CRUD API or content management system. The investment is only justified for genuinely performance-critical workloads.

A decision framework

The choice between bare metal and cloud VMs is not ideological, it is workload-specific. The following framework identifies the characteristics that make bare metal the stronger choice.

Workload Characteristic	Favors Bare Metal	Favors Cloud VM
Demand pattern	Stable / predictable	Spiky / seasonal
Latency sensitivity	p99 matters (trading, gaming)	Average latency is sufficient
Compute intensity	CPU/GPU-bound (ML, HPC, video)	I/O-bound (CRUD, web apps)
Data volume	Large (egress fees dominate)	Small / moderate
Team expertise	Systems engineers on staff	Application-focused team
Lifetime	Long-running production service	Experimental / short-lived
Compliance	Physical isolation required	Standard multi-tenant OK

The strongest candidates for bare metal are workloads that are compute-intensive, latency-sensitive, stable in demand, and long-lived. This describes a surprisingly large fraction of production infrastructure.

Industries Leading the Shift

High-frequency trading firms (Citadel, Jump Trading) have always run bare metal with kernel-bypass networking. Game studios like Riot Games use bare metal for latency-critical game servers. Large ML training clusters, even at cloud providers themselves, run on bare metal internally. The cloud repatriation trend is

accelerating: a 2024 Barclays CIO Survey showed nearly all respondents planned to move some public-cloud workloads back to private infrastructure.

The Gluon advantage

This paper has outlined the performance and economic case for bare metal, but it has also been candid about the barriers. Writing metal-sympathetic code demands deep expertise in concurrency, memory management, and hardware-aware optimization. These skills are scarce, and building an in-house team with the required depth takes years.

That is precisely the gap Gluon was built to fill.

High-performance computing, by design

Gluon is a specialist consultancy focused exclusively on high-performance computing for mission-critical systems. Our practice is built around the disciplines that matter most when every microsecond counts: distributed, low-latency architectures, parallel programming, lock-free concurrency, memory-efficient design, and hardware-sympathetic code that extracts maximum throughput from modern server platforms.

Low-GC and no-GC engineering

Garbage collection pauses are one of the most common sources of latency spikes in managed-language applications. Gluon's consultants are experts in minimizing or eliminating GC pressure through techniques such as object pooling, stack-based allocation, value types, struct-oriented design, off-heap memory management, and arena allocation patterns. We design systems where memory behavior is predictable and deterministic, not left to the whims of a runtime collector.

Stack-based and cache-sympathetic programming

Modern CPUs are fast; main memory is slow. The performance gap between L1 cache and DRAM can be two orders of magnitude. Gluon's engineers design data structures and access patterns that keep hot data on the stack and in cache, minimize pointer chasing, and exploit sequential memory access for hardware prefetching. This is the level of optimization that turns a 2× benchmark advantage into a 5× real-world advantage under production load.

20+ years in critical environments

Every Gluon Senior Consultant brings more than 20 years of hands-on experience in high-performance C#/.NET and Java, matured in environments where failure is not an option and latency is measured in microseconds. Our team's track record spans three of the most demanding sectors in software engineering.

Sector	What we deliver
Banking & Finance	Ultra-low-latency trading systems, real-time risk engines, high-throughput transaction processing, and regulatory-compliant infrastructure where downtime is measured in dollars per second.
Industrial Automation	Real-time control systems, SCADA integration, deterministic processing pipelines, and edge computing platforms where timing guarantees are non-negotiable.
Gaming	High-tick-rate game servers, real-time physics and simulation engines, low-latency networking stacks, and player-facing infrastructure where every frame matters.

Bridging the talent gap

Talent scarcity is the primary barrier to adopting bare metal infrastructure. Gluon exists to eliminate that barrier. Whether you need to optimize an existing application for bare metal deployment, design a new high-performance system

from the ground up, or train your internal team in metal-sympathetic programming techniques, Gluon's consultants bring the depth of experience required to deliver results, without the years of hiring and learning curve.

The market itself validates this point. Even the most aggressive cost-optimization firms in the software industry, companies whose entire business model is acquiring struggling SaaS products and ruthlessly cutting operational expenditure, advertise senior platform engineering roles at more than double the going market rate, fully remote, to attract the caliber of talent needed for genuine re-architecture work.

Routine maintenance can be commoditized. Transformative engineering cannot. The expertise required to re-platform a bloated, over-virtualized application into a lean, metal-sympathetic system commands a premium precisely because so few engineers possess it. That scarcity is Gluon's reason for being.

You don't need to build a team of low-latency specialists from scratch. Gluon brings 20+ years of battle-tested expertise in high-performance .NET, Java, and Rust, ready to deploy on your most demanding workloads.

Conclusion

The cloud revolution delivered genuine value: it democratized infrastructure, enabled startups to launch without capital expenditure, and built a rich ecosystem of managed services. None of this is in question.

What is in question is the assumption that virtualization overhead is negligible, that the hypervisor tax is a rounding error, and that cloud pricing is always economically rational for production workloads.

The benchmarks tell a different story: 2–3× performance gaps in CPU, memory, and storage. The financial data tells a different story: companies saving tens of millions by moving to dedicated infrastructure. The market analysis tells a different story: an estimated $100 billion in suppressed market capitalization across 50 public software companies.

The thesis is not “cloud bad, bare metal good.” The thesis is more precise: the industry has conflated the value of managed infrastructure with the value of virtualization, and they are not the same thing. Modern bare metal hosting providers deliver the operational model of cloud: OpEx billing, managed hardware, API provisioning, without the hypervisor in between.

For workloads with stable demand, performance sensitivity, and meaningful scale, bare metal is not a step backward. It is a deliberate, data-driven choice to stop paying a tax on every CPU cycle for flexibility that was never needed.

The best argument is a benchmark. Take a representative production workload, run it on an equivalently-specced cloud VM and a bare metal server at the same price point. Measure the delta. That delta is the virtualization tax. For the right workloads, it is not small.