PERSPECTIVE

Server Consolidation: Balancing Scalability, Overhead, and Workload Dependencies

Belen Bermejo[1,*], **Carlos Juiz**[1] **and David Cortes**[1]

[1] Computer Science Department, University of the Balearic Islands, Palma 07122, Spain

Abstract

Server consolidation through virtualization is the primary strategy for optimizing data centres, aiming to maximize server efficiency and reduce costs. However, the long-term success of this strategy is compromised by three interconnected factors: virtualization overhead, the difficulty of modelling representative workloads with dependencies, and scalability limitations. This perspective article argues that scalability in consolidated environments is not a hardware problem, but rather a systems engineering problem, requiring a deep understanding of the interactions and constraints imposed by hypervisor overhead.

Keywords: server consolidation, scalability, workload dependence, virtualization overhead.

1 The Duality of Efficiency: Overhead and Maximum Density of Consolidation

The goal of server consolidation is to reduce the number of physical servers by increasing the number of virtual machines per physical server. Consequently, the utilization of physical server resources should increase [3, 4].

However, this increase in virtual machine density comes at an unavoidable cost: the overhead generated by the hypervisor. This intermediate software layer, responsible for isolation, emulation, and the management of shared resources (CPU, memory, I/O), consumes its own compute and memory resources. Even the most efficient hypervisors, such as Type 1 hypervisors, cannot eliminate the negative impact on performance.

This is magnified when aiming for maximum consolidation density. The overhead increases exponentially as the hypervisor must dedicate more time to arbitrating resource access and contention. When attempting to scale vertically (more virtual machines on a single server), a critical point is reached where the overhead significantly degrades the performance of all virtual machines, thus limiting effective scalability [4].

2 The Fragility of Scalability and Load Dependence

Scalability in a virtualized environment is understood in two ways: vertical scalability (scale-up), which increases the virtual resources (vCPU, vRAM) of an individual virtual machine, and horizontal scalability

Submitted: 22 December 2025
Accepted: 22 June 2026
Published: 27 June 2026

Vol. 1, **No.** 2, 2026.
10.62762/JSS.2025.511991

***Corresponding author:**
✉ Belen Bermejo
belen.bermejo@uib.es

Citation
Bermejo, B., Juiz, C., & Cortes, D. (2026). Server Consolidation: Balancing Scalability, Overhead, and Workload Dependencies. *Journal of Systems Scalability*, 1(2), 50–54.

(scale-out), which adds more virtual machines or more physical servers to the cluster [5].

The problem of efficient consolidation is closely related to horizontal scaling and representative load. It is worth noting that modelling interdependent workloads is challenging. The key to scalable consolidation lies in accurately predicting the resource demands of the virtual machines. This is considerably complicated by application dependencies.

An important aspect to highlight is that consolidated environments introduce performance interference between co-located virtual machines [1, 2]. Dependencies in CPU-based workloads differ from dependencies in workloads based on mixed CPU and I/O environments [8]. This is because the CPU is a device that scales best linearly. However, I/O often represents a bottleneck in systems.

For the reasons mentioned above, in this article, we will choose an example based on a mixed workload with dependencies. We will define three entities, "A", "B", and "C", which make up a system. Application 'A' depends on database 'B', which in turn depends on server 'C'. If A, B, and C are consolidated onto different servers, the load is distributed, although the performance of the communication network is critical. Conversely, if A, B, and C are consolidated onto the same server, communication across the network is eliminated, but synchronous resource contention is created. A high-volume request to A could trigger simultaneous demand spikes on B and C, which then compete for the same CPU and I/O subsystem resources. In this case, the representative load must model this concurrent dynamic, not simply the sum of individual demand spikes. Failure to model these interdependencies leads to an underestimation of contention risk and limits the ability to scale out, as each new addition increases the likelihood of an unpredictable latency bottleneck on the overloaded server.

Consolidation enhances data centre scale-out by freeing up physical servers to accommodate more virtual machines (or services). However, at the individual server level, the consolidation process imposes a strict limit on consolidation density.

The hypervisor must arbitrate access to shared physical resources (CPU, memory, I/O). As more virtual machines are added to a host (increased density), the time spent on this arbitration (overhead) increases. The process of increasing density in consolidation goes through the following phases:

- **Phase 1:** Linear scalability. Initially, performance per virtual machine decreases slightly, but overall host performance increases. Therefore, consolidation is efficient.
- **Phase 2:** Saturation point. Upon reaching a critical threshold, resource contention becomes a critical factor. If virtual machines with critical dependencies (A and B) are consolidated, their peak demands will coincide. For example, a high-volume request generates high CPU demand on the frontend (A) and high I/O demand on the backend (B).
- **Phase 3:** Scalability Limit. At this point, the host is overloaded. The hypervisor cannot meet both synchronous demands without increasing latency. Overall host performance stagnates or even declines, making any further increase in consolidation density counterproductive. At this point, scalability has failed.

Therefore, the scalability limit is the point of maximum density where critical latency (e.g., database response time) remains below an acceptable threshold.

2.1 Dependencies and unpredictability of scalability

The key to scalability failure in consolidation lies in the hidden or unallocated dependencies of workloads. When planning for scale-out, it can be assumed that workloads are additive. For example, if the host can handle 100 workload units and each virtual machine consumes 10, it could scale up to 10 virtual machines.

However, due to dependencies, the effective load of synchronously consumed resources can be much greater than the simple sum. If interdependent virtual machines (A, B, C) are consolidated, their peak demand aligns. If the peak synchronous load exceeds the host's capacity, latency spikes occur for all virtual machines on that host, even those without direct dependencies on the failed transaction flow. This creates a single point of performance failure on the consolidated host, drastically limiting safe scale-out.

2.2 Scalability Through Dependency Mapping

To enable efficient scalability, consolidation must incorporate dependency awareness through affinity and anti-affinity placement rules, a mechanism that has proven effective across diverse virtualized deployment scenarios [6]. Anti-affinity rules are

crucial for scalability. They involve forcing virtual machines with critical dependencies (e.g., the web server and its database) to reside on different physical servers. This ensures that a peak application load does not synchronously overwhelm the resources of a single server. This technique mitigates contention and allows for more predictable scale-out. On the other hand, affinity rules can be used to group virtual machines that communicate intensively but have inverse or complementary resource usage patterns (e.g., one with high I/O consumption and another with low CPU consumption). This reduces virtual network latency but must be implemented carefully to avoid violating the principle of synchronous contention. As for vertical scalability (scale-up), it also has its limitations. As servers become more resource-intensive, managing cache consistency and non-uniform memory access (NUMA) becomes more complex for the hypervisor. The overhead required to coordinate access to distributed resources on the same physical server can increase, impacting latency.

The exponential growth in resource management complexity limits the ability to acquire more hardware, ultimately forcing organizations to rely on horizontal scalability (i.e., adding more servers to the cluster). This requires a distributed, cloud-native application architecture that many legacy applications lack.

3 Practical Modelling Case Study

Defining and creating a representative workload is the most critical phase of server consolidation, and often the most underestimated. When dependencies are introduced between service components, the challenge shifts from a simple resource measurement to a complex systems engineering problem.

Accurately modelling the workload in a consolidated virtualized environment, where the primary risk is synchronous saturation (multiple key components consuming resources simultaneously on the same server), requires methodologies that go beyond isolated synthetic performance tests [7].

3.1 Limitations of Isolated Synthetic Benchmarks

Performance benchmarks like the SPEC or TPC suites (SPECint, SPECfp, TPC-C, TPC-E, etc.) are excellent for measuring the performance of a physical server or an isolated virtual machine. The main problem is that these benchmarks are designed to test components in isolation, assume independent workloads, and focus on peak performance.

When two virtual machines, VM-A (a web server running SPECint) and VM-B (a database running TPC-C), are consolidated on the same host, the benchmarks fail to capture the interaction between the different system components. Regarding synchronous contention, the peak demand of SPECint on VM-A will likely coincide with the peak demand of TPC-C on VM-B (since the web layer generates the database load), resulting in extreme contention for CPU cycles and disk I/O. Furthermore, the overall performance of the application depends on latency (response time), not just throughput. Even a small degradation in the response time of VM-B due to contention on the host can cause a significant delay in VM-A.

3.2 Representative Load Methodology with Dependencies

An end-to-end workflow that activates the entire chain of resource dependencies follows these phases.

Phase 1: Transaction Flow Mapping
The process begins by mapping the application topology and identifying the critical transaction flow. First, the load pattern is identified using monitoring tools and network traces to determine how, for example, a user request travels from the presentation layer (VM1), through the business logic (VM2), to the cache service (VM3), and finally to the database (VM4). Next, peak resource usage (CPU, I/O) is measured on each VM over time. This reveals whether dependencies cause competing resource consumption spikes.

Phase 2: Integrated Flow Benchmarking
Instead of running SPECint and TPC-C separately, benchmarking is used to simulate real-world interaction. For real-world load simulation, load testing scripts are used that mimic the behaviour of hundreds or thousands of concurrent users performing actual transactions (login, search, payment). To replicate the consolidated environment, a test environment is set up where virtual machines with critical dependencies (e.g., the web server and its database) are intentionally placed on the same virtualized host. End-to-end transaction latency is then measured as the density of virtual machines on the test host increases. The goal is to identify the precise point where the latency of internal requests (between virtual machines) peaks due to CPU or shared I/O contention, thus defining the safe consolidation threshold.

Phase 3: The Hidden Dependency

The most subtle and dangerous dependency in consolidation is storage I/O. Virtually all consolidated environments use centralized storage (SAN, NAS, software-defined storage). However, the problem with I/O sharing lies in the fact that even if one application (VM-A) consumes a lot of CPU and another (VM-B) consumes a lot of memory, both depend on the same I/O path for their disk operations (reading/writing records, data, etc.). If VM-A's I/O pattern is sequential and VM-B's is random, and both VMs saturate the storage host with requests, the latency of all VMs connected to that storage will skyrocket.

A representative workload should include disk I/O stress tests with mixed patterns (read/write, sequential/random, block size) that simulate the aggregate behaviour of the consolidated set of VMs, since this shared I/O dependency is the most frequent and difficult-to-mitigate scalability constraint.

4 Conclusion

Server consolidation through virtualization is a powerful tool, but the pursuit of maximum virtual machine density ignores resource contention and scalability limits. To scale robustly, the strategy must evolve from simply reducing costs to managing latency and performance.

In a real environment, to identify dependencies, it is necessary to map application dependencies and use VM placement algorithms to group those that critically depend on each other onto separate servers or, conversely, onto the same server only if their load pattern has been shown to be neither synchronous nor competitive.

Performance should be modelled not only as an average but as a probability distribution (latency percentiles), ensuring that even in the worst-case scenario, the server has 95% or 99% capacity to scale without falling into a "performance black hole."

In summary, this perspective reaffirms that scalability in consolidated environments is fundamentally a systems engineering challenge. Addressing it requires moving beyond hardware-centric thinking toward structured methodologies for dependency mapping, representative workload modelling, and intelligent VM placement.

Scalability in consolidated environments is directly proportional to the organization's ability to model and manage workload dependencies. Ignoring the risk of synchronous contention reduces the safe density limit to a conservative level, undermining the consolidation efficiency goal. A scalable strategy requires that the horizontal scaling process be based on rules-based virtual machine placement dynamics, due to the representative load of full transaction flows.

As future work stemming from this paper, we propose developing advanced benchmarking and systems engineering methodologies that go beyond isolated synthetic tests, focusing on designing intelligent orchestration and scheduling algorithms that apply dynamic affinity and anti-affinity rules to proactively manage virtual machine placement.

Data Availability Statement

Not applicable.

Funding

This work was supported without any funding.

Conflicts of Interest

Belen Bermejo served as an Associate Editor, and Carlos Juiz served as an Editor-in-Chief of the *Journal of Systems Scalability* at the time of manuscript submission. To ensure the integrity of the peer-review process, neither Belen Bermejo nor Carlos Juiz was involved in the editorial handling, peer review, or decision-making process for this manuscript, which was handled independently by another editor. The remaining authors declare no conflicts of interest.

AI Use Statement

The authors declare that no generative AI was used in the preparation of this manuscript.

Ethical Approval and Consent to Participate

Not applicable.

References

[1] Lin, W., Xiong, C., Wu, W., Shi, F., Li, K., & Xu, M. (2023). Performance interference of virtual machines: A survey. *ACM Computing Surveys, 55*(12), 1-37. [CrossRef]

[2] Dias, A. H., Correia, L. H., & Malheiros, N. (2021). A systematic literature review on virtual machine consolidation. *ACM Computing Surveys (CSUR)*, *54*(8), 1-38. [CrossRef]

[3] Bermejo, B., & Juiz, C. (2020). Virtual machine consolidation: A systematic review of its overhead influencing factors. *The Journal of Supercomputing*, *76*(1), 324–361. [CrossRef]

[4] Bermejo, B., & Juiz, C. (2022). A general method for evaluating the overhead when consolidating servers: performance degradation in virtual machines and containers. The Journal of Supercomputing, 78(9), 11345-11372. [CrossRef]

[5] Lehrig, S., Eikerling, H., & Becker, S. (2015, May). Scalability, elasticity, and efficiency in cloud computing: A systematic literature review of definitions and metrics. In *Proceedings of the 11th international ACM SIGSOFT conference on quality of software architectures* (pp. 83-92). [CrossRef]

[6] Deng, Q., Zhang, S., Yang, X., Zuo, Q., Wang, Z., & Long, S. (2024, December). MO-DDPG: an affinity and anti-affinity-based container service migration strategy in MEC. In *2024 IEEE International Conference on High Performance Computing and Communications (HPCC)* (pp. 618-626). IEEE. [CrossRef]

[7] Molero, X., Juiz, C., & Rodeño, M. J. (2025). Evaluación y modelado del rendimiento de los sistemas informáticos. *arXiv preprint arXiv:2508.16996*. [CrossRef]

[8] Fernandes, F., Beserra, D., Moreno, E. D., Schulze, B., & Pinto, R. C. G. (2016). A virtual machine scheduler based on CPU and I/O-bound features for energy-aware in high performance computing clouds. *Computers & Electrical Engineering*, *56*, 854-870. [CrossRef]

Belen Bermejo received her Ph.D. in Computer Science in 2020 and is a member of the ACSIC research group. Her research focuses on improving the trade-off between energy consumption and performance in virtualized servers and cloud computing systems, as well as on the energy impact of surveillance capitalism. She has carried out pre-doctoral and post-doctoral research stays at Newcastle University and the University of Seville. She has authored several publications in JCR-indexed journals, international conferences, and book chapters. (Email: belen.bermejo@uib.es)

Carlos Juiz is a Full Professor of Computer Science and holds a Ph.D. in Computer Science from the University of the Balearic Islands, Spain. His research focuses on performance engineering, Green IT, and IT governance, leading the ACSIC research group. He has served as a visiting researcher at the University of Vienna (2003) and a visiting associate professor at Stanford University (2011). He has supervised nine Ph.D. dissertations and authored over 250 international scientific publications and books, and is a Senior Member of IEEE and ACM. Currently, is a visiting professor at Keio University. (Email: cjuiz@uib.es)

David Cortes received his undergraduate degree from the National University of Colombia and an M.Sc. in Intelligent Systems from the University of the Balearic Islands. He is a Ph.D. candidate in Information Technologies at the University of the Balearic Islands. His research focuses on Artificial Intelligence, with emphasis on convolutional neural networks for image and medical image processing. His interests include AR/VR, efficient training, scalability, and energy-efficient computing. (Email: dcortes@uib.es)