RAG on the Decentralized Web: An Empirical Study of Akash and Golem

Suting Chen
Northwestern University
Evanston, United States
suting.chen@u.northwestern.edu

Matteo Varvello
Nokia Bell Labs
Murray Hill, NJ, United States
varvello@gmail.com

Aleksandar Kuzmanovic
Northwestern University
Evanston, United States
akuzma@northwestern.edu

Yunming Xiao
The Chinese University of Hong Kong, Shenzhen
China
yunmingxiao@cuhk.edu.cn

Abstract

Decentralized compute markets claim to be an alternative to the cloud, yet the systems community lacks hard evidence on whether they can run modern workloads end to end. We put this claim to the test by deploying a Retrieval-Augmented Generation (RAG) pipeline on the Akash and Golem networks, two popular decentralized computing markets. Our analysis reveals a striking paradox: supply is abundant, with over 10,000 CPU cores and 50 TiB of memory available, yet utilization remains minimal (about 0.3% on Akash). When mapped carefully, several RAG pipeline stages execute correctly and up to 3× cheaper than AWS; others degrade under node churn and network bottlenecks. We also observe that decentralization in practice is already hybrid, with centralized gateways and offchain services masking blockchain complexity. Taken together, our findings suggest that decentralized markets are neither hype nor drop-in cloud replacements, but an underutilized substrate whose viability depends on better scheduling, reliability mechanisms, and trust primitives.

CCS Concepts

• **Networks** → **Network measurement**; **Network services**; • **Information systems** → **World Wide Web**.

Keywords

Decentralized Applications; Retrieval-Augmented Generation

ACM Reference Format:
Suting Chen, Matteo Varvello, Aleksandar Kuzmanovic, and Yunming Xiao. 2026. RAG on the Decentralized Web: An Empirical Study of Akash and Golem. In *The 10th Asia-Pacific Workshop on Networking (APNet '26), August 6–7, 2026, Singapore, Singapore.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3820441.3820466

1 Introduction

Modern cloud computing is built on a simple assumption: serious applications require expensive, centralized datacenters. In this paper, we question this assumption. Blockchains quietly power a parallel universe of decentralized applications offering compute [1, 9, 13, 16, 31, 40], storage [10, 15, 17, 39, 42], and networking [23, 34, 47]. These platforms promise cryptographic trust, permissionless participation, and market-driven pricing. Their narrative is appealing, yet it is unclear if they can actually replace traditional cloud deployments for modern workloads.

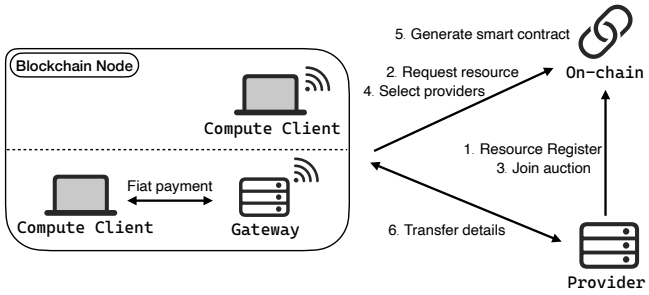
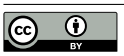


Figure 1: DApp network architecture, client could manage their wallets or use a centralized gateway.

To investigate this question, we deploy a RAG (Retrieval-Augmented Generation) [11] pipeline on two major decentralized compute markets: Akash [1] and Golem [13]. RAG is a complex and realistic workload which stresses GPU/CPU, memory, networking, and latency in different ways. It represents modern AI-backed services; if dApps can support RAG competitively, the implications are broader than Web3 enthusiasm. We measure cost, performance, reliability, and operational friction. Our findings are as follows:

Massive Dormant Capacity. Decentralized compute markets expose tens of thousands of CPU cores and over 50 TiB of memory that sit mostly idle. On Akash, daily lease volume is roughly $5,000, translating to a tiny fraction of overall resource utilization. The supply exists, the demand does not. This mismatch reveals an opportunity: substantial unused global capacity that can be tapped at competitive prices. It also exposes real barriers such as blockchain-native interfaces, payment friction, and uncertainty in node availability.

Workload Sensitivity Is the Real Bottleneck. RAG is not monolithic. Preprocessing stresses parsing and I/O. Embedding generation is compute intensive. Vector retrieval stresses synchronization and distributed indexing. Inference is latency critical. Decentralized markets handle these stages unevenly: some phases scale surprisingly well across heterogeneous providers, while others suffer from



APNet '26, Singapore, Singapore

ACM ISBN 979-8-4007-2664-4/26/08
https://doi.org/10.1145/3820441.3820466

stragglers, churn, or network variability. The key insight is not whether dApps are slow or fast, but where they break and why.

Decentralization Is Already Hybrid. Leading platforms increasingly rely on centralized gateways, off-chain services, and developer-friendly APIs that abstract blockchain mechanics. In practice, decentralized infrastructure is converging toward a hybrid model that combine market-based resource allocation with conventional edge and cloud components. This evolution suggests that the future is not ideological decentralization, but pragmatic integration.

This paper does not advocate abandoning the cloud tomorrow. Instead, it provides the first systematic evidence that decentralized markets are no longer academic curiosities. For certain workloads and cost envelopes, they can compete with centralized datacenters. However, challenges lie ahead for broad adoption. First, the ecosystem must discover a sustainable evolution path: today's markets expose abundant supply but attract limited demand, leaving large amounts of capacity idle. Second, reliability remains a concern, as decentralized environments lack mature fault-tolerant orchestration to handle node churn and heterogeneous performance. Third, current platforms provide little scheduling or placement intelligence, making it difficult to efficiently deploy distributed workloads across diverse providers. Finally, trust remains an open issue, since practical mechanisms for verifying remote execution and resource integrity are still largely missing.

2 Background and Related Work

DApps Architectures. Blockchain-based dApps enable decentralized marketplaces for storage, computation, and bandwidth through smart contracts that coordinate resources, enforce agreements, and manage payments. Storage systems (e.g., Filecoin, Sia) rely on cryptographic integrity proofs [3]; Nardini et al. propose a hybrid compute marketplace with off-chain execution and on-chain verification [24]; dVPN services (e.g., Mysterium, Sentinel, Tachyon) monetize bandwidth via smart contracts [47]. Most dAPP compute networks (e.g., Akash and Golem) separates on-chain coordination from off-chain execution (see Figure 1). *Providers* advertise resources, *clients* submit job requests, and matching is enforced through smart contracts, often via escrowed payments. Tasks run off-chain, typically in containerized environments. To improve usability, developers often build centralized platforms and managed wallets that lower the deployment barrier.

Prior Measurements. Trautwein et al. [44] evaluate IPFS at scale and report favorable publication and retrieval performance. Lajam and Helmy [20] show IPFS incurs higher latency and CPU overhead in private networks compared to FTP. Other studies highlight significant content centralization, with most data hosted by a small set of cloud peers, suggesting room for optimization [37, 38]. Decentralized bandwidth marketplaces have also been surveyed [4, 47]. To the best of our knowledge, prior work did not evaluate usability in decentralized compute networks. We fill this gap by monitoring the scale and activity of the Akash and Golem networks, and benchmarking their real-world usability.

Retrieval-Augmented Generation (RAG). Large language models [5] are often limited by outdated knowledge, hallucinations, and limited grounding. RAG [11] mitigates these issues with external, up-to-date information, extending a model's effective knowledge and improving reliability for response generation. This multi-stage pipeline increases synchronization overhead, complexity, and infrastructure cost.

Recent work has improved retrieval scalability [36], vector database efficiency [45], and end-to-end optimization through caching and adaptive strategies [41]. With growing commercial adoption [2, 6, 25], there is increasing demand for scalable, cost-efficient infrastructure. While most deployments rely on centralized clouds, decentralized compute platforms offer a potential alternative for reducing costs.

Platform	Payment	Comment
Akash Network [1]	$AKT	
Golem Network [13]	$GLM	
Render Network [31]	$RENDER	Focus on rendering
GPU.Net [16]	$GPU	GPU-focused
Spheron Network [40]	$SPON	GPU-focused
Gensyn	-	Under development

Table 1: Overview of compute dApps.

3 DAPP Networks Measurements

Before answering whether compute dApps can truly replace traditional cloud deployments for modern workloads, we first take a step back and characterize what exists today. We measure popular platforms through passive data collection and active experiments to understand price, hardware availability, and delivered performance. This serves two goals: build the measurement tools and primitives needed for the next step, namely deploying a RAG workload, and perform a basic feasibility check. If capacity is limited or unreliable, there is little point in going further. To the best of our knowledge, no prior work has directly benchmarked Akash and Golem, the two most established compute dApps we study.

3.1 Compute dApp Overview

Table 1 provides an overview of six blockchain-enabled computing dApps. Among them, Render Network is a specialized platform focused on 3D rendering tasks [31]. GPU.Net and Spheron Network provide a GPU marketplace [16, 40], but they lack general-purpose computation. Gensyn was still in testnet during our measurement period [9].

As a result, we select two viable, representative platforms: Akash [1] and Golem [13]. Both platforms support familiar developer workflows. Clients submit computing jobs as Docker [7] containers with VM configurations (e.g., CPU cores, RAM, disk size), which are then assigned to providers and executed with platform-specific limitations.

3.2 Measurement Methodology

We collect publicly available data without directly interacting with providers. In addition to on-chain records, most networks offer dashboards or APIs that report real-time statistics on available resources and active deployments, giving a high-level view of overall capacity and utilization. The data monitored includes online nodes, available resources and the number of on-chain transactions.

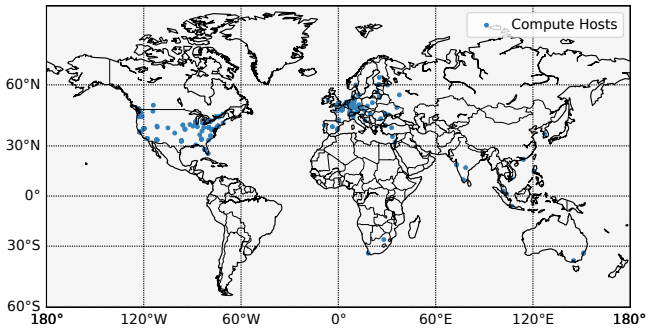


Figure 2: Footprint of compute host location.

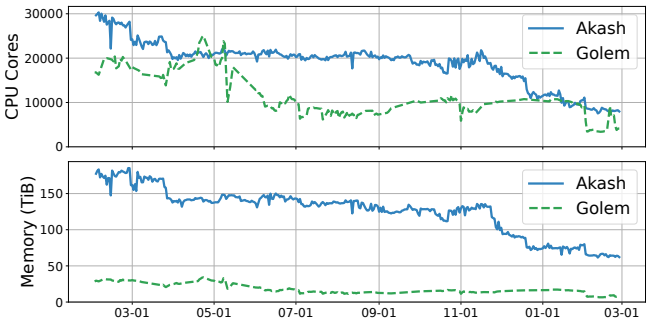


Figure 3: Total CPU (cores) and memory (TiB) availability on Akash and Golem networks (02/2025–03/2026).

3.3 Network Capacity Overview

Figure 2 shows the global footprint of compute host locations. The majority of hosts are located in North America and Europe, with a smaller presence in Asia, while the rest of the world has limited availability. This geographic imbalance suggests potential disparities in latency, accessibility, and cost across regions.

Figure 3 shows the available compute resources on Akash and Golem between February 2025 and March 2026. During the thirteen-month observation period, Golem's resource supply fluctuates noticeably over time, with several distinct episodes where the total available resources surge or drop sharply within a short window. The Akash network shows overall more stable resource availability between April and December 2025, with a constant 20K CPU cores. Note that a major software update was gradually applied in November 2025, causing outdated nodes to be slowly dropped out. This is reflected in a 50% resource drop between December 2025 and March 2026. This observation is consistent with the network utilization rate shown in Figure 5(b). After the upgrade, utilization roughly doubles compared to the pre-upgrade period, indicating that the network experiences no major disruption. Instead, the existing load is effectively absorbed by the available, up-to-date nodes.

To understand how computational power is distributed across the Akash and Golem networks, we collect metadata from active providers and analyze their advertised hardware configurations. Figure 5(a) plots the Cumulative Distribution Function (CDF) of CPU cores per provider and network. We observe that most providers on Akash offer relatively high-end machines, with the distribution peaking at 32 cores, while Golem's resource base consists primarily of smaller nodes (typically 4 cores).

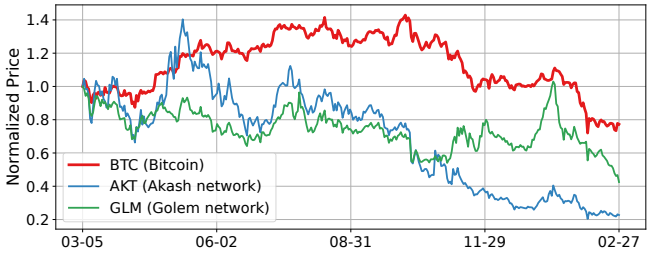


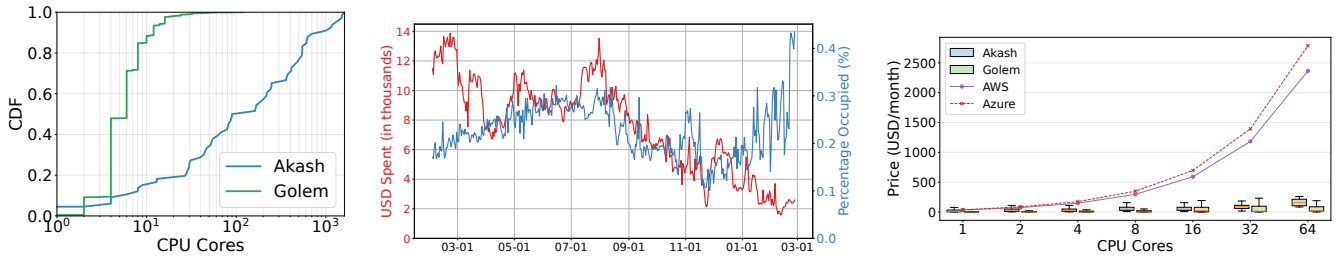
Figure 4: Normalized prices of cryptocurrencies during our measurement period (02/2025–03/2026).

Next, we examine how much of the available capacity is actively utilized. Figure 5(b) shows the total daily value of leases signed on the Akash network, denominated in USD for simplicity. On average, Akash network processes around $5,000 in lease payments per day after the upgrade, with an average CPU core utilization rate of approximately 0.30%. In contrast, the Golem network appears to struggle with resource utilization. We frequently observed entire weeks with total transaction volumes below 20 GLM [14], less than $5 at the beginning of our study. One year later, weekly volume reached only about 1,000 GLM, roughly $135. These figures are negligible when compared to the available capacity. Golem has declared a new partnership with *salad.com* [32], aiming to migrate significant Web2 workloads onto its infrastructure [8]. This integration is expected to drive an increase in demand for Golem's compute resources, as Salad has a large user base and offers a wide range of applications that could benefit from decentralized compute. However, because this integration was still in engineering test phase during our data collection period, the resulting traffic did not yet represent a stabilized market state. Consequently, Golem is omitted from Figure 5(b).

3.4 Pricing Analysis

Figure 5(c) compares the monthly price of compute resources across different platforms, when considering renting different number of CPU cores per VM. We observe that Akash and Golem offer significantly lower prices, approximately 10x cheaper, on average, than centralized cloud providers like AWS [35] and Azure [21]. Further, the price growth is overall sublinear for both Akash and Golem, making them particularly attractive for cost-sensitive workloads that need significant compute power. Golem prices sometimes drop to very low levels, near zero in some cases. We assume this reflects the low overall demand on the network, as discussed above. In such conditions, providers may lower prices to attract workloads and remain active. Indeed, they must maintain high uptime to preserve their reputation and successfully complete leases to accumulate credit within Golem's reputation system [12].

To understand how cryptocurrency price trends influence compute pricing, Figure 4 plots the token values of BTC, AKT and GLM relative to USD over our measurement period; prices are normalized to their value on the first day of observation to highlight relative changes. In Akash, where demand is relatively stable (see Figure 5(b)), providers often adjust their cryptocurrency prices in response to fiat exchange rate fluctuations. Out of 213 providers we randomly tracked, 115 (53.99%) adjust their token rates to maintain a consistent price in USD, while 63 (29.58%) appear to target a stable price in EUR. This suggests that a major portion of providers (84%



(a) CDF of CPU cores per provider. (b) Akash network utilization (02/2025-03/2026). (c) Comparison of price by CPU cores.

Figure 5: Observation of compute dAPPs.

overall) price their services with fiat stability in mind, avoiding exposing users to cryptocurrency volatility.

Takeaway: *Our evaluations reveal a paradox: despite competitive pricing and abundant resources, distributed computing platforms experience low usage: 0.30% CPU utilization for Akash and Golem processing under $20 in daily transactions.*

4 RAG as a Case Study

While our measurements quantify the performance and economics of decentralized compute platforms in isolation, a natural next question is how these capabilities translate to real-world workloads. To explore this, we consider a representative modern application that integrates diverse compute and latency requirements, namely Retrieval Augmented Generation (RAG). RAG is a key architecture for providing large language models (LLMs) with external and up-to-date knowledge. It is particularly suitable for this analysis because a complete RAG pipeline consists of several distinct stages, including vectorizing, storing embeddings, and performing inference, each with unique resource and performance demands.

Although RAG workloads typically benefit from GPU acceleration, current dApp platforms either offer GPUs in scarce supply or restrict them to non-general-purpose tasks (see Table 1); we will monitor these marketplaces and incorporate GPU-based deployments as viable options emerge. Accordingly, we deploy lightweight Gemma 3 4B IT for generation and Gemma 3 300M for embeddings [33, 43]. Both use 4-bit Quantization-Aware Training and are deployed in GGUF format.

4.1 Pipeline Architecture and Workload Characterization

As shown in Figure 6, a typical RAG pipeline consists of two computation-intense stages: *embedding and store*, and *retrieval with inference*.

Embedding and Store. Large blocks of text are cleaned and chunked, then converted into high-dimensional vectors via *text embedding* — a process that encodes semantic meaning into a numerical representation suitable for similarity search. These vectors, alongside their original text chunks, are stored in a *vector database* optimized for efficient indexing and nearest-neighbor retrieval. The vector database must be reachable over a fast and reliable network, and the embedding step itself is compute-intensive, thus benefitting from parallel execution.

System	Cost (per GB / mo)	Notes
Pinecone [27]	$0.33	-
Weaviate [46]	$9.5 per 100M dimensions	$25 min.
Milvus on Zilliz [22, 49]	$0.33	-
Qdrant [29]	$0.25–$0.35	1 GB free
pgvector [26]	$0 (Postgres Extension)	

Table 2: Vector database services and pricing.

Retrieval and Inference. At query time, the system retrieves the most semantically relevant chunks from the vector database and combines them with the user's query as input to the LLM. As this pipeline executes on the critical path of every user request, end-to-end latency is a first-order constraint.

We implement the above pipeline in a disaggregated manner, with data flowing asynchronously across components. This allows stages to execute in parallel, enabling flexible scheduling and deployment across heterogeneous resources. Importantly, this design does not represent a performance compromise but rather reflects common practice in modern AI serving systems. For example, prefill–decode (PD) disaggregation has been widely adopted in LLM serving to separate compute-intensive and latency-sensitive stages across different machines [19, 48].

The phases of a RAG pipeline have diverse resource and performance requirements. The embedding and indexing phases are largely compute-intensive and highly parallelizable, with moderate latency sensitivity but strong dependence on processing throughput and stable networking for vector database operations. In contrast, retrieval and inference are user-facing and latency-critical, where end-to-end response time directly affects user experience. Disaggregation therefore enables each stage to be provisioned and scheduled according to its specific resource and latency requirements, improving overall efficiency and scalability [18].

Such a disaggregated architecture is naturally well suited for decentralized environments, where tasks with different resource demands can be placed on nodes with matching capabilities. This enables efficient utilization of heterogeneous resources while accommodating the diverse performance requirements of the pipeline. These properties make RAG an ideal workload for studying the performance of decentralized compute networks across a range of tasks.

4.2 Design and Implementation

We containerize all components in the RAG pipeline with Docker [7], so they can seamlessly be deployed across dApp nodes and cloud

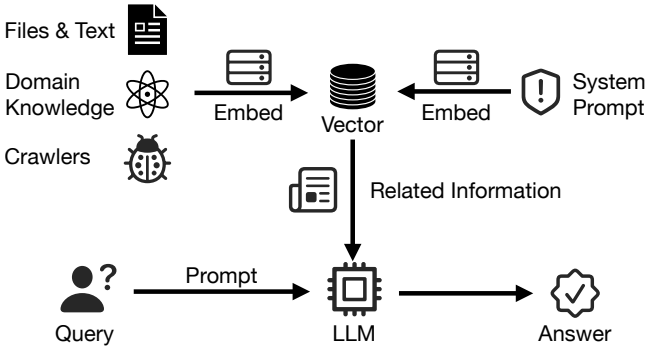


Figure 6: Illustration of typical RAG pipeline.

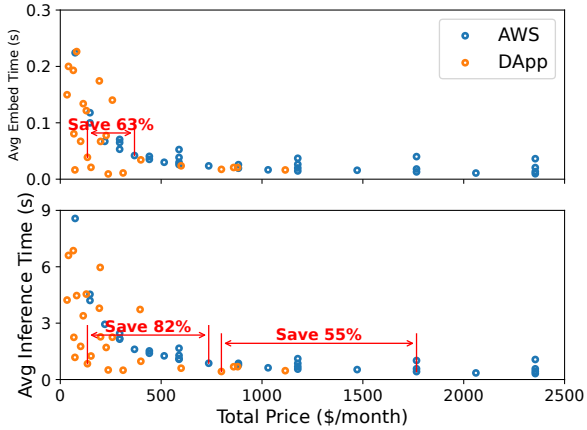


Figure 7: Comparison of costs and performance for RAG pipeline stages.

servers. We adopt Qdrant [28] as the centralized vector database because it provides both an open-source version and a competitively priced managed option, making it one of the most cost-effective choices to start with. A comparison of representative vector storage services and their indicative pricing is shown in Table 2.

4.3 Experiments and Analysis

To evaluate the practicality of our distributed RAG implementation, we measure both performance and cost across decentralized compute networks (Akash and Golem) and a traditional cloud environment (AWS). Performance metrics focus on the time taken for each pipeline stage under a designated workload.

In total, our evaluation collects over 2 million fine-grained measurement records, including detailed per-stage time breakdowns (e.g., embedding, upsert, search, chat, and network overhead) across all configurations. The experiment includes: *embedding* stage, processes 10,000 text documents into vector representations; *inference* stage, handles 1,000 randomly picked queries from the SQuAD2.0 dataset [30].

For the cloud baseline, we deploy the system on AWS [35]. We evaluate a range of general-purpose instance families (m7a, m7i, m8a, and m8i) with sizes from large (2 vCPUs) up to 16xlarge (64 vCPUs). For each instance type, we scale the number of replicas so that the total allocated cores do not exceed 64, with at most eight replicas per configuration. This setup allows us to compare different CPU generations and scaling strategies under a fixed total core budget, enabling a fair analysis of performance and cost trade-offs across instance types. For the vector database, we deploy the open-source version of Qdrant on AWS with sufficient network capacity (5 Gbps) for predictable performance and cost.

For the distributed networks, experiments are conducted on Akash and Golem. We publish resource requests ranging from 2-core to 32-core, and scale deployments to up to 8 replicas per configuration. These settings are chosen to achieve performance comparable to the most cost-efficient cloud configurations described above. Each pipeline component is packaged as a Docker container and deployed using the same container images across all platforms. To minimize bias, we deliberately select different hosts for each run on both Akash and Golem, regardless of whether the host IP belongs to a datacenter or a residential provider, CPU model, or geographical location. We rely on the same setup as the cloud baseline for the vector database (Qdrant), to ensure stable storage performance and focus the experiment on computation behavior.

Results. For decentralized deployments, we sampled 50 quotes for equivalent resources to capture price variability. Each experiment is repeated three times; reported performance metrics are averaged across repetitions, while cost figures reflect market price variations across providers. Figure 7 shows that the performance variation of dApp nodes is much higher than traditional cloud servers, with up to 4x difference in execution time given the same price. However, with careful host selection, embedding tasks on dApp nodes can run up to 20% faster than cloud servers at the same price. For inference tasks, the cost-efficiency advantage is even clearer: a highlighted dApp node achieves similar execution time while costing $134.51, whereas the cheapest AWS configuration with comparable performance costs $735.85, resulting in an 81.7% cost saving.

The total time and breakdown of each stage are shown in Figure 8, with the x-axis representing the number of CPU cores, and price attached to the top of each bar. The number of cores in the dApp figure is picked as comparison with similar performance. The results indicate that dApp nodes often need 4x more CPU cores to achieve the same computation performance (reflected as blue parts of the bars) as traditional cloud servers. Despite that, the cost of dApp nodes is significantly lower, up to one-third in some inference experiments. This suggests that while dApp nodes may be less efficient in terms of single-core raw performance, their cost-effectiveness can make them attractive for workloads that could be parallelized across more cores.

A further analysis of figure 8 reveals that networking in dApp nodes is a significant bottleneck, resulting in higher percentage of upsert and search time (reflected as orange). In our experiment, some lightweight embedding experiment have shown more than 40% of the total time spent on network operations in embedding tasks, compared to less than 10% in traditional cloud servers.

Takeaway: *Our case study shows that dApp nodes can be cost-effective for parallelizable, computation-intensive tasks like embedding and inferencing, where lower prices can offset weaker per-core performance. However, network bottlenecks observed during vector upserts in the embedding stage highlight the importance of stable*

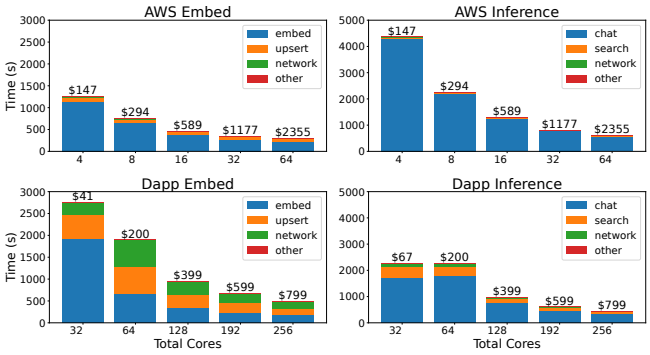


Figure 8: Pricing comparison of dApp and traditional cloud.

connection in traditional cloud infrastructure. In practice, a workload-aware hybrid approach is likely the most effective way to balance cost and performance.

5 Conclusion & Future Work

This work presents the first empirical evaluation of Retrieval-Augmented Generation (RAG) workloads on decentralized compute platforms, focusing on Akash and Golem. Our results highlight both the promise and current limitations of deploying AI pipelines on blockchain-based infrastructure. Despite a large resource pool—over 10,000 CPU cores and 50 TiB of memory—utilization remains extremely low. For example, Akash processes roughly $5,000 in daily leases while operating at only 0.30% CPU utilization, reflecting technical friction and limited ecosystem maturity.

For developers, the main advantage lies in cost. DApp offer compute at roughly 10× lower prices than centralized clouds such as AWS for the same number of CPU cores. While single-core performance is weaker, this price gap favors throughput-oriented workloads. In our RAG case study, parallelizable stages such as embedding achieve 3× cost reductions, whereas latency-sensitive inference remains constrained by network variability. Although networking overheads remain a challenge, workload-aware or hybrid deployment strategies suggest a practical path for integrating decentralized compute into cost-sensitive AI systems.

In future work, we plan to explore several directions toward making decentralized compute markets more practical. First, we will investigate mechanisms to stimulate sustainable demand and improve developer accessibility. Second, we plan to study fault-tolerant orchestration techniques that can improve reliability under node churn and heterogeneous performance. Third, we aim to design more intelligent scheduling and placement strategies for distributed workloads across diverse providers. Finally, we will explore practical trust and verification mechanisms to ensure correct remote execution and resource integrity. Addressing these challenges will help decentralized infrastructure evolve into a reliable complement to conventional cloud platforms.

Acknowledgments

We sincerely thank the anonymous reviewers for their constructive feedback. This paper was funded in part by the US NSF awards CNS-2226107 and CNS-2310927. Yunming Xiao is the corresponding author.

References

[1] Akash Network. 2026. Akash Network - Decentralized Compute Marketplace. https://akash.network/ Accessed May 13, 2026.

[2] Amazon Web Services. 2025. Fully managed Retrieval Augmented Generation options on AWS. https://docs.aws.amazon.com/prescriptive-guidance/latest/retrieval-augmented-generation-options/rag-fully-managed.html. AWS Prescriptive Guidance; Accessed: 2025-10-06.

[3] Nazanin Zahed Benisi, Mehdi Aminian, and Bahman Javadi. 2020. Blockchain-based decentralized storage networks: A survey. *J. Netw. Comput. Appl.* 162 (2020), 102656. doi:10.1016/J.JNCA.2020.102656

[4] Maurantonio Caprolu, Simone Raponi, and Roberto Di Pietro. 2024. Sharing is (s) caring: Security and privacy issues in decentralized physical infrastructure networks (depin). In *International Conference on Network and System Security*. Springer, 301–318.

[5] Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. 2024. A survey on evaluation of large language models. *ACM transactions on intelligent systems and technology* 15, 3 (2024), 1–45.

[6] Cloudflare. 2025. Cloudflare AI Search Documentation. https://developers.cloudflare.com/ai-search/. Accessed: 2025-10-06.

[7] Docker Inc. 2026. Docker. https://www.docker.com/ Accessed: 2026.

[8] Kyle Dodson. 2026. *Golem Network: An Opportunity to Improve Salad's Tech Stack.* SaladCloud Blog. https://blog.salad.com/golem-network-an-opportunity-to-improve-salads-tech-stack/

[9] Ben Fielding and Harry Grieve. 2026. Gensyn: Making compute for AI as accessible as oxygen for humans. https://www.gensyn.ai/ Last accessed May 13, 2026.

[10] Filecoin. 2026. Filecoin. https://filecoin.io/ Last accessed May 14, 2026.

[11] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yixin Dai, Jiawei Sun, Haofen Wang, Haofen Wang, et al. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997* 2, 1 (2023), 32.

[12] Golem Factory GmbH. 2025. Golem Network Reputation System - Provider Performance and. https://docs.golem.network/docs/reputation#golem-network-reputation-system Accessed May 14, 2025.

[13] Golem Network. 2026. Golem Network: Decentralized Computing for Everyone. https://www.golem.network/ Last accessed May 13, 2026.

[14] Golem Network. 2026. Golem Network Statistics and Metrics. https://stats.golem.network/ Last accessed May 13, 2026.

[15] Golem Network. 2026. Storj: Making Data Growth a Source for Good. https://www.storj.io/ Last accessed May 13, 2026.

[16] GPUNET Enterprise. 2026. GPUNET Enterprise — GPU Compute for Business. https://www.gpu.net/ Last accessed May 14, 2026.

[17] IPFS. 2026. IPFS: Building blocks for a better web | IPFS. https://ipfs.tech/ Last accessed May 14, 2026.

[18] Wenqi Jiang, Marco Zeller, Roger Waleffe, Torsten Hoefler, and Gustavo Alonso. 2024. Chameleon: a Heterogeneous and Disaggregated Accelerator System for Retrieval-Augmented Language Models. *Proc. VLDB Endow.* 18, 1 (2024), 42–52. doi:10.14778/3696435.3696439

[19] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient Memory Management for Large Language Model Serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles, SOSP 2023, Koblenz, Germany, October 23-26, 2023*, Jason Flinn, Margo I. Seltzer, Peter Druschel, Antoine Kaufmann, and Jonathan Mace (Eds.). ACM, 611–626. doi:10.1145/3600006.3613165

[20] Omar Abdullah Lajam and Tarek Ahmed Helmy. 2021. Performance Evaluation of IPFS in Private Networks. In *DSDE '21: 2021 4th International Conference on Data Storage and Data Engineering, Barcelona, Spain, February 18-20, 2021.* ACM, 77–84. doi:10.1145/3456146.3456159

[21] Microsoft. 2025. Microsoft Azure. https://azure.microsoft.com/en-us/. Accessed: 2025-05-09.

[22] Milvus. 2025. Milvus: High-Performance Vector Database Built for Scale. https://milvus.io. Accessed: 2025-10-06.

[23] Mysterium VPN. 2025. Mysterium VPN: Residential IPs. No Borders. No Limits. https://www.mysteriumvpn.com/. Last accessed: 2025-10-07.

[24] Matteo Nardini, Sven Helmer, Nabil El Ioini, and Claus Pahl. 2020. A Blockchain-based Decentralized Electronic Marketplace for Computing Resources. *SN Computer Science* 1, 5 (2020), 251. doi:10.1007/s42979-020-00243-7 doi:10.1007/s42979-020-00243-7

[25] OpenAI. 2025. Retrieval - OpenAI API. https://platform.openai.com/docs/guides/retrieval. Accessed: 2025-10-06.

[26] pgvector Developers. 2025. pgvector: Open-source vector similarity search for Postgres. https://github.com/pgvector/pgvector. Version 0.8.1; Accessed: 2025-10-06.

[27] Pinecone. 2025. Pinecone: The Vector Database to Build Knowledgeable AI. https://www.pinecone.io/. Accessed: 2025-10-06.

[28] Qdrant. 2025. Qdrant: Open-Source Vector Database and Similarity Search Engine. https://qdrant.tech/ Accessed: 2025-10-07.

[29] Qdrant. 2025. Qdrant: Vector Database - The leading open source vector database. https://qdrant.tech/. Accessed: 2025-10-06.

[30] Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know What You Don't Know: Unanswerable Questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Iryna Gurevych and Yusuke Miyao (Eds.). Association for Computational Linguistics, Melbourne, Australia, 784–789. doi:10.18653/v1/P18-2124

[31] Render Network. 2026. Render Network. https://rendernetwork.com/ Last accessed May 14, 2026.

[32] Salad Technologies. 2024. Salad Cloud: Affordable, Secure, Community Cloud for AI/ML Inference. https://salad.com/ Accessed: 2026.

[33] Henrique* Schechter Vera, Sahil* Dua, Biao Zhang, Daniel Salz, Ryan Mullins, Sindhu Raghuram Panyam, Sara Smoot, Iftekhar Naim, Joe Zou, Feiyang Chen, Daniel Cer, Alice Lisak, Min Choi, Lucas Gonzalez, Omar Sanseviero, Glenn Cameron, Ian Ballantyne, Kat Black, Kaifeng Chen, Weiyi Wang, Zhe Li, Gus Martins, Jinhyuk Lee, Mark Sherwood, Juyeong Ji, Renjie Wu, Jingxiao Zheng, Jyotinder Singh, Abheesht Sharma, Divya Sreepat, Aashi Jain, Adham Elarabawy, AJ Co, Andreas Doumanoglou, Babak Samari, Ben Hora, Brian Potetz, Dahun Kim, Enrique Alfonseca, Fedor Moiseev, Feng Han, Frank Palma Gomez, Gustavo Hernández Ábrego, Hesen Zhang, Hui Hui, Jay Han, Karan Gill, Ke Chen, Koert Chen, Madhuri Shanbhogue, Michael Boratko, Paul Suganthan, Sai Meher Karthik Duddu, Sandeep Mariserla, Setareh Ariafar, Shanfeng Zhang, Shijie Zhang, Simon Baumgartner, Sonam Goenka, Steve Qiu, Tanmaya Dabral, Trevor Walker, Vikram Rao, Waleed Khawaja, Wenlei Zhou, Xiaoqi Ren, Ye Xia, Yichang Chen, Yi-Ting Chen, Zhe Dong, Zhongli Ding, Francesco Visin, Gaël Liu, Jiageng Zhang, Kathleen Kenealy, Michelle Casbon, Ravin Kumar, Thomas Mesnard, Zach Gleicher, Cormac Brick, Olivier Lacombe, Adam Roberts, Yunhsuan Sung, Raphael Hoffmann, Tris Warkentin, Armand Joulin, Tom Duerig, and Mojtaba Seyedhosseini. 2025. EmbeddingGemma: Powerful and Lightweight Text Representations. (2025). https://arxiv.org/abs/2509.20354

[34] Sentinel. 2025. Sentinel: Decentralized Bandwidth Marketplace and dVPN. https://www.sentinel.co/. Accessed: 2025-10-07.

[35] Amazon Web Services. 2025. Cloud Computing Services - Amazon Web Services (AWS). https://aws.amazon.com/. Accessed May 15, 2025.

[36] Rulin Shao, Jacqueline He, Akari Asai, Weijia Shi, Tim Dettmers, Sewon Min, Luke Zettlemoyer, and Pang Wei Koh. 2024. Scaling Retrieval-Based Language Models with a Trillion-Token Datastore. *arXiv preprint arXiv:2407.12854* (2024).

[37] Ruizhe Shi, Ruizhi Cheng, Yuqi Fu, Bo Han, Yue Cheng, and Songqing Chen. 2025. Centralization in the Decentralized Web: Challenges and Opportunities in IPFS Data Management. In *Proceedings of the ACM on Web Conference 2025, WWW 2025, Sydney, NSW, Australia, 28 April 2025- 2 May 2025*. ACM, 4068–4076. doi:10.1145/3696410.3714627

[38] Ruizhe Shi, Ruizhi Cheng, Bo Han, Yue Cheng, and Songqing Chen. 2024. A Closer Look into IPFS: Accessibility, Content, and Performance. *Proc. ACM Meas. Anal. Comput. Syst.* 8, 2 (2024), 20:1–20:31. doi:10.1145/3656015

[39] Sia Cloud Storage Network. 2026. Sia Cloud Storage Network. https://sia.tech/ Last accessed May 13, 2026.

[40] Spheron Network. 2026. Spheron Network: Decentralized cloud application development platform. https://www.spheron.network/ Last accessed May 13, 2026.

[41] Sakhinana Sagar Srinivas, Akash Das, Shivam Gupta, and Venkataramana Runkana. 2025. Scaling Test-Time Inference with Policy-Optimized, Dynamic Retrieval-Augmented Generation via KV Caching and Decoding. arXiv:2504.01281 [cs.LG] https://arxiv.org/abs/2504.01281

[42] Delta Storage. 2025. Decentralized Cloud Storage for AI & Web3. https://www.delta.storage/ Accessed May 14, 2025.

[43] Gemma Team. 2025. Gemma 3. (2025). https://goo.gle/Gemma3Report

[44] Dennis Trautwein, Aravindh Raman, Gareth Tyson, Ignacio Castro, Will Scott, Moritz Schubotz, Bela Gipp, and Yiannis Psaras. 2022. Design and evaluation of IPFS: a storage layer for the decentralized web. In *SIGCOMM '22: ACM SIGCOMM 2022 Conference, Amsterdam, The Netherlands, August 22 - 26, 2022*. ACM, 739–752. doi:10.1145/3544216.3544232

[45] Nitish Upreti, Harsha Vardhan Simhadri, Hari Sudan Sundar, Krishnan Sundaram, Samer Boshra, Balachandar Perumalswamy, Shivam Atri, Martin Chisholm, Revti Raman Singh, Greg Yang, Tamara Hass, Nitesh Dudhey, Subramanyam Pattipaka, Mark Hildebrand, Magdalen Manohar, Jack Moffitt, Haiyang Xu, Naren Datha, Suryansh Gupta, Ravishankar Krishnaswamy, Prashant Gupta, Abhishek Sahu, Hemeswari Varada, Sudhanshu Barthwal, Ritika Mor, James Codella, Shaun Cooper, Kevin Pilch, Simon Moreno, Aayush Kataria, Santosh Kulkarni, Neil Deshpande, Amar Sagare, Dinesh Billa, Zishan Fu, and Vipul Vishal. 2025. Cost-Effective, Low Latency Vector Search with Azure Cosmos DB. *Proc. VLDB Endow.* 18, 12 (Sept. 2025), 5166–5183. doi:10.14778/3750601.3750635

[46] Weaviate. 2025. Weaviate: The AI-native database developers love. https://weaviate.io. Accessed: 2025-10-06.

[47] Yunming Xiao, Matteo Varvello, and Aleksandar Kuzmanovic. 2022. Monetizing Spare Bandwidth: The Case of Distributed VPNs. *Proc. ACM Meas. Anal. Comput. Syst.* 6, 2 (2022), 33:1–33:27. doi:10.1145/3530899

[48] Yinmin Zhong, Shengyu Liu, Junda Chen, Jianbo Hu, Yibo Zhu, Xuanzhe Liu, Xin Jin, and Hao Zhang. 2024. DistServe: Disaggregating Prefill and Decoding for Goodput-optimized Large Language Model Serving. In *18th USENIX Symposium on Operating Systems Design and Implementation, OSDI 2024, Santa Clara, CA, USA, July 10-12, 2024*, Ada Gavrilovska and Douglas B. Terry (Eds.). USENIX Association, 193–210. https://www.usenix.org/conference/osdi24/presentation/zhong-yinmin

[49] Zilliz. 2025. Zilliz: Vector Database built for enterprise-grade AI applications. https://zilliz.com. Accessed: 2025-10-06.