Advances in Computational Systems, Algorithms, and Emerging Technologies (2023), **8**, 1–12

ORIGINALRESEARCH

Efficient Storage Solutions for Big Data in Cloud Environments: A Comparative Study of Scalability, Cost, and Performance

Alejandro Pérez Gómez

Instituto de Tecnologia do Ceará, Faculdade de Engenharia de Computação, Avenida Dom Luís, Aldeota, Fortaleza, Ceará

Abstract

Cloud computing infrastructures have become the de facto standard for hosting massive data repositories, driven by the exponential expansion in data generation across diverse domains. The pressing need to handle this influx of data has motivated new strategies for storage systems that must remain operationally feasible, scalable, and cost-effective. Designing storage mechanisms for big data in cloud environments requires sophisticated techniques to maintain performance guarantees, allow elastic resource allocation, and ensure minimal latency under shifting workloads. Moreover, providers face challenges associated with ensuring high availability, load balancing, fault tolerance, and consistent data integrity. A key factor in formulating these architectures involves reconciling theoretical models with implementation realities, such that overheads remain within acceptable bounds for both batch workloads and interactive, low-latency queries. In this paper, a thorough comparative investigation is presented, focusing on the interplay among scalability, cost efficiency, and performance optimization in modern storage systems. By exploring cutting-edge theoretical frameworks and examining the interplay between mathematical abstractions and hardware-level constraints, this work aims to shed light on the design choices that best match different application demands. Through a careful synthesis of model formulations, computational analysis, and large-scale practical considerations, this study highlights fundamental performance trade-offs while paving the way for robust, future-proof storage solutions.

1. Introduction

The relentless expansion in data creation has motivated intensive efforts to design, analyze, and refine storage solutions that can adapt to increasingly diverse and voluminous data patterns (Yan et al. 2017). In cloud environments, these solutions must address a broad spectrum of operational requirements, including the management of data-intensive scientific computations, real-time analytics for mission-critical applications, and long-term archival for regulatory compliance. On top of these functional demands lies the nontrivial question of cost: cloud providers charge users for the consumption of storage and computational resources, both of which vary substantially depending on factors such as availability region, redundancy settings, and usage patterns (VoPham et al. 2018). Navigating this complex pricing environment compels system architects to adopt scalable and cost-conscious designs that fulfill service-level objectives without sacrificing responsiveness or throughput.

Underlying the practical considerations of production-grade storage systems is a foundation of theoretical abstractions that enables reasoning about data placement, fault tolerance, and scheduling policies (Hummaida, Paton, and Sakellariou 2022; Kansara 2021). Mathematical structures provide

rigorous techniques to evaluate load dynamics, measure system resilience, and determine optimal partitioning strategies for large-scale datasets. Analytic approaches also aid in the understanding of bandwidth requirements, concurrency control, and latency bounds, guiding decisions on how best to allocate system resources for heterogeneous workloads (Adi et al. 2020). These theoretical insights, however, must be reconciled with the realities of hardware limitations, fluctuating user demand, and the complexities of distributed deployments.

When dealing with petabyte-scale data sets, centralized architectures often prove insufficient due to bottlenecks in single-node processing, leading to the emergence of distributed filesystems, object stores, and block storage abstractions that integrate replication, sharding, and erasure coding for enhanced reliability (Chaudhuri et al. 2021). Over time, the variety of architectural options has grown, with each alternative offering unique trade-offs in terms of performance, consistency, resilience, and cost. System architects must carefully weigh these trade-offs while employing advanced allocation policies and placement algorithms that directly influence response times, resource utilization, and total expenditures. (Xiaocui Sun et al. 2021)

In cloud environments, elasticity further complicates design choices, as horizontal scaling is as much a question of financial feasibility as it is about raw computational capacity. Scaling out large-scale storage clusters imposes added overhead related to data movement, synchronization, and rebalancing (Liu et al. 2017; Shekhar 2016). Additionally, any large-scale storage system must integrate robust monitoring and recovery mechanisms to detect and resolve node failures quickly, preventing data loss and maintaining uninterrupted service for applications. Meanwhile, fine-grained tuning for performance, such as caching or specialized indexing strategies, can substantially reduce query latency and storage overhead but must be devised with long-term growth in mind. (Hu et al. 2018)

As data volumes continue to rise, the performance bottlenecks shift from computation to data transfer and I/O operations, compelling researchers to explore new paradigms such as in-memory computing, custom hardware accelerators, and specialized interconnects. Taken together, these considerations illustrate the multifaceted nature of designing big data storage systems in the cloud (Tong, Bakhshi, and Prabhu 2022; Avula 2018). Within the following sections, this paper provides a rigorous examination of foundational theories, advanced mathematical models, and practical techniques that support scalable, cost-effective, and high-performing storage deployments in the modern cloud landscape.

2. Cloud Storage Foundations

A technical overview of the underlying structures for cloud-based storage solutions begins by considering the baseline design of distributed file systems (Cuzzocrea et al. 2018). In such architectures, data blocks are commonly replicated across multiple nodes, ensuring that in the event of hardware failure, data remains accessible through alternative replicas. This replicated design augments reliability and availability but simultaneously introduces consistency and synchronization challenges (Teing, Dehghantanha, and Choo 2019). From a purely theoretical perspective, one can analyze the relationship between the replication factor and the mean time to data loss by modeling replica failure events as independent stochastic processes. Let T represent the random variable corresponding to the time until all replicas of a given data block fail, and suppose each replica has an exponential lifetime distribution with rate parameter λ (B. Xia et al. 2021). The probability that all r replicas fail before a certain time t can be computed by:

$$P(T \leq t) = \left(1 - e^{-\lambda t}\right)^{r}.$$

This expression captures the increased reliability gained by having multiple copies (Salinas et al. 2018). Extension of this model to correlated failure scenarios or heterogeneous node reliability requires

more intricate probabilistic methods, sometimes involving Markov chains or semi-Markov processes. These theoretical foundations inform the risk management strategies of real-world cloud systems, dictating replication levels, placement policies, and backup intervals. (Gorban, Makarov, and Tyukin 2018)

Beyond replication, erasure coding has emerged as a crucial technique to reduce storage overhead while still retaining the ability to reconstruct lost data through coded fragments. The fundamental concept in erasure coding is the partition of a data object into k fragments, and the generation of m redundant fragments based on algebraic transformations in a finite field (Cheah et al. 2022). From the perspective of linear algebra over GF(2^w), one can represent the original k fragments as a $k \times n$ matrix, and the code generation process can be captured by multiplication with a coding matrix of size $n \times n$. Recovery of any lost fragments is possible as long as the number of lost fragments does not exceed m. Mathematically, the encoding matrix G can be formed by:

$$G = \begin{pmatrix} I_k \\ P \end{pmatrix},$$

where I_k is the $k \times k$ identity matrix, and P is a systematic generator for redundancy (Verma et al. 2017). The overhead in storage is governed by the ratio n/k, which must be balanced carefully against reliability and reconstruction cost.

From a resource management perspective, cloud providers seek to balance their own infrastructure constraints with the performance needs of clients (Brody et al. 2017). One line of theoretical analysis involves modeling the storage system as a multi-queue environment where each queue corresponds to a cluster or region, and tasks arrive in a stochastic fashion. By imposing queueing theory constructs, one can derive expected waiting times, throughput, and tail latency based on arrival rates, service rates, and capacity constraints (He 2020). For example, in an $M/M/c$ queue with arrival rate α and service rate β per server, the average response time depends on both c and the traffic intensity $\rho = \alpha/(c\beta)$. In large-scale deployments, extension to multi-class queueing models or networks of queues becomes necessary for capturing the multi-dimensional aspect of data access patterns, where different data sets may have distinct performance requirements. (Xu, Wang, and Yan 2021)

Another classical problem in cloud storage systems is that of load balancing across nodes. Dynamically placing new data objects or reassigning existing ones to minimize load skew while adhering to operational constraints can be formulated as an online partitioning problem (Chabbouh et al. 2017). Analytical results often rely on the application of the power-of-two-choices principle, which states that giving each incoming placement decision two or more candidate positions reduces the maximum load discrepancy exponentially, in contrast to purely random assignment. While the principle is elegantly captured by probability distributions and combinatorial arguments, real deployments must consider complex constraints, such as data locality, network bandwidth, and node heterogeneity (Holzman et al. 2017). Even so, the theoretical basis provides a starting point for the design of heuristics and policies that achieve near-optimal load distribution.

Having established these theoretical cornerstones, it is evident that cloud storage systems derive their structural and algorithmic complexity from a combination of replication schemes, erasure coding, queueing models, and load balancing algorithms (McCord 2019). Together, these elements create a layered framework where data integrity, performance, and operational efficiency interweave to support the foundational requirements of modern big data applications. In the sections that follow, a deeper exploration of mathematical modeling and performance trade-offs will help illustrate how abstract theories inform practical deployments. (Xie et al. 2021; Kansara 2022a)

3. Advanced Mathematical Modeling

Large-scale storage systems in cloud computing can be represented through the lens of complex dynamical processes that evolve over time according to factors such as arrivals of data, resource

fluctuations, and concurrency in user requests. The complexities arise due to distributed scheduling decisions, interactions among load-balancing policies, and replication or erasure-coding overhead (Yadav et al. 2018). Observing these phenomena with advanced mathematical modeling can offer insights into the long-term behavior of the system and help uncover operational bottlenecks.

A powerful formalism for these analyses is provided by continuous-time Markov chains, which allow the state of the system to be captured as a multidimensional random variable that transitions between configurations at rates specified by system parameters (Vaci et al. 2019). Define a state space $\mathcal{S}$ whose elements describe the distribution of data replicas across nodes, as well as the load or queue length on each node. From any given state $s \in \mathcal{S}$, the system can transition to other states in response to an arrival of new data, a data block failure, a node failure, or a user access request (Avula 2019). Each transition probability or rate is determined by empirical or estimated distributions of inter-arrival times, node reliability, or user demand intensities.

The state transition matrix, often denoted by Q, organizes the infinitesimal rates of moving from state s to s' (Fu et al. 2019). Once this matrix is established, one can attempt to compute the stationary distribution π that satisfies

$$\pi Q = 0, \quad \sum_{s \in \mathcal{S}} \pi(s) = 1.$$

From π, average metrics such as utilization, fraction of lost data, or expected response times can be derived (Ho and Grant 2017). However, given the enormous size of $\mathcal{S}$ in real-world storage systems, exact solutions can be intractable. Approximations, aggregation methods, or fluid-limit approaches are typically introduced to reduce complexity. In a fluid-limit model, discrete events are replaced with continuous flows that represent average arrival and service rates, leading to a set of ordinary differential equations governing the evolution of the system (M. Chen et al. 2020). For instance, let $x_i(t)$ represent the fraction of servers at load level i at time t. Then the dynamics might be described by: (Celesti et al. 2019)

$$\frac{dx_i(t)}{dt} = \Phi_i\big(x(t), \alpha, \beta\big),$$

where α is the arrival rate and β is the service rate. The functions Φ_i encode transitions among load levels due to arrivals and completions of tasks, or data arrivals and data placements in a storage-focused scenario. These fluid models enable the use of differential equation methods, stability analysis, and asymptotic expansions to predict the system's behavior at scale. (Dong and Rudin 2020)

Another sophisticated mathematical tool that can be employed involves partial differential equations (PDEs) for capturing spatiotemporal behaviors in large distributed systems. When data replication or coded fragments move across a network of nodes geographically spread among multiple regions, one can regard the system as a continuum in which local intensities of data requests or node failures vary in space and time (Tian, Qin, and Liu 2017). Consider a function $u(x, t)$ that represents the density of data fragments at location x at time t. Suppose that fragments diffuse according to a coefficient D when rebalancing occurs, and that fragments decay with rate λ if node failures are unmitigated (Sookhak, Yu, and Zomaya 2018). A simplified PDE might be expressed as:

$$\frac{\partial u}{\partial t} = D\nabla^2 u - \lambda u + f(x, t),$$

where $\nabla^2 u$ is the Laplacian capturing diffusion over the spatial domain and $f(x, t)$ represents the arrival of newly introduced data or externally triggered reallocation processes (Shen et al. 2017). Such PDE-based formulations, though abstract, assist in conceptualizing how data migrates throughout cloud regions and how node failures or capacity expansions alter system-wide fragment density.

Furthermore, the computational cost of these mathematical formulations can be linked to high-dimensional partial or ordinary differential equations, prompting the application of numerical methods (B. Li et al. 2020). Techniques such as finite difference methods, finite element methods, or spectral

methods allow one to approximate solutions over discretized time and space. Through careful selection of step sizes, basis functions, or iterative schemes, large-scale storage system dynamics can be studied in silico to identify parameter regimes that yield acceptable performance or risk profiles. (Cahan et al. 2019)

The interplay between advanced mathematical modeling and real-world deployment is an iterative process. Data-driven parameter estimation is typically necessary, requiring the collection of metrics such as node mean time to failure, request arrival patterns, code reconstruction overhead, and network latency distributions (Brink et al. 2017). Once the models are calibrated, they can guide strategic decisions regarding replication policy, erasure-coding configuration, or load balancing algorithms. In turn, revised system designs feed back into the modeling process, updating or refining the underlying equations (Topol 2019). This cyclical process ensures that theoretical insights remain relevant and actionable within ever-evolving cloud infrastructures.

4. Scalability and Cost Analysis

One of the central questions in big data storage involves how systems scale as the volume of stored data and the number of users grow (Y. Chen et al. 2022). Scalability is not merely a matter of adding more hardware resources, but rather ensuring that the incremental addition of resources yields proportional improvements in throughput and latency while keeping cost overhead manageable. Mathematical approaches to scalability often focus on analyzing how system metrics scale with increasing number of servers, data replicas, or request load. (Al-Mohannadi, Awan, and Hamar 2020)

Consider a distributed storage system employing replication with replication factor r. The total storage requirement for a dataset of size S is rS, disregarding indexing or metadata overhead (Appelbaum, Kogan, and Vasarhelyi 2017). If the system has N nodes, each with capacity C, then the feasibility condition is $rS \leq NC$. To achieve higher scalability, one might increase N and proportionally reduce the fraction of data each node holds, but the overhead from cross-node communication, replica synchronization, and concurrency control can degrade performance (T. Li et al. 2018). A mathematically grounded approach to this question can revolve around analyzing the communication complexity for data updates or retrievals. Suppose that each update triggers a synchronization protocol requiring communication with a fraction γ of the nodes holding replicas (Hamman et al. 2020). One can attempt to quantify the total traffic across the system, which is typically proportional to $\gamma \alpha S_u$, where α is the arrival rate of updates and S_u is the average size of an updated data object. If γ depends on N or the network topology, system designers must incorporate that relationship into cost, throughput, and latency estimates. (Subramanian et al. 2021)

Similarly, erasure-coded systems alter the storage overhead to $(k+m)/k \times S$, which can be smaller than r if carefully chosen. However, the reconstruction overhead for read or repair operations grows, reflected in coded systems by matrix multiplication overhead in finite fields. Such overhead can be approximated by analyzing the arithmetic complexity of the coding algorithm (Wang et al. 2019). If the field has size 2^w, the cost of a matrix-vector multiplication for m redundant fragments can scale on the order of $k \times m \times w$ bitwise operations. In large-scale systems, the interplay between lower storage overhead and higher repair overhead must be balanced to optimize total cost (Yang et al. 2017; Kansara 2022b). A typical approach is to find a sweet spot in the space of k and m that addresses both reliability objectives and overhead constraints. This trade-off can be formalized by an optimization problem: (Gai, Qin, and Zhu 2021; Avula 2020)

$$\min_{k,m} \left[C_{\text{storage}}(k, m) + C_{\text{repair}}(k, m) \right],$$

subject to constraints on data loss probability or maximum allowable latency. In some scenarios, one may also incorporate queueing-based metrics to capture the effect of repair concurrency when multiple data blocks simultaneously require reconstruction. (O'Connor et al. 2017)

A further aspect of scalability and cost pertains to multi-regional deployments. Cloud providers typically offer multiple availability zones, each with distinct pricing models for storage, retrieval, and data transfer across regions (Ploton et al. 2020). Mathematically, a multi-region cost function might be expressed as:

$$C_{\text{multi-region}} = \sum_{j=1}^{R} \left(c_j^{\text{store}} \cdot S_j + c_j^{\text{xfer}} \cdot T_j \right),$$

where R is the number of regions, c_j^{store} is the per-gigabyte storage cost in region j, c_j^{xfer} is the data transfer cost for inbound or outbound traffic, and S_j and T_j are the storage volume and transfer volume allocated in region j, respectively. Designing a strategy to place data replicas or coded fragments across regions can be viewed as an optimization problem in the face of uncertain data access patterns and failure events. (Cheng et al. 2017)

Beyond purely cost-centric concerns, cloud providers and users are interested in energy consumption and sustainability. Energy-aware models broaden the scope of analysis by incorporating power usage parameters for each node and potential power saving techniques such as spinning down idle disks or migrating data to lower-power regions when usage is minimal (Mohammed et al. 2019). Such refinements add another layer of complexity to cost-benefit calculations, merging economic factors with environmental considerations.

Empirical tests and real-world measurements often diverge from purely theoretical models, pointing to hidden overheads in metadata management, virtualization layers, or ephemeral network congestion (Xiaobo Sun et al. 2018). Nevertheless, advanced mathematical formulations remain integral to systematically approaching the design of scalable and cost-effective storage architectures. By encompassing replication or coding overhead, network transfer costs, and multi-regional deployment fees, these models enable researchers and architects to evaluate potential solutions before committing to large-scale production changes (Popic and Batzoglou 2017). They also highlight the fundamental trade-offs between performance, cost, and reliability, forcing system designers to prioritize objectives based on the specific demands of their application domains.

5. Performance Evaluation

Performance evaluation in large-scale storage systems encompasses metrics of throughput, latency, reliability, and fault tolerance (Stern et al. 2022). The inherent complexity of distributed environments makes direct analytical derivations challenging, but approximate methods and carefully controlled experiments can provide a rigorous assessment of system behavior. On the analytical side, queueing-based performance models remain a key tool for capturing request arrival processes, service times, concurrency, and queuing delays at each node (Z. Chen et al. 2020). By mapping the request flow to a network of queues, one can apply known results such as the Jackson network model in simpler cases, or more advanced queueing network models for systems with complicated job routing or concurrent read-write operations.

An illustrative example involves a read-intensive workload, in which the arrival rate α_r is primarily for read requests and the arrival rate α_w is for write requests (Vahidy et al. 2021). If each request has an average service time μ_r^{-1} for reads and μ_w^{-1} for writes, then a single-server queue would have expected response time determined by:

$$\frac{1}{\mu_r - \alpha_r} \quad \text{for reads,}$$

and

$$\frac{1}{\mu_w - \alpha_w} \quad \text{for writes,}$$

under the simplifying assumption that reads and writes are served by separate dedicated queues. With replication or coding in place, these service times become more complex, often leading to multi-phase or multi-class queueing networks that incorporate data partitioning, pipeline stages, and concurrency among replicas. (Das et al. 2017)

For empirical evaluation, system prototypes or cloud-based testbeds are often deployed, and standardized workloads are applied to measure throughput, tail latency (for instance, the 99.9th percentile of request completion times), and fault tolerance under stress. From a modeling perspective, one can interpret the empirical performance data through the prism of regression or maximum likelihood estimation to refine parameters in advanced models (Kirsal, Mapp, and Sardis 2019). This synergy between measured data and theoretical analysis helps ensure that performance predictions are not purely abstract, but anchored in verifiable real-world observations.

Another dimension of performance concerns caching and hierarchical storage strategies, where data is dynamically staged in higher-speed storage tiers, such as in-memory caches or solid-state drives, to reduce read latencies for frequently accessed items (Krishna and Elisseev 2020). Mathematical formulations for caching often rely on the independent reference model (IRM) or related variants, in which the request probability for an object is determined by its popularity distribution. One common distribution is the Zipf law, where the probability that the i-th most popular object is requested is proportional to $i^{-\theta}$ for some parameter θ. Under the IRM, the steady-state hit ratio of a cache can be approximated by analyzing the arrival and eviction processes. More nuanced caching approaches, such as segment caching or multi-tier caching, complicate these models, but the fundamental principle remains that well-designed caching mechanisms can dramatically reduce latencies and overall resource usage. (Katz and Plaza 2019)

Network performance is also pivotal, as data must travel between nodes and to and from end-users. High-performance interconnects and wide-area networks introduce additional latency, packet loss, and throughput constraints (Tien 2019). By modeling network links as channels with certain bandwidth and delay parameters, one can incorporate network transmission times into overall request latency calculations. For instance, if each data retrieval involves transferring a data block of size b across a channel with bandwidth B and one-way latency ℓ, the minimal transfer time is $\ell + \frac{b}{B}$. In multi-hop or multi-path routes, these times accumulate or can be partially parallelized, depending on the architecture (Z. Xia et al. 2021). For high-performance systems, transport protocols, congestion control algorithms, and advanced routing schemes must be tuned to avoid bottlenecks, a process that can be guided by queueing-theoretic or fluid-flow models of network traffic.

Fault tolerance and disaster recovery performance also require rigorous evaluation (Makkie et al. 2018). By injecting failure events, either at the node or the disk level, test scenarios gauge how quickly replicas or coded fragments can be reconstructed to maintain data availability. Mathematically, the distribution of reconstruction times depends on factors such as the concurrency level of repair tasks, the network bandwidth available for data movement, and the computational overhead for decoding (Fernández et al. 2019). Analytical bounds for these metrics can be derived from combination of erasure-coding complexity and queueing delays in the background repair processes. One might characterize the overall system reliability or availability by analyzing the fraction of time in which at least one data object is unrecoverable, linking reliability directly to performance constraints. (Youens-Clark et al. 2019)

In sum, performance evaluation for big data storage in cloud settings draws on a rich tapestry of methods: queueing theory, network flow modeling, caching analysis, and empirical benchmarking. By applying rigorous mathematical constructs and implementing test environments, system architects gain critical insights into how well storage designs will perform under realistic or anticipated loads (Du et al. 2017). The interplay between these techniques and the earlier theoretical underpinnings provides a holistic perspective on both the advantages and the potential pitfalls of each design choice.

6. Conclusion

Ultimately, cloud storage systems must operate under conditions of unpredictability, whether due to fluctuating workloads, sudden node outages, or the introduction of new data-intensive applications. Maintaining a robust theoretical and experimental foundation ensures that these systems adapt gracefully, providing scalable, cost-conscious, and high-performance environments for ever-growing data volumes. By recognizing and embracing the intricacies revealed by rigorous mathematical modeling, along with iterative testing in realistic settings, the community can continue to refine approaches that stand at the forefront of storage technology for the cloud era. (Ghinita et al. 2020)

References

Adi, Erwin, Adnan Anwar, Zubair A. Baig, and Sherali Zeadally. 2020. Machine learning and data analytics for the iot. *Neural Computing and Applications* 32, no. 20 (May 11, 2020): 16205–16233. https://doi.org/10.1007/s00521-020-04874-y.

Appelbaum, Deniz, Alexander Kogan, and Miklos A. Vasarhelyi. 2017. Big data and analytics in the modern audit engagement: research needs. *Auditing: A Journal of Practice & Theory* 36, no. 4 (February 1, 2017): 1–27. https://doi.org/10.2308/ajpt-51684.

Avula, Ramya. 2018. Architectural frameworks for big data analytics in patient-centric healthcare systems: opportunities, challenges, and limitations. *Emerging Trends in Machine Intelligence and Big Data* 10 (3): 13–27.

———. 2019. Optimizing data quality in electronic medical records: addressing fragmentation, inconsistencies, and data integrity issues in healthcare. *Journal of Big-Data Analytics and Cloud Computing* 4 (5): 1–25.

———. 2020. Overcoming data silos in healthcare with strategies for enhancing integration and interoperability to improve clinical and operational efficiency. *Journal of Advanced Analytics in Healthcare Management* 4 (10): 26–44.

Brink, James A., Ronald L. Arenson, Thomas M. Grist, Jonathan S. Lewin, and Dieter R. Enzmann. 2017. Bits and bytes: the future of radiology lies in informatics and information technology. *European radiology* 27, no. 9 (March 9, 2017): 3647–3651. https://doi.org/10.1007/s00330-016-4688-5.

Brody, Jennifer A., Alanna C. Morrison, Joshua C. Bis, Jeffrey R. O'Connell, Michael R. Brown, Jennifer E. Huffman, Darren C. Ames, et al. 2017. Analysis commons, a team approach to discovery in a big-data environment for genetic epidemiology. *Nature genetics* 49, no. 11 (October 27, 2017): 1560–1563. https://doi.org/10.1038/ng.3968.

Cahan, Eli M., Tina Hernandez-Boussard, Sonoo Thadaney-Israni, and Daniel L. Rubin. 2019. Putting the data before the algorithm in big data addressing personalized healthcare. *NPJ digital medicine* 2, no. 1 (August 19, 2019): 78–78. https://doi.org/10.1038/s41746-019-0157-2.

Celesti, Antonio, Maria Fazio, Fermín Galán Márquez, Alex Glikson, Hope Mauwa, Antoine Bagula, Fabrizio Celesti, and Massimo Villari. 2019. How to develop iot cloud e-health systems based on fiware: a lesson learnt. *Journal of Sensor and Actuator Networks* 8, no. 1 (January 10, 2019): 7–. https://doi.org/10.3390/jsan8010007.

Chabbouh, Olfa, Sonia Ben Rejeb, Zied Choukair, and Nazim Agoulmine. 2017. A strategy for joint service offloading and scheduling in heterogeneous cloud radio access networks. *EURASIP Journal on Wireless Communications and Networking* 2017, no. 1 (November 21, 2017): 1–11. https://doi.org/10.1186/s13638-017-0978-0.

Chaudhuri, Sheetal, Hao Han, Caitlin Monaghan, John W. Larkin, Peter Waguespack, Brian Shulman, Zuwen Kuang, et al. 2021. Real-time prediction of intradialytic relative blood volume: a proof-of-concept for integrated cloud computing infrastructure. *BMC nephrology* 22, no. 1 (August 9, 2021): 1–10. https://doi.org/10.1186/s12882-021-02481-0.

Cheah, Chor Gene, Wen Yi Chia, Shuet Fen Lai, Kit Wayne Chew, Shir Reen Chia, and Pau Loke Show. 2022. Innovation designs of industry 4.0 based solid waste management: machinery and digital circular economy. *Environmental research* 213 (June 11, 2022): 113619–113619. https://doi.org/10.1016/j.envres.2022.113619.

Chen, Min, Yongfeng Qian, Jing Chen, Kai Hwang, Shiwen Mao, and Long Hu. 2020. Privacy protection and intrusion avoidance for cloudlet-based medical data sharing. *IEEE Transactions on Cloud Computing* 8, no. 4 (October 1, 2020): 1274–1283. https://doi.org/10.1109/tcc.2016.2617382.

Chen, Yong, Yang Lu, Larisa Bulysheva, and Mikhail Yu. Kataev. 2022. Applications of blockchain in industry 4.0: a review. *Information Systems Frontiers* 26, no. 5 (February 11, 2022): 1715–1729. https://doi.org/10.1007/s10796-022-10248-7.

Chen, Zhiqi, Sheng Zhang, Zhuzhong Qian, Can Wang, Mingjun Xiao, Jie Wu, and Sanglu Lu. 2020. Sum of squares: a new metric for nfv service chain placement in edge computing environments and efficient heuristic algorithms. *CCF Transactions on Networking* 3, no. 2 (July 30, 2020): 140–153. https://doi.org/10.1007/s42045-020-00030-1.

Cheng, Ming-Ming, Qibin Hou, Song-Hai Zhang, and Paul L. Rosin. 2017. Intelligent visual media processing: when graphics meets vision. *Journal of Computer Science and Technology* 32, no. 1 (January 11, 2017): 110–121. https://doi.org/10.1007/s11390-017-1681-7.

Cuzzocrea, Alfredo, Mohamed Medhat Gaber, Edoardo Fadda, and Giorgio Mario Grasso. 2018. An innovative framework for supporting big atmospheric data analytics via clustering-based spatio-temporal analysis. *Journal of Ambient Intelligence and Humanized Computing* 10, no. 9 (August 13, 2018): 3383–3398. https://doi.org/10.1007/s12652-018-0966-1.

Das, A., Praveen Kumar Koppa, Sayan Goswami, Richard Platania, and Seung-Jong Park. 2017. Large-scale parallel genome assembler over cloud computing environment. *Journal of bioinformatics and computational biology* 15, no. 3 (May 23, 2017): 1740003–1740003. https://doi.org/10.1142/s0219720017400030.

Dong, Jiayun, and Cynthia Rudin. 2020. Exploring the cloud of variable importance for the set of all good models. *Nature Machine Intelligence* 2, no. 12 (December 10, 2020): 810–824. https://doi.org/10.1038/s42256-020-00264-0.

Du, Xue Kai, Zhi Hui Lu, Qiang Duan, Jie Wu, and Cheng Rong Wu. 2017. Ltss: load-adaptive traffic steering and forwarding for security services in multi-tenant cloud datacenters. *Journal of Computer Science and Technology* 32, no. 6 (December 8, 2017): 1265–1278. https://doi.org/10.1007/s11390-017-1799-7.

Fernández, Alberto, Isaac Triguero, Mikel Galar, and Francisco Herrera. 2019. Guest editorial: computational intelligence for big data analytics. *Cognitive Computation* 11, no. 3 (May 25, 2019): 329–330. https://doi.org/10.1007/s12559-019-09647-x.

Fu, Yinjin, Nong Xiao, Hong Jiang, Guyu Hu, and Weiwei Chen. 2019. Application-aware big data deduplication in cloud environment. *IEEE Transactions on Cloud Computing* 7, no. 4 (October 1, 2019): 921–934. https://doi.org/10.1109/tcc.2017.2710043.

Gai, Keke, Xiao Qin, and Liehuang Zhu. 2021. An energy-aware high performance task allocation strategy in heterogeneous fog computing environments. *IEEE Transactions on Computers* 70, no. 4 (April 1, 2021): 626–639. https://doi.org/10.1109/tc.2020.2993561.

Ghinita, Gabriel, Kien Nguyen, Mihai Maruseac, and Cyrus Shahabi. 2020. A secure location-based alert system with tunable privacy-performance trade-off. *GeoInformatica* 24, no. 4 (June 16, 2020): 1–35. https://doi.org/10.1007/s10707-020-00410-1.

Gorban, Alexander N., Valeri A. Makarov, and Ivan Tyukin. 2018. The unreasonable effectiveness of small neural ensembles in high-dimensional brain. *Physics of life reviews* 29 (October 2, 2018): 55–88. https://doi.org/10.1016/j.plrev.2018.09.005.

Hamman, Joseph, S. T. Henderson, Anthony Arendt, Amanda Tan, Dennis Robert Fatland, Andrew Pawloski, Daniel Pilone, et al. 2020. *The pangeo platform: a community-driven open-source big data environment,* January 17, 2020. https://doi.org/10.1002/essoar.10501751.1.

He, Pinzhen. 2020. Study of economic management forecast and optimized resource allocation based on cloud computing and neural network. *EURASIP Journal on Wireless Communications and Networking* 2020, no. 1 (August 27, 2020): 1–14. https://doi.org/10.1186/s13638-020-01790-6.

Ho, Joshua W. K., and Guy H. Grant. 2017. Modelling, inference and big data in biophysics. *Biophysical reviews* 9, no. 4 (July 30, 2017): 297–298. https://doi.org/10.1007/s12551-017-0282-6.

Holzman, Burt, Lothar At Bauerdick, Brian Bockelman, Dave Dykstra, Ian Fisk, S. Fuess, Gabriele Garzoglio, et al. 2017. Hepcloud, a new paradigm for hep facilities: cms amazon web services investigation. *Computing and Software for Big Science* 1, no. 1 (September 29, 2017): 1–15. https://doi.org/10.1007/s41781-017-0001-9.

Hu, Chunqiang, Wei Li, Xiuzhen Cheng, Jiguo Yu, Shengling Wang, and Rongfang Bie. 2018. A secure and verifiable access control scheme for big data storage in clouds. *IEEE Transactions on Big Data* 4, no. 3 (September 1, 2018): 341–355. https://doi.org/10.1109/tbdata.2016.2621106.

Hummaida, Abdul Rahman, Norman W. Paton, and Rizos Sakellariou. 2022. Scalable virtual machine migration using reinforcement learning. *Journal of Grid Computing* 20, no. 2 (April 28, 2022). https://doi.org/10.1007/s10723-022-09603-4.

Kansara, M. 2021. Cloud migration strategies and challenges in highly regulated and data-intensive industries: a technical perspective. *International Journal of Applied Machine Learning and Computational Intelligence* 11 (12): 78–121.

———. 2022a. A comparative analysis of security algorithms and mechanisms for protecting data, applications, and services during cloud migration. *International Journal of Information and Cybersecurity* 6 (1): 164–197.

———. 2022b. A structured lifecycle approach to large-scale cloud database migration: challenges and strategies for an optimal transition. *Applied Research in Artificial Intelligence and Cloud Computing* 5 (1): 237–261.

Katz, William T., and Stephen M. Plaza. 2019. Dvid: distributed versioned image-oriented dataservice. *Frontiers in neural circuits* 13 (February 5, 2019): 5–. https://doi.org/10.3389/fncir.2019.00005.

Kirsal, Yonal, Glenford Mapp, and Fragkiskos Sardis. 2019. Using advanced handover and localization techniques for maintaining quality-of-service of mobile users in heterogeneous cloud-based environment. *Journal of Network and Systems Management* 27, no. 4 (March 13, 2019): 972–997. https://doi.org/10.1007/s10922-019-09494-z.

Krishna, Ritesh, and Vadim V. Elisseev. 2020. User-centric genomics infrastructure: trends and technologies. *Genome* 64, no. 4 (November 20, 2020): 467–475. https://doi.org/10.1139/gen-2020-0096.

Li, Bo, Joshua Gould, Yiming Yang, Siranush Sarkizova, Marcin Tabaka, Orr Ashenberg, Yanay Rosen, et al. 2020. Cumulus provides cloud-based data analysis for large-scale single-cell and single-nucleus rna-seq. *Nature methods* 17, no. 8 (July 27, 2020): 793–798. https://doi.org/10.1038/s41592-020-0905-x.

Li, Tong, Kezhi Wang, Ke Xu, Kun Yang, Chathura M. Sarathchandra Magurawalage, and Haiyang Wang. 2018. Communication and computation cooperation in cloud radio access network with mobile edge computing. *CCF Transactions on Networking* 2, no. 1 (November 28, 2018): 43–56. https://doi.org/10.1007/s42045-018-0006-x.

Liu, Qin, Yuhong Guo, Jie Wu, and Guojun Wang. 2017. Effective query grouping strategy in clouds. *Journal of Computer Science and Technology* 32, no. 6 (December 8, 2017): 1231–1249. https://doi.org/10.1007/s11390-017-1797-9.

Makkie, Milad, Heng Huang, Yu Zhao, Athanasios V. Vasilakos, and Tianming Liu. 2018. Fast and scalable distributed deep convolutional autoencoder for fmri big data analytics. *Neurocomputing* 325 (October 9, 2018): 20–30. https://doi.org/10.1016/j.neucom.2018.09.066.

McCord, David M. 2019. The multidimensional behavioral health screen 1.0: a translational tool for primary medical care. *Journal of personality assessment* 102, no. 2 (November 4, 2019): 164–174. https://doi.org/10.1080/00223891.2019.1683019.

Mohammed, Bashir, Irfan Awan, Hassan Ugail, and Muhammad Younas. 2019. Failure prediction using machine learning in a virtualised hpc system and application. *Cluster Computing* 22, no. 2 (March 21, 2019): 471–485. https://doi.org/10.1007/s10586-019-02917-1.

Al-Mohannadi, Hamad, Irfan Awan, and Jassim Al Hamar. 2020. Analysis of adversary activities using cloud-based web services to enhance cyber threat intelligence. *Service Oriented Computing and Applications* 14, no. 3 (January 21, 2020): 175–187. https://doi.org/10.1007/s11761-019-00285-7.

O'Connor, Brian, Denis Yuen, Vincent Chung, Andrew G. Duncan, Xiang Kun Liu, Janice Patricia, Benedict Paten, Lincoln Stein, and Vincent Ferretti. 2017. The dockstore: enabling modular, community-focused sharing of docker-based genomics tools and workflows. *F1000Research* 6, no. 6 (January 18, 2017): 52–52. https://doi.org/10.12688/f1000research.10137.1.

Ploton, Pierre, Frédéric Mortier, Maxime Réjou-Méchain, Nicolas Barbier, Nicolas Picard, Vivien Rossi, Carsten F. Dormann, et al. 2020. Spatial validation reveals poor predictive performance of large-scale ecological mapping models. *Nature communications* 11, no. 1 (September 11, 2020): 1–11. https://doi.org/10.1038/s41467-020-18321-y.

Popic, Victoria, and Serafim Batzoglou. 2017. A hybrid cloud read aligner based on minhash and kmer voting that preserves privacy. *Nature communications* 8, no. 1 (May 16, 2017): 15311–15311. https://doi.org/10.1038/ncomms15311.

Salinas, Sergio, Changqing Luo, Xuhui Chen, Weixian Liao, and Pan Li. 2018. Efficient secure outsourcing of large-scale sparse linear systems of equations. *IEEE Transactions on Big Data* 4, no. 1 (March 1, 2018): 26–39. https://doi.org/10.1109/tbdata.2017.2679760.

Shekhar, Suman. 2016. A critical examination of cross-industry project management innovations and their transferability for improving it project deliverables. *Quarterly Journal of Emerging Technologies and Innovations* 1 (1): 1–18.

Shen, Chao, Weiqin Tong, Kim-Kwang Raymond Choo, and Samina Kausar. 2017. Performance prediction of parallel computing models to analyze cloud-based big data applications. *Cluster Computing* 21, no. 2 (November 23, 2017): 1439–1454. https://doi.org/10.1007/s10586-017-1385-3.

Sookhak, Mehdi, F. Richard Yu, and Albert Y. Zomaya. 2018. Auditing big data storage in cloud computing using divide and conquer tables. *IEEE Transactions on Parallel and Distributed Systems* 29, no. 5 (May 1, 2018): 999–1012. https://doi.org/10.1109/tpds.2017.2784423.

Stern, Charles, Ryan Abernathey, Joseph Hamman, Rachel Wegener, Chiara Lepore, Sean Harkins, and Alexander Merose. 2022. Pangeo forge: crowdsourcing analysis-ready, cloud optimized data production. *Frontiers in Climate* 3 (February 10, 2022). https://doi.org/10.3389/fclim.2021.782909.

Subramanian, Malliga, Kogilavani Shanmuga Vadivel, Wesam Atef Hatamleh, Abeer Ali Alnuaim, Mohamed Abdelhady, and Sathishkumar V E. 2021. The role of contemporary digital tools and technologies in covid-19 crisis: an exploratory analysis. *Expert systems* 39, no. 6 (October 6, 2021): e12834–. https://doi.org/10.1111/exsy.12834.

Sun, Xiaobo, Jingjing Gao, Peng Jin, Celeste Eng, Esteban G. Burchard, Terri H. Beaty, Ingo Ruczinski, et al. 2018. Optimized distributed systems achieve significant performance improvement on sorted merging of massive vcf files. *GigaScience* 7, no. 6 (May 10, 2018). https://doi.org/10.1093/gigascience/giy052.

Sun, Xiaocui, Zhijun Wang, Yunxiang Wu, Hao Che, and Hong Jiang. 2021. A price-aware congestion control protocol for cloud services. *Journal of Cloud Computing* 10, no. 1 (November 20, 2021): 1–15. https://doi.org/10.1186/s13677-021-00271-5.

Teing, Yee-Yang, Ali Dehghantanha, and Kim-Kwang Raymond Choo. 2019. Greening cloud-enabled big data storage forensics: syncany as a case study. *IEEE Transactions on Sustainable Computing* 4, no. 2 (April 1, 2019): 204–216. https://doi.org/10.1109/tsusc.2017.2687103.

Tian, Fei, Tao Qin, and Tie-Yan Liu. 2017. Computational pricing in internet era. *Frontiers of Computer Science* 12, no. 1 (March 10, 2017): 40–54. https://doi.org/10.1007/s11704-017-6005-0.

Tien, James M. 2019. Convergence to real-time decision making. *Frontiers of Engineering Management* 7, no. 2 (June 6, 2019): 204–222. https://doi.org/10.1007/s42524-019-0040-5.

Tong, Xiaorui, Roozbeh Bakhshi, and Chetan Prabhu. 2022. Industry 4.0 for aerospace manufacturing: condition based maintenance methodology, implementation and challenges. *Annual Conference of the PHM Society* 14, no. 1 (October 28, 2022). https://doi.org/10.36001/phmconf.2022.v14i1.3189.

Topol, Eric J. 2019. High-performance medicine: the convergence of human and artificial intelligence. *Nature medicine* 25, no. 1 (January 7, 2019): 44–56. https://doi.org/10.1038/s41591-018-0300-7.

Vaci, Nemanja, Dijana Cocić, Bartosz Gula, and Merim Bilalić. 2019. Large data and bayesian modeling-aging curves of nba players. *Behavior research methods* 51, no. 4 (January 25, 2019): 1544–1564. https://doi.org/10.3758/s13428-018-1183-8.

Vahidy, Farhaan S, Stephen L. Jones, Mauricio E. Tano, Juan Carlos Nicolas, Osman Khan, Jennifer Meeks, Alan Pan, et al. 2021. Rapid response to drive covid-19 research in a learning health care system: rationale and design of the houston methodist covid-19 surveillance and outcomes registry (curator). *JMIR medical informatics* 9, no. 2 (February 23, 2021): e26773–. https://doi.org/10.2196/26773.

Verma, Shefali S., Anurag Verma, Anna O. Basile, Marta-Byrska Bishop, and Christian Darabos. 2017. Session introduction: challenges of pattern recognition in biomedical data. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing* 23 (November 17, 2017): 104–110. https://doi.org/10.1142/9789813235533_0010.

VoPham, Trang, Jaime E. Hart, Francine Laden, and Yao-Yi Chiang. 2018. Emerging trends in geospatial artificial intelligence (geoai): potential applications for environmental epidemiology. *Environmental health : a global access science source* 17, no. 1 (April 17, 2018): 40–40. https://doi.org/10.1186/s12940-018-0386-x.

Wang, Lin, Lei Jiao, Jun Li, Julien Gedeon, and Max Mühlhäuser. 2019. Moera: mobility-agnostic online resource allocation for edge computing. *IEEE Transactions on Mobile Computing* 18, no. 8 (August 1, 2019): 1843–1856. https://doi.org/10.1109/tmc.2018.2867520.

Xia, Bin, Tao Li, Qifeng Zhou, Qianmu Li, and Hong Zhang. 2021. An effective classification-based framework for predicting cloud capacity demand in cloud services. *IEEE Transactions on Services Computing* 14, no. 4 (July 1, 2021): 944–956. https://doi.org/10.1109/tsc.2018.2804916.

Xia, Zhihua, Lan Wang, Jian Tang, Neal N. Xiong, and Jian Weng. 2021. A privacy-preserving image retrieval scheme using secure local binary pattern in cloud computing. *IEEE Transactions on Network Science and Engineering* 8, no. 1 (January 1, 2021): 318–330. https://doi.org/10.1109/tnse.2020.3038218.

Xie, Yi, Dongxiao Gu, Xiaoyu Wang, Xuejie Yang, Wang Zhao, Aida K Khakimova, and Hu Liu. 2021. A smart healthcare knowledge service framework for hierarchical medical treatment system. *Healthcare (Basel, Switzerland)* 10, no. 1 (December 24, 2021): 32–32. https://doi.org/10.3390/healthcare10010032.

Xu, Weilin, Jingjuan Wang, and Xiaobing Yan. 2021. Advances in memristor-based neural networks. *Frontiers in Nanotechnology* 3 (March 24, 2021). https://doi.org/10.3389/fnano.2021.645995.

Yadav, Rahul, Weizhe Zhang, Keqin Li, Chuanyi Liu, Muhammad Shafiq, and Nabin Kumar Karn. 2018. An adaptive heuristic for managing energy consumption and overloaded hosts in a cloud data center. *Wireless Networks* 26, no. 3 (November 20, 2018): 1905–1919. https://doi.org/10.1007/s11276-018-1874-1.

Yan, Hehua, Qingsong Hua, Daqiang Zhang, Jiafu Wan, Seungmin Rho, and Houbing Song. 2017. Cloud-assisted mobile crowd sensing for traffic congestion control. *Mobile Networks and Applications* 22, no. 6 (April 29, 2017): 1212–1218. https://doi.org/10.1007/s11036-017-0873-2.

Yang, Chao-Tung, Yu-Wei Chan, Jung-Chun Liu, and Ben-Shen Lou. 2017. An implementation of cloud-based platform with r packages for spatiotemporal analysis of air pollution. *The Journal of Supercomputing* 76, no. 3 (November 14, 2017): 1416–1437. https://doi.org/10.1007/s11227-017-2189-1.

Youens-Clark, Ken, Matt Bomhoff, Alise J. Ponsero, Elisha M. Wood-Charlson, Joshua Lynch, Illyoung Choi, John H. Hartman, and Bonnie L. Hurwitz. 2019. Imicrobe: tools and data-driven discovery platform for the microbiome sciences. *GigaScience* 8, no. 7 (July 1, 2019). https://doi.org/10.1093/gigascience/giz083.