REVIEW ARTICLE

FlexGroup and FlexCache Optimization for AWS-Based CAE/HPC File Workloads: A Latency and Throughput Evaluation

Mohammed Nazir[1]

[1] Honda North America, Raymond, OH, United States

Received	Revised	Accepted	Published
05 February 2023	22 March 2023	28 April 2023	15 September 2023

Abstract

Background: Computer-aided engineering (CAE) and high-performance computing (HPC) workflows place conflicting demands on shared file storage. Large sequential solver I/O, metadata-intensive preprocessing, repeated reads of common models, bursty checkpointing, and geographically separated data sources can expose both latency and aggregate-throughput limits. NetApp FlexGroup and FlexCache provide complementary mechanisms for addressing these limits in Amazon Web Services (AWS)-based ONTAP environments.

Objective: To evaluate the performance mechanisms, workload dependencies, and optimization strategies by which FlexGroup and FlexCache can improve latency and throughput for AWS-based CAE/HPC file workloads.

Methods: A structured narrative review synthesized peer-reviewed systems literature, HPC I/O characterization studies, NetApp technical reports, and AWS technical evaluations published through 2023. Evidence was organized by storage abstraction, workload pattern, metric, and optimization mechanism. Quantitative findings were retained only when directly reported; otherwise, engineering relationships were expressed analytically.

Results: FlexGroup improves scale by distributing files and metadata across constituent volumes while maintaining a single namespace. In a published evaluation, FlexGroup achieved normalized throughput of 0.90 for the SPEC SFS EDA profile relative to a local FlexVol baseline of 1.00, compared with 0.78 for a remote FlexVol placement; scaling across eight nodes reached 5.28 times the one-node EDA throughput. FlexCache addresses a different bottleneck: distance to data. Repeated reads are served from a sparse local cache, whereas first-read misses and write-around operations remain dependent on network and origin response time. AWS guidance therefore emphasizes cache-hit ratio, working-set sizing, prewarming, and grouping clients that process similar datasets.

Conclusions: FlexGroup and FlexCache should not be treated as interchangeable acceleration features. FlexGroup is primarily a scale-out and load-distribution mechanism; FlexCache is primarily a data-locality mechanism. The strongest CAE/HPC design combines workload-aware FlexGroup sizing with FlexCache placement close to compute, explicit cold-versus-warm benchmarking, and continuous observation of cache-hit ratio, network latency, origin response time, constituent balance, throughput, and IOPS.

Keywords: FlexGroup; FlexCache; Amazon FSx for NetApp ONTAP; computer-aided engineering; high-performance computing; latency; throughput; NFS; cloud bursting; storage performance.

How to cite: Nazir M. FlexGroup and FlexCache Optimization for AWS-Based CAE/HPC File Workloads: A Latency and Throughput Evaluation. International Journal of Business & Computational Science. 2023;3(1):25-31.

1. Introduction

CAE and HPC workflows are unusually sensitive to storage behavior because compute scaling can expose I/O bottlenecks that are negligible at workstation scale. A single engineering workflow may combine geometry and mesh files, millions of metadata operations, large solver inputs, sequential result streams, temporary scratch data, checkpoints, restart files, and post-processing scans. The storage system must therefore deliver both low response time for latency-sensitive operations and high aggregate throughput when many clients read or write concurrently. Production HPC studies have repeatedly shown that application I/O is heterogeneous and that effective optimization requires characterization of access pattern, request size, concurrency, and sharing behavior rather than reliance on nominal device bandwidth alone [1,2].

NetApp FlexGroup was designed to extend a file namespace across the resources of multiple constituent FlexVol volumes. Its placement heuristics distribute capacity and load while minimizing remote metadata references. In laboratory and field evaluation, FlexGroup provided a single namespace that scaled across cluster resources with modest overhead relative to manually placed local volumes [3]. This model is well aligned with CAE/HPC environments in which projects can contain very large file counts, large datasets, and many simultaneous clients.

FlexCache addresses a different problem. It creates a sparse, writable cache of an origin volume and retrieves data blocks on demand. The cache is most advantageous when a working set is repeatedly read by compute located away from the authoritative dataset. NetApp describes its benefits as reduced access latency, load distribution, and efficient storage of the active working set; the first uncached read remains sensitive to the network path and origin service time [4]. AWS extends this pattern to cloud bursting by allowing FSx for NetApp ONTAP to cache data from remote ONTAP systems, including on-premises systems and Cloud Volumes ONTAP [5].

The central optimization problem is therefore two-dimensional. FlexGroup determines how effectively storage-side resources are pooled and balanced; FlexCache determines how much of the active data path can be shortened by locality. This review evaluates the interaction of these mechanisms for AWS-based CAE/HPC file workloads and proposes a performance framework centered on latency, throughput, cache-hit behavior, metadata intensity, client concurrency, and working-set reuse.

2. Methods

2.1 Review design and evidence selection

A structured narrative review was conducted using literature and technical evidence available through 2023. Sources were eligible when they addressed at least one of the following: FlexGroup architecture or performance; FlexCache architecture or performance; FSx for NetApp ONTAP performance or caching patterns; HPC file-I/O characterization; standardized NAS/HPC storage benchmarking; or engineering/scientific workloads with characteristics applicable to CAE. Peer-reviewed systems papers were prioritized for architectural and workload claims, while first-party AWS and NetApp technical material was used for product-specific behavior, limits, and deployment patterns that are not generally reported in journals.

The primary outcomes were request latency, aggregate throughput, IOPS or operations per second, scaling efficiency, cache-hit behavior, and the latency contribution of remote data access. Secondary outcomes included capacity scale, metadata sensitivity,

workload concurrency, prewarming, and monitoring requirements. Results from different benchmark families were not pooled into a meta-analysis because their units, client counts, server configurations, access patterns, and storage generations were not directly comparable.

2.2 Analytical performance model

For an uncached read, end-to-end response time can be represented as L_miss = L_client + L_cache + L_network + L_origin + L_return, whereas a valid cached read is approximately L_hit = L_client + L_cache. Thus, the expected read latency is E[L_read] = hL_hit + (1-h)L_miss, where h is the cache-hit ratio. The model predicts that FlexCache benefit rises with hit ratio and with the latency gap between the local cache and origin path. For throughput, the effective application rate is bounded by the minimum of client/network capacity, storage-server network capacity, cache service rate, origin service rate for misses, and disk throughput for non-cached operations. These relationships were used to interpret the evidence without assigning unsupported benchmark values.

FlexGroup scaling was considered separately. Because files are allocated among constituent volumes and cluster resources, aggregate throughput can rise as more resources are engaged, but metadata-heavy operations can incur additional remote-access-layer work. The relevant optimization objective is not maximal distribution at all times; it is sufficient distribution to avoid capacity or load hot spots while limiting unnecessary cross-node metadata activity [3].

3. Results

3.1 FlexGroup throughput and metadata behavior

The most direct peer-reviewed evaluation of FlexGroup compared local FlexVol placement, FlexGroup, and remote FlexVol placement using synthetic data tests and SPEC SFS 2014 profiles. FlexGroup throughput was more than four times that of an earlier indirection-based design, while the alternative design showed 1.9 times higher average request latency at low load because every request incurred an additional network hop [3]. This finding is important for CAE/HPC architecture: a distributed namespace is useful only if the placement mechanism avoids turning distribution metadata into a serial bottleneck.

In the application-level evaluation, normalized operations per second for FlexGroup were 0.79 for random read, 0.88 for random write, 0.79 for sequential read, 0.85 for sequential write, 0.90 for the SPEC SFS EDA profile, 0.89 for SWBUILD, and 0.92 for VDA relative to a local FlexVol baseline of 1.00 [3]. The corresponding remote-FlexVol values were lower for all seven profiles. The EDA result is particularly relevant because the profile combines metadata and data throughput in a pattern representative of electronic design automation. The study further showed that FlexGroup scaling reached 5.28 times one-node throughput for EDA and 5.88 times for VDA on eight nodes, whereas the metadata-heavier SWBUILD profile reached 3.64 times [3]. These results support the principle that scale-out efficiency depends on the mix of metadata and data transfer rather than node count alone.

AWS technical evaluation of upstream engineering workloads similarly emphasizes workload dependency. A 2023 FSx for ONTAP evaluation used a 512 MB/s throughput file system and repeated FIO tests with 64-KB requests, mixed random/sequential behavior, and a 70:30 read/write ratio to explore FlexGroup and NFS choices [6]. The important methodological lesson is that CAE and geoscience applications vary in block size, client count, operating-system support, and protocol features; storage settings must therefore be profiled against representative application behavior rather than selected by label alone.

3.2 FlexCache latency and working-set locality

FlexCache creates sparse writable replicas rather than full dataset copies. When requested blocks are present and valid, they are served from the cache; otherwise, data are retrieved from the origin and retained according to cache policy [4]. The performance consequence is asymmetric: warm reads can avoid remote-distance latency, but the first read of uncached data pays the origin path. NetApp explicitly notes that performance across a latent WAN is more sensitive during the first read and recommends reducing network latency or preloading data when startup speed matters [4].

For AWS cloud bursting, this behavior allows compute to start without copying the entire project repository. AWS guidance describes on-demand population in which only blocks actually accessed in AWS are transferred to the cache. It also recommends grouping HPC clients working on similar datasets on the same FlexCache volume because shared working sets increase cache-hit ratio [5]. This is directly applicable to CAE campaigns in which many solver ranks or parameter-sweep jobs repeatedly read a common mesh, material library, model database, or restart image.

Writes require different interpretation. In the write-around mode described in the reviewed FlexCache technical report, write latency includes forwarding to the origin; the report recommends read-dominant use, with a rule of thumb of at least 80% reads and 20% writes at the cache [4]. Consequently, FlexCache should not be expected to remove origin-distance latency from write-heavy checkpoint streams. For those phases, the design priority shifts toward adequate FSx throughput, network path capacity, and appropriate placement of the write destination.

3.3 AWS scale and placement considerations

FSx for NetApp ONTAP provides the ONTAP file abstraction in AWS and supports the same broad data-management model used for on-premises NAS environments [7]. FlexCache enables a hybrid topology in which an origin can remain on premises or in another ONTAP environment while AWS compute reads a local cache [5]. This architecture can reduce migration delay for burst workloads because storage is populated lazily rather than through an up-front full copy.

FlexGroup became directly manageable in FSx for ONTAP through the AWS management interfaces in 2023, with support for volumes up to 20 PiB and billions of files, targeted at demanding workloads including EDA and seismic analysis [8]. The same period introduced scale-out FSx for ONTAP file systems using multiple high-availability pairs, with published upper bounds of 36 GB/s read throughput, 6.6 GB/s write throughput, and 1.2 million SSD IOPS across six HA pairs [9]. These are service ceilings rather than guaranteed application rates; application throughput remains constrained by protocol, client networking, request size, concurrency, cache state, and workload structure.

3.4 Optimization synthesis for CAE/HPC

The evidence supports a layered optimization sequence. First, characterize the application using representative job phases rather than a single synthetic transfer. HPC I/O tools such as Darshan were developed specifically because real production workloads differ markedly in access pattern and can be characterized with low overhead [1,2]. Second, size and distribute the FlexGroup so that capacity and load do not concentrate in a small subset of constituents. Third, place FlexCache near the compute fleet when the active dataset is remote and repeatedly read. Fourth, measure both cold-cache and warm-cache behavior. Reporting only a warm-cache result overstates first-run performance, while reporting only a cold-cache result understates steady-state performance for iterative engineering workloads.

Prewarming is justified when the required working set is known and the cost of a slow first iteration is operationally significant. ONTAP provides prepopulation mechanisms, and AWS recommends estimating hydration time from available bandwidth and working-set size [4,5]. Conversely, prewarming an entire multi-petabyte project can defeat the purpose of sparse caching. The more efficient strategy is to identify the minimum recurring input set for the next job wave.

Monitoring should distinguish network, data, and disk latency rather than relying only on client-observed elapsed time. AWS guidance for FSx for ONTAP highlights IOPS, throughput, and latency as joint performance measures and exposes ONTAP

statistics useful for locating the latency domain [6]. For FlexCache, cache-hit ratio, packet loss, available bandwidth, and origin response time are particularly important because they determine whether the cache is removing or merely relocating a bottleneck [5].

4. Discussion

4.1 Why FlexGroup and FlexCache are complementary

The principal finding is that FlexGroup and FlexCache solve different parts of the I/O path. FlexGroup improves the ability of a namespace to exploit storage-cluster resources. Its performance value appears when a workload requires more capacity, file count, or aggregate service rate than a single volume placement can provide. FlexCache improves locality. Its value appears when compute and data are separated and the working set has sufficient temporal or cross-client reuse to produce cache hits. A workload may therefore benefit from either mechanism independently, but the combined design is most attractive for large read-intensive engineering datasets that are stored remotely and processed concurrently by many AWS instances.

This distinction prevents two common design errors. The first is using a cache to solve a throughput problem caused by insufficient backend or client concurrency; a cache can reduce origin load, but a low-hit workload may remain throughput-limited. The second is using additional scale-out constituents to solve WAN latency; more backend resources cannot remove geographic round-trip time. Storage optimization must match the mechanism to the dominant term in the response-time equation.

4.2 Workload-specific implications

Preprocessing and model assembly are often metadata-sensitive and can involve large directory trees, many small files, and repeated file-open operations. These phases require attention to FlexGroup metadata overhead and client protocol behavior. Solver input phases are commonly read-dominant and may reuse a stable model across many runs; they are strong FlexCache candidates, especially in cloud-burst scenarios. Checkpoint phases are write-dominant and generally receive less benefit from write-around caching; sustained write capacity and origin placement become more important. Post-processing often re-reads large output files and can again benefit from local caching when the same results are inspected by multiple tools or engineers.

The EDA evidence is useful beyond semiconductor design because it represents a mixed workload in which metadata and data operations coexist. FlexGroup maintained 90% of the local-FlexVol normalized throughput in the published two-node application benchmark and scaled strongly as nodes were added [3]. This suggests that modest distributed-metadata overhead can be an acceptable trade-off when it enables a substantially larger pool of storage and compute resources. However, the lower scaling of SWBUILD relative to EDA and VDA demonstrates that metadata intensity can reduce efficiency; CAE workflows with millions of small files should therefore be benchmarked separately from large sequential solver streams.

4.3 Benchmarking recommendations

A defensible evaluation should report at least four states: local/no-cache baseline, remote-origin cold-cache, remote-origin warm-cache, and the intended production FlexGroup/FlexCache configuration. Each state should be tested at multiple client counts. Metrics should include median and tail latency, aggregate read and write throughput, IOPS or operations per second, cache-hit ratio, origin traffic, and job elapsed time. The storage benchmark should be accompanied by an application-level run because synthetic saturation can identify infrastructure limits but cannot guarantee solver speedup.

The use of standardized tools improves reproducibility. SPEC SFS profiles are valuable for NAS workload mixes, while FIO or IOR can isolate block-size and concurrency effects. Darshan-style instrumentation can identify whether the application is actually generating the assumed pattern [1,2,10]. Repeated tests should separate cache warm-up from steady state, and cache prepopulation should be documented because it materially changes first-access latency.

4.4 Limitations

Direct peer-reviewed experiments that jointly evaluate FlexGroup and FlexCache on AWS with commercial CAE solvers are limited. The evidence base therefore combines a peer-reviewed FlexGroup systems evaluation with first-party technical reports and AWS workload demonstrations. Product-specific throughput ceilings describe provisioned service capability, not guaranteed application performance. Additionally, CAE/HPC is not a single workload class: computational fluid dynamics, finite-element analysis, reservoir simulation, EDA, rendering, and software build pipelines differ in file size, read/write ratio, metadata density, concurrency, and reuse. The proposed optimization framework should therefore be treated as a measurement-driven design method rather than a universal configuration recipe.

5. Conclusion

FlexGroup and FlexCache provide complementary optimization mechanisms for AWS-based CAE/HPC file workloads. FlexGroup expands namespace scale and distributes load across storage resources; peer-reviewed evidence demonstrates competitive throughput for EDA-like mixed workloads and substantial scale-out as nodes are added. FlexCache reduces repeated-read latency by moving the active working set closer to compute, but cold reads and write-around operations remain dependent on network and origin performance. The most effective architecture therefore combines workload-aware FlexGroup sizing, locality-aware FlexCache placement, selective prewarming, grouping of clients with common working sets, and monitoring that separates cache, network, origin, and disk contributions. For journal and engineering evaluation, performance claims should be reported separately for cold and warm cache states and validated with both storage microbenchmarks and end-to-end CAE/HPC job timing.

Declarations

Ethics approval and consent to participate: Not applicable.

Consent for publication: Not applicable.

Data availability: No primary participant dataset was generated. Quantitative values reported in this review were obtained from the cited published literature and technical evaluations.

Competing interests: None declared.

References

[1] Carns P, Harms K, Allcock W, Bacon C, Lang S, Latham R, et al. Understanding and improving computational science storage access through continuous characterization. In: 2011 IEEE 27th Symposium on Mass Storage Systems and Technologies (MSST). IEEE; 2011. p. 1-14. doi:10.1109/MSST.2011.5937212.

[2] Snyder S, Carns P, Harms K, Ross R, Lockwood GK, Wright NJ. Modular HPC I/O characterization with Darshan. In: 2016 5th Workshop on Extreme-Scale Programming Tools (ESPT). IEEE; 2016. p. 9-17. doi:10.1109/ESPT.2016.006.

[3] Kesavan R, Hennessey J, Jernigan R, Macko P, Smith KA, Tennant D, et al. FlexGroup volumes: a distributed WAFL file system. In: 2019 USENIX Annual Technical Conference (USENIX ATC 19). Renton (WA): USENIX Association; 2019. p. 135-148.

[4] Hurley C, Ecton E. FlexCache in ONTAP: ONTAP 9.11.1. Technical Report TR-4743. NetApp; 2022.

[5] Brown E, Dickmeis J. Caching data using Amazon FSx for NetApp ONTAP. AWS Storage Blog. 2022 Feb 15.

[6] McDonald T. Enhance your upstream workloads with Amazon FSx for NetApp ONTAP. AWS Storage Blog. 2023 Jun 5.

[7] Amazon Web Services. Introducing Amazon FSx for NetApp ONTAP. AWS What's New. 2021 Sep 2.

[8] Barr J. FlexGroup volume management for Amazon FSx for NetApp ONTAP is now available. AWS News Blog. 2023 Nov 26.

[9] Barr J. New - Scale-out file systems for Amazon FSx for NetApp ONTAP. AWS News Blog. 2023 Nov 26.

[10] Luu HT, Winslett M, Gropp W, Ross R, Carns P, Harms K, et al. A multiplatform study of I/O behavior on petascale supercomputers. In: Proceedings of the 24th International Symposium on High-Performance Parallel and Distributed Computing. New York: ACM; 2015. p. 33-44.

[11] Chowdhury F, Zhu Y, Heer T, Paredes S, Moody A, Goldstone R, et al. I/O characterization and performance evaluation of BeeGFS for deep learning. In: Proceedings of the 48th International Conference on Parallel Processing. New York: ACM; 2019. Article 80. doi:10.1145/3337821.3337902.

[12] Weil SA, Brandt SA, Miller EL, Long DDE, Maltzahn C. Ceph: a scalable, high-performance distributed file system. In: Proceedings of the 7th USENIX Symposium on Operating Systems Design and Implementation. USENIX Association; 2006. p. 307-320.

[13] Welch B, Unangst M, Abbasi Z, Gibson GA, Mueller B, Small J, et al. Scalable performance of the Panasas parallel file system. In: Proceedings of the 6th USENIX Conference on File and Storage Technologies. USENIX Association; 2008. p. 17-33.

[14] Standard Performance Evaluation Corporation. SPEC SFS 2014 SP2 User Guide. Gainesville (VA): SPEC; 2017.

Tables

Table 1. Included studies and technical evidence.

Source	Year	Evidence type	Workload / system	Principal relevance
Carns et al. [1]	2011	Peer-reviewed HPC I/O study	6,481 jobs / 38 projects on Blue Gene/P	Application I/O patterns are heterogeneous; continuous characterization supports storage tuning.
Snyder et al. [2]	2016	Peer-reviewed profiling study	Darshan on production HPC systems	Modular application-level I/O characterization can be deployed with negligible overhead.
Kesavan et al. [3]	2019	Peer-reviewed distributed file-system evaluation	FlexGroup; SPEC SFS EDA/SWBUILD/VDA; 1-8 nodes	FlexGroup balanced scale and overhead; EDA throughput 0.90 vs local baseline 1.00 and remote 0.78; eight-node EDA scaling 5.28x.
Hurley & Ecton [4]	2022	NetApp technical report	FlexCache ONTAP 9.11.1	Warm reads behave like FlexGroup; first reads across latent links are network/origin sensitive; prewarming and read-dominant workloads recommended.
Brown & Dickmeis [5]	2022	AWS technical evaluation	FSx for ONTAP FlexCache cloud-burst patterns	Lazy loading reduces full-dataset migration; grouping HPC clients by common datasets raises cache-hit ratio.
McDonald [6]	2023	AWS engineering workload evaluation	Upstream G&G; FIO; 64-KB mixed I/O	FlexGroup/NFS performance is workload dependent; latency, throughput, IOPS and protocol choice should be actively profiled.
AWS [7]	2021	Service launch / architecture	FSx for NetApp ONTAP	Established managed ONTAP file service in AWS with NFS/SMB/iSCSI and ONTAP data-management features.
Barr [8]	2023	AWS service evaluation/launch	FSx for ONTAP FlexGroup	FlexGroup support up to 20 PiB and billions of files; positioned for EDA, seismic and build/test workloads.
Barr [9]	2023	AWS scale-out performance specification	2-6 HA pairs	Published ceilings up to 36 GB/s read, 6.6 GB/s write and 1.2M SSD IOPS across six HA pairs.
Luu et al. [10]	2015	Peer-reviewed HPC workload study	Multiple petascale platforms	I/O behavior varies across platforms and applications; scale alone does not determine storage efficiency.
Chowdhury et al. [11]	2019	Peer-reviewed file-system evaluation	BeeGFS / deep-learning I/O	Highlights client concurrency, access pattern and distributed file-system design as determinants of throughput.

Table 2. Functional separation of the principal optimization layers.

Layer	Primary function	Best-fit condition	Key controls	Main failure mode
FlexGroup	Scale-out namespace and capacity/load distribution	Many clients; large file counts; aggregate throughput demand	Constituent count/placement, HA pairs, protocol, client concurrency	Remote metadata operations and imbalance can add latency.
FlexCache	Data locality through sparse caching	Read-dominant repeated access to remote datasets	Cache placement, cache size, hit ratio, prewarm, client grouping	Cold misses and writes remain origin/network dependent.
FSx throughput/IOPS	Provisioned service capacity	Sustained high-throughput or high-IOPS phases	Throughput capacity, SSD IOPS, deployment type	Provisioned ceiling does not guarantee application rate.
Client/network	Path capacity and protocol efficiency	Large EC2 fleets and parallel NFS/SMB access	Instance networking, NFS version/nconnect, request size	Client-side limits can mask storage-side scaling.
Origin storage	Miss/write service rate	Hybrid cloud and cross-region data access	Origin latency, load, network path	Origin bottlenecks propagate to FlexCache miss/write latency.

Table 3. Engineering metrics for latency and throughput evaluation.

Metric	Analytical form	Interpretation
Expected read latency	E[L] = hLhit + (1-h)Lmiss	Quantifies value of cache hit ratio; larger origin-distance penalty increases cache benefit.
Cold-read latency	Lmiss = Lclient + Lcache + Lnetwork + Lorigin + Lreturn	Use to isolate whether the miss penalty is network- or origin-dominated.
Throughput ceiling	Tapp <= min(Tclient, Tnetwork, Tserver, Tdisk/origin)	Prevents interpreting storage scale-out as useful when another layer is already limiting.
Cache hydration time	Thydrate ≈ active working-set bytes / effective origin-to-cache bandwidth	Supports decisions about on-demand loading versus prewarming.
Scaling efficiency	E_n = T_n / (n × T_1)	Separates absolute speedup from efficiency as nodes/HA pairs are added.
Job impact	I/O fraction × storage speedup	Amdahl-style interpretation: storage tuning cannot accelerate compute-dominated phases proportionally.

Table 4. Workload-specific optimization recommendations for CAE/HPC.

CAE/HPC phase	Typical I/O pattern	Preferred optimization	Evaluation emphasis
Model/mesh ingest	Large shared read; moderate metadata	FlexGroup + FlexCache near compute	Prewarm known input set if first-run latency is material.
Parameter sweep / ensemble	Repeated common input across many clients	FlexCache + grouped clients; FlexGroup backend	High reuse can raise hit ratio and reduce origin traffic.
Solver checkpoint	Write-heavy sequential/bursty	FlexGroup / adequate FSx write capacity	Write-around cache does not remove origin acknowledgment path.
Metadata-heavy preprocessing	Many small files / opens / stats	FlexGroup sized for metadata concurrency	Benchmark separately; metadata scaling may be lower than streaming scaling.
Post-processing	Large repeated scans of results	FlexCache when outputs are remote and repeatedly read	Consider bidirectional caching when cloud outputs are consumed elsewhere.
Distributed engineering team	Remote shared project data	Regional/AZ FlexCache with single source of truth	Monitor cache coherency path, network and permissions.

Table 5. Monitoring signals and corrective actions.

Signal	Observed symptom	Likely cause	Corrective action
Cache hit ratio	Falling hit ratio	Cache undersized or clients share dissimilar datasets	Increase cache size; regroup clients by working set; reconsider caching candidate.
Network latency / packet loss	High miss or write latency	WAN/AZ path dominates	Improve path; relocate cache; prewarm critical dataset.
Origin response time	Misses remain slow despite good network	Origin saturated or storage-limited	Scale/tune origin; reduce miss rate.
Constituent utilization	Uneven capacity or load	Suboptimal FlexGroup balance	Review constituent count, capacity distribution and workload placement.
Client throughput	Storage underutilized at high client wait time	EC2 networking/protocol/client serialization	Increase client parallelism; review protocol and request size.
Cold vs warm job time	Large first-run penalty	Hydration dominates startup	Prewarm minimum active set or schedule warm-up before expensive solver allocation.

Figures

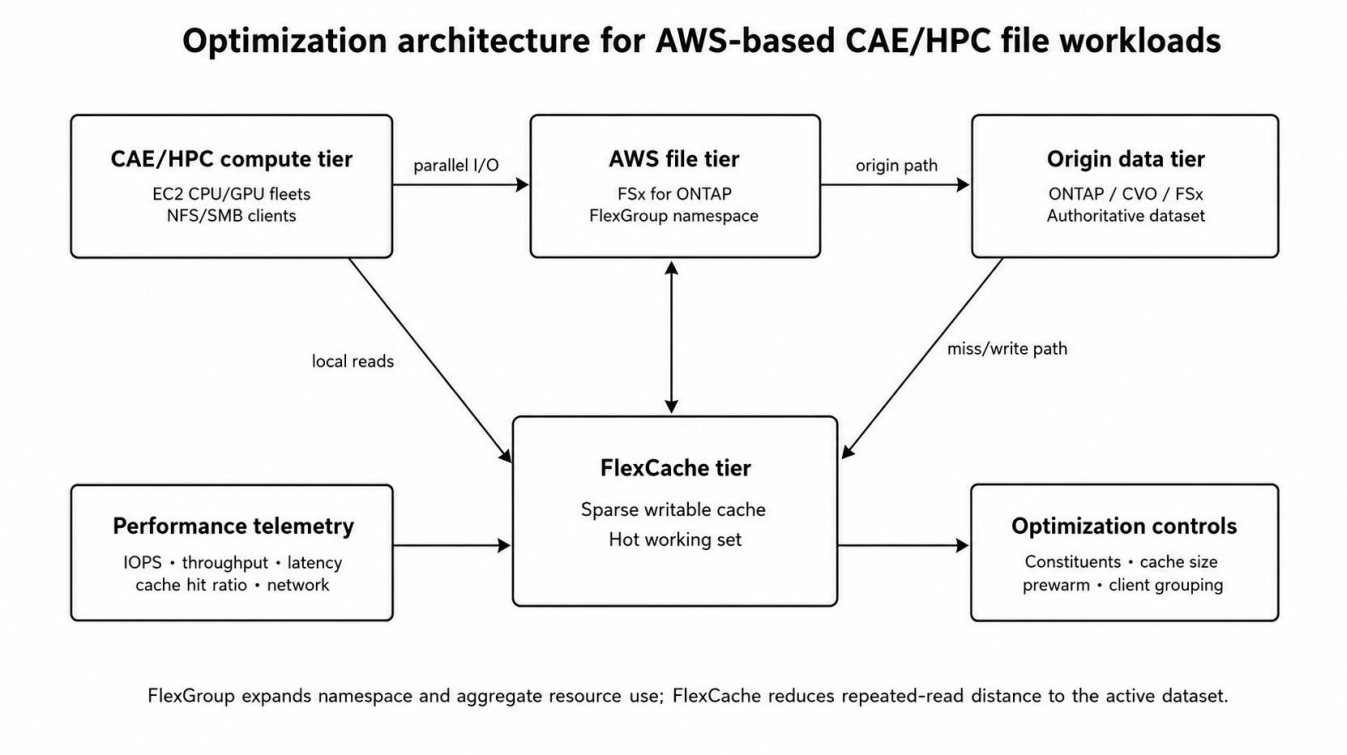

Figure 1. Conceptual architecture for combining FlexGroup scale-out storage with FlexCache locality in AWS-based CAE/HPC workloads. The origin may reside on premises, in Cloud Volumes ONTAP, or in another FSx for ONTAP file system.

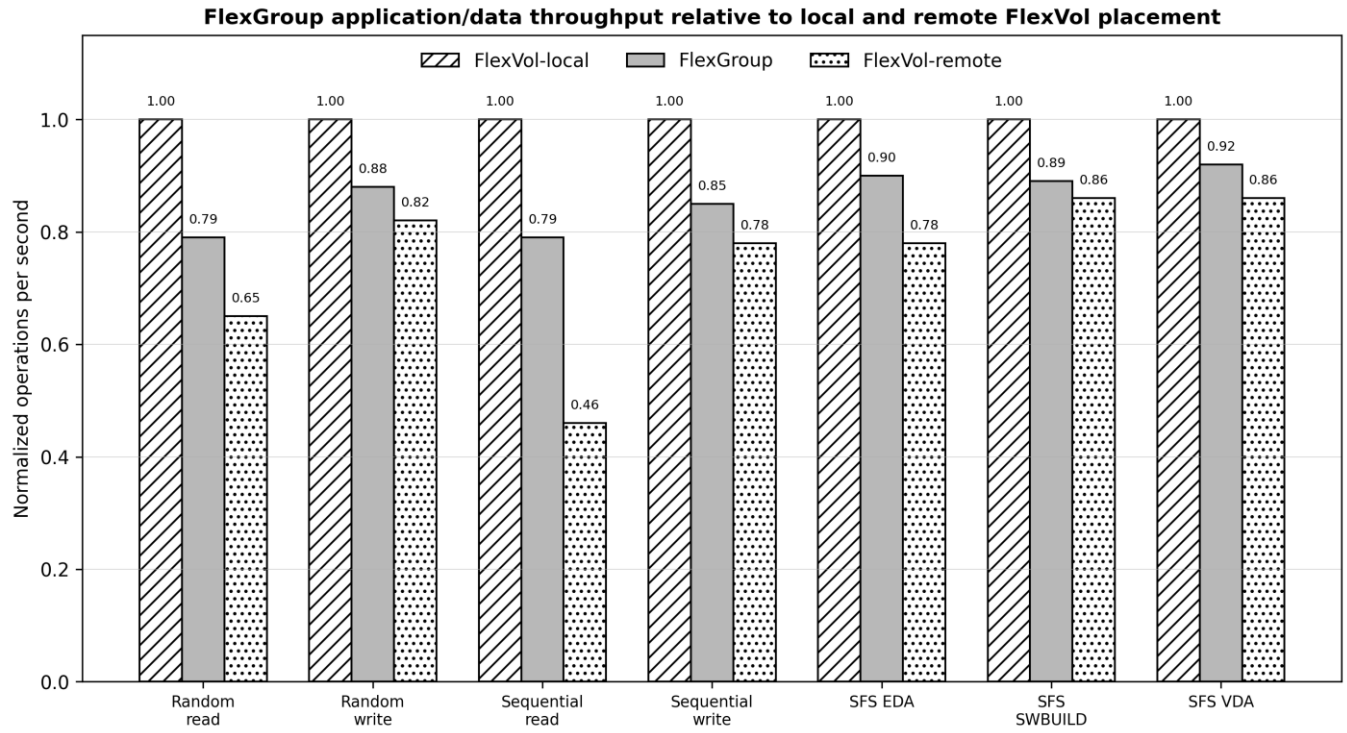

Figure 2. Normalized operations per second for application and data benchmarks reported by Kesavan et al. [3]. Values are normalized to FlexVol-local = 1.00; higher is better. The SPEC SFS EDA profile is representative of an engineering workload with both metadata and data throughput.

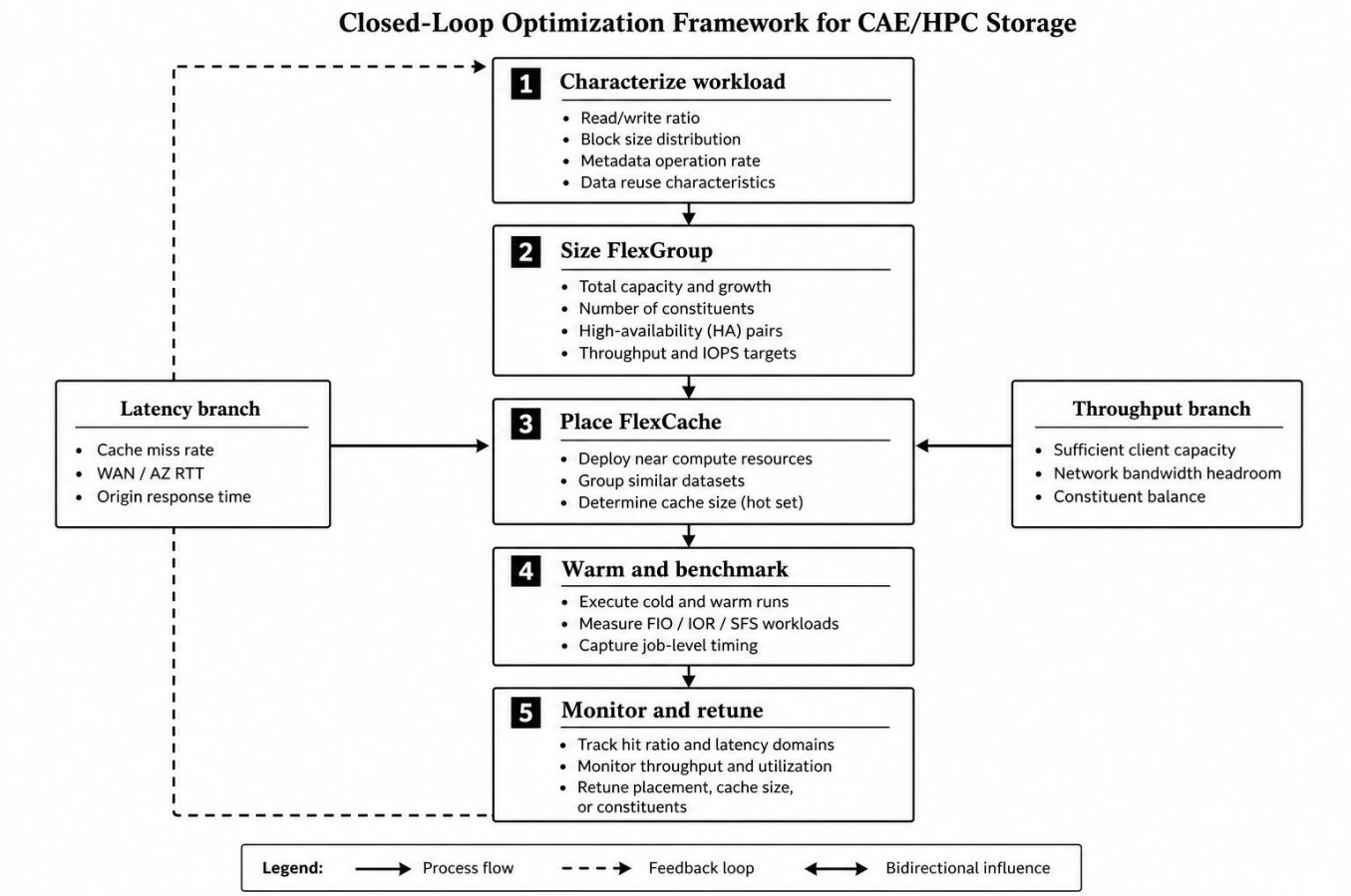

Figure 3. Closed-loop optimization framework. FlexGroup sizing and FlexCache placement should be driven by measured application behavior and retuned from latency, throughput, hit-ratio and load-balance telemetry.