Whitepaper

The Silk Cloud Data Platform Architecture

February 2024

Table of Contents

About Silk 2
Executive Summary 2
Introduction 3
Silk Cloud Data Platform Overview 3
Benefits of Silk's Architecture 6
Silk's Cloud Data Architecture Principles 7
Performance 7
Scaling Your Environment 7
Using Global Adaptive Block Size 9
Deduplicating Your Data 10
Maximizing Low Latency or High Throughput with Read Boosts 10
Running Your Environment with Silk on VMs 10
Resiliency 11
Promoting High Availability 11
Supporting Disaster Recovery through Replication 12
Protecting Data with Distributed Parity 13
Managing Metadata 14
Running Your Environment with Silk on Managed Disks 14
Data Lifecycle Management 14
Using Enterprise Data Services 14
Managing Your Silk Environment 17
Securing Your Data 18
Cost Efficiency 18
Provisioning Resources Dynamically 19
Choosing the Right Infrastructure 19
Summary 19
Appendix 20
Appendix A: Understanding Silk's IO Flow 20
Appendix B: Understanding Silk's RAID Structure 29

About Silk

Silk is more than just a storage solution or a cloud migration tool. We're an intelligent data platform that optimizes cloud resources to significantly improve performance, resiliency, data lifecycle management, and cost efficiency in the cloud. We operate between databases and cloud infrastructure to let our customers manage data in ways cloud providers can't. Our platform gives you the means to shift your mission-critical data to the cloud and operate on par with, or beyond, even the fastest on-prem environments.

Silk lets you use one platform to run multiple databases concurrently, giving you more ways to improve management of your data. The platform features two-tier virtualization with a fully distributed, cloud-native, multi-cloud framework that decouples your compute and storage and promotes self-optimizing for scalability. Our dynamic database block size alignment uses an adaptive, log-structured datastore to improve performance, while our elastic architecture optimizes latency and throughput. Silk uses a shared multi-tenant datastore so transactional, analytical, and mixed workloads can run together on pooled resources for greater efficiency. We are a database-, application-, and cloud-agnostic platform that enables companies to harness the full power of the cloud.

The Silk Cloud Data Platform can give you the tools you need to make your cloud environment run up to 10x faster. Contact our Sales team for a demo to see how Silk can help you make the most of your cloud resources.

Executive Summary

As more organizations move their data to the cloud, Silk has set out on a mission to rewrite the rules of data operations in the cloud. Our platform is architected in a way that makes it easy to migrate and manage your data in any cloud environment. Silk lets customers compose data resources based on application requirements, allocating just the right amount of both performance and capacity while minimizing resource overprovisioning and waste. You can extend your environment's operational flexibility by scaling performance and capacity resources independently of each other in a dynamic cloud framework.

The Silk Cloud Data Platform operates using industry-leading software and services. Our software stack is built on Silk Data Pods (SDPs), Silk Flex, and Silk Clarity, which work together to provide a rich set of data services, machine learning, analytics, and policy-based automation and orchestration. Our tools and services support a comprehensive cloud optimization strategy through performance, resiliency, data lifecycle management, and cost efficiency.

This whitepaper describes the Silk Cloud Data Platform's architecture, including its core features and functionalities.

Introduction

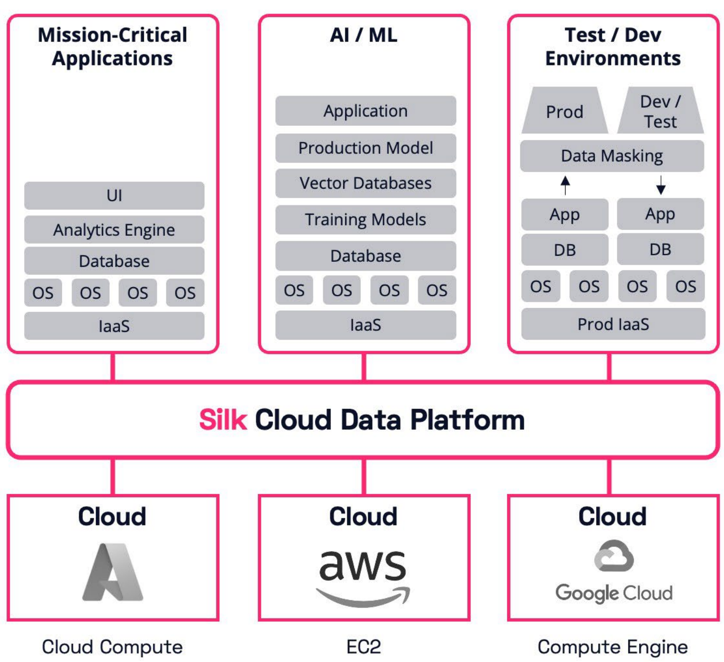
The Silk Cloud Data Platform architecture is fundamentally different from traditional cloud frameworks. Conventional cloud environments don't support shared architectures the way datacenters and on-premises environments do. Instead, the cloud typically uses a 1:1 schema in which each database must be provisioned with the resources it needs and must be updated (often manually) to accommodate any performance shifts. Silk's technology doesn't rely on this limited model. We have separate capacity and performance layers in a shared framework that improves what you can do with your databases and allows you to achieve economies of scale within your cloud environment.

Silk Cloud Data Platform Overview

The Silk platform is made up of three industry-leading technologies:

- **Silk Data Pod:** A data hypervisor for resource virtualization
- **Silk Flex:** A data orchestration platform for managing cloud resources
- **Silk Clarity:** A cloud-based AIOps engine that delivers predictive analytics

These tools work together to help you quickly scale your data framework to meet performance and capacity needs, along with the flexibility to use any cloud infrastructure. They efficiently manage data for any application type and environment, offering a single platform that can run in public, private, hybrid, and multi-cloud systems.



Silk Data Pods (SDPs) allow customers running on any cloud provider to accelerate their cloud journey by delivering enterprise data services with a highly flexible shared data framework. SDPs turn underlying cloud infrastructure into the world's most capable high-performance, scale-out data virtualization and mobility platform. The SDP architecture uses three node types as the building blocks of the Silk Cloud Data Platform:

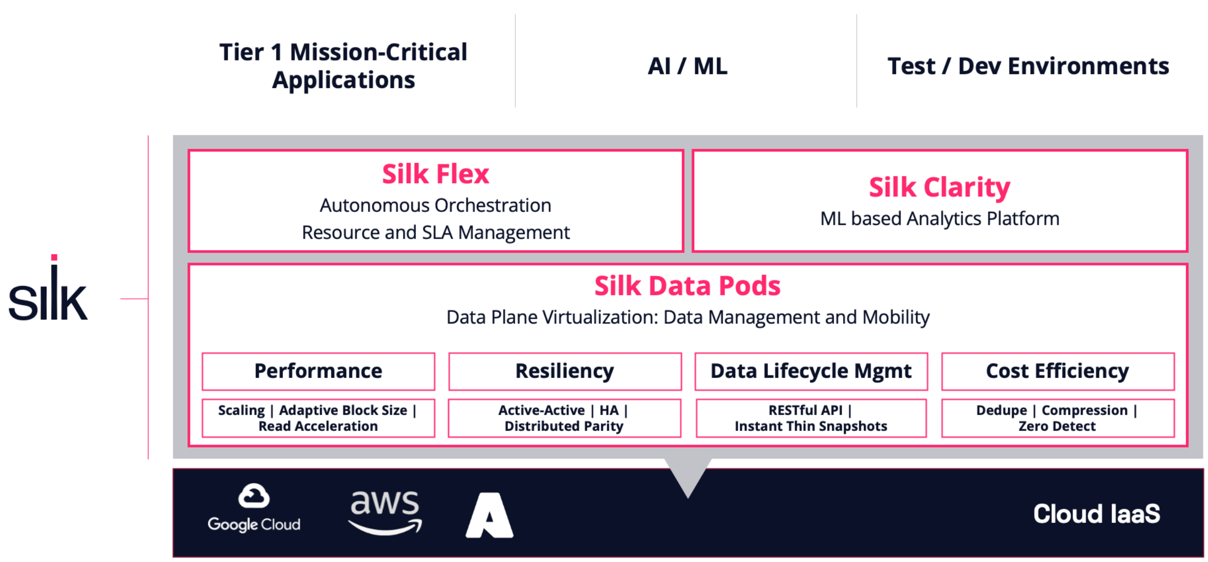
- **Compute/controller node (c.node):** Data management and optimization
- **Data node (d.node):** Data endpoint virtualization
- **Media node (m.node):** d.node management, protected using RAID-based erasure coding

Silk Flex lets you manage resources in your Silk implementation by dynamically composing, optimizing, managing, and reallocating resources for extreme performance and proactive resilience. It orchestrates cloud resources as new workloads emerge, move, and evolve over time through the platform's GUI or RESTful API and infrastructure-as-code operations. The platform deploys and configures Silk SDPs to virtually associate c.nodes and m.nodes and build shared data assets that offer performance, resiliency, and cost efficiency. Flex provides enterprise data management capabilities with unmatched flexibility, allowing customers to build and manage resources with a simple finger swipe or line of code.

Flex is available for download from the Google Cloud or Microsoft Azure marketplaces.

Silk Clarity is a cloud-based analytics platform with a comprehensive set of management and monitoring functionalities enhanced by machine learning. It offers the unique ability to leverage application-level intelligence and big data analytics, enabling you to get more productivity out of your environment and higher performance for business-critical applications.

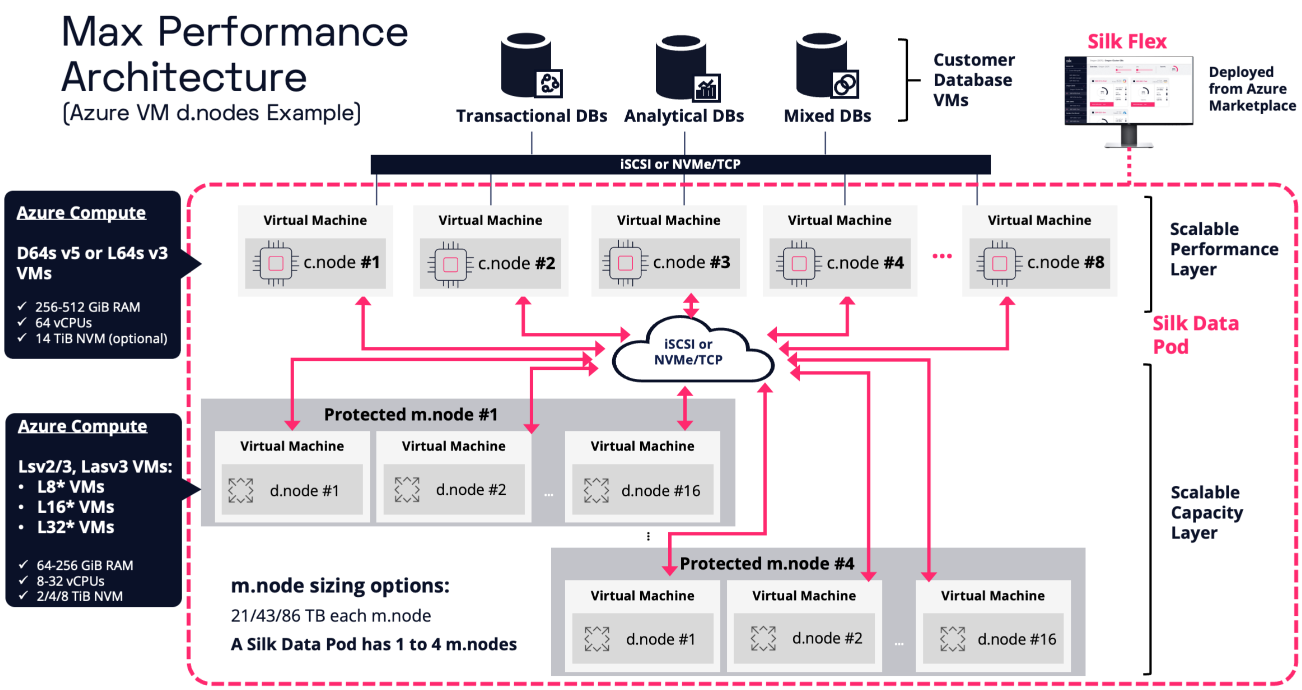
Clarity extends the capabilities of the Silk Cloud Data Platform to make it one of the most advanced cloud data management engines in the industry. Its big data platform collects millions of active call-home data points from its customer base to drive automation and predictive alerting. Using advanced analytics and modeling, Clarity uncovers new insights, automates business management, and provides recommendations for self-healing policies and preemptive resource optimization. Clarity is tightly integrated with SDPs and Flex, allowing it to gather, report on, and act on aggregated performance trends, capacity utilization, data protection metrics, and real-time events.



Silk's configuration requires **at least** two c.nodes per SDP for redundancy in the platform's consistently active-active architecture, and the configuration can scale out to eight c.nodes per SDP. The c.nodes are always deployed on virtual machines, and each c.node is comprised of:

- Up to 64 VCPUs
- Up to 512GB RAM
- Up to 100Gbps connectivity

Silk implementations also require a base m.node, with the ability to scale up additional m.nodes as needed. Every m.node provides 9 or 16 logical d.nodes with up to 8TB of physical capacity each and either ephemeral NVMe media or native protected storage, depending on the cloud provider's capabilities. d.nodes are composed of either virtual machines or managed disks using virtual resources from the cloud provider.



Silk's technology uses SDPs to provision and manage your cloud resources. Use Flex to configure, deploy, and control your SDPs, and use Clarity to monitor your environment's performance. The platform is designed to create a seamless user experience that lets you operate your cloud resources efficiently and effectively.

Benefits of Silk's Architecture

The Silk Cloud Data Platform offers a shared framework in which all cloud data resources can easily move between cloud types and cloud providers, offering a true multi-cloud experience. This framework separates data from the infrastructure it lives on, giving customers far more flexibility, performance, and efficiency with cloud resources. Silk's scalable architecture drives cloud migration and management, and it is built around four principles:

- **Performance:** Silk delivers unmatched, consistent performance levels, and is a top-rated solution for analytics databases and high-performance compute workloads. The platform lets you adjust underlying cloud resources based on application requirements and delivers multi-petabyte levels of scalability over multiple SDPs. Silk excels at serving mixed workloads via a patented global variable block size algorithm, allowing customers to receive real-time reports and faster database queries. The Silk architecture supports horizontal and vertical scaling to meet dynamic performance needs, while exclusive features like global inline deduplication and read boosts minimize latency and maximize IOPS and throughput.

- **Resiliency:** The Silk platform is made up of cloud resources that are packaged into SDPs for easy, efficient, and reliable operation. Each SDP can support extreme performance and capacity requirements. Silk is highly available and fault tolerant, with an active-active framework and full redundancy so there's no single point of failure. Your data is well-protected and recoverable with dual parity through RAID volumes. The platform's availability, fault tolerance, and disaster recovery features ensure that your cloud environment, and every piece of data within it, is as resilient as possible.

- **Data lifecycle management:** Silk's unique scaling architecture supports dynamic performance and capacity needs. The platform can scale performance linearly by adding or removing c.nodes, and can independently scale capacity by adding m.nodes. This provides more flexibility than rigid cloud-native solutions that have stricter scaling limitations and can't use a true shared metadata schema. Silk's technology significantly improves the operational efficiency of cloud resources through data management features like snapshots, replication, and deduplication. Our shared-capacity framework combines data services, data reduction, and data security that spans globally across all Silk resources to continuously optimize your cloud environment while simultaneously simplifying data management.

- **Cost efficiency:** Silk's goal is to make cloud adoption as cost efficient as possible. The platform achieves this by separating data from its native infrastructure, using two-tier virtualization to enable flexible use of cloud resources, and providing resource pooling to allocate resources exactly where they're needed. Use Silk to minimize resource waste and reduce cloud costs by dynamically provisioning, decommissioning, and reallocating resources as your applications' needs change. We also offer data reduction –and thereby cost reduction– through compression, thin provisioning, deduplication, and zero elimination. The platform is designed to optimize costs both within Silk and beyond, extending savings to your cloud environment's native resources. Our product architecture allows us to deliver exceptional performance and reduce the number of resources, as well as the total cost, required to support your applications.

Silk's Cloud Data Architecture Principles

Performance

Silk is on a mission to maximize performance in the cloud. To do this, we borrowed some of our infrastructure from traditional storage solutions. While we share one stack with storage, the way we work is fundamentally different. Our platform lets you scale up and out without significant downtime and without requiring changes to your environment's codebase. We provide the benefits and ease of use of a virtual volume while optimizing data paths for efficiency and stability. When you deploy the Silk Cloud Data Platform from the marketplace, you install Flex to manage your environment and Clarity to monitor your performance. SDPs consistently send messages about their performance to Clarity, so you have a single pane of glass for viewing all your environment's key metrics, including capacity used, data reduction, bandwidth, IOPs, and latency.

The Silk Cloud Data Platform lets you take your cloud performance to the next level with features that aren't natively available from cloud providers, as well as the option to run your SDPs on VMs. Our IO flow controls how data is written to SDPs and how it is read, which is integral to Silk's functionality. This flow allows Silk to perform reads and writes quickly, efficiently, and reliably using advanced technologies like global inline selective deduplication, global adaptive block sizing, inline compression, distributed metadata, and distributed parity. Our scalability, IO flow, and customization options like read transaction boost or read throughput boost keep your environment running smoothly, with consistently low latency and high throughput.

Scaling Your Environment

Silk's uniquely designed scaling architecture lets you expand your environment's performance and capacity separately while maintaining low latency. You can scale in, out, or up with additional controllers for performance or additional disks of capacity, depending on your needs. Any increase in capacity results in automatic rebalancing of data within your SDPs, without human intervention or management. The Silk framework is built for multi-petabyte levels of scalability through snapshots, thin provisioning, and support for multiple SDPs. These features also allow you to transform your production data for dev, testing, or other business use cases, adding another layer to your scaling strategy. This type of scalability is key for building a cost-effective infrastructure that meets exact performance and capacity requirements for your new and existing applications.

By default, each SDP has a minimum configuration of two c.nodes and one m.node. Each c.node is connected to the m.node via one of many available networking protocols. New volumes are automatically distributed between all m.nodes and can be accessed from every c.node in your SDP. Likewise, existing volumes and data are automatically redistributed between all the d.nodes in an SDP as part of the platform's standard write handling.

Silk allows you to scale up capacity by adding more m.nodes to your environment's base configuration, without requiring any additional resources. Vertical scaling in Silk provides several cost and performance benefits, including:

- Increasing capacity density and reducing cost per GB
- Performing scaling operations online with no downtime or performance decreases
- Offering the same performance as your original, smaller configuration
- No changes required to any of the host connections or definitions

- Allowing customers to mix and match with the latest and most cost-efficient cloud infrastructure technology

- Customizing the m.node configuration by mixing and matching m.nodes with different capacities

Silk also lets you scale out by increasing the number of c.nodes within your SDPs, without requiring more m.nodes. The platform handles performance and capacity separately to allow you to add or remove c.nodes linearly as your compute needs evolve. Horizontal scaling includes benefits like:

- Increasing the number of c.nodes and the number of m.nodes independently

- Increasing performance measures like IOPs and throughput

- Keeping latency consistently low

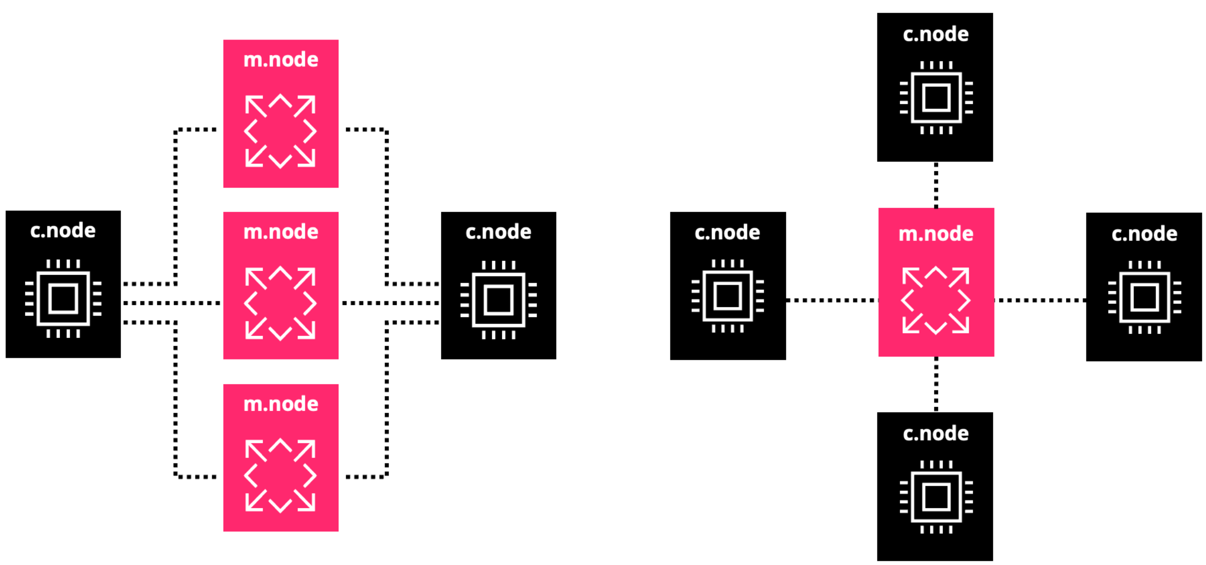
- Reallocating or decommissioning resources

- Performing scaling operations online with no downtime

- Automatically engaging all c.nodes as your environment scales for efficient data deduplication

- Allowing new host connections and letting both new and existing hosts access all volumes

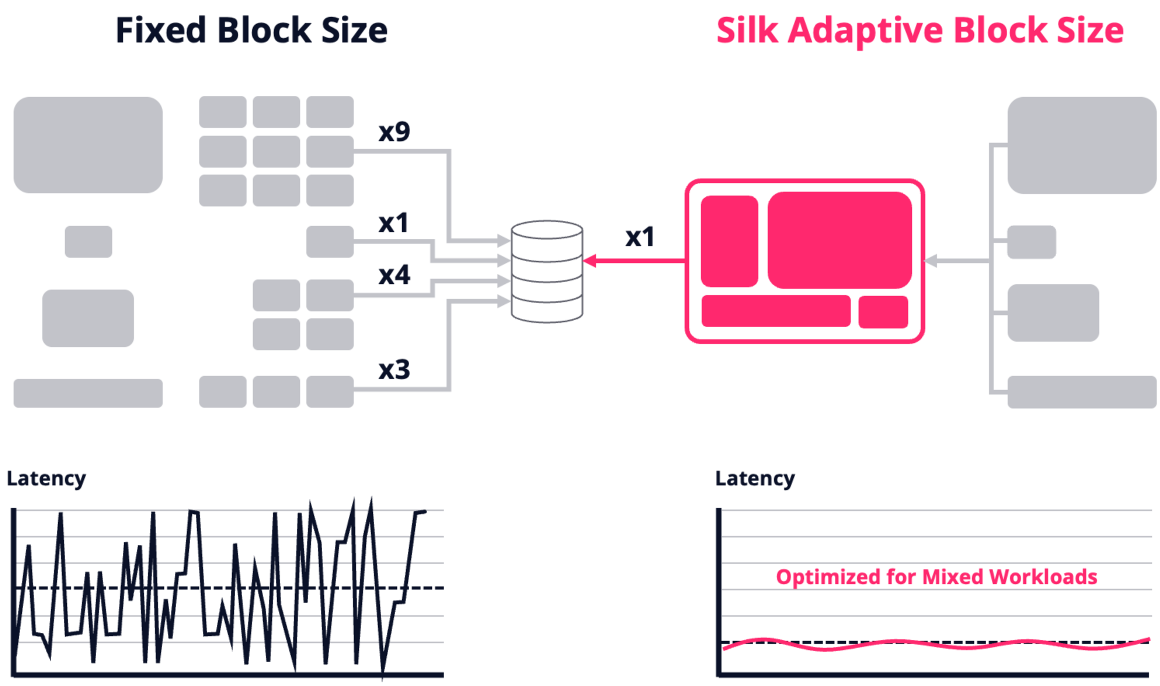
- Maximizing performance by automatically adjusting host connectivity as c.nodes are added and removed



Using Global Adaptive Block Size

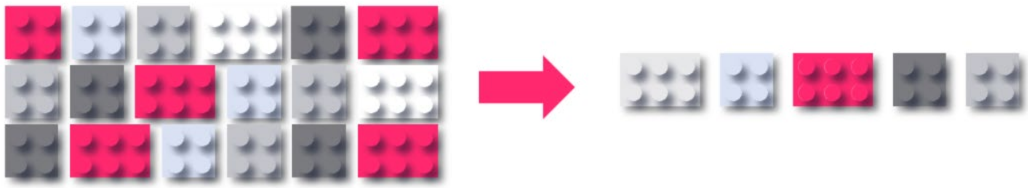
Workloads generated by real applications vary in their block size. Resources –specifically those that utilize deduplication– tend to use a fixed block size of 4KB, 8KB, 16KB, or larger. However, fixed block sizes can't offer the best performance for environments with fluctuating amounts of data, creating an issue known as the IO blender effect. Smaller IO operations may require padding a fixed block with zeros to fill it, while larger IO operations may require data to be fragmented across multiple blocks, resulting in limited bandwidth and an unnecessary increase in the number of IO operations. Silk developed an algorithm that eliminates the IO blender effect from virtual servers, enabling your cloud environment to run smoother and more efficiently.

SDPs let customers maximize data performance with dynamic block sizes that adapt to an application's real workload, without compromising latency, IOPS, or throughput. Silk's global adaptive block size algorithm allows the platform to support the performance requirements of multiple application types running concurrently, as well as multiple BI workloads running simultaneously. It uses Silk's performance layer to receive and copy data for redundancy, combining smaller blocks of data into a larger block that's indexed into 4KB segments so there's no wasted space. When the block is full, the data is written to the d.nodes in your SDP. This patented algorithm plays a significant role in helping Silk's customers scale out cloud resources.



Deduplicating Your Data

SDPs use global inline deduplication to eliminate redundant data, so your data is only stored once. By default, deduplication is performed globally and processing is distributed across all c.nodes in your SDPs, enabling higher deduplication ratios, high performance, and consistently low latency. Our deduplication process scales with your environment as you increase the number of c.nodes in your SDPs, without requiring you to manually move any data.



Selective deduplication is also available for database applications like Oracle or SQL Server. This allows data to be stored without deduplication for applications with negligible data redundancy, where additional performance is needed instead.

Maximizing Low Latency or High Throughput with Read Boosts

Silk lets you use one of two custom optimization features in your environment based on your performance needs. You can choose to reduce latency with read transaction boost **or** enhance throughput with read throughput boost.

Read transaction boost: Transaction boost adds read caching in your c.nodes, so your most accessed data is stored at the c.node-level and is sent directly back to your database. In your SDPs, reads via disk are replaced with reads via cache, which significantly reduces latency and improves IOPS performance. Our large cache ensures a high hit rate, and any cache misses are redirected to the m.node and then written to the cache. Data is written to the cache and the m.node simultaneously, without any additional processing time from the database.

Read throughput boost: Throughput boost works via remote metadata reads, so each c.node in your SDP keeps a mapping of a portion of the data sent to your m.nodes to optimize your backend data flow. This greatly reduces the amount of data sent on reads and allows c.nodes to communicate directly when read requests are received, which can improve bandwidth and throughput in your environment by up to 60%.

Running Your Environment with Silk on VMs

The Silk platform uses m.nodes for capacity within SDPs, built with either VMs for peak performance (available for all cloud providers) or managed disks for peak resilience (currently available for Azure), offering two distinct ways to achieve greater reliability in your cloud environment.

Silk on VMs allows you to use traditional storage resources in a more flexible way. m.nodes built on VMs are connected through compute networking instead of storage networking to significantly enhance performance and optimize speed within your environment. These m.nodes also use attached storage disks to reduce latency overall and eliminate internal latency within the Silk platform. Each VM-based m.node contains 16 d.nodes in a 14+2 configuration, reserving two nodes for data redundancy. Your d.nodes are supported by RAID volumes for parity and protection against data loss. Use Silk on VMs to increase your environment's speed and performance while maintaining data resiliency and security.

Silk combines native cloud solutions with our own industry-leading technology to unlock new levels of performance within your cloud environment. Our scalable architecture works with features like global adaptive block sizing, deduplication, and read transaction boosts or read throughput boosts to ensure you achieve high

IOPS and throughput and consistently low latency. Running Silk on VMs takes advantage of compute networking and attached disks to increase your environment's speed and support the highest levels of performance. Customers have used Silk to achieve more than 14.6GB/s of throughput from a single SDP. Use Silk to enhance your cloud environment's performance far beyond anything native solutions alone can offer.

Resiliency

The Silk Cloud Data Platform is built for the resilience needs of business-critical enterprise applications. Designed with high availability in mind, Silk supports fault domain scaling and provides a rich feature set for maintaining enterprise products and protecting valuable application data.

The platform offers excellent resiliency to ensure your cloud environment doesn't experience any unplanned downtime or data loss. If any c.node in an SDP fails, your environment's performance decreases linearly as a percentage of the total number of active c.nodes, thereby minimizing performance impact. Likewise, if a d.node fails and requires a rebuild or replacement, that work is distributed across all c.nodes to significantly shorten rebuild times. Silk employs features like replication, metadata management, and non-disruptive upgrades to enhance data resiliency, and our cloud-agnostic framework uses RAID volumes and active-active clustering (ensuring no single point of failure) to support full redundancy.

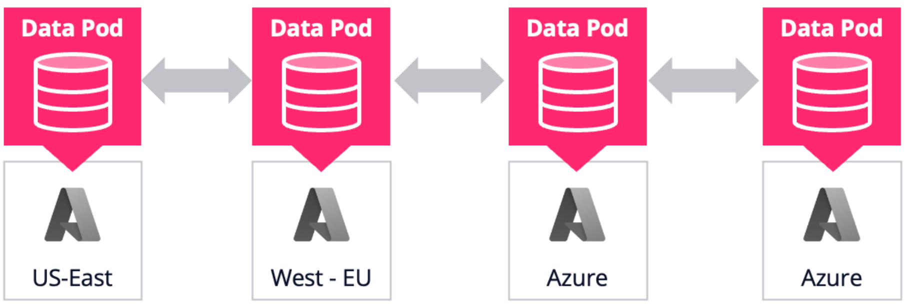
Promoting High Availability

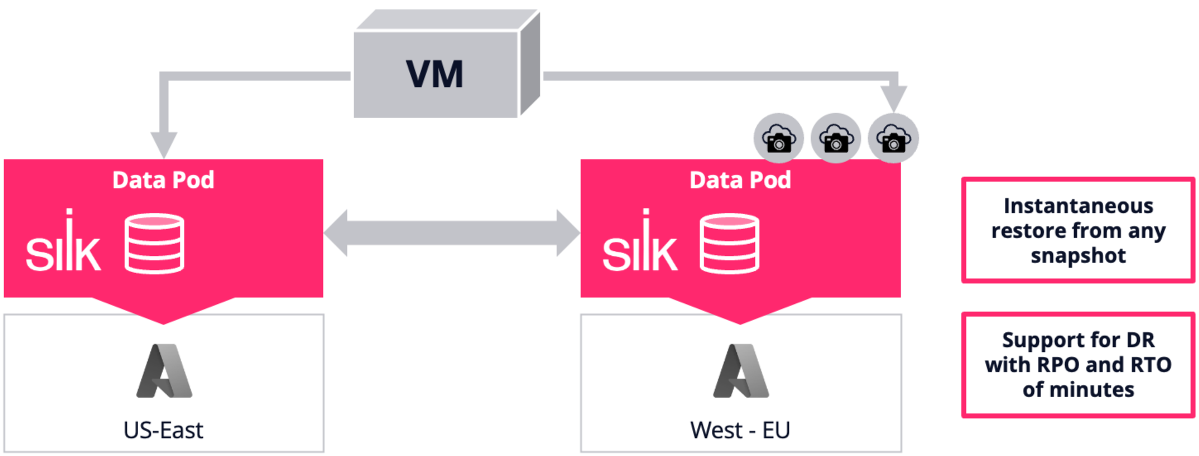
Silk SDPs are highly available and fault tolerant to ensure your data is always accessible. Our non-disruptive upgrades and scalable architecture work together to optimize cloud costs by eliminating fork-lift upgrades and planned downtime. In our active-active framework, each layer of the Silk stack supports at least one node failure while ensuring continuity of service. This means we can automatically upgrade any SDP components without impacting your environment's availability or performance, and without human intervention. We can also deploy enhancements and updates in existing SDPs without dependencies on maintenance windows or running workloads. The platform uses logic that works around predictable events in the cloud by anticipating planned updates from cloud providers. That same logic supports rapid response and resiliency to help resources recover from unpredictable events.

Our active-active architecture utilizes all the platform's resources at all times, with no passive or idle components. We provide full redundancy of every component in the system –both virtualized and physical– and there is not any single component failure that can cause unplanned downtime or data loss. Each SDP is configured and installed with a threshold defining the minimum number of online c.nodes required to operate your environment, and your SDPs can sustain any c.node losses above that predefined minimum while remaining online. Checkpointing is also available to aid in disaster recovery for VM-based Google Cloud instances and managed disk-based Azure instances. Checkpoints are a configurable resource stored on d.nodes, so there is no performance impact and little to no additional cost.

Supporting Disaster Recovery through Replication

The Silk Cloud Data Platform supports enterprise datacenter resiliency and disaster recovery through replication. Silk facilitates asynchronous, snapshot-based replication between SDPs. Since the replication is based on snapshots, there is no impact on your dataset's consistent high performance. Silk replication can copy data across zones and has the flexibility to replicate to different destinations.





All essential disaster recovery capabilities, including recovery point objectives (RPO) and recovery time objectives (RTO) are gained natively through your Silk implementation, without any costly third-party software components adding overhead on the data flow. Additionally, Silk supports database stacks using common database replication solutions like Oracle Data Guard and Microsoft SQL Always On. Silk replication utilizes multiple connections and data compression to enable extreme levels of throughput in cloud environments. Customers have achieved replication bandwidth of over 10Gbits/s of physical data, enabling rates of over 50Gbits/s of logical data transfer (depending on reduction rate).

Silk offers one-to-one replication to let you replicate volume group (VG) data from one SDP to another, or one-to-many replication to let you replicate data from one database to additional target environments for testing or backup and recovery. The platform support two types of one-to-many replication structures for flexibility and high availability in both production and non-production environments:

- **Concurrent replication:** Replicate data to two destinations from the same source in parallel.

- **Cascading replication:** Replicate from one destination to another, and then replicate from the second destination to a third. This type of replication is even more parallel than concurrent replication, as it is distributed further across multiple SDPs.

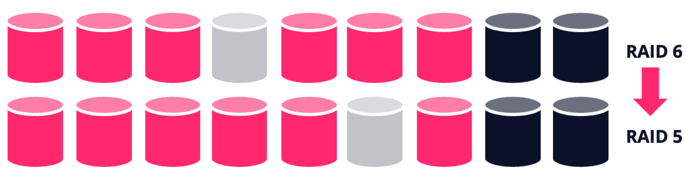
In addition, Silk supports **one-way replication** and **disaster recovery (DR) replication**. One-way replication lets you give data to workloads in non-production environments to allow access to the data without granting access to the production volume. Disaster recovery replication allows you to switch from your primary SDP to a secondary SDP that's receiving DR data to make the data rapidly available in a DR scenario.

Silk's multifaceted replication feature enables a sophisticated replication strategy for various business use cases.

Protecting Data with Distributed Parity

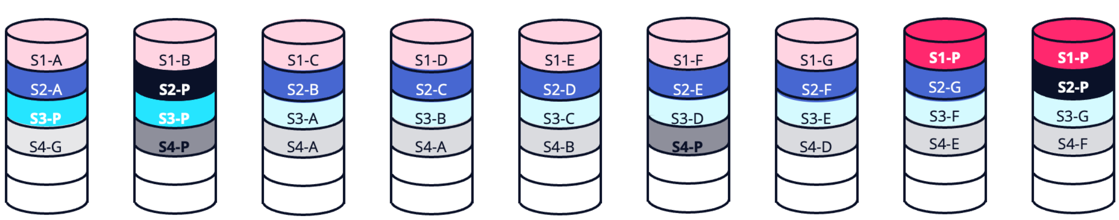
To ensure high fault tolerance and resiliency, metadata at rest is protected by dual-parity RAID volumes. Whether your environment runs on VMs or managed disks, it is always configured with two parity nodes for data redundancy. VM-based instances use 16 d.nodes in a 14+2 configuration, while managed disk-based instances use 9 d.nodes in a 7+2 configuration.

Our m.nodes use a RAID 6 resiliency structure by default. If any d.node in your environment fails, the m.node switches to a RAID 5 scheme until your failed d.nodes are restored, protecting your data durability. Both RAID 5 and RAID 6 use striped blocks to spread data for a single volume over two or more d.nodes. When reading from and writing to a large file, the request is sent to all d.nodes at the same time, which allows the d.nodes to work in parallel and reduces the time for reads and writes.



To further enhance disaster recovery for your environment, our architecture supports self-healing measures like fast failback and fast rebuilds in case of short-term d.node outages. Your Silk implementation can sustain two concurrent d.node failures while maintaining data availability. RAID stripes are pseudo-randomly distributed throughout each m.node to avoid fixed parity or data roles within your environment, meaning that all d.nodes offer consistent data protection and performance. Silk's RAID configuration is automatically applied to your environment and doesn't require any manual setup, which offloads resiliency tasks from your IT team.

For more details on how Silk works with RAID volumes, see Appendix B at the end of this document.



Parity nodes in each stripe are indicated with a darker shade and white bolded text

Managing Metadata

Metadata management is essential for any data system, but its importance grows tenfold when you deploy a data system that supports a scalable architecture and features like snapshots, deduplication, and replication. The way Silk's SDPs manage metadata enhances deduplication and compression performance while allowing scale-out, active-active access to all c.node volumes and snapshots. It also facilitates fast recovery in the event of failures and provides an optimized garbage collection process.

Metadata is kept both on DRAM and NVMe flash media, using a unique cache algorithm for consistent performance. Silk's patented approach to distributing metadata across c.nodes provides parallel processing and redundancy to maintain performance and reinforce your environment's resiliency.

Running Your Environment with Silk on Managed Disks

The Silk platform works with Azure SSD Premium v2 to offer a simplified SDP architecture using shared, persistent SSDs. In this framework, m.nodes use 9 managed disk-based d.nodes in a 7+2 configuration to ensure high performance with better durability and multiple layers of resiliency. Managed disks are stored in triplicate at the d.node level so your environment can sustain up to two simultaneous SSD failures for each individual node, thereby boosting disk reliability and durability. We also proactively monitor and address known Azure updates to avoid disruptive maintenance waves, and our checkpointing feature preserves data even during zone outages. Run Silk on managed disks to take full advantage of our suite of services, including compression, deduplication, instant snapshots, thin provisioning, replication, and encryption at rest.

Silk supports highly resilient and available cloud solutions for any type of cloud implementation. Our set of resiliency features –including replication, distributed parity, self-healing, and metadata management– help you build and maintain a cloud environment that's fully redundant, fault tolerant, and protected at every level. Running Silk on managed disks provides additional layers of resiliency and durability to further reinforce every piece of data in your dataset. Our platform keeps your business-critical applications running systematically and effectively while reducing the amount of hands-on management needed, allowing you to spend less time worrying about your data and more time innovating for your business.

Data Lifecycle Management

We don't just help you store your data or migrate it to the cloud – we give you the tools to optimize and manage your data at every step of your cloud journey. We developed a host of features called Enterprise Data Services to shrink your capacity footprint and simplify data sharing between systems. Our simple-yet-effective user interface lets you visualize and control the flow of your data quickly and easily. We keep your data safe with security features like encryption, SAML for SSO, and CHAP. Silk gives you the tools you need to make data management easy, streamlined, and scalable.

Using Enterprise Data Services

Silk offers performance and cost efficiency through our Enterprise Data Services, which include deduplication, compression, zero elimination, thin provisioning, and snapshots. These features enable highly efficient SDPs without compromising on enterprise resiliency, and they play a major role in Silk's IO processing.

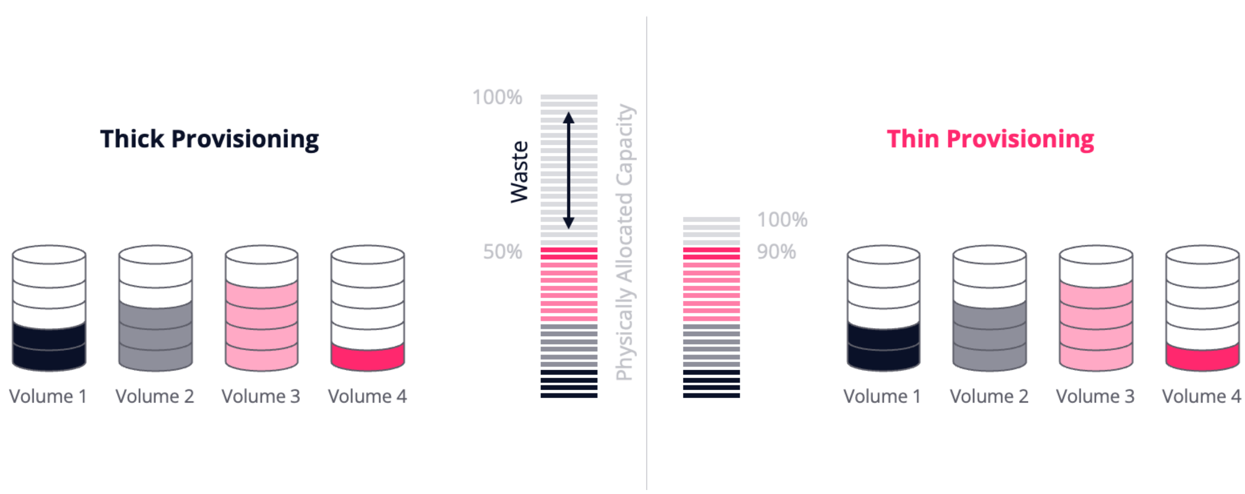
Deduplication: Improve your environment's performance with global inline deduplication for all applications or selective deduplication to specify which applications don't require deduplication. For more details, see our section on deduplicating your data here.

Compression: SDPs use inline, real-time data compression optimized for low-latency performance that significantly reduces the amount of data sent between SDPs. When enabled, compression can occur with or independent from deduplication, even in non-dedupable datasets common in database environments like Oracle and Microsoft SQL Server.

SDPs use byte-aligned compression algorithms that can compress data into a single byte, achieving greater compression ratios and eliminating segment padding. Our compression process places a metadata marker on every 4KB block, rather than on larger data segments, ensuring that small reads don't cause unnecessary data hydration. The output of a compressed 4KB block is rounded to the nearest byte, and this byte-aligned compression prevents internal fragmentation and facilitates better compression ratios.

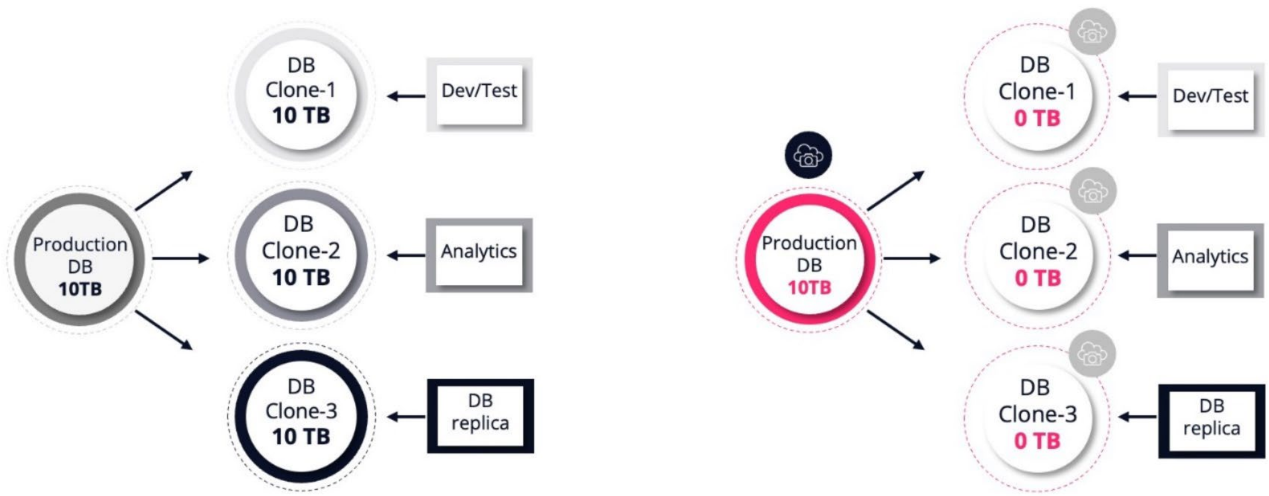
Zero Elimination: Many applications either format data on initialization or store large sections of zeros as part of the data. SDPs use inline zero elimination to avoid storing data that consists of zeros. Instead, the system stores a metadata flag to indicate that a certain data address is logical zero. Zero elimination is fine-grained in 4KB blocks, which ensures that applications won't consume valuable space within the data pod and enables faster data formatting.

Thin provisioning: Thin provisioning allows maximum resource utilization, with the ability to plan capacity provisioning for the long term, by allocating disk space on demand as needed. By default, resources in the cloud are thick provisioned and don't support native thin provisioning, but Silk's scalable architecture mitigates this limitation. All the volumes in Silk SDPs are thin provisioned, regardless of the underlying cloud infrastructure. Our provisioning capabilities offer granular, on-demand growth of 4KB blocks and UNMAP operations. SDPs make thin provisioning effective and enable easy, hassle-free capacity management for volume provisioning.



Snapshots: Silk's patented snapshot architecture is efficient, high-performing, and scalable. Snapshots are created instantly, with no performance impact, and they only use capacity on writes. They track only the deltas from the source volume in 4KB blocks using a redirect-on-write (RoW) approach, minimizing required capacity. Snapshots can be mounted for read/write purposes, which lets you create additional working environments for QA, test and dev, analytics, backup, etc., at an extremely low cost. Read/write snapshots deliver the same performance as production volumes and can be manipulated without impacting production volumes.

Snapshot creation has no dependencies on the number or size of the volumes being snapped or on the size of your SDP. Use snapshots to restore functionality for recovery purposes, without losing any of your snapshot's history, at any time. Snapshots are accessible from any c.node in your SDP, without bottlenecks or load balancing affinity to a specific controller. Silk's snapshot feature streamlines data use across your organization, allowing your business to scale easily while reducing operational costs.



Managing Your Silk Environment

SDPs provide a rich set of tools to monitor and manage your cloud environment. These management tools don't require additional hardware resources since they're designed to natively run on your provisioned c.nodes. With Silk's data management tools, there's no need to configure RAID groups, tune your SDPs to a specific application, or create affinities between volumes and controllers.

Manage your SDPs through Silk's web-based user interface, which doesn't require any software installation. Your management options are displayed in a sleek dashboard that lets you see all your environment's key metrics at a glance, including capacity, events, and performance. Performance indicators (throughput, IOPs, and latency) show your current performance averages and potential maximums based on your Silk configuration. Capacity indicators show your current capacity usage, data reduction scale, and volume details. Use the tabs along the top of the interface to open more detailed and actionable views of your environment. The platform's GUI lets you manage your SDPs by creating volumes, snapshots, and replication/capacity policies, assigning hosts, managing host connections, etc.

The CLI is fully scriptable and allows full control of the platform's features and functionality. The CLI is accessible via SSH using a Linux/Unix shell or PuTTY-like utility.

The platform has a full SDK that supports Microsoft PowerShell, including cmdlets for all native SDP functions. We've also written a Terraform provider that supports the main system user functions needed for common workflows and can easily be extended to support any functionality. The Terraform provider is available at https://github.com/silk-us/terraform-provider-silk. Likewise, we support Datadog integrations for monitoring performance data and performing service checks through the Datadog Agent.

Silk exposes a full set of RESTful APIs that allows the system to be controlled, managed, and monitored via third-party software and in-house management platforms. The Silk REST API is publicly available at https://github.com/silk-us/silk-sdp-api-docs, with examples on how to develop automatic flows. Protocols like SNMP, SMI-S, and Syslog are supported through third-party platforms. Our APIs also allow third-party components and independent software vendors to monitor and manage your SDPs. We support integrations with some of the leading vendors in the enterprise IT industry, including:

- **Containers Storage Interface (CSI) plugin:** The CSI plugin is a native interface between containers like Kubernetes and the Silk Cloud Data Platform. The plugin is easy to install and use, and it's based on industry-standard software infrastructures.

- **OpenStack:** OpenStack's agile and scalable cloud deployments pair well with Silk's architecture. The Cinder driver leverages Silk's APIs to let OpenStack environments provision all-flash datasets.

Securing Your Data

Our platform's cybersecurity features keep your mission-critical systems safe. We use encryption, SAML-based authentication, and CHAP authentication to protect your data.

- **Encryption:** We take advantage of your cloud provider's native encryption at rest and, where available, networking encryption capabilities.

- **SAML for SSO:** Use SAML for SSO as a single authentication tool for Silk Flex instances and SDPs. Silk users are authenticated through an Okta Identity Provider (IdP) or through Microsoft Entra ID. This lets users access Flex and their SDPs easily by logging in once, offering greater security, more control, and ease of use.

- **CHAP:** CHAP authenticates communication between a host and your SDPs, allowing users to authenticate without exposing their password. This protocol supports organizational security policies and locks out potential intruders.

At Silk, we know the value of continuous data management. Moving to the cloud is just the first step in the cloud journey – managing your data in the cloud is the true key to executing an effective cloud solution. Our platform is purpose-built around Enterprise Data Services, environment management, and security best practices to help you realize your data's full potential and keep your environment running as smoothly as possible.

Cost Efficiency

Silk consistently innovates to ensure our platform offers high price performance and the smallest TCO possible, depending on your needs. The cost benefits of using Silk are twofold: you can reduce your native cloud spend **and** achieve cost efficiency within the Silk Cloud Data Platform. Performance pooling maximizes workload efficiency and minimizes resource cost, while our Enterprise Data Services shrink your data and the cost of capacity. Our architecture helps you avoid overprovisioning host VMs and database licenses, and our technology improves data management so you can provision resources precisely and flexibly.

Provisioning Resources Dynamically

Silk lets you provision one platform that runs multiple databases concurrently, rather than the traditional cloud model of running separate databases with separate costs. Use performance pooling to allocate resources where they're needed and enable multiple workloads to use the same SDP, and dynamically provision and decommission resources to ensure you never pay for resources you don't need. SDPs are designed to help minimize your data footprint and its associated costs with enterprise data service features like snapshots. Our compression, zero elimination, and deduplication components work together to provide not only the best possible performance and data management, but also the most cost efficiency. Compared with native cloud solutions, the Silk Cloud Data Platform's TCO includes stronger price performance, database volume performance, scalability, reliability, and fine-grained control over your data management.

Choosing the Right Infrastructure

Whether you run your Silk environment on VMs or on managed disks, you can benefit from the platform's cost-efficient design. Silk on VMs takes advantage of the inherent cost efficiency of VMs, since they require fewer hardware resources, while Silk on managed disks has a lower baseline infrastructure cost for m.nodes, thereby reducing the TCO of your Silk environment. Our ability to reduce VM overprovisioning coupled with our replication capabilities helps you avoid paying for excess, unused database licenses. The platform's scalable architecture uses rightsized host VMs and pooled performance to help you minimize infrastructure costs both within Silk and within your native cloud solution. Our focus on performance and capacity means that features like compression, thin provisioning, and zero-footprint snapshots work to minimize your data footprint, allowing you to achieve economies of scale at a lower price point.

Cloud migration and management isn't a one-size-fits-all solution, and tailoring your cloud deployment to your business-critical needs can come at a high cost. Silk is built to give you more ways to reduce your cloud spend without sacrificing performance or scalability. Both our infrastructure and our services allow you to create a cloud solution that's designed around your business needs and your budget.

Summary

The Silk Cloud Data Platform improves cloud performance, simplifies cloud management, and reduces cloud cost and complexity. In many cases, Silk is the sole cloud solution that can enable you to achieve the performance, resiliency, and manageability that your business needs. We offer up to 26 GB/s of throughput or 2.1M IOPS per Silk Data Pod and latency that's consistently under 1ms for unmatched levels of performance. The platform is redundant at every layer for high availability and resiliency that grows along with your environment. We're the only solution that offers enterprise data services like thin provisioning, built-in compression and deduplication, and instantaneous snapshots to optimize your data management capabilities. Use Silk to make your cloud migration and maintenance simpler, more efficient, and more dynamically tailored to your needs.

Silk

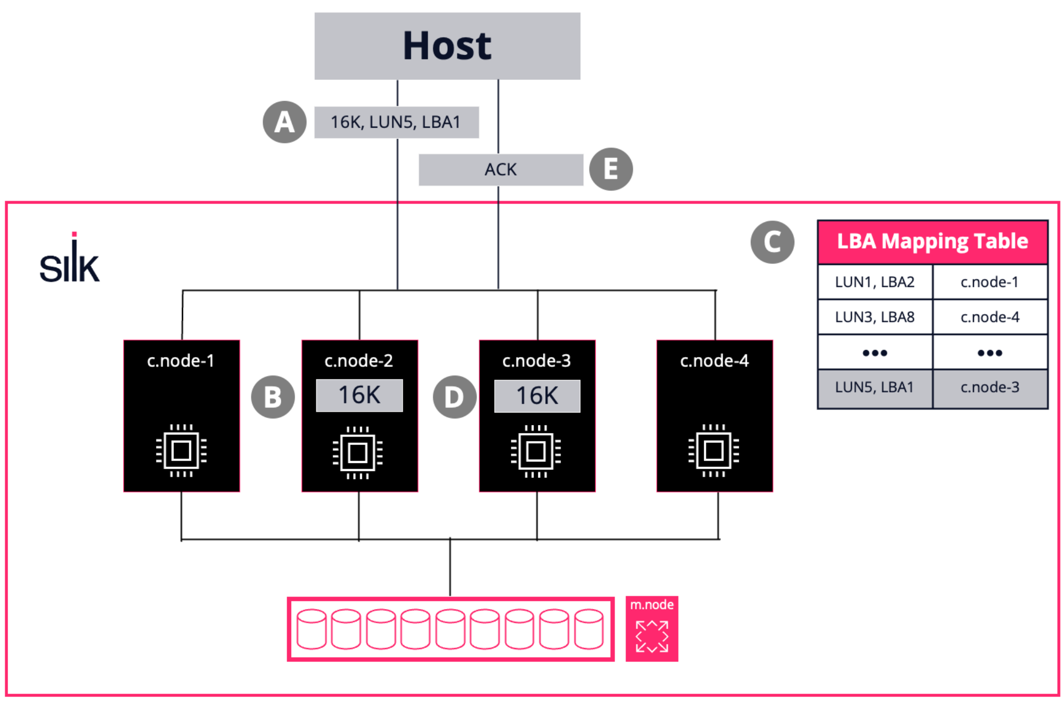
Appendix

Appendix A: Understanding Silk's IO Flow

The following sample configuration demonstrates a simplified IO read/write flow, showing how Silk's global adaptive block size algorithm works to scale out a real application load. This sample flow uses a 16KB IO operation from a host, but also works with other IO sizes. The IO size is a multiple of the volume's logical block size. When hundreds of thousands of IO operations run concurrently through the platform, additional features like batching and queuing also take place to provide further performance optimization.

1. Unique Write: IO is Stored and Acknowledged

- **A.** An application writes a typical 16KB IO to a logical block address (**LBA1**) within a block device (**LUN5**) on an SDP. This data can arrive at any c.node within the SDP because Silk's data pods use a scale-out, active-active scheme. In this example, the data arrived at **c.node-2**.
- **B.** After the data is stored in **c.node-2**, it must also be stored in a second c.node before returning an acknowledgement (**ACK**) to the application.
- **C.** The second c.node is selected according to a mapping table that maps the LUN and LBAs of the incoming write to a specific c.node. This mapping table is identical in all the c.nodes within an SDP. In this example, **c.node-3** is selected.
- **D.** The data is mirrored to, and stored in, **c.node-3**.
- **E.** **c.node-2** can now return an **ACK** to the application's host. Any subsequent work performed while storing the new data is asynchronous and allows for low host-side latencies.

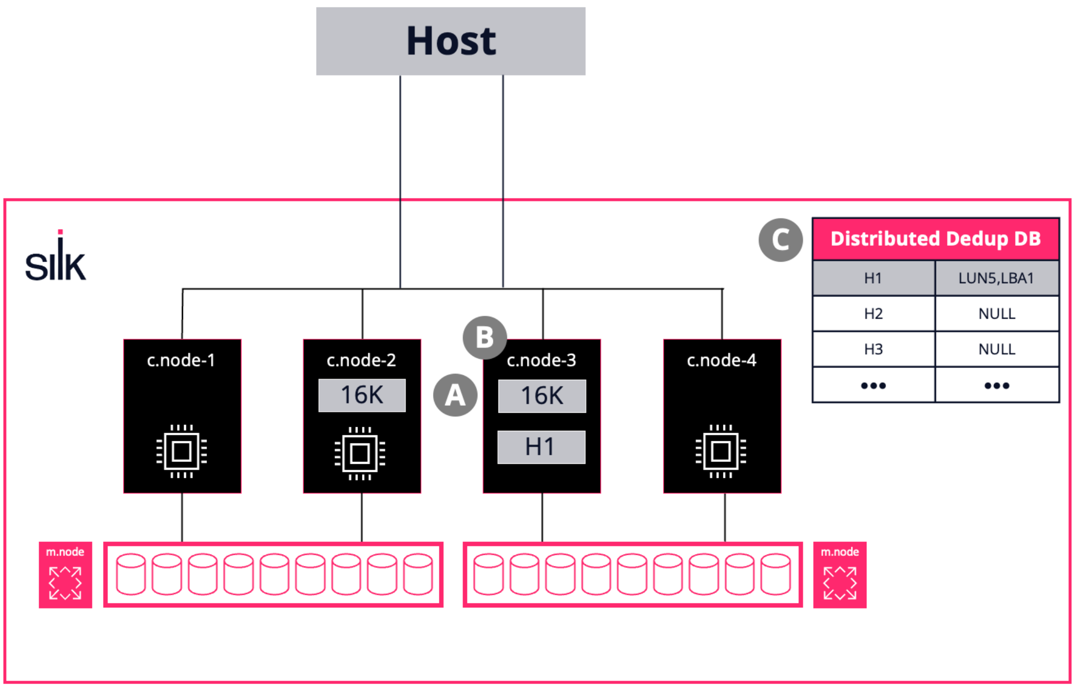


2. Unique Write: Hash Table is Updated

A. The deduplication process starts on **c.node-3**. Deduplication is distributed. A hash value **H1** of the data is created. All possible hash values are divided (sharded) into several hash tables, according to the number of c.nodes in the SDP.

B. The hash value is sent to the appropriate c.node and a lookup is performed for **H1**. This will indicate whether the data was already written to the system. Since this is a unique write to the SDP, the hash lookup returns negative.

The c.node that originated the hash lookups (in this example, **c.node-3**), takes ownership of the data. During the hash lookup, the c.node that was queried updates its hash table with the owner of the 16KB data (**LUN5**, **LBA1**).



![Silk logo]

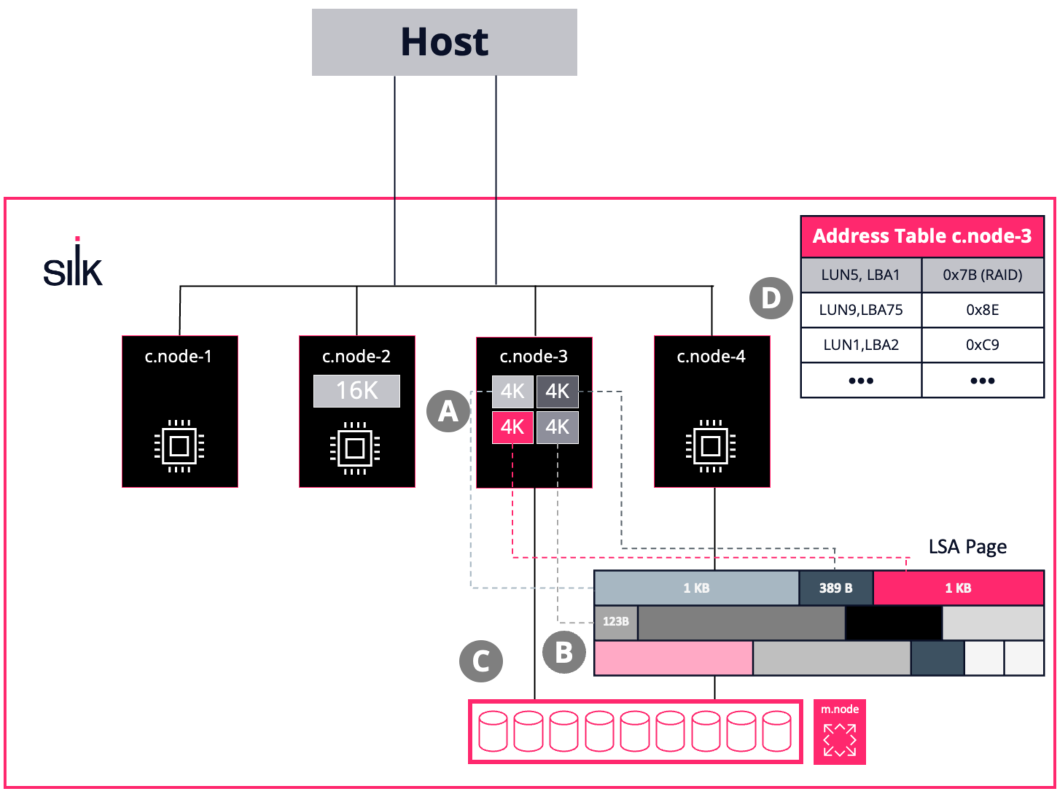
3. Unique Write: Data is Split and Compressed

A. The 16KB is split into 4KB segments before being compressed. Each 4KB segment is compressed separately to the nearest byte using byte-aligned compression to optimize space required and future read operations.

B. The compressed 16KB of data is placed contiguously in a 3MB log-structured array (LSA) page.

C. After the LSA page is filled, **c.node-3** prepares a full RAID stripe, calculates the parity, and writes it to the RAID volume.

D. **c.node-3** keeps the location of the compressed 16KB in its address table, which translates **LUN, LBA** pairs to physical addresses on the RAID volume.



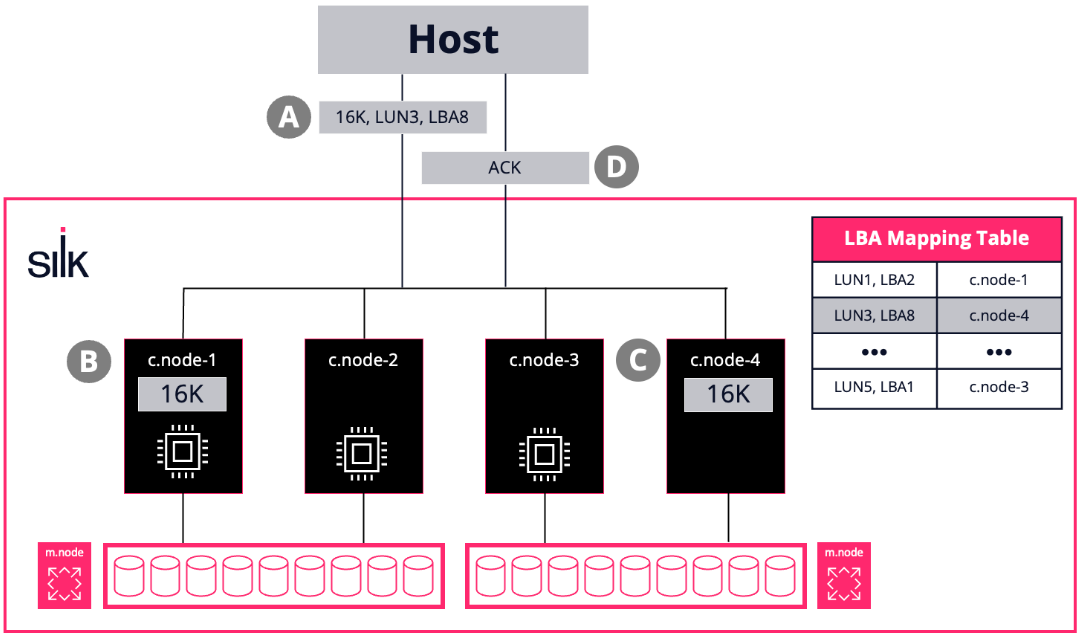
4. Variable Deduplication Write: Data is Stored and Acknowledged

A. The same 16KB IO is written again to the SDP, with the write designated for **LUN3, LBA8**.

B. The IO arrives at **c.node-1** and is stored.

C. The mapping table indicates that the write should be mirrored to **c.node-4**, so the block is mirrored to **c.node-4**.

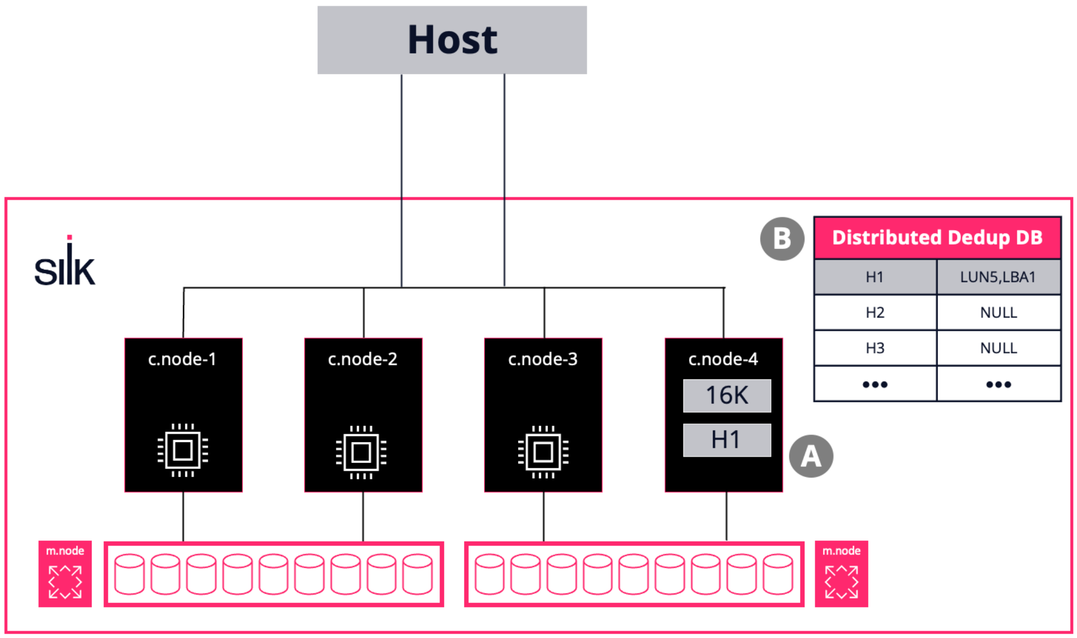
D. The data is now stored in two different c.nodes, and **c.node-1** can return an **ACK** to the application's host.



5. **Variable Deduplication Write: Hash Lookup is Performed**

A. On **c.node-4**, a hash value of **H1** is created for the 16KB. The hash value is sent to the appropriate c.node (in this example, **c.node-1**), and a lookup is performed in the hash table.

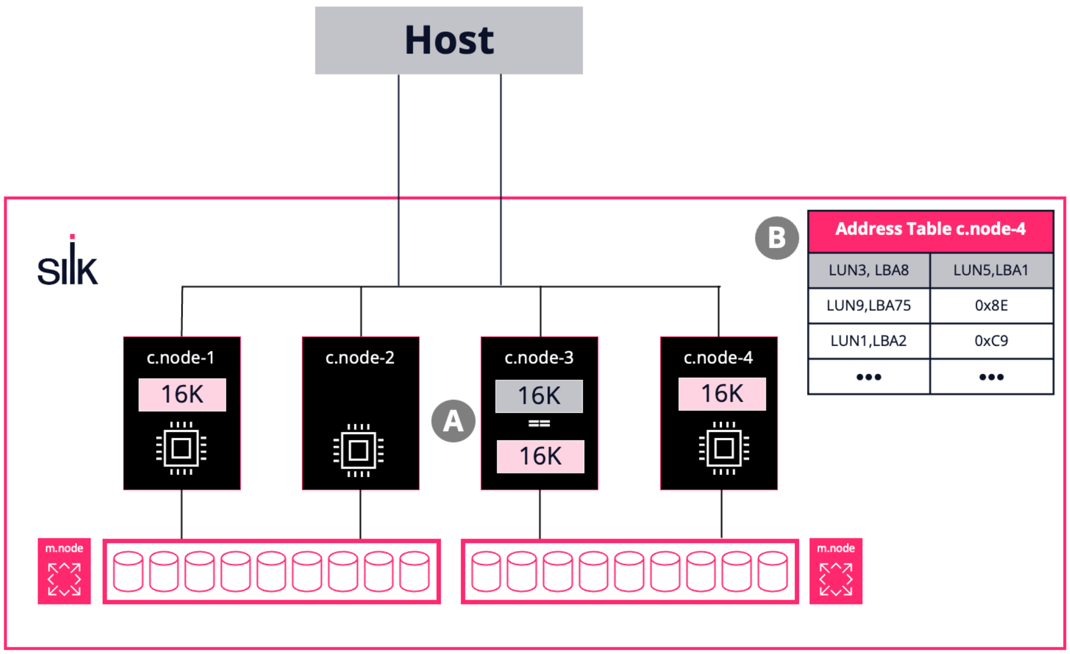
B. **c.node-1** reports back that the hash value was queried with the address that appears as **LUN5, LBA1**.



6. Variable Deduplication Write: Data is Compared

A. To avoid hash collision, the actual data must be compared. **c.node-4** sends the 16KB of data to **c.node-3** for a full comparison. This requires a single read operation by **c.node-3** from the RAID volume, since it was stored as a single 16KB.

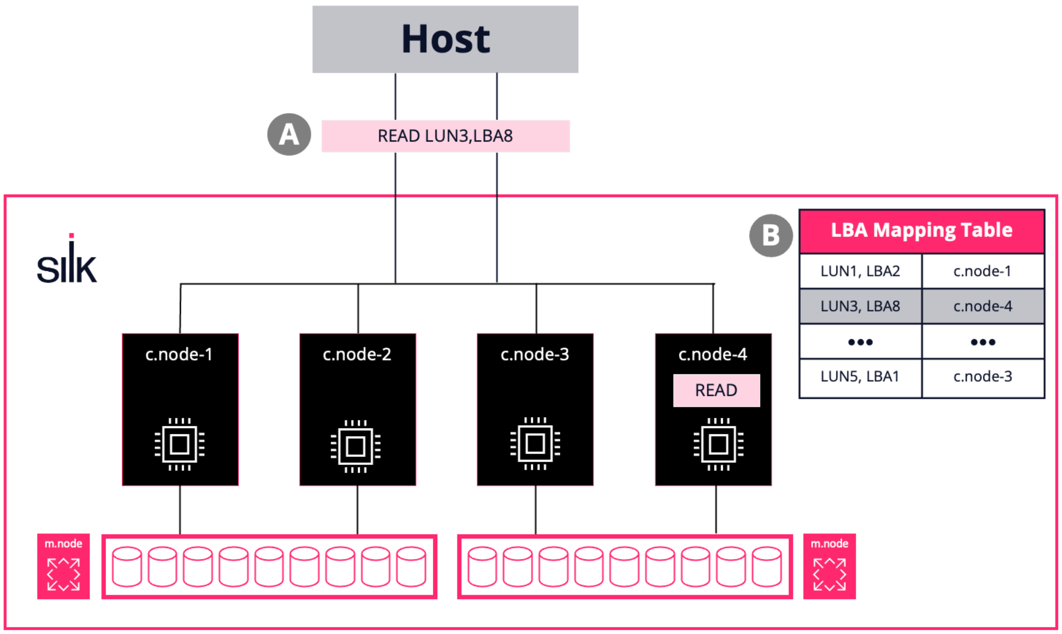
B. If the data is identical after it is compared, **c.node-4** updates its address table at **LUN3, LBA8** to point to **LUN5, LBA1**. These metadata updates are mirrored between c.nodes and eventually de-staged to the RAID volume, so metadata is kept highly available at all times. **c.node-1** and **c.node-4** can now free the 16KB from their DRAM.



7. Read: Mapping Table Lookup is Performed

A. The application reads 16KB from **LUN3, LBA8**.

B. This read request can arrive at any c.node. In this example, it is received by **c.node-4**, which performs a lookup in its mapping table for the owner of **LUN3, LBA8**. According to the mapping table lookup, **c.node-4** is also the owner of **LUN3, LBA8**.



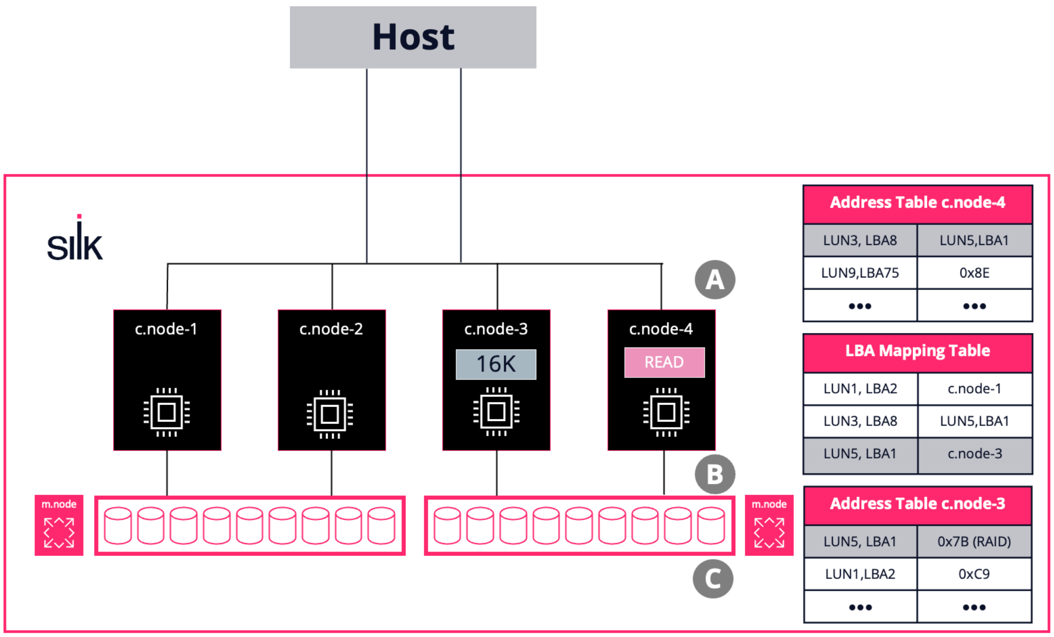
8. Read: Data is Retrieved and Compressed

A. **c.node-4** looks up **LUN3, LBA8** in its address table and finds reference to a logical address (**LUN5, LBA1**) rather than a physical address.

B. **c.node-4** uses the mapping table to request the data from **c.node-3**, the owner of **LUN5, LBA1**.

C. **c.node-3** looks up the physical address.

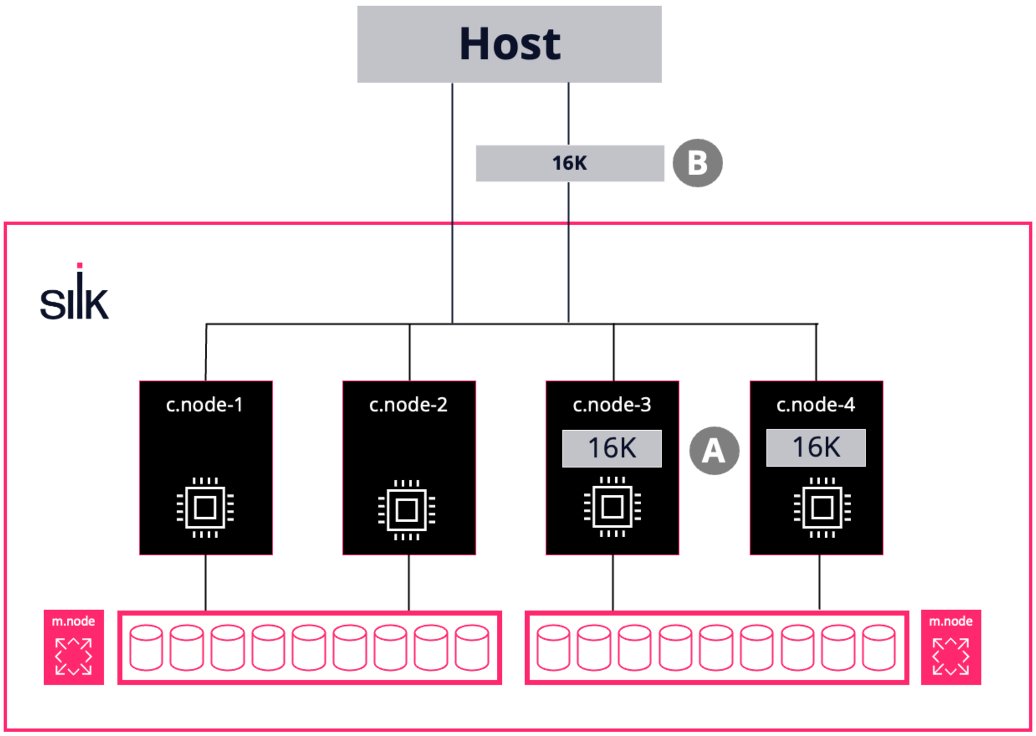
D. **c.node-3** sends the physical address back to **c.node-4**.



9. Read: Data is Sent to the Application's Host

A. c.node-4 retrieves and decompresses the data.

B. c.node-4 sends the requested 16KB back to the application's host.



Appendix B: Understanding Silk's RAID Structure

The Silk RAID parity protection algorithm is both efficient and extremely robust. It can sustain two concurrent d.node failures while maintaining data availability. As an SDP scales capacity, the number of system-wide d.node failures the system can sustain also scales. RAID volumes have triple parity protection that adapt according to the failure scenario. Any d.node failure is quickly recovered thanks to efficient metadata management and real-time system health monitoring. There is minimal performance impact during rebuilds and no impact to Silk's performance once rebuilds are complete. Parity overhead from raw capacity is a very small 12.5%.

Here is an example of how RAID works with d.node failures:

1. After the d.node fails, the RAID volume switches to RAID 5 and a rebuild is performed from the other healthy d.nodes. This method results in considerably faster rebuild times.
2. XOR calculations are performed from the affected RAID group and the parity segment takes over the role of the lost data segment. There is no performance hit when reading from a RAID group that lost a parity segment.
3. After the rebuild is complete, the RAID volume switches back to RAID 6, ensuring the volume is still capable of handling two concurrent d.node failures without data loss.
4. After the failed d.node is restored, the RAID volume is restored to its original structure.