INTEL® SOLUTIONS REFERENCE ARCHITECTURE

Converging Enterprise Storage and Business Intelligence: A Reference Architecture for Data-Centric Infrastructure

COHO DATA

Audience and Purpose

Many corporate environments are faced with supporting both new business intelligence (BI) services such as Apache Hadoop* and Spark* in addition to more traditional facilities for compute and storage. Enterprise IT Organizations, Cloud Hosting Providers, and Cloud Service Providers have all explored the deployment of BI solutions in various forms, and have raised a common set of concerns. When BI solutions are implemented in isolation, experience shows that they are expensive to deploy, add environmental complexity, and integrate poorly with existing and mature facilities for data durability and failure recovery. Business intelligence stacks have matured and offer rich new capabilities for enterprises that deploy them, but they also present a burden to IT organizations as a new and additional "silo" to manage.

This reference architecture demonstrates a data-centric storage platform that deeply integrates hosted business intelligence capabilities directly within a scale-out enterprise storage system. Through a customer implementation at a Fortune 100 financial institution, that has solved a number of problems that have traditionally been associated with Enterprise BI deployments, it outlines the deployment of Coho Data's DataStream* storage nodes with an internal framework for the Docker*-based hosting of Cloudera CDH5* analytics software. This reference architecture brings parallel computation "into the fold" as a first-class and fully integrated aspect of enterprise computing infrastructure.

Andrew Warfield
Coho Data

David Cohen
Intel Corporation

James Younan
UBS

August 2015 (v1.0)

Executive Summary

In today's rapidly changing business environment, companies are generating data at unprecedented rates. Businesses that succeed in their ability to analyze and act on the results of the analysis are able to create competitive differentiation and enter into new markets. Business Intelligence (BI) solutions, such as Cloudera Enterprise CDH5* distribution, represent invaluable tools in analyzing, interpreting, and taking action on organizational data.

Unfortunately, BI environments are frequently deployed as one or more independent silos within enterprise deployments, using independent and dedicated physical hardware. For many customers, this model of analytics deployments presents three specific concerns:

1. **Analytics sprawl.** Where analytics purchases are ad hoc within an organization, one or more smaller deployments may be installed outside the purview of the core IT organization. The "sprawl" of analytics environments results in inefficiency, as data must be copied between traditional enterprise storage and the analytics environment, often on a regular basis. Of even greater concern is the fact that sprawl represents a risk for business continuity, because ad hoc analytics clusters may not have sufficient monitoring of experienced staff to guarantee that data and services remain available.
2. **Investment protection.** New analytics deployments are incredibly difficult to scope. IT directors and CIOs frequently face an initial analytics project that will demand clusters with hundreds of terabytes, or even petabytes of initial capacity need. However, if the systems are not successful, or if early estimates far exceed actual uptake, an IT group may be left with a sunk investment in hardware that has no transferrable value to alternate IT applications.
3. **Lack of management and data services.** While analytics environments have evolved significantly in recent years, they still fall short on many capabilities and features that have become commonplace for existing infrastructure products in both virtualization and storage. For example, analytics purchasers express a desire to support remote replication, metering and chargeback, and the rapid provisioning of per-tenant analytics resources.

In this reference architecture, which was motivated by conversations with multiple large enterprise business who have articulated the above concerns, we demonstrate a deeply integrated combination of traditional NFS-based file storage, with hosted and in situ data analytics using Cloudera CHD5. The resulting system acts concurrently as a scale-out enterprise storage offering and a fully functional analytics cluster. Scale-out means the system has shared-nothing, per-server storage devices that are presented by software running on those servers and arrayed in a cluster configuration. The cluster is specifically designed to satisfy Enterprise storage requirements.

Introduction and Problem Statement: The Enterprise Big Data Dilemma

Virtually all large enterprises today have either deployed, or are in the process of deploying, managed infrastructure for business intelligence and big data workloads. These projects typically start as a collection of ad hoc analytics clusters spread throughout an organization driven by various business groups. Due to the ad hoc nature, insufficient attention and investment have been given to these environments. Once the BI solution is in place and has become an necessary part of the business, Enterprise IT is engaged and required to become responsible and accountable in supporting these tenuous environments

As a result of enterprise deployments of big data analytic solutions, a tension between big data and existing infrastructure architecture has come to light. Big data is a new, and often completely independent silo in terms of purchasing, planning, operations, and evolution. Siloed big data environment can present a serious challenge in terms of efficient resource usage, availability, business continuity, replication and backup. Moreover, as a new technology solving complex unstructured data problems, big data platforms are notoriously difficult to plan for from a use and capacity perspective. For all of these reasons, big data deployments present significant risk in terms of IT project success. According to Gartner , "Through 2018, 70% of Hadoop deployments will fail to meet cost savings and revenue generation objectives due to skills and integration challenges." The success of big data projects hinge critically both on integration with existing IT infrastructure, and the technical capabilities of both the administrators and users of the system to expose and harness new BI capabilities.

The diagram below illustrates several aspects of this big data dilemma. Big data environments are deployed as an independent physical environment necessitating separate hardware, and independent connectivity and planning. Additionally, a significant technical challenge exists in integrating the big data stack into the traditional enterprise IT infrastructure.

Two concrete issues introduce problems that result in big data implementations as configured below in Figure 1. These are:

1. The mismatch between Big Data workloads and Enterprise storage architectures
2. The tension between the desire of AppDev teams deploying Big Data solutions to have high performance and isolation and IT's mission to meet company expectations and standards for data availability, integrity, protection, security, etc.

Now let's dive deeper into each of the problems we outlined. ***First, there is a strong mismatch between big data workloads and traditional enterprise storage architectures.*** In the diagram below, it is important to notice that all of the servers in the big data silo contain independent local storage. The recommended configuration (from vendors such as Cloudera and HortonWorks*) for these environments is to co-locate storage with compute, in order to allow analytics jobs to be scheduled as close to data as possible reducing repetitive data movement and unnecessary processing time. The resulting I/O capabilities (and in turn, aggregate data processing capability) should then scale as new nodes are added to the environment. Traditional enterprise storage presents limited connectivity between the SAN or NAS and the rest of the network. As a result, big data vendors strongly discourage enterprise storage targets as a primary data source for analytics specifically because aggregate data bandwidth may not scale in proportion to the size of the connected analytics cluster.

Secondly, there is a **tension between AppDev teams deploying Big Data applications and IT who is responsible for insuring the data is highly available, secure, backed up on a regular basis, and in compliance with a diverse set of governance restrictions.** IT's response is to advocate using their managed/shared, hypervisor-based virtualization infrastructure. However, AppDev teams are skeptical that such a solution can satisfy their service level requirements.

This reference architecture sets out to describe a fully integrated enterprise big data environment. The system described here has been developed in a collaboration between Intel, Coho Data*, and the end customer. Coho Data's architecture is designed specifically to bridge the gap between traditional monolithic storage architectures, and emerging datacenter technologies such

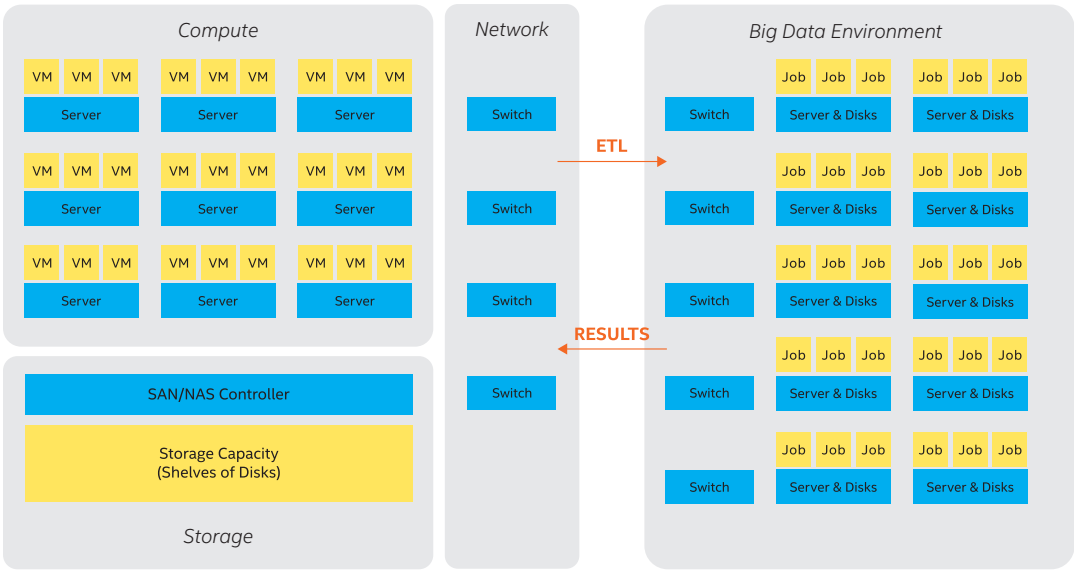


Figure 1 - The Enterprise Big Data Dilemma: Big data environments are frequently deployed as a completely separate silo.

1. http://www.gartner.com/document/2956017?ref=QuickSearch&sthkw=hadoop&refval=147069170&qid=63956a9e94577796094eca2345225e01

as big data frameworks and containerized applications. Coho Data provides a completely scale-out, network-integrated storage platform that allows unmodified clients to access traditional storage protocols such as NFS over a single IP address, and transparently scales connectivity over the full width of an enterprise Ethernet fabric. In combination with Intel CPUs and two tiers of high-performance enterprise flash memories, the resulting design can realize a scalable, high-performance, multi-tenant environment that converges enterprise storage and big data analytics for business intelligence.

Data-Centric Enterprise Infrastructure: A Reference Architecture

The remainder of this document describes the design and deployment of a reference architecture for a scalable IT infrastructure that hosts Cloudera Enterprise (CDH)* directly within a scalable enterprise storage system. The system has been deployed at the customer's site, with an initial scale of 20 nodes, interconnected across four racks using 40 Gbps networking. The reference architecture is a direct result of addressing the concerns outlined above, focusing on allowing the environment to effortlessly share resources between the traditional storage - enterprise data over NFS, and the scheduling of per-tenant analytics jobs on that data in situ using CDH5.

Coho Data Cluster Layout

The cluster configuration used in the deployment is depicted in figure 1. There are four data center racks, one for management services and the other three for Coho Data-based storage services. Each rack has a 1 Gbps Ethernet management switch and a 40 Gbps Top-of-Rack (ToR) switch. These ToR switches are interconnected via a set of 40 Gbps Spine switches.

There are a total of 23 servers in the cluster, three servers are deployed in the management rack and the remaining 20 servers are distributed across the other three racks as seen in Figure 1. Each server is attached to their rack's 1 Gbps management switch and to the rack's ToR switch via a 40 Gigabit Intel® XL710 converged network adapter. Each of the DataStream nodes are fitted with a 1.6TB Intel® P3700 PCIe* SSD using the NVMe product and ten 800GB Intel® S3500 SSDs. These latter devices are 6 Gbps SAS-attached in a JBOD configuration. Unlike traditional storage systems that perform host-side RAID within a server, Coho Data's architecture explicitly manages data locality and placement across the cluster and fabric, thus the devices are virtualized by Coho Data's thin data hypervisor and directly presented to the network as a low-level object store.

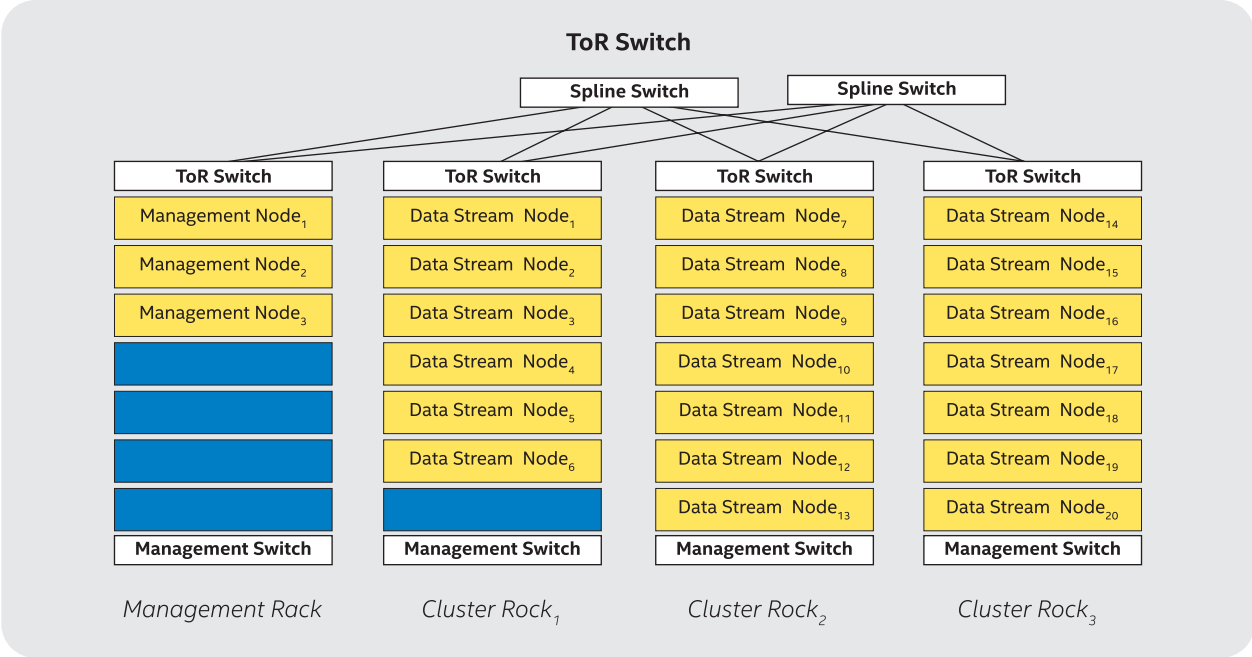


Figure 2 - Reference architecture cluster layout.

Coho's Virtual Management Network and the Storage Controller (SC)

The Coho Data cluster includes a "Virtual Network" for out-of-band control and management operations internal to the cluster. However, in the customer deployment the 1Gbs management network was used to carry this traffic.

The "Storage Controller" (SC) is a dedicated, cluster-wide service, attached to the Virtual Network that:

- Performs large-scale reconfiguration tasks to maintain the health of the system as a whole
- Triggers peer-wise resynchronization and reorganization of objects amongst the Network Addressable Devices (NADs)
- Handles events such as device failures and load imbalance
- Interacts with a Software Defined Network Controller to actively and dynamically manage external client connections into the storage system to maximize aggregate performance and efficiency

The Storage Controller draws inspiration from the similarly named role of a "network controller" or "OpenFlow controller" in the context of software defined networking (SDN). The Storage Controller is a logically centralized function that interacts with distributed storage endpoints for orchestration and management.

Isolating the traffic on a virtual network that is only visible to the storage system effectively creates a private storage fabric without necessitating the deployment of an entirely separate physical network. This technique forms the basis for network-level isolation for multiple containerized analytics tenants described later in this document.

The storage controller (management software) that runs in the management rack and provides configuration and coordination services to the rest of the storage servers in the cluster. When a server is power-cycled via IPMI, it is configured to boot using PXE/DHCP. The PXE packet is received by the SC service, which provides a point-to-point response back to the booting server.

The LLDP daemon is then initialized and initiates the discovery process, which is also handled via the Virtual Network, similar to the PXE process described above. Once the discovery process is complete, the system uses the 40 Gbps and issues a DHCP query that SC service responds to. These interactions continue until the booting server has initialized its network stack to the point it is ready to configure its root file system and initialize its operating system.

Cluster-Wide Configuration Database

Upon completion of being added to the cluster, the DataStream node is ready to complete its cluster membership process and register its NAD with the cluster-wide Configuration Database. This data store uses a Paxos-based distributed co-ordination service to persist information about the current cluster membership and related state.

A NAD is a per-device object store that presents a device as an address space of 2128 sparse objects, each of which may be up to 264 bytes (16 Exabytes) in size. Coho Data then layers on top a platform-wide address space that maps a virtual object onto a set of these physical objects. A virtual object can be made available for use by an end-user's workload, an NFS mount point, an HDFS mount point, etc. The metadata for managing these relationships is maintained in the Configuration Database.

The Configuration Database has a secondary function to implement a coherence protocol insuring the integrity of the cluster's metadata. This protocol is used to coordinate a cluster-wide resynchronization service and orchestrate crash recovery when a node fails or is taken out of service.

Switch Store Application

The cluster's ToR and Spine switches provide a single fabric substrate that delivers any-to-any connectivity between an arbitrarily selected pair of servers attached to the fabric. This connectivity is IP-based. Storage-related traffic is configured with an IP address as part of the Coho Data Storage Network. Two Coho Data Datastream servers communicate via this network with no dependencies on the physical switches. Instead, the switches act as fast forwarding devices.

In order to configure the DataStream nodes into a singular flat Storage Network, an IP substrate is dedicated to the Coho Data cluster. Each server's 40Gbs interface has an IP address on this network, which is used by cluster members to communicate with each other. Effectively, this is a virtualized cluster interconnect.

To bring the Storage Network online, a Switch Store Appliance (SSAPP) runs as a virtual machine on hosts in the management rack. The cluster's Configuration Database allows coordination to ensure that exactly one instance of the SSAPP is always running and operational. The role of the SSAPP is to provide imaging, installation, and cluster membership services to new nodes. This SSAPP instance handles the cluster's PXE, LLDP, and DHCP traffic.

Per-Tenant Application Networking

In addition to the Virtual Management and Storage Networks, the cluster provides an Application Network. Each DataStream server has a local Open vSwitch (OVS) based bridge. There are a set of VXLAN tunnels terminated on the OVS instance with the other end of each tunnel terminated on another DataStream server. The result is an

"any-to-any" mesh of VXLAN tunnels so that all DataStream servers in the cluster are connected to the Application Network. At configuration time, containers are configured with the local IP for NFS and HDFS services for a given tenant. OVS is configured to forward this traffic directly within the physical host, avoiding unnecessary "tromboning" of container traffic off-host. We are investigating implementing a single endpoint-addressable IP for each service by configuring each OVS instance with per-tenant and per-service IP addresses.

With this in place, an end-user provisions a logical application network. This is a VXLAN-based tenant network (aka "overlay"). All traffic on this overlay will be encapsulated inside the VXLAN tunnel. The result is that containers (and/or guest VMs) attached to this tenant network appear to be interconnected on a LAN, i.e. a flat, layer-2 network segment.

An overview of the network configuration on each DataStream server is depicted in Figure 2. The block arrow between the Storage Stack and the OVS Bridge represents the virtual L2 bridge for traffic originating from a container destined for per-tenant HDFS and NFS services. The Virtual Management Network handles low-rate, control-related traffic such as DHCP/PXE and cluster membership. The Application Network handles all the per-tenant traffic that will be carried via VXLAN tunnels.

With the clustering and network aspects of the system explained, we now switch gears to discuss how data is stored and accessed in the system.

The Data Hypervisor: Coho Data's Log-Structured Object Store (CLOS)

Coho Data's storage architecture is motivated by the emergence of very high performance storage-class memories (SCMs), such as NVMe™ flash NVDIMMs. These memories challenge traditional storage system architectures because they effectively invert the design assumptions with which enterprise storage has historically been built. In the past, spinning disks were slow and cheap, and so it was sensible to place large numbers of them behind a single controller and not worry about the software overheads associated with RAID-style redundancy, volume management or file system implementation. Unlike spinning disks, emerging storage hardware is very fast, and often significantly more expensive than other components of the system. A single high-end NVMe product can saturate a 40 Gbps network port, and costs more than four times the price of the CPU that is required to drive that traffic. These memories are so performance dense that they are hard to keep busy. Most application workloads, even a busy, multi-VM server, fail to offer the I/O load to even approach the capabilities of these devices. As a result, this hardware presents a similar utilization

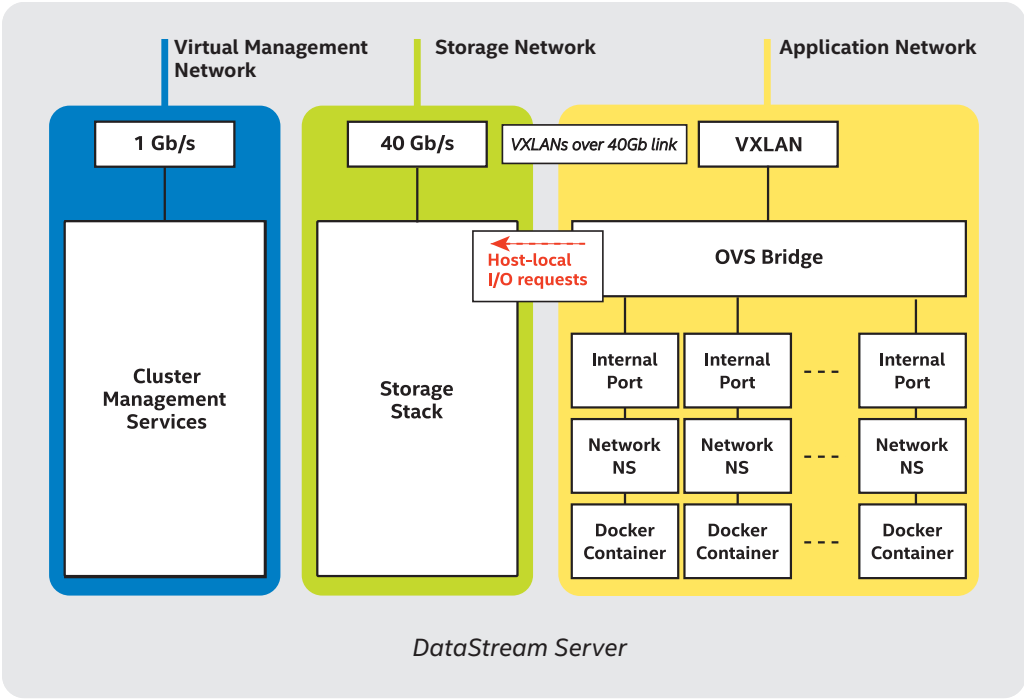


Figure 3 - Network configuration

challenge to one that was presented by idle CPUs (and solved with CPU virtualization) a decade ago.

The bottom layer of the Coho Data stack is referred to as a data hypervisor. It is a thin layer of code that is responsible for making devices directly available over the network (and so providing the NAD interface), and for allowing multiple tenants to share those devices in an isolated manner. The data hypervisor manages on-device data layout using Coho Data's log-structured object store (CLOS) and allows a collection of one or more devices to be presented as an arbitrary number of sparse address spaces. The system has two main design goals:

1. Impose as little overhead as possible. Like a CPU hypervisor, the data hypervisor attempts to expose the underlying hardware as directly as possible, allowing higher layers to implement the specific functionality they need, rather than being burdened with overhead-imposing features that may not be appropriate for all use cases.
2. Allow multiple tenants with different protocols. Also similar to a CPU hypervisor, the data hypervisor is intended to allow new tenants to be deployed alongside existing ones, and to allow the fast development of new storage- and data-access protocols on the system. The integration of HDFS alongside the existing NFS implementation is a good example of this aspect of the system.

CLOS can be configured with a single device, virtualized and exposed directly to clients. Alternatively, it may be configured to aggregate data on multiple devices. This latter configuration is typically used to allow CLOS to move cold (unused) data to lower-cost storage over time. CLOS's implementation does not rely on the existence of battery-backed or otherwise nonvolatile memory, instead electing to use NVMe-based storage as the direct path for persisted writes. Similarly, all metadata is stored in this fastest layer of flash, allowing for fast rebuilds, format updates, and crash recovery.

Figure 4 illustrates CLOS's layout. NVMe flash is dynamically divided into three regions: First, an operation log records changes to stored data, such as new writes. Second, a set of range-maps are indexing data structures that provide fast lookups of data by address. These are generated from the operation logs and so do not need to be kept consistent on disk, allowing for faster performance in the system, they are also aggressively cached in memory on the microarray. Finally, remaining storage is divided into a set of data heaps – areas of device capacity that can be used to store the actual data in the system. CLOS organizes these from fastest to slowest, and tracks the age of data in the system over time, moving the coldest data out to slower and cheaper devices as it ages.

Data hypervisor instances are shared-nothing stores. Metadata on them is entirely limited to local data, avoiding the need for low-level transactions or consistency across NADs.

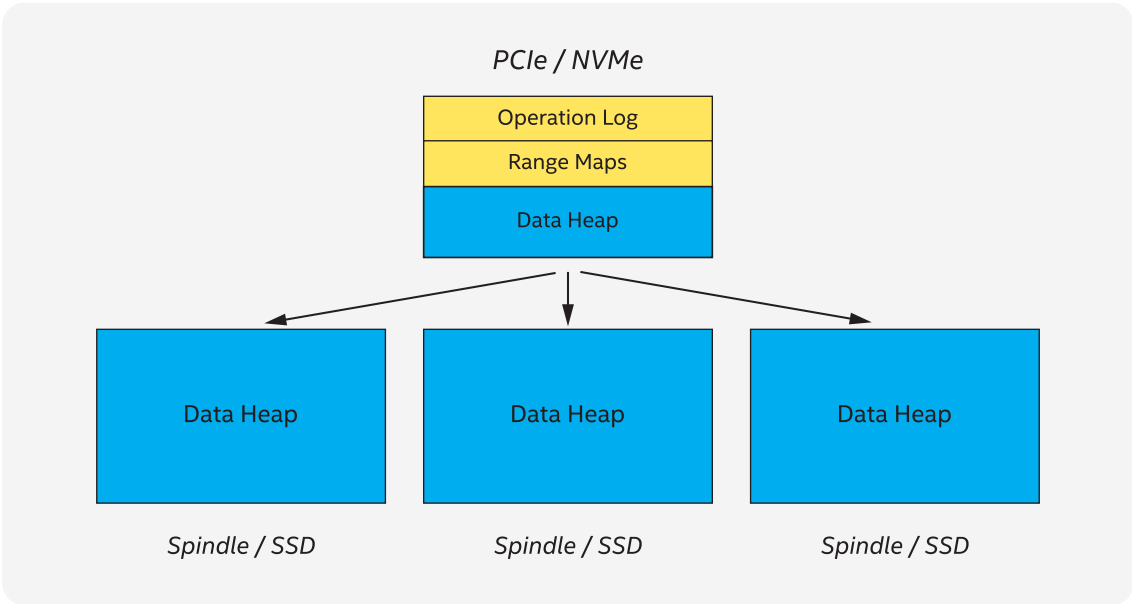


Figure 4 - CLOS virtualizes flash hardware and dynamically de-stages cold data.

Dispatch-Oriented Data Path

With the data hypervisor in place, the second challenge faced by the system is to provide rich enterprise data services, including facilities such as replication, failure recovery, performance monitoring, chargeback, and dynamic scale-out. These features are provided in a flexible dispatch layer that is implemented above the data hypervisor, and that can be used directly by any tenant of the system. The fundamental idea in the dispatch-oriented architecture is that these data services should not be implemented by a single central controller (as has been the case in traditional storage systems) but instead be a set of reusable functionality that can be incorporated into tenants of the data hypervisor. Coho Data's NFS and HDFS implementations, described next, are a scale-out service that are co-hosted on the storage nodes and use the dispatch functionality to allow transparent scale-out of the storage controller. An alternate model for interacting with the dispatch library is to link it directly to applications or OS storage facilities. In this regard, the dispatch-orientation is a strong fit for scalable storage services in container environments using technologies such as Docker.

Using Coho Data's dispatch library, hosted storage services are able to take advantage of component functionality for storage, replication for instance, to build rich objects for use in the storage system. This is characterized in the diagram below: a "regular" object in the deployment is composed using a dispatch policy that specifies 3-way replication. With this policy, the dispatch library presents a single object address space, but concurrently forwards all writes to three independent CLOS object instances, waiting for all three writes to be acknowledged before the request is acknowledged as complete. Other dispatch configurations allow for chunking or striping of objects, allowing the ability to scale to very large object sizes while still carefully controlling locality and placement. This regular object is a first-class container for data within the storage system, and is in turn mapped to a virtual object within, for instance, an NFS or HDFS namespace. Storage protocol implementations (e.g. NFS/HDFS) are hosted directly on the microarrays and interact directly with the associated regular objects in the system. This approach allows for strong partitioning between objects and allows the system to maintain a very clean design that provides low-overhead and high performance.

Dynamism and the Storage Controller (SC)

Both CLOS and the dispatch library instances provide asynchronous interfaces to the SC. When new nodes are added to the system, when nodes fail, or when performance or capacity characteristics of the environment change, the SC adaptively calculates a plan for restoring the durability, performance, and related placement constraints that have been provided for regular objects. It then initiates peer-wise data migration or copy at the data hypervisor layer, and notifies dispatch instances as the locations of CLOS objects are reconfigured.

The SC achieves this goal by solving a constraint-based problem that is expressed as a set of declarative properties representing the policy associated with each object and hardware component in the system. Declarations include properties such as the fact that "all data hypervisor instances should have balanced capacity stored on them," or "replicas of an object must be placed in different fault domains". The SC is activated both periodically and in response to environmental changes such as either component failure or the addition of new hardware.

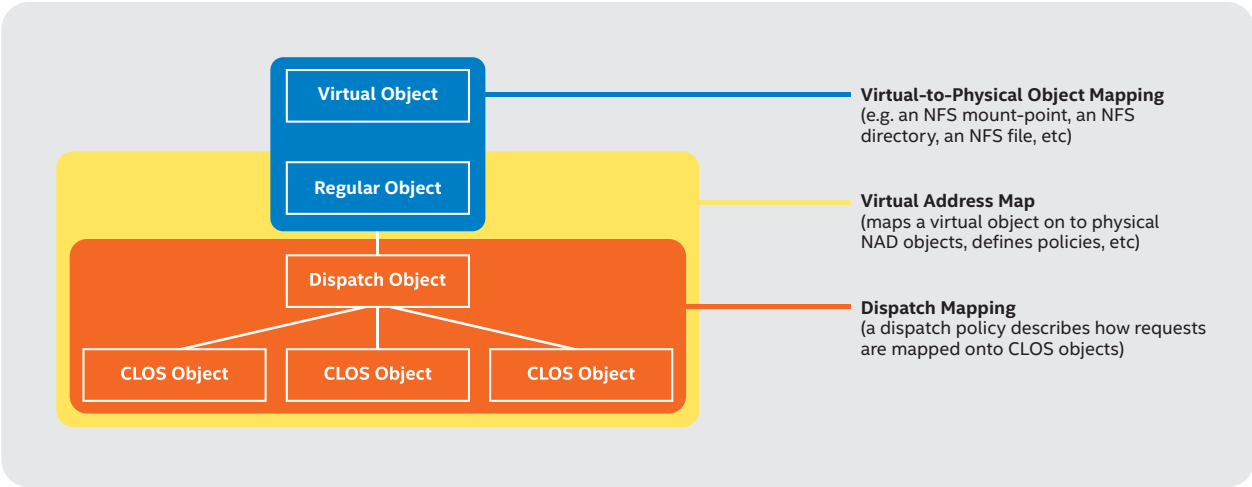


Figure 5 - Object representations across the dispatch-oriented data path.

Scalable Multi-tenant NFS

Coho Data's enterprise NFS implementation is a thin NFSv3 protocol presentation layer that runs on the microarrays, adjacent to the data hypervisor layer. It uses the data dispatch library to forward requests appropriately across data hypervisors on both the local and remote microarrays. SDN integration for protocol scaling allows the single NFS server IP address to transparently scale connection traffic across all nodes in the system, and the SC provides dynamic reconfiguration to scale both connection bandwidth and storage performance as new nodes are added over time.

Common deployments for Coho Data's NFS are virtualized environments using VMware ESX (as an NFS datastore) or OpenStack (as a Cinder storage provider).

Scalable Multi-tenant HDFS

HDFS support has been implemented similarly to NFS. Coho Data implements the HDFS wire protocol, including both metadata management (name node) and storage interactions (data node). The regular objects used by HDFS are configured for 3-way replication (to allow flexibility in the scheduling of compute on data), and objects are configured to a 64 MB chunk size. Both of these parameters are fully configurable on a per-object basis. The on-disk storage of data is presented by the dispatch library and data hypervisor, and not the legacy HDFS file system. As a result, despite allowing HDFS client applications to seamlessly interact with their data, the stored files benefit from rich enterprise storage capabilities such as remote replication, and transparent scalability that is provided by the platform.

HDFS and NFS are typically configured to interact with a single, shared file system namespace. This allows enterprise data to be used directly over NFS, and then to schedule processing jobs to run on that data in situ. Jobs may run using any of the wide variety of existing tools that have been built over the HDFS APIs, including frameworks such as Apache Hadoop and Spark. This approach avoids unnecessary data copies out to a separate silo and brings the data transformation and analytics capabilities from data parallel tool chains such as Cloudera Enterprise to be first-class tools within the enterprise storage environment.

Hosting Cloudera CDH5 using Kubernetes APIs and Docker Containers

The advent of container technologies such as Docker has been a huge asset in managing the deployment, scale, and maintenance of complex software. Cloudera recently began packaging the CDH5 Cloudera Enterprise suite as a set of Docker containers, and this approach has become a common technique for installing CDH5 onto production systems.

A well-known challenge with containerized software deployment is that containers themselves do not solve the entire problem. They are simply a way of managing a single piece of software and its installation and configuration on a single system. For large-scale distributed environments, an additional layer of software, often referred to as an orchestration layer is required to coordinate the deployment of applications that are composed of multiple container instances, that span multiple hosts, and that have complex cluster configurations, including virtual network topologies, load balancing, and shared storage requirements.

For this reference architecture, the CDH5 containers were repackaged with modifications to integrate with the storage and networking aspects of the system as described throughout this document. They were then wrapped in a service description using the Google Kubernetes* interfaces. Kubernetes is one of several emerging orchestration platforms for containerized applications, and Coho Data's implementation was extended to support the deployment of applications against that API.

In hosting CDH5 instances, the incorporation of the Kubernetes APIs allows users to quickly and programmatically instantiate CDH5 clusters specifically for their needs. For example, an 8-node cluster may be provisioned and deployed in a matter of seconds, and used in isolation on private storage and virtual network resources for as long as it is required. Resource use may be audited and charged appropriately, and then resources freed when the cluster is eventually decommissioned. A common use case that we have seen in the environment is that clusters may be provisioned for very short periods of time, often just to run a single job against a data set, and then decommissioned allowing resources to be appropriately applied to production storage or other workloads. This stands in stark contrast to traditional CDH5 deployments, which may sit idle for large periods of time.

The incorporation of orchestration APIs into the Coho Data platform has far broader implications than just data analytics. Users of the system have begun exploring the deployment of additional containerized services, including cluster applications such as Redis* and Elastic Search*, as scalable service-based augmentations of the existing storage and analytics platform.

Evaluation and Experiences in a Data-Centric Enterprise IT Environment

The reference architecture described above has been implemented and deployed in the customer's IT lab environment, and has begun to host initial big data application workloads. In this section we discuss the performance

evaluation that was performed on the platform prior to making it available to application teams, and our experiences in working with those teams to implement applications in the environment.

Does solid state memory improve the performance of big data workloads?

As mentioned above, the reference architecture is composed of 23 DataStream nodes, each with a 1.6TB Intel® P3700 NVMe SSD, and ten 800GB Intel® S3500 SSDs. The resulting cluster has a combined raw capacity of over 200TB, and is likely representative of the composition of big data environments over the coming years as solid state memory prices continue to fall towards the cost of disk.

However, given that this all-flash design represents a potentially higher cost solution today than an equivalent capacity cluster based on spinning disk, we begin by attempting to understand whether the solid state devices add value in this environment.

We began by comparing the native, "bare metal" performance of the CDH5 packages and HDFS between spinning disks and flash. To get a sense of potential gains, we compared cluster nodes that were configured with 12 4TB spinning disks versus identical workloads running entirely on Intel® P3700 NVMe SSDs. These measurements are intended to provide a hardware baseline using the existing CDH5 implementation, and so do not use Coho Data's storage layer.

The two configurations were evaluated using both I/O intensive micro-benchmarks and established big data workload benchmarks: our findings are that the improvement of application performance between these two storage configurations is entirely application dependent. As an example, on the terasort benchmark with a single active tenant in the cluster, the solid-state implementation was roughly 25% faster than disk. However, on workloads that involved significant levels of computation as well as I/O, the ability of solid state devices to contribute to whole-system performance diminished. These findings are consistent with those of other studies[1,2]

Do solid state memories help performance in the face of multi-tenancy?

We continued to evaluate the performance differences between solid state and spinning disk using the bare metal (non-Coho Data) configuration. Given the expectation that application owners desired to configure their own analytics clusters with potentially disparate software, we evaluated the performance of the two configurations in the face of competing workloads.

In this dimension, the flash-based system represents a clear and unequivocal performance win over spinning disks. As shown in Figure 6, the contention for disk spindles introduces a significant burden under even light levels of multi-tenancy: with four competing tenants, each individual job took almost two and a half times as long to compete over disks than they did under the same degree of multi-tenancy on the flash-based system.

The importance of this use case has been reinforced by our experiences in moving applications into the environment. These efforts have involved frequent and ongoing customization of per-application environments, and the ability to use Docker to manage and deploy individual application environments as containers has been highly valuable. The resulting multi-tenant system benefits enormously from the high levels of random access performance offered by the Intel® SSDs.

Is Coho Data's storage system effective for big data workloads?

With hardware baselines established, we move to consider the performance of Coho Data's HDFS protocol implementation onto the DataStream storage system as compared to native HDFS. Remember that in this configuration, Coho Data's architecture offers both

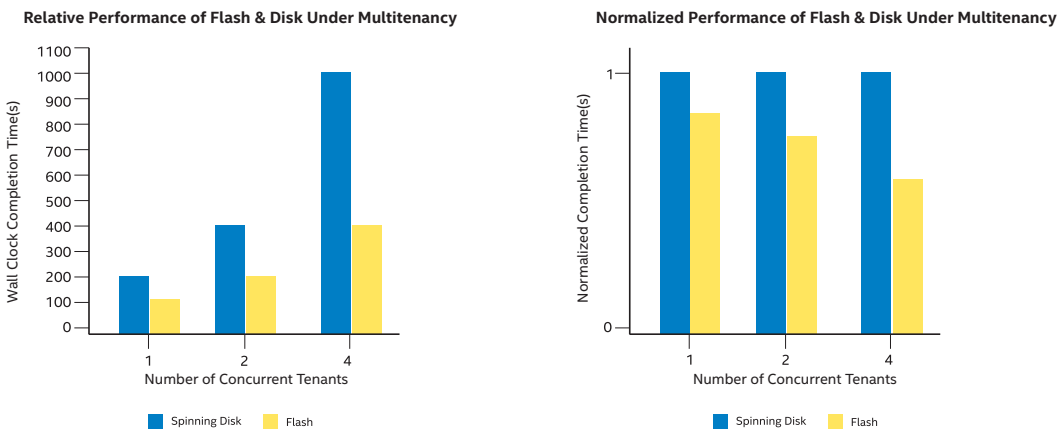


Figure 6 - Relative performance of flash and disk under light levels of multitenancy.

NFS and HDFS protocol implementations as direct access interfaces to its underlying high-performance object store. As a result, the system offers enterprise-class storage capabilities such as consistent and durable writes, snapshots, automatic tiering, and asynchronous remote replication.

As a full-featured enterprise storage implementation, we were initially concerned that Coho Data's implementation contained overheads that would put it at a performance disadvantage to HDFS: For example, HDFS is an eventually consistent system; it acknowledges writes on the first successful write request and then forwards requests through additional replicas afterwards. Coho Data's implementation guarantees durable and committed writes on all replicas before acknowledgement to clients, which is an expected behavior for enterprise-class storage.

Despite the additional features that are present in Coho Data's implementation, the system performs as well as or better than HDFS from a performance perspective in all tests that we ran. Application benchmarks showed performance that was in line with the bare metal flash performance discussed above. Moreover, through the development of the reference architecture, Coho Data's engineering team identified a number of opportunities for performance improvements that lead us to believe that the system will achieve even better performance relative to HDFS in the future.

Is the convergence of big data and enterprise storage a valuable solution?

As a final note in our evaluation of the reference architecture, it is worth summarizing our experiences working with users on the resulting converged system. The most important point to emphasize in this regard is the shift in perspective of the value of the system that occurred as we designed, implemented, deployed, and eventually worked with application teams. Our initial expectations largely surrounded the potential performance advantage of combining Intel® SSDs and Coho Data's software-defined storage implementation for big data. While these performance implications are a clear reality, they have not proven to be the highest impact aspect of the system.

Instead, the most significant win realized in this design has been the ease and efficiency in which the architecture allows users to bring big data analysis tools to their data. By extending a scalable and high-performance NFS implementation, the architecture allows big data tasks to be deployed on primary data where it lives. This can dramatically ease both interaction with existing application ecosystems and the barrier to entry in building out an initial big data cluster. The system has allowed us to engage not only application teams who have existing clusters, but also ones that have been looking to experiment with tools like Spark on their own data.

The reference architecture has received compliments from customer's application owners as making it easier to take action on their data, both because of NFS integration and the ability to quickly provision and also to customize container-based analytics environment templates. It has been reviewed favorably by IT architects as providing a both richer combined feature set and a lower total cost of ownership than independently managed storage and big data silos.

Conclusion

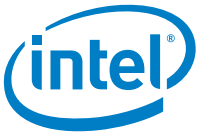
Big data has the potential to dramatically improve the ability of businesses to understand their organizational data, to make effective decisions and to remain competitive in the face of dynamic environments. This reference architecture has presented a fully integrated big data implementation within a scale-out enterprise storage system. The resulting architecture, results in dramatically improved manageability, scalability, and performance in a multi-tenant environment that provides users with the flexibility to deploy their own big data tool chains and to apply those tools directly to stored enterprise data.

Tight integration with software-defined networking provides the control and isolation to present clear and predictable service-level objectives for big data tenants, and to host tenants in secure virtual environments across an enterprise network.

Finally, with the incorporation of emerging interfaces for containerized applications and their orchestration (specifically Docker and Google's Kubernetes APIs), this RA represents a broader realization of a richly converged but carefully designed hosting environment for the emerging set of scalable and distributed enterprise applications.

References

1. K. Kambatla, Y. Chen, "The Truth About MapReduce Performance on SSDs." 28th Large Installation System Administration Conference (LISA14). https://www.usenix.org/conference/lisa14/conference-program/presentation/kambatla
2. "Making Sense of Performance in Data Analytics Frameworks," K. Ousterhout, R. Rasti, S. Ratnasamy, S. Shenker, Byung-Gon Chun. 12th USENIX Symposium on Networked Systems Design and Implementation (NSDI'15). https://www.usenix.org/conference/nsdi15/technical-sessions/presentation/ousterhout



Notices and Disclaimers