A Best-In-Class Data Analytics Platform
Built on a VMware Private Cloud
### Executive summary

A Best-In-Class Data Analytics Platform, Built on a VMware Private Cloud

---

### Introduction

---

### What is Super Collider?

---

### VMware’s Need for Data Analytics

---

### Super Collider’s True Value

---

### Optimizing Data flow

---

### Super Collider’s Design

---

### Architecture

---

### Super Collider Data Lake and Analytics Stack

---

### Super Collider’s Infrastructure

---

### Operations

---

### Lessons Learned: Past and future considerations

---

### Past Decisions

---

### Future Considerations

---

### In Conclusion

---

### Bibliography
Executive summary

Like most large enterprises, VMware needed to get more value out of its data by removing data silos and combining disparate data sources into a centralized and governed Big Data System. As part of this journey, we started with a hyper-scaler service, evaluated a third-party SaaS solution, and ended up with a VMware + open-source software technology stack that is deployed in a private cloud built and operated with VMware software. After having invested over 100 person-years in this endeavor and ending up with a successful outcome, we want to share our learning, our perspective on Data Platforms, what we built, and why we built it this way. The data platform that was deployed, known as Super Collider, achieved an industry benchmark - Net Promoter Score (NPS) of 81.8, which places it well into a “best in class” offering. Hopefully the learnings from this read will help you build a data system in much less than 100 person-years.

Introduction

In today’s competitive environment, business leaders from the C-suite down know that the fastest way to a big idea is to cultivate a data-driven, test and learn culture [1]. The use of data and the analysis of this data are changing the way business is performed in today’s marketplace. According to a Gartner study [2], digital innovation goes hand in hand with data analysis. Every software application that is built and run, be it a web site, or a mobile app, or a back-office information technology (IT) system, produces data that can offer key insights and reveal unseen business value. This data can be things like telemetry, logs, click streams, and metrics related to Service Level Agreements. Our demand and appetite for data has been, and will be, ever increasing, as shown in following chart that highlights the ubiquitous data growth over the last decade, as well as the forecast for growth over the next few years.

![FIGURE 1: Annual Size of the Global Datasphere, IDC.](image)

Just think about this – 90% of the data that we process today has been created in the last few 2-3 years [3]

As VMware journeyed down the data intelligence path, we observed and worked through the data growth identified in the previous chart. Super Collider, our internal analytics service, has had a steady 10% month-over-month growth in data volume for the last 7-8 years. This translates into a 3x annual growth, confirming that 90% of the data in the system was generated in the past two years. The demands on a data analytics platform to be able to store this much data and allow for efficient and effective analysis is not a small task. With so much data in hand, organizations find themselves challenged in processing the quantity of data collected, often escalating the need to build automated analysis tooling, to get to the key insights in a timely manner. More data to store and process typically means more cost. To achieve a positive return on investment (ROI) for data analytics, an enterprise needs a data system whose business value outpaces the costs of the system itself.

The ability to recognize business value from data analytics in not a new revelation in industry. As an example, more a decade ago, NetApp’s automated support infrastructure made it possible for a NetApp customer to arrive at work one morning and find a package awaiting at their desk, containing a hard drive along with replacement instructions, for a failing NetApp storage array drive that the customer had no clue was in trouble. The system had reported metrics back to NetApp and it was determined that the drive needed to be
replaced, prior to a hard failure. The value to the customer was very evident and the value to NetApp was heightened customer satisfaction, the root of many good things.

The possibilities of what you can learn and understand from data are endless. There are only two limiting factors – your imagination and the available toolset.

With Super Collider we worked to eliminate conventional toolset limitations from the equation and to unlock the endless possibilities for our customers – the 30,000+ employees of VMware.

What is Super Collider?

VMware is well known for providing virtualized platforms that are easily configurable, highly performant, elastically scalable and secure; well suited for the needs of a petabyte-scale data analytics platform. The data analytics platform to be developed was to operate as a SaaS application, accessible to all VMware employees. The vision was to provide endless data source integration producing a structured data lake. To satisfy the need for a curated domain-specific data Super Collider offers data warehousing capabilities. Finally, with the addition of a full featured graphical user interface (GUI) and governance capabilities, Super Collider delivered on the promise of being a complete, self-service big data analytics platform

To appreciate this vision, let’s review the initial set of use cases Super Collider set out to address. As we walk through each, we’ll touch on what challenges were faced and addressed and which challenges remain. This understanding will aide in rationalizing the specifics of Super Collider’s design and implementation, presented later in this read.

VMware’s Need for Data Analytics

Several years ago, VMware introduced VMware Cloud (VMC), a viable platform for competing in the SaaS market. It was clear from the beginning that SaaS needs would be very different from that of on-premises offerings. requiring frequent adjustments to scale and resources, based on real-time feedback and dynamic conditions. All sorts of data-driven decisions were a necessity. At that time, there were few data systems in use at VMware mature enough to handle the task of supporting VMC. There was one system available, an in-house data warehouse, combining various back-office data, along with sales, customer, and support information. There was another system in use, functioning as a data lake, storing product telemetry data from all VMware products. For the VMC to leverage this data effectively, the VMC team would have to integrate the multiple data flows, processes, and toolsets of both systems to enable data-driven decisions for the customer’s SaaS applications. Instead of integrating with those workflows, a low-cost solution was quickly deployed, based on Amazon Redshift technology, to address the need to perform data analytics. Initially, the solution worked well, but, as the data analysis quickly showed its value, the hunger for more data grew. Quickly the team found themselves in a position where instead of two systems, they needed to use three. The expansion of the Redshift solution presented challenges, delaying the time-to-market of mission critical analytics. The addition of new systems also complicated the goal of unified, federated data. During this short timeframe it became very clear and obvious that VMC team’s real quest was to have a holistic, 360° view of its operations, unifying datapoints related to customer, product, support, and SaaS utilization, to name a few.

This need drove the creation of a new platform that combines the benefits of the already existing platforms (hence the name Super Collider). Super Collider inherited the Amazon-based system with SaaS data and the in-house system with product data. The positive results were immediate – analysis over Super Collider data identified wins for both customers and VMware.

This increased the adoption of Super Collider across all the departments in the company (full-time data engineers, data scientists, software developers, managers, sales, executives, etc.). The data continued to grow in volume and variety.

As the number of active users moved into the thousands, the needs also grew for enhanced collaboration capabilities, data management, governance, and self-service facilities. These capabilities had to deliver on-demand, real-time analytics, coupled with Artificial Intelligence (AI) processing and discovery.

True to our vision we’ve built Super Collider to serve as a platform that is an enabler for everyone at VMware to innovate and process data, without having to learn a complex toolset, bound by technology limitations. Here are a few examples of Super Collider enabling follow-on innovation:

1. The VMC team’s utilization of machine-learning (ML) algorithms, based on data within Super Collider, to predict the hardware capacity needed by VMware Cloud on Amazon. These capabilities enabled “just in time” hardware purchasing from Amazon, ensuring that the upcoming demands would be ready, without incurring a high cost of “in-reserve” assets.

2. The development of vSphere Health, a powerful feature of vSphere 6.7, which works to identify and resolve potential customer issues before they have an impact on the customer’s environment. Resolutions and recommendations are provided to guide the customer through remediation, avoiding customer platform down time and dissatisfaction.
Super Collider’s True Value

In Buddhism, nirvana is the ultimate goal and marks the end of the vicious cycle of doing things ineffectively, over and over. Super Collider’s nirvana is defined as achieving data democratization, that is, data is available for anyone to use, at any time, without difficulties accessing or understanding it. In a system free of gatekeepers and bottlenecks, collected data can be used to expedite critical decision-making and uncover value and opportunities for an organization to execute on. Data democratization is an enabler for x-analytics and decision intelligence [2].

Data democratization is a complex concept and if we are to oversimplify it, it would boil down to self-service and data quality. This is why Super Collider is a multi-tenant self-service data platform that promotes data mesh principles [4]. Multi-tenancy encourages decentralized data ownership, with each tenant taking responsibility for a data domain. Data domains promote data quality and treating data as a product. When data is a product, owners do their best to make their customers (data users) happy. Being self-service allows for unobstructed interactions between users and owners. Last but not least – the platform facilitates (makes it easy for users but does not enforce) processes and tools aligned with our opinion on how to manage data efficiently, including all aspects of data management and policy compliance.

Super Collider also addresses the limitations of the traditional model where data warehouses (DWH) are built by the IT organization and any change to the DWH (bringing new data, modifying existing data) is implemented by a central team. This model of operation poses challenges with the prioritization of requests by multiple teams, which affects agility and prevents the business from moving with the speed that digital transformation requires. In this traditional workflow, multiple business teams would submit work requests, which would first be prioritized by IT for implementation. Longer time frames would typically be observed, as time is needed to communicate objectives, requirements, validation needs, and sign off for the deployed changes.

The Super Collider team decided to offer a platform with a self-serve model, thus leaving it up to the business to decide how much resources will be invested in implementing a given data feature. The team that needs the results, can insert the data, apply the analysis, dedicating the resources needed to achieve the timelines required. It is up to them to decide the balance between need, cost, and value. This self-service platform also works in an environment with centralized IT, data science, and data engineering teams, so that teams with limited budget can benefit from data too.

With support for multi-tenancy, data democratization, and the enablement of data mesh, all in a self-serving manner, Super Collider has truly differentiated itself from the other available solutions in this space and has cemented its value in VMware’s core business operations.

Optimizing Data flow

The evolution of data platforms has introduced us to multiple data housing terms, sometimes clearly distinguishable, and sometimes not. One of first recognized structure was that of a data warehouse – a centrally managed monolithic store for structured data, where data is first cleaned and then organized into marts. Then, Massively Parallel Processing (MPP) architectures allowed for cost-efficient systems that were able to handle larger data sizes and organizations started building data lakes containing all the raw data. And in recent years, the data mesh shifts the focus towards unlocking value out of data by organizing data and processes into bounded data domains.

Super Collider combines all three options to unleash the ultimate data analytics platform. From architectural perspective, Super Collider is rather a structured data lake, but it also offers warehousing capabilities, and the opinionated tooling and recommendations we provide facilitate the adoption of data mesh-like practices – for example, domain-driven pipelines, and cross-team collaboration.

Super Collider’s architecture presents a common ingestion application program interface (API) for all data sources to utilize in sending data to Super Collider. Using a common API to source all the data is the first part of transforming the data into a consistent structure for data ingestion. Sourced data is inserted into Kafka for downstream processing. Super Collider’s data mesh realization starts with the utilization of Kafka consumers, typically Spark based processes or jobs, to apply schemas to transform the raw data into structured tables which are then housed in the data lake, representative of the sourced raw data. Data domain owners or teams then have access to the system to generate customized data processing jobs to transform the raw data into data domain products persisted in data marts. From these marts, business intelligence (BI) tools can query and analyze housed data to deliver key business insights. Having the raw (lake) and processed (mart) data eases data science activities too.

The following is a representation of a typical data flow within Super Collider:
A Best-In-Class Data Analytics Platform, Built on a VMware Private Cloud

Super Collider users are self-organized into teams (so far we have 100+ teams working with the Super Collider platform). Those teams ingest and process data with the end goal of providing usable data and accurate business insights. For regular ingestion and data processing, the teams use an in-house built feature called Data Pipelines. Data Pipelines consists of two things – a workflow engine (Apache Airflow) and an abstraction layer over Super Collider APIs, a layer whose purposes is to make data engineers more efficient. That is, instead of opening an ODBC connection to Super Collider, dealing with credentials, retries, reconnects, etc., Data Pipelines allows you to just invoke a method called “execute_query”. Everything exposed by Data Pipelines is bundled with built-in monitoring, troubleshooting and smart notifications capabilities. Using Data Pipelines also allows data engineers to experience higher SLAs of Super Collider compared to the SLAs they’d get if they were using Super Collider APIs directly. Often the gain is ½ to 1 additional “9” of availability, e.g. from 99.9% original API availability, Data Pipeline users get 99.97% actual availability, because Data Pipelines have a sophisticated error detection mechanism and can retry on system-related failures such as network errors, for example.

Teams also manage data visibility; by default, all data – both raw and processed – is public to everyone. Data is only restricted when policy compliance requires so. Every piece of data and every process is owned by exactly one team, and members of teams, including their roles is available to all. This means, for example, that if you need access to certain data – it is clear who owns it and you can contact this person and ask them for access.

For higher developer productivity and to enable iterative development, Super Collider offers staging/production support where staging and production data are logically separated. All data – staging, production, lake, and marts - can be accessed by a single SQL query, which allows for rapid ad-hoc analysis and hypothesis validation. Although this approach is more expensive to implement (e.g. traditionally, staging and prod are different deployments, accessible via different URLs), it enables extremely fast prototyping and speeds up the overall data management process.

Super Collider’s Design
Building a Data Analytics platform is far from being a trivial task. There are hundreds of solutions to choose from and then integrate before any true data analytics can be achieved.
A Best-In-Class Data Analytics Platform, Built on a VMware Private Cloud

Few organizations have the time or skills, or resources needed to build such a platform. A recent paper [6] by Google highlighted the huge engineering debt involved in building and maintaining the platform behind their machine learning (ML) services. They claimed that much of their debt had nothing to do with the ML function itself, but more on the choice and implementation of the technologies used.

From the very beginning a clear goal for Super Collider was to provide a system whose value outperforms its cost. This translates not only to low-cost technical implementations but also to implementations with open-ended flexibility that allow making decisions with minimal impact in the future.

We continuously challenge our architecture and make changes to it. For example, we moved off of Hadoop, moved off of Amazon, considered shifting to a commercial solution like Snowflake®, etc. After a few iterations in this manner, Super Collider is built entirely on open-source and VMware technologies and is deployed in OneCloud. OneCloud is VMware’s internal cloud platform, an IaaS cloud built using VMware Cloud Director and operated by the IT Organization.

Using proven and readily accessible open-source software (OSS) technology stack gives us flexibility and ensures the best speed of innovation; this, combined with our expertise in operating OSS keeps cost down. Another significant factor for cost reduction is going to the cloud [7] and using cloud-native technologies; we’ve tried public, private, and hybrid, and finally settled with private cloud (more details in this section).

Architecture
A key requirement for the Super Collider architecture was to be able to handle large amounts of data at a relatively low cost – doubling the size of the data should not mean doubling the cost of the entire system. This translated into the requirement for storage and compute to scale independently. Well known relational database management systems (RDBMS), however, are expensive at scale, mainly because their compute and storage need to scale together. This meant that we could not just use a RDBMS, because their storage and compute are coupled, and as a result RDBMS at scale is expensive. To provide a better value, the decision was taken to utilize a scalable storage solution coupled with a query engine that could interrogate the storage. Separation of these services was vital for Super Collider to succeed.

To accomplish the above, Super Collider was built into three distinct service layers of implementation:

- **Infrastructure as a Service (IaaS):** Provides core compute, storage, and network infrastructure
- **Platform as a Service (PaaS):** Collection of reusable Open-Source Software (OSS) services and additional software components.
- **Software as a Service (SaaS):** Application layer integrating all services into a solution that exposes the data ingestion and analytics capabilities to end users.

The SaaS layer reduces complexity by hiding implementation details from the customer while exposing capabilities like a SQL interface to all data, data pipelines, data lake and data warehousing capabilities, multitenancy, data discovery and governance, etc.

The advantage of realizing and abstracting the architecture into three distinct layers was that it promoted separation of function, with clear and defined interfaces, enabling flexibility and future proofing of the design. As previously highlighted, technologies in the data domain are numerous and change very rapidly. Switching to a new technology is costly, especially if it requires a migration of petabytes of data. Layering the architecture was imperative to establishing contracts and functionality – on one hand among the engineering teams, and on another – for our users, abstracting away details for how data would be stored and migrated if ever needed.

The layers also aided in simplifying extensibility, which was common when data or even whole systems were consumed into Super Collider, such as the initial Amazon Redshift platform absorption, or when VMware acquires another company with its own data solution. Having separate layers helped us with project management and operations too – because the problem domain is huge it is impossible for one persona to be an expert in every area. Having distinct teams with targeted skillsets proved to be a much more efficient way of running Super Collider.

As we double click into the architecture, we’ll show how every facet of Super Collider is done with an eye towards utilization of pluggable services, layer over layer.

**Super Collider Data Lake and Analytics Stack**

Across the different layers of implementation (SaaS, PaaS, and IaaS), Super Collider was factored into two distinct entities defining a Control plane and a Data Plane. The Control Plane provided user controls and services such as management of teams, data sources, data governance controls. The Data Plane supported data ingestion and processing. A high-level overview of the architecture is defined in the following figure:
The access to the data lake is through Apache Impala – an MPP SQL query engine. The lake provides virtual data warehouse capabilities where users can create data pipelines, usually used for batch processing and data preparation for ML, or to maintain high-quality data marts that feed downstream reporting/analysis. Permission management is solved with Apache Sentry, where group membership (authZ) is provided by FreeIPA. This setup allows us to integrate a Super Collider-internal Kerberos instance (for Super Collider service users) with the VMware Corporate LDAP. This solution offered simplicity and flexibility, favoring it over other potential options, like Apache Ranger.

Workflows are deployed to IaaS Kubernetes clusters and the technology of choice for workflow management is Apache Airflow, which is exposed to users. Metadata management is delegated to Apache Atlas.

Super Collider’s Infrastructure

Administration model

Having a clearly defined layers is critical for management efficiency.

Let’s briefly describe the responsibilities of each layer and the organizational structure, so that we can see how this administration model promotes efficiency:

The lowest layer of the system, the physical layer (IaaS), is a product of OneCloud. The VMware OneCloud team is responsible for provisioning hardware and for the continuous operations of running our private IaaS Cloud.

On top of OneCloud, the Super Collider Infrastructure team is responsible for packaging and deploying the Super Collider analytics stack into the Virtual Machines (VMs) and containers within the IaaS Cloud. The team uses Terraform to manage VM lifecycles, and Ansible for in-VM configurations. The Super Collider Infrastructure team ensure that the PaaS layer offers a reproducible set of services with high availability and uptime to the upper SaaS layer.
The third layer – the SaaS layer – consists of all the in-house built software – things like control plane and all kinds of components needed to combine all the OSS and in-house built components into one system that looks seamless to the end user. This layer is developed and supported by the software development teams of Super Collider.

It is key for the middle PaaS layer to be defined by clear interfaces that include contracts and SLAs. This shapes the other two layers and guarantees that a change in any of the three layers is confined within that layer. This way when a change in one layer affects another layer communication overhead is minimal, as it is clear who breaks their contract and who is supposed to provide a fix. Also, communication overhead is reduced, because IaaS and SaaS people need not communicate as they have no touchpoints.

The collaboration between OneCloud and Super Collider is a win-win for VMware. If you have heard of Cloud Director, then you already know what OneCloud is. VMware provides datacenter design recommendations to help its partners build and operate their own VMware-powered clouds. To validate and test these designs, VMware operates large private clouds to run production workloads within the company. One such cloud is used by Super Collider. Super Collider not only provides a valuable service as a Data Analytics platform, but it also qualifies the cloud design and vSphere products offered. As the design and product improves, Super Collider can offer higher performance and availability to its customers. Super Collider has directly benefited from latest vSphere optimizations in vSAN and NSX, which have helped squeezing improved performance out of Impala. vSphere’s native support for the Kubernetes (K8S) containers delivers easy access to better performing data processing jobs. On the other hand – Super Collider team is constantly trying to minimize its OneCloud bill, thus constantly pushing the limits of the IaaS layer and sometimes opportunities for improvements emerge – either in the designs VMware recommends to customers, or in the VMware products that constitute OneCloud.

In other words, OneCloud is not something proprietary, it is just an IaaS technology stack built and operated by VMware for VMware, and VMware customers can spawn their own OneCloud-like systems by following the VMware design recommendations.

**Optimizing Compute**

The Super Collider team optimizes VM CPU utilization by running several less CPU-intensive OSS services in one VM (e.g., Hue and Zookeeper). Further optimizations have been made by the team to combine CPU-intensive and non-CPU-intensive machines on the same physical host using VM affinity rules.

The team evaluated where performance and memory consumption optimizations could be made. As expected, Impala with its metadata cache was the heaviest memory consumer (see the image below), followed by Kafka – the backbone of the data pipe. Real-time data processing workflows followed, as they work with data that is in memory is not yet persisted in the Hadoop File System (HDFS). To increase performance of the memory-hungry Impala service, the team disabled memory ballooning and memory overcommitment of the VMs that host Impala. Configuration tuning like this allowed the team to achieve the desired balance between performance, stability, and cost.

**FIGURE 5**: Super Collider memory distribution.
Managing Storage
As you would expect, utilization and management of storage in big data platform is a primary concern. The following chart shows how Super Collider services use internal block storage:

![Super Collider block storage distribution chart](image)

**FIGURE 6:** Super Collider block storage distribution.

Within Super Collider, the largest consumer of block storage is the data pipe (implemented as a Kafka service). Since the data pipe must safeguard against data loss, we rely on Kafka to handle redundancy, driving up the need for additional storage. To ensure the data pipe is performant, Kafka utilizes low latency block storage, configured by OneCloud. The second largest consumer of block storage is Impala, which does aggressive data caching (ephemeral writes). In a distributed SaaS platform like Super Collider, utilization of a virtual Storage Array Network (vSAN) to move data is vital. To eliminate the replication load of those ephemeral writes on the vSAN, Super Collider utilized **vSAN Direct Configuration™** which allows direct access to the attached storage for optimized I/O and storage efficiency.

One of the primary concerns in a big data platform is the cost to store and access the ingested data. The Super Collider data lake storage system utilizes a Dell EMC Isilon appliance to provide an HDFS service to the platform. This appliance is managed and provided as part of the infrastructure included in OneCloud. The main reason the Isilon appliance was chosen was to drive the cost point down without sacrificing scalability. Alternatively, if the decision was made to use Amazon’s S3 storage, which is widely used for data lake solutions, the total cost of ownership (TCO) would be the sum of 5 separate factors. Excluding the factor of pure storage cost per GB, which is comparable between Amazon and OneCloud, OneCloud asserted no additional costs for processing data requests and retrievals; for utilizing management and analytics features, and for pure data transfers (except for the overall network cost).

Network
Every big data platform is extremely I/O and network-intensive among its components. The generic environments offered by cloud vendors are not optimized for such scenarios. Grouping of workloads for network proximity both on virtual and physical level is critical for performance and cost optimization.

Using the latest version of VMware NSX-T Data Center allowed network segmentation to be configured across Software Defined Data Centers (SDDC), which allows the network to scale beyond the limits of a single SDDC. This option to span across SDDCs also allows for network maintenance without downtime.

Requesting dedicated physical environment and hosts that are physically close to each other may be possible with some cloud vendors, and it comes at a higher cost. Here again cost was managed by using a private cloud. With the OneCloud solution, Super Collider got this feature almost for free – we only had to explain our needs early enough to allow the OneCloud team to plan accordingly.

Management of inbound Internet traffic (for example, VMware SaaS and products deployed at customer premises pushing telemetry data to Super Collider) was important as well. Super Collider tried a few Content Delivery Network (CDN) providers, and even experimented...
without a CDN. Finally, Super Collider chose Akamai services for its ability to offer DDoS protection, invalid request filtering, and rate control.

Operations
With more than a hundred teams using the Super Collider platform, it is not uncommon for a team to double their load on the system within a few days. Additionally, during the development phase of a data product, it is not uncommon for a query or a data pipeline to be implemented in a way that is very resource-intensive. To help users utilize Super Collider’s resources efficiently, Super Collider offers showback capabilities that allow data engineers to optimize their workloads to use minimum capacity, lowering the demand for resource and overall operational cost.

Super Collider mission is to offer a self-service platform that allows any VMware employee to implement any kind of data application. This might open the door to congestion points and resource starvation. Super Collider heavily relies on Kubernetes namespaces to solve the noisy neighborhood problem. As an example, the data pipelines of one team can be physically separated from the data pipelines of another team. On the SQL side, we rely on Impala resource pools, and for the network we use VMware NSX network segmentation when needed.

Super Collider is designed to offer self-serve decentralized data management, which creates challenges in the platform’s resource planning and capacity utilization. Due to the enterprise demand, Super Collider is a centrally funded system (not coupled with an individual product team) and thus requires efficient resource planning and capacity management. Despite the usage spikes mentioned above, on aggregate level the growth curve looks smooth enough, and using Super Collider’s analytical capabilities we’ve predicted the demand accurately enough to be able to manage our budget.

Support and availability
As called out, Super Colliders success has made it a vital tool, critical for support of customer facing products and business-critical features. This has resulted in the demand for a high availability system with round the clock (24x7) on-call support. Luckily, VMware has a centralized product support team that can provide Level1 and Level 2 support to Super Collider’s customers. To augment this support, Super Collider’s software developers and infrastructure engineers, across the globe, take support shifts to provide laser focused Level3 support.

In Super Collider, a “data loss” incident is a critical event (P0), and we have designed our system architecture and backup/restore strategy in a way that prevents data loss even when there is a failure in a given part of the system. For example, Kafka topics are persisted on a solid-state drives (SSD), so that if a given broker fails, data is not lost. As an extra safeguard, a backup of Kafka data is kept for a week. This allows us to restore and replay data in case of a failure in downstream data processing, even when the problem was found hours or days later. System monitoring is mission critical. Investment in monitoring has always paid off and always will. Super Collider initially started with a monitoring solution using Grafana and InfluxDB. Recent developments and the extended usage of Kubernetes has driven the team to integrate the use of the Prometheus stack. Every component is being both monitored in isolation and via end-to-end monitoring. Monitoring starts with a single-page Grafana dashboard showing the overall system stability for user-facing components (such as SQL endpoint, Data Pipelines, ingestion endpoint, control plane, etc.). Each component panel is clickable and allows drill-down to ease troubleshooting. That main Grafana dashboard is projected on a large central display in the office.

Lessons Learned: Past and future considerations
In the above sections, we explained how the Super Collider system has been designed and built. In this section, we’ll shed some light on the reasoning behind some of our architectural decisions.

While each topic below deserves a separate article, discussing the high-level rationale and decision points would be valuable to solidify the direction Super Collider has taken. Many of the decisions ultimately come back to TCO, we’ll explore the key factors that steered our decisions in architecture.

Past Decisions
Two key decisions that drove Super Collider’s direction.

Cloud-native vs. hybrid vs. private cloud
In 2017, it was time to decide what the Super Collider deployment model would be for the next 5 years. We had to choose a vendor for lake and warehousing capabilities. The three options for vendors were Amazon, Snowflake, or building an in-house system based on open-source software.

Amazon vs. Snowflake: From a product perspective, Snowflake is an integrated solution with streamlined usage practices. Their vision is aligned with our vision for integrated and democratized lake and warehouse capabilities. On the other hand, Amazon with its multiple data services is like a LEGO® – the way you assemble the blocks determines what your solution and cost will be.
The cost of Snowflake at the scale we were in 2017 was estimated to be a bit lower than an on-prem solution. However, Snowflake did not offer an on-prem deployment, so we had to add to the total cost the effort required for migrating our existing workloads (both VMware-internal and workloads we had on Amazon). Also, given that the Snowflake data format is proprietary, we had to calculate the cost of migrating from Snowflake if we should make a change later. This made the total cost of Snowflake higher than the cost of the on-prem deployment of an OSS stack.

With Amazon, the calculation was not that simple because we had to validate and estimate several combinations of technologies including Presto, Athena, Kinesis, etc. For our use-cases, Redshift Spectrum + Glue was the winning combination, and still, the TCO of the Amazon-based stack was higher (in our case significantly higher) than running an OSS-based stack deployed and operated by VMware.

Hadoop

We had 2 main challenges with Hadoop-based components:
1. HDFS
2. YARN

With HDFS, we used to run a version that turned out inefficient above a few million files. This challenge was addressed in newer HDFS versions, so we upgraded to a newer version (back then it was Hadoop 2.6). The other challenge with HDFS was its cost. HDFS is cheap, but modern technologies can be even cheaper. For example, HDFS was originally designed to run on local drives, but running local drives in the cloud is expensive. So, we initially started with EMC VNX (shared iSCSI). Later, with the release of Dell EMC Isilon, we got the same SLAs at a 4.5x lower cost, because A) Isilon does not need a 3x replication factor, and B) Isilon is more cost-effective per gigabyte.

We had challenges utilizing the memory and CPU resources by YARN-managed workloads. Migrating to K8S resulted in savings of up to 90%, depending on the service being migrated, with an average of 50% overall, thus halving our IaaS resource utilization.

Speaking about technologies – a noteworthy recent optimization is the migration from Hadoop YARN to Kubernetes. Because we had lots of very simple data pipelines, where each pipeline needs a fraction of the resources that YARN needs to run it, migrating our data pipelines from YARN to K8S saved ~90% of the used CPU cores, and ~80% of the used memory. For example, here’s the memory and CPU consumption of the jobs that implement the vsphere Health feature before and after the migration to K8S:

<table>
<thead>
<tr>
<th></th>
<th>Streaming Workflows V1 (YARN)</th>
<th>Streaming Workflows V2 (K8S)</th>
<th>Reduction</th>
<th>Reduction %</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU CORES</td>
<td>538</td>
<td>40</td>
<td>498</td>
<td>92%</td>
</tr>
<tr>
<td>RAM (GB)</td>
<td>187</td>
<td>30</td>
<td>157</td>
<td>84%</td>
</tr>
</tbody>
</table>

Future Considerations

Read uncommitted = no isolation

As mentioned above, we picked a query engine instead of RDBMS. The problem is that this setup offers no transaction isolation (i.e. a transaction isolation level “read uncommitted” as defined by the SQL-92 standard) capabilities. Lack of transactions is a problem for every data lake out there and the problem is left to application developers to solve, so every enterprise building their own big data system is solving it somehow, to some extent. And in Super Collider we have implemented multiple workarounds to this problem, however with time and scale those workarounds become costly and less efficient. We are still looking to fill in the gap between what the market offers – file storage, and what we actually need – a table storage. And there are solutions out there, usually contributing with innovations in the contact area between the SQL engine, metadata, and storage. Take Apache Iceberg or Delta Lake as an example.

Data Federation

Data attracts data, hence the term “data gravity”. With the current rate of data growth data sources proliferate and data landscapes change. It is impossible for all the data to reside in a single data lake backed by a single file system.

Trying to keep Super Collider as a single filesystem-backed data lake is a doomed mission.
There are attempts in different directions that are aiming at providing a federated view over datasets. Distributed query engines like Presto solve the “data read” problem quite well, but the challenge with writes and transactions is still a work in progress.

Another approach is data orchestration, e.g. Alluxio.

Cloudera has invested in a toolset that makes it easy to copy data together with its metadata and permissions, etc., which makes it easy to solve use-cases that require subsets of all the data in the lake. Although data copying is a workaround, not a solution - this workaround is good enough when the data volume is in the GB ranges.

In Super Collider, we are still searching for a solution. Ideally, data orchestration, data federation, and transactional support will be solved at once. Clearly, the paradigm of a centralized data lake is getting increasingly more expensive to maintain. Only time will tell whether a more distributed approach, such as data mesh is the right direction to go.

Stay tuned!

In Conclusion

Data volume worldwide is getting bigger exponentially. At the same time the need for data-assisted capabilities supporting the business is becoming vital. To address those business needs, enterprises need a system, and the key question is “what system”. Given the technology landscape it is obvious that the technological part of the answer is far from trivial. Besides the technological aspect there is also a variety of possibilities for the operational model of the system, too.

Initially it might sound tempting to go with hyperscalers, but hyperscalers are expensive, especially when system matures [8] and it is expensive to migrate away. Initially it might sound like a safe bet to lean towards the traditional centralized operational model, but it lacks the agility needed by the dynamic world of business.

After spending years designing, building, operating, and evangelizing such analytical system we believe that operating a private cloud based on VMware stack and using OSS is the technological approach that gives agility at a good price. For this approach to be sustainable the interfacing surfaces provided to system users should be very carefully selected, so that it is possible for the underlying technical solution to be changed transparently to (or with a very minimal cost for) the users. And to maximize the potential of the data – the self-serve model is key step to achieving data democratization. From our experience – system users are most effective when organized in data mesh-like teams. In the last chapters of this article, we also shared some technical details and decisions, with the hope to spare you effort that we've already made.

Bibliography


