Sizing Splunk SmartStore - Spend Less and Get More out of Splunk

Make your infra $$ work harder for you
Bharath Aleti
Director Product Management | Splunk Inc.
Use this if there will be two speakers for your session.

Jon Rust  
Splunk Architect | ADP

Jane Joki  
Offering Manager | IBM Cloud Object Storage
During the course of this presentation, we may make forward-looking statements regarding future events or plans of the company. We caution you that such statements reflect our current expectations and estimates based on factors currently known to us and that actual events or results may differ materially. The forward-looking statements made in the this presentation are being made as of the time and date of its live presentation. If reviewed after its live presentation, it may not contain current or accurate information. We do not assume any obligation to update any forward-looking statements made herein.

In addition, any information about our roadmap outlines our general product direction and is subject to change at any time without notice. It is for informational purposes only, and shall not be incorporated into any contract or other commitment. Splunk undertakes no obligation either to develop the features or functionalities described or to include any such feature or functionality in a future release.
Splunk SmartStore ?
Data Technology Transitions
Trending with data growth and business needs

1990s
Gigabyte-Terabyte Era
Bring Data to Compute
Databases and File-systems

Data Ponds

2005+
Terabyte-Petabyte Era
Bring Compute to Data
Hadoop, Splunk, DistFS

Data Lakes

2018+
Exabyte-Zetabyte Era

Data Oceans

Will the same scale-out colocation model work?
Growing data volumes requires $$$ infra spend

Adding new indexers in response to data growth is expensive => High cost
Searches typically run over only a partial subset of data => Inefficient utilization
Distributed scale out architecture => No longer a good fit for growing data volumes
Splunk SmartStore
Achieve massive scale with lower TCO

- Decoupled compute and storage
- Scale storage for longer retention & indexers on performance demand
- Reduced indexer footprint for warm/cold data

Lower TCO

- Brings in data closer to compute on-demand
- Application and data aware cache
- Cache data based on age, priority and access patterns

Performance at Scale

- Faster indexer recovery
- Faster data rebalance

Faster Failure Recovery

- Add/remove indexers on-demand
- Setup/teardown cluster on-demand

On-Demand Cluster
SmartStore Overview
SmartStore
Decoupled Compute and Storage

- Decoupled storage and compute
- Warm/cold data in remote storage
- Hot and recently access data on indexers

- Longer data retention by independently scaling storage
- Scale out compute based on performance demands
- Lower TCO with S3 & S3 API compliant object stores

S3 or S3 API compliant object stores

Decoupled Compute and Storage

Search Tier

Indexer Tier

Remote storage (warm/cold data)
SmartStore
Reduced Indexer Footprint & Faster Node Recovery

- 1 Full copy + RF-1 Metadata copies of warm/cold on indexers
- Fewer indexers required with only one full copy of warm/cold
- Faster node recovery & data rebalance with metadata copy

Search Tier

Indexer Tier (Hot and recent data)

Remote storage (warm/cold data)

S3 or S3 API compliant object stores
SmartStore
Application & data aware cache brings in data on-demand

Search Tier

Indexer Tier (Hot and recent data)
- Hot and recently access data on indexers
- Cache Manager
  - A1, A2, A3, B1, C1
  - B2, B3, C2
  - C3
- Remote storage (warm/cold data)
- Warm/cold data in remote storage
  - B1, B2, B3, B4, B5, B6, B7, B8
  - C1, C2, C3, C4, C5, C6, C7, C8
  - Bn
- S3 or S3 API compliant object stores

- Fills up indexer storage cache until available capacity
- When cache is full, buckets are evicted based on LRU, data age and priority
- Loads active dataset on indexers
SmartStore Cache Manager
Similar to CPU memory caching

Cache and Main Memory

(a) Simple cache:

(b) Three-level cache organization:

Splunk
Hot/Cache Storage
Remote Storage
SmartStore Architectural Advantages

Storage Tier is no longer tied to hardware
- Separation of storage and compute
- Indexer failures is no longer tied to storage failure

Local Storage is now simply a Search-Cache
- No longer need to size local storage to hold long-term retention
- Just need enough local storage for search
  - Majority of searches are typically over last 7 days
Monitoring Console Additions

SmartStore Activity: Instance

Online

Remote Storage Connectivity

ONLINE

Migration Progress

100

Bucket Activity

Metric

Each new bucket adds uploaded to remote storage. Buckets are only downloaded if they’re required for a search and are not already in the local cache. High download rates are a negative indicator of cache health.
SmartStore Architectural Advantages

Scalability & High Availability
- Architectured for massive scale
- High data availability with remote storage tier
- Performance at scale with cached active dataset

TCO Reduction
- Scale compute and storage independently
- Lower TCO with reduced indexer footprint
- Leverage cost benefits of cloud/storage innovations

Simplified Management
- Instant indexer failure recovery
- Faster data rebalance
- Upgrade/replace indexer infrastructure with simple bootstrap from remote store
- New global size based retention policies
SmartStore in Production

• 95% of Splunk Cloud prod stacks running on SmartStore

• Successful adoption at key customer accounts and more in the pipeline
  • ADP, Lawrence Livermore National Labs speaking at Conf ….
  • 100+ on-prem deployments based on Splunk telemetry and support info

• Quotes
  • “SmartStore working like a dream”
  • “Saving many millions per year in AWS storage”
  • “No longer worried about running out of disk space for long term retention”
Sizing, Performance & TCO
SmartStore Cache Sizing Guidelines

- Daily Ingestion Rate (I)
- Search timespan for majority of your searches
  - Cache Retention (C) = 1 day / 10 days / 30 days or more
- Available disk space (D) on your indexers (assuming homogenous disk space)
- Replication Factor (R) = 2
- Min required cache size: \[I*R + (C-1)*I\]
- Min required indexers = Min required cache size / D
- Also factor in ingestion throughput requirements (~300GB/day/indexer) to determine the number of indexers

<table>
<thead>
<tr>
<th>SmartStore Sizing Summary</th>
<th>1TBDay_7DayCache</th>
<th>1TBDay_10DayCache</th>
<th>1TBDay_30DayCache</th>
<th>10TBday_10DayCache</th>
<th>10TBDay_30DayCache</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ingest/Day (GB)</td>
<td>1,000</td>
<td>1,000</td>
<td>1,000</td>
<td>10,000</td>
<td>10,000</td>
</tr>
<tr>
<td>Storage/Indexer (GB)</td>
<td>2,000</td>
<td>2,000</td>
<td>2,000</td>
<td>2,000</td>
<td>2,000</td>
</tr>
<tr>
<td>Cache Retention</td>
<td>7</td>
<td>10</td>
<td>30</td>
<td>10</td>
<td>30</td>
</tr>
<tr>
<td>Replication Factor</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Min Required Cache (GB)</td>
<td>8000</td>
<td>11000</td>
<td>31000</td>
<td>110000</td>
<td>310000</td>
</tr>
<tr>
<td>Min Required #Indexers</td>
<td>4</td>
<td>6</td>
<td>16</td>
<td>55</td>
<td>155</td>
</tr>
</tbody>
</table>
Performance: Cache Miss
Performance: Cache Miss

100% cached: Search time grows linearly along with time range

Cache miss: Sharp spikes when hitting non-cached data
- Impact is lower for dense searches due to data locality and prefetch
- On a cache miss, the search time may increase from 2s to >100s, depending on the search
  - E.g. To fetch a single bucket of 750MB on 1 Gbps network, the latency is 7.5s.
  - Prefetching reduces the overall search response impact by overlapping with CPU/IO operations
Impact of Network Latency

Upload/migration:
- 0ms latency: 500MB/s (1.5s per 750MB bucket)
- 30ms latency: 100MB/s (7.5s per 750MB bucket)
- 100ms latency: 30MB/s (25s per 750MB bucket)

Download/localization:
- 0ms latency: 800MB/s (0.94s per 750MB bucket)
- 30ms latency: 100MB/s (7.5s per 750MB bucket)
- 100ms latency: 30MB/s (25s per 750MB bucket)

- Total impact is lower with parallel download/upload
- By default, Splunk will upload/download 8 operations at a time.
- With multi-part upload, this will be 48 operations in parallel
Object Store Performance Specs

Object Store to per-Splunk-indexer throughput

<table>
<thead>
<tr>
<th></th>
<th>Minimum Specs</th>
<th>Performance Specs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Download Throughput</td>
<td>100MB/s or higher</td>
<td>800MB/s or higher</td>
</tr>
<tr>
<td>Upload Throughput</td>
<td>30MB/s or higher</td>
<td>500MB/s or higher</td>
</tr>
<tr>
<td>Network Connectivity</td>
<td>1Gbps or higher</td>
<td>10Gbps or higher</td>
</tr>
</tbody>
</table>

Scalable/modular network backplane of the Object Store

- Must support network connectivity reqs of all connected indexers
- e.g. for 100 indexers with minimum specs, the backplane must support 100Gbps or higher

Object Store must support at least 1K per second API operations

- (GET/PUT/POST/DELETE) operations to a bucket
## SmartStore Cost Savings

Reference only, may vary based on your pricing

<table>
<thead>
<tr>
<th>Deployment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ingestion Rate: 1TB/day</td>
</tr>
<tr>
<td>Total Retention: 365 days</td>
</tr>
<tr>
<td>Replication Factor: 2</td>
</tr>
<tr>
<td>Max Search Concurrency: 64</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Non-SmartStore Infrastructure Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>At 1TB/day for 365 days and RF=2, storage capacity req is 365TB</td>
</tr>
<tr>
<td>With 12TB per indexer, this would require 31 indexers</td>
</tr>
<tr>
<td>At a server cost of $12K/year, this comes to $374K</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SmartStore Infrastructure Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>With 30 days cache retention, indexer footprint is reduced to 8</td>
</tr>
<tr>
<td>With 2TB per indexer (SSD), annual cost of indexers is $43K</td>
</tr>
<tr>
<td>Storage cost is $46K cost/year, with total cost = $90K</td>
</tr>
<tr>
<td>SmartStore approx cost savings: 75%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Non-SmartStore Infrastructure Costs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non-SmartStore Server On-demand Pricing/Hr</td>
</tr>
<tr>
<td>Non-SmartStore Server Cost/Year</td>
</tr>
<tr>
<td>Non-SmartStore Storage Per Node (GB)</td>
</tr>
<tr>
<td>Non-SmartStore Indexers Required</td>
</tr>
<tr>
<td>Non-SmartStore Indexer Cost/Year</td>
</tr>
<tr>
<td>Non-SmartStore Total Cost/Year</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SmartStore Infrastructure Costs</th>
</tr>
</thead>
<tbody>
<tr>
<td>SmartStore Server (SSD) On-demand Pricing/Hr</td>
</tr>
<tr>
<td>SmartStore Server (SSD) Cost/Year</td>
</tr>
<tr>
<td>SmartStore Cache Required</td>
</tr>
<tr>
<td>SmartStore Min Indexers Required</td>
</tr>
<tr>
<td>SmartStore Indexer Cost/Year</td>
</tr>
<tr>
<td>SmartStore remote storage pricing/GB/month</td>
</tr>
<tr>
<td>SmartStore Remote Storage Cost/Year</td>
</tr>
<tr>
<td>SmartStore Total Cost/Year</td>
</tr>
</tbody>
</table>

More performance => Add indexers
More storage Capacity => Add storage

Cost savings go down with increase in number of indexers and increases with higher ingest rate/retention requirements
SmartStore in Production at ADP

Jon Rust
Splunk Admin.
Overview - Usage

20 TB license, 11 TB avg day, 19 TB recent peak

500 TB of retention (growing since implementing S2)

600,000 searches per day
  • Avg runtime 4.0s, unchanged since S2

5500 users

80 groups (each group gets a Splunk app)

1000 indexes (each group gets multiple indexes)
  • Largest cluster has 300
Overview - Infrastructure

72 physical indexers, 2 VM (lab) in 7 environments
  • Largest clusters are 25 and 29 indexers

16 VM search heads
  • Largest cluster is 9
Overview – Basic Cluster

Most traffic still comes through SUF

Growing HEC, close to 50% lately

Separate HEC HF farm
  • Flexibility
  • HEC overuse doesn’t impact indexers

COS: Cloud Object Store from IBM
  • Formerly known as CleverSafe
Overview – Production
“Indexers are too expensive”

Management unhappy with the cost of Splunk

- $50k per indexer, 20 cores
- 15 TB of usable RAID10 SSD

With SmartStore (S2)

- $12k per indexer, 36 cores
- 7 TB of usable RAID0 SSD
  - BUT! S2 redundancy
- COS disk cost is about $0.35/GB
- 2x indexer count, almost 4x core count
  - Still < 50% the $$
More than money management: Agility!

• Increase or decrease peer count very quickly
• Random other example, “re-RAID project Q12019”
  – Management forced us to use RAID5 during initial build-out
  – RAID5 needs to die in a fire
  – We eventually hit the IO wall
  – With S2, rebuilding RAID volumes was pretty painless!

```
splunk offline
Take mount offline, rebuild the volume as RAID10
splunk restart
<repeat for each indexer>
```

12 indexers in the cluster, less than 2 hours of work, no service interruption
But how does it search?

Most common searches are unchanged

- Recent data is in cache, performs exactly as before but faster with more h/w
- Historic searches are okay, depends
  - Big window searches over old data can trigger large downloads from remote store
- We’ve had zero complaints about search performance since updating to S2
  - Most users have no idea
Was migration difficult?

Mostly turn-key

• A few beta/early release issues (since solved)

• When migrating a cluster
  – Chose 1 index first and verified
  – Good? Chose 5 more and verified
  – Good? Rolled the rest

• Upload concurrency during migration
  – We turned this down (from default of 8, to 4)
  – Our COS infra wasn’t designed to handle so much upload data all at once
  – Consider your network and S3 limits before migration
  – Normal day-to-day use spreads out uploads pretty nicely
Sample config

[volume:remote_store]
storageType = remote
path = s3://splunk-s2-webtier-dc2
remote.s3.access_key = **key**
remote.s3.secret_key = **key**
remote.s3.endpoint = https://internalS3.endpoint
remote.s3.signature_version = v2

[some_index]
remotePath = volume:remote_store/$_index_name
homePath = volume:hot/$_index_name
maxGlobalDataSizeMB = 175000
frozenTimePeriodInSecs = 12096000
# required, but only used during migration; no data will land here after migration
coldPath = volume:cold/$_index_name
Dashboard: SmartStore Traffic

https://github.com/camrunr/s2_traffic_report
Splunk SmartStore and IBM Cloud Object Storage

A Gamechanger for Your Splunk Environment

Jane Jokl
Offering Manager, IBM Cloud Object Storage Solutions
Topics

• Brief Overview of IBM Cloud Object Storage
• Solution Highlights
• Key Takeaways
Efficiency of IBM Cloud Object Storage

Example: How to build a highly reliable storage system for 1 Petabyte of usable data?

RAID 6 + Replication

Original
1.20 PB Raw

Onsite mirror
1.20 PB Raw

Remote copy
1.20 PB Raw

Software Defined Solutions

<table>
<thead>
<tr>
<th>1 PB</th>
<th>Usable Storage</th>
<th>1 PB</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.6 PB</td>
<td>Raw Storage</td>
<td>1.7 PB</td>
</tr>
<tr>
<td>900</td>
<td>4TB Disks</td>
<td>432</td>
</tr>
<tr>
<td>3.6x</td>
<td>Racks Required</td>
<td>1.7x</td>
</tr>
<tr>
<td>3.6x</td>
<td>Floor Space</td>
<td>1.7x</td>
</tr>
<tr>
<td>3 FTE</td>
<td>Ops Staffing</td>
<td>.5 FTE</td>
</tr>
<tr>
<td>Replication/backup</td>
<td>Extra Software</td>
<td>None</td>
</tr>
</tbody>
</table>

TCO Savings

70% + $
Why is Cloud Object Storage a good fit for Unstructured Data?

IBM Cloud Object Storage Industry Leader
IDC and Gartner Market leader for over 6 years

Simplified Distributed Architecture
Access from anywhere
Reduce points of failure
Enhanced durability w/ consistency checks

Simplify management
Much less to tune (no controller nodes or replication)
No snapshots or backup copies

Virtually infinite scalability
Scale Capacity to Exabytes
Flexible addition/removal

Reduced cost
Commodity hardware
Single copy protection

No file system limitations
Number of files per directories – no limit
Total objects in a volume and max size
Single volume max capacity

Custom metadata
Ready for AI/Analytics
Stored with object for new use cases

Notes:
• All deployment models supported – On Premise, Hybrid, Public Cloud
• Available as Software only; Supported on approved customer x86 platforms
• IBM appliances also available
How IBM Cloud Object Storage Works

Content Transformation
IBM COS software encrypts, slices and applies Information Dispersal Algorithms, otherwise known as erasure coding policies to the data.

Data Ingest

Accesser Software

Physical Distribution
Slices are distributed to separate disks on industry standard x86 hardware across geographic locations.

Site 1  Site 2  Site 3

Storage Nodes

Data Retrieval

Benefits
The level of resiliency is fully customizable resulting in a massively reliable and efficient way to store data at scale as opposed to RAID and replication techniques.

Reliable Retrieval
An operator defined subset of slices is needed to retrieve data bit perfectly in real time.

Site 1  Site 2  Site 3

Storage Nodes
Example of 1PB Data Use Case with SmartStore and COS

COS Configuration
- IDA: 12/7/9
- Data Reliability: 10 9's
- Expansion: 1.71
- 12 TB HDDs
- Usable: 1008 TB
- Primary Raw: 1728 TB
- Managers: 1
- Accessers: 6
- Slicestors: 12

- Number of Accessers can be scaled to handle throughput
- Each accesser handles approx 750MB/sec; varies depending on object size
- Slicestors can be scaled for capacity
Highlights of Splunk SmartStore with IBM COS

Splunk administrators can seamlessly increase storage as well as storage performance with IBM COS without having to scale up compute at the same time.

Both Splunk and IBM COS highly flexible and extremely scalable without any downtime:

- Scaling COS performance is as simple as adding more Accessers serving the storage pool.
- If the dsNet becomes storage pool constrained, IBM COS allows realtime addition of additional sets of Slicestors to the storage pool to increase storage pool performance.
- Additional method of scaling performance from a COS perspective: use SmartStore’s ability to have different endpoints for each volume; Ex: One set of indices use one dsNet, and other indices use another dsNet.

Performance:

- Can be as performant as Splunk’s traditional architecture – minimal performance delta with SmartStore remote storage.
- ADP use case success story.

Benefits of On Prem deployments:

- Less capacity costs.
- No retrieval charges (egress bandwidth and operational requests).
- Higher reliability.
- Data in your control.
- Performance you control and more predictable.
Unlock the Value of Splunk SmartStore with IBM COS Key Takeaways

Take advantage of the SmartStore feature in Splunk Enterprise which has native S3 integration with IBM Cloud Object Storage

Lower TCO
- Scale Warm tier (IBM COS) independent of adding more indexing servers
- Optimize Hot tier Servers for Performance

Extend Data Retention and Maximize Data Accessibility
- Hot tier remains the same as classic architecture
- Everything else is IBM COS which is WARM and SEARCHABLE (Warm/Cold = Warm)

Agility of Infrastructure – Data not tied to Servers; No Downtime; Seamless Scalability

Take advantage of intrinsic HA capabilities provided by IBM COS as Warm tier remote storage

Simplify Data Management and Deployment model with only 2 tiers – Hot and Warm

Architected for Massive Scale

No size limitations on ingest with SmartStore; Setup parameters will need to be set according to either architecture

Can be implemented on a per Index basis, i.e. deployments do not have to be “all Classic” or “all SmartStore”
Key Takeaways

Splunk SmartStore

1. Decoupled compute and storage w/ SmartStore provides scale and performance at low cost

2. Supported with both cloud and on-prem object storage

3. Drives business insights with longer retention and large data volumes
Thank You!

Go to the .conf19 mobile app to RATE THIS SESSION
Q&A
Alternatives

Option #1: Reduce data retention or reduce ingest rate
Option #2: Multiple data copies in NFS (dedup offers respite)
  - Searches over older datasets limited by NFS network bandwidth