TCO Reduction Through Storage

Mustafa Ahamed

Director, Product Management

Ashish Mathew

Software Engineer

.conf2016

splunk>

Disclaimer

During the course of this presentation, we may make forward looking statements regarding future events or the expected performance 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 could differ materially. For important factors that may cause actual results to differ from those contained in our forward-looking statements, please review our filings with the SEC. 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, this presentation may not contain current or accurate information. We do not assume any obligation to update any forward looking statements we may make. 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 functionality described or to include any such feature or functionality in a future release.

Agenda

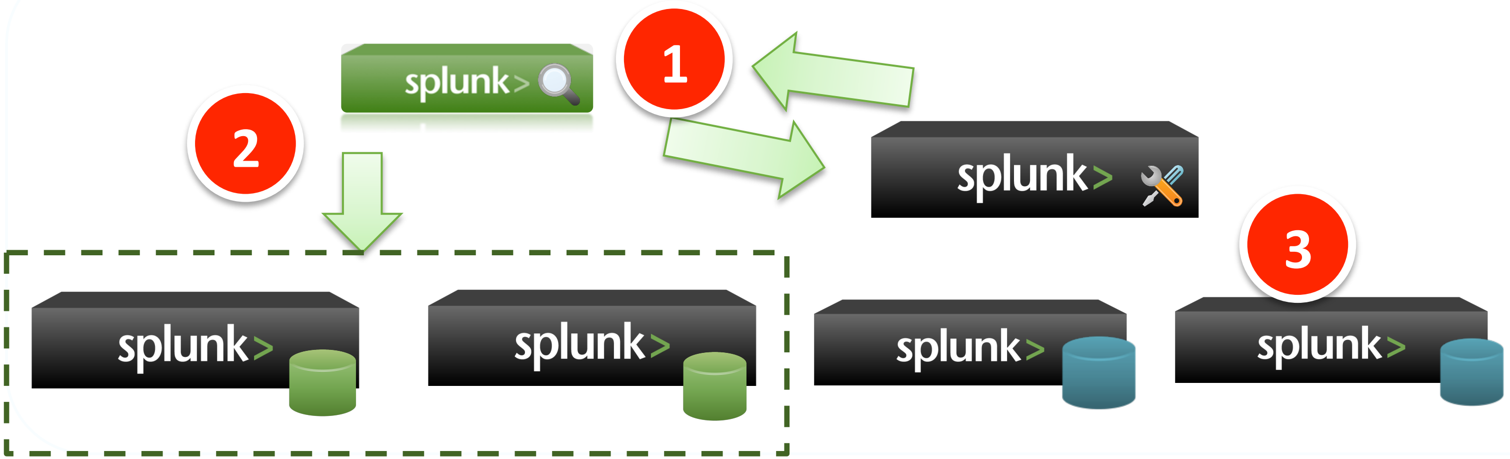
- Introduction To Data Storage In Splunk
- TSIDX Reduction – Overview
- TSIDX Reduction – Set Up
- Performance Comparisons
- Tips & Tricks

Introduction To Data Storage In Splunk

.conf2016

splunk>

Splunk Architecture

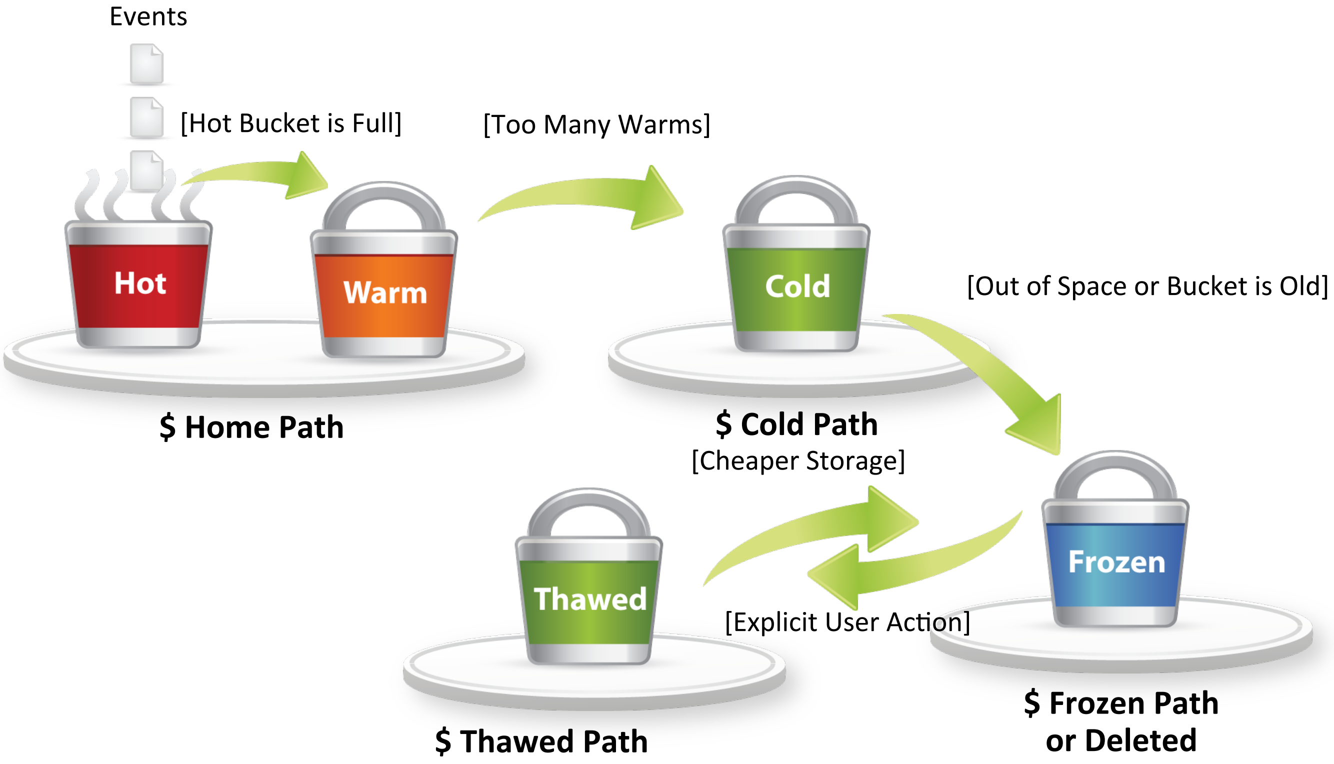


1 Search Head gets the peer list from Cluster Master

2 Search Head sends the search queries to peers

3 Redundant copies of raw data are available

Bucket Lifecycle



Storage Requirements

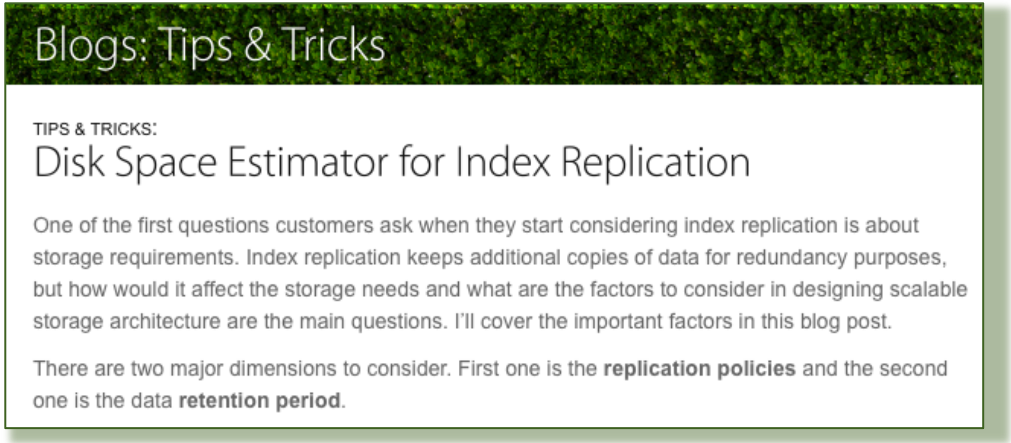
Raw data on disk = ~ 15% of indexed data
Index files on disk = ~35% of indexed data

Index data = 100GB, RF = 3, SF = 2

- Raw data = 15 * 3 = 45 GB
- Index files = 35 * 2 = 70 GB

Total size across cluster = 115 GB

Per peer storage = 38 GB



Blogs: Tips & Tricks

TIPS & TRICKS:

Disk Space Estimator for Index Replication

One of the first questions customers ask when they start considering index replication is about storage requirements. Index replication keeps additional copies of data for redundancy purposes, but how would it affect the storage needs and what are the factors to consider in designing scalable storage architecture are the main questions. I'll cover the important factors in this blog post.

There are two major dimensions to consider. First one is the **replication policies** and the second one is the data **retention period**.

http://blogs.splunk.com/2013/01/31/disk-space-estimator-for-index-replication/

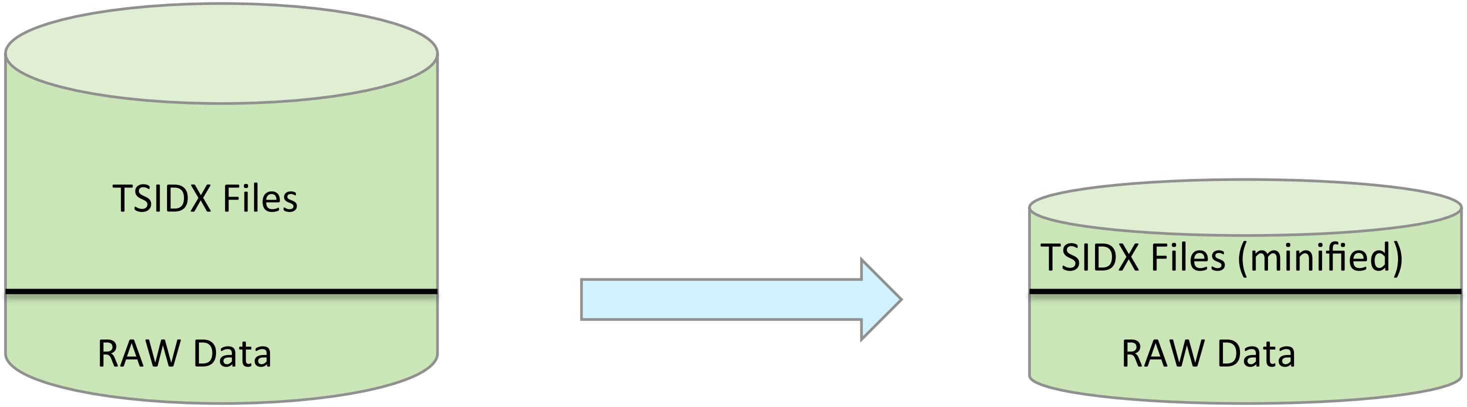
TSIDX Reduction Overview

.conf2016

splunk>

TSIDX Retention Policy

Ability to remove TSIDX file contents for historical data to save disk space



Deep Dive

Reduce What ?

- Lexicon and Postings list

  Raw data:
    - Event 1: Happy kitty
    - Event 2: Sad kitty
  - Lexicon:
    - Happy: Term-id 1
    - Kitty: Term-id 2
    - Sad: Term-id 3

  - Postings List:
    - Term-1:
      - [Event-1]
    - Term-2:
      - [Event-1,Event-2]
    - Term-3:
      - [Event-2]

So How Do We Search ?

- Brute Force !
  - Read EVERYTHING from disk, filter raw in memory

- Some optimizations by retaining the following
  - Bloom filters : Eliminate buckets that do not contain the terms
  - Reduced TSIDX : Eliminate events that fall outside the time range
  - *.data Files : Eliminate events that don't match host/source/sourcetype

Won’t Searches Be Slow ?

- It Depends !!!
  - Dense searches not affected at all
  - Sparse searches affected significantly

- Assumption : Old data is less searched
- Before configuring determine a cutoff point

Numbers

- Disk Savings : 60-70% on average
  - Better for numerical data
  - Better for larger lexicons

- Search Times:
  - Dense : Not affected
  - Sparse/Rare
    - Goes from seconds to minutes
    - Scales with data volume

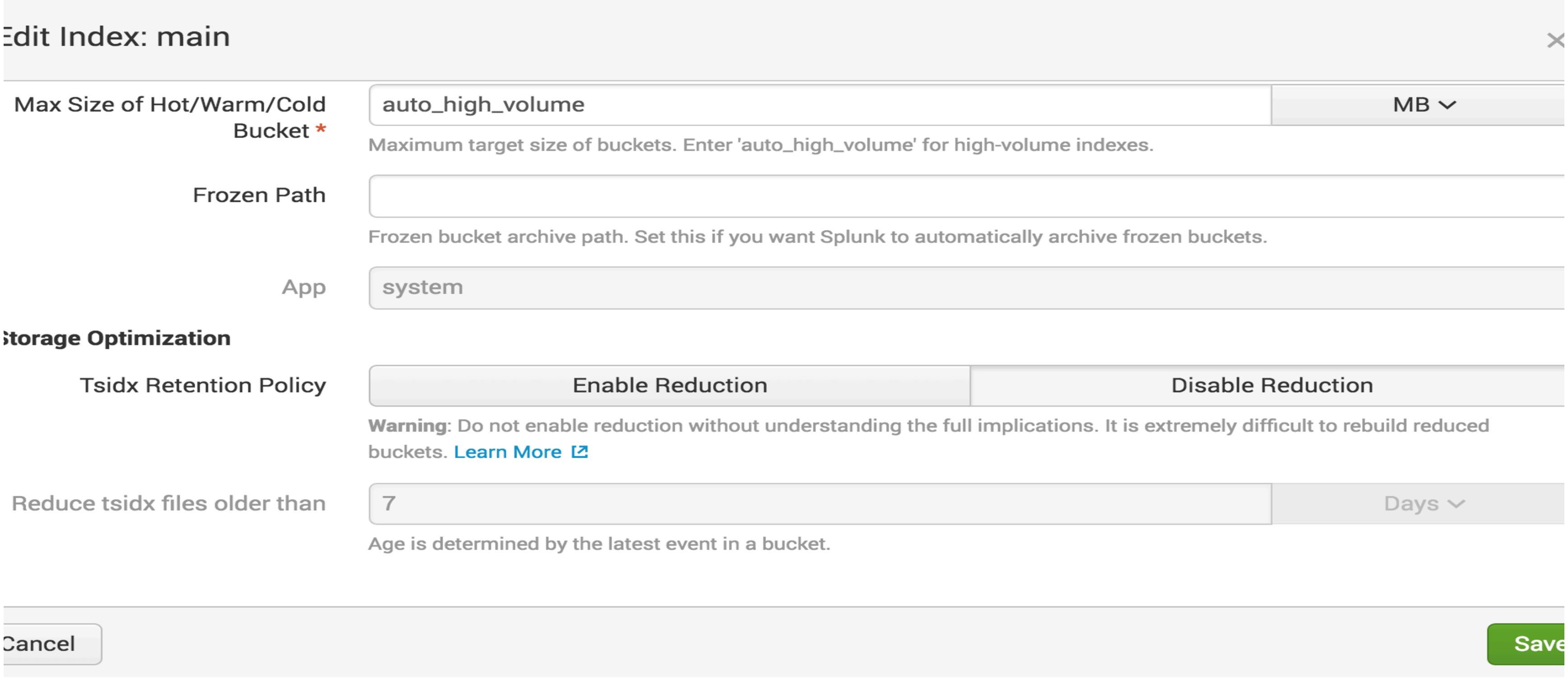
Configuration

Per-index settings in indexes.conf

REST/CLI/UI: No restart required

- enableTsidxResuction : true|false
  - Enable the feature. Off by default
- timePeriodInSecBeforeTsidxReduction
  - Age at which bucket eligible for reduction
- tsidxReductionCheckPeriodInSec
  - Frequency of scans for eligible buckets

UI



Edit Index: main

Max Size of Hot/Warm/Cold Bucket * auto_high_volume MB

Maximum target size of buckets. Enter 'auto_high_volume' for high-volume indexes.

Frozen Path

Frozen bucket archive path. Set this if you want Splunk to automatically archive frozen buckets.

App system

Storage Optimization

Tsidx Retention Policy: Enable Reduction | Disable Reduction

Warning: Do not enable reduction without understanding the full implications. It is extremely difficult to rebuild reduced buckets. Learn More

Reduce tsidx files older than 7 Days

Age is determined by the latest event in a bucket.

Reduction Process

- Eligibility
  - Bucket is not HOT
  - No more splunk-optimize runs scheduled on the bucket
  - Bucket is the right age

- Create reduced files in a tmp directory in the bucket
- Copy over reduced files, delete the full files
- Ongoing searches uninterrupted
- **NOTE:** Marginal disk usage increase when first enabled

DANGER !

- Once a bucket is reduced going back is very expensive

- Two ways:
  - Disable reduction, then wait for the reduced buckets to be phased out
  - Stop Splunk and rebuild the bucket

Clustering

- indexes.conf Is consistent across slaves.
- Reduction does not happen in lock step across all slaves
- *Eventually* all copies of the bucket will have the same state across peers
- Bucket is *SEARCHABLE* if it has either full or mini-TSIDX files

Debug Options

- Undocumented CLI to manually minify a specific bucket
  - Stop splunk
    - splunk fsck minify-tsidx --one-bucket --bucket-path=<path>

- New field in dbinspect :
  - tsidxState : full | mini

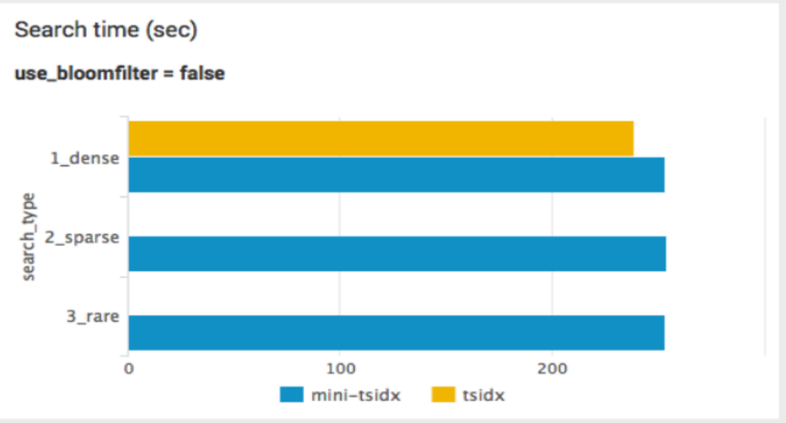
- Log channels
  - Minification scheduler
    - category.OnlineFsck
  - New filtering layers in Search
    - category.ISearchOperator
    - category.FastSearchFilter
    - category.LispyPostFilter

Performance Testing Results

.conf2016

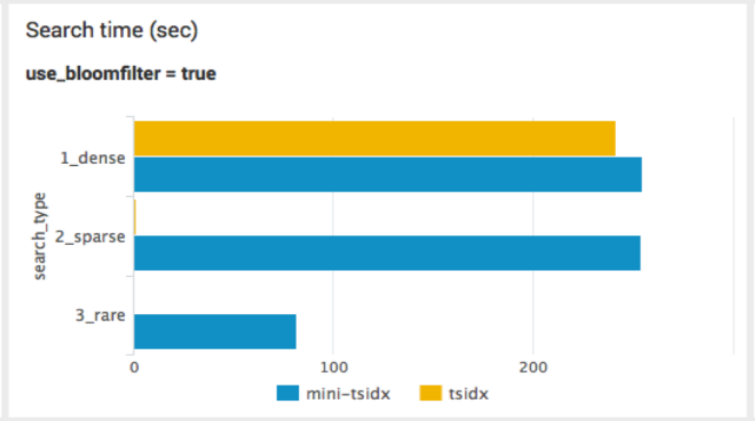
splunk>

Performance



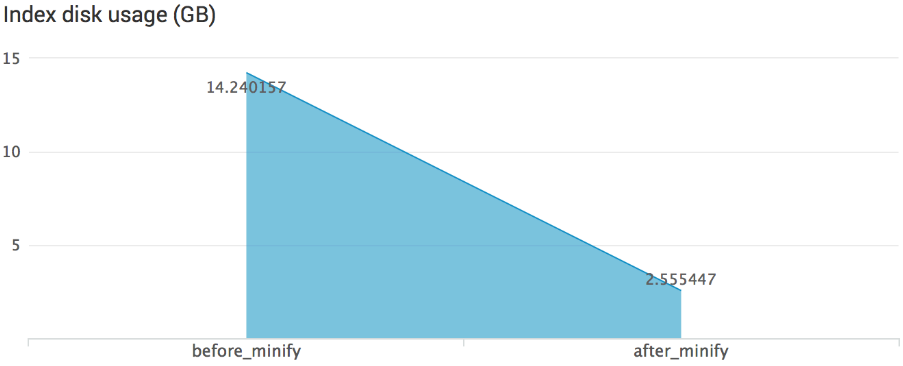
Search time (sec)

use_bloomfilter = false

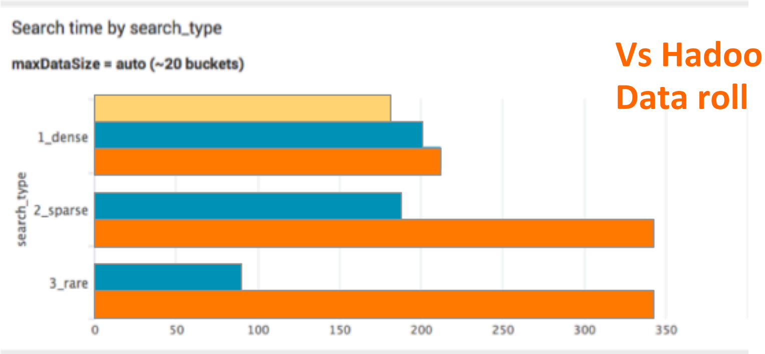


Search time (sec)

use_bloomfilter = true



Index disk usage (GB)



Search time by search_type

maxDataSize = auto (~20 buckets)

Vs Hadoop
Data roll

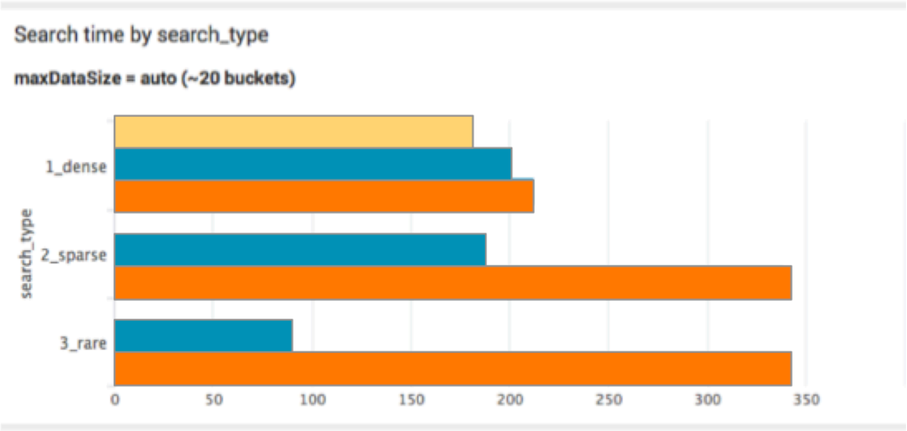
Comparison To Hadoop Data Roll

.conf2016

splunk>

Hadoop Data Roll

- Moves raw data from Splunk to Hadoop infrastructure
- Useful if you already have Hadoop in your env
- Performance wise TSIDX reduction is faster due to Bloom filters



Best Practice Recommendations

.conf2016

splunk>

Key Details

- Per-index configuration
  - Can be enabled globally or per-index basis
- Cluster-aware
- Bloom filter
  - Always Use Bloom Filters
- Performance

THANK YOU

.conf2016

splunk>