

Rebalancing Data Across An Indexer Cluster

Bharath Aleti

Sr. Product Manager, Splunk

Da Xu

Sr. Software Engineer, Splunk

.conf2016

splunk>

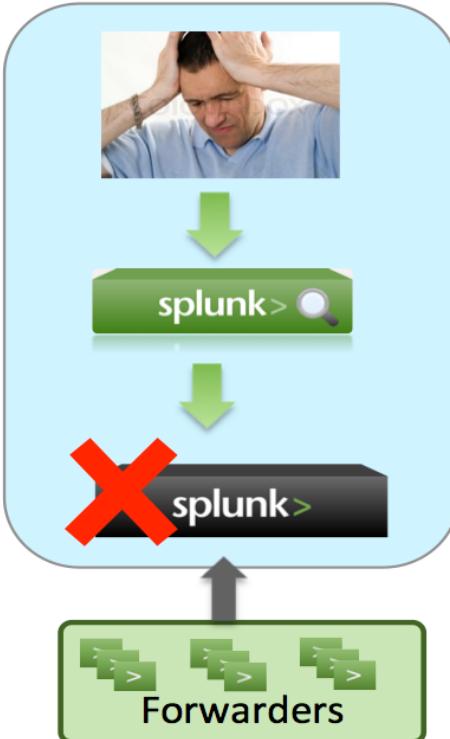
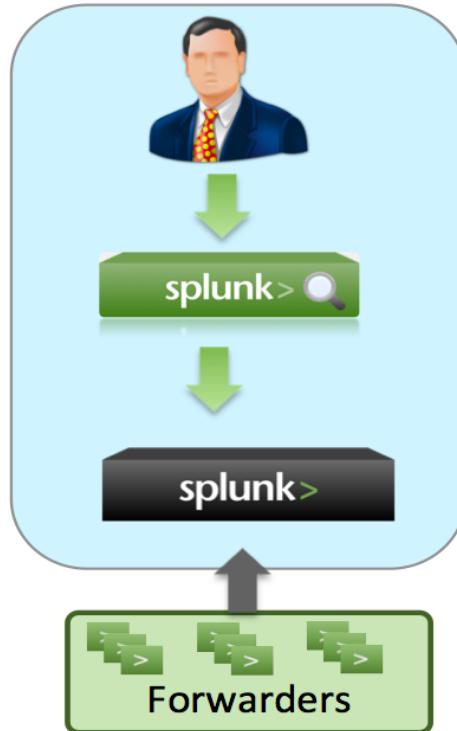
Disclaimer

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 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

- What Is Indexer Clustering ?
- Why Rebalance Data ?
- What Is Data Rebalance ?
- How Does It Work ?
- Benefits And Limitations
- Demo
- Q&a

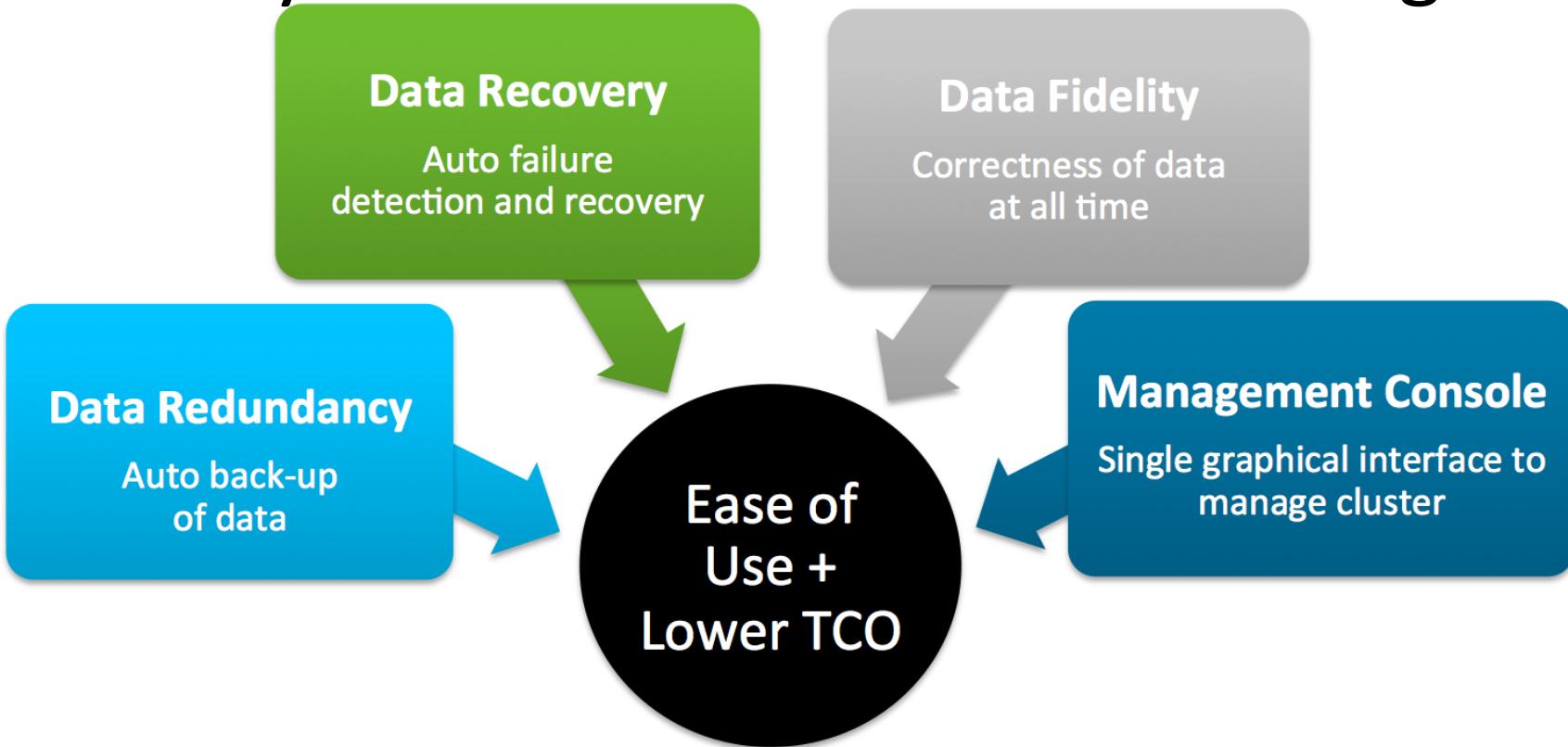
Why Do We Need Clustering ?



Work-Arounds

- 1. Index and Forward**
 - Addl. Licensing Costs
- 2. Simultaneous forward**
 - Data Sync Issue

Key Benefits Of Indexer Clustering



Why Rebalance Data ?

.conf2016

splunk>

Uneven Data Distribution

- Addition of new indexers
- Forwarders sending incoming traffic to select indexers
- Multiple forwarders randomly choosing select indexers
- Unbalanced Multisite configuration
- Random node selection by indexer replication
- Node failure or going offline

- Higher load on existing indexers
- Poor utilization of new indexers
- Node detention, on reaching max available storage on a single indexer
- Incomplete searches due to nodes in auto-detention state
- Higher per node storage requirement

Existing Guardrails

- Use Indexer discovery to load-balance across all available indexers
- Monitor/Update '*minFreeSpace*' (*diskUsage stanza*)
- Random replication evens out over a period of time
- Setting '*advertised_disk_capacity*' to specify static threshold w/ indexer discovery
- Investigate reasons for biased data forwarding
 - Note: chunk breaker (Ivory) solves the case for large inputs

What Is Data Rebalance ?

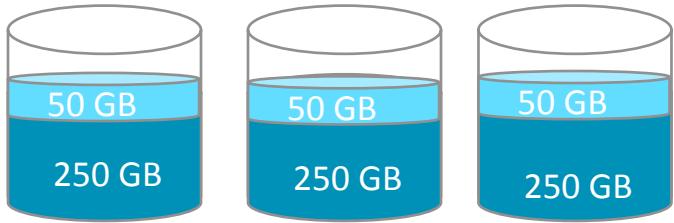
.conf2016

splunk>

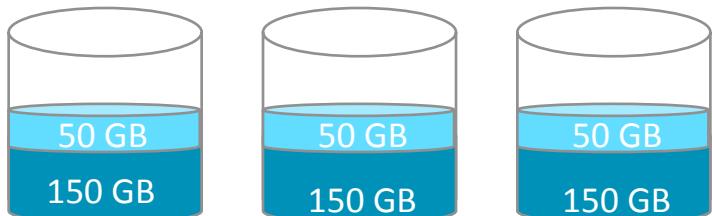
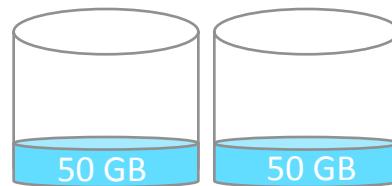
Data Rebalance

Balanced Data Distribution between New and Existing indexers

Existing Indexers



New Indexers



Problem

- Uneven data distribution after adding new indexers
 - Higher load on existing indexers
 - Poor utilization of new indexers
- Even data and search load distribution
- Lower storage requirement per node
 - Improved search performance

Data Rebalance

- Redistributions bucket copies so that each peer has approximately the same number of buckets (within a given threshold)
 - Rebalances all (non-searchable, searchable and primary) buckets
 - Supports multi-site cluster configurations
-
- Option to monitor the status of data rebalance
 - Configurable to tune desired rebalance threshold for optimal storage utilization
 - Optional timer to stop rebalance, within a given time period.

Data Rebalance CLI And REST Calls

- `splunk rebalance cluster-data -action start [-index index_name] [-max_runtime 100]`
- `splunk rebalance cluster-data -action stop`
- `splunk rebalance cluster-data -action status`

REST Calls

```
curl -k -u admin:changeme https://master:mgmt/services/cluster/master/control/control/rebalance-data -d action=[start/stop/status]
```

```
curl -k -u admin:changeme https://master:mgmt/services/cluster/config/config -d rebalance_threshold=0.9
```

Data Rebalance UI

The screenshot shows the Splunk Data Rebalance UI for a Master Node. The top navigation bar includes links for Administrator, Messages (with 1 notification), Settings, Activity, Help, and Find. The main title is "Indexer Clustering: Master Node". A central message says "No Peers Configured" with a link to learn more. On the right, a context menu for "Node Type" is open, listing options: Master Node Configuration, Distribute Configuration Bundle, Data Rebalance (which is selected and highlighted in blue), and Disable Indexer Clustering. Below the menu is a table titled "Search Heads (1)". The table has two columns: "Search head name" and "Status". It lists one entry: "dlu-mbp.sv.splunk.com" with a status of "Up". A "filter" input field and a "10 per page" dropdown are also visible.

i	Search head name	Status
>	dlu-mbp.sv.splunk.com	✓ Up

About Support File a Bug Documentation Privacy Policy © 2005-2016 Splunk Inc. All rights reserved.

Data Rebalance UI

The screenshot shows the Splunk Data Rebalance UI. A modal dialog box is centered over the main interface. The dialog contains fields for 'Threshold' (set to 0.9), 'Max Runtime' (set to 'optional'), and 'Index' (set to 'optional'). Below these fields, a message states 'Data Rebalance is not running.' At the bottom right of the dialog is a green 'Start' button. The background of the main interface shows a warning message: 'Some Data is Not Searchable' with counts of 1 searchable and 0 not searchable peers. There are tabs for 'Peers (1)', 'Indexes (0)', and 'Search Help'. A 'filter' input field is also visible.

How Does It Work ?

.conf2016

splunk>

Data Rebalance Process

1. Cluster Master (CM) removes all excess bucket copies from the cluster
2. CM adds all buckets to a fix-up list. If the master is balancing just a single index, it adds only the buckets for that index to the fix-up list.
3. CM determines the peers below and above the average threshold
4. CM rebalances non-searchable buckets moving buckets from a peer with above average threshold, to nodes with lower than average threshold
5. CM rebalances searchable buckets
6. CM removes any excess copies, as a result of the rebalancing action and moves the processed bucket to the end of the fixup list.

Data Rebalance Process (Cont'd)

7. CM recalculates the over/under bucket averages for each peer in the cluster and continues rebalancing. Max number of buckets that are simultaneously rebalancing is subject to *max_peer_rep_load* and *max_peer_build_load*
8. Data rebalancing ends when the number of copies on all peers are within the range of average or if prematurely terminated.
9. CM performs primary rebalancing for all buckets in the cluster.
10. In a multi site cluster, buckets are first balanced across sites based on permitted site configuration and then balanced within each site.

Best Practices

- Run data rebalance after adding a new indexer(s)
- Run in maintenance window, to avoid potentially partial search results w/ data rebalance in progress
- Enable forwarder discovery, to minimize the need for data rebalance
- Set to desired '*rebalance_threshold*' (≤ 0.95) prior to kicking off data rebalance, through REST calls or *server.conf* changes.
- Configure the rebalancing load using *max_peer_rep_load* and *max_peer_build_load* in the [clustering] stanza of *server.conf*

Benefits And Limitations

.conf2016

splunk>

Benefits

- Reduced storage consumption per node, w/ even distribution to all nodes.
- Potential costs savings for cloud/on-prem customers to use instances with smaller storage footprint
- Newly added indexers, with resident primaries, immediately available for new search requests.
- Improved search performance by harnessing I/O throughput across all available indexers

Limitations

- **Optimal but not perfectly balanced**
 -
- **Based on number of buckets**
 -
- **Assumes similar total disk capacity per node**
 -
- **Concurrent searches**
 -

Demo

.conf2016

splunk>

Related Sessions

- TODO
- References to other Indexer Clustering sessions

Data Rebalance - Key Takeaways

Balanced Data Distribution

- Normalizes storage consumption across nodes
- Potential cost savings for with smaller per node storage footprint

Optimized Performance

- Distributes search traffic to newly added indexers
- Improved search performance by harnessing I/O throughput across all available indexers

Simplified Management

- Invoke using UI or REST calls
- Status/Progress indicator
- Set max time window for completion

Documentation

<http://docs.splunk.com/Documentation/Splunk/6.5.0/Indexer/Rebalancethecluster>



THANK YOU

.conf2016

splunk®