Splunk Metric Store Index Sizing

BSides SPL22

Oliver Hoppe

October 2022

Continuation of https://conf.splunk.com/watch/conf-online.html?search=PLA1627B#/

BSides SPL

AGENDA

- WHO AM I
- WHAT IS THE SPLUNK METRICS STORE
- CONFIGURATION OF A METRIC INDEX
- LOAD TESTING THE METRICS STORE FOR COMPRESSION
- PROBLEMS FOUND
- OVERCOMING THE PROBLEMS
- Q&A

POW!

HELLO!

OLIVER HOPPE
SPLUNK TRUST MEMBER
SPLUNK USER GROUP LEADER
LEAD ARCHITECT & DATA STRATEGIST @ LC SYSTEMS
FATHER OF 4

BSIDES SPL

WHAT IS THE SPLUNK METRICS STORE

- Released with 7.0 Sept 2017
- Dedícated type of index
- Support for statsD, collectD and other metric types
- Multi-metric support
- Conversion of logs to metrics at ingest time
- Special commands to search metric data e.g. mstats, mpreview, mcatalog

CONIFGURATION OF THE SPLUNK METRIC STORE

- via UI
- via indexes.conf (recommended)
  - datatype = metric
  - metric.timestampResolution = <s | ms>
  - metric.maxHotBuckets = auto (6)
  - metric.splitByIndexKeys
  - https://docs.splunk.com/Documentation/Splunk/9.0.1/admin/Indexesconf

LOAD TESTING THE METRIC STORE FOR COMPRESSION

- Setup with multiple senders generating metric data for 5000 hosts, 68 different multi metrics like cpu, disk, memory, netstat etc.
- Initial setup included all 68 metrics with a huge variety on values from 0.00 to millions, precission was always limited to 2 digits
- Later setup with reduced metrics to 46 just for memory and cpu
- Especially netstat metrics were removed with hundreds of different metrics within the netstat multi metric

LOAD TESTING THE METRIC STORE FOR COMPRESSION

- Findings 1st run
- After rolling hot to warm compression is between 300 and 1000%
- Buckets are far earlier rolled to warm and then to cold because of Strings.data / MaxMetaEntries and maxWarmDBCount

LOAD TESTING THE METRIC STORE FOR COMPRESSION

- Findings 2nd run
- After rolling hot to warm compression is between 95% and 150%%
- Buckets are far later rolled to warm and then to cold because of smaller Strings.data / MaxMetaEntries and therefore later hit of maxWarmDBCount

Problems found

- Buckets roll too early independent of maxDataSize because of Strings.data
- Compression is quite bad

OVERCOMING THE PROBLEMS

- Remove precission where it is not required
- Roll up the unit from e.g. bytes to kbytes or bigger where the precission is not required
- Do not send too many different metrics into the same index
- Split different types of metrics in different indexes
- Be careful with increasing MaxMetaEntries and it can massively impact your compute

1MIO METRIC POINTS =~20MBYTE

Plus possible replication

Q&A

- Further readings on the topic by Brett Adams & Chris Younger https://www.linkedin.com/pulse/perfecting-perfmon-splunk-brett-adams
- Join #metrics on the Splunk community Slack for a good discussion

Q&A

- Using dbsinpect to check your compression

```
| dbinspect index=metrics_a| eval sizeOnDiskMB=round(sizeOnDiskMB,0)| eval
rawSizeMB = round(rawSize/1024/1024,0)| eval compression=round((sizeOnDiskMB /
rawSizeMB) * 100,2)| rename index AS title
```

Q&A

- Using rest to check your index configs (from the monitoring console)

```
| rest splunk_server=ip-172-31-33-134.eu-central-1.compute.internal /services/data/indexes datatype=all  | join type=outer title [    | rest splunk_server=ip-172-31-33-134.eu-central-1.compute.internal /services/data/indexes-extended datatype=all  ]| eval warm_bucket_size = coalesce('bucket_dirs.home.warm_bucket_size', 'bucket_dirs.home.size')| eval cold_bucket_size = coalesce('bucket_dirs.cold.bucket_size', 'bucket_dirs.cold.size')| eval hot_bucket_size = if(isnotnull(cold_bucket_size), total_size - cold_bucket_size - warm_bucket_size, total_size - warm_bucket_size)| eval thawed_bucket_size = coalesce('bucket_dirs.thawed.bucket_size', 'bucket_dirs.thawed.size')| eval warm_bucket_size_gb = coalesce(round(warm_bucket_size / 1024, 2), 0.00)| eval hot_bucket_size_gb = coalesce(round(hot_bucket_size / 1024, 2), 0.00)| eval cold_bucket_size_gb = coalesce(round(cold_bucket_size / 1024, 2), 0.00)| eval thawed_bucket_size_gb = coalesce(round(thawed_bucket_size / 1024, 2), 0.00)| eval warm_bucket_count = coalesce('bucket_dirs.home.warm_bucket_count', 0)| eval hot_bucket_count = coalesce('bucket_dirs.home.hot_bucket_count', 0)| eval cold_bucket_count = coalesce('bucket_dirs.cold.bucket_count', 0)| eval thawed_bucket_count = coalesce('bucket_dirs.thawed.bucket_count', 0)| eval home_event_count = coalesce('bucket_dirs.home.event_count', 0)| eval cold_event_count = coalesce('bucket_dirs.cold.event_count', 0)| eval thawed_event_count = coalesce('bucket_dirs.thawed.event_count', 0)| eval home_bucket_size_gb = coalesce(round((warm_bucket_size + hot_bucket_size) / 1024, 2), 0.00)| eval homeBucketMaxSizeGB = coalesce(round('homePath.maxDataSizeMB' / 1024, 2), 0.00)| eval home_bucket_capacity_gb = if(homeBucketMaxSizeGB > 0, homeBucketMaxSizeGB, "unlimited")| eval home_bucket_usage_gb = home_bucket_size_gb." / ".home_bucket_capacity_gb| eval cold_bucket_capacity_gb = coalesce(round('coldPath.maxDataSizeMB' / 1024, 2), 0.00)| eval cold_bucket_capacity_gb = if(cold_bucket_capacity_gb > 0, cold_bucket_capacity_gb, "unlimited")| eval cold_bucket_usage_gb = cold_bucket_size_gb." / ".cold_bucket_capacity_gb| eval currentDBSizeGB = round(currentDBSizeMB / 1024, 2)| eval maxTotalDataSizeGB = if(maxTotalDataSizeMB > 0, round(maxTotalDataSizeMB / 1024, 2), "unlimited")| eval disk_usage_gb = currentDBSizeGB." / ".maxTotalDataSizeGB| eval currentTimePeriodDay = coalesce(round((now() - strptime(minTime,"%Y-%m-%dT%H:%M:%S%z")) / 86400, 0), 0)| eval frozenTimePeriodDay = coalesce(round(frozenTimePeriodInSecs / 86400, 0), 0)| eval frozenTimePeriodDay = if(frozenTimePeriodDay > 0, frozenTimePeriodDay, "unlimited")| eval freeze_period_viz_day = currentTimePeriodDay." / ".frozenTimePeriodDay| eval total_bucket_count = toString(coalesce(total_bucket_count, 0), "commas")| eval totalEventCount = toString(coalesce(totalEventCount, 0), "commas")| eval total_raw_size_gb = round(total_raw_size / 1024, 2)| eval avg_bucket_size_gb = round(currentDBSizeGB / total_bucket_count, 2)| eval compress_ratio = round(total_raw_size_gb / currentDBSizeGB, 2)." : 1" | search title=_metrics OR title=metrics_a OR title=metrics_ahv
```

THANKS!