splunk> .conf2017

How Did You Get So Big?

Tips & Tricks for Growing your Splunk Deployment from 50 GB/day to 1 TB/day

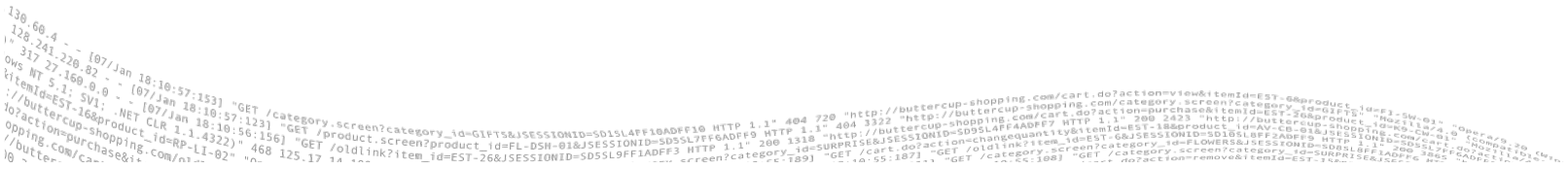
Gareth Anderson – IT Specialist

2017-08-19 | Washington, DC

Forward-Looking Statements

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.



splunk> .conf2017

What is this presentation about?

- After working on a Splunk instance for over four years there have been a variety of challenges encountered and many lessons learned
- This presentation aims to provide examples of what actions you can take to ensure your Splunk environment grows successfully
- At the time of writing Splunk 6.5.2 was in use with 6.6.2 in non-production

Agenda

- Indexer performance tuning and troubleshooting
  - Data Ingestion tuning
  - Data Distribution within an indexer cluster
  - Miscellaneous issues
- Search Head performance tuning
  - Optimising scheduled searches and dashboards
  - Tuning data models
- Universal Forwarder Issues

Indexer Performance/ Configuration Issues

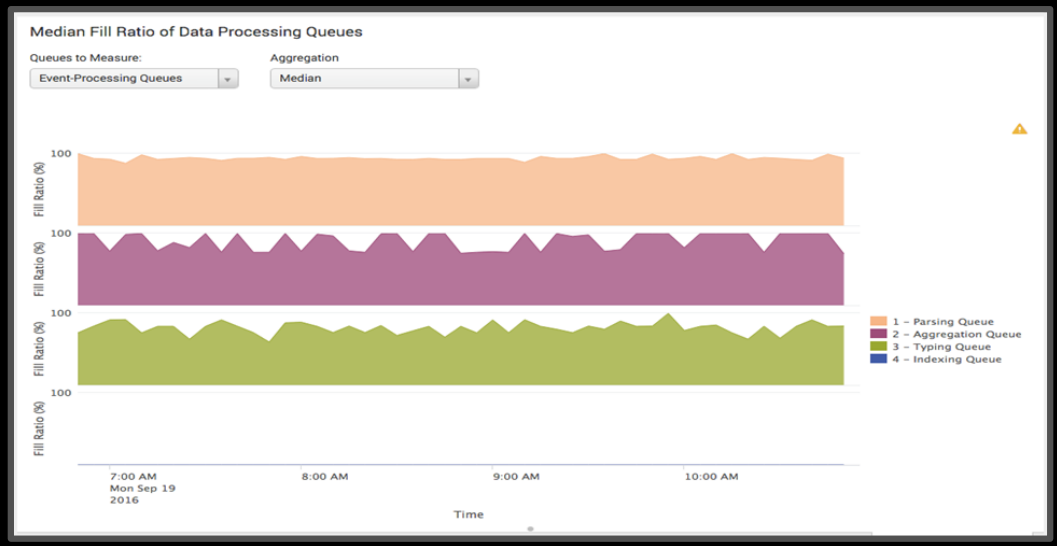
Part 1 - Parsing incoming data

splunk> .conf2017

Data ingestion – What can go wrong?

Indexer Data Processing Queues are blocked

- Data processing queues are in a pipeline
- For example a block in the typing queue (pictured) effects the aggregation queue and all queues above



Data ingestion Issues

- Aggregation queue blocked, the root cause was merging unusually large events (over 10K lines) using **SHOULD_LINEMERGE = true**
  - This issue took down all indexers in the cluster at the same time
  - Switching to a **LINE_BREAKER** resolved this issue
- Poorly written transforms can also cause blocking in the typing queue

Data ingestion - Timestamp Parsing

- Multiple timestamps in a single line can confuse Splunk
  - The **_time** field may be set in the future or in the past
  - XML events with whitespace & dates in the file will cause this issue
  - If the issue occurs the events may be split incorrectly
- Manually parsing timestamps (**TIME_PREFIX**, **TIME_FORMAT** in ***props.conf***) improves indexing performance

Data ingestion – Timestamp parsing

- What issues might occur?

- Timestamps parsed with a date in the future
- Timestamps parsed with a date in the past

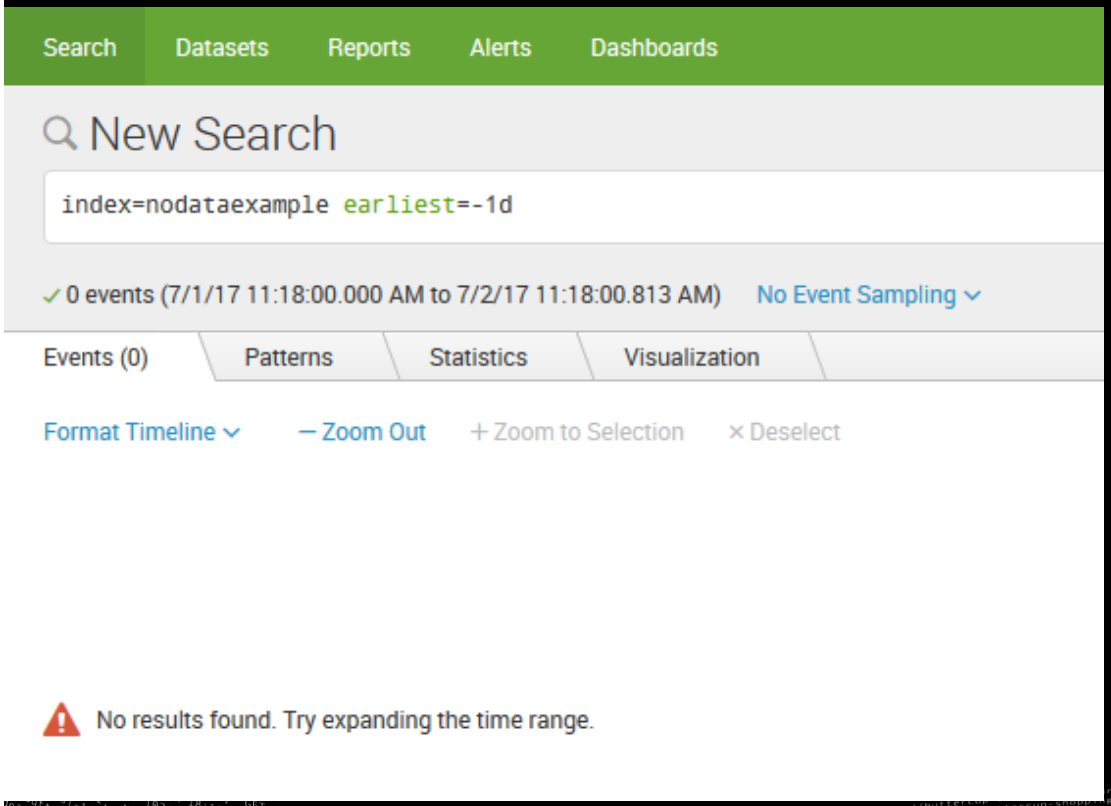
- Enable alerting to detect the issue, please refer to the appendix for an example

Indexer Performance/ Configuration Issues

Part 2 - Index sizing

splunk> .conf2017

Data ingestion – Index Sizing Issues



- Events rolled to frozen bucket too quickly

splunk> .conf2017

Data ingestion – Index Sizing Issues

- Due to not having **TIME_PREFIX**, **TIME_FORMAT** configured incoming data had a parsed timestamp of two years in the past
- Buckets rolled to frozen contained both recent data and data from what appeared to be “two years ago”
  - Some of the frozen buckets contained data from only one day ago, Splunk’s reputation was damaged within the company
- Recommendation – ensure timestamp parsing is working, avoid reaching the index size limit if possible

Data ingestion – Index Sizing Issues

- By default buckets will roll to frozen when the oldest timestamp of a bucket is past the **frozenTimePeriodInSecs** (default of 6 years)

- When an index reaches the size limit the behaviour is different, buckets will be frozen based on the oldest date of data in the bucket
  - It does not matter if the bucket contains both new and old data
  - This can become an issue if your data has timestamp parsing issues

Data ingestion – Index Sizing

- ***indexes.conf*** can be used to limit the size of the hot, cold and overall size per index and per volume

- The scenario of a flood of non-prod logging data has caused an index to roll data older than a few days to frozen
  - The root cause was an infinite loop in the application logging
  - Without appropriate index sizing and index separation we would have lost production data
  - Recommend that both the **frozenTimePeriodInSecs** setting is used in addition to setting a maximum index/volume size

Data ingestion – Index Configuration

- Consider creating subdirectories for sets of Splunk indexes (var/lib/splunk/oper/) for flexibility
- For example separating the security specific indexes from operational indexes
- When setting volume sizing keep in mind that the data model acceleration files use a different volume (**_splunk_summaries**) compared to the index hot/cold path
- Configure sizing within ***indexes.conf*** to handle the loss of an indexer

splunk> .conf2017

Indexer Performance/ Configuration Issues

Part 3 - Bucket sizing

splunk> .conf2017

Data ingestion – Bucket Tuning

- Splunk bucket sizes by default are set to auto (750MB per bucket) they can also use **auto_high_volume** which is 10GB per bucket on 64bit machines
- A rule of thumb is to have a bucket roll per day per indexer (**maxDataSize**), the **dbinspect** command is useful to determine the bucket sizing settings
- The number of buckets searched impacts search performance

Data ingestion – Bucket Tuning

- If the hot path is not large enough then…
  - This can result in the creation of many smaller buckets due to lack of disk space to store the number of hot buckets required (the default settings are 3 hot buckets and 750MB/bucket)
- If the timestamp parsing is not working correctly then…
  - Buckets can roll due to the difference in the timestamps of the incoming data and therefore you may have smaller buckets than expected

splunk> .conf2017

Indexer/ Heavy Forwarder

Miscellaneous Issues

splunk> .conf2017

Indexer/Heavy Forwarder

- Splunkweb can become unresponsive due to reaching the number of sockets/threads (dynamically configured)
  - A deployment server handling many incoming clients may require tuning
- Watch for high CPU utilisation on Splunk servers
  - We have alerts setup utilising the NMON for Splunk application
  - The Splunk introspection data or the Monitoring Console can be used

Indexer Search Optimisation

splunk> .conf2017

Search Optimisation

- Batch mode search parallelisation
  - The default value is 1
  - Changing the value to 2 will increase historical batch search performance at the cost of thread & memory consumption on the indexers

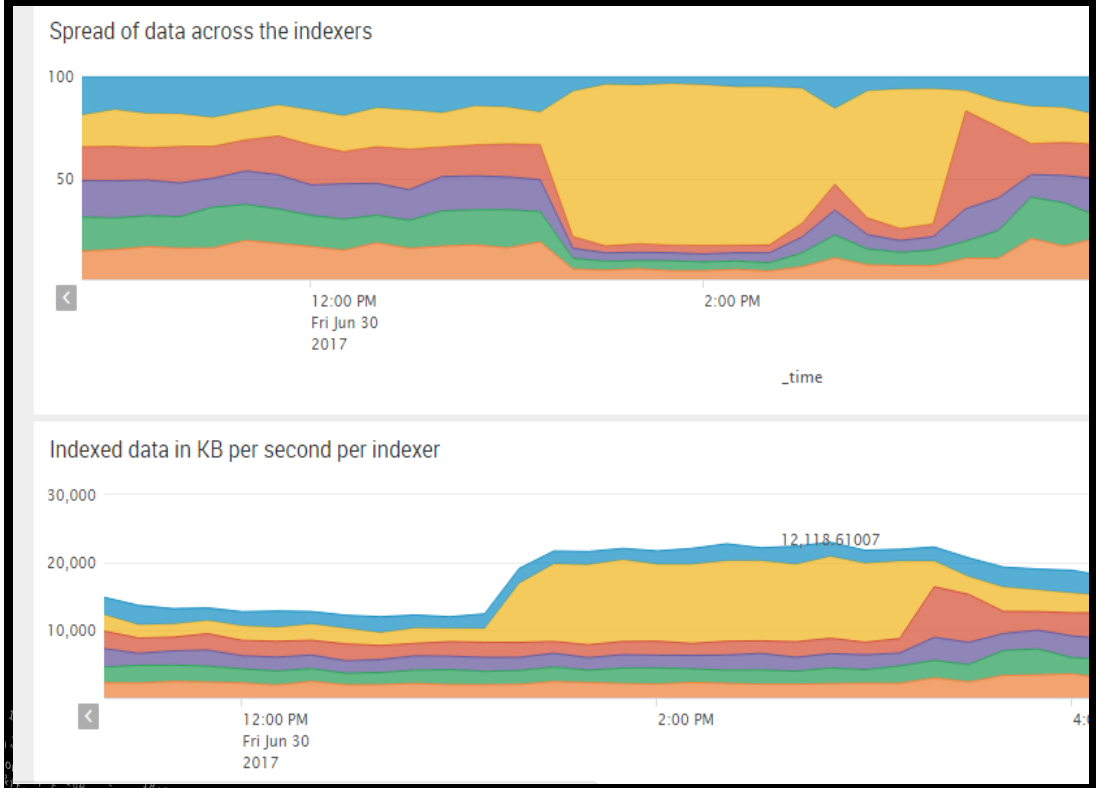
Indexer Cluster Issues

Data distribution within the cluster

splunk> .conf2017

Data Distribution Per Indexer

What can go wrong?



- In this example the spread of data over the cluster changes during the day…

splunk> .conf2017

Data Distribution Per Indexer

- An uneven data balance can cause more CPU & I/O usage on some indexers
- 80% of the network traffic on weekends got sent to 2 indexers in the 5 node indexer cluster
  - The data model acceleration searches resulted in 100% CPU utilisation on the indexers that received this extra volume of data.
  - The remaining 3 indexers in the 5 node cluster were relatively quiet (<50% CPU)

Data Distribution Per Indexer

- You should aim for an even data spread within the cluster
  - **autoLBFrequency** – attempt to divide data evenly over one or more minutes, for example 20 seconds across 6 indexers
    - Dropping the setting below 15 seconds can fill up your queues
  - In 6.5+ the **EVENT_BREAKER** setting (***props.conf***) will split large/rapidly growing files across the indexer cluster
  - In Splunk 6.6 there is a new feature in ***outputs.conf*** that allows load balancing of data by size (**autoLBVolume**)

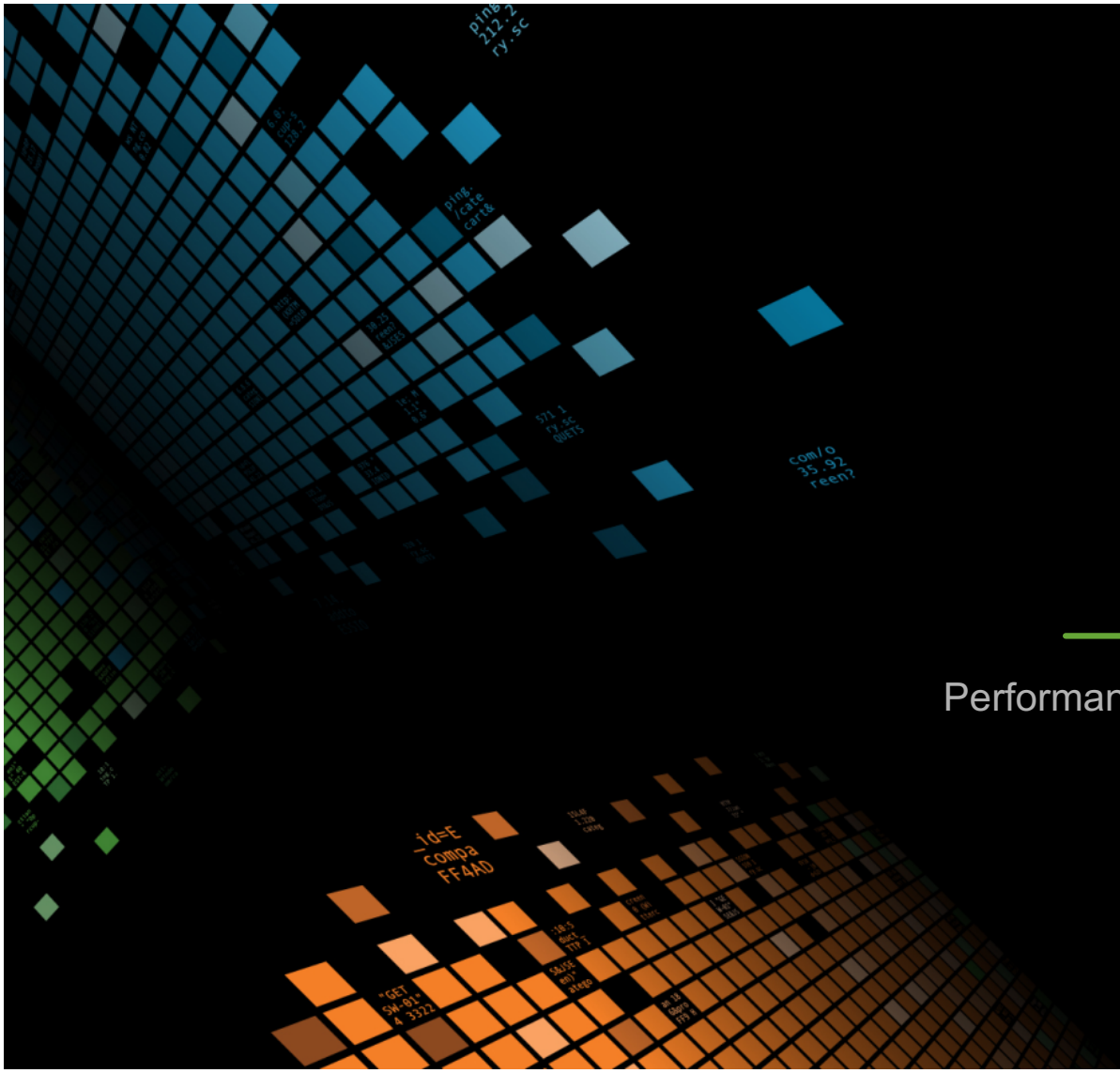
Data Distribution Per Indexer

- Consider dedicating a universal forwarder for ingesting large files
  - For files >2GB consider lowering the **maxKpbs**
  - This prevents a single indexer getting most of the data
  - Indexer queues will not be blocked if the ingestion is throttled

Data Distribution Per Indexer

- Consider increasing **parallelIngestionPipelines** on heavy forwarders to improve the data distribution
  - This allows the forwarder to talk to 2 indexers at the same time
- Use a **tstats** query to monitor the data distribution among indexer cluster members
  - | tstats count where index="*" by splunk_server, _time span=10m | timechart span=10m sum(count) by splunk_server

splunk> .conf2017



Search Heads

Performance issues relating to user behaviour

splunk> .conf2017

Search Head Tuning

- The default search time range in pre-6.6 versions of Splunk is “All time”, the ***ui-prefs.conf*** can be changed to “Today” to reduce the load on both search heads and indexers
- User roles can be limited in terms of search time window, run time, and the number of concurrent searches
- Roles can also be used to control the default indexes searched for a particular user role

Search Head Tuning – What to watch for?

- Scheduled searches should use specific indexes (do not use index=*)
- Limit search time period to prevent all time scheduled searches
- Consider configuring Max Historic Search Limits (user role configurations)

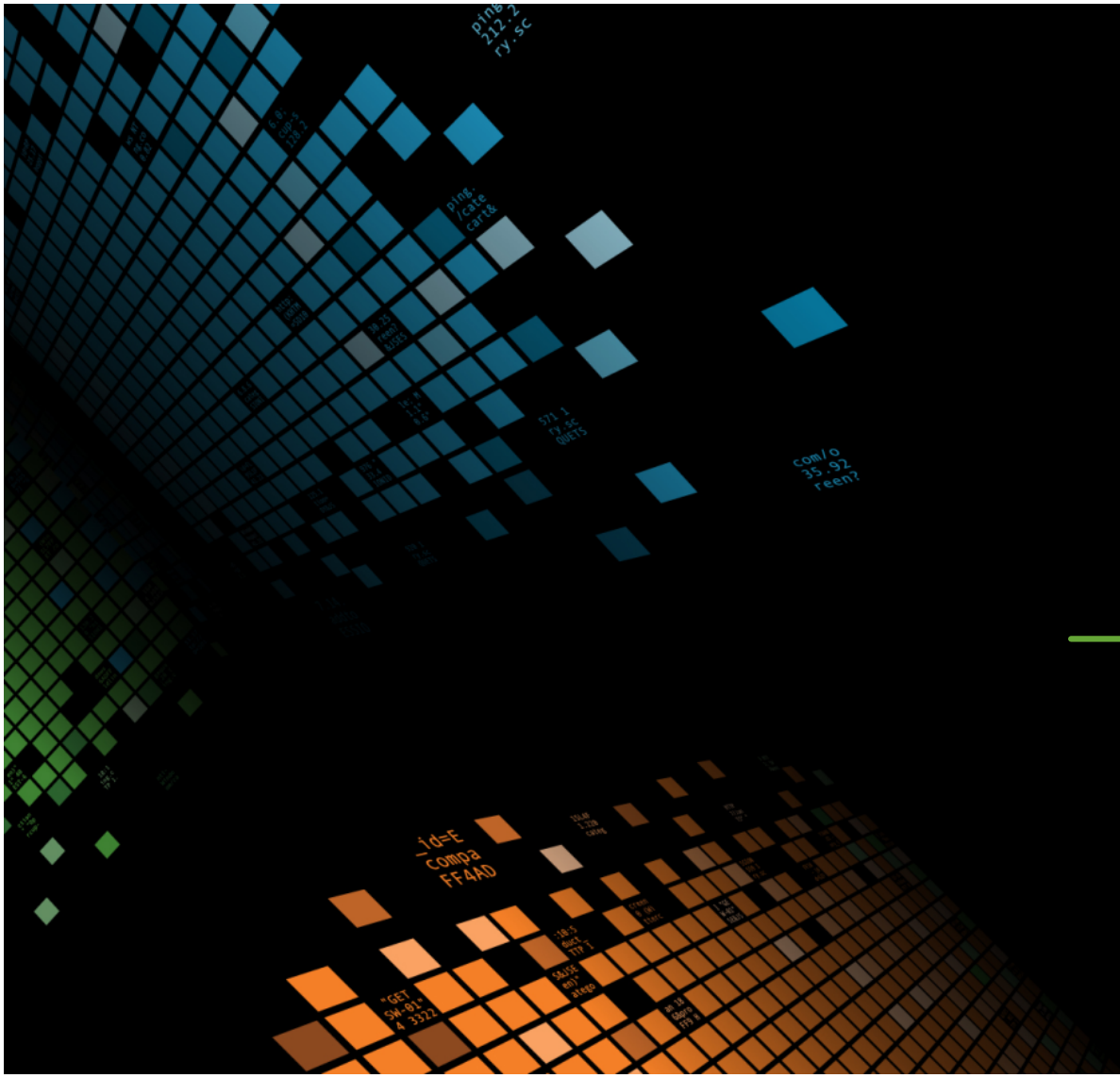
Search Head Tuning

- If allowing real time searches, consider indexed realtime mode (Splunk Enterprise Security does this by default)
  - Note that this particular setting applies to a user only if the setting is readable/exported to the user

- Review the **max_searches_per_cpu** setting in the ***limits.conf*** to ensure you are appropriately utilising your indexers/search heads (the default can require tuning)

Data Model Acceleration

- Data Model & Report Acceleration can create many CPU/IO intensive searches on the indexers
- Avoid using index=* in data models
  - Acceleration searches run every 5 minutes therefore narrowing down the indexes searched can make a measurable difference in performance
  - Shortening the data model acceleration period will also reduce both disk usage & CPU/IO on the indexers
  - If you do tweak the index settings, consider regular checks to see if any new indexes contain data relevant to the data model



Search Heads

Issues that will effect users

splunk> .conf2017

Detecting potential issues for users

- Scheduled searches that have become orphaned
  - When the owner of a report/alert gets deleted, the report/alert will stop running
- Consider opening permissions on private alerts (shared link)
- Scheduled searches not running due to a syntax error

Things to watch for

- Splunk scheduler skipping searches (or excessive scheduling delays)
- Splunk users violating the search quota
  - The user may not be aware of the messages appearing in Splunk
- Splunk users violating the disk quota
  - The user may be confused as to why their searches are no longer working

Universal Forwarders

Performance/CPU usage issues

- Splunk universal forwarders generally use more CPU when configured to monitor directories or file names with wildcards

- Recommend using the complete filenames and avoiding wildcards
  - In our testing we have found 2-3% CPU is normal for monitoring exact file names, while 20-30% CPU is not unusual for monitoring multiple directories with a lot of files getting created/deleted

Universal Forwarders

Troubleshooting the forwarder

- Learn how to use the btool command for troubleshooting
  - For example - **splunk btool inputs list --debug**
- Use the REST API endpoints to troubleshoot inputs/outputs
  - URL https://localhost:8089/services/admin/inputstatus/TailingProcessor:FileStatus
  - Or use the CLI command - **splunk list inputstatus**

Universal Forwarders

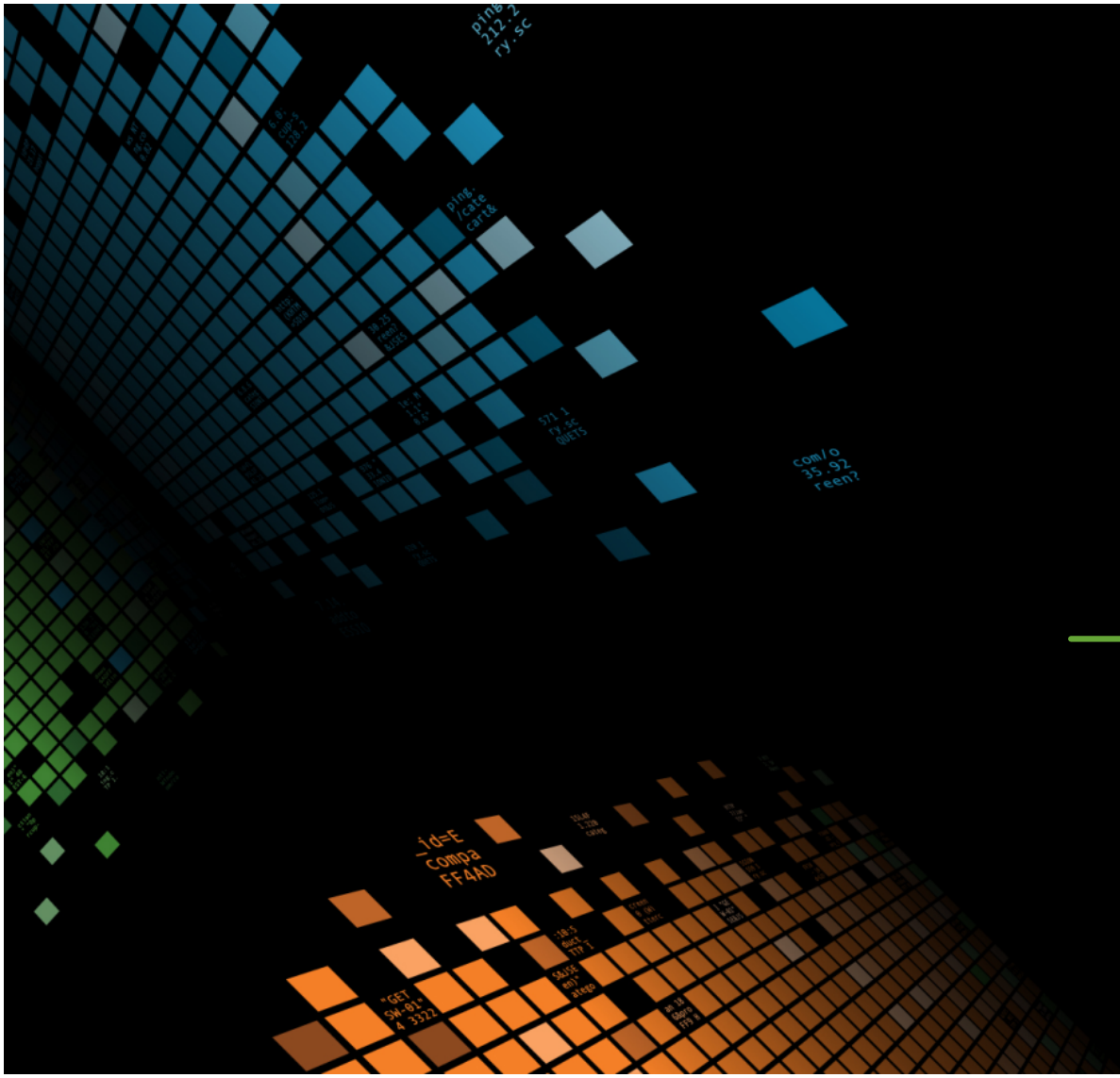
Troubleshooting the forwarder

- Impact of using **ignoreOlderThan** setting in the ***inputs.conf*** file, files are ignored until the next restart
  - Use this setting with great caution
- Watch the OS **ulimit** on all forwarders, 8192 file descriptors is a reasonable limit on most Linux/AIX forwarders
  - Having the ulimit set too low can result in the forwarder crashing

Universal Forwarders

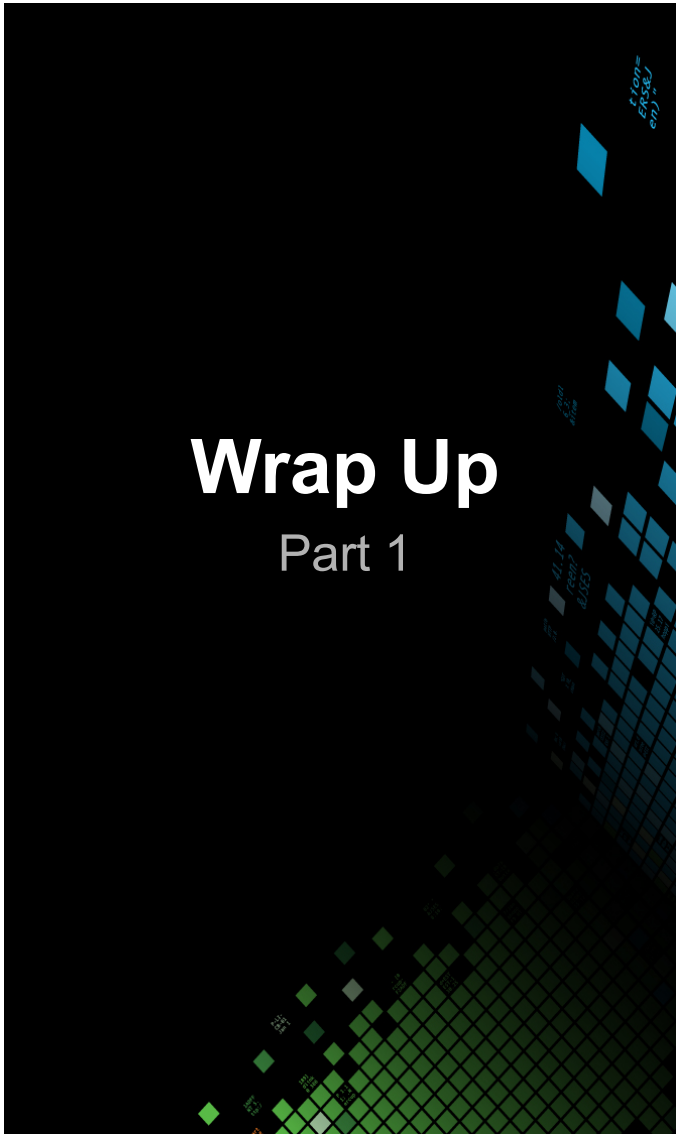
What to watch for

- Watch for excessive **checkCRC** errors, this can cause files to either not be read, or files to be re-read resulting in a duplication of data
- The ***inputs.conf*** setting of **initCrcLength** may need tweaking if the start of the monitored file has a lot of repetitive data (e.g. IBM WebSphere SystemOut)
- **crcSalt <SOURCE>** may be appropriate in some scenarios



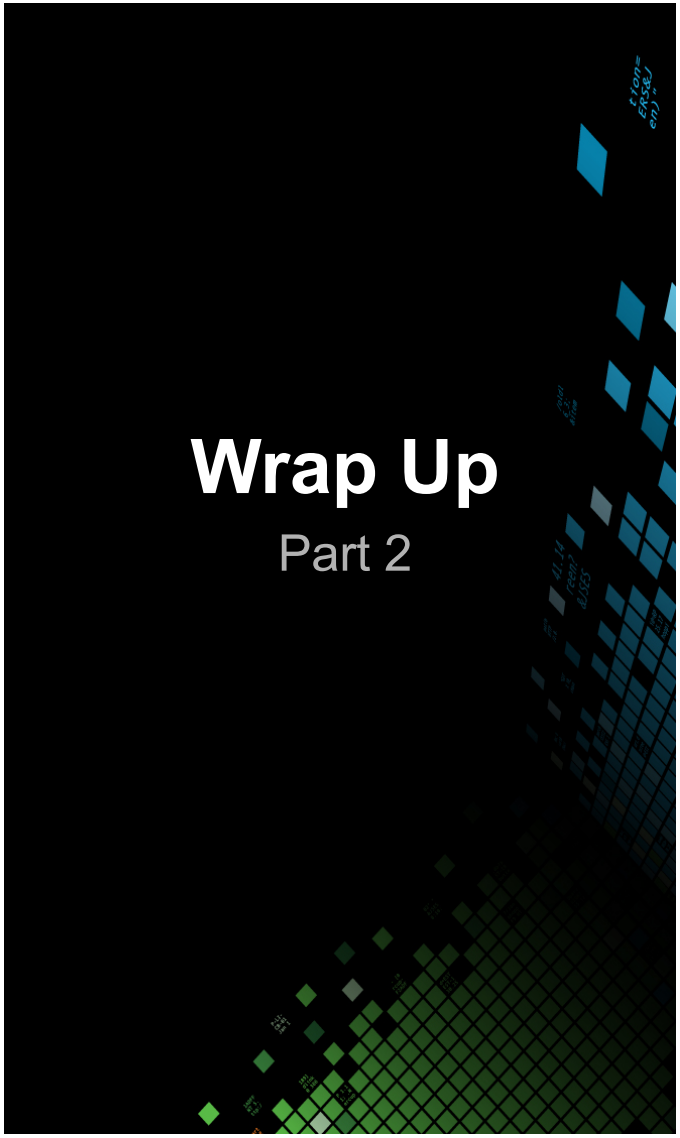
Wrap Up

splunk> .conf2017



1. Manually configure timestamp parsing (for both performance and correct timestamping)
2. Appropriately size indexes, buckets and volumes
3. Aim for even data balancing across all indexers within a cluster

splunk> .conf2017



Wrap Up

Part 2

4. Tune data model acceleration
5. Alert on poorly configured scheduled searches and dashboards
6. Alert on user level issues

splunk> .conf2017

The Splunk Admins application

Why would I want to download this?

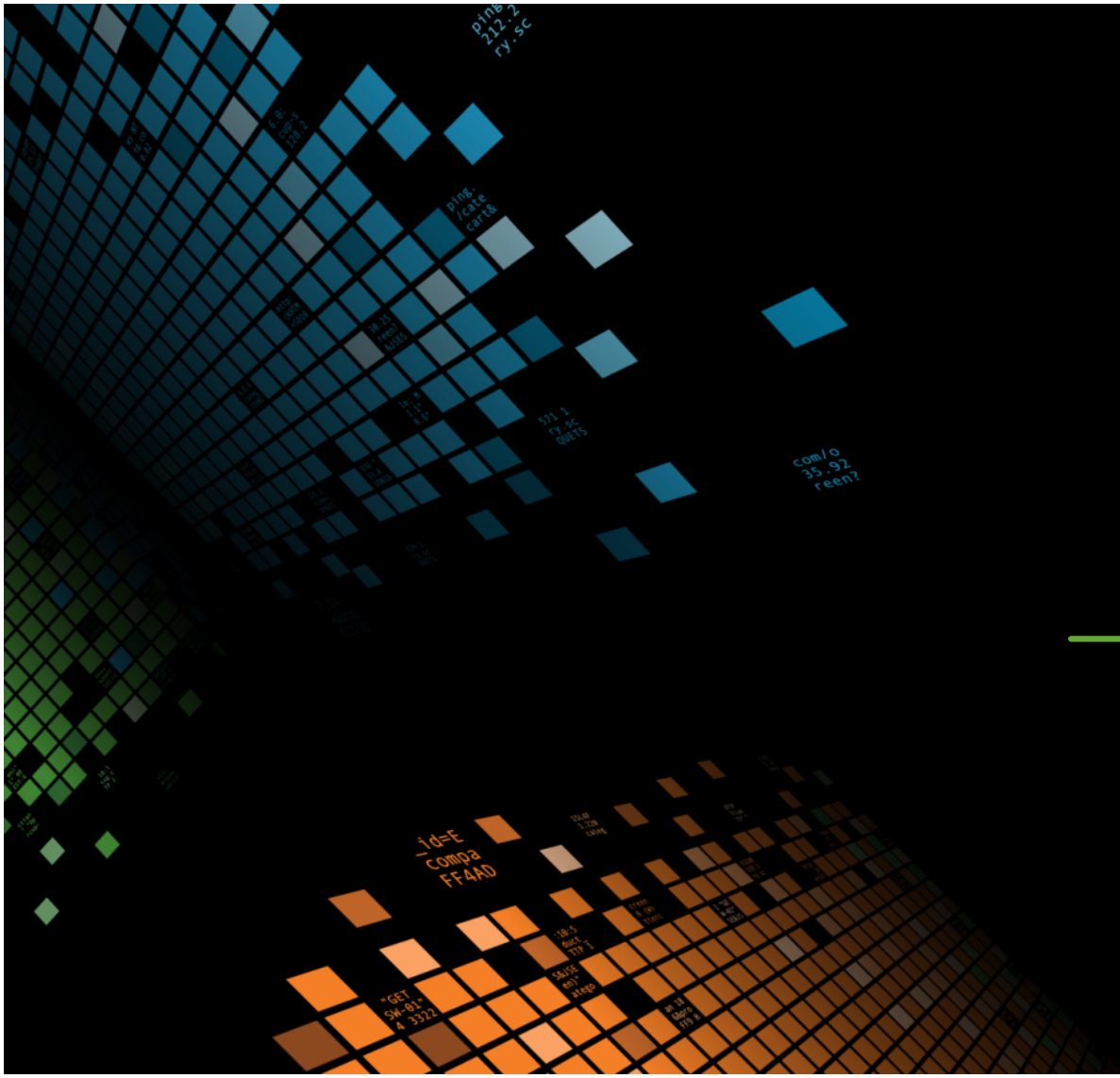
- All dashboards and alerts mentioned in this presentation can be found at:
- http://github.com/gjanders/SplunkAdmins

splunk> .conf2017

Thanks & QA

Don't forget to rate this session in the .conf2017 mobile app

splunk> .conf2017



Appendix

splunk> .conf2017

OS Level Tuning

splunk> .conf2017

OS Tuning

Linux Only

- What are the most important settings to watch?
- transparent huge pages
  - Should always be disabled on Splunk enterprise servers, this is optional for forwarders
- ulimits
  - Should be appropriately set within the limits.conf of the OS & this may be required within the init.d file as well, note that 8192 file descriptors is not enough for a busy indexer
  - The nproc setting (number of processes per user) becomes an issue in Redhat Linux (the default is 1024)

OS Tuning

Linux Only

- The tuned application
  - Adjusts various OS settings for your chosen tuning level, such as the swapiness level
- The IO Scheduler
  - Always choose the appropriate scheduler for your hardware
    - noop is likely the best fit for VMWare guests
    - deadline often works better with bare metal
    - Performance test Splunk if possible

Capacity Planning for Splunk

splunk> .conf2017

Capacity planning for Splunk

General Tips

- Bare metal hardware with locally attached storage is the fastest performance we have obtained for the indexing tier
- If using VMWare hardware do not share CPU/memory (or ideally storage)
  - CPU ready and co-stop times need to remain low on VMWare for Splunk to perform well
  - Cluster masters and deployers are good candidates for virtualisation, indexers are not
- Search heads can become a bottleneck if they have underlying storage or CPU issues

Capacity planning for Splunk

General Tips

- A Splunk environment utilising data model acceleration without fast I/O performance will reach full CPU utilisation quickly!
  - 100GB/day in our case on VMWare 32 VCPU/16 core machines
- Splunk indexer clusters can scale with faster I/O and more CPU (vertically) or with more indexers (horizontally)
- Refer to the Splunk capacity planning documentation for detailed information

Appendix

What does the application include?

- All mentioned alerts exist within the application available on https://github.com/gjanders/SplunkAdmins

- The application covers a large number of potential scenarios that can cause issues within a Splunk environment, the alerts are grouped into categories:
  - Issues that exist at the search head level only
  - Issues that exist at the indexer level only
  - Issues that exist at the forwarder level
  - Issues that may exist in any Splunk enterprise or non-enterprise instance
  - Issues specific to the deployment server

Appendix

What does the application include?

The application also includes a few dashboards

- Dashboard - Troubleshooting indexer CPU
  - This complex dashboard has multiple views of which searches are utilising the indexer CPU time and memory
- Dashboard - Indexer data spread
  - An attempt to visualise the distribution of incoming data and searchable data within an indexer cluster
- Dashboard - Scheduled Searches Distribution
  - A simple visualisation of the searches running per search head