Behind the Magnifying Glass: How Search Works

Jeff Champagne
Staff Architect, 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.
Who’s This Dude?

Jeff Champagne
jchampagne@splunk.com
Staff Architect

- Started with Splunk in the fall of 2014
- Former Splunk customer in the Financial Services Industry
- Lived previous lives as a Systems Administrator, Engineer, and Architect
- Loves Skiing, traveling, photography, and a good Sazerac
Am I In The Right Place?

Some familiarity with...

- Splunk Components
  - Search Head, Indexer, Forwarder
- Splunk Search Interface
- Search Processing Language (SPL)
What Will I Learn?

1. What is going on when you click search
2. How to improve searches so they run faster

- Splunk Architecture Overview
- How Splunk stores events
- Components of a search
- Search tips and SPL command alternatives
- Search command examples
Send data from thousands of servers using any combination of Splunk forwarders

Auto load-balanced forwarding to Splunk Indexers

Distributed Search coordinated by Splunk Search Head(s)
Index Vs. Index

An Overloaded Term

- **Logical grouping for data**
  - You or your Splunk admin create these
  - You reference these in your searches
    - Implicitly or explicitly

- **TSIDX File**
  - Time-series Index
  - Splunk’s “secret sauce”
  - A logical Index is made up of many indexes/TSIDX files
  - This is how we search for your data
    - More on this later...
How Are Events Stored?

Buckets, Indexes, and Indexers

Events \(\rightarrow\) Buckets \(\rightarrow\) Indexes (Logical Grouping) \(\rightarrow\) Indexers
How Are Events Stored?

Bucket Aging Process

Hot/Warm Storage
- Fast Storage
- Recent data

Cold Storage
- Slower “bulk” storage
- Older data

Archive Storage
- Historical/Compliance data
- Online (searchable)/Offline

Delete

-OR-
What’s In A Bucket?

- Bloom filter
- .tsidx
- journal.gz
What’s In A Bucket?

Journal.gz

- Your events go here
- Journal.gz is made up of many smaller compressed slices
- Raw data is collected and saved into slices
  - ~128KB of uncompressed data make up a slice
What’s In A Bucket?

TSIDX

<table>
<thead>
<tr>
<th>Raw Events</th>
<th>Lexicon</th>
<th>Values Array</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jim likes Mickey</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Suzie likes Donald</td>
<td>Donald: 1</td>
<td>Posting Value: 0, 1, 2</td>
</tr>
<tr>
<td>Pat likes Pluto</td>
<td>Jim: 0</td>
<td>Seek Address: 34, 87, 132</td>
</tr>
<tr>
<td></td>
<td>likes: 0, 1, 2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Mickey: 0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Pat: 2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Pluto: 2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Suzie: 1</td>
<td></td>
</tr>
</tbody>
</table>

Unique terms from the raw events are written to the lexicon. The postings list tells us where we can find a specific term in the values array. The seek address tells us where we can find the matching event(s) in the journal.gz slices.

*The overall structure of a TSIDX file has been simplified for illustrative purposes.
What’s In A Bucket?

Bloom Filter

- Determines whether a term is likely to exist in the TSIDX of a bucket
  - False positives are possible, false negatives are not

Lexicon

| Term | Donald | Jim | likes | Mickey |

Each term from the lex is run through a set of hashing algorithms

The output of each hash sets a bit in the array to ON

- Regardless of the # of terms, bit array size remains fixed
- Binary format
- Fast to read vs. TSIDX, which grows with more unique terms
How Search Works...

An Example
How Search Works

Components of a Search String

```
index=world name=waldo glasses=yes | eval miles=km*0.62 | stats count by countries
```

- **Base Search**
  - Retrieves & filters events

- **SPL Commands**
  - Evaluate, transform, and format events

Events are retrieved → Results move linearly through SPL commands
How Search Works

Where’s Waldo?

index=world name=waldo
How Search Works
Where’s Waldo?

1. Search

   index=world name=waldo

2. Hash the value waldo to create a bloom filter for our search
   010101010001001

3. Begin searching world buckets containing events from the Last 4 hours

4. Compare our filter to the one in each bucket

5. Locate the value waldo in the TSIDX

6. Retrieve events with waldo using the seek address in the TSIDX

I have been trying to find Waldo looking all over these books. I'm not sure I'll

The individual you are looking for does not exist in this dataset. We banished him. He

Oh yeah, Waldo comes in this joint all the time. The last time I saw him was probably

*The internal structure of Bloom filters, TSIDX, and Journal files has been simplified for illustrative purposes*
How Search Works...
Distributed Search
How Search Works

Distributed Search

1. Search Head parses search into distributed and centralized parts

2. Distributed parts of search are sent to indexers

3. Indexers fetch events from disk

4. Schema is applied to events (Schema-on-the-fly)

5. Events are filtered based on KV pairs

6. Distributed commands are applied

7. Results are sent to Search Head

8. Search Head applies centralized streaming & transforming commands, then displays results
How Search Works

Types of Search Commands

● **Streaming Commands**
  - Distributable (Remote Streaming)
    - Operate on individual events
    - Run on indexers (distributed)
    - Ex: eval, rex, where, rename, fields…
  - Centralized (Stateful Streaming)
    - Operate on at least a sub-set of the entire result set
    - Run on Search Head (centralized)
    - Ex: head, streamstats

● **Transforming Commands**
  - Create a reporting data structure
  - Operate on the entire event set
    - Non-streaming
    - Typically run on the search head
  - Ex: transaction, stats, top, timechart…
How Search Works

Command Ordering

- Events are retrieved
- Results move linearly through SPL commands

- Commands are processed in the order you write them
- Placing centralized or transforming commands before distributable commands may force unnecessary data and/or processing to the Search Head
Want To Know More?

**Search: Under the Hood** by Chris Pride
– Wednesday, Sept. 28<sup>th</sup> 4:35PM – 5:20PM
Search Tips
Just Because You Can...doesn’t Mean You Should

Plan your search to leverage the power of Splunk!
Search Tips

- Reduce the amount of data Splunk has to Search
  - Specify and limit the index(es)
  - Limit the time range
  - Search for values that are unique to your events where possible
    ‣ Reduce the number of events filtered after schema-on-the-fly

- Distributed Search
  - Ensure events are well distributed
  - Place distributed commands before centralized commands
Thou shalt not use index=* or All Time

-Moses
# Search Tips

<table>
<thead>
<tr>
<th>Avoid</th>
<th>Explanation</th>
<th>Suggested Alternative</th>
</tr>
</thead>
<tbody>
<tr>
<td>All Time</td>
<td>• Events are grouped by time&lt;br&gt;• Reduce searched buckets by being specific about time</td>
<td>• Use a specific time range&lt;br&gt;• Narrow the time range as much as possible</td>
</tr>
<tr>
<td>index=*/</td>
<td>• Events are grouped into indexes&lt;br&gt;• Reduce searched buckets by specifying an index</td>
<td>• Always specify an index in your search</td>
</tr>
<tr>
<td>Wildcards</td>
<td>• Wildcards are not compatible with Bloom Filters&lt;br&gt;• Wildcard matching of terms in the index takes time</td>
<td>• Varying levels of suck-itude&lt;br&gt;  &gt; myterm* → Not great&lt;br&gt;  &gt; *myterm → Bad&lt;br&gt;  &gt; <em>myterm</em> → Death&lt;br&gt;• Use the OR operator i.e.: MyTerm1 OR MyTerm2</td>
</tr>
</tbody>
</table>
# Search Tips

<table>
<thead>
<tr>
<th>Avoid</th>
<th>Explanation</th>
<th>Suggested Alternative</th>
</tr>
</thead>
</table>
| **NOT !=** | • Bloom filters & indexes are designed to quickly locate terms that exist  
      • Searching for terms that don’t exist takes longer | • Use the OR/AND operators  
           (host=c OR host=d)  
           (host=f AND host=h)  
           vs.  
           (host!=a host!=b)  
           NOT host=a host=b |
| **Verbose Search Mode** | • Verbose search mode causes full event data to be sent to the search head, even if it isn’t needed | • Use Smart Mode or Fast Mode |
| **Real-time Searches** | • RT Searches put an increased load on search head and indexers  
      • The same effect can typically be accomplished with a 1 min. or 5 min. scheduled search | • Use a scheduled search that occurs more frequently  
      • Use Indexed-Realtime searches  
      (Set by Splunk admin) |
## Search Tips

<table>
<thead>
<tr>
<th>Avoid</th>
<th>Explanation</th>
<th>Suggested Alternative</th>
</tr>
</thead>
</table>
| Transaction    | • Not distributed to indexers  
• Typically only needed if using additional parameters (maxSpan, startsWith, etc...) | • Use the stats command to link events where possible             |
| Joins/Sub-searches | • Joins can be used to link events by a common field value, but this is an intensive search command | • Use the stats (preferred) or transaction command to link events |
| Search after first | • Filtering search results using a second “| search” command in your query is inefficient | • As much as possible, add all filtering criteria before the first  
  i.e.: >index=main foo bar vs.  
  >index=main foo | search bar |
The TERM Directive

Why does it matter?

• Splunk breaks terms by Major and Minor Segmenters
  - When writing to the TSIDX and searching
  - Default minor segmenters:
    / : = @ . – $ # % \ _

• TERM prevents breaking on Minor segmenters

[ AND 0 10 6 index::myindex ]
The TERM Directive

What about quotes?

- TERM controls how we search the lexicon and which events are retrieved from disk
- Quotes can help filter after the events are retrieved from disk
- Use quotes when the value in your key-value pair has major breakers

```
[ AND wonka willy index::myindex ]
```
The TERM Directive

How do I use it?

- Your term **MUST** be bounded by major segmenters
  - Example: Spaces, tabs, carriage returns
    - See Segmenters.conf spec for full details
  - Your term cannot contain major segmenters

<table>
<thead>
<tr>
<th>Term Definition</th>
<th>Example Event Log</th>
</tr>
</thead>
<tbody>
<tr>
<td>ip=TERM(10.0.0.6)</td>
<td>ip 10.0.0.6 - 807256800 GET /images/launchlogo.gif</td>
</tr>
<tr>
<td>TERM(ip=10.0.0.6)</td>
<td>ip=10.0.0.6 - 807256804 GET /shuttle/missions.html</td>
</tr>
<tr>
<td>TERM(ip10.0.0.6)</td>
<td>ip10.0.0.6 - 807256944 GET /history/history.html</td>
</tr>
<tr>
<td>TERM(10.0.0.6*)</td>
<td>10.0.0.6:80 - 807256966 GET /skylab/skylab-4.html</td>
</tr>
<tr>
<td>TERM(“Willy Wonka”)</td>
<td>9/28/16 1:30 PM - name=Willy Wonka sex=m age=46</td>
</tr>
</tbody>
</table>
Search Tips

Indexed Extractions

- Special Key-Value pairs that are stored in the TSIDX file
- Default Extractions
  - source, host, sourcetype
  - Use these whenever possible
- TSTATS
  - Super-fast command
  - Doesn’t search or return raw data
  - Can be used on report/data model accelerations AND indexed extractions
Want To Know More?

How to Scale: From _raw to tstats by David Veuve
   – Wednesday, Sept 28th 2:15 PM – 3:00 PM

Previous Session:

• Fields, Indexed Tokens and You by Martin Müller
Commands In Action
Command Abuse
Fields vs. Table

**Goal:** Remove fields I don’t need from results

**BAD:**
```
index=myIndex field1=value1 | table field1, field2, field4 | head 10000
| table field2, field4
```

**GOOD:**
```
index=myIndex field1=value1 | fields field1, field2, field4 | head 10000
| table field2, field4
```

- Table is a formatting command NOT a filtering command
  - If used improperly, it will cause unnecessary data to be transferred to the search head from search peers
- Fields tells Splunk to explicitly drop or retain fields from your results
Command Abuse
Fields vs. Table Example

<table>
<thead>
<tr>
<th>Search Term</th>
<th>Status</th>
<th>Artifact Size</th>
<th># of Events</th>
<th>Run Time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>table</td>
<td>Running (1%)</td>
<td>624.93MB</td>
<td>2,037,500</td>
</tr>
<tr>
<td></td>
<td>fields</td>
<td>Done</td>
<td>9.95MB</td>
<td>10,000</td>
</tr>
</tbody>
</table>
Goal: Group multiple events by a common field value

• If you’re not using any of the Transaction command parameters, the same results can usually be accomplished using Stats
  – startswith, endswith, maxspan, maxpause, etc…
Goal: Return the latest JSESSIONID across two sourcetypes

GOOD:
```
sourcetype=access_combined OR sourcetype=applogs | stats latest(*) AS * BY JSESSIONID
```

NOT GREAT:
```
sourcetype=access_combined | join type=inner JSESSIONID [search sourcetype=applogs | dedup JSESSIONID | table JSESSIONID, clienip, othervalue]
```
Resources

• Splunk Docs
  – Write Better Searches
    http://docs.splunk.com/Documentation/Splunk/latest/Search/Writebettersearches
  – Wiki: How Distributed Search Works
    http://wiki.splunk.com/Community:HowDistSearchWorks
  – Splunk Search Types
    http://docs.splunk.com/Documentation/Splunk/6.2.3/Capacity/HowsearchtypesaffectSplunkEnterpriseperformance
  – Blog: When to use Transaction and when to use Stats
  – Segmenters.conf Spec
    http://docs.splunk.com/Documentation/Splunk/latest/Admin/Segmentersconf
  – Splunk Book: Exploring Splunk

• How Bloom Filters Work: An Interactive Demo
  https://www.jasondavies.com/bloomfilter/
Resources

Training

• eLearning
  – What is Splunk (Intro to Splunk)
    ‣ http://www.splunk.com/view/SP-CAAAH9U

• Instructor Led Courses with Labs
  – Using Splunk
    ‣ http://www.splunk.com/view/SP-CAAAH9A
  – Searching & Reporting with Splunk
    ‣ http://www.splunk.com/view/SP-CAAAH9C
  – Advanced Searching & Reporting
    ‣ http://www.splunk.com/view/SP-CAAAH9D
What Now?

Related breakout sessions and activities...

- **How to Scale: From _raw to tstats** by David Veuve
  - Wednesday, Sept 28th 2:15 PM – 3:00 PM

- **Search: Under the Hood** by Chris Pride
  - Wednesday, Sept. 28th 4:35PM – 5:20PM

- **Best Practices and Better Practices for Users** by Burch Simon
  - Thursday, Sept. 29th 12:25PM – 1:10PM

Previous Sessions...

- **Fields, Indexed Tokens and You** by Martin Müller
- **Worst Practices...and How to Fix Them** by Jeff Champagne
- **Best and Better Practices for Admins** by Burch Simon
- **Observations and Recommendations on Splunk Performance**
  by Dritan Bitincka
Questions?
THANK YOU