How to Troubleshoot Blocked Ingestion Pipeline Queues with Indexers and Forwarders

Harendra Rawat
Senior Principal Software Engineer | Splunk Inc.
During the course of this presentation, we may make forward-looking statements regarding future events or plans 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 may differ materially. 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, it may not contain current or accurate information. We do not assume any obligation to update any forward-looking statements made herein.

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 functionalities described or to include any such feature or functionality in a future release.

Splunk, Splunk>, Turn Data Into Doing, The Engine for Machine Data, Splunk Cloud, Splunk Light and SPL are trademarks and registered trademarks of Splunk Inc. in the United States and other countries. All other brand names, product names, or trademarks belong to their respective owners. © 2019 Splunk Inc. All rights reserved.
How to troubleshoot blocked ingestion pipeline queues with Indexers and Forwarders

Troubleshoot blocked ingestion pipeline queues
Introduction to Splunk ingestion pipeline queues

Typical problems associated with queues
- Blocked queues, low ingestion throughput and broken/orphaned events

How to find problematic queue

Debugging problematic queue

How to fix problematic queue
Splunk Ingestion Pipeline Queues

Forwarder pipeline queues
Splunk Ingestion Pipeline Queues

Queue

- Queue size bounded by memory
- Holds variable sized Pipeline Data
Forwarder Ingestion Pipeline

Forwarder ingestion pipeline queues

- Remember the order of queues and processor in the pipeline.
Indexer Ingestion Pipeline

Indexer pipeline queues
Indexer Ingestion Pipeline

Pipeline Processor threads

Remember the order of queues and processor in the pipeline.
Typical Problems Associated with Queues

Problems with ingestion pipeline queues
Typical Problems Associated with Queues

Problems with queues

Blocked queue

Low ingestion throughput

Missing Broken/Orphaned Events
How to Find Problematic Queue

Identifying the queue causing blocked ingestion pipeline
How to Find Problematic Queue With
Identifying the queue responsible for blocked ingestion pipeline

- Click on >Monitoring Console (MC)

- Remember MC.
How to Find Problematic Queue
Identifying the queue responsible for blocked ingestion pipeline

- Click on >(MC)Indexing->Performance->Indexing Performance: Instance.

- Remember the order of queues in the pipeline
- Parsing/Aggregation queues are blocked due to Typing queue
- Typing Queue is the bottleneck
How to Find Problematic Queue
Identifying the queue responsible for blocked ingestion pipeline

- Click on > (MC) Indexing -> Performance -> Indexing Performance : Advanced.

- In limits.conf add [default]
  regex_cpu_profiling = true

- Restart splunk

- How to enable identify CPU usage using MC
- How to enable regex cpu profiling
Debugging Blocked Queue

Troubleshooting typing queue

- Syslog responsible for blocked queue
- Due to high volume syslog

• Take away> Use MC to find offending source/sourcetype causing blocked typing queue.
How to Find Problematic Queue

Identify the queue responsible for blocked ingestion pipeline
How to Find Problematic Queue

Identifying the queue responsible for blocked ingestion pipeline

Using “grep” cli command

>grep blocked metrics.log

- Parsing/Aggregation queues are blocked due to Typing queue.
- Typing Queue is the bottleneck
- Using ‘grep’ cli find blocked queues.
### How to Find Problematic Queue

Identifying the queue responsible for blocked ingestion pipeline

- Remember the order of queues in the pipeline.
- Parsing/Aggregation queues are blocked due to Typing queue.
- Using `grep` cli find blocked queues for a specific time range. Find the queues that are not blocked.

```bash
> grep '02-28-2017 23:44:20' metrics.log| grep ingest_pipe=1|grep group=queue
```

<table>
<thead>
<tr>
<th>Time</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>02-28-2017 23:42:46.899 +0000 INFO</td>
<td>Metrics - group=queue, ingest_pipe=1, name=indexqueue, max_size_kb=500, current_size_kb=0</td>
</tr>
<tr>
<td>02-28-2017 23:42:46.899 +0000 INFO</td>
<td>Metrics - group=queue, ingest_pipe=1, name=parsingqueue, max_size_kb=6144, current_size_kb=4353, current_size=0, largest_size=715, smallest_size=252</td>
</tr>
<tr>
<td>02-28-2017 23:42:46.899 +0000 INFO</td>
<td>Metrics - group=queue, ingest_pipe=1, name=splunktopin, max_size_kb=500, current_size_kb=476, current_size=37, largest_size=112, smallest_size=0</td>
</tr>
<tr>
<td>02-28-2017 23:42:46.899 +0000 INFO</td>
<td>Metrics - group=queue, ingest_pipe=1, name=typingqueue, blocked=true, max_size_kb=500, current_size_kb=499, current_size=499, largest_size=1467, smallest_size=0</td>
</tr>
<tr>
<td>02-28-2017 23:44:20.893 +0000 INFO</td>
<td>Metrics - group=queue, ingest_pipe=1, name=indexqueue, max_size_kb=1024, current_size_kb=1023, current_size=2/66, largest_size=2497, smallest_size=0</td>
</tr>
<tr>
<td>02-28-2017 23:44:20.893 +0000 INFO</td>
<td>Metrics - group=queue, ingest_pipe=1, name=parsingqueue, max_size_kb=1024, current_size_kb=1023, current_size=2610, largest_size=3025, smallest_size=105</td>
</tr>
<tr>
<td>02-28-2017 23:44:20.893 +0000 INFO</td>
<td>Metrics - group=queue, ingest_pipe=1, name=splunktopin, max_size_kb=500, current_size_kb=458, current_size=28, largest_size=105, smallest_size=0</td>
</tr>
<tr>
<td>02-28-2017 23:44:20.893 +0000 INFO</td>
<td>Metrics - group=queue, ingest_pipe=1, name=typingqueue, max_size_kb=500, current_size_kb=477, current_size=1202, largest_size=1515, smallest_size=0</td>
</tr>
</tbody>
</table>
Finding processor causing blocked queue

Troubleshooting typing queue

- Remember the processors of typing queue.
- Search cpu usage (cpu_seconds) in metrics.log for name=typing.
- Typing Queue is blocked due to regexreplacement.
- Using `grep name=typing metrics.log |grep "02-28-2017 23:44:20"` find all processors associated with typing queue.
- Find the processor consuming most of the cpu seconds.
Finding sourcetype causing blocked queue

Using regexreplacement processor cpu profiling

- Breakdown total time spent by regexreplacement processor.
- Search ‘per_sourcetype_regex_cpu’ in metrics.log
- ‘syslog’ is the reason typing Queue is blocked.

- In limits.conf add following and Restart splunk
  ```conf
  [default]
  regex_cpu_profiling = true
  ```

- Using ‘grep group=per_sourcetype_regex_cpu metrics.log |grep "02-28-2017 23:44:20”’
  find total regex cpu usage for each sourcetype for a given time range.
- Find the sourcetype consuming most of the cpu in milliseconds.
TcputQ to TcpinQ
Troubleshooting Blocked Tcput queue

- netstat
- ping
- metrics.log

- Tools/log possibly needed to narrow down the cause of TcputQ blockage
Check TcpoutQ status on Monitoring Console
Troubleshooting Blocked Tcpout queue

- Click on >(MC)Indexing->Performance->Indexing Performance : Instance.

- Monitoring console indicates TcpoutQ is blocked.
- There is missing metrics.log data due to indexing latency caused by blocked TcpoutQ
Live troubleshooting Tcpout queue
With metrics.log/netstat/sysctl/ping

 tcpout queue fluctuating.
 tcp send buffer is growing.
 tcp layer buffers are low and might be an issue.

> sysctl –an | grep wmem
Investigate TcpinQ concurrently.

> grep name=tcpout_metrics.log

 Tcp send-Q growing
tcp 0 1304
tcp 0 418827

Low tcp buffer settings

netstat –an | grep “:9997”

bash-4.2$ netstat -a | grep "9997"
tcp 0 147.107.156.15:47558 169.122.212.55:9997 TIME_WAIT
tcp 0 147.107.156.15:60044 169.122.220.28:9997 ESTABLISHED

bash-4.2$ netstat -na | grep ":9997"
tcp 0 147.107.156.15:48094 169.122.212.55:9997 TIME_WAIT

Live troubleshooting Tcpin queue

With metrics.log/netstat

- tcp recv buffer is empty.
- Splunk tcpin queue is fine as well.
- Tune net.core.wmem_max setting on forwarder. (see previous slide)

```
netstat -an | grep "::9997"
```

Tcp recv-Q is empty on indexer

Splunk tcpin queue is empty on indexer

```
grep name=tcpin_ metrics.log
```

current_size_kb=0, current_size=0
Debugging blocked tcpout queue

Troubleshoot blocked ingestion pipeline queues
Live troubleshooting Tcpout queue

Troubleshooting Blocked Tcpout queue

- tcpout queue is full.
- netstat -an | grep :9997
- tcp layer send buffer is also full.

If TcpoutputQ is blocked, it’s very likely tcp send-Q is also full (netstat output)
Live troubleshooting Tcpin queue
Troubleshooting Blocked Tcpout queue

grep tcpin metrics.log | grep group=queue

grep name=pipelineinputchannel metrics.log

current_size_kb=0, current_size=0

tcp 675170 0
tcp 576132 0
tcp 895074 0

tcpin queue is empty.
tcp layer receive buffer is full.
TcpInputProcessor is busy.
Tune autoLBFrequency on forwarder to reduce channels.

new_channels=60290, removed_channels=62712
Debugging blocked queue/missing broken events

Troubleshoot blocked ingestion pipeline queues
Debugging blocked queue

Troubleshooting Blocked Tcpout queue/Orphaned Broken events

- Connection goes away
- Teared down instance
- UF crashed

Universal Forwarder (useACK=true)
Load Balancing

Broken connection results in Broken orphaned missing pData stuck in TcpoutQ.
Detect broken missing events

Troubleshooting Blocked Tcpout queue/Orphaned Broken events

Query to detect broken events stuck in tcpoutQ

```
index_internal source=splunkd.log "Possible duplication of events with channel"
```

Solution - use Heavy Forwarder (useACK=true) as intermediate forwarder (Load Balancing)
Thank You!

Go to the .conf19 mobile app to RATE THIS SESSION