splunk> .conf2017

Scaling Indexer Clustering

5 Million Unique Buckets and Beyond

Cher-Hung Chang | Principal Software Engineer

Tameem Anwar | Software Engineer

09/26/2017 | 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

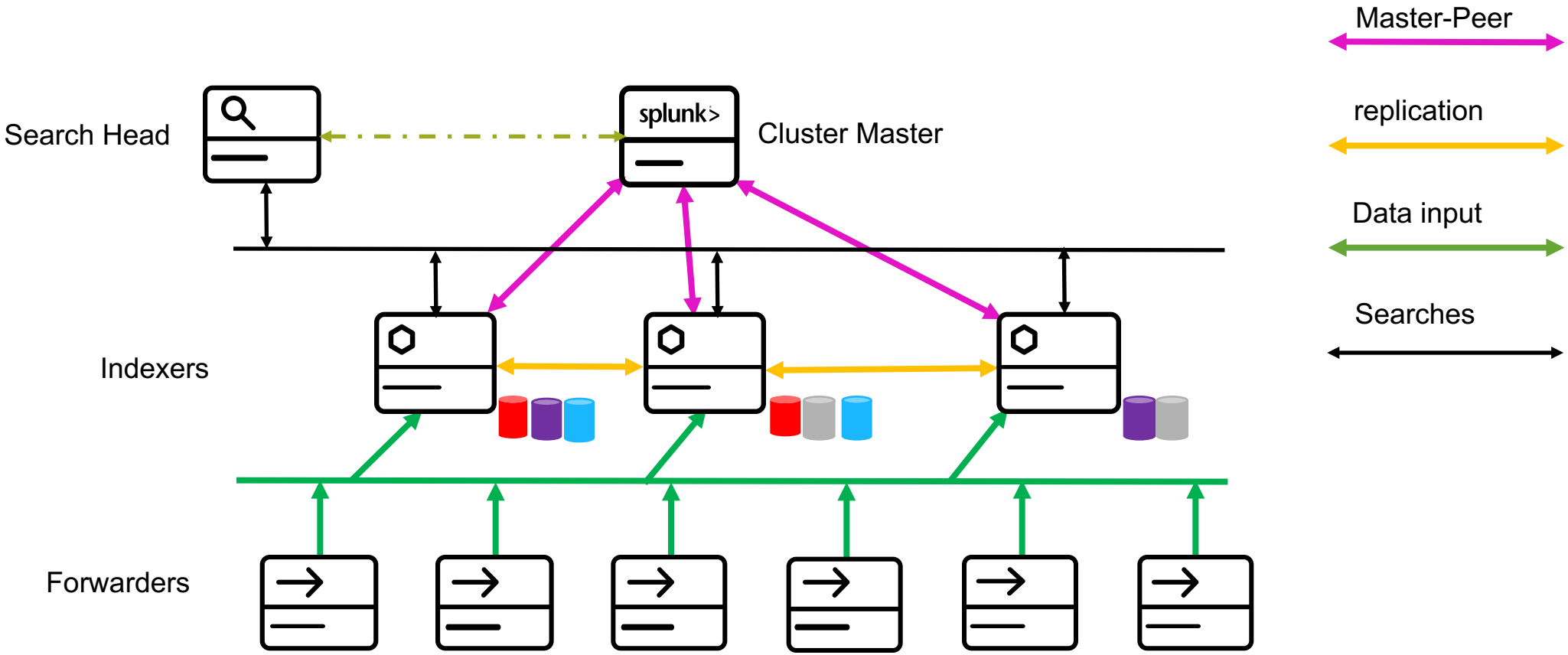
Agenda

1. Introduction to Indexer Clustering
2. Scalability Improvements
3. Performance

Introduction to Indexer Clustering

splunk> .conf2017

Indexer Clustering Topology



Splunk Server

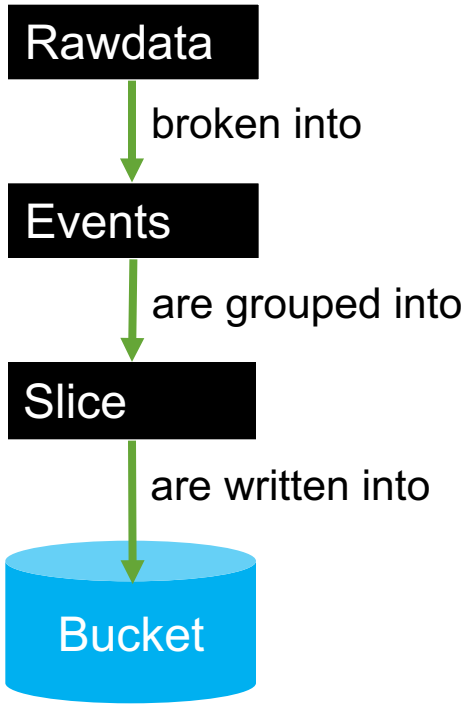
Cluster Nodes

- Cluster Master (CM)
  - Stateless
  - Maintains in-memory state of all the peers and buckets
  - Coordinates the replicating activities of the peer nodes
  - Tells the search head where to find data
- Cluster Peer (Indexer)
  - Indexes, replicates data
  - Responds to the incoming search requests
  - Report its state and all its buckets to CM
- Search head
- Forwarder

Buckets

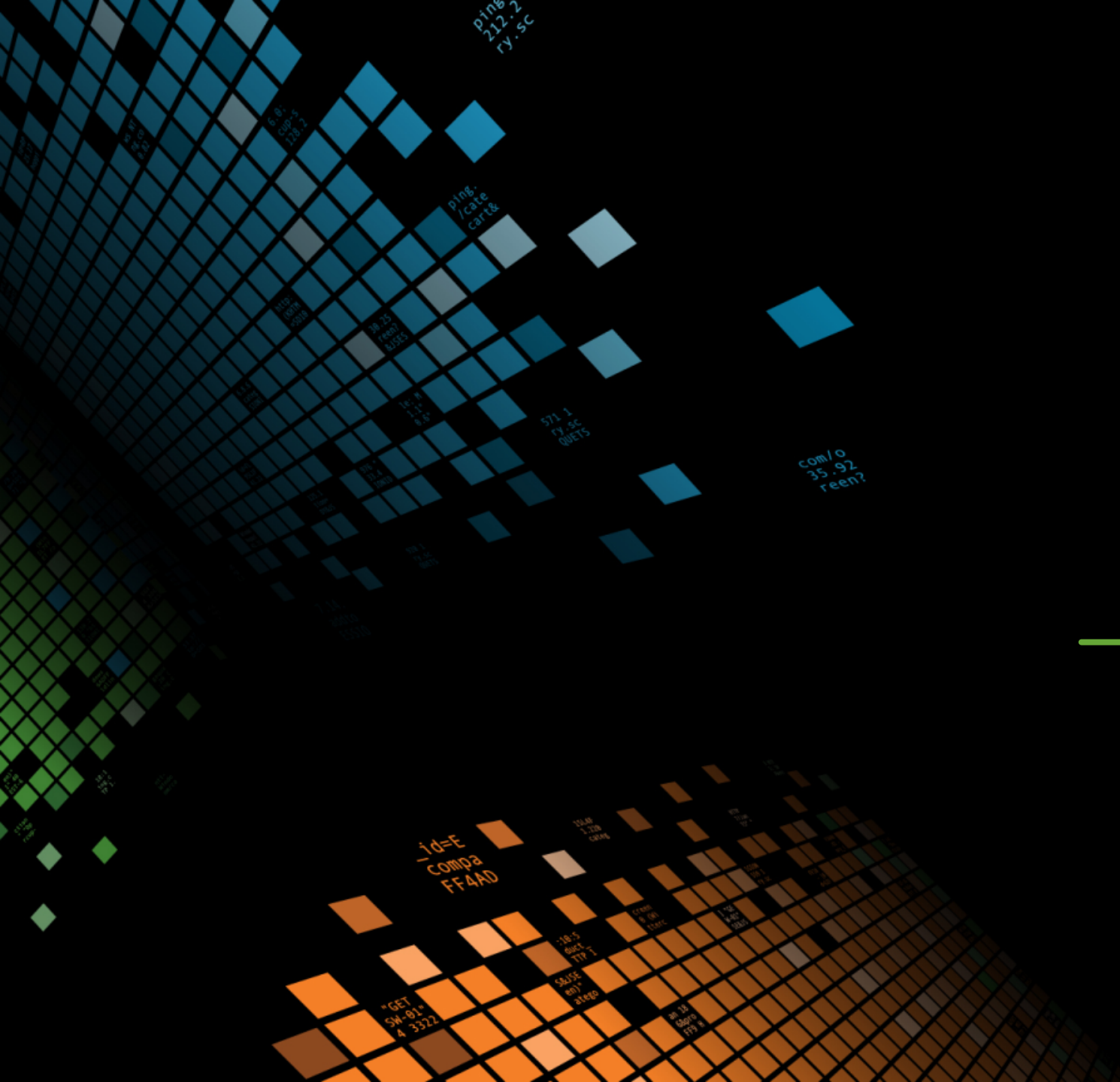
- Unit of data the cluster is aware of
- Created on the indexer
  - Indexer notifies CM upon every state transition of its bucket
- Configurable size
- Bucket manipulation is centric of CM's responsibility
- More data, more buckets

- Flow of bucket creation



Heartbeats

- One of the mechanisms CM uses to communicate with peers
- Status synchronization
- Once peer registers to master, it starts to heartbeat to master every `heartbeat_period` seconds (defaults to 1)
- CM utilizes `heartbeat_timeout` to consider if peer is offline and perform fixup if necessary



Scalability Improvements

splunk> .conf2017

Quick Glance of Improvements since 6.6

- 3x more cluster-wide buckets (vs 6.5)
  - 15M cluster-wide buckets, or 150 PB data (10GB/bucket)
- Faster cluster management with scale
  - rolling restart: ~2x faster
  - peer failure recovery: ~2x faster
  - Improved responsiveness
    - Better average turn-around time: 2x faster and more
  - and more…
- Less memory (vs Splunk 6.6)
  - ~15% reduction in CM

Challenges to scale up

- More data, more buckets
  - more complexity
  - more fixes to reach healthy state
  - more memory consumed, less cache hit-rate, higher cache miss penalty
- Extreme conditions are difficult to fulfill in time
  - Long-running transactions
  - When CM's CPU is burned out, requests could be possibly timeout and result in failure recovery
    - Add into cluster, huge amount of fix jobs
    - More fix ups, higher demand of CPU
  - False positive scenarios due to above reasons
    - unnecessary fixes
- Network latency

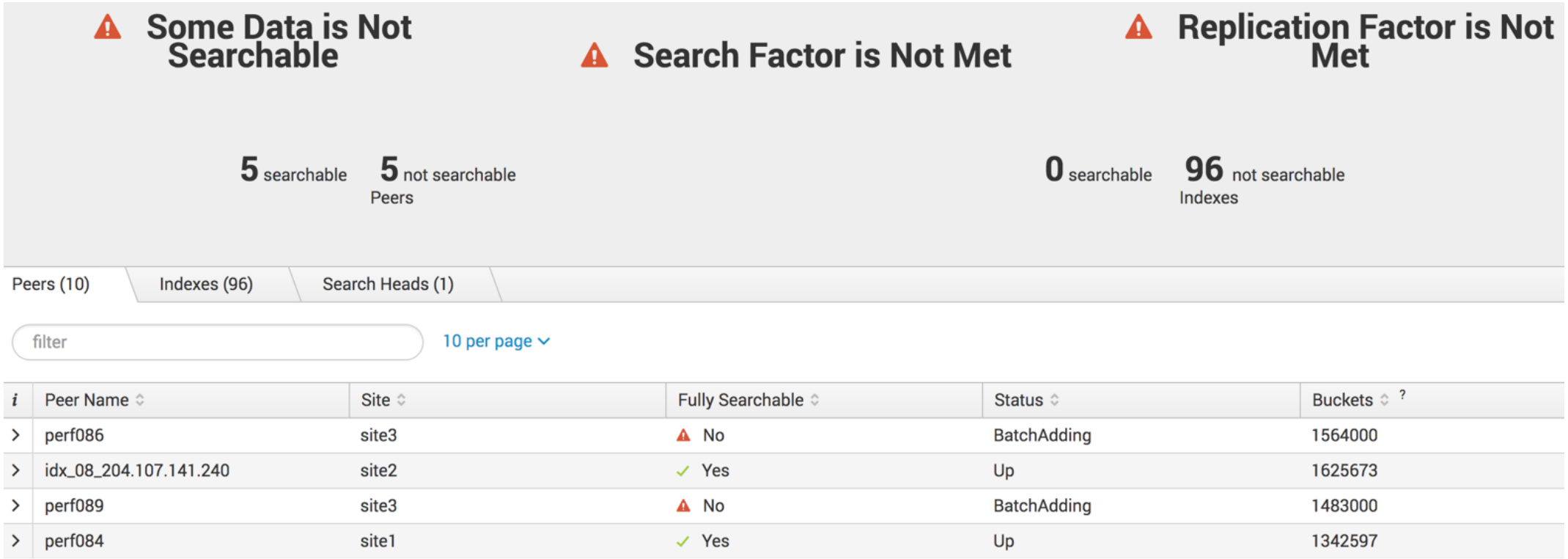
Improvements

- Peer adding with configurable amount of buckets
- Limit job processing time
- Lockless heartbeat
- Enhanced bucket management

Peer adding - configurable amount of buckets

- Splunk 6.5 and prior
  - Peer sends all buckets within a single operation
  - High requirement for CM processing power
  - Might result in unresponsive CM
  - Long-running transaction is fragile
- Splunk 6.6+
  - Configurable amount of buckets to replace long-running add-peer
  - Reduce the requirement of CM computing power
  - Reduce the variation caused by network latency
  - Better turn-around time
  - More responsive CM

Peer adding - configurable amount of buckets



Peer adding - configurable amount of buckets

- buckets_per_addpeer = <non-negative integer> in server.conf
- Defaults to 1000
- The more buckets in add-peer, the quicker it adds to cluster
  - Needs larger REST receive timeout `**rcv_timeout**` and heartbeat timeout `**heartbeat_timeout**`
  - Less responsive CM
  - Less up-to-date cluster
- Needs tweaking along with
  - Computing power of CM
  - Related timeout settings
  - Number of buckets
  - Number of indexers

Limit job processing time

- Splunk 6.5 and prior
  - CM used to process all jobs within a single service duration, which could be many seconds or even minutes
    - Replication (to meet RF)
    - Search (to meet SF)
    - Primality (all buckets need to have a primary copy per site)
    - Others (freezing, checksum, rolling, etc)
- Splunk 6.6+
  - An option to limit how long CM spends per fixup
  - CM is forced to context switch to serve other requests
  - More responsive CM, however fixups take longer

Limit job processing time

- max_fixup_time_ms = <zero or positive integer>
- Defaults to 0 (unused)
- Limits how long each fixup level runs
- Useful on larger clusters having massive amount of buckets
- Enable it when there are massive amount of buckets, and cluster failures result in extra busy cluster master fixup activity (where service duration takes more than 10 seconds)

Lockless heartbeat

- Splunk 6.5 and prior
  - The heartbeat processing in CM was blocking
  - Frequent heartbeat and massive indexers will result in less responsive CM
  - False positive errors
- Splunk 6.6+
  - Re-designed heartbeat mechanism
  - Not blocking anymore
  - Incoming heartbeats are stored, cached, and processed later
  - Scaling up with increasing number of indexers won't significantly affect CM responsiveness

Enhanced bucket management

- Bucket is centric of CM's responsibility
  - Bucket manipulation is required by mostly all processing in CM
  - Improving it will generally improve all critical paths
- Available since Splunk 7.0
- Bonus: ~15% memory saving in master node

Test Case	Splunk 6.6	Splunk 7.0
1M bucket Insertions	60ms	2.42ms
1M bucket Lookups	60ms	2.26ms

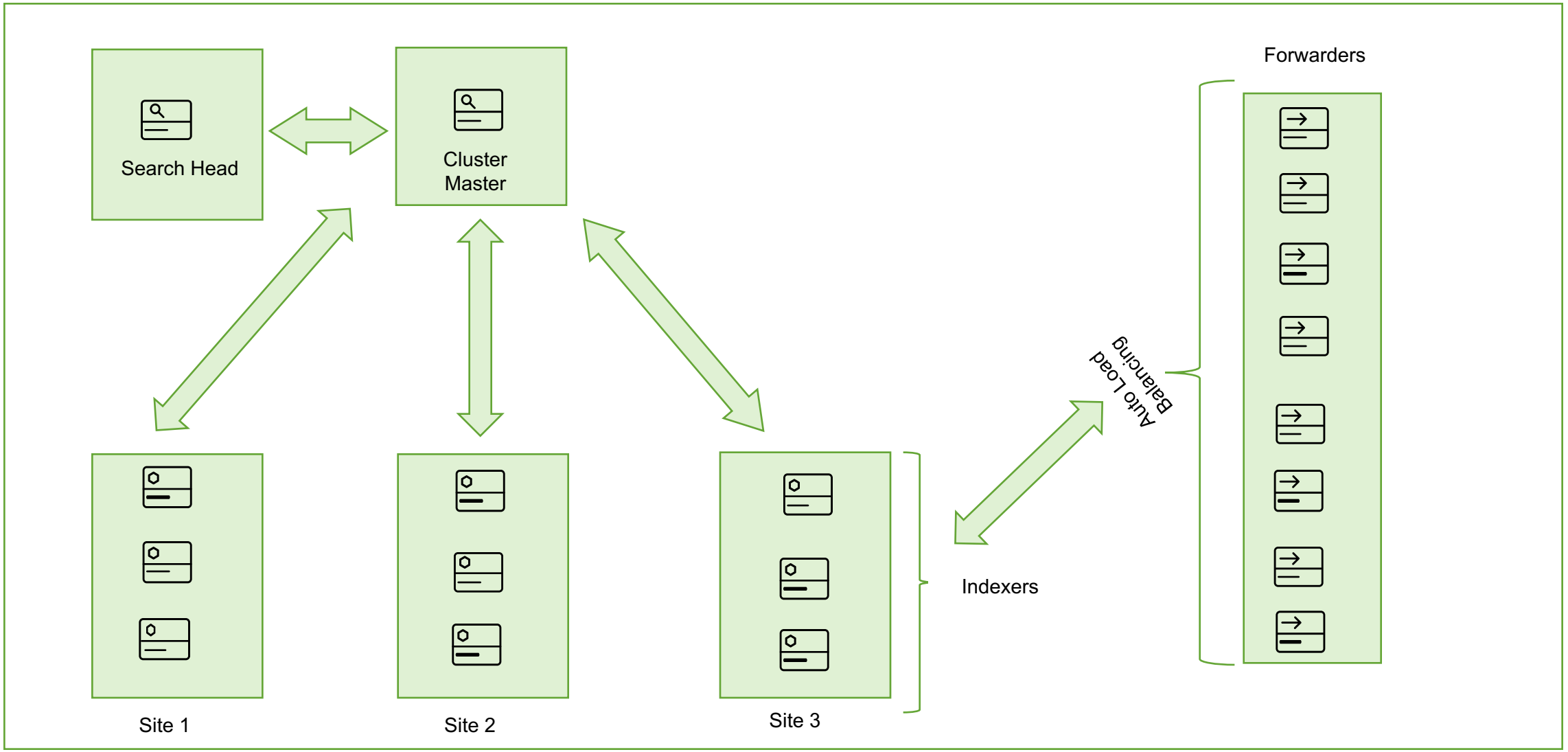
splunk> .conf2017

Indexer Clustering Performance

Scale tests with 5 Million Unique Buckets

splunk> .conf2017

Deployment



Test Configuration

Machine Specs

- 2x12 Xeon 2.30 GHz
- 24 cores (48 w/HT)
- 128 GB RAM
- 8 x 300GB 15k RPM disks in RAID-0
- 1 Gb Ethernet NIC
- CentOS 7.6

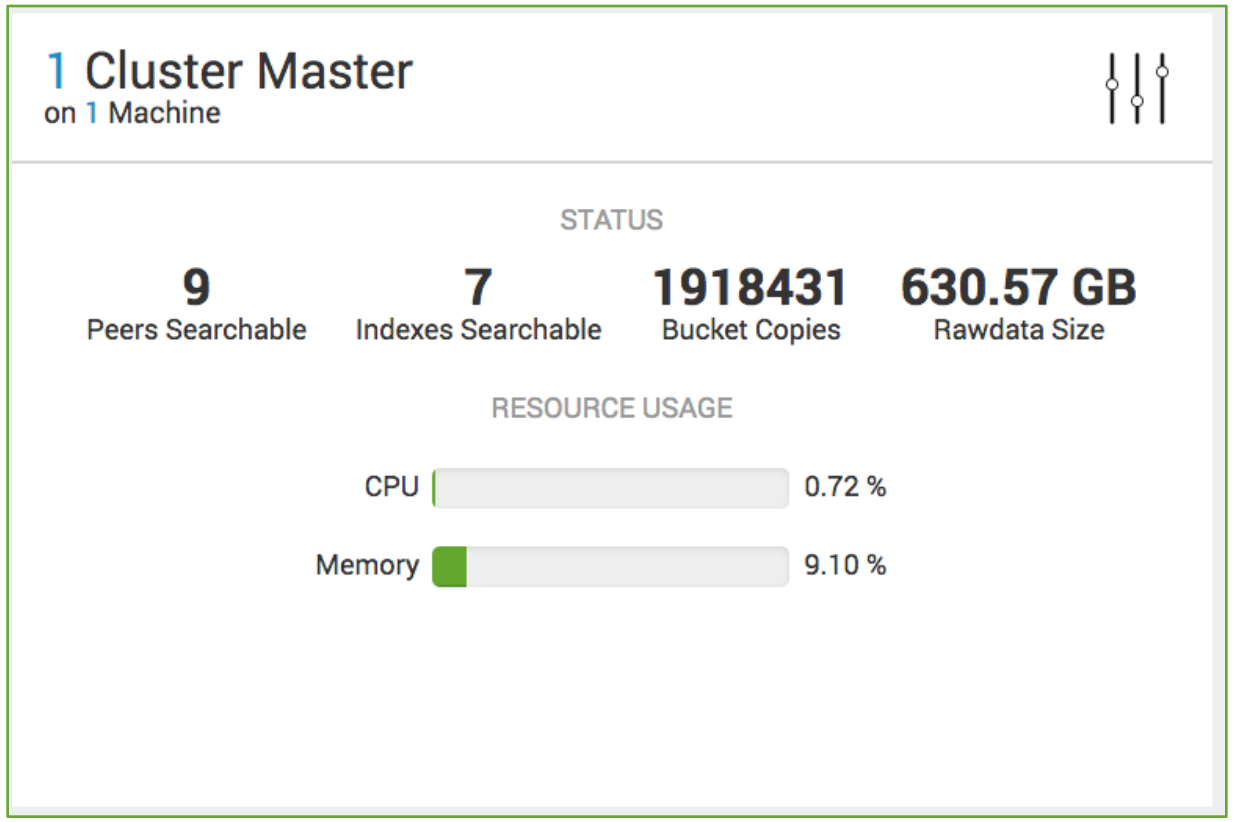
Multi-site Cluster Configuration

- 3 sites
- Replication factor – origin:2, total:3
- Search factor – origin:1, total:2

No other load on the box

1.9 Million Bucket Test

Splunk 6.5 vs Splunk 7.0



- 640,000 unique buckets
- 200,000 buckets/peer
- Default cluster timeout configurations
- No Search or Indexing load

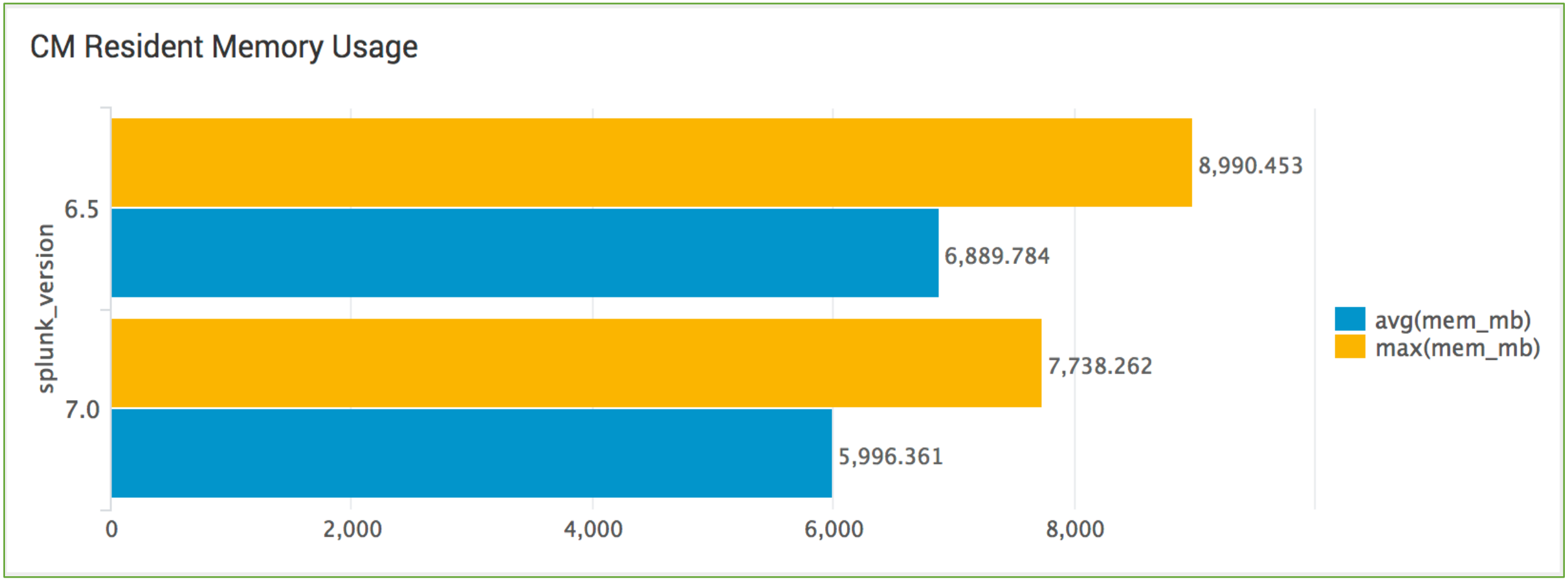
Test Results

Test Case	Success Condition	Time for Completion	
		Splunk 6.5	Splunk 7.0
CM Restart	Cluster Complete	43.08 secs	36.7 secs
Peer Failure	Cluster Searchable	60 + 73.8 secs	60 + 8.6 secs
Rolling Restart	Cluster Complete	19.1 mins	10.1 mins
Bundle Push	100 MB Pushed	49.2 secs	25.8 secs

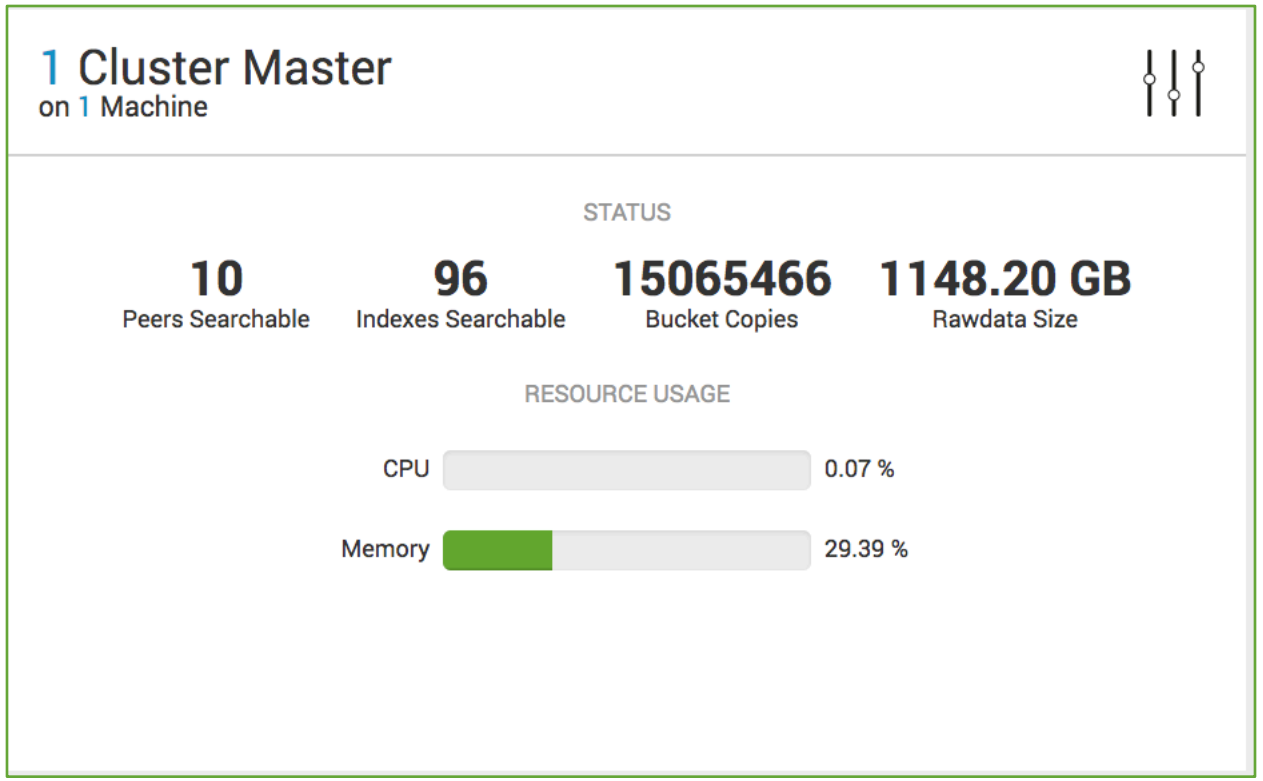
Improved CM responsiveness

Endpoints	Average Time (milliseconds)	
	Splunk 6.5	Splunk 7.0
/services/cluster/master/generation	778 ms	262 ms
/services/cluster/master/peers	1972 ms	119 ms
/services/cluster/master/buckets	818 ms	440 ms
/services/cluster/master/info	4293 ms	1796 ms
/services/cluster/master/indexes	1874 ms	598 ms

CM Memory Footprint

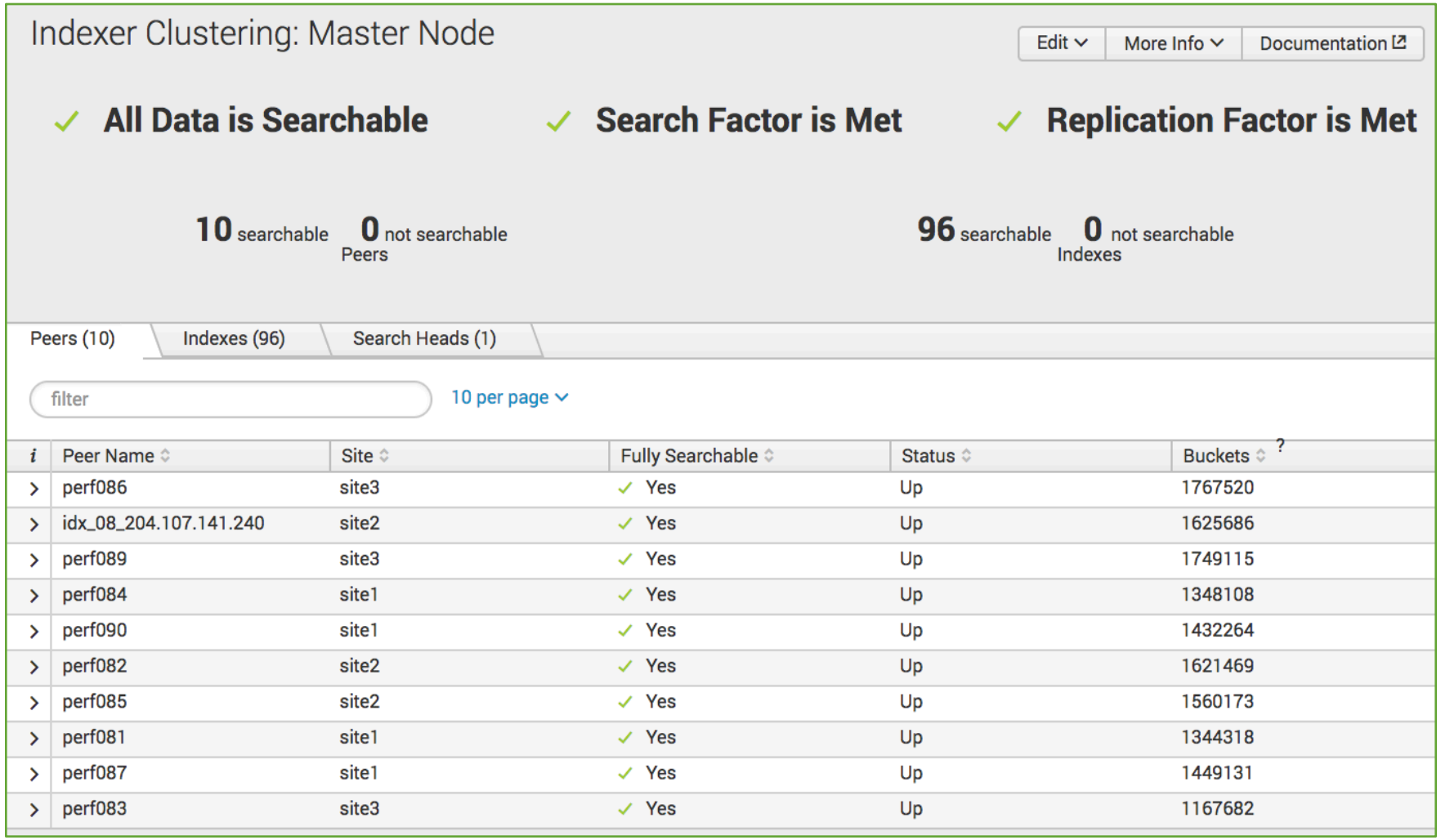


15 million Bucket Test



- 5 million unique buckets
- 3x more buckets cluster-wide compared to Splunk 6.5
- No Search or Indexing Load

Buckets per Peer



i	Peer Name	Site	Fully Searchable	Status	Buckets
>	perf086	site3	✓ Yes	Up	1767520
>	idx_08_204.107.141.240	site2	✓ Yes	Up	1625686
>	perf089	site3	✓ Yes	Up	1749115
>	perf084	site1	✓ Yes	Up	1348108
>	perf090	site1	✓ Yes	Up	1432264
>	perf082	site2	✓ Yes	Up	1621469
>	perf085	site2	✓ Yes	Up	1560173
>	perf081	site1	✓ Yes	Up	1344318
>	perf087	site1	✓ Yes	Up	1449131
>	perf083	site3	✓ Yes	Up	1167682

splunk> .conf2017

Test Results

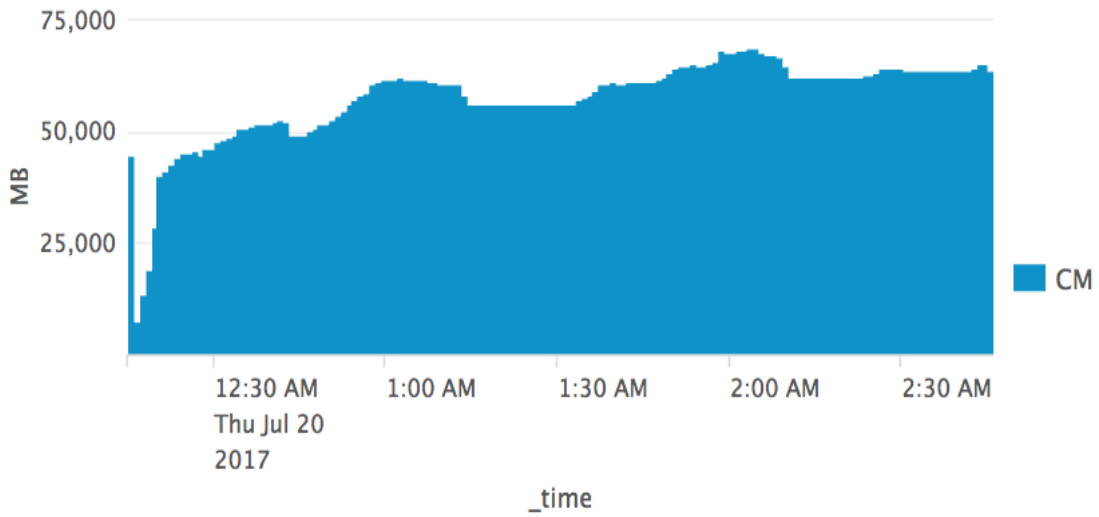
Test Case	Success Condition	Time for Completion
CM Restart	Cluster Complete	5.73 mins
Peer Failure	Cluster Searchable	600 + 140.1 secs
Rolling Restart	Cluster Complete	31.91 mins
Bundle Push	100 MB Pushed	112.2 secs

CM Memory Footprint

- 18.5 % reduction in avg. Resident Memory & 23 % reduction in max Resident Memory

Splunk 6.6

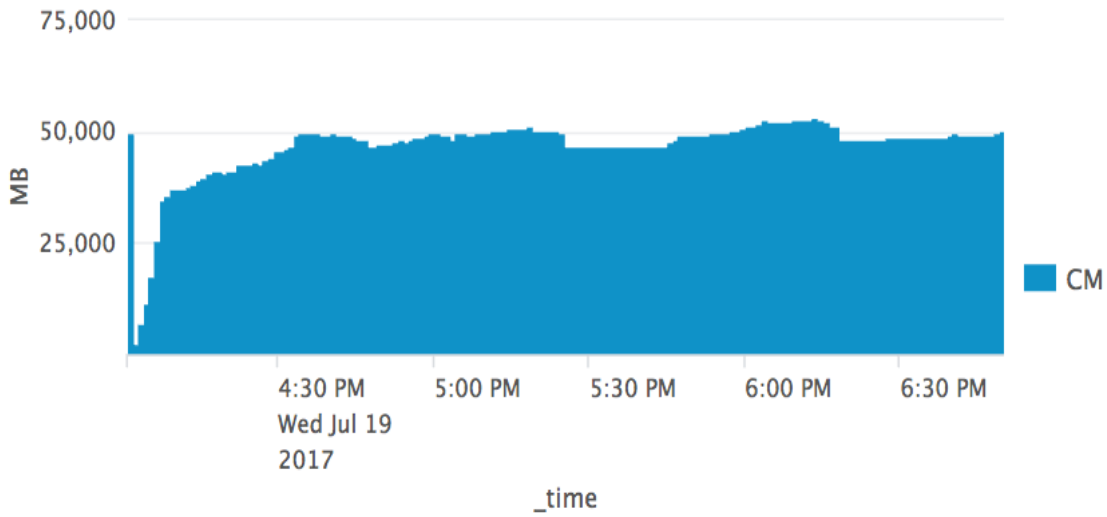
Cluster Master



splunk_server	avg(data.mem_used)	perc90(data.mem_used)	max(data.mem_used)
CM	57450.70899113083	65030.000	69074.469

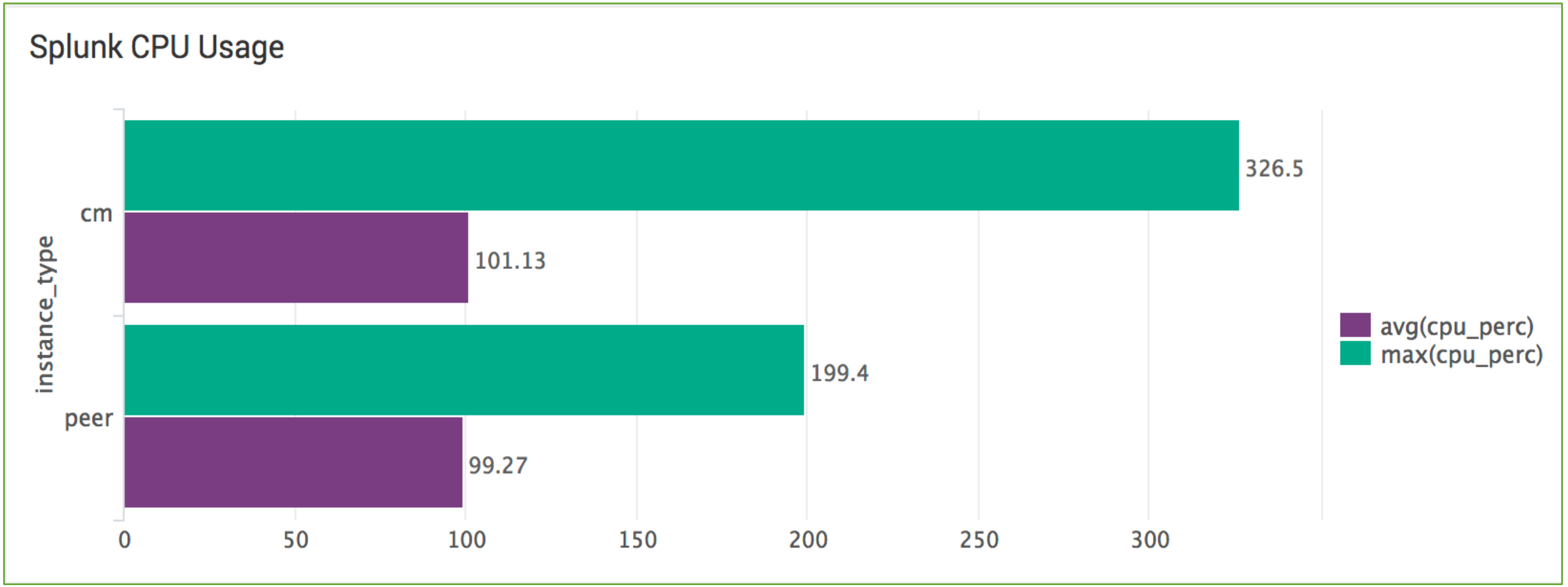
Splunk 7.0

Cluster Master



splunk_server	avg(data.mem_used)	perc90(data.mem_used)	max(data.mem_used)
CM	46807.2043313492	51060.000	53186.609

Splunk CPU Usage



Cluster Configuration

Cluster Master - server.conf

[clustering]

- heartbeat_timeout = 600
- rcv_timeout = 600
- send_timeout = 600
- cxn_timeout = 600
- max_fixup_time_ms = 5000

Indexers - server.conf

[clustering]

- heartbeat_period = 40
- cxn_timeout = 600
- send_timeout = 600
- rcv_timeout = 600
- buckets_per_addpeer = 1000

splunk> .conf2017

Key Takeaways

1. Enhancements - Peer adding with configurable amount of buckets, Lockless heartbeat, Limit job processing time etc.
2. CM scales up to 5+ million unique buckets (3x more compared to Splunk 6.5)
3. Faster Cluster recovery and Improved CM responsiveness.
4. ~15% CM memory reduction.

Thank You

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