

What's New in Splunk 7.3

Dave Shpritz, Aplura Splunk Practice Lead

Baltimore Splunk User Group

June 17th, 2019

32616	splunkd	%CPU	0.3	USER	splunk	UID	1000	TIME+	0:02.80	MEM	4.3	PR	20	NI	0	VIRT	135356	RES	43956	SHR	S	PPID	1000	PPID	1000	USER	splunk	MEM	4.3	PR	20	NI	0	VIRT	135356	RES	43956	SHR	S	PPID	1000	PPID	1000	USER	splunk
32695	tcpdump	%CPU	0.3	USER	tcpdump	UID	72	TIME+	0:02.45	MEM	0.7	PR	20	NI	0	VIRT	28588	RES	6872	SHR	S	PPID	1000	PPID	1000	USER	tcpdump	MEM	0.7	PR	20	NI	0	VIRT	28588	RES	6872	SHR	S	PPID	1000	PPID	1000	USER	tcpdump
590	named	%CPU	0.3	USER	named	UID	25	TIME+	7:31.12	MEM	1.8	PR	20	NI	0	VIRT	548272	RES	18872	SHR	S	PPID	1000	PPID	1000	USER	named	MEM	1.8	PR	20	NI	0	VIRT	548272	RES	18872	SHR	S	PPID	1000	PPID	1000	USER	named
1602	vim	%CPU	0.3	USER	dan	UID	1028	TIME+	0:00.08	MEM	0.6	PR	20	NI	0	VIRT	152828	RES	6248	SHR	S	PPID	1000	PPID	1000	USER	dan	MEM	0.6	PR	20	NI	0	VIRT	152828	RES	6248	SHR	S	PPID	1000	PPID	1000	USER	dan
1243	python	%CPU	0.3	USER	python	UID	1000	TIME+	0:00.00	MEM	0.0	PR	20	NI	0	VIRT	0	RES	0	SHR	S	PPID	1000	PPID	1000	USER	python	MEM	0.0	PR	20	NI	0	VIRT	0	RES	0	SHR	S	PPID	1000	PPID	1000	USER	python

Splunk 7.3

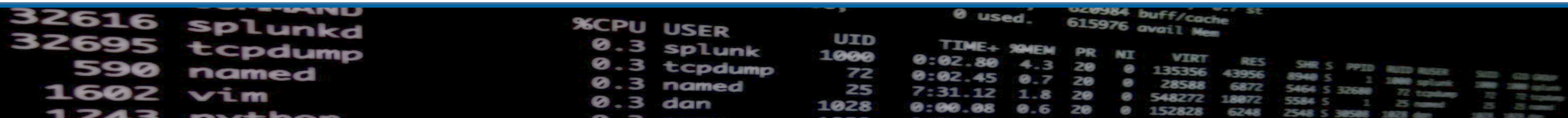
- Codename: PinkyPie
- “Dark Data”
- More getting data in (and getting in more data)
- Other market-y stuff (AR, Mobile)
- Not really interested in this stuff, so we aren’t going to cover it



0 used.		0.00984 buff/cache		0.00984 avail Mem														
32616	splunkd	%CPU	USER	UID	TIME+	XMEM	PR	NI	VIRT	RES	SHR	S	PPID	USED	INUSE	SWIO	CDD	GROUP
32695	tcpdump	0.3	splunk	1000	0:02.80	4.3	20	0	135356	43956	8940	S	1	1000	splunk	1000	splunk	splunk
590	named	0.3	tcpdump	72	0:02.45	0.7	20	0	28588	6872	5464	S	32680	tcpdump	72	tcpdump	72	tcpdump
1602	vim	0.3	named	25	7:31.12	1.8	20	0	548272	18072	5584	S	1	25	named	85	named	named
1243	python	0.3	dan	1028	0:00.08	0.6	20	0	152828	6248	2548	S	30580	python	1028	python	1028	python

What are we going to cover?

- SmartStore (S2)
- Searchable Data Rebalance
- Indexer Clustering Performance
- Search Performance
- Some Cloud
- SHC Deployer changes
- Indexing Pipeline
- Metrics
- Workload Management (WLM)
- Token Authentication (finally)
- LDAP
- Time Fields

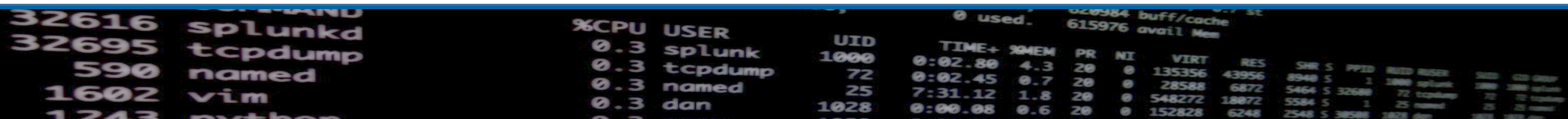


A terminal window screenshot showing system metrics and a process list. The top part of the terminal displays memory usage statistics: 0 used, 620984 buff/cache, and 615976 avail Mem. Below this, a table lists system metrics for various processes. The table has columns for %CPU, USER, UID, TIME+, %MEM, PR, NI, VIRT, RES, SHR, S, PPID, and several other columns. The processes listed include splunkd, tcpdump, named, vim, and python.

%CPU	USER	UID	TIME+	%MEM	PR	NI	VIRT	RES	SHR	S	PPID	...
0.3	splunk	1000	0:02.80	4.3	20	0	135356	43956	8940	S	1	...
0.3	tcpdump	72	0:02.45	0.7	20	0	28588	6872	5464	S	32680	...
0.3	named	25	7:31.12	1.8	20	0	548272	18872	5584	S	1	...
0.3	dan	1028	0:00.08	0.6	20	0	152828	6248	2548	S	38580	...

SmartStore (S2)

- Now supports Report and Data Model accelerations
- This means ES is now supported on SmartStore
- New retention settings (size)
- Support for non-clustered indexers and indexes
- Better resiliency (?)
- Better scalability (?)

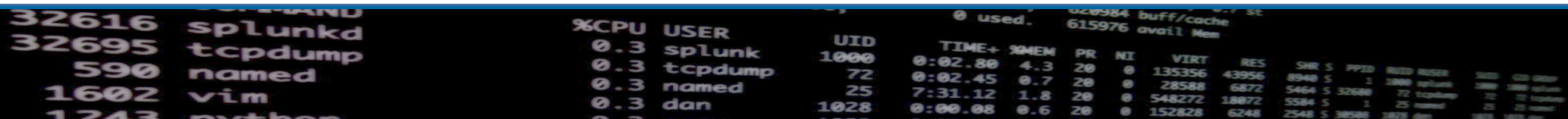


A terminal window screenshot showing system statistics and a list of processes. The top part of the terminal displays memory usage statistics: '0 used', '620984 buff/cache', and '615976 avail Mem'. Below this, a table lists processes with columns for PID, CPU usage, user, UID, TIME+, MEM, PR, NI, VIRT, RES, SHR, S, PPID, and other system metrics. The processes listed include splunkd, tcpdump, named, vim, and python.

PID	%CPU	USER	UID	TIME+	MEM	PR	NI	VIRT	RES	SHR	S	PPID	OTHER
32616	0.3	splunkd	1000	0:02.80	4.3	20	0	135356	43956	8940	S	1	named
32695	0.3	splunk	72	0:02.45	0.7	20	0	28588	6872	5464	S	32680	72
590	0.3	tcpdump	25	7:31.12	1.8	20	0	548272	18872	5584	S	1	25
1602	0.3	named	1028	0:00.08	0.6	20	0	152828	6248	2548	S	38584	1602
1243	0.3	vim											
1243	0.3	python											

Searchable Data Rebalance

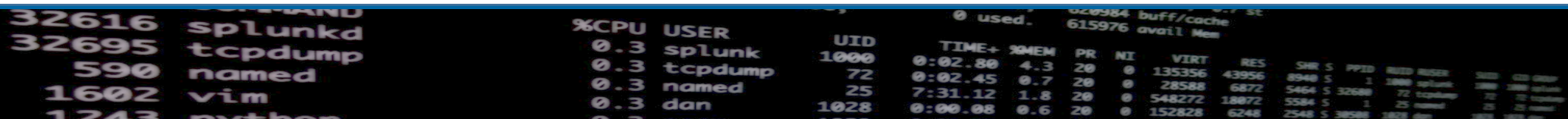
- Everyone has to be on 7.3
- Previously, data rebalance is not search safe
- The removal of buckets could cause differences/inaccuracy in results
- 7.3 added workflow to make removal of excess buckets search safe
- Available as a checkbox when performing rebalance
- Other things aren't available when doing this
 - Excess bucket removal
 - Rolling restart
 - Rolling upgrade
- There is a timeout, so longer running searches will still be subject, as will indexed real-time searches (mostly in hot, so mostly should be ok)



	%CPU	USER	UID	TIME+	PMEM	PR	NI	VIRT	RES	SHR	S	PPID	PPID	NAME
32616	0.3	splunkd	1000	0:02.80	4.3	20	0	135356	43956	8940	S	1	2880	splunkd
32695	0.3	splunk	72	0:02.45	0.7	20	0	28588	6872	5464	S	32680	72	tcpdump
590	0.3	named	25	7:31.12	1.8	20	0	548272	18872	5584	S	1	25	named
1602	0.3	vim	1028	0:00.08	0.6	20	0	152828	6248	2548	S	38584	1602	vim
1243	0.3	python												

Indexer Clustering Performance

- Changes to how the UI displays data
- Used to show intermediate changes, now caches
- More logging (event=rfMet, event=sfMet, event=allSearchable)
- Could mean that on a rolling restart, SF/RF appears to “flap” more
- tsidxWritingLevel = 3



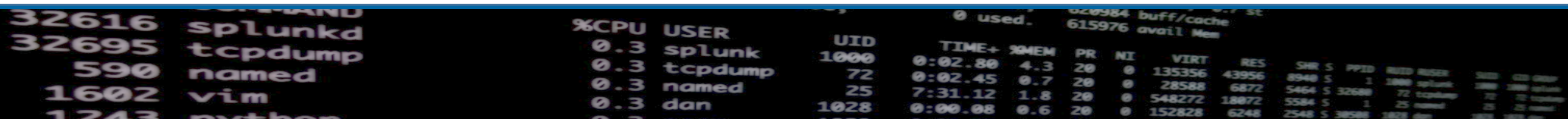
The screenshot shows a terminal window with two main sections of output. The top section lists system metrics including CPU usage (0.3), memory usage (4.3), and various system parameters. The bottom section lists running processes with columns for PID, username, and other details.

System Metrics			
%CPU	USER	UID	TIME+ MEM PR NI VIRT RES
0.3	splunk	1000	0:02.80 4.3 20 0 135356 43956
0.3	tcpdump	72	0:02.45 0.7 20 0 28588 6872
0.3	named	25	7:31.12 1.8 20 0 548272 18872
0.3	dan	1028	0:00.08 0.6 20 0 152828 6248

PID	USER	Other Info
32616	splunkd	
32695	tcpdump	
590	named	
1602	vim	
1243	python	

Search Performance

- Lots of smaller improvements
- Some larger ones too
 - Stats to tstats
 - tsidxWritingLevel and Data Model Acceleration
 - Data Model UI index constraints
 - CIDR matching
 - Compression
 - Datamodel command
 - Post Process

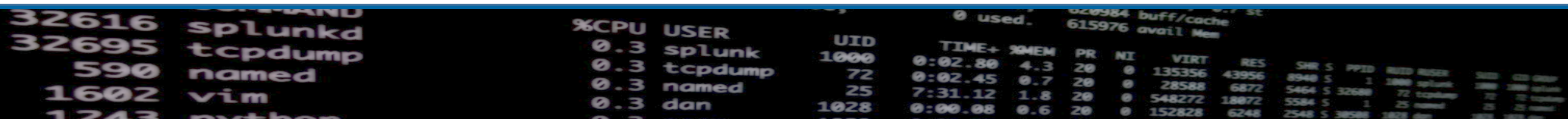


A terminal window screenshot showing system statistics and a list of processes. The top line displays memory usage: 0 used, 620984 buff/cache, 615976 avail Mem. Below this is a table of processes with columns for PID, COMMAND, %CPU, USER, UID, TIME+, MEM, PR, NI, VIRT, RES, SHR, S, PPID, and PGRP. The processes listed are splunkd, tcpdump, named, vim, and python.

PID	COMMAND	%CPU	USER	UID	TIME+	MEM	PR	NI	VIRT	RES	SHR	S	PPID	PGRP
32616	splunkd	0.3	splunk	1000	0:02.80	4.3	20	0	135356	43956	8940	S	1	3000
32695	tcpdump	0.3	tcpdump	72	0:02.45	0.7	20	0	28588	6872	5464	S	32680	72
590	named	0.3	named	25	7:31.12	1.8	20	0	548272	18872	5584	S	1	25
1602	vim	0.3	dan	1028	0:00.08	0.6	20	0	152828	6248	2548	S	30580	1602
1243	python	0.3	python	1000	0:00.00	0.0	20	0	152828	6248	2548	S	30580	1602

Search Performance - Stats to tstats

- Optimizes searches that use stats command
- Converts them to tstats under-the-hood
- On by default, but can be disabled using the “noop” command
- Will work with any indexed field
 - As long as they are in fields.conf
 - Remember, fields.conf is not sourcetype scoped, so, careful with that ax

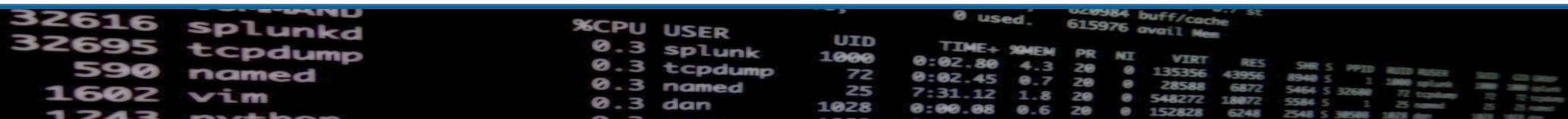


The screenshot shows a Splunk search results table with columns: PID, COMMAND, %CPU, USER, UID, TIME+, MEM, PR, NI, VIRT, RES, SHR, S, PPID, PWD, PATH, and others. The data rows show processes like splunkd, tcpdump, named, vim, and python.

PID	COMMAND	%CPU	USER	UID	TIME+	MEM	PR	NI	VIRT	RES	SHR	S	PPID	PWD	PATH
32616	splunkd	0.3	splunk	1000	0:02.80	4.3	20	0	135356	43956	8940	S	1	/usr/bin	/usr/bin/splunkd
32695	tcpdump	0.3	tcpdump	72	0:02.45	0.7	20	0	28588	6872	5464	S	32680	72	/usr/sbin/tcpdump
590	named	0.3	named	25	7:31.12	1.8	20	0	548272	18872	5584	S	1	25	/usr/sbin/named
1602	vim	0.3	dan	1028	0:00.08	0.6	20	0	152828	6248	2548	S	30580	1602	/usr/bin/vim
1243	python	0.3	python	1028	0:00.08	0.6	20	0	152828	6248	2548	S	30580	1602	/usr/bin/python

Search Performance - tsidxWritingLevel and Data Model Acceleration

- Writing level in 7.2 (level 2 introduced in 7.2, 7.3 adds a level 3)
- All indexers and search heads have to have this set
- A collection of enhancements to how tsidx files are written/structured
- Large space and search performance gains
- Previously the DMA tsidx files were only using level 1 (even if level 2 was set for the index)
- DMA now will use the same enhancements

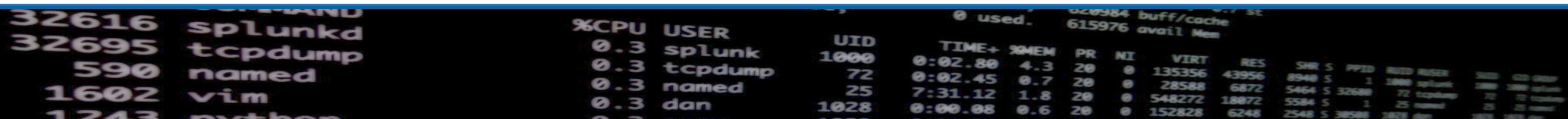


The image shows a terminal window with two main sections. The top section displays system status information, including memory usage (620984 buff/cache, 615976 avail Mem) and a list of processes. The bottom section shows a detailed process list with columns for PID, CPU, USER, UID, TIME+, MEM, PR, NI, VIRT, RES, SHR, S, PPID, and others.

PID	CPU	USER	UID	TIME+	MEM	PR	NI	VIRT	RES	SHR	S	PPID
32616	0.3	splunkd	1000	0:02.80	4.3	20	0	135356	43956	8940	S	1
32695	0.3	tcpdump	72	0:02.45	0.7	20	0	28588	6872	5464	S	32680
590	0.3	named	25	7:31.12	1.8	20	0	548272	18872	5584	S	1
1602	0.3	vim	1028	0:00.08	0.6	20	0	152828	6248	2548	S	38580
1243	0.3	python										

Search Performance - Data Model UI index constraints

- Best practice is to have index constraints in place
- CIM app uses macros to implement this (please check yo' self)
- 7.3 enforces that a DM must have a constraint in place
- Should be macro aware (like the macros in CIM)
- Can still have non-constrained searches in JSON
- Index=* is a valid constraint ☹️



The screenshot shows a terminal window with two main sections. The top section lists processes with their PIDs and names: 32616 splunkd, 32695 tcpdump, 590 named, 1602 vim, and 1243 python. The bottom section displays a table of system metrics.

%CPU	USER	UID	TIME+	PMEM	PR	NI	VIRT	RES	SHR	S	PPID	PPID	PPID	PPID	PPID	PPID	PPID	PPID	PPID
0.3	splunk	1000	0:02.80	4.3	20	0	135356	43956	8940	S	1	2880	tcpdump	2880	2880	tcpdump	2880	2880	tcpdump
0.3	tcpdump	72	0:02.45	0.7	20	0	28588	6872	5464	S	32680	72	tcpdump	72	72	tcpdump	72	72	tcpdump
0.3	named	25	7:31.12	1.8	20	0	548272	18872	5584	S	1	25	named	25	25	named	25	25	named
0.3	dan	1028	0:00.08	0.6	20	0	152828	6248	2548	S	38584	1028	dan	1028	1028	dan	1028	1028	dan

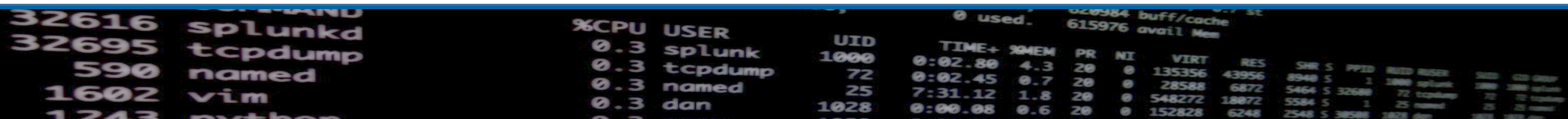
Search Performance - CIDR matching

- General performance improvements
- tstats would not perform negated CIDR, now it does
- Search is now IPv6 CIDR aware (no love for tstats)

	%CPU	USER	UID	TIME+	XMEM	PR	NI	VIRT	RES	SHR	S	PID	NEXT USER:	KIDS	CMD GROUP
32616 splunkd	0.3	splunk	1000	0:02.80	4.3	20	0	135356	43956	8940	5	1	1800 splunk	1800	1800 splunk
32695 tcpdump	0.3	tcpdump	72	0:02.45	0.7	20	0	28588	6872	5464	5	32695	tcpdump	72	tcpdump
590 named	0.3	named	25	7:31.12	1.8	20	0	548272	18072	5584	5	1	25 named	85	named
1602 vim	0.3	dan	1028	0:00.08	0.6	20	0	152828	6248	2548	5	38580	vim dan	1028	vim dan
1243 python	0.3	python	1028	0:00.08	0.6	20	0	152828	6248	2548	5	38580	vim dan	1028	vim dan

Search Performance - Compression

- Zstandard compression (look, Facebook did something good!)
- Less space, less CPU usage
- 7.2 introduced this for journals
- Search results still used gzip
- Now defaults to zstd
- You can use a splunkd command to decompress
- Alert actions still get gzip
- Note that there is no zstandard decompression module in the bundled python



A terminal window screenshot showing system resource usage. The left side lists processes with their PIDs and names. The right side shows a detailed table of resource usage for selected processes.

COMMAND	%CPU	USER	UID	TIME+	MEM	PR	NI	VIRT	RES	SHR	S	PPID	PPID	PPID	PPID	PPID	PPID	PPID	PPID
32616 splunkd	0.3	splunk	1000	0:02.80	4.3	20	0	135356	43956	8940	S	1	2880	splunk	2880	2880	2880	2880	2880
32695 tcpdump	0.3	tcpdump	72	0:02.45	0.7	20	0	28588	6872	5464	S	32680	72	tcpdump	72	72	72	72	72
590 named	0.3	named	25	7:31.12	1.8	20	0	548272	18872	5584	S	1	25	named	25	25	25	25	25
1602 vim	0.3	dan	1028	0:00.08	0.6	20	0	152828	6248	2548	S	38584	1602	vim	1602	1602	1602	1602	1602
1243 python	0.3																		

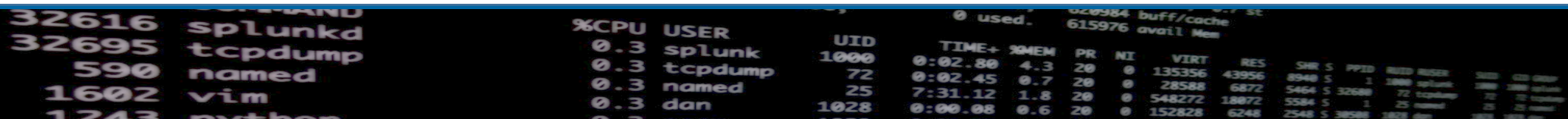
Search Performance - Datamodel command

- New flag for the “datamodel” command to allow the use of “summariesonly=true”
- Should allow for faster drilldowns
- Not available for the “from” command

[illegible]

Search Performance - Post Process

- Post process searches used to be run by the same splunkd process
- Could cause memory issues
- Makes the execution of them smarter, moves them to search pipelines
- Config options can disable this if there are problems

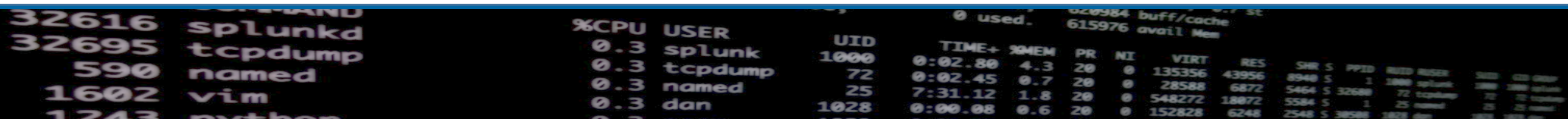


A terminal window screenshot showing system statistics and a list of processes. The top part of the image shows memory usage: 020984 buff/cache, 615976 avail Mem. Below this is a table of processes with columns for PID, PPID, USER, CPU, MEM, and other metrics. The processes listed are splunkd, tcpdump, named, vim, and python.

PID	PPID	USER	CPU	MEM	Other
32616		splunkd	0.3		
32695		tcpdump	0.3		
590		named	0.3		
1602		vim	0.3		
1243		python	0.3		

Some Cloud

- Better interface for the index manager page
 - Makes SmartStore retention easier
- Relative Search Concurrency
 - Now in the UI
 - On prem too
 - Includes "max_searches_perc" and "auto_summary_perc"
 - Check yo self

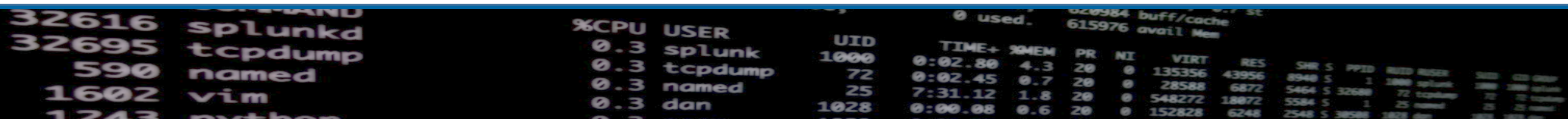


A terminal window screenshot showing system statistics and a list of processes. The top part shows memory usage: 0 used, 620984 buff/cache, 615976 avail Mem. Below that is a table of processes with columns for PID, PPID, USER, and COMMAND. The processes listed are splunkd, tcpdump, named, vim, and python.

PID	PPID	USER	COMMAND
32616		splunkd	
32695		tcpdump	
590		named	
1602		vim	
1243		python	

SHC Deployer Changes

- Review: config merging
- Now we get some control on this via app.conf and the [shclustering] stanza
- `deployer_lookups_push_mode`
 - `preserve_lookups` (honors CLI)
 - `always_preserve` (ignores CLI)
 - `always_overwrite` (ignores CLI)
- `deployer_push_mode`
 - `merge_to_default`
 - `local_only`
 - `default_only`
 - `full`

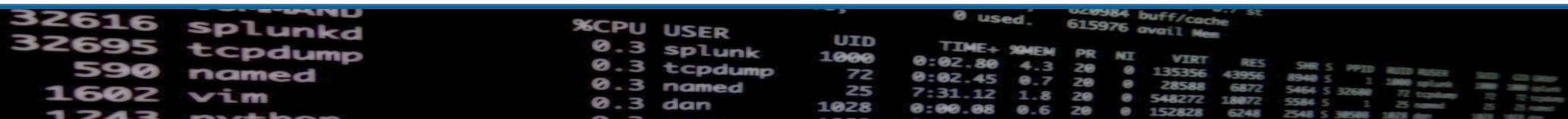


A terminal window showing system statistics and a list of processes. The top part shows memory usage: 0 used, 620984 buff/cache, 615976 avail Mem. Below that is a table of processes with columns for PID, COMMAND, %CPU, USER, UID, TIME+, MEM, PR, NI, VIRT, RES, SHR, S, PPID, and PWD. The processes listed are splunkd, tcpdump, named, vim, and python.

PID	COMMAND	%CPU	USER	UID	TIME+	MEM	PR	NI	VIRT	RES	SHR	S	PPID	PWD
32616	splunkd	0.3	splunk	1000	0:02.80	4.3	20	0	135356	43956	8940	S	1	/usr/sbin/splunkd
32695	tcpdump	0.3	tcpdump	72	0:02.45	0.7	20	0	28588	6872	5464	S	32680	/usr/sbin/tcpdump
590	named	0.3	named	25	7:31.12	1.8	20	0	548272	18872	5584	S	1	/usr/sbin/named
1602	vim	0.3	dan	1028	0:00.08	0.6	20	0	152828	6248	2548	S	30580	/usr/bin/vim
1243	python	0.3	python	1028	0:00.08	0.6	20	0	152828	6248	2548	S	30580	/usr/bin/python

SHC Push Modes

- merge_to_default
 - Default
 - Like what we currently have
- local_only
 - Only pushes /local configs
 - Could be used for something like built-in apps (“search”)
 - Only delivered to the captain
- default_only
 - Only pushes /default configs
 - Gets delivered to all nodes/members
- full
 - No merging
 - default to default, local to local

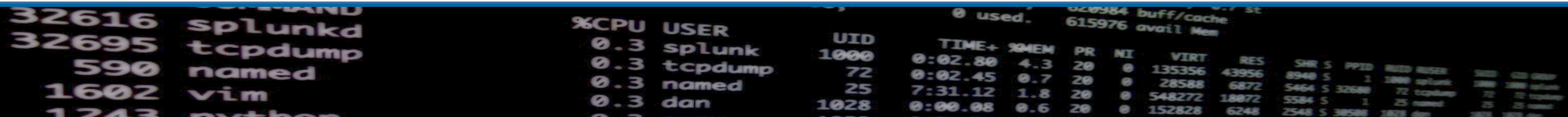


A terminal window showing system status and process information. The top line displays memory usage: 0 used, 620984 buff/cache, 615976 avail Mem. Below this is a table of processes with columns for PID, COMMAND, %CPU, USER, UID, TIME+, MEM, PR, NI, VIRT, RES, SHR, S, PPID, and others. The processes listed are splunkd, tcpdump, named, vim, and python.

PID	COMMAND	%CPU	USER	UID	TIME+	MEM	PR	NI	VIRT	RES	SHR	S	PPID	OTHERS
32616	splunkd	0.3	splunk	1000	0:02.80	4.3	20	0	135356	43956	8940	S	1	2000
32695	tcpdump	0.3	splunk	72	0:02.45	0.7	20	0	28588	6872	5464	S	32680	72
590	named	0.3	named	25	7:31.12	1.8	20	0	548272	18872	5584	S	1	25
1602	vim	0.3	dan	1028	0:00.08	0.6	20	0	152828	6248	2548	S	30580	1628
1243	python	0.3	python	1028	0:00.08	0.6	20	0	152828	6248	2548	S	30580	1628

Indexing Pipeline - Metrics

- Better metrics on pipeline usage
 - Better instrumentation
 - “metrics.log” “group=dutycycle”
 - Includes management, ingest, misc types
 - The “ratio” field is a measurement of busyness (via maths)
 - Will fluctuate at first before it stabilizes
- Why do we need this? (aside from better logging of a logging product)

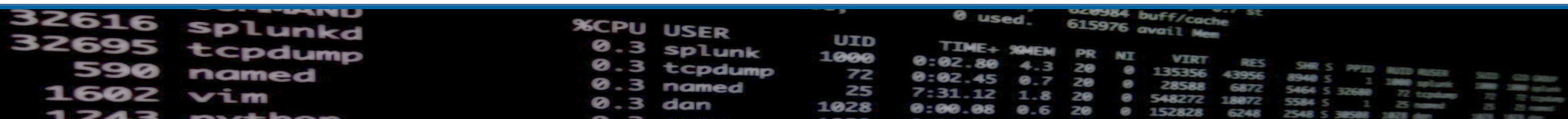


The image shows a terminal window with two distinct sections. The top section displays system metrics including CPU usage (0.3), memory usage (4.3), and various system statistics. The bottom section shows a list of processes with columns for PID, command, %CPU, USER, and UID. The processes listed include splunkd, tcpdump, named, vim, and python.

PID	COMMAND	%CPU	USER	UID
32616	splunkd	0.3	splunk	1000
32695	tcpdump	0.3	tcpdump	72
590	named	0.3	named	25
1602	vim	0.3	dan	1028
1243	python	0.3	python	1028

Indexing Pipeline – Pipeline Set Selection

- Currently just uses round-robin
- Could lead to stuffed and starved pipelines
- New server.conf config for “pipelineSetSelectionPolicy”
- “round_robin” or “weighted_random”
- “weighted_random”
 - Uses more maths
 - Should improve throughput
 - There are settings to change some of the variables on this selection process

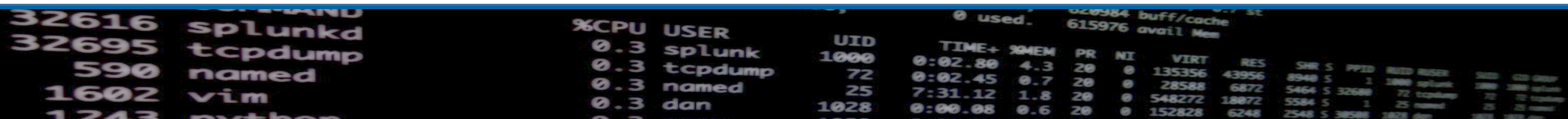


A terminal window showing system statistics and a list of processes. The top part shows memory usage: 0 used, 620984 buff/cache, 615976 avail Mem. Below that is a table of processes with columns: PID, COMMAND, %CPU, USER, UID, TIME+, MEM, PR, NI, VIRT, RES, SHR, S, PPID, PWD, USER, and COMMAND. The processes listed are splunkd, tcpdump, named, vim, and python.

PID	COMMAND	%CPU	USER	UID	TIME+	MEM	PR	NI	VIRT	RES	SHR	S	PPID	PWD	USER	COMMAND
32616	splunkd	0.3	splunk	1000	0:02.80	4.3	20	0	135356	43956	8940	S	1	/usr/sbin	splunk	splunkd
32695	tcpdump	0.3	tcpdump	72	0:02.45	0.7	20	0	28588	6872	5464	S	32680	/usr/sbin	tcpdump	tcpdump
590	named	0.3	named	25	7:31.12	1.8	20	0	548272	18872	5584	S	1	/usr/sbin	named	named
1602	vim	0.3	dan	1028	0:00.08	0.6	20	0	152828	6248	2548	S	38580	/usr/bin	vim	vim
1243	python	0.3	python	1028	0:00.08	0.6	20	0	152828	6248	2548	S	38580	/usr/bin	python	python

Metrics

- Reduced storage footprint and increased search performance on metrics indexes
- Metrics Workspace now included with Splunk Enterprise
- Also available via Splunkbase
- Added multi-series charting
- Better aggregation of common fields across indexes
- Better accessibility and localization
- Metrics rollup
 - Think “summary indexing, but for metrics”
 - Take very fine measurements, roll them up into aggregates for faster searching

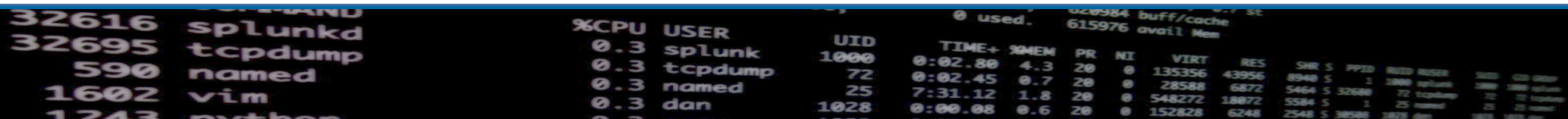


A terminal window screenshot showing system metrics and a process list. The top part of the terminal displays various system statistics including CPU usage, memory usage, and disk I/O. Below this, a table lists running processes with columns for PID, PPID, USER, and COMMAND. The processes listed include splunkd, tcpdump, named, vim, and python.

PID	PPID	USER	COMMAND
32616		splunkd	
32695		tcpdump	
590		named	
1602		vim	
1243		python	

Workload Management (WLM)

- Uses Pools
 - Get assigned CPU and memory resources
- Linux only (uses Linux cgroups under the hood)
- Prioritize searches (by app, user, type)
- Resource reservation
- System protection
- Splunkd processes run under pools
- Assignment by manual addition or by rules
- Now with a better UI



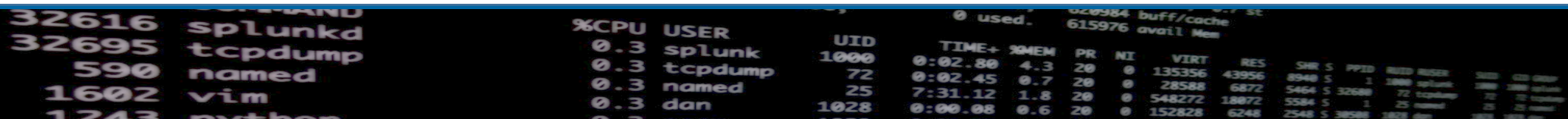
A terminal window screenshot showing system resource usage. The top part displays a list of processes with their PIDs and names. The bottom part shows a detailed table of resource usage for specific processes.

PID	Process Name
32616	splunkd
32695	tcpdump
590	named
1602	vim
1243	python

%CPU	USER	UID	TIME+	PMEM	PR	NI	VIRT	RES	SHR	S	PPID	PPID	USER	GROUP
0.3	splunk	1000	0:02.80	4.3	20	0	135356	43956	8940	S	1	1000	splunk	splunk
0.3	tcpdump	72	0:02.45	0.7	20	0	28588	6872	5464	S	72	72	tcpdump	tcpdump
0.3	named	25	7:31.12	1.8	20	0	548272	18872	5584	S	1	25	named	named
0.3	dan	1028	0:00.08	0.6	20	0	152828	6248	2548	S	30584	1028	dan	dan

Token Authentication

- On prem only
- Previously Splunk didn't have a great way for API usage (REST)
- This lead to people doing some pretty gross things
- JWT (JSON Web Tokens)
- Token gets put in the `Authorization` headers for requests
- New settings in authorize.conf for [tokens_auth]
- New role capabilities for token viewing and management

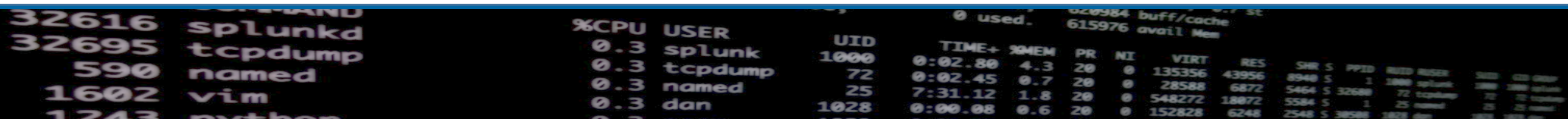


A terminal window screenshot showing system metrics and a list of processes. The top part of the terminal displays system statistics including CPU usage (0.3%), memory usage (0.3%), and various system parameters. Below this, a list of processes is shown with columns for PID, username, and process name. The processes listed include splunkd, tcpdump, named, vim, and python.

PID	USER	NAME
32616	splunkd	
32695	tcpdump	
590	named	
1602	vim	
1243	python	

LDAP

- Now has caching
- Caching has paging (lots of tweaking available)
- On prem and Cloud
- Should allow for very large LDAP queries (thousands of users/groups)

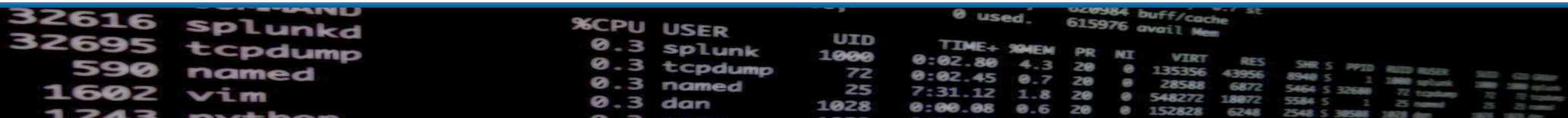


A terminal window screenshot showing system statistics and a list of processes. The top part of the image shows a list of processes with their PIDs, names, and some additional details. The bottom part shows a table with columns for %CPU, USER, UID, TIME+, MEM, PR, NI, VIRT, RES, SHR, S, PPID, and PGRP. The processes listed are splunkd, tcpdump, named, vim, and python. The system statistics show 0.3% CPU usage for splunk, tcpdump, named, and dan, with various memory and time values.

PID	NAME	%CPU	USER	UID	TIME+	MEM	PR	NI	VIRT	RES	SHR	S	PPID	PGRP
32616	splunkd	0.3	splunk	1000	0:02.80	4.3	20	0	135356	43956	8940	S	1	3000
32695	tcpdump	0.3	tcpdump	72	0:02.45	0.7	20	0	28588	6872	5464	S	32680	72
590	named	0.3	named	25	7:31.12	1.8	20	0	548272	18872	5584	S	1	25
1602	vim	0.3	dan	1028	0:00.08	0.6	20	0	152828	6248	2548	S	30580	1602
1243	python													

Time fields

- ADD_EXTRA_TIME_FIELDS in props.conf
- Used to be “true” and “false”
- Includes “date_hour, date_mday, date_minute, date_month, date_second, date_wday, date_year, date_zone, timestartpos, timeendpos, timestamp”
- Now has options
 - “none” or “false”
 - Um. None. Including sub-second info.
 - “all” or “true” (default)
 - Buddhist at a hot-dog stand
 - “subseconds”
 - None of the extra fields, but still the sub-second info



A terminal window showing system statistics and a list of processes. The top part displays memory usage: 0 used, 620984 buff/cache, 615976 avail Mem. Below is a table of processes with columns for PID, COMMAND, %CPU, USER, UID, TIME+, MEM, PR, NI, VIRT, RES, SHR, S, PPID, and others. The processes listed are splunkd, tcpdump, named, vim, and python.

PID	COMMAND	%CPU	USER	UID	TIME+	MEM	PR	NI	VIRT	RES	SHR	S	PPID	OTHER
32616	splunkd	0.3	splunk	1000	0:02.80	4.3	20	0	135356	43956	8940	S	1	2000
32695	tcpdump	0.3	tcpdump	72	0:02.45	0.7	20	0	28588	6872	5464	S	32680	72
590	named	0.3	named	25	7:31.12	1.8	20	0	548272	18872	5584	S	1	25
1602	vim	0.3	dan	1028	0:00.08	0.6	20	0	152828	6248	2548	S	30580	1602
1243	python	0.3	python	1028	0:00.08	0.6	20	0	152828	6248	2548	S	30580	1602