

# What's New in Splunk 7.3

Dave Shpritz, Aplura Splunk Practice Lead

June/July, 2019

```
020984 buff/cache
0 used, 615976 avail Mem
```

PID	PPID	%CPU	USER	UID	TIME+	MEM	PR	NI	VIRT	RES	SHR	S	PPID	RUSER	RUID	ST
32616		0.3	splunkd	1000	0:02.80	4.3	20	0	135356	43956	8948	S	1			
32695		0.3	tcpdump	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			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

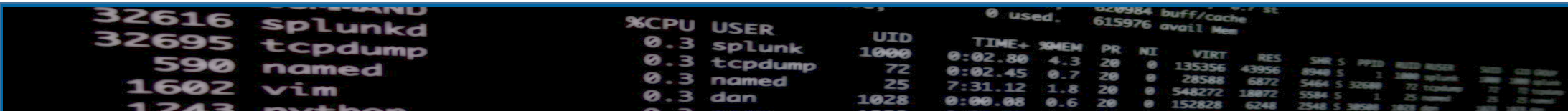


```
32616 splunkd
32695 tcpdump
590 named
1602 vim
1243 other

%CPU USER      UID      TIME+  %MEM  PR  NI  VIRT  RES  SHR  S  PPID  RSS  RSSD  RSSM  RSSV  RSSO  RSSP  RSSC  RSSD  RSSM  RSSV  RSSO  RSSP  RSSC
0.3 splunk  1000    0:02.80  4.3  20  0  135356  43956  8948  S  1  2888  2888  2888  2888  2888  2888  2888  2888  2888  2888  2888  2888  2888  2888
0.3 tcpdump  72     0:02.45  0.7  20  0  28588  6872  5464  S  1  72  72  72  72  72  72  72  72  72  72  72  72  72  72
0.3 named    25     7:31.12  1.8  20  0  548272  18872  5584  S  1  25  25  25  25  25  25  25  25  25  25  25  25  25  25
0.3 dan     1028   0:00.08  0.6  20  0  152828  6248  2548  S  1  25  25  25  25  25  25  25  25  25  25  25  25  25  25
```

# 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 showing system metrics and process information. The top part shows memory usage: 0 used, 615976 avail Mem. Below that is a table with columns: %CPU, USER, UID, TIME+, %MEM, PR, NI, VIRT, RES. The table lists processes like splunkd, tcpdump, named, vim, and python.

%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

# 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 (?)

```
0 used, 620984 buff/cache 0 used, 615976 avail Mem
```

	%CPU	USER	UID	TIME+	MEM	PR	NI	VIRT	RES	SHR	S	PPID	RSSD	RSSK	...
32616	0.3	splunkd	1000	0:02.80	4.3	20	0	135356	43956	8948	S	1	...	...	...
32695	0.3	tcpdump	72	0:02.45	0.7	20	0	28588	6872	5464	S	32689	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	38588	1828	vim	...
1243	0.3	python	...	...	...	...	...	...	...	...	S	...	...	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 of indexers
  - Rolling upgrade of indexers
- There is a timeout, so longer running searches will still be subject to kill, as will indexed real-time searches (mostly in hot, so mostly should be ok, but edge cases)

```
32616 splunkd
32695 tcpdump
590 named
1602 vim
1243 python
```

%CPU	USER	UID	TIME+	PMEM	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

# 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

```
0 used, 615976 avail Mem
%CPU USER      UID      TIME+  %MEM  PR  NI  VIRT  RES  SHR  S  PPID  RSS  RSIZE  CHILD  STATE  COMMAND
0.3 splunk    1000    0:02.80  4.3  20  0  135356  43956  8948  S  1  2888  2888  2888  2888  2888  2888  splunk
0.3 tcpdump   72     0:02.45  0.7  20  0  28588  6872  5464  S  1  72  12288  72  72  12288  72  tcpdump
0.3 named     25     7:31.12  1.8  20  0  548272  18872  5584  S  1  25  25  25  25  25  25  named
0.3 dan      1028    0:00.08  0.6  20  0  152828  6248  2548  S  1  38588  1828  400  25  25  25  dan
```

# 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

```
020984 buff/cache 0 used, 615976 avail Mem
```

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

# 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

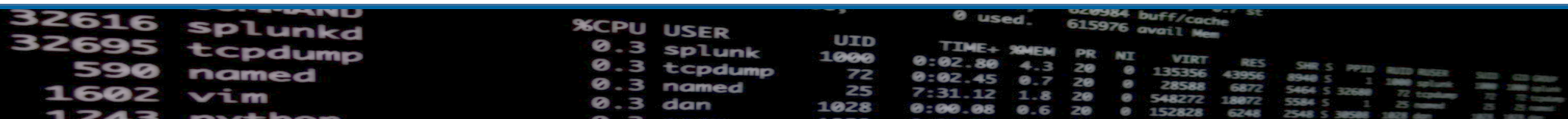
```
COMMAND
32616 splunkd
32695 tcpdump
590 named
1602 vim
1243 python

%CPU USER      UID      TIME+  %MEM  PR  NI  VIRT  RES  SHR  S  PPID  RSS  RSIZE  RMEM  ...
0.3 splunk  1000    0:02.80  4.3  20  0  135356  43956  8948  S  1  2888  2888  2888  ...
0.3 tcpdump  72     0:02.45  0.7  20  0  28588  6872  5464  S  32688  72  12288  72  ...
0.3 named    25     7:31.12  1.8  20  0  548272  18872  5584  S  1  25  25  25  ...
0.3 dan     1028   0:00.08  0.6  20  0  152828  6248  2548  S  38588  1828  4000  1828  ...
```



# 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 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 a process list on the left and system performance metrics on the right. The process list includes:

PID	Command
32616	splunkd
32695	tcpdump
590	named
1602	vim
1243	python

The system performance metrics on the right include:

%CPU	USER	UID	TIME+	MEM	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	8948	S	1	2888	splunk	2888	2888	2888	2888	2888	2888
0.3	tcpdump	72	0:02.45	0.7	20	0	28588	6872	5464	S	32688	72	tcpdump	72	72	72	72	72	72
0.3	named	25	7:31.12	1.8	20	0	548272	18872	5584	S	1	25	named	25	25	25	25	25	25
0.3	dan	1028	0:00.08	0.6	20	0	152828	6248	2548	S	38588	1028	dan	1028	1028	1028	1028	1028	1028

# 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 ☹️

```
020984 buff/cache 0 used, 615976 avail Mem
```

PPID	COMMAND	%CPU	USER	UID	TIME+	PMEM	PR	NI	VIRT	RES	SHR	S	PPID	PMEM	PR	NI	VIRT	RES	
32616	splunkd	0.3	splunk	1000	0:02.80	4.3	20	0	135356	43956	8948	S	1	2888	splunk	20	0	135356	43956
32695	tcpdump	0.3	tcpdump	72	0:02.45	0.7	20	0	28588	6872	5464	S	32688	72	tcpdump	20	0	28588	6872
590	named	0.3	named	25	7:31.12	1.8	20	0	548272	18872	5584	S	1	25	named	20	0	548272	18872
1602	vim	0.3	dan	1028	0:00.08	0.6	20	0	152828	6248	2548	S	38588	1602	vim	20	0	152828	6248
1243	python	0.3	python	1028	0:00.08	0.6	20	0	152828	6248	2548	S	38588	1243	python	20	0	152828	6248

# 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)

```
0 used, 615976 avail Mem
```

COMMAND	%CPU	USER	UID	TIME+	PMEM	PR	NI	VIRT	RES	SHR	S	PPID	RUSER	RUID	ST	TTY	TTYP	STRT	END	EXIT
32616 splunkd	0.3	splunk	1000	0:02.80	4.3	20	0	135356	43956	8948	S	1	splunk	splunk	2000					
32695 tcpdump	0.3	tcpdump	72	0:02.45	0.7	20	0	28588	6872	5464	S	32689	72	tcpdump	72					
590 named	0.3	named	25	7:31.12	1.8	20	0	548272	18872	5584	S	1	named	named	25					
1602 vim	0.3	dan	1028	0:00.08	0.6	20	0	152828	6248	2548	S	38588	1602	vim	1602					
1243 other																				

# 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

```
0 used, 615976 avail Mem
COMMAND
32616 splunkd
32695 tcpdump
590 named
1602 vim
1243 python
%CPU USER      UID      TIME+  %MEM  PR  NI  VIRT  RES  SHR  S  PPID  RSS  RSIZE  INCR  OOM  STATE
0.3 splunk  1000    0:02.80  4.3  20  0  135356  43956  8948  S  1  2888  1024  1024  0  0  0  0  0  0  0
0.3 tcpdump  72     0:02.45  0.7  20  0  28588  6872  5464  S  1  72  1024  1024  0  0  0  0  0  0  0
0.3 named    25     7:31.12  1.8  20  0  548272  18872  5584  S  1  25  1024  1024  0  0  0  0  0  0  0
0.3 dan     1028   0:00.08  0.6  20  0  152828  6248  2548  S  1  25  1024  1024  0  0  0  0  0  0  0
```



# 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

```
0 used, 615976 avail Mem
%CPU USER      UID      TIME+  %MEM  PR  NI  VIRT  RES  SHR  S  PPID  RSS  RSIZE  STATE  CWD  PID  NAME
32616 splunkd    1000     0:02.80  4.3  20  0  135356 43956 8948 S  1  2888 1000 1000 1000 1000 1000 1000
32695 tcpdump    72      0:02.45  0.7  20  0  28588  6872 5464 S  1  72  1000  1000 1000 1000 1000 1000
590  named     25      7:31.12  1.8  20  0  548272 18872 5584 S  1  25  1000  1000 1000 1000 1000 1000
1602 vim       1028    0:00.08  0.6  20  0  152828  6248 2548 S  1  25  1000  1000 1000 1000 1000 1000
1243 other
```

# 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

```
0 used, 615976 avail Mem
%CPU USER      UID      TIME+  %MEM  PR  NI  VIRT  RES  SHR  S  PPID  RSS  RSIZE  RMEM  ...
32616 splunkd   1000     0:02.80  4.3  20  0  135356 43956 8948 S  1  2888  splunk  1880  1880  splunk
32695 tcpdump   72      0:02.45  0.7  20  0  28588  6872  5464 S  1  72  tcpdump  712  712  tcpdump
590  named     25      7:31.12  1.8  20  0  548272 18872 5584 S  1  25  named    25  25  named
1602 vim       1028    0:00.08  0.6  20  0  152828  6248  2548 S  1  25  vim      25  25  vim
1243 other
```

# 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`

```
0 used, 615976 avail Mem
```

	%CPU	USER	UID	TIME+	MEM	PR	NI	VIRT	RES	SHR	S	PPID	PPID	USER	MEM	MEM	MEM
32616	0.3	splunkd	1000	0:02.80	4.3	20	0	135356	43956	8948	S	1	2888	splunk	2888	2888	splunk
32695	0.3	tcpdump	72	0:02.45	0.7	20	0	28588	6872	5464	S	32688	72	tcpdump	72	72	tcpdump
590	0.3	named	25	7:31.12	1.8	20	0	548272	18872	5584	S	1	25	named	25	25	named
1602	0.3	vim	1028	0:00.08	0.6	20	0	152828	6248	2548	S	38588	1602	vim	1602	1602	vim
1243	0.3	python															



# SHC Push Modes

- merge\_to\_default
  - Default
  - Like current behavior in < 7.2
- 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

```
020984 buff/cache 0 used, 615976 avail Mem
```

PPID	USER	%CPU	MEM	TIME+	PR	NI	VIRT	RES	SHR	S	PPID	USER	%CPU	MEM	TIME+	PR	NI	VIRT	RES	SHR	S																														
32616	splunkd	0.3		0:02.80	20	0	135356	43956	8948	S	1	3888	splunk	0.3		0:02.45	20	0	28588	6872	5464	S	32688	72	tcpdump	0.3		7:31.12	1.8	20	0	548272	18872	5584	S	1	25	named	0.3		0:00.08	0.6	20	0	152828	6248	2548	S	38588	1828	vim

# 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)...

```
32616 splunkd
32695 tcpdump
590 named
1602 vim
1243 python
```

%CPU	USER	UID	TIME+	PMEM	PR	NI	VIRT	RES	SHR	S	PPID	PPID	PPID	PPID	PPID	PPID	PPID	PPID	PPID	PPID
0.3	splunk	1000	0:02.80	4.3	20	0	135356	43956	8948	S	1	2888	splunk	2888	2888	2888	2888	2888	2888	2888
0.3	tcpdump	72	0:02.45	0.7	20	0	28588	6872	5464	S	32689	72	tcpdump	72	72	72	72	72	72	72
0.3	named	25	7:31.12	1.8	20	0	548272	18872	5584	S	1	25	named	25	25	25	25	25	25	25
0.3	dan	1028	0:00.08	0.6	20	0	152828	6248	2548	S	38588	1028	dan	1028	1028	1028	1028	1028	1028	1028

# Indexing Pipeline – Pipeline Set Selection

- 7.2 and prior 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

```
020984 buff/cache 615976 avail Mem
0 used.

PID PPID %CPU USER      UID      TIME+  %MEM  PR  NI  VIRT  RES  SHR  S  PPID  RSS  RSIZE  STATE  CWD          PID  PNAME
32616 0.3 0.00  splunkd  1000    0:02.80 4.3  20  0  135356 43956 8948 S  1  2888  2888  2888  /usr/share/splunk/bin/splunkd
32695 0.3 0.00  tcpdump  72      0:02.45 0.7  20  0  28588  6872  5464 S  1  32689  72  32689  /usr/share/splunk/bin/tcpdump
590  0.3 0.00  named    25      7:31.12 1.8  20  0  548272 18872 5584 S  1  25  25  25  /usr/sbin/named
1602 0.3 0.00  vim      1028    0:00.08 0.6  20  0  152828  6248  2548 S  1  38588  1602  1602  /usr/bin/vim
1243 0.3 0.00  python  1028    0:00.08 0.6  20  0  152828  6248  2548 S  1  38588  1243  1243  /usr/bin/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

The image shows a terminal window with two sections of output. The left section is a process list with columns for PID, Username, and Process Name. The right section is a system status report with columns for CPU usage, user, UID, Time, Mem, PR, NI, VIRT, RES, SHR, S, PPID, PGRP, RUSER, and TTY.

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

%CPU	USER	UID	TIME+	MEM	PR	NI	VIRT	RES	SHR	S	PPID	PGRP	RUSER	TTY
0.3	splunk	1000	0:02.80	4.3	20	0	135356	43956	8948	S	1	3888	splunk	
0.3	tcpdump	72	0:02.45	0.7	20	0	28588	6872	5464	S	32689	72	tcpdump	
0.3	named	25	7:31.12	1.8	20	0	548272	18872	5584	S	1	25	named	
0.3	dan	1028	0:00.08	0.6	20	0	152828	6248	2548	S	38588	1028	dan	

# 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

```
020984 buff/cache 0 used, 615976 avail Mem
```

PPID	USER	%CPU	MEM	TIME+	PR	NI	VIRT	RES	SHR	S	PPID	USER	%CPU	MEM	TIME+	PR	NI	VIRT	RES	SHR	S
32616	splunkd	0.3	4.3	0:02.80	20	0	135356	43956	8948	S	1	splunkd	0.3	4.3	0:02.80	20	0	135356	43956	8948	S
32695	tcpdump	0.3	0.7	0:02.45	20	0	28588	6872	5464	S	72	tcpdump	0.3	0.7	0:02.45	20	0	28588	6872	5464	S
590	named	0.3	1.8	7:31.12	20	0	548272	18872	5584	S	1	named	0.3	1.8	7:31.12	20	0	548272	18872	5584	S
1602	vim	0.3	0.6	0:00.08	20	0	152828	6248	2548	S	38584	vim	0.3	0.6	0:00.08	20	0	152828	6248	2548	S
1243	python	0.3										python	0.3								

# 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

```
0 used, 615976 avail Mem
```

	%CPU	USER	UID	TIME+	MEM	PR	NI	VIRT	RES	SHR	S	PPID	PPID	NAME	STATE	TIME	TIME	TIME
32616	0.3	splunkd	1000	0:02.80	4.3	20	0	135356	43956	8948	S	1	2888	splunkd	2888	2888	splunkd	
32695	0.3	tcpdump	72	0:02.45	0.7	20	0	28588	6872	5464	S	32688	72	tcpdump	2888	2888	splunkd	
590	0.3	named	25	7:31.12	1.8	20	0	548272	18872	5584	S	1	25	named	25	25	named	
1602	0.3	vim	1028	0:00.08	0.6	20	0	152828	6248	2548	S	38588	1602	vim	2888	2888	splunkd	
1243	0.3	python																

# LDAP

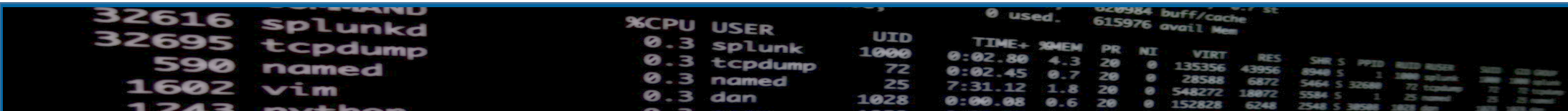
- Now has caching
- Caching has paging (lots of tweaking available)
- On prem and Cloud
- Should allow for very large LDAP queries (thousands of users/groups)

```
0 used, 615976 avail Mem
```

	%CPU	USER	UID	TIME+	PMEM	PR	NI	VIRT	RES	SHR	S	PPID	RUID	RUSER	ST	ST	ST	ST
32616	0.3	splunkd	1000	0:02.80	4.3	20	0	135356	43956	8948	S	1	2888	splunk	2888	2888	splunk	
32695	0.3	tcpdump	72	0:02.45	0.7	20	0	28588	6872	5464	S	32688	72	tcpdump	72	72	tcpdump	
590	0.3	named	25	7:31.12	1.8	20	0	548272	18872	5584	S	1	25	named	25	25	named	
1602	0.3	vim	1028	0:00.08	0.6	20	0	152828	6248	2548	S	38588	1602	vim	1602	1602	vim	
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



The image shows a terminal window with two sections of output. The top section displays system statistics, including memory usage (0 used, 615976 avail Mem) and a list of processes with their PIDs and names. The bottom section shows a table of system statistics with columns for %CPU, USER, UID, TIME+, %MEM, PR, NI, VIRT, and RES.

%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