



# Agenda

- Splunk Data Summarization Techniques
- Data Models
- Building Data Models
- Using Data Models

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

%CPU USER      UID      TIME+  %MEM  PR  NI  VIRT  RES  SHR  S  PPID  RSS  RUSR  RDSR  ST
0 used, 615976 avail Mem
0.3 splunk 1000 0:02.80 4.3 20 0 135356 43956 8940 S 1 2880 splunk 2880 2880 splunk
0.3 tcpdump 72 0:02.45 0.7 20 0 28588 6872 5464 S 32680 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
0.3 dan 1028 0:00.08 0.6 20 0 152828 6248 2548 S 38584 1828 dan 38584 1828 dan
```



# The need

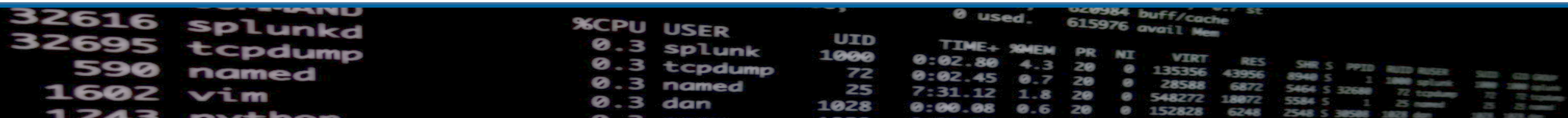
- Big data gets, well, big
- Going through all of that data for dashboards is heavy lifting
- For generating reports or dashboards over extended periods of time, it gets even bigger (read: slower)
- Needed a way to make dashboards respond faster, but still allow for the flexibility of Splunk for things like time ranges, as well as keeping data current
- Give newer users the ability to intuitively explore data and generate their own reports

```
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	8948	S	1	3888	splunk	3888	3888	3888	3888	3888	
32695	0.3	tcpdump	72	0:02.45	0.7	20	0	28588	6872	5464	S	32688	72	tcpdump	72	72	72	72	72	
590	0.3	named	25	7:31.12	1.8	20	0	548272	18872	5584	S	1	25	named	25	25	25	25	25	
1602	0.3	vim	1028	0:00.08	0.6	20	0	152828	6248	2548	S	38588	1602	vim	1602	1602	1602	1602	1602	
1243		python																		

# Ye Olde Fashion Summary Indexing

- Only take the parts of an event that you care about (reduce the volume)
- Put them in a format that is easily parsed (reduce the variety)
- Summarize the data in a manner that lends itself to calculating statistics on later (reduce the velocity)
- Accomplish this by running faster searches over a shorter period of time, to build up an index of summarized data
- This summary index will now take less effort for longer term reporting



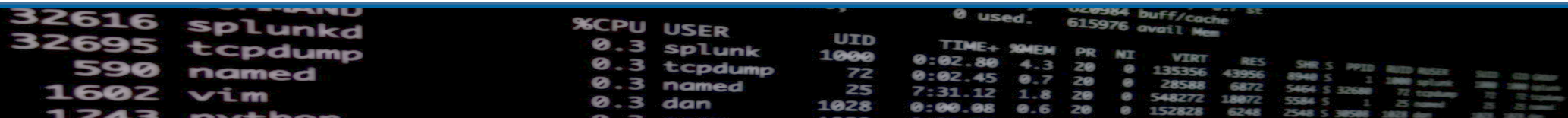
The image shows a terminal window with two sections of output. The top section lists system statistics, and the bottom section shows a process list.

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

%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

# But there were problems

- Highly dependent on search schedules
- Highly dependent on data arriving on time
- Planning ahead for report time spans
- No access controls to summarized events (only the summary index itself)
- Required running scripts at the command line to fill gaps
- Searches taking longer than the schedule range
- Writing searches for summary indexes is harder for most users

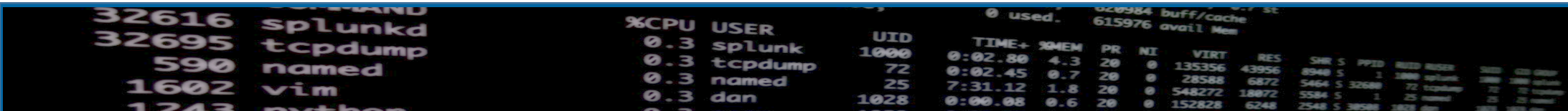


A terminal window showing system statistics and a process list. The top part shows memory usage: 0 used, 620984 buff/cache, 615976 avail Mem. Below that is a table of processes with columns for %CPU, USER, UID, TIME+, %MEM, PR, NI, VIRT, RES, and others.

%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	...

# Solutions

- Report Acceleration
  - Used to accelerate individual reports (when possible)
  - Takes care of its own scheduling
  - Takes care of its own backfill
  - The accelerated results *may* be available to other similar searches
  - Not so good for general dashboarding/reporting without having a lot of them run
  - Still requires know-how to create the original search to accelerate
- Data Model Acceleration ...



A terminal window showing system metrics and a list of processes. The top part shows memory usage: 0 used, 615976 avail Mem. Below that is a table of processes with columns for PID, CPU, USER, UID, TIME+, MEM, PR, NI, VIRT, RES, SHR, S, PPID, and other fields. 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	8948	S	1	...
32695	0.3	tcpdump	72	0:02.45	0.7	20	0	28588	6872	5464	S	32689	...
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	38588	...
1243	0.3	python											...

# Data Models

How late-binding schema helps solve the problem



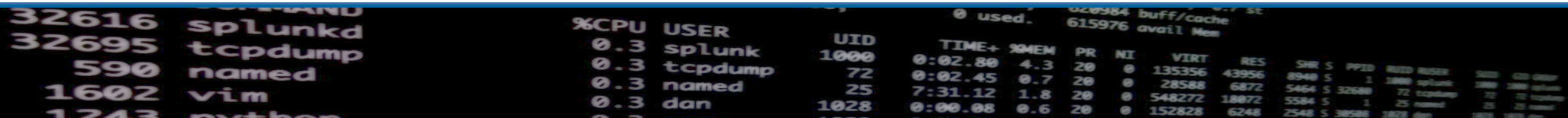
```
32616 splunkd
32695 tcpdump
590 named
1602 vim
1243 python
```

%CPU	USER	UID	TIME+	MEM	PR	NI	VIRT	RES	SHR	S	PPID	PPID	MEM	MEM	MEM	MEM	MEM	MEM	MEM	MEM
0.3	splunk	1000	0:02.80	4.3	20	0	135356	43956	8940	S	1	3880	splunk	3880	3880	splunk	3880	3880	splunk	
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	18072	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	38580	1028	dan	1028	1028	dan	1028	1028	dan	



# How data models solve (some) problems

- Provides summarization of large amounts of data with acceleration
- Can include enrichment of the data along the way
- Allow for ad-hoc acceleration (more on that later)
- Take care of their own scheduling and backfill
- Provide new (and not so new) users ways to explore data with Pivot and Datasets in a more intuitive manner
- Access to data is still limited by the underlying Splunk access controls

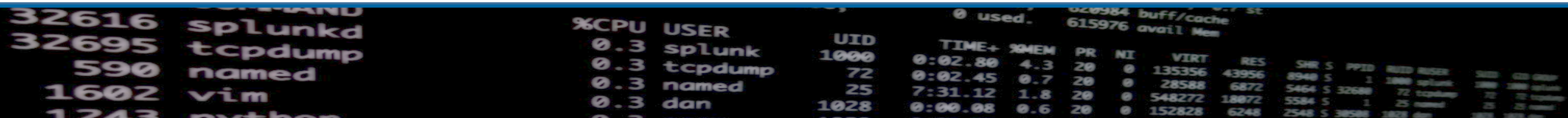


The image shows a terminal window with system statistics and a process list. The statistics include CPU usage (0.3 used), memory usage (615976 used, 615976 avail), and various system metrics. The process list shows the following:

PID	PPID	%CPU	USER	UID	TIME+	MEM	PR	NI	VIRT	RES	SHR	S	PPID	NAME
32616		0.3	splunkd	1000	0:02.80	4.3	20	0	135356	43956	8948	S	1	splunkd
32695		0.3	tcpdump	72	0:02.45	0.7	20	0	28588	6872	5464	S	32689	tcpdump
590		0.3	named	25	7:31.12	1.8	20	0	548272	18872	5584	S	1	named
1602		0.3	vim	1028	0:00.08	0.6	20	0	152828	6248	2548	S	30588	vim
1243		0.3	python											python

# Let's talk schemas

- If you know traditional databases, you may know this term
- Defines what makes up the structure of a database
- Traditional databases use early-binding or star schema
- Think “definition on write”
- Splunk uses late-binding or schema-on-read
- Data models help formalize that late-binding schema
- Data model acceleration then turns it to structured data in the “high performance analytics store”



The image shows a terminal window with two sections of output. The left section is a process list with columns for PID, COMMAND, %CPU, USER, and UID. The right section is a system statistics output with columns for TIME+, %MEM, PR, NI, VIRT, RES, SHR, S, PPID, PWD, RUSER, and RGROUP.

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	dan	1028

TIME+	%MEM	PR	NI	VIRT	RES	SHR	S	PPID	PWD	RUSER	RGROUP
0:02.80	4.3	20	0	135356	43956	8948	S	1	/usr/bin	splunk	splunk
0:02.45	0.7	20	0	28588	6872	5464	S	32689	/usr/bin	tcpdump	tcpdump
7:31.12	1.8	20	0	548272	18872	5584	S	1	/usr/bin	named	named
0:00.08	0.6	20	0	152828	6248	2548	S	38588	/usr/bin	python	python

# So what's a data model?

- A data model is made up of one or more datasets
- Datasets represent events, but have a standard set of fields for each event
- There are four types of datasets
  - Event
  - Search
  - Transaction
  - Child

Select a Table	
<a href="#">Back</a>	
<b>i</b>	23 Tables in Call Detail Records
▶	<a href="#">All Calls</a>
▶	<a href="#">Voice</a>
▶	<a href="#">SMS</a>
▶	<a href="#">Data</a>
▶	<a href="#">Roaming</a>
▶	<a href="#">All Switch Records</a>
▶	<a href="#">ATT Carrier</a>
▶	<a href="#">Metro Carrier</a>
▶	<a href="#">SWB Carrier</a>
▶	<a href="#">VER Carrier</a>
▶	<a href="#">Virgin Carrier</a>
▶	<a href="#">Conversations (1 day maxspan, 5 hours maxpause)</a>
▶	<a href="#">Outgoing Calls (1 day maxspan)</a>
▶	<a href="#">All Calls and Switches</a>

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

%CPU USER      UID      TIME+  %MEM  PR  NI  VIRT  RES  SHR  S  PPID  RSS  RUSR  RDSK  RNET  ST
0.3 splunk  1000    0:02.80  4.3  20  0  135356 43956 8940 S  1  2880  1000  1000  1000  1000  1000  1000
0.3 tcpdump  72     0:02.45  0.7  20  0  28588  6872  5464 S  1  25  25  25  25  25  25  25
0.3 named    25     7:31.12  1.8  20  0  548272 18872 5584 S  1  25  25  25  25  25  25  25
0.3 dan     1028    0:00.08  0.6  20  0  152828  6248  2548 S  1  38588  1828  400  25  25  25  25
```



# Search Datasets

- The base search can have additional fanciness
- For example, transformation to aggregate search results
- The search can be arbitrary
- Can be accelerated in some cases

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

%CPU USER      UID      TIME+  %MEM  PR  NI  VIRT  RES  SHR  S  PPID  RSS  RSD  RDSR  ST
0 used, 615976 avail Mem
0.3 splunk 1000 0:02.80 4.3 20 0 135356 43956 8940 S 1 2880 splunk 2880 2880 splunk
0.3 tcpdump 72 0:02.45 0.7 20 0 28588 6872 5464 S 32689 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
0.3 dan 1028 0:00.08 0.6 20 0 152828 6248 2548 S 38588 1828 dan 38588 1828 dan
```

# Transaction Datasets

- Use other types of datasets to form transactions
- Can't be accelerated

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

	%CPU	USER	UID	TIME+	PMEM	PR	NI	VIRT	RES	SHR	S	PPID	RUSER	RGROUP	ST	TTY	DESCRIPTORS
32616	0.3	splunkd	1000	0:02.80	4.3	20	0	135356	43956	8948	S	1	splunk	splunk	0	0	0
32695	0.3	tcpdump	72	0:02.45	0.7	20	0	28588	6872	5464	S	32680	72	tcpdump	72	72	tcpdump
590	0.3	named	25	7:31.12	1.8	20	0	548272	18872	5584	S	1	named	named	25	25	named
1602	0.3	vim	1028	0:00.08	0.6	20	0	152828	6248	2548	S	30584	1602	vim	1602	1602	vim
1243	0.3	python															

# Child Datasets

- Apply additional constraints or filtering to their parent datasets

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

PID	COMMAND	%CPU	USER	UID	TIME+	MEM	PR	NI	VIRT	RES	SHR	S	PPID	RUSER	RUID	GROUP
32616	splunkd	0.3	splunk	1000	0:02.80	4.3	20	0	135356	43956	8948	S	1	splunk	splunk	splunk
32695	tcpdump	0.3	tcpdump	72	0:02.45	0.7	20	0	28588	6872	5464	S	32680	72	tcpdump	tcpdump
590	named	0.3	named	25	7:31.12	1.8	20	0	548272	18872	5584	S	1	named	named	named
1602	vim	0.3	dan	1028	0:00.08	0.6	20	0	152828	6248	2548	S	38588	1028	dan	dan

# When Should I Use Data Models?

- For normalizing data
- For applying additional, often used, evaluations (evals) or lookups to data
- With accelerations, can provide the groundwork for great generalized reporting and dashboarding in your own application

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

PPID	PID	USER	%CPU	MEM	TIME+	PR	NI	VIRT	RES	SHR	S	PPID	PPID	USER	%CPU	MEM	TIME+	PR	NI	VIRT	RES	SHR	S
32616	32616	splunkd	0.3	4.3	0:02.80	20	0	135356	43956	8948	S	1	3888	splunk	0.3	4.3	0:02.80	20	0	135356	43956	8948	S
32695	32695	tcpdump	0.3	0.7	0:02.45	20	0	28588	6872	5464	S	32688	72	tcpdump	0.3	0.7	0:02.45	20	0	28588	6872	5464	S
590	590	named	0.3	1.8	7:31.12	20	0	548272	18872	5584	S	1	25	named	0.3	1.8	7:31.12	20	0	548272	18872	5584	S
1602	1602	vim	0.3	0.6	0:00.08	20	0	152828	6248	2548	S	38588	1602	vim	0.3	0.6	0:00.08	20	0	152828	6248	2548	S
1243	1243	python	0.3																				



# When Shouldn't I Use Data Models?

- Up-to-the-second (near real-time) results
- Small short searches
- ITSI is an example
- High performance searching (without acceleration)

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

PPID	PID	USER	%CPU	MEM	TIME+	PR	NI	VIRT	RES	SHR	S	PPID	PPID	USER	%CPU	MEM	TIME+	PR	NI	VIRT	RES	SHR	S
32616	32616	splunkd	0.3	4.3	0:02.80	20	0	135356	43956	8948	S	1	3888	splunk	0.3	4.3	0:02.80	20	0	135356	43956	8948	S
32695	32695	tcpdump	0.3	0.7	0:02.45	20	0	28588	6872	5464	S	32688	72	tcpdump	0.3	0.7	0:02.45	20	0	28588	6872	5464	S
590	590	named	0.3	1.8	7:31.12	20	0	548272	18872	5584	S	1	25	named	0.3	1.8	7:31.12	20	0	548272	18872	5584	S
1602	1602	vim	0.3	0.6	0:00.08	20	0	152828	6248	2548	S	38588	1602	vim	0.3	0.6	0:00.08	20	0	152828	6248	2548	S
1243	1243	python	0.3																				

# A Note on Acceleration

- There are limits
- Trades disk space for speed
  - High cardinality data can dramatically increase this disk space usage
- Generates acceleration summary searches
  - Can be resource intensive
- Once accelerated, you cannot edit a data model
  - Must be de-accelerated (it's a word) for editing, and then regenerated
- Accelerations are tied to the search head(s) generating them
  - This means if you want to use the same data model on multiple search heads, you need to accelerate it on each, using more resources and space



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

%CPU USER      UID      TIME+  %MEM  PR  NI  VIRT  RES  SHR  S  PPID  RSSD  RSSZ  RSSM  RSSV  RSSP  RSSC  RSSD  RSSZ  RSSM  RSSV  RSSP  RSSC
0.3 splunk  1000    0:02.80  4.3  20  0  135356  43956  8940  S  1  2888  splunk  2888  2888  splunk
0.3 tcpdump  72     0:02.45  0.7  20  0  28588  6872  5464  S  32688  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
0.3 dan     1028   0:00.08  0.6  20  0  152828  6248  2548  S  38588  1828  dan  1828  1828  dan
```

# Building Data Models



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

%CPU USER      UID      TIME+  %MEM  PR  NI  VIRT  RES  SHR  S  PPID  RSSD  RSSR  RSSZ  SZD  SZR  SZZ  DROPPED
0 used, 615976 avail Mem
0.3 splunk 1000 0:02.80 4.3 20 0 135356 43956 8940 S 1 2880 splunk 2880 2880 splunk
0.3 tcpdump 72 0:02.45 0.7 20 0 28588 6872 5464 S 32680 72 tcpdump 72 72 tcpdump
0.3 named 25 7:31.12 1.8 20 0 548272 18072 5584 S 1 25 named 25 25 named
0.3 dan 1028 0:00.08 0.6 20 0 152828 6248 2548 S 38580 1828 dan 25 25 dan
```

# We're going to talk in abstract terms

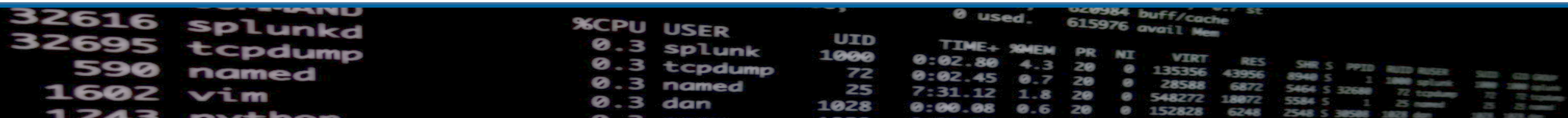
- Splunk has a great tutorial  
<http://docs.splunk.com/Documentation/Splunk/latest/PivotTutorial/>
- And a Cheat Sheet  
<https://www.splunk.com/blog/2014/02/26/data-model-cheat-sheet.html>

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

%CPU USER      UID      TIME+  %MEM  PR  NI  VIRT  RES  SHR  S  PPID  RSS  RSIZE  CHILD  OTHER
0.3 splunk  1000    0:02.80  4.3  20  0  135356  43956  8948  S  1  2888  2888  2888  2888  2888
0.3 tcpdump  72     0:02.45  0.7  20  0  28588  6872  5464  S  72  12288  72  12288  72  12288
0.3 named    25     7:31.12  1.8  20  0  548272  18872  5584  S  1  25  25  25  25  25
0.3 dan     1028    0:00.08  0.6  20  0  152828  6248  2548  S  38588  2628  4088  25  25  25
```

# You need a good base

- Prep your data
- Check your field extractions to make sure they are as good as possible
- Make your field names meaningful and consistent
  - Look at Splunk's Common Information Model (CIM) for ideas
  - If you need to create and document your own Model for your data
- Work with domain experts



The image shows a terminal window with two sections of output. The top section displays system statistics, including memory usage (615976 avail Mem) and CPU usage (0.3 used). The bottom section shows the output of the 'top' command, listing processes with columns for PID, USER, %CPU, and UID. The processes listed are splunkd, tcpdump, named, vim, and others.

PID	USER	%CPU	UID
32616	splunkd	0.3	1000
32695	tcpdump	0.3	72
590	named	0.3	25
1602	vim	0.3	1028
1243	others	0.3	

# Step Back, and think about the problem

- The cheat sheet has great guidance on this
- Look at what questions you need answered
- Look at the noun, verbs, and modifiers of those questions
- What kind of reports do you want to create? Dashboards?
- What parts of the data (events and then fields) would you need to answer those questions?

```
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

# Now go to your data

- Ask your friendly Splunk administrator for help if you need it
- Start looking at the data and creating a search that gathers the events you need
- That search will be the root event search
- You may want to encapsulate this search in an eventtype (your admin can help you with that)
- Remember to watch out for high cardinality fields!

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

%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	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	32689	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

# About your fields...

- You can use auto-extracted fields
- These are field pulled from the original events
- What if an event doesn't have that field?
- Would you like to replace them with "unknown" or some other value?
- Look at using eval based fields
- Enrich your events using lookups (sorry, that's another talk)

```
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
0.3	splunk	1000	0:02.80	4.3	20	0	135356	43956	8948	S	1	2888	aplunk	2888	2888	aplunk	2888	2888	
0.3	tcpdump	72	0:02.45	0.7	20	0	28588	6872	5464	S	32689	72	tcpdump	72	72	tcpdump	72	72	
0.3	named	25	7:31.12	1.8	20	0	548272	18872	5584	S	1	25	named	25	25	named	25	25	
0.3	dan	1028	0:00.08	0.6	20	0	152828	6248	2548	S	38588	1028	dan	1028	1028	dan	1028	1028	



# Using Data Models



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

%CPU USER      UID      TIME+  %MEM  PR  NI  VIRT  RES  SHR  S  PPID  RSSD  RSSZ  DSD  DSDR
0 used, 615976 buff/cache
615976 avail Mem
0.3 splunk 1000 0:02.80 4.3 20 0 135356 43956 8948 S 1 2880 splunk 2880 2880 splunk
0.3 tcpdump 72 0:02.45 0.7 20 0 28588 6872 5464 S 32680 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
0.3 dan 1028 0:00.08 0.6 20 0 152828 6248 2548 S 38588 1828 dan 25 25 dan
```

# Using Splunk's GUI

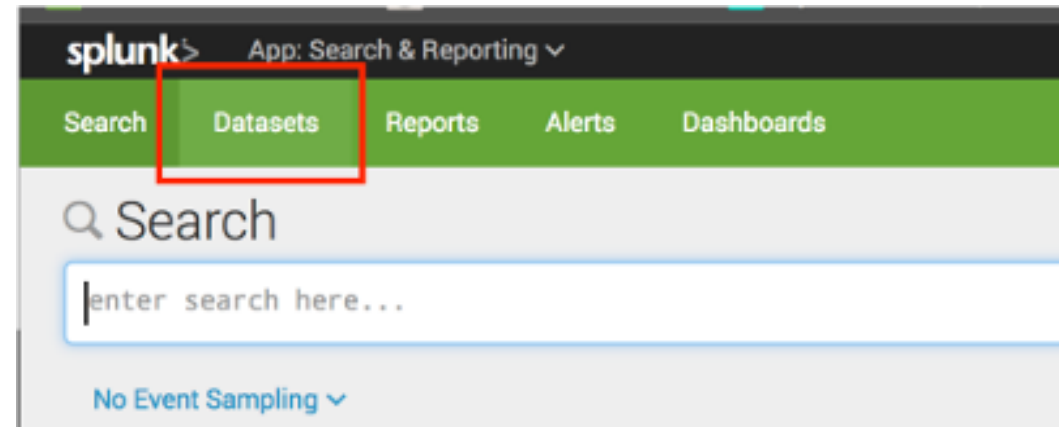
- Splunk provides different ways of looking at your data
- Datasets
- Pivot

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

	%CPU	USER	UID	TIME+	MEM	PR	NI	VIRT	RES	SHR	S	PPID	PID	PPID	USER	MEM	SHR	S
32616	0.3	splunkd	1000	0:02.80	4.3	20	0	135356	43956	8948	S	1	2880	2880	splunk	2880	2880	splunk
32695	0.3	tcpdump	72	0:02.45	0.7	20	0	28588	6872	5464	S	32680	72	72	tcpdump	72	72	tcpdump
590	0.3	named	25	7:31.12	1.8	20	0	548272	18872	5584	S	1	25	25	named	25	25	named
1602	0.3	vim	1028	0:00.08	0.6	20	0	152828	6248	2548	S	38588	1602	1602	vim	1602	1602	vim
1243	0.3	python																

# Datasets

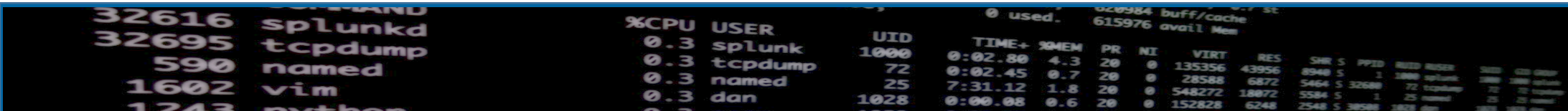
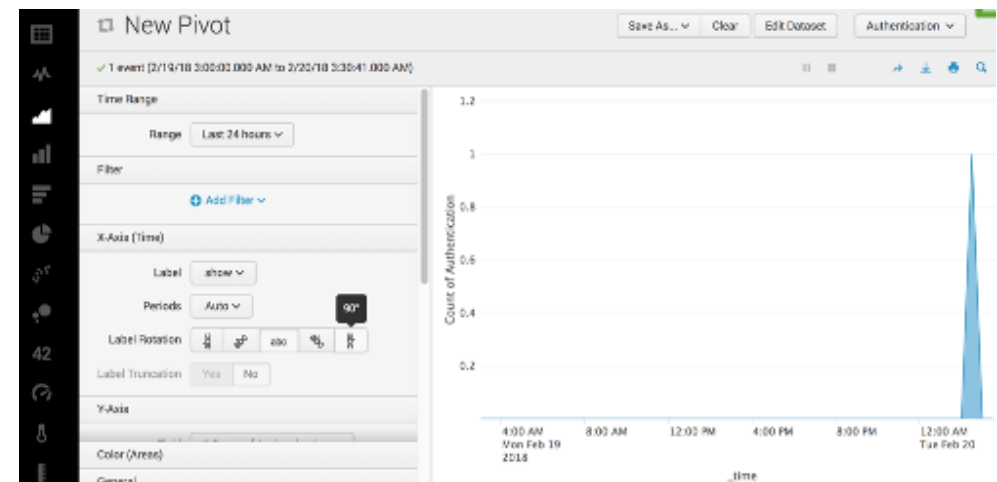
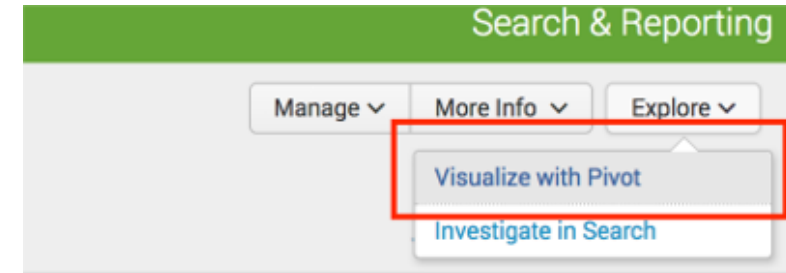
- Available in the Search and Reporting app in Splunk
- An extended version with more functionality for creating “tables” is available on Splunkbase
- Lets you look at what data is in a datamodel, including getting some summaries of the field values
- Datasets aren't limited to datamodels, lookups are datasets as well
- Leads you to opening the dataset in the Pivot interface or Search



PID	USER	%CPU	MEM	PR	NI	VIRT	RES	SHR	S	PPID	PPID	PPID	PPID	PPID	PPID	PPID	PPID	PPID	PPID
32616	splunkd	0.3		20	0	135356	43956	8948	S	1	2888	splunk	2888	2888	2888	2888	2888	2888	2888
32695	tcpdump	0.3		20	0	28588	6872	5464	S	32688	72	tcpdump	72	72	72	72	72	72	72
590	named	0.3		20	0	548272	18872	5584	S	1	25	named	25	25	25	25	25	25	25
1602	vim	0.3		20	0	152828	6248	2548	S	38588	1828	vim	1828	1828	1828	1828	1828	1828	1828
1243	python	0.3		20	0				S										

# Pivot

- Available from the Dataset explorer, or the Datamodels settings page
- Pivot lets you aggregate datasets
- You can then use these to easily create graphs and charts for inclusion in reports or dashboards



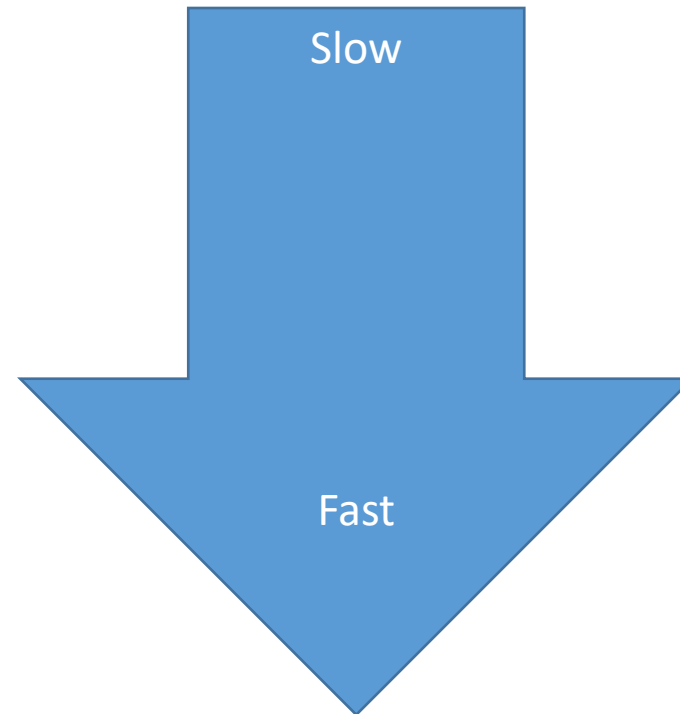
The background image shows a terminal window with system metrics. The visible text includes:

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

%CPU USER      UID      TIME+  %MEM  PR  NI  VIRT  RES  SHR  S  PPID  RSSD  RSSZ  RSSM  ...
0.3 splunk  1000    0:02.80 4.3  20  0  135356 43956 8940 S  1  2880  splunk  2880  2880  ...
0.3 tcpdump  72     0:02.45 0.7  20  0  28588  6872  5464 S  1  25  tcpdump  25  25  ...
0.3 named    25     7:31.12 1.8  20  0  548272 18872 5584 S  1  25  named    25  25  ...
0.3 dan     1028   0:00.08 0.6  20  0  152828  6248  2548 S  38588  1828  ...
```

# Using Splunk Commands

- datamodel
- from
- pivot
- tstats

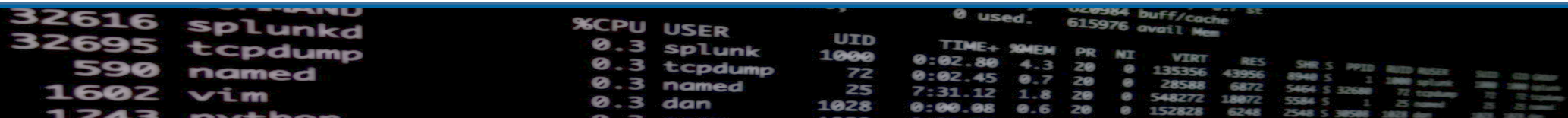


```
COMMAND
32616 splunkd
32695 tcpdump
590 named
1602 vim
1243 python

%CPU USER      UID      TIME+  %MEM  PR  NI  VIRT  RES  SHR  S  PPID  RSSZ  RUSER  ST  ST  ST  ST
0.3 splunk  1000    0:02.80 4.3  20  0  135356 43956 8940 S  1  2880 splunk  2880  2880 splunk
0.3 tcpdump  72     0:02.45 0.7  20  0  28588  6872  5464 S  1  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
0.3 dan     1028    0:00.08 0.6  20  0  152828  6248  2548 S  1  25 dan     25  25 dan
```

# The datamodel Command

- Can be used to view the JSON definition of the data model
- Usually used with the “search” option to gather events
- Works against raw data (non-accelerated)
- Returns all of the fields in the events, including the datamodel fields, prepended with their dataset title



The image shows a terminal window with two sections of output. The top section is a process list with columns for PID, COMMAND, %CPU, USER, and UID. The bottom section is a system resource usage table with columns for %CPU, USER, UID, TIME+, %MEM, PR, NI, VIRT, RES, SHR, S, PPID, and other metrics.

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	dan	1028

%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	8948	S	1	...
0.3	tcpdump	72	0:02.45	0.7	20	0	28588	6872	5464	S	32689	...
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	38588	...

# The from Command

- Also used against raw data
- Performs a little better
- Returns all fields, but does not prepend the dataset name
- Can be used against other dataset types, such as lookups

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

%CPU USER      UID      TIME+  %MEM  PR  NI  VIRT  RES  SHR  S  PPID  RSS  RUSR  RDSR  ST
0 used, 620984 buff/cache, 615976 avail Mem
0.3 splunk 1000 0:02.80 4.3 20 0 135356 43956 8940 S 1 2880 splunk 2880 2880 splunk
0.3 tcpdump 72 0:02.45 0.7 20 0 28588 6872 5464 S 32680 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
0.3 dan 1028 0:00.08 0.6 20 0 152828 6248 2548 S 38580 1828 dan 38580 1828 dan
```

# The pivot command

- More complex, mostly used by searches generated by the Pivot interface
- Has tons of options
- I've never seen one written by hand

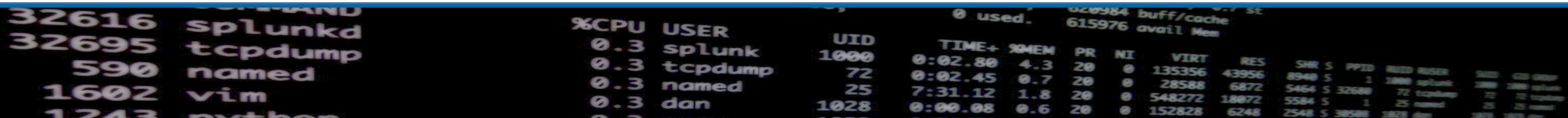
```
COMMAND
32616 splunkd
32695 tcpdump
590 named
1602 vim
1243 python

%CPU USER      UID      TIME+  %MEM  PR  NI  VIRT  RES  SHR  S  PPID  RSS  RSIZE  RUSER  ...
0.3 splunk  1000    0:02.80  4.3  20  0  135356  43956  8940  S  1  2880  splunk  2880  2880  splunk
0.3 tcpdump  72     0:02.45  0.7  20  0  28588  6872  5464  S  32680  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
0.3 dan     1028   0:00.08  0.6  20  0  152828  6248  2548  S  38584  1828  dan     25  25  dan
```



# The tstats Command

- SQL-like syntax
- Takes some getting used to
- Works against raw and accelerated events
- Can be limited to only use accelerated data by using the summariesonly flag
- Many Splunk apps which may give you example searches
- Can be used against index-time fields



The image shows a terminal window with a process list on the left and system statistics on the right. The process list includes:

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

The system statistics 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

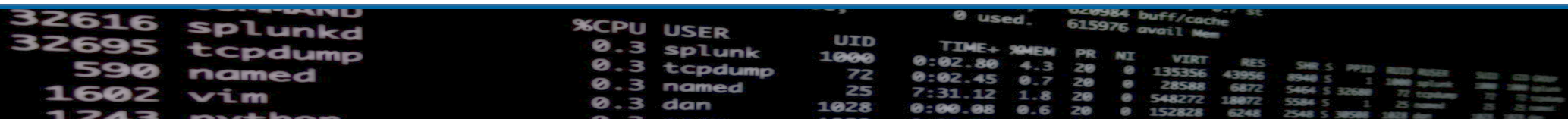
# In Summary (ha ha ha)

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

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

# Key Points

- Splunk has a few different acceleration techniques
- Data models can be useful for exploring data, even when not accelerated
- Data models can be useful for general reporting, when accelerated, even against large amounts of data. The price is resource usage
- Data models can provide beginners with a way to start working with data, but the building of the data model must be on a good base
- Splunk provides different interfaces and commands for using data models



The image shows a terminal window with system metrics and a process list. 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, 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	%CPU	USER	UID	TIME+	MEM	PR	NI	VIRT	RES	SHR	S	PPID	OTHER
32616	0.3	splunk	1000	0:02.80	4.3	20	0	135356	43956	8948	S	1	2888
32695	0.3	splunk	1000	0:02.45	0.7	20	0	28588	6872	5464	S	32688	72
590	0.3	named	72	7:31.12	1.8	20	0	548272	18872	5584	S	1	25
1602	0.3	vim	25	0:00.08	0.6	20	0	152828	6248	2548	S	38588	1828
1243	0.3	python	1028										

# Additional Resources

- Data Model and Pivot Tutorial  
<http://docs.splunk.com/Documentation/Splunk/7.0.2/PivotTutorial/WelcometothePivotTutorial>
- About data models from the Knowledge Manager Manual  
<http://docs.splunk.com/Documentation/Splunk/7.0.2/Knowledge/Aboutdatamodels>
- Accelerate data models from the Knowledge Manager Manual  
<http://docs.splunk.com/Documentation/Splunk/7.0.2/Knowledge/Acceleratedatamodels>
- Pivot Manual  
<http://docs.splunk.com/Documentation/Splunk/7.0.2/Pivot/IntroductiontoPivot>
- Using Data Models presentation  
[https://conf.splunk.com/session/2014/conf2014\\_DavidClawson\\_Splunk\\_WhatsNew.pdf](https://conf.splunk.com/session/2014/conf2014_DavidClawson_Splunk_WhatsNew.pdf)
- Using Data Sets  
<https://conf.splunk.com/files/2017/slides/using-datasets-for-easier-data-exploration-preparation-and-analysis.pdf>
- Speed Up Your Searches  
<https://conf.splunk.com/files/2017/slides/speed-up-your-searches.pdf>
- From \_raw to tstats  
<https://conf.splunk.com/files/2016/slides/how-to-scale-from-raw-to-tstats.pdf>
- Searching FAST: How to Start Using tstats and Other Acceleration Techniques  
<http://conf.splunk.com/files/2017/slides/searching-fast-how-to-start-using-tstats-and-other-acceleration-techniques.pdf>
- Lesser Known Search Commands  
<http://conf.splunk.com/files/2017/slides/lesser-known-search-commands.pdf>
- Answers  
<http://answers.splunk.com/>
- Docs  
<http://docs.splunk.com/Documentation>
- Baltimore Area User Group  
<https://usergroups.splunk.com/group/baltimore-splunk-user-group.html>
- Slack Signup  
<http://splk.it/slack>

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

%CPU	USER	UID	TIME+	%MEM	PR	NI	VIRT	RES	SHR	S	PPID	PWD
0.3	splunk	1000	0:02.80	4.3	20	0	135356	43956	8940	S	1	/usr/local/splunk/bin
0.3	tcpdump	72	0:02.45	0.7	20	0	28588	6872	5464	S	32680	/usr/local/bin
0.3	named	25	7:31.12	1.8	20	0	548272	18872	5584	S	1	/usr/sbin
0.3	dan	1028	0:00.08	0.6	20	0	152828	6248	2548	S	38588	/usr/bin