



>>

Everything Everywhere All At Once

Rajiv Gupta

AU^rOCON 2

THE NETWORK AUTOMATION CONFERENCE

Gather>>

- Simple Command Line Utility that multiplexes the output of show commands from many devices into a single file...FAST.
- Function like grep, sort, uniq, select, join, add, count, average, **gather**
- Large or small infrastructure snapshots
- Multiple technologies Network, Firewalls, Load Balancers, Servers, etc, etc,

A little More Information...

- Netmiko wrapped in a CLI with multithreading, and output formatting.

~~#Netbox #Batfish #SuzieQ #Pandas #sqlite #walloftext~~

Design Philosophy

- Do one thing as fast as possible
- Do the opposite. Simple solution for complex problems
- Use as much of the standard library as possible
- Fewest lines of code as possible
- Inspired by Alain Degreffé's ciscocmd utility
- Using Netmiko - THANK YOU KIRK BYERS!!!



Why...

- Everything - I need to know every state and configuration detail.
- Everywhere - In every location
- All At Once - Fast on demand large or small infrastructure snapshots.
- Complicated and Complex

Do we have any devices with these serial numbers, how many ports are connected on our network, can I get an inventory of all of our equipment, can you locate this device, I need a list of all of our access points, I need a custom configuration for 500 devices.



About Me



CISCO SYSTEMS



```
pip install gogather
```

```
pip install gogather
```

```
gather -h to get started
```

```
gather [-h] [-c COMMANDFILE] [-sc SINGLECOMMAND] [-o OUTFILE]
       [-u USERNAME] [-t TARGETFILE] [-st SINGLETARGET] [-p PASSWD]
       [-mt MAXTHREAD] [-en ENABLE] [-notag] [-pc PINGCHECK]
       [-datetime] [-os PLATFORM]
```

Examples:

```
gather -u username -p password -c cmdfile.txt -t targetfile.txt -o outputfile.txt
```

```
gather -u username -p password -sc "show version" -st switch1 -o outputfile.txt
```

```
gather -u username -p password -sc "show version" -t targetfile.txt
```

```
gather -u username -p password -c cmdfile.txt -st switch1 (DefaultDB name)
```

```
gather -u username -p password -c cmdfile.txt -t targetfile.txt -notag
```

pip install gogather

gather -help to get started

gather [-h] [-c COMMANDFILE] [-sc SINGLECOMMAND] [-o OUTFILE] [-u USERNAME] [-t TARGETFILE] [-st SINGLETARGET] [-p PASSWD]
[-mt MAXTHREAD] [-en ENABLE] [-notag] [-pc PINGCHECK] [-datetime] [-os PLATFORM]

optional arguments:

-h, --help	show this help message and exit
-c COMMANDFILE	Enter Command File - One Per Line
-sc SINGLECOMMAND	Enter A Single Command Enclosed in Quotes - Example "show version"
-o OUTFILE	Enter Output Log File Name - If not specified default is GatherDB...
-u USERNAME	Username
-t TARGETFILE	Host File - One Per Line
-st SINGLETARGET	Enter One Target Host Only
-p PASSWD	Enter Password
-mt MAXTHREAD	Number of simultaneous threads - maximum value is 30
-en ENABLE	Please enter the enable password only if you explicitly wish to be in enable mode
-notag	NO TAG places untagged output into separate files
-pc PINGCHECK	YOU MUST ENTER THE NAME OF A GatherDB containing the output of the "show ip interface brief" command. This option searches the GatherDB and compiles all Layer 3 Interfaces present in the "show ip interface brief" command. If you use the same filename specified in the -o option, a GatherDB will be generated with the specified filename and then the same file will be used for the ping check. You can also use this option alone to ping check an existing GatherDB.
-datetime	Date Time adds the date and time to the output tags
-os PLATFORM	Enter OS Type - Default value is cisco_ios - Other available platform values are. [SSH OPTIONS] a10, accedian, adtran_os, alcatel_aos, alcatel_sros, allied_tesis_awplus, apresia_aeos, arista_eos, aruba_os, aruba_ossswitch, aruba_procurve, avaya_ers, avaya_vsp, broadcom_icos, brocade_fastiron, brocade_fos, brocade_netiron, brocade_nos, brocade_vdx, brocade_vyos, calix_b6, cdot_cros, centec_os, checkpoint_gaia, ciena_saos, cisco_asa, cisco_ftd, cisco_ios, cisco_nxos, cisco_s300, cisco_tp, cisco_viptela, cisco_wlc, cisco_xe, cisco_xr, cloudgenix_ion, coriant, dell_dnos9, dell_force10, dell_isilon, dell_os10, dell_os6,

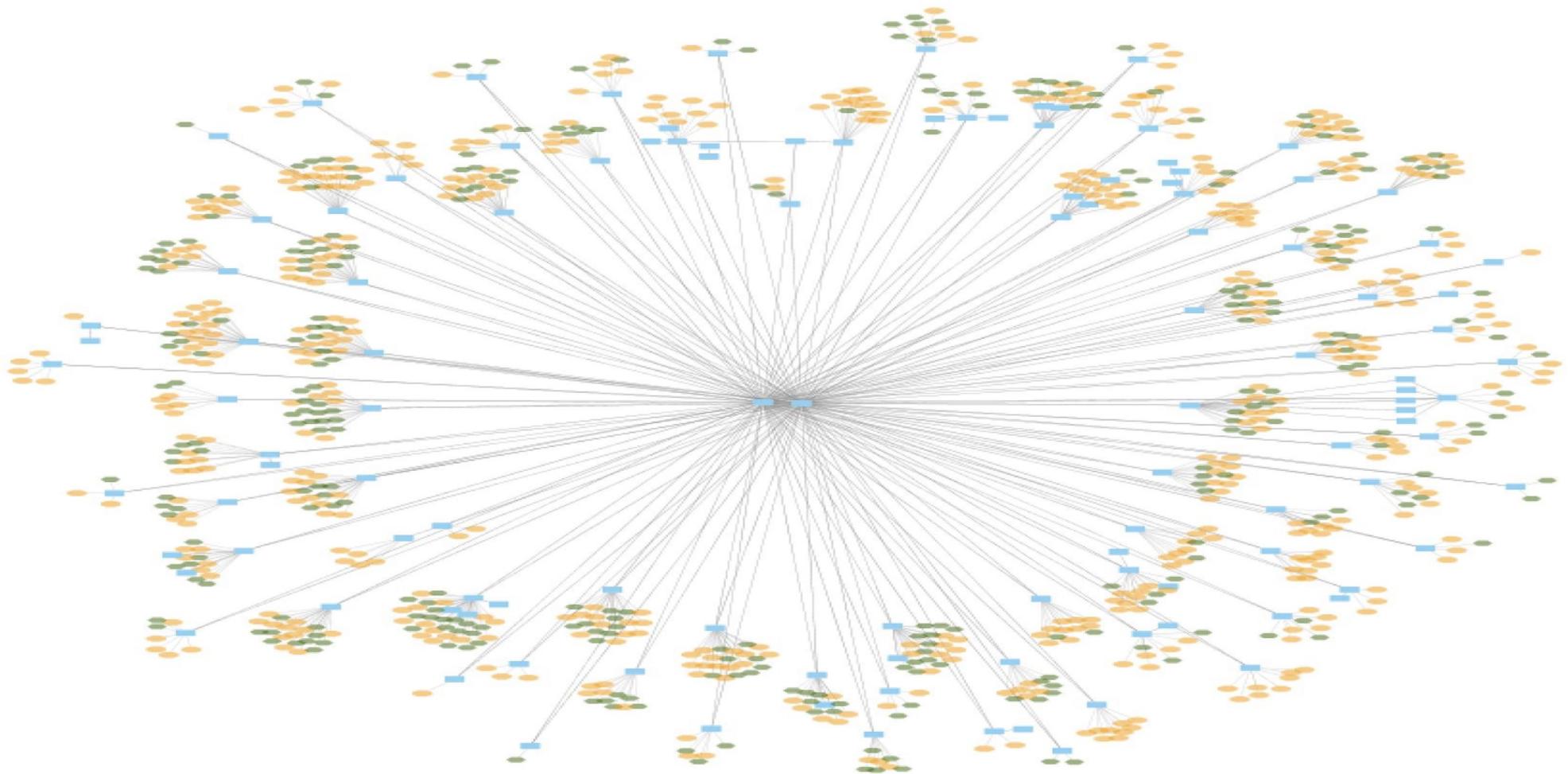
Demo Small Ad Hoc DB show version, show running - 453 devices

```
AUTOCON2 ~
AUTOCON2 ~
AUTOCON2 ~ cat command.txt
show version
show running
AUTOCON2 ~
AUTOCON2 ~ wc -l target.txt
    453 target.txt
AUTOCON2 ~
AUTOCON2 ~ gather -u `cat user.txt` -p `cat pass.txt` -t target.txt -c command.txt
```

Demo 60 commands - 453 devices

```
show logging warnings
show mac address-table
show mlag
show mlag config-sanity
show mlag detail
show module
show platform
show platform hardware chassis fantray detail
show platform hardware chassis power-supply detail all
show port-channel summary
show power
show power inline
show process cpu history
show processes memory
show snmp community
show snmp group
show spanning-tree
show system environment all
show version
show version detail
show vlan
show vxlan address-table
[AUTOCON2 ~
[AUTOCON2 ~
[AUTOCON2 ~ wc -l commands.txt
      60 commands.txt
[AUTOCON2 ~
[AUTOCON2 ~ wc -l target.txt
      453 target.txt
[AUTOCON2 ~
[AUTOCON2 ~
[AUTOCON2 ~ gather -u `cat user.txt` -p `cat pass.txt` -t target.txt -c commands.txt
```

Before



After - One Small Text File



GatherDB

GatherDB - Contents

R3	lshow interface status Port	Name	Status	Vlan	Duplex	Speed	Type	Flags	Encapsulation
R3	lshow interface status Et1		connected	routed	full	unconf	EbraTestPhyPort		
R3	lshow interface status Et2		connected	routed	full	unconf	EbraTestPhyPort		
R3	lshow interface status Et3		connected	routed	full	unconf	EbraTestPhyPort		
R3	lshow interface status Et4		connected	routed	full	unconf	EbraTestPhyPort		
R3	lshow interface status Et5		connected	1	full	unconf	EbraTestPhyPort		
R3	lshow interface status Et6		connected	1	full	unconf	EbraTestPhyPort		
R3	lshow interface status Et7		connected	F3	full	unconf	EbraTestPhyPort		
R3	lshow interface status Et8		connected	1	full	unconf	EbraTestPhyPort		
R3	lshow interface status Ma1		connected	routed	full	1G	1G/100/1000		
R3	lshow interface status								
R3	lshow interfaces counters errors Port			FCS	Align	Symbol	Rx	Runts	Giants
R3	lshow interfaces counters errors Et1			0	0	0	0	0	0
R3	lshow interfaces counters errors Et2			0	0	0	0	0	0
R3	lshow interfaces counters errors Et3			0	0	0	0	0	0
R3	lshow interfaces counters errors Et4			0	0	0	0	0	0
R3	lshow interfaces counters errors Et5			0	0	0	0	0	0
R3	lshow interfaces counters errors Et6			0	0	0	0	0	0
R3	lshow interfaces counters errors Et7			0	0	0	0	0	0
R3	lshow interfaces counters errors Et8			0	0	0	0	0	0
R3	lshow interfaces counters errors Ma1			0	0	0	0	0	0
R3	lshow ip bgp								
R3	lshow ip bgp summary								
R3	lshow ip int brief								
R3	lshow ip int brief Interface			IP Address					
R3	lshow ip int brief -----			-----					
R3	lshow ip int brief Ethernet1			1.1.1.5/31					
R3	lshow ip int brief Ethernet2			1.1.1.3/31					
R3	lshow ip int brief Ethernet3			1.1.1.8/31					
R3	lshow ip int brief Ethernet4			1.1.1.12/31					

Command Output

GatherDB - Contents

```
R3 lshow interface statusl Port
R3 lshow interface statusl Et1
R3 lshow interface statusl Et2
R3 lshow interface statusl Et3
R3 lshow interface statusl Et4
R3 lshow interface statusl Et5
R3 lshow interface statusl Et6
R3 lshow interface statusl Et7
R3 lshow interface statusl Et8
R3 lshow interface statusl Ma1
R3 lshow interface statusl

R3 lshow interfaces counters errorsl Port
R3 lshow interfaces counters errorsl E 1
R3 lshow interfaces counters errorsl E 2
R3 lshow interfaces counters errorsl E 3
R3 lshow interfaces counters errorsl E 4
R3 lshow interfaces counters errorsl E 5
R3 lshow interfaces counters errorsl E 6
R3 lshow interfaces counters errorsl E 7
R3 lshow interfaces counters errorsl E 8
R3 lshow interfaces counters errorsl Ma1

Command Issued

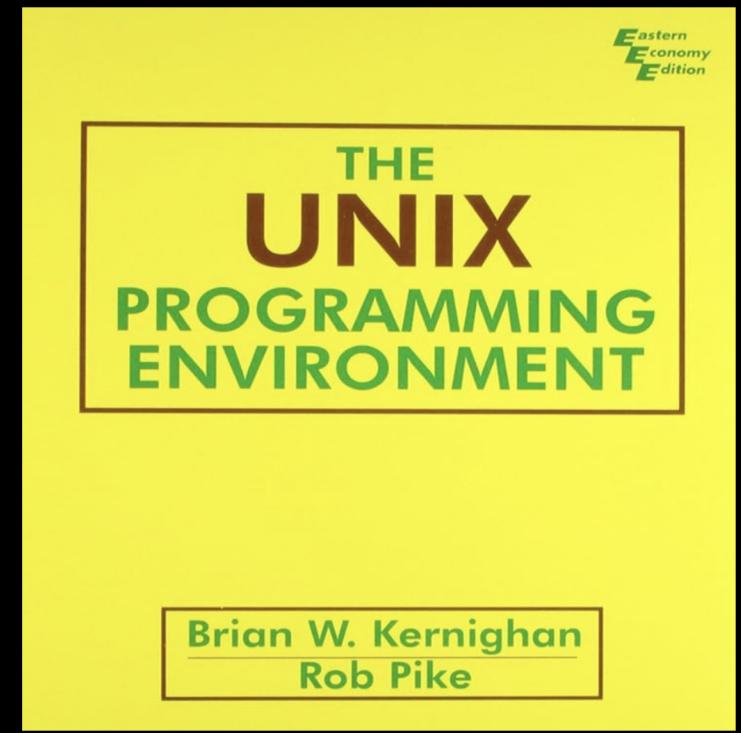
R3 lshow ip bgp summaryl
R3 lshow ip int briefl
R3 lshow ip int briefl Interface
R3 lshow ip int briefl -----
R3 lshow ip int briefl Ethernet1
R3 lshow ip int briefl Ethernet2
R3 lshow ip int briefl Ethernet3
R3 lshow ip int briefl Ethernet4
```

GatherDB - Contents

R3 show interface status| Port
R3 show interface status| Et1
R3 show interface status| Et2
R3 show interface status| Et3
R3 show interface status| Et4
R3 show interface status| Et5
R3 sh
Device Name
R3 show interface status| Et8
R3 show interface status| Ma1
R3 show interface status|
R3 show interfaces counters errors| Port
R3 show interfaces counters errors| Et1
R3 show interfaces counters errors| Et2
R3 show interfaces counters errors| Et3
R3 show interfaces counters errors| Et4
R3 show interfaces counters errors| Et5
R3 show interfaces counters errors| Et6
R3 show interfaces counters errors| Et7
R3 show interfaces counters errors| Et8
R3 show interfaces counters errors| Ma1
R3 show ip bgpl
R3 show ip bgp summary|
R3 show ip int brief|
R3 show ip int brief| Interface
R3 show ip int brief| -----
R3 show ip int brief| Ethernet1
R3 show ip int brief| Ethernet2
R3 show ip int brief| Ethernet3
R3 show ip int brief| Ethernet4

Unix Paradigm

- Simple utilities that perform specific tasks
- Combine utilities with pipes to perform complex operations
- Gather, **Observe**, Search, Count, Sort, Unique, Text Formatting, Column Formatting, Simple Loops, Simple Math



Headspace - Ten functions in your head

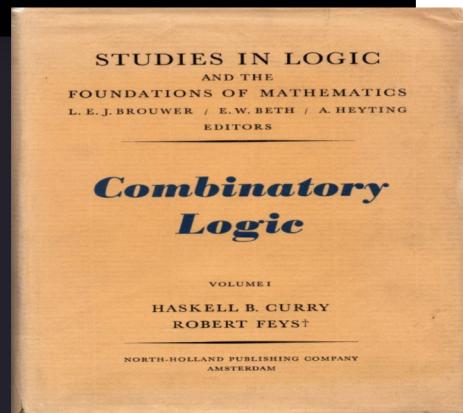
- Gather
- **Observe** - Just look at the file
- Search - grep
- Count - wc -l, uniq -c
- Sort - sort
- Unique elements - uniq, uniq -c
- Text Formatting - **sed**
- Column Formatting - **awk**
- Simple Loops - tiny **shell** loops
- Simple Math - **awk**,bc

Turing Complete

Start with Gather, Observe, Search

Functions as a tool of thought

- Use functions and combinations of functions to reason about your infrastructure.
- This idea has been around since the 60's. Before Unix



```
StDev ← 0.5 * √(+/2 * √(+-+/÷ρ)) ÷ ρ
StDev ← (+' ÷ ≠) {Fx ⊖ (Fx =) - o Fx}
StDev ← √(÷) # / + n 2 - ÷) # / +.
```

$$\lambda g. (\lambda x. g(x\ x))(\lambda x. g(x\ x))$$

1979
Turing
Award
Lecture



Notation as a Tool of Thought

KENNETH E. IVERSON
IBM Thomas J. Watson Research Center

The 1979 ACM Turing Award was presented to Kenneth E. Iverson by Walter Carlson, Chairman of the Awards Committee, at the ACM Annual Conference in Detroit, Michigan, October 29, 1979.

Simple Search - Find a mac address

Simple Search

```
AUTOCON % grep "show mac" DEMODB.txt | grep "XXXX"
```

Result

```
Device250 |show mac address-table| 213 XXXX.773a.XXXX DYNAMIC Gi2/0/1
```

Simple Search Across Time - Find a mac address

```
AUTOCON % grep "show mac" DEMODB*.txt | grep "XXXX"
```

Result

```
DEMODB_093024.txt:Device250 |show mac address-table| 213 XXXX.773a.XXXX DYNAMIC Gi2/0/1
DEMODB_100724.txt: MISSING
DEMODB_101424.txt:Device250 |show mac address-table| 213 XXXX.773a.XXXX DYNAMIC Gi2/0/1
```

Audit Config Changes -notag option

```
gather -u <user> -p <pass> -t targetfile -sc "show running" -notag
git add .
git commit -m "Post Change Commit"
git diff HEAD~1 HEAD

diff --git a/R3.txt b/R3.txt
index a7bacbb..e3f21b0 100644
--- a/R3.txt
+++ b/R3.txt
@@ -21,6 +21,7 @@ no aaa root
!
interface Ethernet1
+ description GIT DEMO
  no switchport
  ip address 1.1.1.5/31
!
```

Counting / Reporting - A little more complex

- How many wired connections on your campus total - 11004

```
grep "show interface status" DEMODB.txt | grep connected | wc -l
```

- How many ports on your campus. - 25456

```
grep "show interface status" DEMODB.txt | grep "connected|notconnect|disabled" | wc -l
```

- How many ports are connected per device

```
grep "show interface status" DEMODB.txt | grep connected | awk '{print $1}' | sort | uniq -c | sort -rn
```

snip...

206 Device447
194 Device342
194 Device261
193 Device262
186 Device332

Custom Config Changes - Down the Rabbit Hole

- Find every port on your campus with a wireless AP connected to it and configure the AP name on the port description.

```
grep WIRELESS_AP DEMODB.txt | grep ACCESSPOINT | \
awk '{print $1,$5,$6,$7}' | \
while read host ap porta portb; do \
echo "int $porta$portb \n description $ap on this port" >> "$host"-config.txt ; done
```

...

Device247-config.txt
Device250-config.txt
Device251-config.txt
Device258-config.txt
Device259-config.txt

```
more Device247-config.txt
int Ten 8/0/41
description ACCESSPOINTXX4386 on this port
int Ten7/0/26
description ACCESSPOINTXX4408 on this port
int Ten4/0/34
description ACCESSPOINTXX4382 on this port
int Ten4/0/28
description ACCESSPOINTXX4411 on this port
int Ten8/0/36
```

Custom Config Changes - Explanation

- REPL DRIVEN DEVELOPMENT - READY FIRE AIM
- One step at a time. Do something, Assess, and Repeat.
- Automation in “the small” vs “the big” - A line or two.
- The solution emerges

Custom Config Changes - Explanation

```
grep WIRELESS_AP DEMODB.txt | grep ACCESSPOINT | more
```

Device299	show cdp neighbor	ACCESSPOINTXX5316B	Ten	7/0/28	175	R T	WIRELESS_AP	Gig	0
Device299	show cdp neighbor	ACCESSPOINTXX5316A	Ten	7/0/30	136	R T	WIRELESS_AP	Gig	0
Device299	show cdp neighbor	ACCESSPOINTXX5309	Ten	7/0/31	178	R T	WIRELESS_AP	Gig	0
Device299	show cdp neighbor	ACCESSPOINTXX5304	Ten	1/0/26	146	R T	WIRELESS_AP	Gig	0
Device299	show cdp neighbor	ACCESSPOINTXX5315	Ten	7/0/29	156	R T	WIRELESS_AP	Gig	0
Device299	show cdp neighbor	ACCESSPOINTXX5161	Ten	2/0/26	173	R T	WIRELESS_AP	Gig	0

Column 1 Device Name

Column 5 AP Name

Column 6 Port

Column 7 Port

Next step format the output to print only columns needed

Custom Config Changes - Explanation

```
grep WIRELESS_AP DEMODB.txt | grep ACCESSPOINT | awk '{print $1,$5,$6,$7}' | more
```

```
Device299 ACCESSPOINTXX5316B Ten 7/0/28
Device299 ACCESSPOINTXX5316A Ten 7/0/30
Device299 ACCESSPOINTXX5309 Ten 7/0/31
Device299 ACCESSPOINTXX5304 Ten 1/0/26
Device299 ACCESSPOINTXX5315 Ten 7/0/29
Device299 ACCESSPOINTXX5161 Ten 2/0/26
```

Next step. Create the configuration for each devices

Custom Config Changes - Explanation

```
grep WIRELESS_AP DEMODB.txt | grep ACCESSPOINT |  
awk '{print $1,$5,$6,$7}' | while read host ap porta portb;  
do echo "int $porta$portb\ndescription $ap on this port" >> "$host"-config.txt ; done
```

host	ap	porta	portb
Device299	ACCESSPOINTXX5316B	Ten	7/0/28

Int Ten7/0/28

Description ACCESSPOINTXX5316B on this port >> Device299-config.txt

```
int Ten 7/0/28  
description ACCESSPOINTXX5316B on this port  
int Ten 7/0/30  
description ACCESSPOINTXX5316A on this port  
int Ten 7/0/31  
description ACCESSPOINTXX5309 on this port  
int Ten 1/0/26  
description ACCESSPOINTXX5304 on this port  
int Ten 7/0/29....
```

Recursive Searches - No Way Out

Output “show interface status” for all ports on devices running OS 17.X.X with interface module serial numbers starting with SN123.

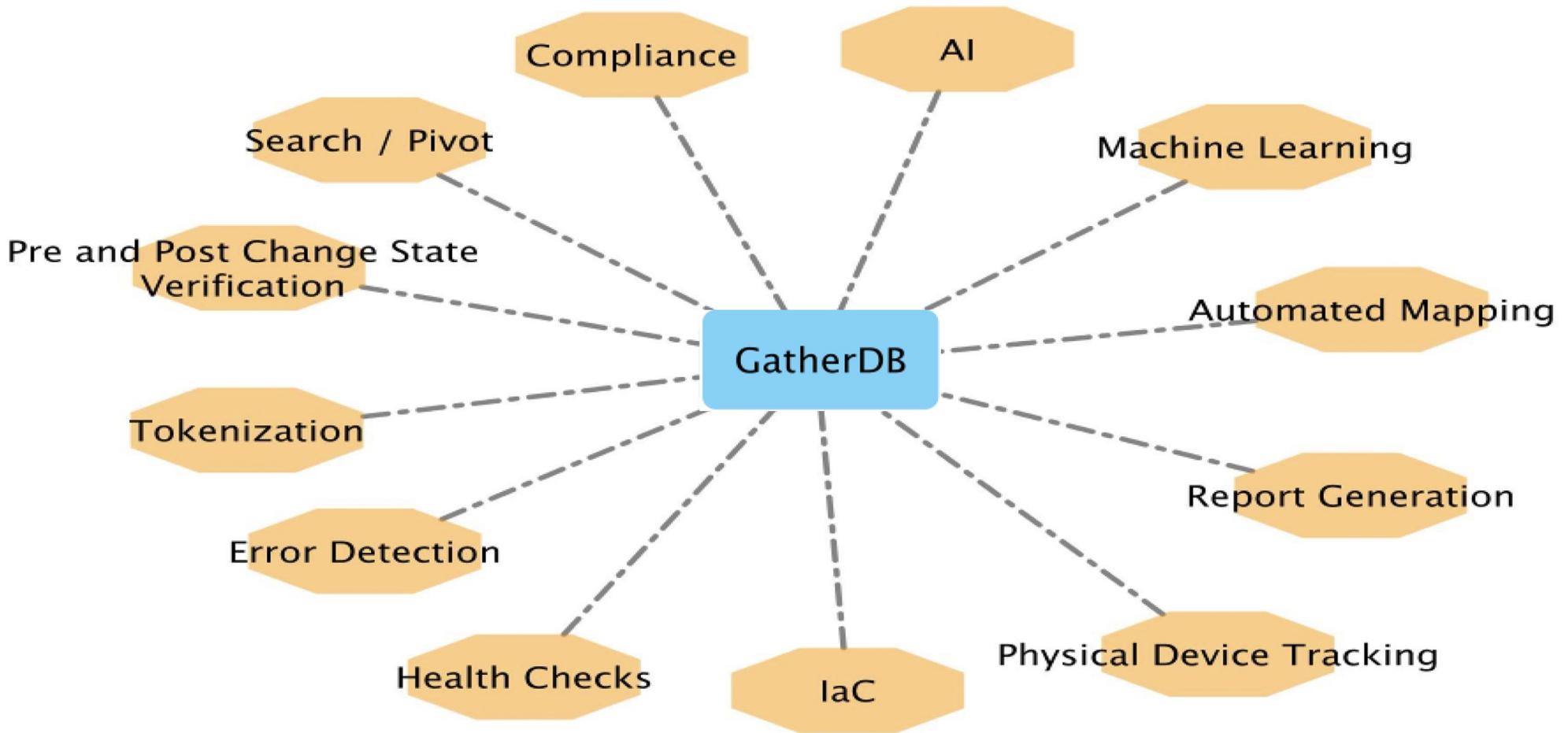
```
#Find Devices with version 17.X.X
grep "show version" DEMODB.txt | grep "Software Version" | grep "17.X.X" | awk '{print $1}' |

#From devices with version 17.X.X, Search for modules with serial numbers starting with SN123
while read host; do grep $host DEMODB.txt | grep "show module";done | grep SN123 | awk '{print $1,$4}' |

#From devices with serial number starting with SN123, Display show interface status
while read host module; do grep $host DEMODB.txt | grep "show interface status" | grep Gi$module; done
```

Complicated operations built easily step by step.

Emergent Properties of combining functions - Wonderland



Emergent Properties of combining functions - Wonderland

- Huge detailed searchable interactive maps in a few hours - Extract CDP / LLDP tables and put them into a link mapper like Cytoscape.
- Spot audit your network using meta information - Example search for NTP config on your devices and count the number of lines per device. They should all be the same.
- Have an intuitive understanding about the value and cost of vendor solutions.
- **THE TRUTH On Demand > SOT**
- **Low cognitive demand** - Reflexively turning to a set of functions or tools because they are easy and fast. (Where do you instinctively turn during an emergency?)
- Look into the past by querying archived GatherDBs. Very useful because troubleshooting a current issue because it does not show the working state. **Extremely useful for wireless.**
- **Multitechnology** - Extend your view beyond the network
- AI / ML RAG Corpus on demand

Whats Next

- Refactor from 300 lines of code to 100
- 1000 simultaneous threads - Currently 30
- Improve file handling.
- Improve error handling
- Create a binary executable version - Go, Gleam, Mojo
- Expose the netmiko config_set for configuration push and JSON
- API front end to search a GatherDB
- Work on command line utility approach to massive telemetry data sets with q and kdb+ code.kx.com

References

- pip install gogather; gather -h
- PYPI reference URL <https://pypi.org/project/gogather/>
- Youtube RANTLABS @rantlabs2982
- Github repo <https://github.com/rantlabs/RANT>
- PYPI reference URL <https://pypi.org/project/gogather/>
- REPL driven development Clojure
- The Array Cast - Amazing array programming languages podcast
- code_report polyglot coding youtube channel
- Reach out. Linkedin rajivgupta2

A black and white photograph showing a group of people from behind, their heads tilted back and hands raised towards the sky. They appear to be in a dimly lit room, possibly a theater or auditorium, with a bright light source visible in the background. The scene conveys a sense of awe, inquiry, or participation.

Questions?