LINBIT

Diagnosing Performance Bottlenecks in Production Systems

by Julia Iacoviello

Performance Issue? What to do?

- ✗ Restart system/services
- ✗ Reapply thermal paste
- ✗ Reseat components
- ✗ Blow out the case with compressed air
- ✗ Kill random processes
- ✗ Percussive maintenance
- ✗ Upgrade hardware

Julia Iacoviello
Systems Engineer
LINBIT

- Implements and supports Highly Available (HA) production environments
- High availability: Groups of servers (clusters) where one is configured to automatically switch (fail over) to a working node if there is an issue
- Many LINBIT clients have tight SLAs for service uptime

Are These *Good* Numbers?

???

```
top - 14:22:38 up 4 days, 54 min,  6 users,  load average: 0.86, 0.65, 0.67
Tasks: 462 total,   1 running, 461 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.4 us,  1.3 sy,  0.4 ni, 97.9 id,  0.0 wa,  0.0 hi,  0.0 si,  0.
MiB Mem :  64024.7 total,   767.2 free,  39476.7 used,  23780.8 buff/cache
MiB Sw     1952.0 total,   1898.2 free,     53.8 used.  22141.6 avail Mem

   PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
 280366 julia     20   0 3881384   2.4g   2.3g S   5.0   3.8  52:26.59 VBoxHeadless
 324571 julia     20   0 3338132   1.1g   1.0g     3.3   1.7  39:16.85 VBoxHeadless
 192119 julia     20   0 1131.4g 263676 123100     2.3   0.4   3:38.69 chrome
 204131 julia     20   0 3575688   1.1g   1.       2.3   1.7  52:31.33 VBoxHeadless
 124239 julia     20   0 3938736   1.7g        S   1.7   2.7  48:20.99 VBoxHeadless
 205264 julia     20   0 3639692   2.1g    0g  S   1.7   3.3  24:46.57 VBoxHeadless
 439911 julia     20   0 1133.4g 133668  1180  S   1.7   0.2   0:00.41 chrome
 195656 julia     20   0 3317652   1.     1.8g S   1.3   2.9  46:47.50 VBoxHeadless
 202935 julia     20   0 3645320 95901  912072 S   1.3   1.5  27:00.41 VBoxHeadless
 248563 julia     20   0 3873200   2.4g   2.3g S   1.3   3.8  21:37.27 VBoxHeadless
   4046 julia     20   0   32.8g 514784 229032 S   1.0   0.8  42:03.31 chrome
  90600 julia     20   0 3873204   2.4g   2.3g S   1.0   3.8  32:16.93 VBoxHeadless
 157996 julia     20   0 3873192   2.4g   2.3g S   1.0   3.8  32:23.11 VBoxHeadless
 216105 julia     20   0 3883436   2.4g   2.3g S   1.0   3.8  39:42.41 V        ss
 322225 julia     20   0 3479444 291464 246648 S   1.0   0.4            xHeadless
 339865 julia     20   0 5092560   3.0g   2.9g S   1.0   4.8     .28 VBoxHeadless
 371382 julia     20   0 5327464   3.0g   2.9g S   1.0   4.8  36:29.76 VBoxHeadless
 403177 julia     20   0 4275812   1.9g   1.8g S   1.0   3.0   3:53.63 VBoxHeadless
    397 root     -51   0       0      0      0 S   0.7   0.0  53:50.96 irq/96-DLL0945:
   2883 root      20   0 3751084 129896  82996 S   0.7   0.2  71:51.12 Xorg
 323388 julia     20   0 3331988 234648 189512 S   0.7   0.4  15:48.16 VBoxHeadless
   1884 root      20   0  276256  10944   9896 S   0.3   0.0  11:15.47 thermald
```

???

Talk Outline

- Introduction *(you are here)*
- Basic Concepts & Methodologies
- Demos 

Overhead

- The impact that *gathering the data itself* has on the system
- Different methods have more or less

```
440036 julia     20   0   13480   4560   3464 R   0.3   0.0   0:02.95 top
440057 julia     20   0 2527284 167616 104808 S   0.3   0.3   0:06.24 spectacle
440222 root      20   0       0      0      0 I   0.3   0.0   0:01.49 kworker/12:1-mm_percpu_wq
     1 root      20   0  167156  12776   8420 S   0.0   0.0   0:05.80 systemd
     2 root      20   0       0      0      0 S   0.0   0.0   0:00.24 kthreadd
     3 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_gp
     4 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_par_gp
     5 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 slub_flushwq
```

Observability

- The ability to measure the system state based on what it is *already doing*
- Can also refer to static configuration of the system
- Preferred (in most cases) for systems already deployed to production

```
julia@julia-XPS-15-9510:/$ cat /proc/pressure/cpu
some avg10=0.00 avg60=0.00 avg300=0.00 total=390403417
full avg10=0.00 avg60=0.00 avg300=0.00 total=0
julia@julia-XPS-15-9510:/$ cat /proc/pressure/memory
some avg10=0.00 avg60=0.00 avg300=0.00 total=1248900
full avg10=0.00 avg60=0.00 avg300=0.00 total=1189250
julia@julia-XPS-15-9510:/$ ifconfig wlp0s20f3
wlp0s20f3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.15  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::4971:74b0:db73:8939  prefixlen 64  scopeid 0x20<link>
        ether 4c:79:6e:d3:93:a2  txqueuelen 1000  (Ethernet)
        RX packets 14660848  bytes 18705361239 (18.7 GB)
```

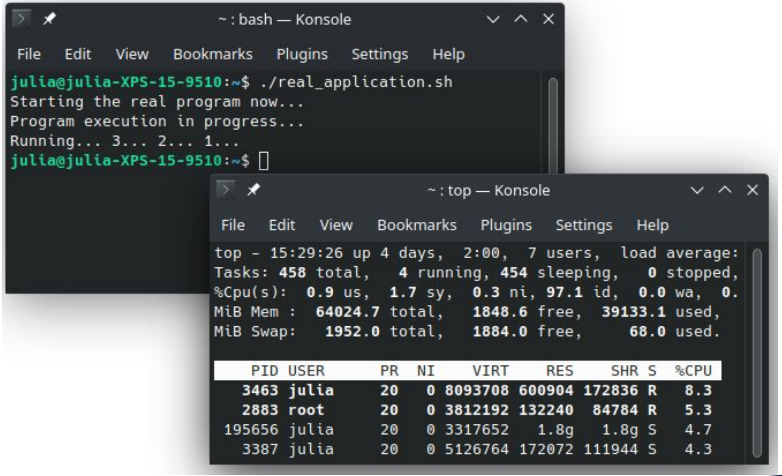
Micro-Benchmarking

- Metrics derived from a *simulated workload* applied to one component or one subset of system components
- Less *overhead* than Macro-Benchmarking

```
julia@julia-XPS-15-9510:/$ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=120 time=7.25 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=120 time=7.00 ms
^C
--- 8.8.8.8 ping statistics ---
8 packets transmitted, 2 received, 75% packet loss, time 7098ms
rtt min/avg/max/mdev = 7.000/7.123/7.247/0.123 ms
```

Macro-Benchmarking

- Applying the application workload fully as it would flow through the data path
- Run explicitly as a test to observe system metrics while it is applied



Workload Characterization

- **Determining quantitative aspects of the production workload to better simulate and observe how it performs on the system**

```
julia@julia-XPS-15-9510:~$ ./real_application.sh
Starting the real program now...
Program execution in progress...
(I am writing 512 sequential bytes to the disk, by the way...
Running... 3... 2... 1...
julia@julia-XPS-15-9510:~$
```

```
julia@julia-XPS-15-9510:~$ dd if=/dev/zero of=/dev/null obs=bytes bs=512 count=1
```

LINBIT

Performance Tuning

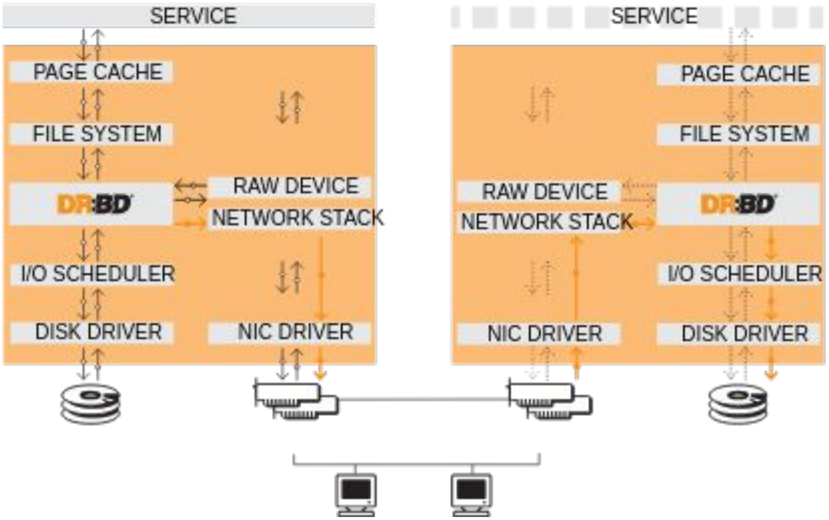
- Changing aspects of the environment and configuration with the intent to improve performance
- Especially disruptive/risky for production systems, so prior diagnostics via other methods are critical

```
root@sar-1:/home/vagrant# ps -eo pid,ppid,ni,comm | grep backup.sh
1223286 1217870  -2 backup.sh
root@sar-1:/home/vagrant# renice -n 5 1223286
1223286 (process ID) old priority -2, new priority 5
root@sar-1:/home/vagrant# 
```

The USE Method

- Developed by Brendan Gregg
- Focus on *Utilization*, *Saturation*, and *Errors* of system resources to quickly diagnose performance issues
- Learn more at www.brendangregg.com/usemethod.html
- Or his book, *Systems Performance*, 2nd ed.

Functional Block Diagram



The USE Method, Cont'd

- **Utilization**: The percentage of time the resource is doing work
- **Saturation**: The degree to which a resource has more work than it can process
- **Errors**: An discrete incident of a system not working as intended. In this case, refers to logged/loggable errors.

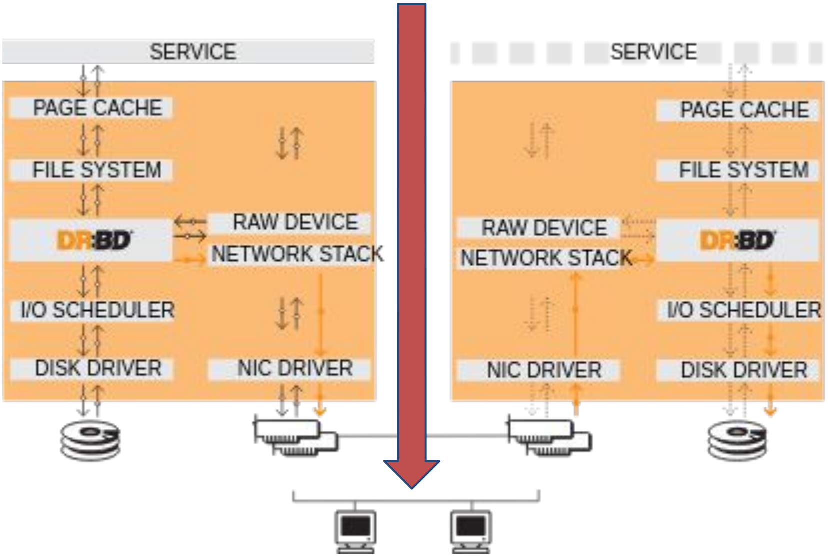
LINBIT

Demo #1

Demo #2

Demo #3

Demo #4


SERVICE
PAGE CACHE
FILE SYSTEM
DRBD
RAW DEVICE
NETWORK STACK
I/O SCHEDULER
DISK DRIVER
NIC DRIVER
SERVICE
PAGE CACHE
FILE SYSTEM
RAW DEVICE
NETWORK STACK
DRBD
I/O SCHEDULER
NIC DRIVER
DISK DRIVER

Demo #5

Demo #6

Considerations for Disk I/O Benchmarking

- **Random vs sequential**
- **Ratio of reads/writes**
- **Size of individual writes performed**
- ***Working Set Size*: How much memory is needed by the application to perform the work**
- **Flash SSDs vs rotational HDDs**
- **RAID configuration (striped, parity?)**

Disk Performance Tools & Metrics

- **iostat** - used to measure disk I/O (or determine if disks are performing I/O at all). Use with the -x flag
- **iowait** - a measure of the time CPUs are waiting for disk I/O to complete. Can be misleading!
- **smartctl** - can be used to report health metrics from disks, if supported by the disks used

LINBIT

Where to go from here?

- perf: The Linux CPU profiler. Lightweight, powerful
- eBPF: Extended Berkeley Packet Filter. Kernel technology (available since 4.4) to run programs in kernel space and has numerous use cases for observability, tracing and profiling

LINBIT

Thanks for listening!

Julia Iacoviello
julia@linbit.com