

# Linux Tools for Monitoring and Performance

Khalid Baheyeldin

November 2009

KWLUG

<http://2bits.com>

The logo for '2bits' features the number '2' in a bright green color, followed by the word 'bits' in a dark grey, stylized font. The '2' has a small green square above it. The entire logo is reflected below it on a light grey surface.



# Agenda



- Introduction
- Definitions
- Tools, with demos
  - Focus on command line, servers, web
  - Exclude GUI tools (Gnome/KDE)
- Questions, discussion





# About Khalid



- 25 years in software development and consulting
- Sinclair ZX Spectrum, mainframe, then UNIX since 1987
- Linux discovered 1990, using it regularly since 1995, “LAMP” since 1999
- Open source developer, contributor since 2003
- Full time open source consulting





# About 2bits.com



- Founded in 1999, based in Waterloo, Ontario.
- Using and contributing to Drupal since 2003
- Full time consulting
- Services
  - Drupal development
  - LAMP performance optimization and tuning
  - Server provisioning for performance and uptime
  - Manage huge sites for various clients
- <http://2bits.com>





# Definitions



- **Faster/Speed**
  - Ability to handle a given request fast, taking the least time possible
- **Scalable**
  - Ability to handle more requests per unit of time
- They are related, but not the same.
- Resource utilization matters





# powertop



- Not really performance, but just wanted to mention it ...
- Created by Intel
- Useful for laptops (maximize battery life).
- Shows “wakeups from idle” per second, preventing power savings







powertop

abits

Demo ...





top

abits

- A process, CPU, memory viewer
- Curses based
- Comprehensive (and hard to remember) options
- Use the ? to get help







top

abits

Demo ...





# htop



- A better top alternative
- Color coded
- Shows multiple CPUs
- Help on the screen (Function keys)
- More configurable (columns to show, meters, ...etc.)





htop

abits

Demo ...





# vmstat



- System wide view of resource utilization
- Processes (runnable, blocked)
- Memory (free, buffers, cache)
- Swap (swap ins, swap outs)
- I/O (blocks in, blocks out)
- Interrupts and context switches
- CPU (system, user, idle, wait for I/O)
- First row is since boot!





vmstat



Demo ... (CPU and disk)





# netstat



- Network connections
- -t tcp, -u udp, -n no DNS
- Mainly for aggregates
  - How many ESTABLISHED connections?
  - How many FIN\_WAIT/CLOSE\_WAIT

```
netstat -tunp | grep :80 | grep EST
```

```
netstat -tunp | grep :80 | grep WAIT
```







# ntop



- Daemon that reports network activity
- Opens a port, and you can connect a browser to it
- Lots of options, and lots of info





# vnstat



- Network adapter traffic (in/out)
- By hour, day, week, month
- Initialize by: `vnstat -u -i eth0`
  - Hourly: `vnstat -h`
  - Daily: `vnstat -d`
  - Monthly: `vnstat -m`
- `-tr` : sample traffic and display
- `-l` : Live real time traffic





vnstat

abits

Demo ...





# iostat



- Disk reads and writes
- Also some CPU info
- Can also report on NFS
- First row is since boot!





# iostat



- Disk characteristics (approximate)
  - 7200RPM SATA drives ~ 100 IOPS
  - 15,000RPM SAS drives ~ 180 IOPS
- Mainly for:
  - Which disk is being hit hardest?
  - Can I move stuff to another less used disk?
  - e.g. Web server: Operating system, MySQL, files, and logs





# iostat



- Example:

```
iostat -t -x 15 sda sdb
```







iostat

abits

Demo ...





# sar/atsar



- System Activity Report
- UNIX System V origin (sar)
- Samples taken at cron
- Reports produced on demand
  - Many types of reports





# atop



- Advanced Top
- Daemon as well as interactive
- Screen is loaded with info ... interpretation is the key ...





# strace



- Single process tracer
- Can trace from command line
- Can also trace running processes
- Timing option
- See what system calls are done, and what arguments are passed
  - Under the hood calls over the network, e.g. Trying to cache a javascript file for Google Analytics





# strace example



- `strace -tt -f -v -s 128 -o /tmp/k.txt -p 123`
- `tt` Timestamp, with microsecond
- `f` follow forks
- `v` verbose output (some syscalls)
- `s` show 128 bytes of strings
- `o` output file
- `p` process ID to trace





# strace output



- 17386 14:57:40.**266864** read(3, "\17\23HTTP\_USER\_AGENTw3m/0.5.1+cvs-1.968\v8HTTP\_ACCEPTtext/html, text/\*;q=0.5, image/\*, application/\*, audio/\*\24\$HTTP\_ACCEPT\_ENCODINGg"..., 749) = 749
- 17386 14:57:40.**302662** write(3, "\1\6\0\1\22\350\0\0X-Powered-By: PHP/5.2.4-2ubuntu5.7\r\nSet-Cookie: SESSd2d9a1cda25904d4196f2913e45db14b=b4c7592ffee7764d61082594f35b01d4; e"..., 4864) = 4864
- 35.79 milliseconds. Fast because Drupal's cache is enabled.





# DTrace



- Solaris technology by Sun Microsystems
- Dynamic tracing framework
- Not available for Linux





# systemtap



- DTrace like tool for Linux
- An interpreter for a special language
- Works with systemwide “probes” that you set
- Requires kernel symbols (Not Ubuntu)
- Site: <http://sourceware.org/systemtap/wiki>
- Lots of examples, e.g.
  - List Executables Reading and Writing the Most Data
  - Track Cumulative I/O Activity by Process Name





# systemtap example



Watch all opens, and print which process is opening what file

```
probe syscall.open {  
    printf ("%s(%d) open (%s)\n", execname(),  
pid(), argstr)  
}  
  
probe timer.ms(4000) { # after 4 seconds  
    exit()  
}
```





# oprofile



- Systemwide profiler
- Kernel driver + daemon for collecting data
- Does not require symbols for some things it does
- Symbols will help in certain cases (callgraphs)





# oprofile workflow



- Start profiling, excluding the kernel

```
opcontrol --no-vmlinux
```

```
opcontrol --start
```
- Do what you want profiled

```
opcontrol --shutdown
```
- View the report (next slide)
- Remember to reset the data between runs

```
opcontrol --reset
```





# oprofile



- Report full file names  
`opreport -f`
- Limit report to a minimum percentage  
`opreport -t 5`







# Valgrind & friends



- Tool for memory debugging, memory leak detection, and profiling
- Useful for C/C++ programmers for memory leaks
- Virtual Machine with Just In Time compiler
- Built for Linux and x86, but found its way elsewhere (MacOS/X, and some BSDs)
- Related tool: Cachegrind
- Related tool: Callgrind, and Kcachegrind





# Xdebug (PHP)



- PHP remote debugger
  - Local IDE or editor can interact with a program being debugged on another server
  - Over the network protocol for debugging
  - Hard to configure, but worth it for PHP apps
- Has profiling feature
  - Outputs defacto standard profiling format
  - Can be analyzed using valgrind/Kcachegrind front end





# Xdebug configuration



- For profiling, you need to add the following to your php.ini, or /etc/php5/conf.d directory:

```
xdebug.profiler_append=0
```

```
xdebug.profiler_enable_trigger=1
```

```
xdebug.profiler_enable=1
```

```
xdebug.profiler_output_dir=/tmp
```

```
xdebug.profiler_output_name=cachegrind.out
```





# Xdebug and Cachegrind **abits**

Demo ...





# MySQL processlist



- Shows queries running now and how long they take
- Shows only seconds, not milli or micro seconds

```
SHOW PROCESSLIST;
```

```
SHOW FULL PROCESSLIST;
```

```
mysqladmin processlist
```

```
mysqladmin -v processlist
```





# MySQL Processlist



Demo ...







# ApacheTop



- Reports how many requests per second Apache is doing
- Can monitor URLs, IP addresses, referrers
- Type ? for commands

```
apachetop -f /var/log/apache2/access.log
```

Options:

-T seconds





# ApacheTop



Demo ...





# Munin



- Graphical presentation of resource usage data over time
- Extensible framework, via scripts (perl, shell, python, ...)
  - Number of FastCGI PHP processes
  - Memcached customizations
- Browser based for data display
- By day, week, month, up to one year of data





# Munin monitors



- CPU (system, user, idle, wait for I/O)
- Disk
- Memory, Swapping
- Context switches, interrupts
- Network
- Load average





# Munin monitors



- Postfix / Exim
- Apache (number processes, bytes, ...etc.)
- MySQL (number of queries, connections, bytes)
- Many in your distro's package
- More on their web site
- Custom
  - Memcached
  - PHP processes





# Munin



Demo ...







- A web log analysis tool, e.g. Apache's access log
- Shows many things
  - Hits (every HTTP request, whether .css, .png, .gif, .ico, .jpg, ...etc.)
  - Page views
  - Unique visitors
  - 404s
  - Search engine keywords





- Top requesting IP addresses (bots, leechers, ...etc.)
- Operating system
- Browser information
- By month, day
- Keeps history
- Bandwidth usage





# Awstats



Demo ...





# Google Analytics



- Similar to Awstats in purpose, but operates differently and shows different results
- Uses Javascripts, so shows less figures
  - Mainly humans
  - Excludes bots, crawlers
- More marketing oriented





# Google Analytics



Demo ...





# Benchmarking



- Web benchmarking
  - Apache Bench (ab)
  - Siege
  - httpperf
    - autobench (wrapper around httpperf)
  - OpenWebLoad
  - Jmeter (Java application, with a GUI)







# Other Benchmarking



- “Quick and dirty”
  - For CPU hogging

```
bzip2 -c < /dev/urandom > /dev/null
```

- For Disk I/O hogging

```
cd /; grep -r blah .
```





# Conclusion



- Questions?
- Comments?
- Discussions?

