datto

ZFS Debugging Techniques

Tom Caputi

Userspace Lockups: What they look like

- **Process appears to be stuck**
  - **Process CAN be terminated with Ctrl + C**
  - **`top` / `htop` / `ps aux`**
    - **Process is not in `D` state**
    - **Process may just be slow**
  - **`strace`**
    - **shows no system calls are occurring**

datto

Userspace Lockups: What causes them

- **Process is usually waiting for something else**
  - **A response from another thread?**
  - **More data from `stdin`?**
  - **More buffer room `stdout`?**
  - **A rogue call to `sleep()`**
- **Process may just be doing a lot of work**
  - **Confirm with `top / htop`**

datto

Userspace Crashes: What they look like

- **Process ends abruptly**
  - **Process usually prints a message about `SIGSEGV`**
  - **Core file may be dumped (depending on `ulimit -c`)**
    - **Core files can be debugged / inspected with `gdb`**
  - **System GUI may ask if you want to submit a bug report**
    - **You will definitely click “no”**

Userspace Crashes: What causes them

- **Any kind of programmer error**
  - **`NULL` pointer dereference**
  - **Divide by zero**
  - **`assert` triggered**
- **Failure to allocate enough memory**
- **Signal received**

Kernel Crashes: What they look like

- **Process appears to be stuck**
  - **Process CAN NOT be terminated with Ctrl + C**
  - **Process IS (usually) in `D` state**
  - **Process may print `Killed` before becoming unresponsive**
  - **`strace` shows no system calls are occuring**
  - **specific info WILL appear in `dmesg`**
  - **system may become completely unresponsive**

Kernel Crashes: What causes them

- **Any kind of programmer error (similar to in userspace)**
  - **`NULL` pointer dereference**
  - **Divide by zero**
  - **`ASSERT` triggered**
- **Kernel cannot fail to allocate memory (in theory)**
- **Kernel cannot receive signals**

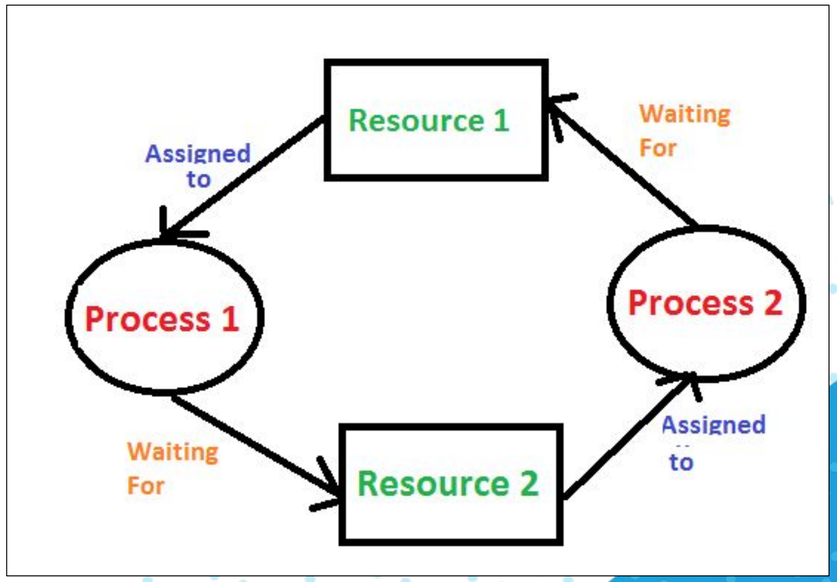
Kernel Lockups: What they look like

- **Process appears to be stuck (similar to crash)**
  - **Process CAN NOT be terminated with Ctrl + C**
  - **Process IS (usually) in `D` state**
  - **Process WILL NOT print `Killed` before becoming unresponsive**
  - **`strace` shows no system calls are occuring**
  - **specific info WILL NOT appear in `dmesg`**
  - **if dmesg isn't helpful check `/proc/<pid>/stack`**

Kernel Lockups: What causes them

<3 Wikipedia

- Process may have crashed
  - Kernel Oops
  - ZFS `ASSERT` / `VERIFY`
  - Calls to `panic()`
- Process may be waiting for something else
  - A rogue call to `msleep()`
  - A response from another thread
    - That thread may not exist anymore
    - That thread may be waiting on us! (deadlock)



datto

Performance Issues: What they look like

- **Process isn’t moving as quickly as you would like**
  - **Process may be bottlenecked by**
    - **CPU: check `top` / `htop`**
    - **RAM: check `top` / `htop` / `free -m`**
    - **Disk IO: check `iostat -mx 1` / `iotop`**
    - **Network IO: check `iftop`**
    - **Another Process: check for other slow processes**
    - **Something else?**

datto

Performance Issues: Finding the Culprit

- **CPU Bottlenecks**
  - **`perf top`: find functions using the most CPU**
  - **`FlameGraph`: analyze how CPU time is spent**
- **Memory bottlenecks**
  - **Usually indicates a memory leak**
  - **ZFS can be built with memory debugging!**

Our Lord and Savior, Brendan Gregg

datto

Performance Issues: Finding the Culprit

- **Disk Bottlenecks**
  - `zpool iostat [-lrw] [-v] 1`: **info about IO size, latency, queuing**
  - `/proc/spl/kstat/zfs/arcstats`: **stats about the ARC**
  - `arcstat.py`: **summarized info from above**
  - `iotop`: **info about which processes are issuing IO**
- **Network bottlenecks**
  - **Send less data over the network**
  - **Do more compression or buy faster networking**

Performance Issues: Finding the Culprit

- **Something else?**
  - **`bpftrace`: print info about kernel function calls**
  - **`funcgraph`: observe how all functions are called**
  - **`/proc/spl/kstat/zfs/dbgmsg`: debug messages from the kernel**

Resources

datto

Where to Get These Tools

- **Most of these tools can just be `apt installed`**
- `perf-tools`: https://github.com/brendangregg/perf-tools.git
- `FlameGraph`: https://github.com/brendangregg/FlameGraph.git
- `bpftrace`: https://github.com/iovisor/bpftrace.git
  - **requires newer kernels for all features (4.15 recommended)**
- **Dump deduplicated stack traces of all processes:**
  - ```
    md5sum /proc/*/stack | \
    sort -k1 | \
    uniq -w32 -c | \
    sort -n | \
    awk '{print $0; system("cat " $3); print "--" }'
    ```

<3 Sri Ramanujam

datto
    ```

Questions?

datto