OS X 10.9.3 Recurring Panics

23 May 2014

I give up. After upgrading to OS X 10.9.3, I've experienced the least reliable computer of my life. I've had 16 kernel panics in 4 days, usually when plugging in remote displays, but sometimes just spontaneously.

In a meeting yesterday, a coworker exclaimed how reliable and awesome 10.9.3 was, and that he didn't have any kernel panics at all. I fantasized that his macbook spontaneously panic'd as he said that, proving him wrong in theatrical fashion, preferably with a loud bang and smoke. (It would also show that it wasn't just me.)

I settled for some debugging to see why his macbook was different, but he shooed me away from his keyboard as he was about to do an IMPORTANT DEMO, and didn't want me to jinx it. He plugged in a data projector.

Instant kernel panic.

He was now the third co-worker to experience this after upgrading to 10.9.3.

tl;dr: Jump to Update 8 for the workaround.

OS X saves diagnostic reports for each panic in /Library/Logs/DiagnosticReports. Note the dates:

```
/Library/Logs/DiagnosticReports> ls -1 Kernel_2014-05-2*
Kernel_2014-05-20-120403_lgml-bgregg.panic
Kernel_2014-05-20-120755_lgml-bgregg.panic
Kernel_2014-05-20-120907_lgml-bgregg.panic
Kernel_2014-05-20-141252_lgml-bgregg.panic
Kernel_2014-05-20-154123_lgml-bgregg.panic
Kernel_2014-05-20-154336_lgml-bgregg.panic
Kernel_2014-05-20-160644_lgml-bgregg.panic
Kernel_2014-05-20-161112_lgml-bgregg.panic
Kernel_2014-05-20-181513_lgml-bgregg.panic
Kernel_2014-05-22-005541_lgml-bgregg.panic
Kernel_2014-05-22-111027_lgml-bgregg.panic
Kernel_2014-05-22-112412_lgml-bgregg.panic
Kernel_2014-05-22-140044_lgml-bgregg.panic
Kernel_2014-05-22-155309_lgml-bgregg.panic
Kernel_2014-05-23-095455_lgml-bgregg.panic
Kernel_2014-05-23-103228_lgml-bgregg.panic
```

These can be useful to browse for patterns. Maybe it's just one application which I can stop using?

```
/Library/Logs/DiagnosticReports> sed -n '/BSD/s/.*thread: //p' Kernel_2014-05-2*
Google Chrome He
Dock
Google Chrome He
launchd
Terminal
WindowServer
WindowServer
Google Chrome
SophosManagement
update_dyld_shar
WindowServer
Dock
mdworker
UserEventAgent
Adium
nsupdate
```

No, it's completely random. This looks a lot like bad or misseated DRAM, so our helpdesk performed what they called a "**shell swap**". This means they replace everything except the SSD. Or put differently, they take out the SSD and put it in a "known to be good" macbook pro, to see if the panics continue.

They continued.

Here's an example report:

```
Anonymous UUID:       D8327F80-E08D-3888-3350-4270EEFAB36C

Fri May 23 10:32:28 2014
panic(cpu 0 caller 0xffffff8007cdbf5e): Kernel trap at 0xffffff8007ca6968, type 14=page fault, re
CR0: 0x0000000080010033, CR2: 0x000000000000007a, CR3: 0x00000000465ce071, CR4: 0x00000000001606e
RAX: 0x0000000000000000, RBX: 0xffffff8028576630, RCX: 0x0000000000000000, RDX: 0x0000000000000000
RSP: 0xffffff81f295b9d0, RBP: 0xffffff81f295ba00, RSI: 0x0000000000001000, RDI: 0xffffff802857663
R8:  0x0000000000000000, R9:  0x0000000000001000, R10: 0x0000000000000000, R11: 0x0000000000000024
R12: 0x0000000000000000, R13: 0x0000000000000000, R14: 0x0000000000000000, R15: 0xffffff802857663
RFL: 0x0000000000010246, RIP: 0xffffff8007ca6968, CS:  0x0000000000000008, SS:  0x0000000000000001
Fault CR2: 0x000000000000007a, Error code: 0x0000000000000000, Fault CPU: 0x0

Backtrace (CPU 0), Frame : Return Address
0xffffff81f295b660 : 0xffffff8007c22fa9
0xffffff81f295b6e0 : 0xffffff8007cdbf5e
0xffffff81f295b8b0 : 0xffffff8007cf3456
0xffffff81f295b8d0 : 0xffffff8007ca6968
0xffffff81f295ba00 : 0xffffff8007c6cf58
0xffffff81f295bb90 : 0xffffff8007dd2ae3
0xffffff81f295bbf0 : 0xffffff8007e1568d
0xffffff81f295bcf0 : 0xffffff8007f6b6fa
0xffffff81f295bd80 : 0xffffff8007dfdd21
0xffffff81f295be00 : 0xffffff8007df38d5
0xffffff81f295be50 : 0xffffff8007ff1cfe
0xffffff81f295bef0 : 0xffffff8007ff1e79
0xffffff81f295bf50 : 0xffffff8008040653
0xffffff81f295bfb0 : 0xffffff8007cf3c56

BSD process name corresponding to current thread: nsupdate

Mac OS version:
13D65

Kernel version:
Darwin Kernel Version 13.2.0: Thu Apr 17 23:03:13 PDT 2014; root:xnu-2422.100.13~1/RELEASE_X86_64
Kernel UUID: ADD73AE6-88B0-32FB-A8BB-4F7C8BE4092E
Kernel slide:     0x0000000007a00000
Kernel text base: 0xffffff8007c00000
System model name: MacBookPro11,2 (Mac-3CBD00234E554E41)

System uptime in nanoseconds: 2243382927147
last loaded kext at 2230133368380: com.apple.driver.AppleUSBCDC 4.2.1b5 (addr 0xffffff7f89d04000,
last unloaded kext at 1237316107990: com.apple.driver.AppleIntelMCEReporter 104 (addr 0xffffff7f8
loaded kexts:
com.apple.driver.AppleUSBCDC    4.2.1b5
com.apple.driver.AppleUSBOHCI   656.4.1
com.apple.driver.AppleIntelMCEReporter  104
com.apple.filesystems.smbfs 2.0.2
com.apple.filesystems.autofs    3.0
com.apple.driver.AppleUpstreamUserClient    3.5.13
com.apple.iokit.IOBluetoothSerialManager    4.2.4f1
com.apple.driver.AppleGraphicsDevicePolicy  3.5.26
com.apple.driver.AudioAUUC  1.60
com.apple.driver.ApplePlatformEnabler   2.0.9d1
com.apple.driver.AGPM   100.14.15
com.apple.driver.X86PlatformShim    1.0.0
com.apple.iokit.IOUserEthernet  1.0.0d1
com.apple.driver.AppleHDA   2.6.1f2
com.apple.Dont_Steal_Mac_OS_X   7.0.0
com.apple.driver.AppleHWAccess  1
com.apple.driver.AppleIntelHD5000Graphics   8.2.6
com.apple.driver.AppleUSBDisplays   360.8.14
```

com.apple.driver.AppleUSBDisplays 360.8.14
com.apple.driver.AppleSMCLMU 2.0.4d1
com.apple.driver.AppleLPC 1.7.0
com.apple.driver.AppleCameraInterface 4.26.0
com.apple.driver.AppleThunderboltIP 1.1.2
com.apple.driver.AppleIntelFramebufferAzul 8.2.6
com.apple.iokit.BroadcomBluetoothHostControllerUSBTransport 4.2.4f1
com.apple.driver.AppleBacklight 170.3.5
com.apple.driver.AppleMCCSControl 1.1.12
com.apple.driver.AppleUSBTCButtons 240.2
com.apple.driver.AppleUSBTCKeyboard 240.2
com.apple.driver.AppleUSBCardReader 3.4.1
com.apple.AppleFSCompression.AppleFSCompressionTypeDataless 1.0.0d1
com.apple.AppleFSCompression.AppleFSCompressionTypeZlib 1.0.0d1
com.apple.BootCache 35
com.apple.driver.XsanFilter 404
com.apple.iokit.IOAHCIBlockStorage 2.5.1
com.apple.driver.AppleAHCIPort 3.0.0
com.apple.driver.AppleUSBHub 666.4.0
com.apple.driver.AppleUSBEHCI 660.4.0
com.apple.iokit.AppleBCM5701Ethernet 3.8.1b2
com.apple.driver.AppleFWOHCI 5.0.2
com.apple.driver.AirPort.Brcm4360 831.21.63
com.apple.driver.AppleUSBXHCI 677.4.0
com.apple.driver.AppleSmartBatteryManager 161.0.0
com.apple.driver.AppleRTC 2.0
com.apple.driver.AppleACPIButtons 2.0
com.apple.driver.AppleHPET 1.8
com.apple.driver.AppleSMBIOS 2.1
com.apple.driver.AppleACPIEC 2.0
com.apple.driver.AppleAPIC 1.7
com.apple.nke.applicationfirewall 153
com.apple.security.quarantine 3
com.apple.kext.triggers 1.0
com.apple.iokit.IOSerialFamily 10.0.7
com.apple.iokit.IOBluetoothFamily 4.2.4f1
com.apple.driver.DspFuncLib 2.6.1f2
com.apple.vecLib.kext 1.0.0
com.apple.iokit.IOSurface 91.1
com.apple.driver.AppleHDAController 2.6.1f2
com.apple.iokit.IOHDAFamily 2.6.1f2
com.apple.driver.X86PlatformPlugin 1.0.0
com.apple.driver.IOPlatformPluginFamily 5.7.0d11
com.apple.AppleGraphicsDeviceControl 3.5.26
com.apple.iokit.IOAcceleratorFamily2 98.20
com.apple.driver.AppleSMC 3.1.8
com.apple.driver.AppleUSBAudio 2.9.5f4
com.apple.iokit.IOAudioFamily 1.9.7fc2
com.apple.kext.OSvKernDSPLib 1.14
com.apple.iokit.IOBluetoothHostControllerUSBTransport 4.2.4f1
com.apple.iokit.IOFireWireIP 2.2.6
com.apple.driver.AppleGraphicsControl 3.5.26
com.apple.driver.AppleBacklightExpert 1.0.4
com.apple.iokit.IONDRVSupport 2.4.1
com.apple.driver.AppleSMBusController 1.0.11d1
com.apple.iokit.IOGraphicsFamily 2.4.1
com.apple.driver.AppleUSBMultitouch 240.9
com.apple.iokit.IOUSBHIDDriver 660.4.0
com.apple.iokit.IOSCSIBlockCommandsDevice 3.6.6
com.apple.iokit.IOUSBMassStorageClass 3.6.0
com.apple.iokit.IOSCSIArchitectureModelFamily 3.6.6
com.apple.driver.AppleThunderboltPCIUpAdapter 1.4.5
com.apple.driver.AppleThunderboltDPInAdapter 3.1.7
com.apple.driver.AppleThunderboltDPOutAdapter 3.1.7
com.apple.driver.AppleThunderboltDPAdapterFamily 3.1.7
com.apple.driver.AppleThunderboltPCIDownAdapter 1.4.5
com.apple.driver.AppleUSBMergeNub 650.4.0
com.apple.driver.AppleUSBComposite 656.4.1
com.apple.iokit.IOAHCIFamily 2.6.5
com.apple.iokit.IOUSBUserClient 660.4.2
com.apple.iokit.IOEthernetAVBController 1.0.3b4
com.apple.iokit.IOFireWireFamily 4.5.5
com.apple.driver.AppleThunderboltNHI 2.0.1
com.apple.iokit.IOThunderboltFamily 3.2.7
com.apple.iokit.IO80211Family 630.35
com.apple.driver.mDNSOffloadUserClient 1.0.1b5
com.apple.iokit.IONetworkingFamily 3.2
com.apple.iokit.IOUSBFamily 677.4.0
com.apple.driver.AppleEFINVRAM 2.0
com.apple.driver.AppleEFIRuntime 2.0
com.apple.iokit.IOHIDFamily 2.0.0
com.apple.iokit.IOSMBusFamily 1.1
com.apple.security.sandbox 278.11
com.apple.kext.AppleMatch 1.0.0d1
com.apple.security.TMSafetyNet 7
com.apple.driver.AppleKeyStore 2
com.apple.driver.DiskImages 371.1
com.apple.iokit.IOStorageFamily 1.9
com.apple.iokit.IOReportFamily 23
com.apple.driver.AppleFDEKeyStore 28.30
com.apple.driver.AppleACPIPlatform 2.0
com.apple.iokit.IOPCIFamily 2.9
com.apple.iokit.IOACPIFamily 1.4
com.apple.kec.corecrypto 1.0

```
com.apple.kec.pthread   1

System Profile:
Model: MacBookPro11,2, BootROM MBP112.0138.B07, 4 processors, Intel Core i7, 2 GHz, 16 GB, SMC 2.
Graphics: Intel Iris Pro, Intel Iris Pro, Built-In
Memory Module: BANK 0/DIMM0, 8 GB, DDR3, 1600 MHz, 0x80AD, 0x484D5434314753364146523841 2D50422020
Memory Module: BANK 1/DIMM0, 8 GB, DDR3, 1600 MHz, 0x80AD, 0x484D5434314753364146523841 2D50422020
AirPort: spairport_wireless_card_type_airport_extreme (0x14E4, 0x134), Broadcom BCM43xx 1.0 (6.30
Bluetooth: Version 4.2.4f1 13674, 3 services, 15 devices, 1 incoming serial ports
Network Service: Wi-Fi, AirPort, en0
Serial ATA Device: APPLE SSD SM0256F, 251 GB
USB Device: Internal Memory Card Reader
USB Device: BRCM20702 Hub
USB Device: Bluetooth USB Host Controller
USB Device: Apple Internal Keyboard / Trackpad
Thunderbolt Bus: MacBook Pro, Apple Inc., 17.1
```

So, CR2 (0x000000000000007a) looks bogus, leading to the panic. I can't infer much more than that, since I'm missing kernel symbols, and the stack trace is hex.

It would be nice if the kernel wasn't stripped, or, if it was easier to get the Kernel Debug Kit (please put them on opensource.apple.com). I'd help debug this further, but can't without symbols (at least, easily).

A few of us are now have the recurring 10.9.3 panics. After sending Apple many diagnostic reports with additional details, I'm switching back to 10.9.2. 10.9.3 is toxic with remote displays.

A coworker suggested the fix.

While I hope Apple fix this in 10.9.4, I'm now leery of OS X updates. I'd feel a lot better if I could debug this on my own further.

I should add that I've used Apple products and OS X for many years, and have been impressed by the reliability and quality of their work. 10.9.2 on the same laptop worked fine, and I'd have prior OS X releases with hundreds of days of uptime. I'd still recommend their products, as I hope this experience was an outlier.

Update 1

Symbols for this stack below (thanks Rasmus!)

```
panic (in mach_kernel) (debug.c:353)
kernel_trap (in mach_kernel) (trap.c:790)
trap_from_kernel (in mach_kernel) + 38
vm_page_lru (in mach_kernel) (vm_resident.c:3238)
memory_object_control_uiomove (in mach_kernel) (bsd_vm.c:499)
cluster_copy_ubc_data_internal (in mach_kernel) (vfs_cluster.c:5816)
decmpfs_read_compressed (in mach_kernel) (decmpfs.c:1227)
hfs_vnop_read (in mach_kernel) (hfs_readwrite.c:154)
VNOP_READ (in mach_kernel) (kpi_vfs.c:3247)
vn_read (in mach_kernel) (vfs_vnops.c:939)
dofileread (in mach_kernel) (sys_generic.c:377)
pread_nocancel (in mach_kernel) (sys_generic.c:266)
unix_syscall64 (in mach_kernel) (systemcalls.c:370)
hndl_unix_scall64 (in mach_kernel) + 22
```

Update 2

Based on the hackernews comments, some people have hit this and others haven't, and using external displays is a factor. Some have said that they have had issues with 10.9.2 as well, or all the 10.9 series. Perhaps it is a problem with a particular graphics card driver (myself and my coworkers have the retina Macbook Pros), but that's just a guess. The decoded stack above includes HFS calls, but that makes no sense, unless the graphics driver was stepping on random memory (which are among the worst panics to debug).

EDIT: After learning that this doesn't affect everyone, I changed the title of this post from "Is Toxic" to "Recurring Panics". I also removed the words "infected" and "disease", which were off-putting.

Update 3

I got another new macbook pro, running 10.9.2 (although, a different kernel version), and switched using migration assistant. It worked great, initially. Then I had five panics in a row when connecting to different remote displays. In case migration assistant moved over some corrupted preferences, I got *another* new macbook pro, with 10.9.2, and just began using it fresh, and was still able to reproduce the panics (running only Firefox and Chrome, this time). I don't know if these panics are the same as what I had on 10.9.3, since I only have hex dumps to compare. 10.9.3 seemed to panic much more easily.

I'm a little fed up of the typical macbook debugging technique: switch things until the problem goes away. I'm also fed up with seeing stack traces that are inscruitable hex. I want to *read* the stack traces, and understand what the kernel is doing that *led to the panic*. So I studied how Rasmus had translated my earlier stack, and I learned that the default kernel (mach_kernel) does have some symbols, which aren't used in the diagnostic reports. Excellent. I can write a quick helper tool for translating stacks.

Update 4

I wrote **kernel_diagreport2text.ksh**, a tool that translates symbols from OS X kernel diagnostic reports using two different techniques. It is here on github.

Here's my new 10.9.2 macbook panics, summarized by kernel_diagreport2text.ksh:

```
$ ./kernel_diagreport2text.ksh /Library/Logs/DiagnosticReports/Kernel_2014*
File /Library/Logs/DiagnosticReports/Kernel_2014-05-26-110638_lgml-bgregg.panic
panic(cpu 4 caller 0xffffff801aedbe2e): Kernel trap at 0xffffff801aea30a3, type 14=page fault, re
Stack:
    panic (in mach_kernel) + 201
    kernel_trap (in mach_kernel) + 2046
    0xffffff80002f3326 (in mach_kernel) + 38
    vm_page_remove (in mach_kernel) + 115
    vm_page_free_prepare_object (in mach_kernel) + 24
    vm_page_free_list (in mach_kernel) + 103
    0xffffff800028fe03 (in mach_kernel) + 227
    0xffffff80002958b3 (in mach_kernel) + 195
    0xffffff800028f05d (in mach_kernel) + 621
    0xffffff800028e2ac (in mach_kernel) + 652
    0xffffff800027e24c (in mach_kernel) + 1212
    vm_map_remove (in mach_kernel) + 111
    _kernelrpc_mach_vm_deallocate_trap (in mach_kernel) + 71
    0xffffff80002c962d (in mach_kernel) + 301
    hndl_mach_scall (in mach_kernel) + 216
BSD process name: Opera
Mac OS version: 13C64

File /Library/Logs/DiagnosticReports/Kernel_2014-05-26-111056_lgml-bgregg.panic
panic(cpu 2 caller 0xffffff801e4a4f5a): "VM_PAGE_QUEUES_REMOVE: unmarked page on Q"@/SourceCache/
Stack:
    panic (in mach_kernel) + 201
    vm_page_free_prepare_queues (in mach_kernel) + 602
    0xffffff800028ffa8 (in mach_kernel) + 648
    0xffffff80002958b3 (in mach_kernel) + 195
    0xffffff800028f05d (in mach_kernel) + 621
    0xffffff800028e2ac (in mach_kernel) + 652
    mach_destroy_memory_entry (in mach_kernel) + 85
    ipc_port_destroy (in mach_kernel) + 401
    ipc_kobject_notify (in mach_kernel) + 162
    ipc_kobject_server (in mach_kernel) + 270
    ipc_kmsg_send (in mach_kernel) + 117
    mach_msg_send_from_kernel_proper (in mach_kernel) + 66
    mach_notify_no_senders (in mach_kernel) + 65
    IOGeneralMemoryDescriptor::free() (in mach_kernel) + 312
    IOBufferMemoryDescriptor::free() (in mach_kernel) + 157
    0xffffff7fa001231a (in com.apple.iokit.IOAcceleratorFamily2(98.14))
    0xffffff7fa0027b01 (in com.apple.iokit.IOAcceleratorFamily2(98.14))
    0xffffff7fa0034116 (in com.apple.iokit.IOAcceleratorFamily2(98.14))
    0xffffff7fa0031acb (in com.apple.iokit.IOAcceleratorFamily2(98.14))
    0xffffff7fa0033d58 (in com.apple.iokit.IOAcceleratorFamily2(98.14))
    0xffffff7fa002fb73 (in com.apple.iokit.IOAcceleratorFamily2(98.14))
    0xffffff7fa0033fcc (in com.apple.iokit.IOAcceleratorFamily2(98.14))
    IOUserClient::externalMethod(unsigned int, IOExternalMethodArguments*, IOExternalMethodDispat
    is_io_connect_method (in mach_kernel) + 415
    0xffffff80002b6558 (in mach_kernel) + 392
    ipc_kobject_server (in mach_kernel) + 241
    ipc_kmsg_send (in mach_kernel) + 117
    mach_msg_overwrite_trap (in mach_kernel) + 195
    0xffffff80002c976d (in mach_kernel) + 237
    hndl_mach_scall64 (in mach_kernel) + 22
BSD process name: WindowServer
Mac OS version: 13C64

File /Library/Logs/DiagnosticReports/Kernel_2014-05-26-113518_lgml-bgregg.panic
panic(cpu 0 caller 0xffffff800429bb3b): "vm_object_iopl_request: missing/bad page in kernel objec
Stack:
```

```
Stack:
    panic (in mach_kernel) + 201
    vm_object_iopl_request (in mach_kernel) + 1355
    vm_map_create_upl (in mach_kernel) + 1519
    IOGeneralMemoryDescriptor::wireVirtual(unsigned int) (in mach_kernel) + 969
    IOGeneralMemoryDescriptor::prepare(unsigned int) (in mach_kernel) + 78
    IOGeneralMemoryDescriptor::initWithOptions(void*, unsigned int, unsigned int, task*, unsigned
    IOMemoryDescriptor::withAddress(void*, unsigned long long, unsigned int) (in mach_kernel) + 1
    0xffffff7f84d558c2 (in com.apple.iokit.IOUSBFamily(675.4))
    0xffffff7f84d74ba0 (in com.apple.iokit.IOUSBFamily(675.4))
    0xffffff7f84d60fe6 (in com.apple.iokit.IOUSBFamily(675.4))
    0xffffff7f84d6a40d (in com.apple.iokit.IOUSBFamily(675.4))
    IOCommandGate::runAction(int (*)(OSObject*, void*, void*, void*, void*), void*, void*, void*,
    0xffffff7f84d6a301 (in com.apple.iokit.IOUSBFamily(675.4))
    0xffffff7f84eda602 (in com.apple.driver.AppleUSBHub(666.4))
    0xffffff7f84ed5f56 (in com.apple.driver.AppleUSBHub(666.4))
    0xffffff7f84ee3cca (in com.apple.driver.AppleUSBHub(666.4))
    0xffffff7f84edeb8f (in com.apple.driver.AppleUSBHub(666.4))
    0xffffff800024a23a (in mach_kernel) + 506
    call_continuation (in mach_kernel) + 23
BSD process name: kernel_task
Mac OS version: 13C64

File /Library/Logs/DiagnosticReports/Kernel_2014-05-26-115647_lgml-bgregg.panic
panic(cpu 6 caller 0xffffff8019cdbe2e): Kernel trap at 0xffffff8019ca3278, type 14=page fault, re
Stack:
    panic (in mach_kernel) + 201
    kernel_trap (in mach_kernel) + 2046
    0xffffff80002f3326 (in mach_kernel) + 38
    vm_page_lookup (in mach_kernel) + 56
    vm_fault (in mach_kernel) + 782
    user_trap (in mach_kernel) + 748
    hndl_alltraps (in mach_kernel) + 219
BSD process name: WindowServer
Mac OS version: 13C64

File /Library/Logs/DiagnosticReports/Kernel_2014-05-26-124827_lgml-bgregg.panic
panic(cpu 0 caller 0xffffff800e6a4f5a): "VM_PAGE_QUEUES_REMOVE: unmarked page on Q"@/SourceCache/
Stack:
    panic (in mach_kernel) + 201
    vm_page_free_prepare_queues (in mach_kernel) + 602
    0xffffff800028ffa8 (in mach_kernel) + 648
    0xffffff80002958b3 (in mach_kernel) + 195
    0xffffff800028f05d (in mach_kernel) + 621
    0xffffff800028e2ac (in mach_kernel) + 652
    0xffffff800027e24c (in mach_kernel) + 1212
    vm_map_remove (in mach_kernel) + 111
    munmap (in mach_kernel) + 89
    unix_syscall (in mach_kernel) + 471
    hndl_unix_scall (in mach_kernel) + 216
BSD process name: Helper
Mac OS version: 13C64
```

I wish I had this tool earlier! It uses atos(1) for symbol translation, and decorates remaining addresses with kernel extension names (eg, "in com.apple.driver.AppleUSBHub") if available in the diag report. It does not need the Kernel Debug Kit installed, although if it is, you should get more symbols translated.

That output is for a default 10.9.2 system, and while many symbols are missing, we can still learn a lot. All of these panics are in VM, and don't look the same as the 10.9.3 panic that was translated earlier.

To run this yourself, download (or save) the raw script. Then open up Terminal (which is under Applications->Utilities) for a command line, and you can run it on your saved kernel diagnostic reports. The steps are likely something like this (depends where your browser has downloaded the file):

```
cd Downloads
mv kernel_diagreport2text.ksh.txt kernel_diagreport2text.ksh    # may not be necessary
chmod 755 kernel_diagreport2text.ksh
./kernel_diagreport2text.ksh /Library/Logs/DiagnosticReports/Kernel*.panic
```

This script is (obviously) not an official Apple diagnostic tool, and is provided as-is with no warranties or guarantees. It does not need to be run as root.

Update 5

29-May-2014. I've partially translated my 10.9.3 panics, using my kernel_diagreport2text.ksh tool described earlier. Here are some key examples:

```
File ../DiagnosticReports/Kernel_2014-05-20-154336_lgml-bgregg.panic
panic(cpu 0 caller 0xffffff80170a50aa): "VM_PAGE_QUEUES_REMOVE: unmarked page on Q"@/SourceCache/
Stack:
```

```
Stack:
    panic (in mach_kernel) + 201
    vm_page_free_prepare_queues (in mach_kernel) + 602
    0xffffff80002900b8 (in mach_kernel) + 648
    0xffffff8000295a03 (in mach_kernel) + 195
    0xffffff800028f16d (in mach_kernel) + 621
    0xffffff800028e3bc (in mach_kernel) + 652
    0xffffff800027e35c (in mach_kernel) + 1212
    vm_map_remove (in mach_kernel) + 111
    _kernelrpc_mach_vm_deallocate_trap (in mach_kernel) + 71
    0xffffff80002c989d (in mach_kernel) + 237
    hndl_mach_scall64 (in mach_kernel) + 22
BSD process name: WindowServer
Mac OS version: 13D65

File ../DiagnosticReports/Kernel_2014-05-20-160644_lgml-bgregg.panic
panic(cpu 6 caller 0xffffff800aca50aa): "VM_PAGE_QUEUES_REMOVE: unmarked page on Q"@/SourceCache/
Stack:
    panic (in mach_kernel) + 201
    vm_page_free_prepare_queues (in mach_kernel) + 602
    0xffffff80002900b8 (in mach_kernel) + 648
    0xffffff8000295a03 (in mach_kernel) + 195
    0xffffff800028f16d (in mach_kernel) + 621
    0xffffff800028e3bc (in mach_kernel) + 652
    mach_destroy_memory_entry (in mach_kernel) + 85
    ipc_port_destroy (in mach_kernel) + 401
    ipc_kobject_notify (in mach_kernel) + 162
    ipc_kobject_server (in mach_kernel) + 270
    ipc_kmsg_send (in mach_kernel) + 117
    mach_msg_send_from_kernel_proper (in mach_kernel) + 66
    mach_notify_no_senders (in mach_kernel) + 65
    IOGeneralMemoryDescriptor::free() (in mach_kernel) + 312
    IOBufferMemoryDescriptor::free() (in mach_kernel) + 157
    0xffffff7f8c81024a (in com.apple.iokit.IOAcceleratorFamily2(98.20))
    0xffffff7f8c825a6a (in com.apple.iokit.IOAcceleratorFamily2(98.20))
    0xffffff7f8c832046 (in com.apple.iokit.IOAcceleratorFamily2(98.20))
    0xffffff7f8c82f9fb (in com.apple.iokit.IOAcceleratorFamily2(98.20))
    0xffffff7f8c831c88 (in com.apple.iokit.IOAcceleratorFamily2(98.20))
    0xffffff7f8c82daa3 (in com.apple.iokit.IOAcceleratorFamily2(98.20))
    0xffffff7f8c831efc (in com.apple.iokit.IOAcceleratorFamily2(98.20))
    IOUserClient::externalMethod(unsigned int, IOExternalMethodArguments*, IOExternalMethodDispat
    is_io_connect_method (in mach_kernel) + 415
    0xffffff80002b66a8 (in mach_kernel) + 392
    ipc_kobject_server (in mach_kernel) + 241
    ipc_kmsg_send (in mach_kernel) + 117
    mach_msg_overwrite_trap (in mach_kernel) + 195
    0xffffff80002c989d (in mach_kernel) + 237
    hndl_mach_scall64 (in mach_kernel) + 22
BSD process name: WindowServer
Mac OS version: 13D65

File ../DiagnosticReports/Kernel_2014-05-22-155309_lgml-bgregg.panic
panic(cpu 2 caller 0xffffff80160dbf5e): Kernel trap at 0xffffff8016093f39, type 14=page fault, re
Stack:
    panic (in mach_kernel) + 201
    kernel_trap (in mach_kernel) + 2046
    0xffffff80002f3456 (in mach_kernel) + 38
    0xffffff8000293f39 (in mach_kernel) + 233
    mach_memory_entry_get_page_counts (in mach_kernel) + 110
    IOMemoryDescriptor::getPageCounts(unsigned long long*, unsigned long long*) (in mach_kernel)
    0xffffff7f97c1108d (in com.apple.iokit.IOAcceleratorFamily2(98.20))
    0xffffff7f97c3924c (in com.apple.iokit.IOAcceleratorFamily2(98.20))
    0xffffff7f97c40fb2 (in com.apple.iokit.IOAcceleratorFamily2(98.20))
    0xffffff7f97c40ddc (in com.apple.iokit.IOAcceleratorFamily2(98.20))
    IOUserClient::externalMethod(unsigned int, IOExternalMethodArguments*, IOExternalMethodDispat
    is_io_connect_method (in mach_kernel) + 415
    0xffffff80002b66a8 (in mach_kernel) + 392
    ipc_kobject_server (in mach_kernel) + 241
    ipc_kmsg_send (in mach_kernel) + 117
    mach_msg_overwrite_trap (in mach_kernel) + 195
    0xffffff80002c989d (in mach_kernel) + 237
    hndl_mach_scall64 (in mach_kernel) + 22
BSD process name: UserEventAgent
Mac OS version: 13D65

File ../DiagnosticReports/Kernel_2014-05-23-103228_lgml-bgregg.panic
panic(cpu 0 caller 0xffffff8007cdbf5e): Kernel trap at 0xffffff8007ca6968, type 14=page fault, re
Stack:
    panic (in mach_kernel) + 201
    kernel_trap (in mach_kernel) + 2046
    0xffffff80002f3456 (in mach_kernel) + 38
    vm_page_lru (in mach_kernel) + 408
    memory_object_control_uiomove (in mach_kernel) + 680
    0xffffff80003d2ae3 (in mach_kernel) + 195
    decmpfs_read_compressed (in mach_kernel) + 237
    hfs_vnop_read (in mach_kernel) + 186
    VNOP_READ (in mach_kernel) + 225
    0xffffff80003f38d5 (in mach_kernel) + 245
    0xffffff80005f1cfe (in mach_kernel) + 174
    pread_nocancel (in mach_kernel) + 137
    unix_syscall64 (in mach_kernel) + 499
    hndl_unix_scall64 (in mach_kernel) + 22
BSD process name: nsupdate
Mac OS version: 13D65
```

The others showed similar stacks. These are also all in VM, either page faults or an explicit panic() call in the VM_PAGE_QUEUES_REMOVE macro.

To translate my 10.9.3 diag reports with the kernel_diagreport2text.ksh script, I needed a copy of the 10.9.3 mach_kernel (I'm back on 10.9.2), and to edit the script to point to it (update: that's now the -f option). Apple's auto update had already downloaded the 10.9.3 update, putting it in /Library/Updates/031-02348/OSXUpd10.9.3.pkg. That pkg file turned out to be Russian dolls: a xar, containing a bzip2 file, containing a cpio archive, which contained the 10.9.3 mach_kernel.

If you ever want to do something similar yourself, you really want to make sure the mach_kernel matches what you have in the diag report, otherwise the translations will be incorrect. Eg:

```
$ grep 'Kernel Version' Kernel_2014-05-23-103228_lgml-bgregg.panic
Darwin Kernel Version 13.2.0: Thu Apr 17 23:03:13 PDT 2014; root:xnu-2422.100.13~1/RELEASE_X86_64
$ strings 10.9.3/mach_kernel | grep 'Kernel Version'
Darwin Kernel Version 13.2.0: Thu Apr 17 23:03:13 PDT 2014; root:xnu-2422.100.13~1/RELEASE_X86_64
```

Those match! The source for xnu-2422.100.13 isn't out yet, but when it is, it should be under opensource.apple.com/source/xnu. I've been browsing the earlier version to get a handle on the VM code.

Update 6

After a quick browse of the VM code, it looks like a double free, based on the VM_PAGE_QUEUES_REMOVE panic. It's possible the other VM panics are manifestations of the same bug. These are nasty bugs to debug, as the engineer must track down the earlier free. This is harder than it sounds, as free's occur so frequently. I could run out of memory trying to log them for later lookup.

But after using DTrace on the vm_page_free_prepare_queues path, I'm not sure it's a double free using the vm_* interface, as mem->free was not set. Which suggests something even *nastier* – someone else is stepping on memory, perhaps zero'ing it out. Now if two paths are fighting over the same memory, and if I'm lucky, they do so in either order with the 2nd hitting the panic. Which could mean I've already captured both paths in the earlier panics. The IOBufferMemoryDescriptor::free() could be the non-VM path that's freeing memory, coming from IOAcceleratorFamily2.

This is just speculation - I don't know what the real cause is yet. But given the nature of the panics (external displays), the partially-translated stacks, the open source xnu kernel, and DTrace to test theories, I have a lot of clues. The hardest part is finding time to put into this.

Update 7

Here's an older panic excerpt (OS X 10.5.8):

```
Backtrace (CPU 0), Frame : Return Address (4 potential args on stack)
0x343a2a88 : 0x12b4c6 (0x45f91c 0x343a2abc 0x13355c 0x0)
0x343a2ad8 : 0x1ab0fe (0x469a98 0x19d7a7 0xe 0x469248)
0x343a2bb8 : 0x1a1713 (0x343a2bcc 0x343a2c18 0x19d7a7 0xe)
0x343a2bc4 : 0x19d7a7 (0xe 0x48 0x31f000c 0x343a000c)
0x343a2c18 : 0xbb17c4 (0x0 0x3ba823e 0xfe825ec8 0x19fed4)
[...]
```

Notice something? This has *arguments* for the stack functions, which are incredibly useful when doing panic analysis, especially when all we have to go on is the panic report file. So where did these args go in 10.9?

From the xnu-2422.90.20 source, osfmk/i386/AT386/model_dep.c:

```
        kdb_printf("Backtrace (CPU %d), "
#if PRINT_ARGS_FROM_STACK_FRAME
        "Frame : Return Address (4 potential args on stack)\n", cn);
#else
        "Frame : Return Address\n", cn);
#endif
```

PRINT_ARGS_FROM_STACK_FRAME needs to be set. It's set in the same file:

```
[...]
volatile int panic_double_fault_cpu = -1;

#define PRINT_ARGS_FROM_STACK_FRAME     0

typedef struct _cframe_t {
[...]
```

(shakes fist.) No comment, no explanation, just zero that guy. Why?

I think the reason can be explained by an earlier kernel version, xnu-2050.9.2:

```
#if defined (__i386__)
#define PRINT_ARGS_FROM_STACK_FRAME 1
#elif defined (__x86_64__)
#define PRINT_ARGS_FROM_STACK_FRAME 0
#else
#error unsupported architecture
#endif
```

Ah. Stack frame arguments are architecture specific, and the code was written for i386, but not x86.

It should be a fairly easy enhancement for Apple to add the x86 code, and greatly enhance *all* kernel panic reports.

Update 8

30-May-2014. I have a workaround in hand for Mavericks 10.9.2: **turn off Firefox hardware acceleration**. With this off, my panics have now stopped. Perhaps this works on 10.9.3 as well, if it is the same panic. The setting is under Preferences->Advanced->General.

Other applications use hardware acceleration as well, so you may need to disable it elsewhere, if you believe you have this bug. For me, the easiest way to duplicate the panic was to run a youtube video full screen in Firefox, then plug in a remote display, and close the laptop lid. A few cycles of that usually led to the same type of panic I was hitting earlier. And with hardware acceleration disabled, it worked fine.

I still feel this was much worse in 10.9.3 (if I had more time, I'd confirm). The 10.9.3 update did say that it "Improves 4K display support", which may have modified the hardware acceleration routines.

I suspect hardware acceleration was freeing or stepping on memory it shouldn't, which led to the VM panics. I'd like to write more about this, including where I was in the analysis, and ideally identify the root cause, but now that I have a workaround it's no longer a priority to work on. I may do a follow up blog post later when I have the time. I wanted to share the workaround ASAP.

This information has been filed as Apple bug ID 17082120. (This is in addition to the 30 or so Kernel diag reports I've sent their way.)

You can comment here, but I can't guarantee your comment will remain here forever: I might switch comment systems at some point (eg, if disqus add advertisements).