DragonFlyBSD - Bug #1029

Dragonfly under KVM

06/05/2008 07:43 PM - dragonfly1

Status:	Closed	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:			

Description

Hello,

I've been experiencing lock-ups using DragonFly HEAD SMP under kvm. Running "make -j8 buildworld" triggers a completely unresponsive state and 100.00% CPU usage on all four cores (Seen from host OS). Could someone offer advice on how to debug this? Either triggering DragonFly to drop to gdb or getting the kvm attached gdb to interrupt the OS.

So far I've turned up nothing on the Wiki's or Google.

Regards

Gary

History

#1 - 06/06/2008 10:43 PM - dragonfly1

I've managed to get gdb attached and get some information.

The kernel is getting caught in a while loop in lwkt_acquire. I can reliably trigger this with with a "make -j8 buildworld" under a SMP kernel (Otherwise identical to GENERIC, no optimisations.) The OS is completely unresponsive and all four cpu cores are running at 100%.

I've included the debug information.

```
Program received signal SIGINT, Interrupt.
lwkt_acquire (td=0xc6a59e70) at /usr/src/sys/kern/lwkt_thread.c:1048
1048  while (td->td_flags & (TDF_RUNNING|TDF_PREEMPT_LOCK))
(gdb) l
1043     mygd = mycpu;
1044     if (gd != mycpu) {
1045  cpu_lfence();
1046  KKASSERT((td->td_flags & TDF_RUNQ) == 0);
1047  crit_enter_gd(mygd);
1048  while (td->td_flags & (TDF_RUNNING|TDF_PREEMPT_LOCK))
1049      cpu_lfence();
1050  td->td_gd = mygd;
1051  TAILQ_INSERT_TAIL(&mygd->gd_tdallq, td, td_allq);
1052  td->td_flags &= ~TDF_MIGRATING;
(gdb) p td->td_flags
$1 = 8390177
(gdb) p td
$2 = (thread_t) 0xc6a59e70
(gdb) bt
#0  lwkt_acquire (td=0xc6a59e70) at /usr/src/sys/kern/lwkt_thread.c:1048
#1  0xc02c66af in bsd4_select_curproc (gd=0xff800000) at
/usr/src/sys/kern/usched_bsd4.c:358
#2  0xc02c6829 in bsd4_release_curproc (lp=0xea634c00) at
/usr/src/sys/kern/usched_bsd4.c:322
#3  0xc04b8239 in passive_release (td=0xdfe8aba0) at
/usr/src/sys/platform/pc32/i386/trap.c:212
#4  0xc02c870b in lwkt_switch () at /usr/src/sys/kern/lwkt_thread.c:491
#5  0xc02c8b3b in lwkt_mp_lock_contested () at
/usr/src/sys/kern/lwkt_thread.c:1374
#6  0xc04b0751 in get_mplock () at
```

/usr/src/sys/platform/pc32/i386/mplock.s:168
#7 0xe9ef6d34 in ?? ()
#8 0xc04b94a4 in syscall2 (frame=0xe9ef6d40) at
/usr/src/sys/platform/pc32/i386/trap.c:1371
#9 0xc04a3396 in Xint0x80_syscall () at
/usr/src/sys/platform/pc32/i386/exception.s:876
#10 0xe9ef6d40 in ?? ()
Backtrace stopped: previous frame inner to this frame (corrupt stack?)
(gdb) jump 1050
Continuing at 0xc02c8bbb.

Continuing execution does not appear to cause any problems.
I can provide additional debugging info if required but I'm unsure of
how to proceed with this myself.

Regards

Gary

#2 - 06/06/2008 10:52 PM - dillon

:I've managed to get gdb attached and get some information.
:
:The kernel is getting caught in a while loop in lwkt_acquire. I can
:reliably trigger this with with a "make -j8 buildworld" under a SMP
:kernel (Otherwise identical to GENERIC, no optimisations.) The OS is
:completely unresponsive and all four cpu cores are running at 100%.
:
:I've included the debug information.
:
:Program received signal SIGINT, Interrupt.
:lwkt_acquire (td=0xc6a59e70) at /usr/src/sys/kern/lwkt_thread.c:1048
:1048 while (td->td_flags & (TDF_RUNNING|TDF_PREEMPT_LOCK))
:(gdb) l
:1043   mygd = mycpu;
:1044   if (gd != mycpu) {
:1045 cpu_lfence();
:1046 KKASSERT((td->td_flags & TDF_RUNQ) == 0);
:1047 crit_enter_gd(mygd);
:1048 while (td->td_flags & (TDF_RUNNING|TDF_PREEMPT_LOCK))
:1049   cpu_lfence();
:1050 td->td_gd = mygd;
:1051 TAILQ_INSERT_TAIL(&mygd->gd_tdallq, td, td_allq);
:1052 td->td_flags &= ~TDF_MIGRATING;
:(gdb) p td->td_flags
:$1 = 8390177
:(gdb) p td
:$2 = (thread_t) 0xc6a59e70
:(gdb) bt
:#0 lwkt_acquire (td=0xc6a59e70) at /usr/src/sys/kern/lwkt_thread.c:1048
:#1 0xc02c66af in bsd4_select_curproc (gd=0xff800000) at
:/usr/src/sys/kern/usched_bsd4.c:358
:#2 0xc02c6829 in bsd4_release_curproc (lp=0xea634c00) at
:/usr/src/sys/kern/usched_bsd4.c:322
:#3 0xc04b8239 in passive_release (td=0xdfe8aba0) at
:...
:
:Continuing execution does not appear to cause any problems.
:I can provide additional debugging info if required but I'm unsure of
:how to proceed with this myself.
:
:Regards
:
:Gary

This is great info. One thing, what do you mean by 'KVM'? What
is DragonFly running under, exactly?

I think once I understand the environment I may be able to figure out
why the acquisition loop is blowing up.

-Matt
Matthew Dillon
<dillon@backplane.com>

#3 - 06/07/2008 04:59 AM - fjwcash

http://kvm.qumranet.com
Linux Kernel-based Virtual Machine.

Uses a modified QEmu and hardware virtualisation support in AMD and
Intel CPUs to effectively turn the Linux kernel into a hypervisor.
The VMs run as user processes on top of a (fairly) standard Linux
install.

Freddie

#4 - 06/07/2008 12:03 PM - dragonfly1

As Freddie says I'm running the QEmu KVM software on Ubuntu x64.
DragonFly is running 32bit with 3.5Gb RAM and 4 CPUs (AMD-Phenom).

I can also trigger SMP DragonFly to lock up during a buildworld on
VMWare Server 1.0.5 (Dual Core, 2Gb RAM) and VirtualBox. I haven't tried
the bare metal but could do so if anyone thinks it would be useful. I'm
running HEAD as of yesterday but 1.12.2 also locks up.

The kernel is GENERIC with "options SMP" and "options APIC_IO" enabled
and no modifications to /etc/make.conf. I have also tried enabling
"options ACPI_QUIRK_VMWARE".

For debugging I copied the kernel and source to my Ubuntu home directory
and ran.

kvm -smp 4 -m 3500 -hda "~/img/dfly.img" -net nic,model=e1000,vlan=1
-net user,vlan=1 -redir tcp:2201::22 -s
gdb -s kernel.SMP -d ~/source/
(gdb) target remote localhost:1234
(gdb) c

Regards

Gary

#5 - 06/07/2008 12:14 PM - sepherosa

I have a box using Phenom 9550 and 2GB ram. Dfly HEAD runs directly
on the box (no virtual tech involved). Kernel is configured with SMP
and APIC_IO. I build world and kernel with -j 8 or -j 16 w/o any
issues.

#6 - 06/07/2008 01:14 PM - cedric

I've had issues like this with OpenBSD on VMWare a couple year ago.
Actually, VMs emulate "bare metal" systems very well, but they introduce
very unusual timings. So if you've a race that is "to short to be a
serious problem", than that race might well become a real problem on
VMs. To sum up, testing OSes on virtualized environment is a good way
to find hidden/rare bugs :)

Cedric

#7 - 06/07/2008 03:38 PM - dragonfly1

I've triggered another lock up and this time got a different trace.
Execution appears to be looping indefinitely inside LWKT code.

Debugging gives the output below. Again all four core are running at 100%.

Program received signal SIGINT, Interrupt.
lwkt_process_ipiq_core (sgd=<value optimized out>, ip=0xc67a7000, frame=0x0)
at /usr/src/sys/kern/lwkt_ipiq.c:522
522   while (wi - (ri = ip->ip_rindex) > 0) {
(gdb) l
517   *
518   * Note: due to additional IPI operations that the callback function
519   * may make, it is possible for both rindex and windex to
advance and
520   * thus for rindex to advance passed our cached windex.
521   */
522   while (wi - (ri = ip->ip_rindex) > 0) {

```
523  ri &= MAXCPUFIFO_MASK;
524  copy_func = ip->ip_func[ri];
525  copy_arg1 = ip->ip_arg1[ri];
526  copy_arg2 = ip->ip_arg2[ri];
(gdb) p wi
$5 = 356278
(gdb) p ip->ip_rindex
$6 = 356278
(gdb) bt
#0  lwkt_process_ipiq_core (sgd=<value optimized out>, ip=0xc67a7000,
frame=0x0) at /usr/src/sys/kern/lwkt_ipiq.c:522
#1  0xc02c94ad in lwkt_process_ipiq () at /usr/src/sys/kern/lwkt_ipiq.c:452
#2  0xc02c9830 in lwkt_send_ipiq3 (target=0xff808000,
func=0xc02c8519 <lwkt_schedule>, arg1=0xc0600170, arg2=0)
at /usr/src/sys/kern/lwkt_ipiq.c:185
#3  0xc02c863c in lwkt_schedule (td=0xc0600170)
at /usr/src/sys/sys/thread2.h:244
#4  0xc02c71e6 in bsd4_setrunqueue (lp=0xe5f8b400)
at /usr/src/sys/kern/usched_bsd4.c:551
#5  0xc02c72be in bsd4_acquire_curproc (lp=0xe5f8b400)
at /usr/src/sys/kern/usched_bsd4.c:271
#6  0xc04b9603 in syscall2 (frame=0xe5a02d40)
at /usr/src/sys/platform/pc32/i386/trap.c:349
#7  0xc04a3396 in Xint0x80_syscall ()
at /usr/src/sys/platform/pc32/i386/exception.s:876
#8  0xe5a02d40 in ?? ()
Backtrace stopped: previous frame inner to this frame (corrupt stack?)
(gdb) c
Continuing.

Program received signal SIGINT, Interrupt.
lwkt_process_ipiq_core (sgd=<value optimized out>, ip=0xc67ac000, frame=0x0)
at /usr/src/sys/kern/lwkt_ipiq.c:558
558     return(wi != ip->ip_windex);
(gdb) l
553      * ipiq.  ip_npoll is left set as long as possible to reduce the
554      * number of IPIs queued by the originating cpu, but must be cleared
555      * *BEFORE* checking windex.
556      */
557     atomic_poll_release_int(&ip->ip_npoll);
558     return(wi != ip->ip_windex);
559 }
560
561 static void
562 lwkt_sync_ipiq(void *arg)
(gdb) p wi
$7 = 357733
(gdb) p ip->ip_windex
$8 = 357733
(gdb) bt
#0  lwkt_process_ipiq_core (sgd=<value optimized out>, ip=0xc67ac000,
frame=0x0) at /usr/src/sys/kern/lwkt_ipiq.c:558
#1  0xc02c94ad in lwkt_process_ipiq () at /usr/src/sys/kern/lwkt_ipiq.c:452
#2  0xc02c9830 in lwkt_send_ipiq3 (target=0xff808000,
func=0xc02c8519 <lwkt_schedule>, arg1=0xc0600170, arg2=0)
at /usr/src/sys/kern/lwkt_ipiq.c:185
#3  0xc02c863c in lwkt_schedule (td=0xc0600170)
at /usr/src/sys/sys/thread2.h:244
#4  0xc02c71e6 in bsd4_setrunqueue (lp=0xe5f8b400)
at /usr/src/sys/kern/usched_bsd4.c:551
#5  0xc02c72be in bsd4_acquire_curproc (lp=0xe5f8b400)
at /usr/src/sys/kern/usched_bsd4.c:271
#6  0xc04b9603 in syscall2 (frame=0xe5a02d40)
at /usr/src/sys/platform/pc32/i386/trap.c:349
#7  0xc04a3396 in Xint0x80_syscall ()
at /usr/src/sys/platform/pc32/i386/exception.s:876
#8  0xe5a02d40 in ?? ()
Backtrace stopped: previous frame inner to this frame (corrupt stack?)
(gdb) c
Continuing.

Program received signal SIGINT, Interrupt.
lwkt_process_ipiq_core (sgd=<value optimized out>, ip=0xc67b1000, frame=0x0)
at /usr/src/sys/kern/lwkt_ipiq.c:558
558     return(wi != ip->ip_windex);
```

```
(gdb) p wi
$9 = 372884
(gdb) p  ip->ip_windex
$10 = 372884
(gdb) bt
#0  lwkt_process_ipiq_core (sgd=<value optimized out>, ip=0xc67b1000,
frame=0x0) at /usr/src/sys/kern/lwkt_ipiq.c:558
#1  0xc02c94ad in lwkt_process_ipiq () at /usr/src/sys/kern/lwkt_ipiq.c:452
#2  0xc02c9830 in lwkt_send_ipiq3 (target=0xff808000,
func=0xc02c8519 <lwkt_schedule>, arg1=0xc0600170, arg2=0)
at /usr/src/sys/kern/lwkt_ipiq.c:185
#3  0xc02c863c in lwkt_schedule (td=0xc0600170)
at /usr/src/sys/sys/thread2.h:244
#4  0xc02c71e6 in bsd4_setrunqueue (lp=0xe5f8b400)
at /usr/src/sys/kern/usched_bsd4.c:551
#5  0xc02c72be in bsd4_acquire_curproc (lp=0xe5f8b400)
at /usr/src/sys/kern/usched_bsd4.c:271
#6  0xc04b9603 in syscall2 (frame=0xe5a02d40)
at /usr/src/sys/platform/pc32/i386/trap.c:349
#7  0xc04a3396 in Xint0x80_syscall ()
at /usr/src/sys/platform/pc32/i386/exception.s:876
#8  0xe5a02d40 in ?? ()
Backtrace stopped: previous frame inner to this frame (corrupt stack?)
(gdb) c
Continuing.

Program received signal SIGINT, Interrupt.
0xc02c93a5 in lwkt_process_ipiq_core (sgd=<value optimized out>,
ip=0xff8001e8, frame=0x0) at /usr/src/sys/kern/lwkt_ipiq.c:522
522     while (wi - (ri = ip->ip_rindex) > 0) {
(gdb) p wi
$11 = 1343383299
(gdb) p ip->ip_rindex
$12 = 1343383298
(gdb) bt
#0  0xc02c93a5 in lwkt_process_ipiq_core (sgd=<value optimized out>,
ip=0xff8001e8, frame=0x0) at /usr/src/sys/kern/lwkt_ipiq.c:522
#1  0xc02c94df in lwkt_process_ipiq () at /usr/src/sys/kern/lwkt_ipiq.c:458
#2  0xc02c9830 in lwkt_send_ipiq3 (target=0xff808000,
func=0xc02c8519 <lwkt_schedule>, arg1=0xc0600170, arg2=0)
at /usr/src/sys/kern/lwkt_ipiq.c:185
#3  0xc02c863c in lwkt_schedule (td=0xc0600170)
at /usr/src/sys/sys/thread2.h:244
#4  0xc02c71e6 in bsd4_setrunqueue (lp=0xe5f8b400)
at /usr/src/sys/kern/usched_bsd4.c:551
#5  0xc02c72be in bsd4_acquire_curproc (lp=0xe5f8b400)
at /usr/src/sys/kern/usched_bsd4.c:271
#6  0xc04b9603 in syscall2 (frame=0xe5a02d40)
at /usr/src/sys/platform/pc32/i386/trap.c:349
#7  0xc04a3396 in Xint0x80_syscall ()
at /usr/src/sys/platform/pc32/i386/exception.s:876
#8  0xe5a02d40 in ?? ()
Backtrace stopped: previous frame inner to this frame (corrupt stack?)
(gdb) c
Continuing.

Program received signal SIGINT, Interrupt.
0xc02c9495 in lwkt_process_ipiq () at /usr/src/sys/kern/lwkt_ipiq.c:452
452   while (lwkt_process_ipiq_core(sgd, &ip[gd->gd_cpuid], NULL))
(gdb) l
447     for (n = 0; n < ncpus; ++n) {
448  if (n != gd->gd_cpuid) {
449       sgd = globaldata_find(n);
450       ip = sgd->gd_ipiq;
451       if (ip != NULL) {
452   while (lwkt_process_ipiq_core(sgd, &ip[gd->gd_cpuid], NULL))
453       ;
454       }
455  }
456     }
(gdb) bt
#0  0xc02c9495 in lwkt_process_ipiq () at /usr/src/sys/kern/lwkt_ipiq.c:452
#1  0xc02c9830 in lwkt_send_ipiq3 (target=0xff808000, Cannot access
memory at address 0x8
func=0xc02c8519 <lwkt_schedule>, arg1=0xc0600170, arg2=0)
```

```
at /usr/src/sys/kern/lwkt_ipiq.c:185
#2  0xc02c863c in lwkt_schedule (td=0xc0600170)
at /usr/src/sys/sys/thread2.h:244
#3  0xc02c71e6 in bsd4_setrunqueue (lp=0xe5f8b400)
at /usr/src/sys/kern/usched_bsd4.c:551
#4  0xc02c72be in bsd4_acquire_curproc (lp=0xe5f8b400)
at /usr/src/sys/kern/usched_bsd4.c:271
#5  0xc04b9603 in syscall2 (frame=0xe5a02d40)
at /usr/src/sys/platform/pc32/i386/trap.c:349
#6  0xc04a3396 in Xint0x80_syscall ()
at /usr/src/sys/platform/pc32/i386/exception.s:876
#7  0xe5a02d40 in ?? ()
Backtrace stopped: previous frame inner to this frame (corrupt stack?)
(gdb) c
Continuing.

Program received signal SIGINT, Interrupt.
0xc02c944c in lwkt_process_ipiq_core (sgd=<value optimized out>,
ip=<value optimized out>, frame=0x0) at
/usr/src/sys/kern/lwkt_ipiq.c:559
559 }
(gdb) l
554      * number of IPIs queued by the originating cpu, but must be cleared
555      * *BEFORE* checking windex.
556      */
557     atomic_poll_release_int(&ip->ip_npoll);
558     return(wi != ip->ip_windex);
559 }
560
561 static void
562 lwkt_sync_ipiq(void *arg)
563 {
(gdb) p wi
$13 = 357733
(gdb) p ip->ip_windex
Cannot access memory at address 0x8
(gdb) bt
#0  0xc02c944c in lwkt_process_ipiq_core (sgd=<value optimized out>,
ip=<value optimized out>, frame=0x0) at
/usr/src/sys/kern/lwkt_ipiq.c:559
#1  0xc02c94ad in lwkt_process_ipiq () at /usr/src/sys/kern/lwkt_ipiq.c:452
#2  0xc02c9830 in lwkt_send_ipiq3 (target=0xff808000,
func=0xc02c8519 <lwkt_schedule>, arg1=0xc0600170, arg2=0)
at /usr/src/sys/kern/lwkt_ipiq.c:185
#3  0xc02c863c in lwkt_schedule (td=0xc0600170)
at /usr/src/sys/sys/thread2.h:244
#4  0xc02c71e6 in bsd4_setrunqueue (lp=0xe5f8b400)
at /usr/src/sys/kern/usched_bsd4.c:551
#5  0xc02c72be in bsd4_acquire_curproc (lp=0xe5f8b400)
at /usr/src/sys/kern/usched_bsd4.c:271
#6  0xc04b9603 in syscall2 (frame=0xe5a02d40)
at /usr/src/sys/platform/pc32/i386/trap.c:349
#7  0xc04a3396 in Xint0x80_syscall ()
at /usr/src/sys/platform/pc32/i386/exception.s:876
#8  0xe5a02d40 in ?? ()
Backtrace stopped: previous frame inner to this frame (corrupt stack?)
(gdb) c
Continuing.

Program received signal SIGINT, Interrupt.
lwkt_process_ipiq_core (sgd=<value optimized out>, ip=0xc67ac000, frame=0x0)
at /usr/src/sys/kern/lwkt_ipiq.c:558
558     return(wi != ip->ip_windex);
(gdb) l
553      * ipiq.  ip_npoll is left set as long as possible to reduce the
554      * number of IPIs queued by the originating cpu, but must be cleared
555      * *BEFORE* checking windex.
556      */
557     atomic_poll_release_int(&ip->ip_npoll);
558     return(wi != ip->ip_windex);
559 }
560
561 static void
562 lwkt_sync_ipiq(void *arg)
(gdb) p wi
```

$24 = 357733
(gdb) p ip->ip_windex
$25 = 357733
(gdb) bt
#0 lwkt_process_ipiq_core (sgd=<value optimized out>, ip=0xc67ac000,
frame=0x0) at /usr/src/sys/kern/lwkt_ipiq.c:558
#1 0xc02c94ad in lwkt_process_ipiq () at /usr/src/sys/kern/lwkt_ipiq.c:452
#2 0xc02c9830 in lwkt_send_ipiq3 (target=0xff808000,
func=0xc02c8519 <lwkt_schedule>, arg1=0xc0600170, arg2=0)
at /usr/src/sys/kern/lwkt_ipiq.c:185
#3 0xc02c863c in lwkt_schedule (td=0xc0600170)
at /usr/src/sys/sys/thread2.h:244
#4 0xc02c71e6 in bsd4_setrunqueue (lp=0xe5f8b400)
at /usr/src/sys/kern/usched_bsd4.c:551
#5 0xc02c72be in bsd4_acquire_curproc (lp=0xe5f8b400)
at /usr/src/sys/kern/usched_bsd4.c:271
#6 0xc04b9603 in syscall2 (frame=0xe5a02d40)
at /usr/src/sys/platform/pc32/i386/trap.c:349
#7 0xc04a3396 in Xint0x80_syscall ()
at /usr/src/sys/platform/pc32/i386/exception.s:876
#8 0xe5a02d40 in ?? ()
Backtrace stopped: previous frame inner to this frame (corrupt stack?)
(gdb) c
Continuing.

#8 - 06/07/2008 04:13 PM - dillon

:http://kvm.qumranet.com
:Linux Kernel-based Virtual Machine.
:
:Uses a modified QEmu and hardware virtualisation support in AMD and
:Intel CPUs to effectively turn the Linux kernel into a hypervisor.
:The VMs run as user processes on top of a (fairly) standard Linux
:install.
:
:Freddie

Try putting a pause instruction in those loops.

I am hitting a lockup on one of my test boxes but I do not yet know
if it is due to a bug in HAMMER (which is what I'm testing on that box),
or due to recent SMP work, or whether it is related to your report.

Is there any chance you can get a backtrace on each one of the cpu's
when it gets into that lockup?

-Matt

#9 - 06/13/2008 10:37 PM - msylvan

@Gary:

| I can also trigger SMP DragonFly to lock up during a buildworld on
| VMWare Server 1.0.5 (Dual Core, 2Gb RAM) and VirtualBox. I haven't tried
| the bare metal but could do so if anyone thinks it would be useful. I'm
| running HEAD as of yesterday but 1.12.2 also locks up.

Pardon me asking, but how did you get DragonFly booting on VirtualBox? I tried
it again (with 1.6.2 on Linux x64) after reading your post. With ACPI enabled it
freezes during the countdown; with ACPI disabled, just after countdown. Bug #995
still appears to be active too.

Would love to get DragonFly working, and I don't feel like using KVM.

--
Michel

#10 - 06/14/2008 04:05 PM - dillon

:..
:> :#0 lwkt_acquire (td=0xc6a59e70) at /usr/src/sys/kern/lwkt_thread.c:1048
:> :#1 0xc02c66af in bsd4_select_curproc (gd=0xff800000) at
:> :/usr/src/sys/kern/usched_bsd4.c:358
:> :#2 0xc02c6829 in bsd4_release_curproc (lp=0xea634c00) at
:> :/usr/src/sys/kern/usched_bsd4.c:322

:> :#3 0xc04b8239 in passive_release (td=0xdfe8aba0) at
:> :..
:Execution appears to be looping indefinitely inside LWKT code.
:
:Debugging gives the output below. Again all four core are running at 100%.
:
:
:Program received signal SIGINT, Interrupt.
:lwkt_process_ipiq_core (sgd=<value optimized out>, ip=0xc67a7000, frame=0x0)
:   at /usr/src/sys/kern/lwkt_ipiq.c:522
:522   while (wi - (ri = ip->ip_rindex) > 0) {
:(gdb) l
:...
:   at /usr/src/sys/sys/thread2.h:244
:#4 0xc02c71e6 in bsd4_setrunqueue (lp=0xe5f8b400)
:   at /usr/src/sys/kern/usched_bsd4.c:551
:#5 0xc02c72be in bsd4_acquire_curproc (lp=0xe5f8b400)
:   at /usr/src/sys/kern/usched_bsd4.c:271
:#6 0xc04b9603 in syscall2 (frame=0xe5a02d40)
:   at /usr/src/sys/platform/pc32/i386/trap.c:349
:#7 0xc04a3396 in Xint0x80_syscall ()
:...
:(gdb) bt
:#0 lwkt_process_ipiq_core (sgd=<value optimized out>, ip=0xc67b1000,
:   frame=0x0) at /usr/src/sys/kern/lwkt_ipiq.c:558
:#1 0xc02c94ad in lwkt_process_ipiq () at /usr/src/sys/kern/lwkt_ipiq.c:452
:#2 0xc02c9830 in lwkt_send_ipiq3 (target=0xff808000,
:   func=0xc02c8519 <lwkt_schedule>, arg1=0xc0600170, arg2=0)
:   at /usr/src/sys/kern/lwkt_ipiq.c:185
:#3 0xc02c863c in lwkt_schedule (td=0xc0600170)
:   at /usr/src/sys/sys/thread2.h:244
:#4 0xc02c71e6 in bsd4_setrunqueue (lp=0xe5f8b400)
:   at /usr/src/sys/kern/usched_bsd4.c:551
:#5 0xc02c72be in bsd4_acquire_curproc (lp=0xe5f8b400)
:   at /usr/src/sys/kern/usched_bsd4.c:271
:#6 0xc04b9603 in syscall2 (frame=0xe5a02d40)
:   at /usr/src/sys/platform/pc32/i386/trap.c:349
:#7 0xc04a3396 in Xint0x80_syscall ()
:   at /usr/src/sys/platform/pc32/i386/exception.s:876
:#8 0xe5a02d40 in ?? ()

I think I see what may be happening here, and I am starting to wonder
if it is also the cause of the system lockups I am getting when testing
HAMMER under extreme loads (with hundreds of user threads which are
sometimes cpu-bound).

I think it may be deadlocking between lwkt_acquire() and lwkt_schedule().
The thread trying to migrate between cpu's is getting stuck and the
acquisition loop is not processing incoming IPIs while it is waiting for
the thread to deschedule on the other cpu.

Please try this patch.

-Matt
Matthew Dillon
<dillon@backplane.com>
Index: lwkt_thread.c
===
RCS file: /cvs/src/sys/kern/lwkt_thread.c,v
retrieving revision 1.115
diff -u -p -r1.115 lwkt_thread.c
--- lwkt_thread.c 2 Jun 2008 16:54:21 -0000 1.115
+++ lwkt_thread.c 14 Jun 2008 15:56:28 -0000
@@ -1045,8 +1045,12 @@   if (gd != mycpu) {
cpu_lfence();
KKASSERT((td->td_flags & TDF_RUNQ) == 0);
crit_enter_gd(mygd);
- while (td->td_flags & (TDF_RUNNING|TDF_PREEMPT_LOCK))
+ while (td->td_flags & (TDF_RUNNING|TDF_PREEMPT_LOCK)) {
+#ifdef SMP
+   lwkt_process_ipiq();
+#endif
cpu_lfence();
+ }
td->td_gd = mygd;

```
TAILQ_INSERT_TAIL(&mygd->gd_tdallq, td, td_allq);
td->td_flags &= ~TDF_MIGRATING;
@@ -1222,8 +1226,12 @@ {
thread_t td = arg;
globaldata_t gd = mycpu;

-    while (td->td_flags & (TDF_RUNNING|TDF_PREEMPT_LOCK))
+    while (td->td_flags & (TDF_RUNNING|TDF_PREEMPT_LOCK)) {
+#ifdef SMP
+ lwkt_process_ipiq();
+#endif
cpu_lfence();
+    }
td->td_gd = gd;
cpu_sfence();
td->td_flags &= ~TDF_MIGRATING;
```

#11 - 06/15/2008 08:03 PM - dragonfly1

Hello,

I've applied the patch and left the system running a "make -j8 buildworld" in a loop, cleaning out /usr/obj between each run.

It has been running for over 10 hours with no lock-ups with each build taking ~45mins. I have also tested -j32 while taxing the disk and CPU of the Linux host OS.

All now appears to be Ok.

Thanks

Gary

#12 - 06/15/2008 08:23 PM - dragonfly1

Hello Michel,

I attached the .vmdk file I've used for ages to run DragonFly. However I've tested it again and it hangs just after the boot menu, as does the 1.12.2 install CD.

I've done a fresh install of DFly since I last tried VirtualBox, the previous install had a custom make.conf and kernel config to build for a VIA C3 system and had been upgraded in-place for several versions.

VirtualBox like KVM is partly based on QEmu but I've looked on the website and it doesn't seem to support debugging of guest OSes. Any testing is going to be trial and error.

Sending a bug report a Sun is probably the quickest route to getting it fixed.

Regards

Gary

#13 - 06/16/2008 02:04 AM - dillon

```
:Hello,
:
:I've applied the patch and left the system running a "make -j8
:buildworld" in a loop, cleaning out /usr/obj between each run.
:
:It has been running for over 10 hours with no lock-ups with each build
:taking ~45mins. I have also tested -j32 while taxing the disk and CPU of
:the Linux host OS.
:
:All now appears to be Ok.
:
:Thanks
:
:Gary
```

The patch also appears to fix the system lockups I was experiencing

testing HAMMER. My worst-case stress test ran over the weekend without
any problems.

The two things in common here are that both KVM and HAMMER (under stress)
could impose very long latencies on particular cpus, creating a
deadlock between the IPI sending code and the thread cpu-migration
code.

-Matt
Matthew Dillon
<dillon@backplane.com>