

Re: scheduler (sched_4bsd) questions

Source: <http://unix.derkeiler.com/Mailing-Lists/FreeBSD/arch/2004-09/0165.html>

From: Stephan Uphoff (*ups_at_tree.com*)

Date: 09/21/04

To: Julian Elischer <julian@elischer.org>

Date: Tue, 21 Sep 2004 13:41:08 -0400

On Mon, 2004-09-20 at 21:02, Julian Elischer wrote:

> >

> >

> > *On Sunday 19 September 2004 03:10 am, Julian Elischer wrote:*

> >

> >

> > > *John Baldwin wrote:*

> > >

> > >

> > > > *On Saturday 18 September 2004 01:42 pm, Stephan Uphoff wrote:*

> > > >

> > > >

> > > > > *On Fri, 2004-09-17 at 21:20, Julian Elischer wrote:*

> > > > >

> > > > >

> > > > > > *Stephan Uphoff wrote:*

> > > > > >

> > > > > >

> > > > > > > *If this is true kernel threads can be preempted while holding*

> > > > > > > *for example the root vnode lock (or other important kernel*

> > > > > > > *resources) while not getting a chance to run until there are no more*

> > > > > > > *user processes with better priority.*

> > > > > >

> > > > > >

> > > > > > > *This is also true, though it is a slightly more complicated thing than*

> > > > > > > *that.*

> > > > > > > > *Preempting threads are usually interrupt threads and are thus usually*

> > > > > > > > *short lived,.*

> > > > > >

> > > > > >

> > > > > > > *But interrupt threads often wake up other threads ...*

> > > > > >

> > > > > >

> > > > > > > *That are lower priority and thus won't be preempted to. Instead, they*

> > > > > > > *run when the interrupt thread goes back to sleep after it finishes.*

> > > >

> > > >

freebsd-arch: Re: scheduler (sched_4bsd) questions

> >> *though that may still be before the original preempted thread gets run..*
> >>
> >> *that reminds me..*
> >> *John, we should add a flag to setrunqueue to add a preempted thread back at*
> >> *the FRONT of it's runqueue.. So that it gets a chance to use the rest of*
> >> *its slot..*
> >>
> >>
> >
> > *The scheduler can already detect this by noting that curthread is being passed*
> > *to sched_add() when it has quantum left and then put it at the head of the*
> > *queue for its priority but only with the remaining quanta. This only really*
> > *works for schedulers that actually track quanta, i.e., not 4BSD. Given that,*
> > *I think the scheduler is free to implement this how it chooses and doesn't*
> > *require another flag to setrunqueue().*
> >
>
> *Just for kicks I implemented this in my p4 branch to see if it would help the problem*
> *with atapi cdroms failing to work when preemption is turned on..*
>
> *but it didn't help..*
>
> *(it's a very strange problem.. turning on preemption makes my test machine's cdrom*
> *time out so much in boot that it takes 5 minutes to probe during boot.)*

Great news ! – I actually switched target machines just when this started to happen and thought the cdrom of the new target was bad and disconnected it ;-)

I reconnected it this morning and did a little debugging.
Looks like an interrupt race condition in the ata code.
Can you try the following patch?

Stephan

Index: sys/dev/ata/ata-lowlevel.c

```
=====  
RCS file: /cvsroot/src/sys/dev/ata/ata-lowlevel.c,v  
retrieving revision 1.47  
diff -u -r1.47 ata-lowlevel.c  
--- sys/dev/ata/ata-lowlevel.c 3 Sep 2004 12:10:44 -0000 1.47  
+++ sys/dev/ata/ata-lowlevel.c 21 Sep 2004 17:36:27 -0000  
@@ -88,11 +88,15 @@  
     (ATA_R_ATAPI | ATA_R_DMA | ATA_R_WRITE)))  
     request->flags &= ~ATA_R_DMA;  
  
+ ch->running = request;  
+  
+     switch (request->flags & (ATA_R_ATAPI | ATA_R_DMA)) {  
  
     /* ATA PIO data transfer and control commands */
```

Re: scheduler (sched_4bsd) questions

freebsd-arch: Re: scheduler (sched_4bsd) questions

```
default:
{
+
+
/* record command direction here as our request might be gone later */
int write = (request->flags & ATA_R_WRITE);

@@ -136,7 +140,7 @@
}

/* record the request as running and return for interrupt */
- ch->running = request;
+ // ch->running = request;
return ATA_OP_CONTINUES;

/* ATA DMA data transfer commands */
@@ -167,7 +171,7 @@
}

/* record the request as running and return for interrupt */
- ch->running = request;
+ //ch->running = request;
return ATA_OP_CONTINUES;

/* ATAPI PIO commands */
@@ -192,7 +196,7 @@

/* command interrupt device ? just return and wait for interrupt */
if ((request->device->param->config & ATA_DRQ_MASK) == ATA_DRQ_INTR) {
- ch->running = request;
+ // ch->running = request;
return ATA_OP_CONTINUES;
}

@@ -226,7 +230,7 @@
ATA_PROTO_ATAPI_12 ? 6 : 8);

/* record the request as running and return for interrupt */
- ch->running = request;
+ //ch->running = request;
return ATA_OP_CONTINUES;

case ATA_R_ATAPI|ATA_R_DMA:
@@ -292,9 +296,11 @@
}

/* record the request as running and return for interrupt */
- ch->running = request;
+ // ch->running = request;
return ATA_OP_CONTINUES;
}
}
```

freebsd-arch: Re: scheduler (sched_4bsd) questions

```
+  
+ ch->running = NULL;  
  
/* request finish here */  
if (ch->dma && ch->dma->flags & ATA_DMA_LOADED)
```

freebsd-arch@freebsd.org mailing list

<http://lists.freebsd.org/mailman/listinfo/freebsd-arch>

To unsubscribe, send any mail to "freebsd-arch-unsubscribe@freebsd.org"