

[PATCH] Configure IDE probe delays

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2004-07/6246.html>

From: Todd Poynor (tpoynor_at_mvista.com)

Date: 07/30/04

Date: Fri, 30 Jul 2004 12:11:00 -0700

To: B.Zolnierkiewicz@elka.pw.edu.pl, linux-kernel@vger.kernel.org

IDE initialization and probing makes numerous calls to sleep for 50 milliseconds while waiting for the interface to return probe status and such. The value is probably rather conservative for modern hardware: Tim Bird, David Singleton, and I have tried reducing this value to 1-5ms on various systems (desktop, laptop, and embedded MIPS with PCI-to-IDE bridge) with no observed ill effect (and delays in loops such as at `actual_try_to_identify()` continued to deliver results after a single delay in my trials). But I haven't looked into ATA standards or hardware specs enough to be very confident that these results would apply to most or all of today's IDE hardware.

The 50ms delays (I counted 82 of 'em on my system) add up to about 4 seconds of kernel startup time on my laptop and Tim reports about 5 seconds on his workstation. This is a more important matter for consumer electronics devices, where these delays may constitute a large fraction of the time required to get the gadget running. Embedded system designers have therefore asked for the ability to customize the value for known hardware. Tim discussed this during his OLS presentation on improving boot times.

Any comments on this suggested patch that allows kernel command line parameter `ide-delay=2` to set the probing delay to 2ms, or any insight into the risks involved in modifying this value? Another possibility would be to configure the value in the IDE interface and device drivers according to known hardware characteristics. Thanks.

```
--- linux-2.6.8-rc2-orig/drivers/ide/ide.c 2004-07-27 14:41:02.000000000 -0700
+++ linux-2.6.8-rc2-ide-delay/drivers/ide/ide.c 2004-07-29 18:38:59.000000000 -0700
@@ -196,6 +196,9 @@
```

```
EXPORT_SYMBOL(ide_hwifs);

+int ide_delay = 50; /* milliseconds */
+EXPORT_SYMBOL(ide_delay);
+
extern ide_driver_t idedefault_driver;
static void setup_driver_defaults(ide_driver_t *driver);
```

Linux-Kernel: [PATCH] Configure IDE probe delays

```

@@ -1766,6 +1769,12 @@
    }
#endif /* CONFIG_BLK_DEV_IDEPCI */

+ if (!strncmp(s, "ide-delay=", 10)) {
+ ide_delay = simple_strtoul(s+10, NULL, 0);
+ printk(" : Delay set to %dms\n", ide_delay);
+ return 1;
+ }
+
+ /*
+  * Look for drive options: "hdx="
+  */
--- linux-2.6.8-rc2-orig/drivers/ide/ide-iops.c 2004-07-27 14:28:26.000000000 -0700
+++ linux-2.6.8-rc2-ide-delay/drivers/ide/ide-iops.c 2004-07-29 18:07:49.000000000 -0700
@@ -30,6 +30,8 @@
#include <asm/io.h>
#include <asm/bitops.h>

+extern int ide_delay;
+
+/*
+ * Conventional PIO operations for ATA devices
+ */
@@ -767,7 +769,7 @@
    SELECT_MASK(drive, 1);
    if (IDE_CONTROL_REG)
        hwif->OUTB(drive->ctl, IDE_CONTROL_REG);
- msleep(50);
+ msleep(ide_delay);
    hwif->OUTB(WIN_IDENTIFY, IDE_COMMAND_REG);
    timeout = jiffies + WAIT_WORSTCASE;
    do {
@@ -775,9 +777,9 @@
        SELECT_MASK(drive, 0);
        return 0; /* drive timed-out */
    }
- msleep(50); /* give drive a breather */
+ msleep(ide_delay); /* give drive a breather */
    } while (hwif->INB(IDE_ALTSTATUS_REG) & BUSY_STAT);
- msleep(50); /* wait for IRQ and DRQ_STAT */
+ msleep(ide_delay); /* wait for IRQ and DRQ_STAT */
    if (!OK_STAT(hwif->INB(IDE_STATUS_REG), DRQ_STAT, BAD_R_STAT)) {
        SELECT_MASK(drive, 0);
        printk("%s: CHECK for good STATUS\n", drive->name);
@@ -827,7 +829,7 @@
    u8 stat;

// while (HWGROUP(drive)->busy)
-// msleep(50);

```

Linux-Kernel: [PATCH] Configure IDE probe delays

```
+// msleep(ide_delay);

#ifdef CONFIG_BLK_DEV_IDEDMA
    if (hwif->ide_dma_check) /* check if host supports DMA */
--- linux-2.6.8-rc2-orig/drivers/ide/ide-probe.c 2004-07-27 14:28:26.000000000 -0700
+++ linux-2.6.8-rc2-ide-delay/drivers/ide/ide-probe.c 2004-07-29 18:59:47.000000000 -0700
@@ -56,6 +56,8 @@
#include <asm/uaccess.h>
#include <asm/io.h>

+extern int ide_delay;
+
/**
 * generic_id - add a generic drive id
 * @drive: drive to make an ID block for
@@ -273,7 +275,7 @@
    u8 s = 0, a = 0;

    /* take a deep breath */
- msleep(50);
+ msleep(ide_delay);

    if (IDE_CONTROL_REG) {
        a = hwif->INB(IDE_ALTSTATUS_REG);
@@ -312,11 +314,11 @@
        return 1;
    }
    /* give drive a breather */
- msleep(50);
+ msleep(ide_delay);
    } while ((hwif->INB(hd_status)) & BUSY_STAT);

    /* wait for IRQ and DRQ_STAT */
- msleep(50);
+ msleep(ide_delay);
    if (OK_STAT((hwif->INB(IDE_STATUS_REG)), DRQ_STAT, BAD_R_STAT)) {
        unsigned long flags;

@@ -445,15 +447,15 @@
    /* needed for some systems
     * (e.g. crw9624 as drive0 with disk as slave)
     */
- msleep(50);
+ msleep(ide_delay);
    SELECT_DRIVE(drive);
- msleep(50);
+ msleep(ide_delay);
    if (hwif->INB(IDE_SELECT_REG) != drive->select.all && !drive->present) {
        if (drive->select.b.unit != 0) {
            /* exit with drive0 selected */
            SELECT_DRIVE(&hwif->drives[0]);
```

Linux-Kernel: [PATCH] Configure IDE probe delays

```

        /* allow BUSY_STAT to assert & clear */
- msleep(50);
+ msleep(ide_delay);
    }
    /* no i/f present: mmm.. this should be a 4 -ml */
    return 3;
@@ -476,14 +478,14 @@
        printk("%s: no response (status = 0x%02x), "
               "resetting drive\n", drive->name,
               hwif->INB(IDE_STATUS_REG));
- msleep(50);
+ msleep(ide_delay);
        hwif->OUTB(drive->select.all, IDE_SELECT_REG);
- msleep(50);
+ msleep(ide_delay);
        hwif->OUTB(WIN_SRST, IDE_COMMAND_REG);
        timeout = jiffies;
        while (((hwif->INB(IDE_STATUS_REG)) & BUSY_STAT) &&
               time_before(jiffies, timeout + WAIT_WORSTCASE))
- msleep(50);
+ msleep(ide_delay);
        rc = try_to_identify(drive, cmd);
    }
    if (rc == 1)
@@ -498,7 +500,7 @@
    if (drive->select.b.unit != 0) {
        /* exit with drive0 selected */
        SELECT_DRIVE(&hwif->drives[0]);
- msleep(50);
+ msleep(ide_delay);
        /* ensure drive irq is clear */
        (void) hwif->INB(IDE_STATUS_REG);
    }
@@ -515,7 +517,7 @@

        printk("%s: enabling %s --- ", hwif->name, drive->id->model);
        SELECT_DRIVE(drive);
- msleep(50);
+ msleep(ide_delay);
        hwif->OUTB(EXABYTE_ENABLE_NEST, IDE_COMMAND_REG);
        timeout = jiffies + WAIT_WORSTCASE;
        do {
@@ -523,10 +525,10 @@
            printk("failed (timeout)\n");
            return;
        }
- msleep(50);
+ msleep(ide_delay);
    } while ((hwif->INB(IDE_STATUS_REG)) & BUSY_STAT);

- msleep(50);

```

Linux-Kernel: [PATCH] Configure IDE probe delays

```
+ msleep(ide_delay);

    if (!OK_STAT((hwif->INB(IDE_STATUS_REG)), 0, BAD_STAT)) {
        printk("failed (status = 0x%02x)\n", hwif->INB(IDE_STATUS_REG));
@@ -767,7 +769,7 @@
        udelay(10);
        hwif->OUTB(8, hwif->io_ports[IDE_CONTROL_OFFSET]);
        do {
- msleep(50);
+ msleep(ide_delay);
            stat = hwif->INB(hwif->io_ports[IDE_STATUS_OFFSET]);
        } while ((stat & BUSY_STAT) && time_after(timeout, jiffies));

--- linux-2.6.8-rc2-orig/Documentation/ide.txt 2004-07-27 14:29:10.000000000 -0700
+++ linux-2.6.8-rc2-ide-delay/Documentation/ide.txt 2004-07-29 18:33:44.000000000 -0700
@@ -304,6 +304,9 @@
```

"ide=reverse" : formerly called to pci sub-system, but now local.

+ "ide-delay=xx" : set delay in milliseconds for initialization and
+ probing. Defaults to 50ms.

+

The following are valid ONLY on ide0 (except dc4030), which usually corresponds
to the first ATA interface found on the particular host, and the defaults for
the base,ctl ports must not be altered.

--

Todd Poynor
MontaVista Software

-

To unsubscribe from this list: send the line "unsubscribe linux-kernel" in
the body of a message to majordomo@vger.kernel.org
More majordomo info at <http://vger.kernel.org/majordomo-info.html>
Please read the FAQ at <http://www.tux.org/lkml/>