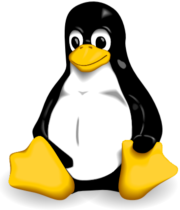
Contributing to the Linux Kernel

Omar Sandoval
osandov@osandov.com

CSE 451, Autumn 2015

November 23, 2015



Goals

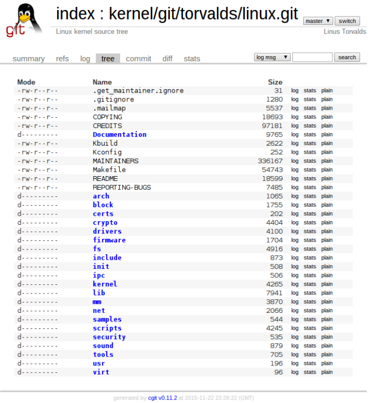
- Introduce you to what it’s like to work on a production operating system
- Shed some light on how Linux in particular is developed
- Give you a starting point if you’re interested in working on Linux

What is Linux?

- Operating system kernel written by Linus Torvalds, first released in 1991
- Free and open source (GNU GPLv2)
- General-purpose
  - Desktops, servers, smartphones, embedded systems
  - x86, ARM, MIPS, . . .

What is Linux? (cont.)

- Huge software engineering endeavor
- For 4.3:
  - Over 50,000 files
  - Almost 20,000,000 lines of code
  - About 1600 developers and 200 companies



index : kernel/git/torvalds/linux.git

Linux kernel source tree

Linus Torvalds

Mode	Name	Size			
-rw-r--r--	.get_maintainer.ignore	31	log	stats	plain
-rw-r--r--	.gitignore	1280	log	stats	plain
-rw-r--r--	.mailmap	5537	log	stats	plain
-rw-r--r--	COPYING	18693	log	stats	plain
-rw-r--r--	CREDITS	97181	log	stats	plain
d---------	Documentation	9765	log	stats	plain
-rw-r--r--	Kbuild	2622	log	stats	plain
-rw-r--r--	Kconfig	252	log	stats	plain
-rw-r--r--	MAINTAINERS	336167	log	stats	plain
-rw-r--r--	Makefile	54743	log	stats	plain
-rw-r--r--	README	18599	log	stats	plain
-rw-r--r--	REPORTING-BUGS	7485	log	stats	plain
d---------	arch	1065	log	stats	plain
d---------	block	1755	log	stats	plain
d---------	certs	202	log	stats	plain
d---------	crypto	4404	log	stats	plain
d---------	drivers	4100	log	stats	plain
d---------	firmware	1704	log	stats	plain
d---------	fs	4916	log	stats	plain
d---------	include	873	log	stats	plain
d---------	init	508	log	stats	plain
d---------	ipc	506	log	stats	plain
d---------	kernel	4265	log	stats	plain
d---------	lib	7941	log	stats	plain
d---------	mm	3870	log	stats	plain
d---------	net	2066	log	stats	plain
d---------	samples	544	log	stats	plain
d---------	scripts	4245	log	stats	plain
d---------	security	535	log	stats	plain
d---------	sound	879	log	stats	plain
d---------	tools	705	log	stats	plain
d---------	usr	196	log	stats	plain
d---------	virt	96	log	stats	plain

generated by cgit v0.11.2 at 2015-11-22 23:28:22 (GMT)

Requisites for Working on the Kernel

1. Technical knowledge
2. Familiarity with the development process
3. Motivation

Technical Requirements

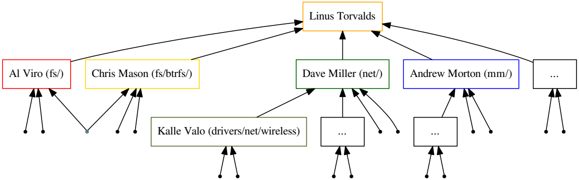
- C
- Unix shell
- Git

Technical Requirements

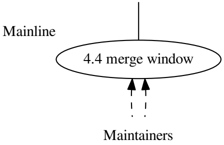
- C
- Unix shell
- Git
- Optional
  - Specific domain knowledge, e.g., filesystems, schedulers, memory management
  - Scripting, working with VMs

Development Process: Maintainers

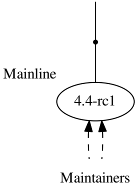
- Kernel source tree divided into subsystems with different maintainers, submaintainers



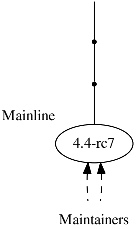
Development Process: Release Cycle



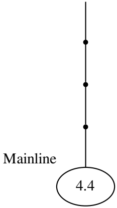
Development Process: Release Cycle



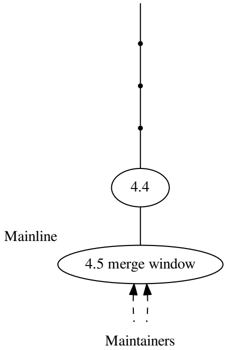
Development Process: Release Cycle



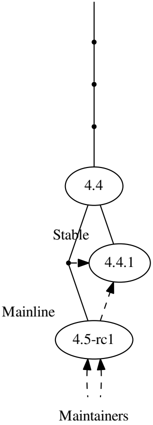
Development Process: Release Cycle



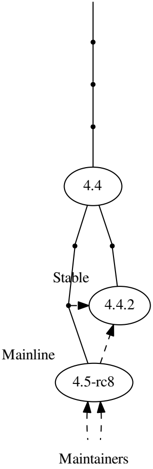
Development Process: Release Cycle



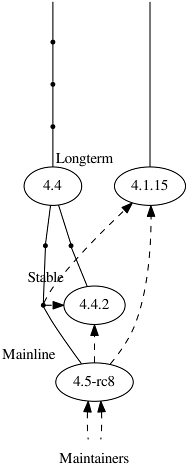
Development Process: Release Cycle



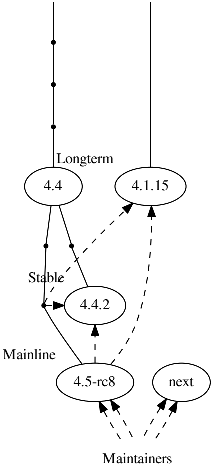
Development Process: Release Cycle



Development Process: Release Cycle



Development Process: Release Cycle



Development Process: Release Cycle Summary

Development Process: Release Cycle Summary

- Mainline
  - Linus Torvalds’s `linux.git` tree
  - Merge window
  - Weekly release candidates

Development Process: Release Cycle Summary

- Mainline
  - Linus Torvalds’s `linux.git` tree
  - Merge window
  - Weekly release candidates
- Stable
  - Greg Kroah-Hartman’s `linux-stable` tree
  - Backports of bug fixes for recent releases

Development Process: Release Cycle Summary

- Mainline
  - Linus Torvalds's `linux.git` tree
  - Merge window
  - Weekly release candidates
- Stable
  - Greg Kroah-Hartman's `linux-stable` tree
  - Backports of bug fixes for recent releases
- Longterm
  - Greg Kroah-Hartman, others
  - Backports of select versions for years

Development Process: Release Cycle Summary

- Mainline
  - Linus Torvalds's `linux.git` tree
  - Merge window
  - Weekly release candidates
- Stable
  - Greg Kroah-Hartman's `linux-stable` tree
  - Backports of bug fixes for recent releases
- Longterm
  - Greg Kroah-Hartman, others
  - Backports of select versions for years
- Next
  - Stephen Rothwell's `linux-next` tree
  - Integration tree combining everything maintainers have in their own trees for the next merge window

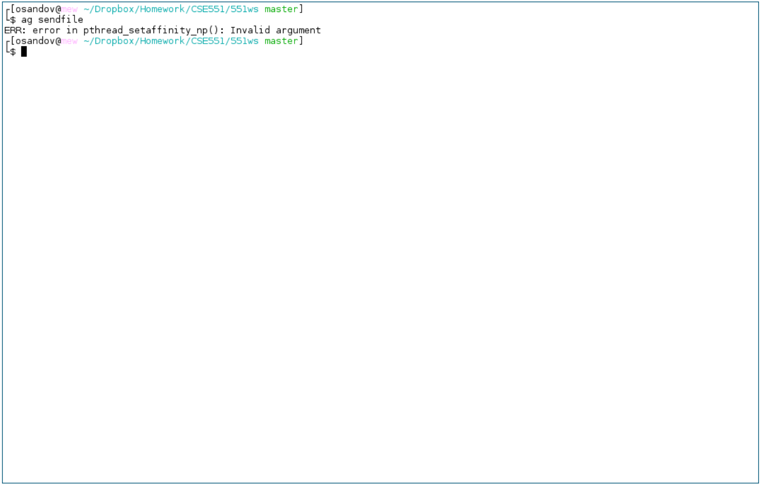
Motivation

- Can’t really help you with this one

Finding Something to Work On

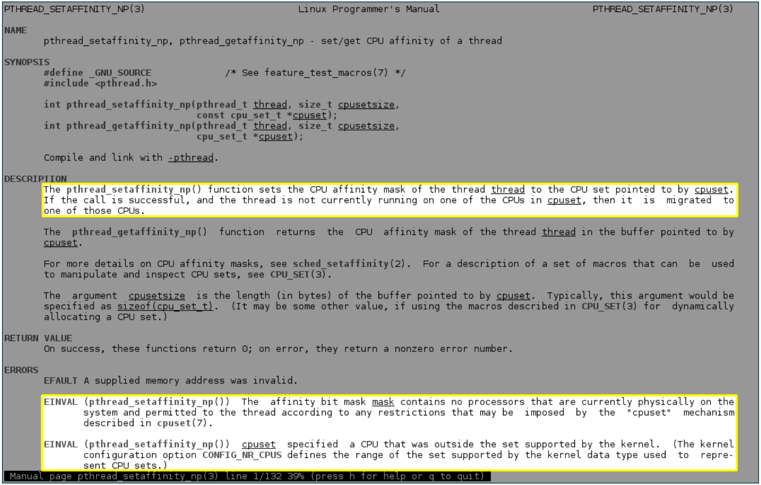
- Decide on a few specific subsystems
- Code cleanup (coding style, whitespace)
- Follow the mailing lists
- Try the release candidates
- Reproduce bugs

Case Study



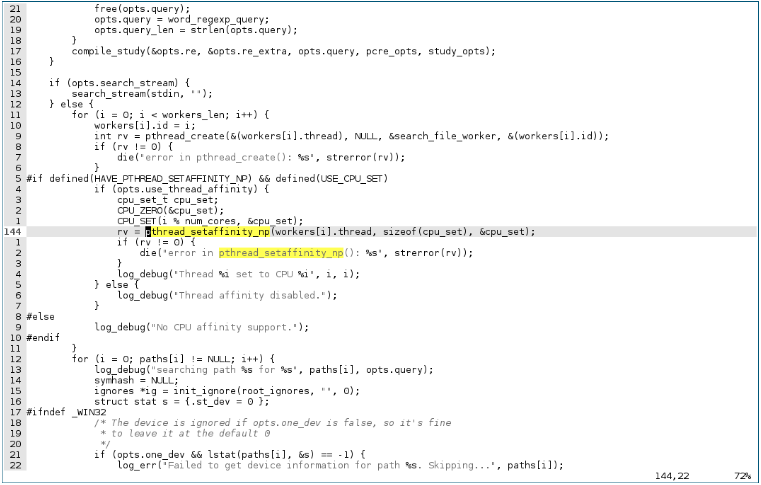
```
┌[osandov@mew ~/Dropbox/Homework/CSE551/551ws master]
└$ ag sendfile
ERR: error in pthread_setaffinity_np(): Invalid argument
┌[osandov@mew ~/Dropbox/Homework/CSE551/551ws master]
└$ █
```

Case Study



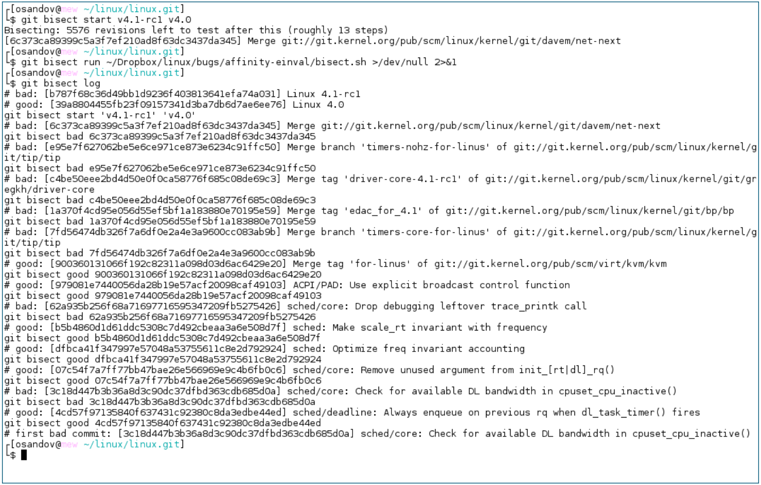
```
PTHREAD_SETAFFINITY_NP(3)              Linux Programmer's Manual              PTHREAD_SETAFFINITY_NP(3)

NAME
       pthread_setaffinity_np, pthread_getaffinity_np - set/get CPU affinity of a thread

SYNOPSIS
       #define _GNU_SOURCE             /* See feature_test_macros(7) */
       #include <pthread.h>

       int pthread_setaffinity_np(pthread_t thread, size_t cpusetsize,
                                  const cpu_set_t *cpuset);
       int pthread_getaffinity_np(pthread_t thread, size_t cpusetsize,
                                  cpu_set_t *cpuset);

       Compile and link with -pthread.

DESCRIPTION
       The pthread_setaffinity_np() function sets the CPU affinity mask of the thread thread to the CPU set pointed to by cpuset.
       If the call is successful, and the thread is not currently running on one of the CPUs in cpuset, then it  is  migrated  to
       one of those CPUs.

       The  pthread_getaffinity_np()  function  returns  the  CPU  affinity mask of the thread thread in the buffer pointed to by
       cpuset.

       For more details on CPU affinity masks, see sched_setaffinity(2).  For a description of a set of macros that can  be  used
       to manipulate and inspect CPU sets, see CPU_SET(3).

       The  argument  cpusetsize  is the length (in bytes) of the buffer pointed to by cpuset.  Typically, this argument would be
       specified as sizeof(cpu_set_t).  (It may be some other value, if using the macros described in CPU_SET(3) for  dynamically
       allocating a CPU set.)

RETURN VALUE
       On success, these functions return 0; on error, they return a nonzero error number.

ERRORS
       EFAULT A supplied memory address was invalid.

       EINVAL (pthread_setaffinity_np())  The  affinity bit mask mask contains no processors that are currently physically on the
              system and permitted to the thread according to any restrictions that may be  imposed  by  the  "cpuset"  mechanism
              described in cpuset(7).

       EINVAL (pthread_setaffinity_np())  cpuset  specified  a CPU that was outside the set supported by the kernel.  (The kernel
              configuration option CONFIG_NR_CPUS defines the range of the set supported by the kernel data type used  to  repre-
              sent CPU sets.)
 Manual page pthread_setaffinity_np(3) line 1/132 39% (press h for help or q to quit)
```

Case Study



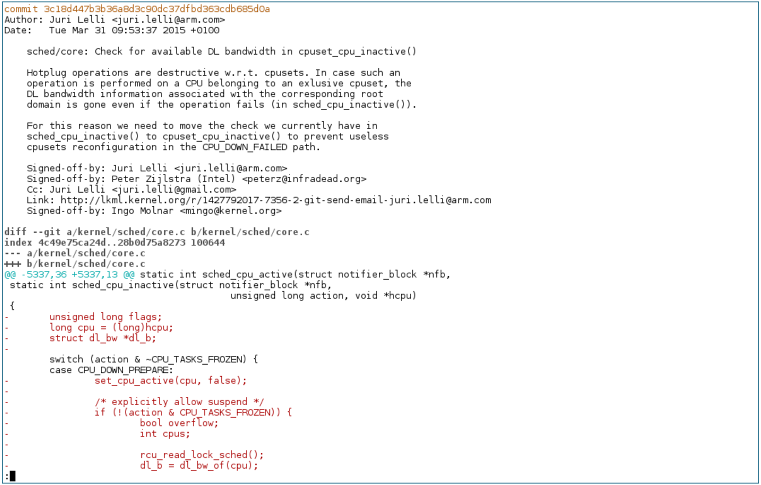
```
            free(opts.query);
            opts.query = word_regexp_query;
            opts.query_len = strlen(opts.query);
        }
        compile_study(&opts.re, &opts.re_extra, opts.query, pcre_opts, study_opts);
    }

    if (opts.search_stream) {
        search_stream(stdin, "");
    } else {
        for (i = 0; i < workers_len; i++) {
            workers[i].id = i;
            int rv = pthread_create(&(workers[i].thread), NULL, &search_file_worker, &(workers[i].id));
            if (rv != 0) {
                die("error in pthread_create(): %s", strerror(rv));
            }
#if defined(HAVE_PTHREAD_SETAFFINITY_NP) && defined(USE_CPU_SET)
            if (opts.use_thread_affinity) {
                cpu_set_t cpu_set;
                CPU_ZERO(&cpu_set);
                CPU_SET(i % num_cores, &cpu_set);
                rv = pthread_setaffinity_np(workers[i].thread, sizeof(cpu_set), &cpu_set);
                if (rv != 0) {
                    die("error in pthread_setaffinity_np(): %s", strerror(rv));
                }
                log_debug("Thread %i set to CPU %i", i, i);
            } else {
                log_debug("Thread affinity disabled.");
            }
#else
            log_debug("No CPU affinity support.");
#endif
        }
        for (i = 0; paths[i] != NULL; i++) {
            log_debug("searching path %s for %s", paths[i], opts.query);
            symhash = NULL;
            ignores *ig = init_ignore(root_ignores, "", 0);
            struct stat s = {.st_dev = 0 };
#ifndef _WIN32
            /* The device is ignored if opts.one_dev is false, so it's fine
             * to leave it at the default 0
             */
            if (opts.one_dev && lstat(paths[i], &s) == -1) {
                log_err("Failed to get device information for path %s. Skipping...", paths[i]);
```

Case Study



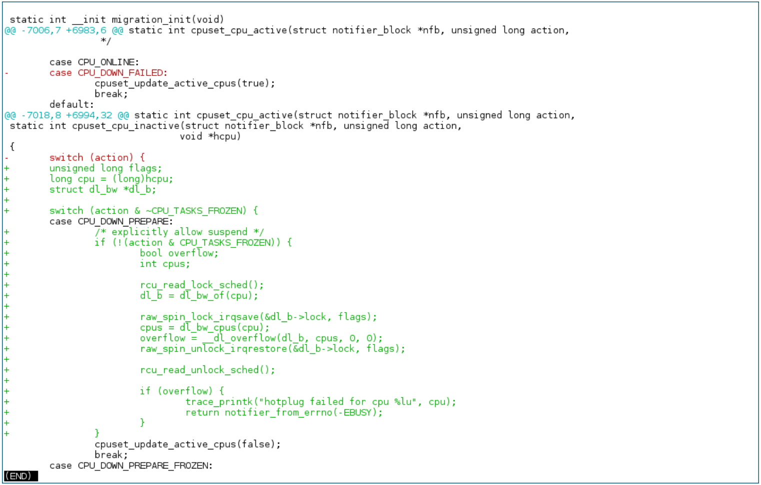
```
┌[osandov@mew ~/linux/linux.git]
└$ git bisect start v4.1-rc1 v4.0
Bisecting: 5576 revisions left to test after this (roughly 13 steps)
[6c373ca89399c5a3f7ef210ad8f63dc3437da345] Merge git://git.kernel.org/pub/scm/linux/kernel/git/davem/net-next
┌[osandov@mew ~/linux/linux.git]
└$ git bisect run ~/Dropbox/linux/bugs/affinity-einval/bisect.sh >/dev/null 2>&1
┌[osandov@mew ~/linux/linux.git]
└$ git bisect log
# bad: [b787f68c36d49bb1d9236f403813641efa74a031] Linux 4.1-rc1
# good: [39a8804455fb23f09157341d3ba7db6d7ae6ee76] Linux 4.0
git bisect start 'v4.1-rc1' 'v4.0'
# bad: [6c373ca89399c5a3f7ef210ad8f63dc3437da345] Merge git://git.kernel.org/pub/scm/linux/kernel/git/davem/net-next
git bisect bad 6c373ca89399c5a3f7ef210ad8f63dc3437da345
# bad: [e95e7f627062be5e6ce971ce873e6234c91ffc50] Merge branch 'timers-nohz-for-linus' of git://git.kernel.org/pub/scm/linux/kernel/g
it/tip/tip
git bisect bad e95e7f627062be5e6ce971ce873e6234c91ffc50
# bad: [c4be50eee2bd4d50e0f0ca58776f685c08de69c3] Merge tag 'driver-core-4.1-rc1' of git://git.kernel.org/pub/scm/linux/kernel/git/gr
egkh/driver-core
git bisect bad c4be50eee2bd4d50e0f0ca58776f685c08de69c3
# bad: [1a370f4cd95e056d55ef5bf1a183880e70195e59] Merge tag 'edac_for_4.1' of git://git.kernel.org/pub/scm/linux/kernel/git/bp/bp
git bisect bad 1a370f4cd95e056d55ef5bf1a183880e70195e59
# bad: [7fd56474db326f7a6df0e2a4e3a9600cc083ab9b] Merge branch 'timers-core-for-linus' of git://git.kernel.org/pub/scm/linux/kernel/g
it/tip/tip
git bisect bad 7fd56474db326f7a6df0e2a4e3a9600cc083ab9b
# good: [900360131066f192c82311a098d03d6ac6429e20] Merge tag 'for-linus' of git://git.kernel.org/pub/scm/virt/kvm/kvm
git bisect good 900360131066f192c82311a098d03d6ac6429e20
# good: [979081e7440056da28b19e57acf20098caf49103] ACPI/PAD: Use explicit broadcast control function
git bisect good 979081e7440056da28b19e57acf20098caf49103
# bad: [62a935b256f68a716977165953472O9fb5275426] sched/core: Drop debugging leftover trace_printk call
git bisect bad 62a935b256f68a7169771659534720 9fb5275426
# good: [b5b4860d1d61ddc5308c7d492cbeaa3a6e508d7f] sched: Make scale_rt invariant with frequency
git bisect good b5b4860d1d61ddc5308c7d492cbeaa3a6e508d7f
# good: [dfbca41f347997e57048a53755611c8e2d792924] sched: Optimize freq invariant accounting
git bisect good dfbca41f347997e57048a53755611c8e2d792924
# good: [07c54f7a7ff77bb47bae26e566969e9c4b6fb0c6] sched/core: Remove unused argument from init_[rt|dl]_rq()
git bisect good 07c54f7a7ff77bb47bae26e566969e9c4b6fb0c6
# bad: [3c18d447b3b36a8d3c90dc37dfbd363cdb685d0a] sched/core: Check for available DL bandwidth in cpuset_cpu_inactive()
git bisect bad 3c18d447b3b36a8d3c90dc37dfbd363cdb685d0a
# good: [4cd57f97135840f637431c92380c8da3edbe44ed] sched/deadline: Always enqueue on previous rq when dl_task_timer() fires
git bisect good 4cd57f97135840f637431c92380c8da3edbe44ed
# first bad commit: [3c18d447b3b36a8d3c90dc37dfbd363cdb685d0a] sched/core: Check for available DL bandwidth in cpuset_cpu_inactive()
┌[osandov@mew ~/linux/linux.git]
└$ █
```

Case Study



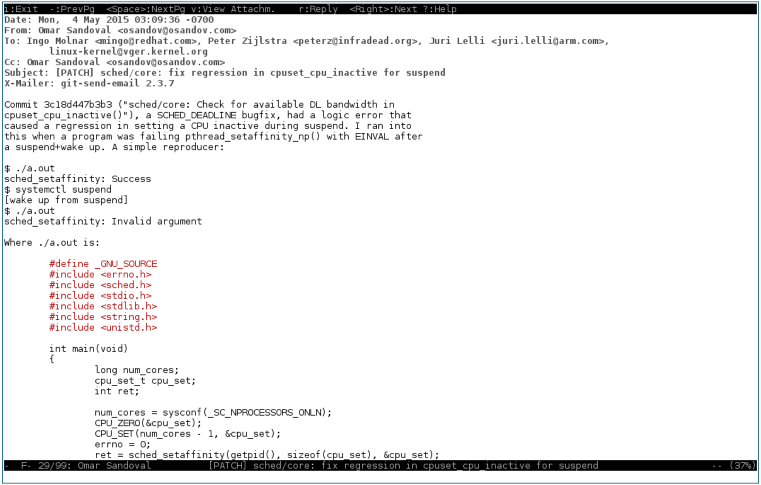
```
commit 3c18d447b3b36a8d3c90dc37dfbd363cdb685d0a
Author: Juri Lelli <juri.lelli@arm.com>
Date:   Tue Mar 31 09:53:37 2015 +0100

    sched/core: Check for available DL bandwidth in cpuset_cpu_inactive()

    Hotplug operations are destructive w.r.t. cpusets. In case such an
    operation is performed on a CPU belonging to an exlusive cpuset, the
    DL bandwidth information associated with the corresponding root
    domain is gone even if the operation fails (in sched_cpu_inactive()).

    For this reason we need to move the check we currently have in
    sched_cpu_inactive() to cpuset_cpu_inactive() to prevent useless
    cpusets reconfiguration in the CPU_DOWN_FAILED path.

    Signed-off-by: Juri Lelli <juri.lelli@arm.com>
    Signed-off-by: Peter Zijlstra (Intel) <peterz@infradead.org>
    Cc: Juri Lelli <juri.lelli@gmail.com>
    Link: http://lkml.kernel.org/r/1427792017-7356-2-git-send-email-juri.lelli@arm.com
    Signed-off-by: Ingo Molnar <mingo@kernel.org>

diff --git a/kernel/sched/core.c b/kernel/sched/core.c
index 4c49e75ca24d..28b0d75a8273 100644
--- a/kernel/sched/core.c
+++ b/kernel/sched/core.c
@@ -5337,36 +5337,13 @@ static int sched_cpu_active(struct notifier_block *nfb,
 static int sched_cpu_inactive(struct notifier_block *nfb,
                                        unsigned long action, void *hcpu)
 {
-       unsigned long flags;
-       long cpu = (long)hcpu;
-       struct dl_bw *dl_b;
-
        switch (action & ~CPU_TASKS_FROZEN) {
        case CPU_DOWN_PREPARE:
-               set_cpu_active(cpu, false);
-
-               /* explicitly allow suspend */
-               if (!(action & CPU_TASKS_FROZEN)) {
-                       bool overflow;
-                       int cpus;
-
-                       rcu_read_lock_sched();
-                       dl_b = dl_bw_of(cpu);
:
```

Case Study



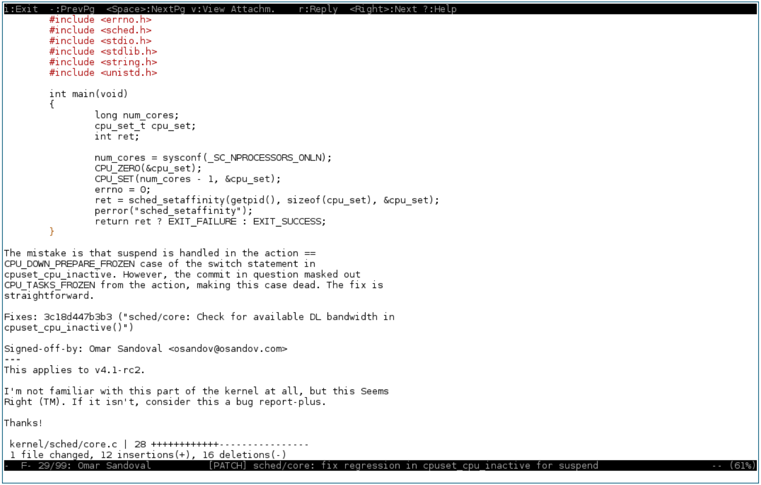
```
 static int __init migration_init(void)
@@ -7006,7 +6983,6 @@ static int cpuset_cpu_active(struct notifier_block *nfb, unsigned long action,
                 */

        case CPU_ONLINE:
-       case CPU_DOWN_FAILED:
                cpuset_update_active_cpus(true);
                break;
        default:
@@ -7018,8 +6994,32 @@ static int cpuset_cpu_active(struct notifier_block *nfb, unsigned long action,
 static int cpuset_cpu_inactive(struct notifier_block *nfb, unsigned long action,
                               void *hcpu)
 {
-       switch (action) {
+       unsigned long flags;
+       long cpu = (long)hcpu;
+       struct dl_bw *dl_b;
+
+       switch (action & ~CPU_TASKS_FROZEN) {
        case CPU_DOWN_PREPARE:
+               /* explicitly allow suspend */
+               if (!(action & CPU_TASKS_FROZEN)) {
+                       bool overflow;
+                       int cpus;
+
+                       rcu_read_lock_sched();
+                       dl_b = dl_bw_of(cpu);
+
+                       raw_spin_lock_irqsave(&dl_b->lock, flags);
+                       cpus = dl_bw_cpus(cpu);
+                       overflow = __dl_overflow(dl_b, cpus, 0, 0);
+                       raw_spin_unlock_irqrestore(&dl_b->lock, flags);
+
+                       rcu_read_unlock_sched();
+
+                       if (overflow) {
+                               trace_printk("hotplug failed for cpu %lu", cpu);
+                               return notifier_from_errno(-EBUSY);
+                       }
+               }
                cpuset_update_active_cpus(false);
                break;
        case CPU_DOWN_PREPARE_FROZEN:
(END)
```

Case Study



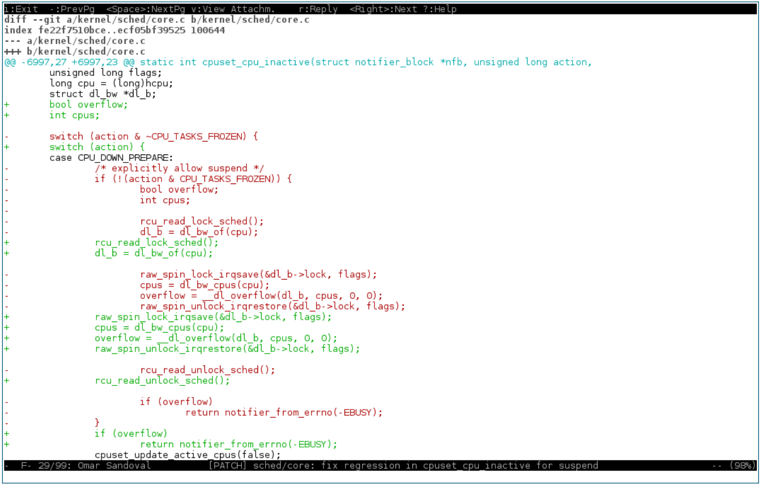
```
i:Exit  -:PrevPg  <Space>:NextPg v:View Attachm.    r:Reply  <Right>:Next ?:Help
Date: Mon,  4 May 2015 03:09:36 -0700
From: Omar Sandoval <osandov@osandov.com>
To: Ingo Molnar <mingo@redhat.com>, Peter Zijlstra <peterz@infradead.org>, Juri Lelli <juri.lelli@arm.com>,
        linux-kernel@vger.kernel.org
Cc: Omar Sandoval <osandov@osandov.com>
Subject: [PATCH] sched/core: fix regression in cpuset_cpu_inactive for suspend
X-Mailer: git-send-email 2.3.7

Commit 3c18d447b3b3 ("sched/core: Check for available DL bandwidth in
cpuset_cpu_inactive()"), a SCHED_DEADLINE bugfix, had a logic error that
caused a regression in setting a CPU inactive during suspend. I ran into
this when a program was failing pthread_setaffinity_np() with EINVAL after
a suspend+wake up. A simple reproducer:

$ ./a.out
sched_setaffinity: Success
$ systemctl suspend
[wake up from suspend]
$ ./a.out
sched_setaffinity: Invalid argument

Where ./a.out is:

        #define _GNU_SOURCE
        #include <errno.h>
        #include <sched.h>
        #include <stdio.h>
        #include <stdlib.h>
        #include <string.h>
        #include <unistd.h>

        int main(void)
        {
                long num_cores;
                cpu_set_t cpu_set;
                int ret;

                num_cores = sysconf(_SC_NPROCESSORS_ONLN);
                CPU_ZERO(&cpu_set);
                CPU_SET(num_cores - 1, &cpu_set);
                errno = 0;
                ret = sched_setaffinity(getpid(), sizeof(cpu_set), &cpu_set);
-  F- 29/99: Omar Sandoval          [PATCH] sched/core: fix regression in cpuset_cpu_inactive for suspend        -- (37%)
```

Case Study



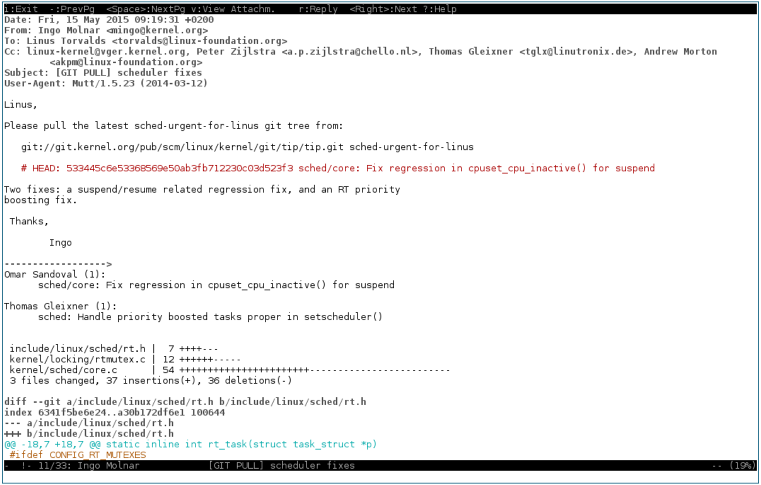
```
i:Exit  -:PrevPg  <Space>:NextPg v:View Attachm.    r:Reply  <Right>:Next ?:Help
        #include <errno.h>
        #include <sched.h>
        #include <stdio.h>
        #include <stdlib.h>
        #include <string.h>
        #include <unistd.h>

        int main(void)
        {
                long num_cores;
                cpu_set_t cpu_set;
                int ret;

                num_cores = sysconf(_SC_NPROCESSORS_ONLN);
                CPU_ZERO(&cpu_set);
                CPU_SET(num_cores - 1, &cpu_set);
                errno = 0;
                ret = sched_setaffinity(getpid(), sizeof(cpu_set), &cpu_set);
                perror("sched_setaffinity");
                return ret ? EXIT_FAILURE : EXIT_SUCCESS;
        }

The mistake is that suspend is handled in the action ==
CPU_DOWN_PREPARE_FROZEN case of the switch statement in
cpuset_cpu_inactive. However, the commit in question masked out
CPU_TASKS_FROZEN from the action, making this case dead. The fix is
straightforward.

Fixes: 3c18d447b3b3 ("sched/core: Check for available DL bandwidth in
cpuset_cpu_inactive()")

Signed-off-by: Omar Sandoval <osandov@osandov.com>
---
This applies to v4.1-rc2.

I'm not familiar with this part of the kernel at all, but this Seems
Right (TM). If it isn't, consider this a bug report-plus.

Thanks!

 kernel/sched/core.c | 28 ++++++++++++----------------
 1 file changed, 12 insertions(+), 16 deletions(-)
-  F- 29/99: Omar Sandoval          [PATCH] sched/core: fix regression in cpuset_cpu_inactive for suspend          -- (61%)
```

Case Study



```
i:Exit  -:PrevPg  <Space>:NextPg v:View Attachm.   r:Reply  <Right>:Next ?:Help
diff --git a/kernel/sched/core.c b/kernel/sched/core.c
index fe22f7510bce..ecf05bf39525 100644
--- a/kernel/sched/core.c
+++ b/kernel/sched/core.c
@@ -6997,27 +6997,23 @@ static int cpuset_cpu_inactive(struct notifier_block *nfb, unsigned long action,
        unsigned long flags;
        long cpu = (long)hcpu;
        struct dl_bw *dl_b;
+       bool overflow;
+       int cpus;

-       switch (action & ~CPU_TASKS_FROZEN) {
+       switch (action) {
        case CPU_DOWN_PREPARE:
-               /* explicitly allow suspend */
-               if (!(action & CPU_TASKS_FROZEN)) {
-                       bool overflow;
-                       int cpus;
-
-                       rcu_read_lock_sched();
-                       dl_b = dl_bw_of(cpu);
+               rcu_read_lock_sched();
+               dl_b = dl_bw_of(cpu);

-                       raw_spin_lock_irqsave(&dl_b->lock, flags);
-                       cpus = dl_bw_cpus(cpu);
-                       overflow = __dl_overflow(dl_b, cpus, 0, 0);
-                       raw_spin_unlock_irqrestore(&dl_b->lock, flags);
+               raw_spin_lock_irqsave(&dl_b->lock, flags);
+               cpus = dl_bw_cpus(cpu);
+               overflow = __dl_overflow(dl_b, cpus, 0, 0);
+               raw_spin_unlock_irqrestore(&dl_b->lock, flags);

-                       rcu_read_unlock_sched();
+               rcu_read_unlock_sched();

-                       if (overflow)
-                               return notifier_from_errno(-EBUSY);
-               }
+               if (overflow)
+                       return notifier_from_errno(-EBUSY);
                cpuset_update_active_cpus(false);
-  F- 29/99: Omar Sandoval          [PATCH] sched/core: fix regression in cpuset_cpu_inactive for suspend                     -- (98%)
```

Case Study



```
i:Exit  -:PrevPg  <Space>:NextPg v:View Attachm.    r:Reply  <Right>:Next ?:Help
Date: Fri, 15 May 2015 09:19:31 +0200
From: Ingo Molnar <mingo@kernel.org>
To: Linus Torvalds <torvalds@linux-foundation.org>
Cc: linux-kernel@vger.kernel.org, Peter Zijlstra <a.p.zijlstra@chello.nl>, Thomas Gleixner <tglx@linutronix.de>, Andrew Morton
        <akpm@linux-foundation.org>
Subject: [GIT PULL] scheduler fixes
User-Agent: Mutt/1.5.23 (2014-03-12)

Linus,

Please pull the latest sched-urgent-for-linus git tree from:

   git://git.kernel.org/pub/scm/linux/kernel/git/tip/tip.git sched-urgent-for-linus

   # HEAD: 533445c6e53368569e50ab3fb712230c03d523f3 sched/core: Fix regression in cpuset_cpu_inactive() for suspend

Two fixes: a suspend/resume related regression fix, and an RT priority
boosting fix.

 Thanks,

        Ingo

------------------>
Omar Sandoval (1):
      sched/core: Fix regression in cpuset_cpu_inactive() for suspend

Thomas Gleixner (1):
      sched: Handle priority boosted tasks proper in setscheduler()


 include/linux/sched/rt.h |  7 ++++---
 kernel/locking/rtmutex.c | 12 ++++++-----
 kernel/sched/core.c      | 54 ++++++++++++++++++++++++--------------------------
 3 files changed, 37 insertions(+), 36 deletions(-)

diff --git a/include/linux/sched/rt.h b/include/linux/sched/rt.h
index 6341f5be6e24..a30b172df6e1 100644
--- a/include/linux/sched/rt.h
+++ b/include/linux/sched/rt.h
@@ -18,7 +18,7 @@ static inline int rt_task(struct task_struct *p)
 #ifdef CONFIG_RT_MUTEXES
-  !- 11/33: Ingo Molnar            [GIT PULL] scheduler fixes                                                      -- (19%)
```

Where to Go From Here

- Look at `Documentation/HOWTO`
- Build and install a kernel from scratch
- Watch “Write and Submit your first Linux kernel Patch” on YouTube
- Take the Eudyptula Challenge
- Read LWN.net
- Email me at osandov@osandov.com