
Subject: [PATCH 5/10] BC: user interface (syscalls)
Posted by [Kirill Korotaev](#) on Thu, 05 Oct 2006 15:50:57 GMT
[View Forum Message](#) <> [Reply to Message](#)

Add the following system calls for BC management:

1. sys_get_bcid - get current BC id
2. sys_set_bcid - change BC on task
3. sys_set_bclimit - set limits for resources consumptions
4. sys_get_bcstat - return br_resource_parm on resource

Signed-off-by: Pavel Emelianov <xemul@openvz.org>

Signed-off-by: Kirill Korotaev <dev@openvz.org>

```
arch/i386/kernel/syscall_table.S | 4 +
arch/ia64/kernel/entry.S         | 4 +
arch/sparc/kernel/entry.S        | 2
arch/sparc/kernel/systbls.S      | 6 +
arch/sparc64/kernel/entry.S      | 2
arch/sparc64/kernel/systbls.S    | 10 ++-
include/asm-i386/unistd.h        | 6 +
include/asm-ia64/unistd.h        | 6 +
include/asm-powerpc/systbl.h     | 4 +
include/asm-powerpc/unistd.h     | 6 +
include/asm-sparc/unistd.h       | 4 +
include/asm-sparc64/unistd.h     | 4 +
include/asm-x86_64/unistd.h      | 10 ++-
kernel/bc/sys.c                  | 130 +++++
kernel/sys_ni.c                  | 6 +
15 files changed, 195 insertions(+), 9 deletions(-)
```

--- ./arch/i386/kernel/syscall_table.S.bc_syscalls 2006-10-05 11:42:39.000000000 +0400

+++ ./arch/i386/kernel/syscall_table.S 2006-10-05 12:09:09.000000000 +0400

@@ -322,3 +322,7 @@ ENTRY(sys_call_table)

.long sys_kevent_get_events /* 320 */

.long sys_kevent_ctl

.long sys_kevent_wait

+ .long sys_get_bcid

+ .long sys_set_bcid

+ .long sys_set_bclimit /* 325 */

+ .long sys_get_bcstat

--- ./arch/ia64/kernel/entry.S.bc_syscalls 2006-10-05 11:42:39.000000000 +0400

+++ ./arch/ia64/kernel/entry.S 2006-10-05 12:09:09.000000000 +0400

@@ -1610,5 +1610,9 @@ sys_call_table:

data8 sys_sync_file_range // 1300

data8 sys_tee

data8 sys_vmsplice

```

+ data8 sys_get_bcid
+ data8 sys_set_bcid
+ data8 sys_set_bclimit // 1305
+ data8 sys_get_bcstat

.org sys_call_table + 8*NR_syscalls // guard against failures to increase NR_syscalls
--- ./arch/sparc/kernel/entry.S.bc_syscalls 2006-09-20 14:46:17.000000000 +0400
+++ ./arch/sparc/kernel/entry.S 2006-10-05 12:09:09.000000000 +0400
@@ -37,7 +37,7 @@

#define curptr    g6

-#define NR_SYSCALLS 300    /* Each OS is different... */
+#define NR_SYSCALLS 304    /* Each OS is different... */

/* These are just handy. */
#define _SV save %sp, -STACKFRAME_SZ, %sp
--- ./arch/sparc/kernel/systbls.S.bc_syscalls 2006-09-20 14:46:17.000000000 +0400
+++ ./arch/sparc/kernel/systbls.S 2006-10-05 12:09:09.000000000 +0400
@@ -78,7 +78,8 @@ sys_call_table:
/*285*/ .long sys_mkdirat, sys_mknodat, sys_fchownat, sys_futimesat, sys_fstatat64
/*290*/ .long sys_unlinkat, sys_renameat, sys_linkat, sys_symlinkat, sys_readlinkat
/*295*/ .long sys_fchmodat, sys_faccessat, sys_pselect6, sys_ppoll, sys_unshare
-/*300*/ .long sys_set_robust_list, sys_get_robust_list
+/*300*/ .long sys_set_robust_list, sys_get_robust_list, sys_get_bcid, sys_set_bcid,
sys_set_bclimit
+/*305*/ .long sys_get_bcstat

#ifdef CONFIG_SUNOS_EMUL
/* Now the SunOS syscall table. */
@@ -192,4 +193,7 @@ sunos_sys_table:
.long sunos_nosys, sunos_nosys, sunos_nosys
.long sunos_nosys, sunos_nosys, sunos_nosys

+ .long sunos_nosys, sunos_nosys, sunos_nosys,
+ .long sunos_nosys
+
#endif
--- ./arch/sparc64/kernel/entry.S.bc_syscalls 2006-09-20 14:46:17.000000000 +0400
+++ ./arch/sparc64/kernel/entry.S 2006-10-05 12:09:09.000000000 +0400
@@ -25,7 +25,7 @@

#define curptr    g6

-#define NR_SYSCALLS 300    /* Each OS is different... */
+#define NR_SYSCALLS 304    /* Each OS is different... */

.text

```

```

.align 32
--- ./arch/sparc64/kernel/systbls.S.bc_syscalls 2006-09-20 14:46:17.000000000 +0400
+++ ./arch/sparc64/kernel/systbls.S 2006-10-05 12:09:09.000000000 +0400
@@ -79,7 +79,8 @@ sys_call_table32:
.word sys_mkdirat, sys_mknodat, sys_fchownat, compat_sys_futimesat, compat_sys_fstatat64
/*290*/.word sys_unlinkat, sys_renameat, sys_linkat, sys_symlinkat, sys_readlinkat
.word sys_fchmodat, sys_faccessat, compat_sys_pselect6, compat_sys_ppoll, sys_unshare
/*300*/.word compat_sys_set_robust_list, compat_sys_get_robust_list
+/*300*/.word compat_sys_set_robust_list, compat_sys_get_robust_list, sys_nis_syscall,
sys_nis_syscall, sys_nis_syscall
+ .word sys_nis_syscall

#endif /* CONFIG_COMPAT */

@@ -149,7 +150,9 @@ sys_call_table:
.word sys_mkdirat, sys_mknodat, sys_fchownat, sys_futimesat, sys_fstatat64
/*290*/.word sys_unlinkat, sys_renameat, sys_linkat, sys_symlinkat, sys_readlinkat
.word sys_fchmodat, sys_faccessat, sys_pselect6, sys_ppoll, sys_unshare
/*300*/.word sys_set_robust_list, sys_get_robust_list
+/*300*/.word sys_set_robust_list, sys_get_robust_list, sys_get_bcid, sys_set_bcid,
sys_set_bclimit
+ .word sys_get_bcstat
+

#if defined(CONFIG_SUNOS_EMUL) || defined(CONFIG_SOLARIS_EMUL) || \
    defined(CONFIG_SOLARIS_EMUL_MODULE)
@@ -263,4 +266,7 @@ sunos_sys_table:
.word sunos_nosys, sunos_nosys, sunos_nosys
.word sunos_nosys, sunos_nosys, sunos_nosys
.word sunos_nosys, sunos_nosys, sunos_nosys
+
+ .word sunos_nosys, sunos_nosys, sunos_nosys
+ .word sunos_nosys
#endif
--- ./include/asm-i386/unistd.h.bc_syscalls 2006-10-05 11:42:42.000000000 +0400
+++ ./include/asm-i386/unistd.h 2006-10-05 12:09:09.000000000 +0400
@@ -328,10 +328,14 @@
#define __NR_kevent_get_events 320
#define __NR_kevent_ctl 321
#define __NR_kevent_wait 322
+#define __NR_get_bcid 323
+#define __NR_set_bcid 324
+#define __NR_set_bclimit 325
+#define __NR_get_bcstat 326

#ifdef __KERNEL__

-#define NR_syscalls 323

```

```

+#define NR_syscalls 327
#include <linux/err.h>

/*
--- ./include/asm-ia64/unistd.h.bc_syscalls 2006-10-05 11:42:42.000000000 +0400
+++ ./include/asm-ia64/unistd.h 2006-10-05 12:09:09.000000000 +0400
@@ -291,11 +291,15 @@
#define __NR_sync_file_range 1300
#define __NR_tee 1301
#define __NR_vmsplice 1302
+#define __NR_get_bcid 1303
+#define __NR_set_bcid 1304
+#define __NR_set_bclimit 1305
+#define __NR_get_bcstat 1306

#ifdef __KERNEL__

-#define NR_syscalls 279 /* length of syscall table */
+#define NR_syscalls 283 /* length of syscall table */

#define __ARCH_WANT_SYS_RT_SIGACTION

--- ./include/asm-powerpc/systbl.h.bc_syscalls 2006-09-20 14:46:39.000000000 +0400
+++ ./include/asm-powerpc/systbl.h 2006-10-05 12:09:09.000000000 +0400
@@ -304,3 +304,7 @@ SYSCALL_SPU(fchmodat)
SYSCALL_SPU(faccessat)
COMPAT_SYS_SPU(get_robust_list)
COMPAT_SYS_SPU(set_robust_list)
+SYSCALL(sys_get_bcid)
+SYSCALL(sys_set_bcid)
+SYSCALL(sys_set_bclimit)
+SYSCALL(sys_get_bcstat)
--- ./include/asm-powerpc/unistd.h.bc_syscalls 2006-10-05 11:42:42.000000000 +0400
+++ ./include/asm-powerpc/unistd.h 2006-10-05 12:09:09.000000000 +0400
@@ -323,10 +323,14 @@
#define __NR_faccessat 298
#define __NR_get_robust_list 299
#define __NR_set_robust_list 300
+#define __NR_get_bcid 301
+#define __NR_set_bcid 302
+#define __NR_set_bclimit 303
+#define __NR_get_bcstat 304

#ifdef __KERNEL__

-#define __NR_syscalls 301
+#define __NR_syscalls 305

```

```

#define __NR_exit __NR_exit
#define NR_syscalls __NR_syscalls
--- ./include/asm-sparc/unistd.h.bc_syscalls 2006-10-05 11:42:43.000000000 +0400
+++ ./include/asm-sparc/unistd.h 2006-10-05 12:09:09.000000000 +0400
@@ -318,6 +318,10 @@
#define __NR_unshare 299
#define __NR_set_robust_list 300
#define __NR_get_robust_list 301
+#define __NR_get_bcid 302
+#define __NR_set_bcid 303
+#define __NR_set_bclimit 304
+#define __NR_get_bcstat 305

#ifdef __KERNEL__
/* WARNING: You MAY NOT add syscall numbers larger than 301, since
--- ./include/asm-sparc64/unistd.h.bc_syscalls 2006-10-05 11:42:43.000000000 +0400
+++ ./include/asm-sparc64/unistd.h 2006-10-05 12:09:09.000000000 +0400
@@ -320,6 +320,10 @@
#define __NR_unshare 299
#define __NR_set_robust_list 300
#define __NR_get_robust_list 301
+#define __NR_get_bcid 302
+#define __NR_set_bcid 303
+#define __NR_set_bclimit 304
+#define __NR_get_bcstat 305

#ifdef __KERNEL__
/* WARNING: You MAY NOT add syscall numbers larger than 301, since
--- ./include/asm-x86_64/unistd.h.bc_syscalls 2006-10-05 11:42:43.000000000 +0400
+++ ./include/asm-x86_64/unistd.h 2006-10-05 12:09:09.000000000 +0400
@@ -625,8 +625,16 @@ __SYSCALL(__NR_kevent_get_events, sys_ke
__SYSCALL(__NR_kevent_ctl, sys_kevent_ctl)
#define __NR_kevent_wait 282
__SYSCALL(__NR_kevent_wait, sys_kevent_wait)
+#define __NR_get_bcid 283
+__SYSCALL(__NR_get_bcid, sys_get_bcid)
+#define __NR_set_bcid 284
+__SYSCALL(__NR_set_bcid, sys_set_bcid)
+#define __NR_set_bclimit 285
+__SYSCALL(__NR_set_bclimit, sys_set_bclimit)
+#define __NR_get_bcstat 286
+__SYSCALL(__NR_get_bcstat, sys_get_bcstat)

-#define __NR_syscall_max __NR_kevent_wait
+#define __NR_syscall_max __NR_get_bcstat

#ifdef __KERNEL__

```

```

#include <linux/err.h>
--- /dev/null 2006-07-18 14:52:43.075228448 +0400
+++ ./kernel/bc/sys.c 2006-10-05 12:10:31.000000000 +0400
@@ -0,0 +1,130 @@
+/*
+ * kernel/bc/sys.c
+ *
+ * Copyright (C) 2006 OpenVZ SWsoft Inc
+ *
+ */
+
+#include <linux/sched.h>
+
+#include <asm/uaccess.h>
+
+#include <bc/beancounter.h>
+#include <bc/task.h>
+
+asmlinkage long sys_get_bcid(void)
+{
+ struct beancounter *bc;
+
+ bc = get_exec_bc();
+ return bc->bc_id;
+}
+
+asmlinkage long sys_set_bcid(bcid_t id, pid_t pid)
+{
+ int error;
+ struct task_struct *tsk;
+ struct beancounter *bc;
+
+ error = -EPERM;
+ if (!capable(CAP_SETUID))
+ goto out;
+
+ error = -ENOMEM;
+ bc = bc_findcreate(id, BC_ALLOC);
+ if (bc == NULL)
+ goto out;
+
+ read_lock(&tasklist_lock);
+ tsk = find_task_by_pid(pid);
+ if (tsk != NULL)
+ get_task_struct(tsk);
+ read_unlock(&tasklist_lock);
+
+ error = -ESRCH;

```

```

+ if (tsk == NULL)
+ goto out_putbc;
+
+ error = bc_task_move(tsk, bc);
+
+ put_task_struct(tsk);
+out_putbc:
+ bc_put(bc);
+out:
+ return error;
+}
+
+asmlinkage long sys_set_bclimit(bcid_t id, unsigned long resource,
+ unsigned long __user *limits)
+{
+ int error;
+ struct beancounter *bc;
+ unsigned long new_limits[2];
+
+ error = -EPERM;
+ if(!capable(CAP_SYS_RESOURCE))
+ goto out;
+
+ error = -EINVAL;
+ if (resource >= BC_RESOURCES)
+ goto out;
+
+ error = -EFAULT;
+ if (copy_from_user(&new_limits, limits, sizeof(new_limits)))
+ goto out;
+
+ error = -EINVAL;
+ if (new_limits[0] > BC_MAXVALUE || new_limits[1] > BC_MAXVALUE ||
+ new_limits[0] > new_limits[1])
+ goto out;
+
+ error = -ENOENT;
+ bc = bc_findcreate(id, BC_LOOKUP);
+ if (bc == NULL)
+ goto out;
+
+ spin_lock_irq(&bc->bc_lock);
+ error = 0;
+ if (bc_resources[resource]->bcr_change)
+ error = bc_resources[resource]->bcr_change(bc,
+ new_limits[0], new_limits[1]);
+ if (error < 0)
+ goto out_unlock;

```

```

+
+ bc->bc_parms[resource].barrier = new_limits[0];
+ bc->bc_parms[resource].limit = new_limits[1];
+ spin_unlock_irq(&bc->bc_lock);
+out_unlock:
+ bc_put(bc);
+out:
+ return error;
+}
+
+asmlinkage long sys_get_bcstat(bcid_t id, unsigned long resource,
+ struct bc_resource_parm __user *uparm)
+{
+ int error;
+ struct beancounter *bc;
+ struct bc_resource_parm parm;
+
+ error = -EINVAL;
+ if (resource >= BC_RESOURCES)
+ goto out;
+
+ error = -ENOENT;
+ bc = bc_findcreate(id, BC_LOOKUP);
+ if (bc == NULL)
+ goto out;
+
+ spin_lock_irq(&bc->bc_lock);
+ parm = bc->bc_parms[resource];
+ spin_unlock_irq(&bc->bc_lock);
+ bc_put(bc);
+
+ error = 0;
+ if (copy_to_user(uparm, &parm, sizeof(parm)))
+ error = -EFAULT;
+
+out:
+ return error;
+}
--- ./kernel/sys_ni.c.bc_syscalls 2006-10-05 11:42:43.000000000 +0400
+++ ./kernel/sys_ni.c 2006-10-05 12:09:09.000000000 +0400
@@ -143,3 +143,9 @@ cond_syscall(compat_sys_move_pages);
cond_syscall(sys_bdflush);
cond_syscall(sys_ioprio_set);
cond_syscall(sys_ioprio_get);
+
+/* user resources syscalls */
+cond_syscall(sys_set_bcid);
+cond_syscall(sys_get_bcid);

```

```
+cond_syscall(sys_set_bclimit);  
+cond_syscall(sys_get_bcstat);
```
