Subject: [PATCH 5/7] BC: user interface (syscalls)
Posted by dev on Tue, 29 Aug 2006 14:52:39 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 exec_ and fork_ BCs on current
3. sys_set_bclimit - set limits for resources consumtions
4. sys_get_bcstat - return br_resource_parm on resource

Signed-off-by: Pavel Emelianov <xemul@sw.ru>
Signed-off-by: Kirill Korotaev <dev@sw.ru>

```
 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/Makefile               |    1
 kernel/bc/sys.c                  |  120 ++++++++++++++++++++++++++++++++++++++++
 kernel/sys_ni.c                  |    6 +
 16 files changed, 186 insertions(+), 9 deletions(-)

--- ./arch/i386/kernel/syscall_table.S.ve3 2006-08-21 13:15:37.000000000 +0400
+++ ./arch/i386/kernel/syscall_table.S 2006-08-21 14:15:47.000000000 +0400
@@ -318,3 +318,7 @@ ENTRY(sys_call_table)
  .long sys_vmsplice
  .long sys_move_pages
  .long sys_getcpu
+ .long sys_get_bcid
+ .long sys_set_bcid  /* 320 */
+ .long sys_set_bclimit
+ .long sys_get_bcstat
--- ./arch/ia64/kernel/entry.S.ve3 2006-08-21 13:15:37.000000000 +0400
+++ ./arch/ia64/kernel/entry.S 2006-08-21 14:17:07.000000000 +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.ve3 2006-07-10 12:39:10.000000000 +0400
+++ ./arch/sparc/kernel/entry.S 2006-08-21 14:29:44.000000000 +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.ve3 2006-07-10 12:39:10.000000000 +0400
+++ ./arch/sparc/kernel/systbls.S 2006-08-21 14:30:43.000000000 +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.ve3 2006-07-10 12:39:10.000000000 +0400
+++ ./arch/sparc64/kernel/entry.S 2006-08-21 14:29:56.000000000 +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.ve3 2006-07-10 12:39:11.000000000 +0400
+++ ./arch/sparc64/kernel/systbls.S 2006-08-21 14:32:26.000000000 +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.ve3 2006-08-21 13:15:39.000000000 +0400
+++ ./include/asm-i386/unistd.h 2006-08-21 14:22:53.000000000 +0400
@@ -324,10 +324,14 @@
 #define __NR_vmsplice  316
 #define __NR_move_pages  317
 #define __NR_getcpu  318
+#define __NR_get_bcid  319
+#define __NR_set_bcid  320
+#define __NR_set_bclimit 321
+#define __NR_get_bcstat  322

 #ifdef __KERNEL__
```

```
-#define NR_syscalls 318
+#define NR_syscalls 323
 #include <linux/err.h>

 /*
--- ./include/asm-ia64/unistd.h.ve3 2006-07-10 12:39:19.000000000 +0400
+++ ./include/asm-ia64/unistd.h 2006-08-21 14:24:29.000000000 +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.ve3 2006-07-10 12:39:19.000000000 +0400
+++ ./include/asm-powerpc/systbl.h 2006-08-21 14:28:46.000000000 +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.ve3 2006-07-10 12:39:19.000000000 +0400
+++ ./include/asm-powerpc/unistd.h 2006-08-21 14:28:24.000000000 +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.ve3 2006-07-10 12:39:19.000000000 +0400
+++ ./include/asm-sparc/unistd.h 2006-08-21 14:33:20.000000000 +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.ve3 2006-07-10 12:39:19.000000000 +0400
+++ ./include/asm-sparc64/unistd.h 2006-08-21 14:34:10.000000000 +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.ve3 2006-08-21 13:15:39.000000000 +0400
+++ ./include/asm-x86_64/unistd.h 2006-08-21 14:35:19.000000000 +0400
@@ -619,10 +619,18 @@ __SYSCALL(__NR_sync_file_range, sys_sync
 __SYSCALL(__NR_vmsplice, sys_vmsplice)
 #define __NR_move_pages  279
 __SYSCALL(__NR_move_pages, sys_move_pages)
+#define __NR_get_bcid  280
+__SYSCALL(__NR_get_bcid, sys_get_bcid)
+#define __NR_set_bcid  281
+__SYSCALL(__NR_set_bcid, sys_set_bcid)
+#define __NR_set_bclimit 282
+__SYSCALL(__NR_set_bclimit, sys_set_bclimit)
+#define __NR_get_bcstat  283
+__SYSCALL(__NR_get_bcstat, sys_get_bcstat)

 #ifdef __KERNEL__

-#define __NR_syscall_max __NR_move_pages
```

```
+#define __NR_syscall_max __NR_get_bcstat
 #include <linux/err.h>

 #ifndef __NO_STUBS
--- ./kernel/sys_ni.c.ve3 2006-07-10 12:39:20.000000000 +0400
+++ ./kernel/sys_ni.c 2006-08-21 14:12:49.000000000 +0400
@@ -134,3 +134,9 @@ cond_syscall(sys_madvise);
 cond_syscall(sys_mremap);
 cond_syscall(sys_remap_file_pages);
 cond_syscall(compat_sys_move_pages);
+
+/* user resources syscalls */
+cond_syscall(sys_set_bcid);
+cond_syscall(sys_get_bcid);
+cond_syscall(sys_set_bclimit);
+cond_syscall(sys_get_bcstat);
--- ./kernel/bc/Makefile.ve3 2006-08-21 13:49:49.000000000 +0400
+++ ./kernel/bc/Makefile 2006-08-21 13:55:39.000000000 +0400
@@ -6,3 +6,4 @@

 obj-$(CONFIG_BEANCOUNTERS) += beancounter.o
 obj-$(CONFIG_BEANCOUNTERS) += misc.o
+obj-$(CONFIG_BEANCOUNTERS) += sys.o
--- ./kernel/bc/sys.c.ve3 2006-08-21 13:49:49.000000000 +0400
+++ ./kernel/bc/sys.c 2006-08-21 14:43:04.000000000 +0400
@@ -0,0 +1,120 @@
+/*
+ *  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)
```

```
+{
+ int error;
+ struct beancounter *bc;
+ struct task_beancounter *task_bc;
+
+ task_bc = &current->task_bc;
+
+ /* You may only set an bc as root */
+ error = -EPERM;
+ if (!capable(CAP_SETUID))
+  goto out;
+
+ /* Ok - set up a beancounter entry for this user */
+ error = -ENOMEM;
+ bc = beancounter_findcreate(id, BC_ALLOC);
+ if (bc == NULL)
+  goto out;
+
+ /* install bc */
+ put_beancounter(task_bc->exec_bc);
+ task_bc->exec_bc = bc;
+ put_beancounter(task_bc->fork_bc);
+ task_bc->fork_bc = get_beancounter(bc);
+ error = 0;
+out:
+ return error;
+}
+
+asmlinkage long sys_set_bclimit(bcid_t id, unsigned long resource,
+  unsigned long __user *limits)
+{
+ int error;
+ unsigned long flags;
+ 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 = beancounter_findcreate(id, BC_LOOKUP);
+ if (bc == NULL)
+   goto out;
+
+ spin_lock_irqsave(&bc->bc_lock, flags);
+ bc->bc_parms[resource].barrier = new_limits[0];
+ bc->bc_parms[resource].limit = new_limits[1];
+ spin_unlock_irqrestore(&bc->bc_lock, flags);
+
+ put_beancounter(bc);
+ error = 0;
+out:
+ return error;
+}
+
+int sys_get_bcstat(bcid_t id, unsigned long resource,
+   struct bc_resource_parm __user *uparm)
+{
+ int error;
+ unsigned long flags;
+ struct beancounter *bc;
+ struct bc_resource_parm parm;
+
+ error = -EINVAL;
+ if (resource >= BC_RESOURCES)
+   goto out;
+
+ error = -ENOENT;
+ bc = beancounter_findcreate(id, BC_LOOKUP);
+ if (bc == NULL)
+   goto out;
+
+ spin_lock_irqsave(&bc->bc_lock, flags);
+ parm = bc->bc_parms[resource];
+ spin_unlock_irqrestore(&bc->bc_lock, flags);
+ put_beancounter(bc);
+
+ error = 0;
+ if (copy_to_user(uparm, &parm, sizeof(parm)))
+   error = -EFAULT;
+
+out:
```

+ return error;
+}