Subject: [PATCH] msr.c: use smp_call_function_single()
Posted by adobriyan on Wed, 17 Jan 2007 12:52:46 GMT
View Forum Message <> Reply to Message

It will execute rdmsr and wrmsr only on the cpu we need.

Signed-off-by: Alexey Dobriyan <adobriyan@openvz.org>
---

 arch/i386/kernel/msr.c | 13 ++++----------
 1 file changed, 4 insertions(+), 9 deletions(-)

--- a/arch/i386/kernel/msr.c
+++ b/arch/i386/kernel/msr.c
@@ -68,7 +68,6 @@ static inline int rdmsr_eio(u32 reg, u32
 #ifdef CONFIG_SMP

 struct msr_command {
- int cpu;
  int err;
  u32 reg;
  u32 data[2];
@@ -78,16 +77,14 @@ static void msr_smp_wrmsr(void *cmd_bloc
 {
  struct msr_command *cmd = (struct msr_command *)cmd_block;

- if (cmd->cpu == smp_processor_id())
- cmd->err = wrmsr_eio(cmd->reg, cmd->data[0], cmd->data[1]);
+ cmd->err = wrmsr_eio(cmd->reg, cmd->data[0], cmd->data[1]);
 }

 static void msr_smp_rdmsr(void *cmd_block)
 {
  struct msr_command *cmd = (struct msr_command *)cmd_block;

- if (cmd->cpu == smp_processor_id())
- cmd->err = rdmsr_eio(cmd->reg, &cmd->data[0], &cmd->data[1]);
+ cmd->err = rdmsr_eio(cmd->reg, &cmd->data[0], &cmd->data[1]);
 }

 static inline int do_wrmsr(int cpu, u32 reg, u32 eax, u32 edx)
@@ -99,12 +96,11 @@ static inline int do_wrmsr(int cpu, u32
   ret = wrmsr_eio(reg, eax, edx);
  } else {
- cmd.cpu = cpu;
   cmd.reg = reg;
   cmd.data[0] = eax;

```
   cmd.data[1] = edx;

-  smp_call_function(msr_smp_wrmsr, &cmd, 1, 1);
+  smp_call_function_single(cpu, msr_smp_wrmsr, &cmd, 1, 1);
   ret = cmd.err;
  }
  preempt_enable();
@@ -120,10 +116,9 @@ static inline int do_rdmsr(int cpu, u32
  if (cpu == smp_processor_id()) {
   ret = rdmsr_eio(reg, eax, edx);
  } else {
-  cmd.cpu = cpu;
   cmd.reg = reg;

-  smp_call_function(msr_smp_rdmsr, &cmd, 1, 1);
+  smp_call_function_single(cpu, msr_smp_rdmsr, &cmd, 1, 1);

   *eax = cmd.data[0];
   *edx = cmd.data[1];
```