[v13,03/12] perf/x86/intel/pt: Introduce intel_pt_validate_cap()

Message ID	1540368316-12998-4-git-send-email-luwei.kang@intel.com
State	New
Headers	show
Series	Intel Processor Trace virtualization enabling
Related	show

10654361 | diff (/patch/10654361/raw/) | mbox (/patch/10654361/mbox/) | series (/series/34463/mbox/)

Commit Message

Kang, Luwei (/project/kvm/list/?submitter=168537) Oct. 24, 2018, 8:05 a.m. UTC

```
intel_pt_validate_hw_cap() validates whether a given PT capability is
supported by the hardware. It checks the PT capability array which
reflects the capabilities of the hardware on which the code is executed.

For setting up PT for KVM guests this is not correct as the capability
array for the guest can be different from the host array.

Provide a new function to check against a given capability array.

Acked-by: Song Liu <songliubraving@fb.com>
Signed-off-by: Luwei Kang <luwei.kang@intel.com>
---
 arch/x86/events/intel/pt.c      | 12 +++++++++---
 arch/x86/include/asm/intel_pt.h |  2 ++
 2 files changed, 11 insertions(+), 3 deletions(-)
```

Patch

10654361 | diff (/patch/10654361/raw/) | mbox (/patch/10654361/mbox/) | series (/series/34463/mbox/)

```
diff --git a/arch/x86/events/intel/pt.c b/arch/x86/events/intel/pt.c
index 309bb1d..53e481a 100644
--- a/arch/x86/events/intel/pt.c
+++ b/arch/x86/events/intel/pt.c
@@ -75,14 +75,20 @@
        PT_CAP(psb_periods,             1, CPUID_EBX, 0xffff0000),
 };

-u32 intel_pt_validate_hw_cap(enum pt_capabilities cap)
+u32 intel_pt_validate_cap(u32 *caps, enum pt_capabilities capability)
 {
-       struct pt_cap_desc *cd = &pt_caps[cap];
-       u32 c = pt_pmu.caps[cd->leaf * PT_CPUID_REGS_NUM + cd->reg];
+       struct pt_cap_desc *cd = &pt_caps[capability];
+       u32 c = caps[cd->leaf * PT_CPUID_REGS_NUM + cd->reg];
        unsigned int shift = __ffs(cd->mask);

        return (c & cd->mask) >> shift;
 }
+EXPORT_SYMBOL_GPL(intel_pt_validate_cap);
+
+u32 intel_pt_validate_hw_cap(enum pt_capabilities cap)
+{
+       return intel_pt_validate_cap(pt_pmu.caps, cap);
+}
 EXPORT_SYMBOL_GPL(intel_pt_validate_hw_cap);

 static ssize_t pt_cap_show(struct device *cdev,
diff --git a/arch/x86/include/asm/intel_pt.h b/arch/x86/include/asm/intel_pt.h
index fa4b4fd..00f4afb 100644
--- a/arch/x86/include/asm/intel_pt.h
+++ b/arch/x86/include/asm/intel_pt.h
@@ -26,9 +26,11 @@  enum pt_capabilities {
 #if defined(CONFIG_PERF_EVENTS) && defined(CONFIG_CPU_SUP_INTEL)
 void cpu_emergency_stop_pt(void);
 extern u32 intel_pt_validate_hw_cap(enum pt_capabilities cap);
+extern u32 intel_pt_validate_cap(u32 *caps, enum pt_capabilities cap);
 #else
 static inline void cpu_emergency_stop_pt(void) {}
 static inline u32 intel_pt_validate_hw_cap(enum pt_capabilities cap) { return 0; }
+static inline u32 intel_pt_validate_cap(u32 *caps, enum pt_capabilities capability) { return 0; }
 #endif

 #endif /* _ASM_X86_INTEL_PT_H */
```