[v13,00/12] Intel Processor Trace virtualization enabling

10654355	mbox (/cover/10654355/mbox/)	series (/series/34463/mbox/)

Message ID	1540368316-12998-1-git-send-email-luwei.kang@intel.com
Headers	show
Series	Intel Processor Trace virtualization enabling
Related	show

Message

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

```
>From V12
 - Refine the title and description of patch 1~3. -- Thomas Gleixner
 - Rename the function of validate the capabilities of Intel PT. -- Thomas Gleixner
 - Add more description of Intel PT work mode. -- Alexander Shishkin

>From V11:
 - In patch 3, arguments caps vs. cap is not good. Spell second one out. -- Thomas Gleixner

>From V10: (This version don't have code change)
 - move the patch 5 in version 9 to patch 3 (reorder patch 5) -- Alexander Shishkin
 - refind the patch description of patch 5 (add new capability for Intel PT) -- Alexander Shishkin
 - CC all the maintainers, reviewers and submitters in each patch of this patch set -- Alexander Shishkin

>From V9:
 - remove redundant initialize for "ctl_bitmask" in patch 9;
 - do some changes for patch's description.

>From V8:
 - move macro definition MSR_IA32_RTIT_ADDR_RANGE from msr-index.h to intel_pt.h;
 - initialize the RTIT_CTL bitmask to ~0ULL.

>From V7:
 - remove host only mode since it can be emulated by perf code;
 - merge patch 8 and 9 to make code and data in the same patch;
 - rename __pt_cap_get() to pt_cap_decode();
 - other minor change.

>From V6:
 - split pathes 1~2 to four separate patches (these patches do 2 things) and add more descriptions.

>From V5:
 - rename the function from pt_cap_get_ex() to __pt_cap_get();
 - replace the most of function from vmx_pt_supported() to "pt_mode == PT_MODE_HOST_GUEST"(or !=).

>From V4:
 - add data check when setting the value of MSR_IA32_RTIT_CTL;
 - Invoke new interface to set the intercept of MSRs read/write after "MSR bitmap per-vcpu" patches.

>From V3:
 - change default mode to SYSTEM mode;
 - add a new patch to move PT out of scattered features;
 - add a new fucntion kvm_get_pt_addr_cnt() to get the number of address ranges;
 - add a new function vmx_set_rtit_ctl() to set the value of guest RTIT_CTL, GUEST_IA32_RTIT_CTL and MSRs intercept.

>From v2:
 - replace *_PT_SUPPRESS_PIP to *_PT_CONCEAL_PIP;
 - clean SECONDARY_EXEC_PT_USE_GPA, VM_EXIT_CLEAR_IA32_RTIT_CTL and VM_ENTRY_LOAD_IA32_RTIT_CTL in SYSTEM mode. These bits must be all set or
 - move processor tracing out of scattered features;
 - add a new function to enable/disable intercept MSRs read/write;
 - add all Intel PT MSRs read/write and disable intercept when PT is enabled in guest;
```

```
 - disable Intel PT and enable intercept MSRs when L1 guest VMXON;
 - performance optimization.
   In Host only mode. we just need to save host RTIT_CTL before vm-entry and restore host RTIT_CTL after vm-exit;
   In HOST_GUEST mode. we need to save and restore all MSRs only when PT has enabled in guest.
 - use XSAVES/XRESTORES implement context switch.
   Haven't implementation in this version and still in debuging. will make a separate patch work on this.

>From v1:
 - remove guest-only mode because guest-only mode can be covered by host-guest mode;
 - always set "use GPA for processor tracing" in secondary execution control if it can be;
 - trap RTIT_CTL read/write. Forbid write this msr when VMXON in L1 hypervisor.

Chao Peng (7):
  perf/x86/intel/pt: Move Intel PT MSRs bit defines to global header
  perf/x86/intel/pt: Export pt_cap_get()
  KVM: x86: Add Intel PT virtualization work mode
  KVM: x86: Add Intel Processor Trace cpuid emulation
  KVM: x86: Add Intel PT context switch for each vcpu
  KVM: x86: Implement Intel PT MSRs read/write emulation
  KVM: x86: Set intercept for Intel PT MSRs read/write

Luwei Kang (5):
  perf/x86/intel/pt: Introduce intel_pt_validate_cap()
  perf/x86/intel/pt: Add new bit definitions for PT MSRs
  perf/x86/intel/pt: add new capability for Intel PT
  KVM: x86: Introduce a function to initialize the PT configuration
  KVM: x86: Disable Intel PT when VMXON in L1 guest

 arch/x86/events/intel/pt.c       |  60 +++---
 arch/x86/events/intel/pt.h       |  58 -----
 arch/x86/include/asm/intel_pt.h  |  39 ++++
 arch/x86/include/asm/kvm_host.h  |   1 +
 arch/x86/include/asm/msr-index.h |  37 ++++
 arch/x86/include/asm/vmx.h       |   8 +
 arch/x86/kvm/cpuid.c             |  22 +-
 arch/x86/kvm/svm.c               |   6 +
 arch/x86/kvm/vmx.c               | 446 ++++++++++++++++++++++++++++++++++++++-
 arch/x86/kvm/x86.c               |  33 ++-
 10 files changed, 620 insertions(+), 90 deletions(-)
```