QEMU-Devel Archive mirror

search help / color / mirror / Atom feed

```
From: Igor Mammedov <imammedo@redhat.com>
To: Salil Mehta <salil.mehta@huawei.com>
Cc: <qemu-devel@nongnu.org>, <qemu-arm@nongnu.org>, <maz@kernel.org>,
        <jean-philippe@linaro.org>, <jonathan.cameron@huawei.com>,
        <lpieralisi@kernel.org>, <peter.maydell@linaro.org>,
        <richard.henderson@linaro.org>, <andrew.jones@linux.dev>,
        <david@redhat.com>, <philmd@linaro.org>, <eric.auger@redhat.com>,
        <oliver.upton@linux.dev>, <pbonzini@redhat.com>, <mst@redhat.com>,
        <will@kernel.org>, <gshan@redhat.com>, <rafael@kernel.org>,
        <alex.bennee@linaro.org>, <linux@armlinux.org.uk>,
        <darren@os.amperecomputing.com>, <ilkka@os.amperecomputing.com>,
        <vishnu@os.amperecomputing.com>, <karl.heubaum@oracle.com>,
        <miguel.luis@oracle.com>, <salil.mehta@opnsrc.net>,
        <zhukeqian1@huawei.com>, <wangxiongfeng2@huawei.com>,
        <wangyanan55@huawei.com>, <jiakernel2@gmail.com>,
        <maobibo@loongson.cn>, <lixianglai@loongson.cn>,
        <linuxarm@huawei.com>
Subject: Re: [PATCH V6 1/9] accel/kvm: Extract common KVM vCPU {creation,parking} code
Date: Fri, 27 Oct 2023 14:56:52 +0200   [thread overview]
Message-ID: <20231027145652.44cc845c@imammedo.users.ipa.redhat.com> (raw)
In-Reply-To: <20231013105129.25648-2-salil.mehta@huawei.com>

On Fri, 13 Oct 2023 11:51:21 +0100
Salil Mehta <salil.mehta@huawei.com> wrote:

> KVM vCPU creation is done once during the initialization of the VM when Qemu
                            ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
> thread is spawned. This is common to all the architectures.

is it really true fox x86?

> 
> Hot-unplug of vCPU results in destruction of the vCPU object in QOM but the
> corresponding KVM vCPU object in the Host KVM is not destroyed and its
                                                                  ^
since KVM doesn't support vCPU removal

> representative KVM vCPU object/context in Qemu is parked.
> 
> Refactor common logic so that some APIs could be reused by vCPU Hotplug code.
'reused' part doesn't happen within this series. So a reason
why patch exists is not clear/no one can deduce the reason
without the actual user here.

Suggest to move it to a series that actually will use this patch.


> Update new/old APIs with trace events instead of DTRACE.
> 
> Signed-off-by: Salil Mehta <salil.mehta@huawei.com>
> Reviewed-by: Gavin Shan <gshan@redhat.com>
> Tested-by: Vishnu Pajjuri <vishnu@os.amperecomputing.com>
> Reviewed-by: Jonathan Cameron <Jonathan.Cameron@huawei.com>
> Tested-by: Xianglai Li <lixianglai@loongson.cn>
> ---
>  accel/kvm/kvm-all.c      | 64 ++++++++++++++++++++++++++++++++----------
>  accel/kvm/trace-events   |  4 +++
>  include/sysemu/kvm.h     | 16 ++++++++++++
>  3 files changed, 69 insertions(+), 15 deletions(-)
> 
> diff --git a/accel/kvm/kvm-all.c b/accel/kvm/kvm-all.c
> index 72e1d1141c..bfa7816aaa 100644
> --- a/accel/kvm/kvm-all.c
> +++ b/accel/kvm/kvm-all.c
> @@ -137,6 +137,7 @@ static QemuMutex kml_slots_lock;
>  #define kvm_slots_unlock()  qemu_mutex_unlock(&kml_slots_lock)
> 
>  static void kvm_slot_init_dirty_bitmap(KVMSlot *mem);
> +static int kvm_get_vcpu(KVMState *s, unsigned long vcpu_id);
> 
>  static inline void kvm_resample_fd_remove(int gsi)
>  {
> @@ -320,14 +321,53 @@ err:
>      return ret;
>  }
> 
> +void kvm_park_vcpu(CPUState *cpu)
> +{
> +    struct KVMParkedVcpu *vcpu;
> +
> +    trace_kvm_park_vcpu(cpu->cpu_index, kvm_arch_vcpu_id(cpu));
> +
> +    vcpu = g_malloc0(sizeof(*vcpu));
> +    vcpu->vcpu_id = kvm_arch_vcpu_id(cpu);
> +    vcpu->kvm_fd = cpu->kvm_fd;
> +    QLIST_INSERT_HEAD(&kvm_state->kvm_parked_vcpus, vcpu, node);
> +}
> +
> +int kvm_create_vcpu(CPUState *cpu)
> +{
> +    unsigned long vcpu_id = kvm_arch_vcpu_id(cpu);
> +    KVMState *s = kvm_state;
> +    int kvm_fd;
> +
> +    trace_kvm_create_vcpu(cpu->cpu_index, kvm_arch_vcpu_id(cpu));
> +
> +    /* check if the KVM vCPU already exist but is parked */
```

```
> +    kvm_fd = kvm_get_vcpu(s, vcpu_id);
> +    if (kvm_fd < 0) {
> +        /* vCPU not parked: create a new KVM vCPU */
> +        kvm_fd = kvm_vm_ioctl(s, KVM_CREATE_VCPU, vcpu_id);
> +        if (kvm_fd < 0) {
> +            error_report("KVM_CREATE_VCPU IOCTL failed for vCPU %lu", vcpu_id);
> +            return kvm_fd;
> +        }
> +    }
> +
> +    cpu->kvm_fd = kvm_fd;
> +    cpu->kvm_state = s;
> +    cpu->vcpu_dirty = true;
> +    cpu->dirty_pages = 0;
> +    cpu->throttle_us_per_full = 0;
> +
> +    return 0;
> +}
> +
>  static int do_kvm_destroy_vcpu(CPUState *cpu)
>  {
>      KVMState *s = kvm_state;
>      long mmap_size;
> -    struct KVMParkedVcpu *vcpu = NULL;
>      int ret = 0;
>
> -    DPRINTF("kvm_destroy_vcpu\n");
> +    trace_kvm_destroy_vcpu(cpu->cpu_index, kvm_arch_vcpu_id(cpu));
>
>      ret = kvm_arch_destroy_vcpu(cpu);
>      if (ret < 0) {
> @@ -353,10 +393,7 @@ static int do_kvm_destroy_vcpu(CPUState *cpu)
>          }
>
> -    vcpu = g_malloc0(sizeof(*vcpu));
> -    vcpu->vcpu_id = kvm_arch_vcpu_id(cpu);
> -    vcpu->kvm_fd = cpu->kvm_fd;
> -    QLIST_INSERT_HEAD(&kvm_state->kvm_parked_vcpus, vcpu, node);
> +    kvm_park_vcpu(cpu);
>  err:
>      return ret;
>  }
> @@ -377,6 +414,8 @@ static int kvm_get_vcpu(KVMState *s, unsigned long vcpu_id)
>          if (cpu->vcpu_id == vcpu_id) {
>              int kvm_fd;
>
> +            trace_kvm_get_vcpu(vcpu_id);
> +
>              QLIST_REMOVE(cpu, node);
>              kvm_fd = cpu->kvm_fd;
>              g_free(cpu);
> @@ -384,7 +423,7 @@ static int kvm_get_vcpu(KVMState *s, unsigned long vcpu_id)
>          }
>      }
>
> -    return kvm_vm_ioctl(s, KVM_CREATE_VCPU, (void *)vcpu_id);
> +    return -ENOENT;
>  }
>
>  int kvm_init_vcpu(CPUState *cpu, Error **errp)
> @@ -395,19 +434,14 @@ int kvm_init_vcpu(CPUState *cpu, Error **errp)
>
>      trace_kvm_init_vcpu(cpu->cpu_index, kvm_arch_vcpu_id(cpu));
>
> -    ret = kvm_get_vcpu(s, kvm_arch_vcpu_id(cpu));
> +    ret = kvm_create_vcpu(cpu);
>      if (ret < 0) {
> -        error_setg_errno(errp, -ret, "kvm_init_vcpu: kvm_get_vcpu failed (%lu)",
> +        error_setg_errno(errp, -ret,
> +                         "kvm_init_vcpu: kvm_create_vcpu failed (%lu)",
>                           kvm_arch_vcpu_id(cpu));
>          goto err;
>      }
>
> -    cpu->kvm_fd = ret;
> -    cpu->kvm_state = s;
> -    cpu->vcpu_dirty = true;
> -    cpu->dirty_pages = 0;
> -    cpu->throttle_us_per_full = 0;
> -
>      mmap_size = kvm_ioctl(s, KVM_GET_VCPU_MMAP_SIZE, 0);
>      if (mmap_size < 0) {
>          ret = mmap_size;
> diff --git a/accel/kvm/trace-events b/accel/kvm/trace-events
> index 399aaeb0ec..cdd0c95c09 100644
> --- a/accel/kvm/trace-events
> +++ b/accel/kvm/trace-events
> @@ -9,6 +9,10 @@ kvm_device_ioctl(int fd, int type, void *arg) "dev fd %d, type 0x%x, arg %p"
>  kvm_failed_reg_get(uint64_t id, const char *msg) "Warning: Unable to retrieve ONEREG %" PRIu64 " from KVM: %s"
>  kvm_failed_reg_set(uint64_t id, const char *msg) "Warning: Unable to set ONEREG %" PRIu64 " to KVM: %s"
>  kvm_init_vcpu(int cpu_index, unsigned long arch_cpu_id) "index: %d id: %lu"
> +kvm_create_vcpu(int cpu_index, unsigned long arch_cpu_id) "index: %d id: %lu"
> +kvm_get_vcpu(unsigned long arch_cpu_id) "id: %lu"
> +kvm_destroy_vcpu(int cpu_index, unsigned long arch_cpu_id) "index: %d id: %lu"
> +kvm_park_vcpu(int cpu_index, unsigned long arch_cpu_id) "index: %d id: %lu"
>  kvm_irqchip_commit_routes(void) ""
>  kvm_irqchip_add_msi_route(char *name, int vector, int virq) "dev %s vector %d virq %d"
>  kvm_irqchip_update_msi_route(int virq) "Updating MSI route virq=%d"
> diff --git a/include/sysemu/kvm.h b/include/sysemu/kvm.h
```

```
> index ee9025f8e9..8137e6a44c 100644
> --- a/include/sysemu/kvm.h
> +++ b/include/sysemu/kvm.h
> @@ -465,6 +465,22 @@ void kvm_set_sigmask_len(KVMState *s, unsigned int sigmask_len);
>  int kvm_physical_memory_addr_from_host(KVMState *s, void *ram_addr,
>                                         hwaddr *phys_addr);
>
> +/**
> + * kvm_create_vcpu - Gets a parked KVM vCPU or creates a KVM vCPU
> + * @cpu: QOM CPUState object for which KVM vCPU has to be fetched/created.
> + *
> + * @returns: 0 when success, errno (<0) when failed.
> + */
> +int kvm_create_vcpu(CPUState *cpu);
> +
> +/**
> + * kvm_park_vcpu - Park QEMU KVM vCPU context
> + * @cpu: QOM CPUState object for which QEMU KVM vCPU context has to be parked.
> + *
> + * @returns: none
> + */
> +void kvm_park_vcpu(CPUState *cpu);
> +
>  #endif /* NEED_CPU_H */
>
>  void kvm_cpu_synchronize_state(CPUState *cpu);
```

---

next prev parent reply other threads:[~2023-11-07 13:51 UTC|newest]

Thread overview: 28+ messages / expand[flat|nested] mbox.gz Atom feed top

---

Reply instructions:

You may reply publicly to this message via plain-text email
using any one of the following methods:

* Save the following mbox file, import it into your mail client,
  and reply-to-all from there: mbox

  Avoid top-posting and favor interleaved quoting:
  https://en.wikipedia.org/wiki/Posting_style#Interleaved_style

* Reply using the **--to**, **--cc**, and **--in-reply-to**
  switches of git-send-email(1):

```
  git send-email \
    --in-reply-to=20231027145652.44cc845c@imammedo.users.ipa.redhat.com \
    --to=imammedo@redhat.com \
    --cc=alex.bennee@linaro.org \
    --cc=andrew.jones@linux.dev \
    --cc=darren@os.amperecomputing.com \
    --cc=david@redhat.com \
    --cc=eric.auger@redhat.com \
    --cc=gshan@redhat.com \
    --cc=ilkka@os.amperecomputing.com \
    --cc=jean-philippe@linaro.org \
    --cc=jiakernel2@gmail.com \
    --cc=jonathan.cameron@huawei.com \
    --cc=karl.heubaum@oracle.com \
    --cc=linux@armlinux.org.uk \
    --cc=linuxarm@huawei.com \
    --cc=lixianglai@loongson.cn \
    --cc=lpieralisi@kernel.org \
    --cc=maobibo@loongson.cn \
    --cc=maz@kernel.org \
```

```
    --cc=miguel.luis@oracle.com \
    --cc=mst@redhat.com \
    --cc=oliver.upton@linux.dev \
    --cc=pbonzini@redhat.com \
    --cc=peter.maydell@linaro.org \
    --cc=philmd@linaro.org \
    --cc=qemu-arm@nongnu.org \
    --cc=qemu-devel@nongnu.org \
    --cc=rafael@kernel.org \
    --cc=richard.henderson@linaro.org \
    --cc=salil.mehta@huawei.com \
    --cc=salil.mehta@opnsrc.net \
    --cc=vishnu@os.amperecomputing.com \
    --cc=wangxiongfeng2@huawei.com \
    --cc=wangyanan55@huawei.com \
    --cc=will@kernel.org \
    --cc=zhukeqian1@huawei.com \
    /path/to/YOUR_REPLY

  https://kernel.org/pub/software/scm/git/docs/git-send-email.html

* If your mail client supports setting the In-Reply-To header
  via mailto: links, try the mailto: link

Be sure your reply has a Subject: header at the top and a blank line before the message body.
```
