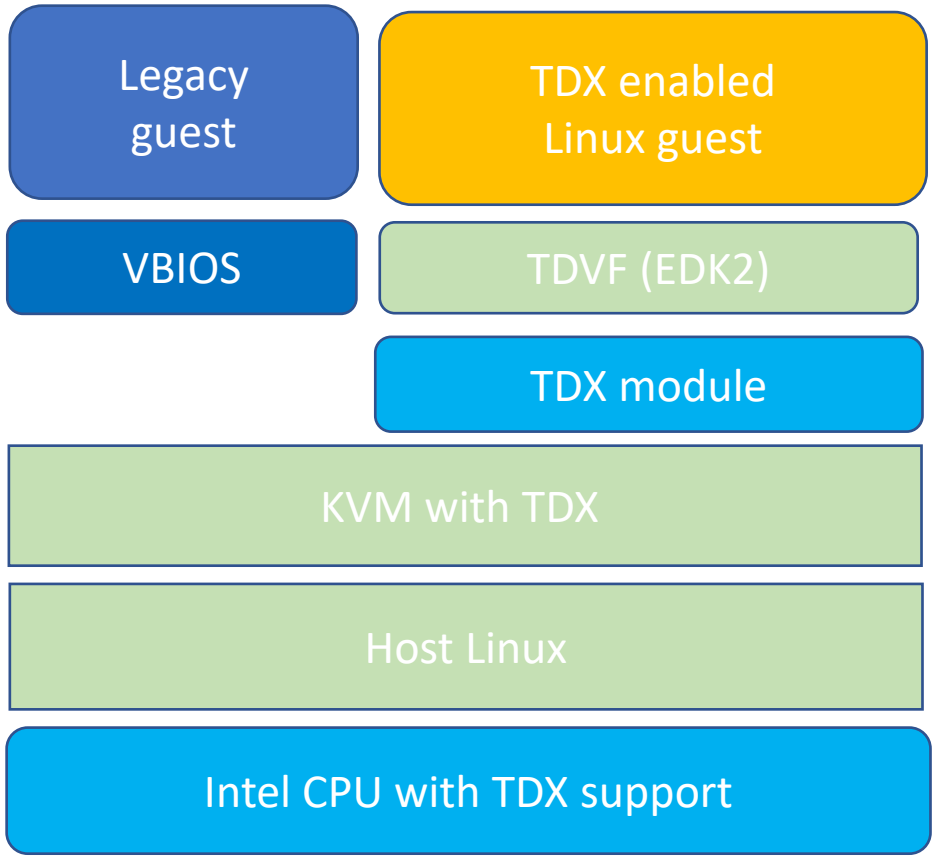
Linux TDX guest for plumbers

Andi Kleen, Sathya Kuppuswamy, Kirill Shutemov, Elena Reshetova, Shiny Sebastian, Isaku Yamahata, Chao Gao, Casey Schaufler, others

Based on earlier work from Sean Christopherson

TDX overview



- Guest only uses TDCALL and shared memory to communicate
- Guest kernel handles MSRs/MMIO/Port IO/CPUID through TDCALLs
- IO through virtio

Performance

- Exits to the host have extra overhead due to TDX module
  - Normal timer interrupt slightly more expensive, likely due to TSC deadline MSR write
    - Use periodic mode when possible?
  - In general optimizations to reduce exits are good
- Untuned swiotlb for virtio adds overhead
  - Split up spinlocks: single lock significant bottleneck
  - Better reuse of swiotlb buffers
  - Further tuning possible?

Lazy accept

- Guest to “accept” memory before it can be used
  - Doing it upfront in TDVF is a serious boot time performance problem
- Kernel has to track what memory is already accepted
- Using 2MB granularity bitmap allocated in decompressor
- Then page allocator accepts in 2MB chunks as needed

- Open issue:
  - How to pass bitmap to kexec, including handling shared memory

Security

- Guest is protected, but can be still attacked through host communication
  - Like “server on untrusted network”
- Disabling as much code as possible
- Device filtering to minimize drivers
  - Mostly done at driver model level using allow list, but some need manual changes
  - Also using opt-in for shared MMIO and IO port filter
- Hardening allowed drivers and non-driver communication
- Adding fuzzing hooks for more testing
- How can the security be ensured long term?

- Elena’s separate talk at Security Summit going into more details