QEMU

- Creates the machine
- Emulates devices
  - some mimic real devices
  - some are special: paravirtualized
- Entire guest is contained within QEMU
- Uses several services from host kernel
  - KVM for guest control
  - Linux for resources
- Runs unprivileged
KVM

- Do one thing, do it right
- Linux kernel module
- Exposes hardware features for virtualization to userspace
- Emulates some devices
  - Like APIC
- Enables several features needed by QEMU
  - Like keeping track of pages guest changes
libvirt

- Management of VMs, storage, network
- Provides a stable API
- Remote management
- `virsh` – command-line interface
- cgroups
- sVirt
- Makes it possible for QEMU to run unprivileged
  - Opens files, connections and passes them on to QEMU
Note on higher layers

- OpenStack
  - Cloud or IaaS management
- oVirt
  - Data centre management
- virt-manager / GNOME Boxes
  - PC management
- libguestfs
  - Nifty tool to perform several operations on VM images
KVM Today

• Very good performance and scalability
  – Consistently tops SPECVirt results
• Default hypervisor for oVirt, OpenStack
• Out-of-box support in all distributions
Live Migration

- Pick guest state from one QEMU process and transfer it to another
  - while the guest is running
- The guest shouldn't realize the world is changing beneath its feet
  - in other words, the guest isn't involved in the process
  - might notice degraded performance, though
- Useful for load balancing, hardware / software maintenance, power saving, checkpointing, ...
KVM Today

• Very good performance and scalability
  – Consistently tops SPECVirt results
• Default hypervisor for oVirt, OpenStack
• Out-of-box support in all distributions
Workstations

• Main interaction with guests
• Migration is triggered by admins
• Don't need anything more fancy / heavyweight
Data Centres / Clouds

- Main interaction with hosts
- Migration is triggered by policies, transparent to admins
- Policies optimise resource usage; host maintenance, etc.
Data Centres

- Scale-up
- Traditional workloads
  - large databases

- Many vCPUs
- Lots of RAM
- Critical data
- Shared storage

Amit Shah  FOSDEM 2016
IaaS / Clouds

- Scale-out
- Custom applications

- Compute, storage separate
- Sometimes compute has storage
  - Needs block migration
Block Migration

- Using only QEMU
  - Take a snapshot of disk image
  - Migrate base disk image
  - Migrate RAM and new snapshot
  - Iterate till VM converges

- Using libvirt
  - Setup NBD connection between hosts
  - Transfer block device contents across hosts
Big VMs

- Performance drop while migration in progress
- Customers don't like this
QEMU Main Loop (old)

```c
main_loop()
{
    while (1) {
        service_guest_requests();
        service_guest_io();
        migration_pass();
    }
}
```
QEMU Main Loop (new)

```c
thread1

main_loop()
{
    while (1) {
        service_guest_requests();
        service_guest_io();
    }
}

thread 2

migration_thread()
{
    while (1) {
        migration_pass();
    }
}
```

- Add migration thread
- Helps parallelise guest IO and migration passes
Oops

- Guest doesn't migrate
- Slowness was a feature!
Restrict Guest

- Throttle guest vCPUs
  - Hope the rate of dirtying memory reduces
  - Autoconverge
  - cgroups
- Offline guest vCPUs
Compression

• Multi-threaded compression
  – Compress pages before sending
  – Do this in multiple threads

• xbzrle
  – Send diffs of pages from previous iteration
  – Means we have to maintain a cache of pages sent in previous iteration
Postcopy

- Migrate guest before all RAM has been transferred
- Keep transferring pages from src to dest on a new channel
- Remote-page-fault pages which don't exist on dest
  - Special OOB mode of transferring pages on the new channel
- userfaultfd in Linux implements remote page fault functionality
Other Challenges

- **Multiple migrations**
  - Logs get left behind on older hosts
  - 24th migration might be failing, 23 prior ones have succeeded
    - but we don't know it's the 24th attempt

- **Multiple layers**
  - Have to check logs for each layer top->down to find cause

- **QEMU defaults**
  - Not suitable for all projects
  - QEMU devels don't know about deployment scenarios
  - More communication between projects to understand options
  - New focus on feature pages to expose more info to higher levels
Thank You!