Google Summer of Code 2016 proposal: Improving bhyve support for libvirt

Proposal

The primary aim of this project is to implement missing calls and functionality in the libvirt bhyve driver. According to the libvirt API Support Matrix, there are a large number of calls not yet implemented. While some missing API calls are not applicable to bhyve, a number of them are, among them the following calls:

- General calls:
    - virConnectDomainXMLFromNative
      This would mostly be an argument parser for a bhyve command line. I would reuse parts of the bhyvectl argument parser.
    - virConnectGetCPUModelNames
    - virConnectGetLibVersion
- Connection calls:
  Most of these include some form of authentication handling and are therefore not applicable. However, the following do apply to bhyve and are easy to implement:
    - virConnectGetType
      Trivially return "BHYVE"
    - virConnectIsAlive
      Check for /dev/vmm presence
    - virConnectIsEncrypted
      Trivially return 0, since bhyve does not support encrypted interfaces
    - virConnectIsSecure
      Check for access rights on /dev/vmm
- General Domain calls
    - virDomainGetMaxMemory
      Since bhyve does not support memory ballooning, just return the amount of memory allocated here
    - virDomainGetMaxVcpus, virDomainGetVcpus
    - virDomainGetCPUStats
    - virDomainGetTime
    - virDomainInjectNMI
      This one is a bit of a catch - I don't know yet if injecting an NMI is possible from the userland utilities. Possibly an ioctl has to be implemented for this. I'd like to discuss on the implications of implementing this first though.
    - virDomainReset
      Kill a bhyve VM with SIGKILL and re-start it
    - virDomainReboot
    - virDomainShutdown
      Kill a bhyve VM with SIGTERM
- Block-Device level calls
  These would implement access to the vdev block storage layer. I plan to implement support for both file-backed and zvol-backed virtual machines for the following API calls:
    - virDomainGetBlockInfo
    - virDomainBlockPeek

    - virDomainBlockCopy
    - virDomainBlockStats
    - virDomainBlockStatsFlags

  Going further, since zvols support snapshotting, I plan to implement the following for zvol-backed storage
    - virDomainBlockCommit
    - virDomainBlockPull
- VirtFS layer
  I would like to create patches to support specifieng filesystems when creating the domain as well as the following calls to be merged at a later time when VirtFS-9p support for bhyve becomes ready:
    - virDomainFSFreeze
    - virDomainFSThaw
- Memory inspection
    - virDomainMemoryPeek
      A guest's memory space is exposed in /dev/vmm, so this call would have to read from that.
    - virDomainMemoryStats

About me
My name is Fabian Freyer, I am a physics student at the TU Berlin interested in IT security, hardware and low-level programming. Aside from my studies, I have been working as a FreeBSD Sysadmin for several years. I am fluent in C and Python and am interested in OS development; in fact I have been working on a toy operating system and boot loader as a side project for a while now.
I playing CTFs with Team Tasteless and am working on my skills at reading and writing x86, amd64 and arm assembly.
I have not yet participated in a GSOC due to other projects I was pursuing in the last years, such as a BEXUS Project to develop a compact sensor for cosmic radiation.
My motivation to pursue this project is that I would greatly appreciate VM migration in bhyve which would make bhyve viable for several of my projects and that I have been using virtualization-based debuggers in several CTF challenges, which has sparked my interest in virtualization. I would also like to use this project as a chance to learn more about practical hypervisor architecture.
While I am happy to individually study the bhyve source code, I would greatly appreciate advice from someone familiar with the bhyve project.

During the summer I have not planned any other projects aside from the occasional CTF contest, so I will be able to commit full-time to this project.

Contact Details:
Name: Fabian Freyer
Email: fabian.freyer@physik.tu-berlin.de
Jabber: ffreyer@jabber.physik.tu-berlin.de
IRC: plonk on irc.freenode.net