Agenda

• Project Introduction
• Ceph
• USB Storage
• Demo!
• Azure Blob Storage
• Linux I/O Target in Userspace
Hack Week 13

• What to hack on?
  - ARM board gathering dust
  - Learn something new
  - Storage is my day job

•  I know...
Project Idea

Ceph USB Storage Gateway
Goals

• Access cloud storage from anything
  - Stereos, TVs, Phones, etc.

• Boot from cloud backed disk images
  - Ceph
  - Azure

• Simple device configuration
Ceph

- Aggregate, manage and share storage resources
- Highly available
  - No single point of failure
- Self managing and self healing
- Scalable
Ceph

LIBRADOS
A library allowing apps to directly access RADOS, with support for C, C++, Java, Python, Ruby, and PHP

RADOSGW
A bucket-based REST gateway, compatible with S3 and Swift

RBD
A reliable and fully-distributed block device, with a Linux kernel client and a QEMU/KVM driver

CEPHFS
A POSIX-compliant distributed file system, with a Linux kernel client and support for FUSE

RADOS
A reliable, autonomous, distributed object store comprised of self-healing, self-managing, intelligent storage nodes.
Ceph RADOS Block Device

- Block device backed by RADOS objects
- Thin provisioned
- Resizeable
- Supports snapshots and clones
- Linux kernel and user-space clients
Hardware
Hardware

• Mainline kernel support
  - sunxi community

• openSUSE Tumbleweed port

• Relatively performant
  - 2x1GHz ARMv7 with 2GB RAM
  - USB2 and “gigabit” Ethernet
Hardware

Alternatives
USB Storage

• SCSI transport
  - Bulk-Only transport (BOT)
  - USB Attached SCSI (UAS)
    - Faster: high-speed and super-speed specs

• Kernel support
  - f_mass_storage.ko
  - f_tcm.ko
    - Support for BOT and UAS
USB Gateway

Plug-in

Boot Linux

Provision Conf-FS

Await Eject

Commit Conf-FS

Connect Network

Map RBD Image

Expose RBD Via USB

Fast Path

Ceph.conf Keyring
Azure Blob Storage

• Public cloud storage
  - RESTful protocol
  - Pay for what you use

• Page Blobs and Block Blobs
  - Page Blobs ideal for disk images
    - Sparse object
    - Accepts 512-byte aligned I/Os at arbitrary offsets

• Premium accounts with QoS constraints
• Linux I/O Target (LIO)

  • In-kernel SCSI target
    - Pluggable transport and storage engine layers
    - Transports: FC, iSCSI, loopback, USB, etc.
    - Storage engines: file, block device, tcm-user (TCMU)

  • TCMU
    - LIO storage engine in user-space
    - SCSI pass-through
• **Linux I/O Target with Azure**

  • Elasto Cloud project
    - Azure Page Blob client written in C
    - Also supports Azure File Service and Amazon S3 protocols
  
  • TCMU Elasto handler
    - Maps SCSI I/O to Azure Page Blob REST requests
    - Page Blobs accessible as regular block devices
      - Exposable via supported LIO transports
• Linux I/O Target with Azure

Microsoft Azure

HTTP(S)

Elasto Client Library

TCMU

Linux I/O Target

Filesystem

SCSI

USB
Future Challenges

• Performance
  - USB3+ and GbE/802.11ac

• Power
  - Battery to reduce reliance on USB supply

• f_tcm
  - Works in VM (loopback) but fails on board
  - Needs super-speed support?

• Caching

• Transparent encryption
Questions?

Code:
https://github.com/ddiss/rbd-usb.git
http://www.elastocloud.org
Thank you.

https://en.opensuse.org/Portal:ARM
https://linux-sunxi.org
Have a Lot of Fun, and Join Us At:

www.opensuse.org
General Disclaimer

This document is not to be construed as a promise by any participating organisation to develop, deliver, or market a product. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. openSUSE makes no representations or warranties with respect to the contents of this document, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. The development, release, and timing of features or functionality described for openSUSE products remains at the sole discretion of openSUSE. Further, openSUSE reserves the right to revise this document and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes. All openSUSE marks referenced in this presentation are trademarks or registered trademarks of SUSE LLC, in the United States and other countries. All third-party trademarks are the property of their respective owners.

License

This slide deck is licensed under the Creative Commons Attribution-ShareAlike 4.0 International license. It can be shared and adapted for any purpose (even commercially) as long as Attribution is given and any derivative work is distributed under the same license. Details can be found at https://creativecommons.org/licenses/by-sa/4.0/

Credits

Template
Richard Brown
rbrown@opensuse.org

Design & Inspiration
openSUSE Design Team
http://opensuse.github.io/branding-guidelines/

https://creativecommons.org/licenses/by-sa/4.0/

rbrown@opensuse.org

http://opensuse.github.io/branding-guidelines/