Mindcastle.io

VERTIGO.AI

A Virtual Disk for Edge, Cloud & HPC

Jacob Gorm Hansen, jacob@vertigo.ai

Bio

- Computer Science Ph.D. from DIKU
- Inventor of VM *Live Migration* >4300 citations
- Hitman (lol), VMware, Bromium
- Founder at Vertigo.ai

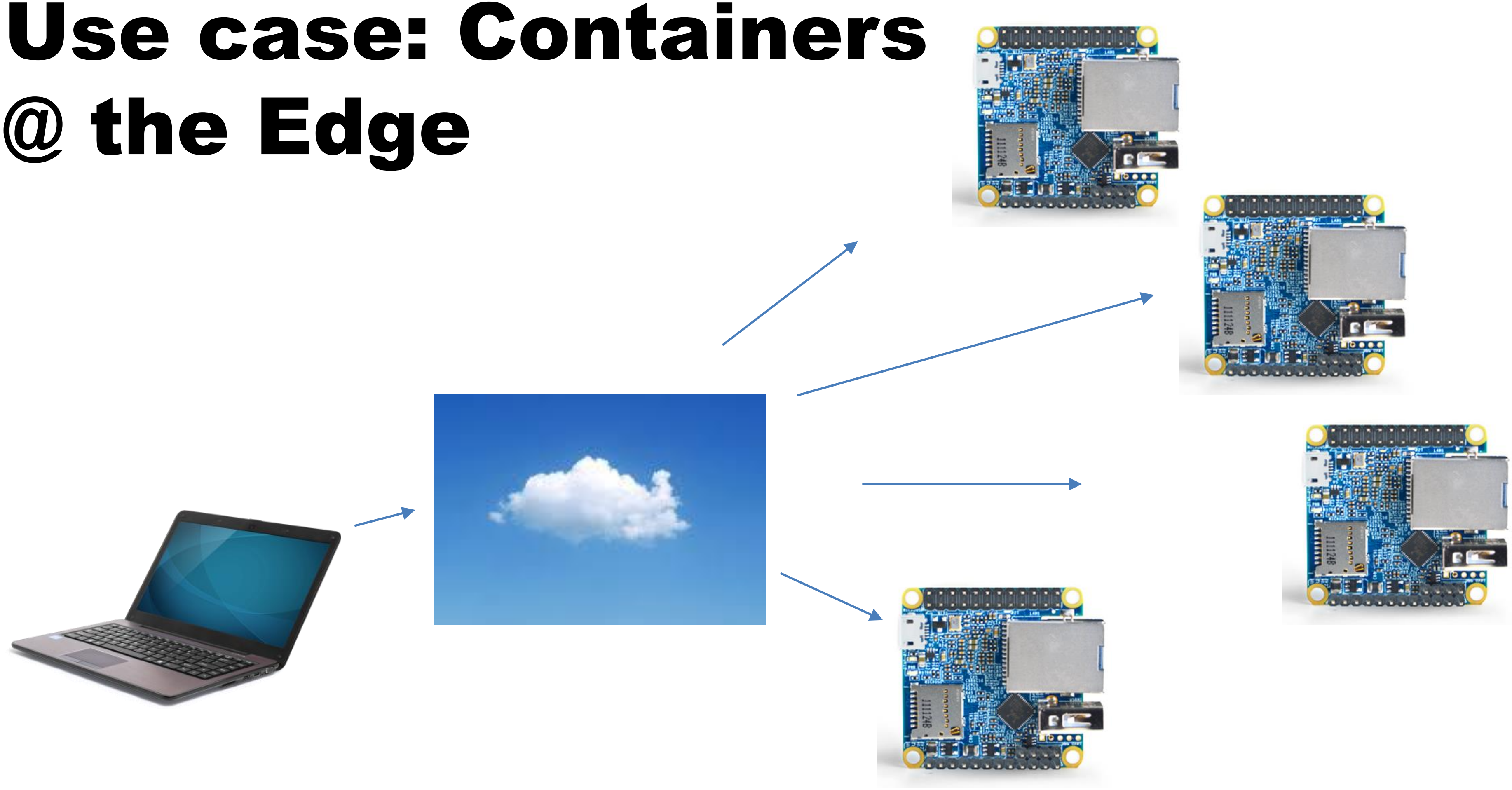
vmware®

Br Bromium®

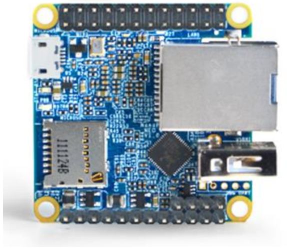
What is Mindcastle?

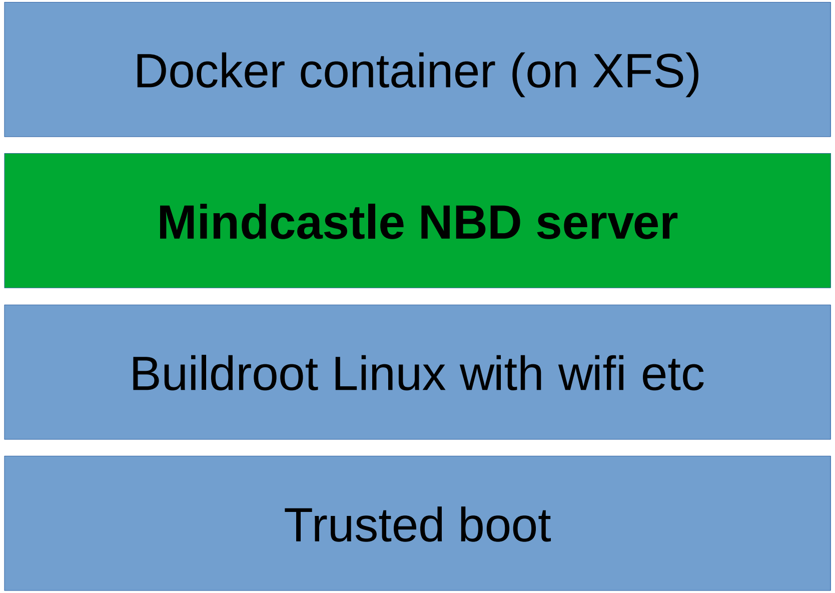
- “An encrypted distributed block device”
- “A server-less storage system”
- “Git for your storage”

Use case: Containers @ the Edge



Brick-safe Containers on Edge





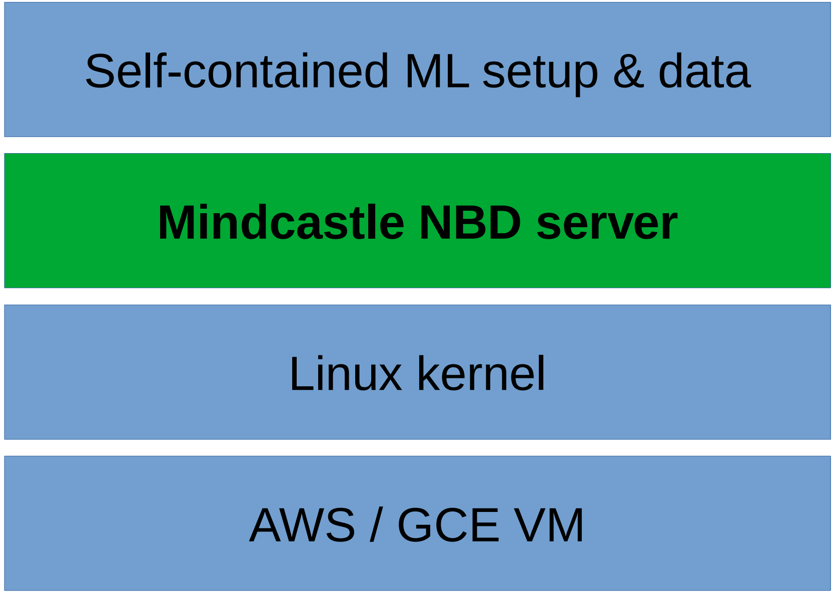
Docker container (on XFS)

Mindcastle NBD server

Buildroot Linux with wifi etc

Trusted boot

ML Training in the Cloud



Self-contained ML setup & data

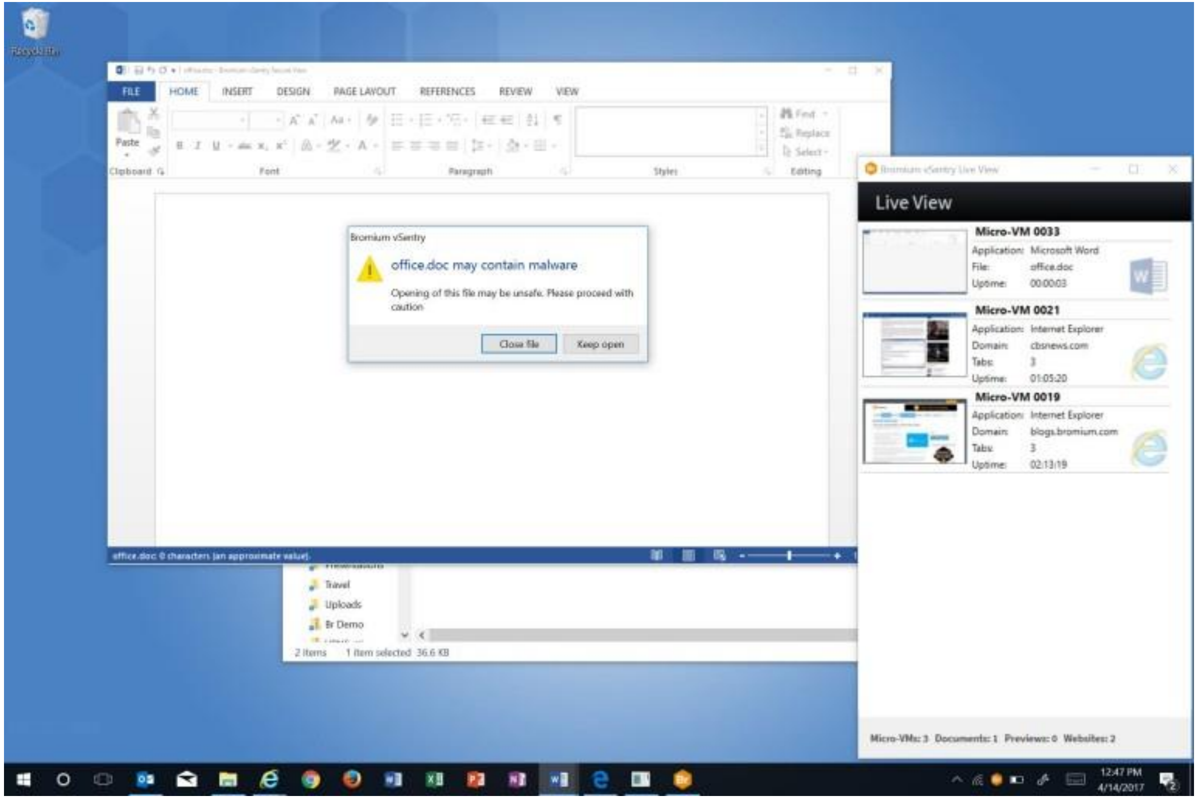
Mindcastle NBD server

Linux kernel

AWS / GCE VM

Precursor: Bromium’s SWAP disk

VM-based Isolation



Bromium vSentry

office.doc may contain malware

Opening of this file may be unsafe. Please proceed with caution

Close file | Keep open

office.doc: 0 characters (an approximate value).

Bromium vSentry Live View

Live View

Micro-VM 0033
Application: Microsoft Word
File: office.doc
Uptime: 00:00:03

Micro-VM 0021
Application: Internet Explorer
Domain: cbsnews.com
Tabs: 3
Uptime: 01:05:20

Micro-VM 0019
Application: Internet Explorer
Domain: blogs.bromium.com
Tabs: 3
Uptime: 02:13:19

Micro-VMs: 3 Documents: 1 Previews: 0 Websites: 2

Lots of VMs need lots of IO

- Possibly 100s of VMs per user
- 4GiB RAM, HDD or small SSD
- Windows needs ~20GB disk per VM
- Each VM needs ~100 IOPS
- Laptop HDD delivers ~100 IOPS

Could we use VHD or similar formats?

- Generally built like page-tables with large (e.g, 2MiB) page sizes
- Problems:
  - For every VM IO, there is a host-side IO
  - Slow on HDD, random writes kill SSDs
  - Sparse random write patterns cause space blowup

Virtual disks are like databases!

- Simple dictionaries mapping LBAs to their contents
- Databases have been solving similar problems since forever with **B-trees**
  - Lookup in **$O(\log_M N)$** IOs instead of **$O(\log_2 N)$** IOs for a binary tree
  - For **N=1M** and **M=1000** this means **2 IOs** instead of 2**0 IOs**
  - But point updates amplify writes **(M-1)** times

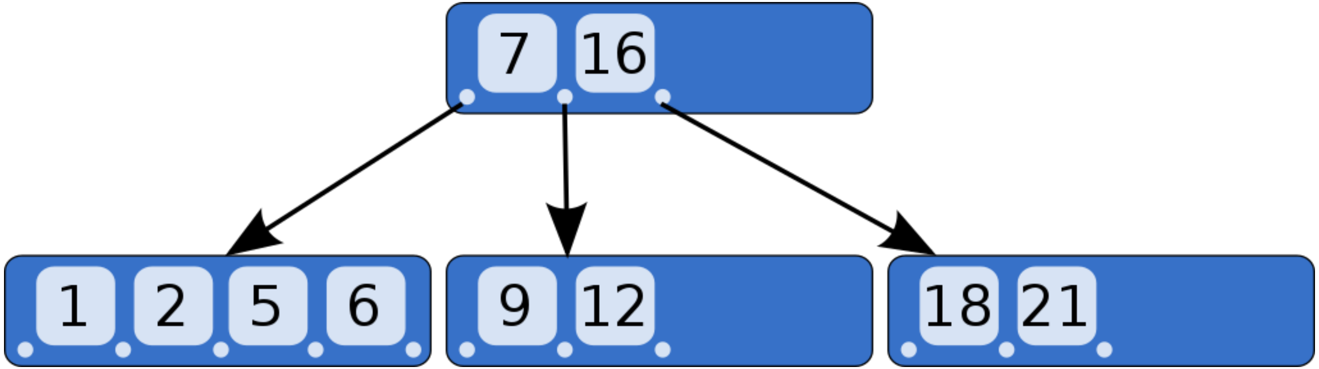
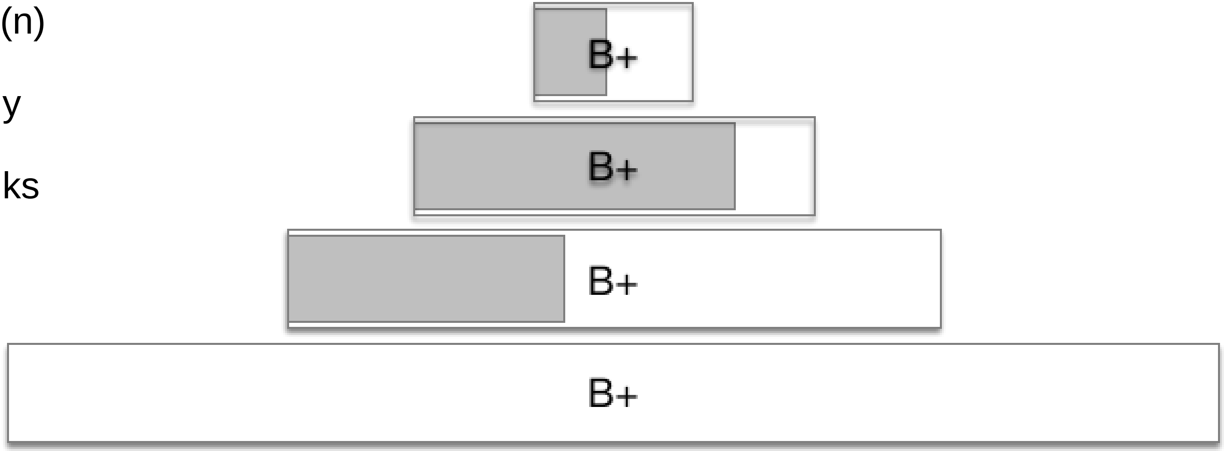


Image credit:
Wikipedia

LSM-like “dubtree” data structure for Swap

- Use a stack of B+-tree indexed logs
- Levels grow by some constant factor (16)
- When one fills up, you merge into the next
- B+-trees generated afresh in O(n)
- Keys & values stored separately
- Levels split into fixed-size chunks
- One chunk per B+-tree



Perf: SWAP vs VHD (i7-4600 SSD)

Format	VHD	SWAP
100k random 4kiB writes	426/s	75495/s
100k random 4kiB reads	30191/s	50701/s
Space used after test	16GiB	131MiB

- Using “img-test” 100k random writes, followed by 100k random reads, repeated 10 times
- 1.68X random read throughput
- 117X random write throughput
- 119X disk space reduction

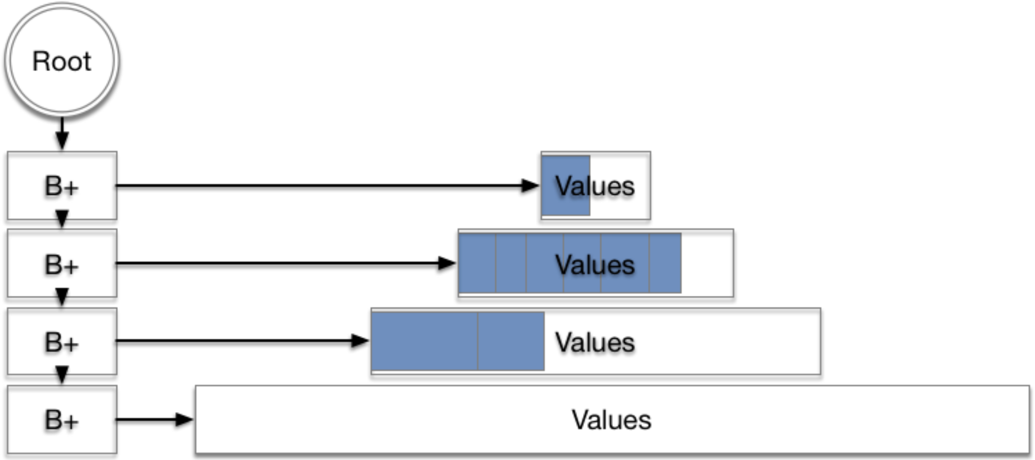
Present day: Mindcastle

From SWAP to Mindcastle

Based on Open Source release of Bromium’s SWAP, adding:

- Linux port
- Remote HTTP chunk storage
- Content hashing & encryption

SWAP += encryption and distribution



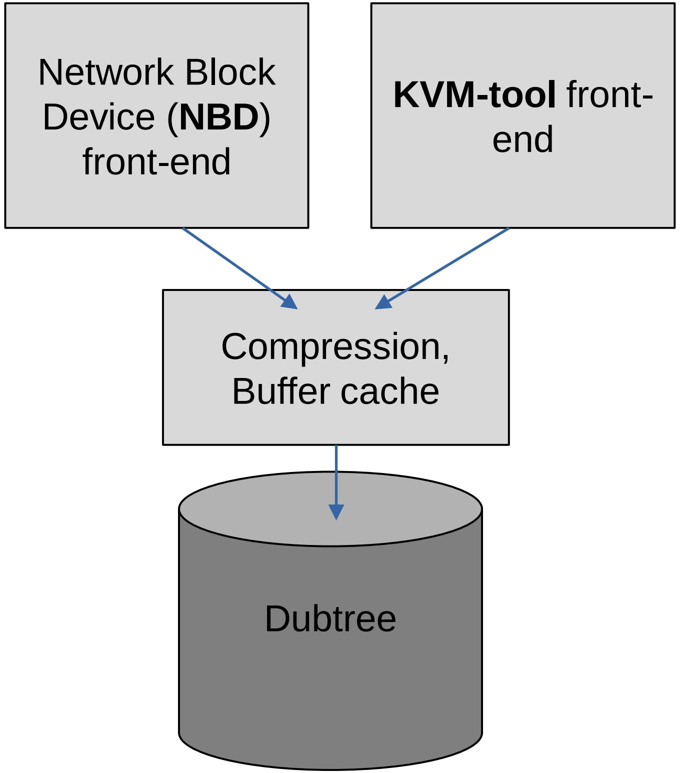
- Store Index B-trees and Levels as content-addressable chunks
- Encrypt B-tree nodes and data values individually
- Entire structure forms a Merkle-tree
- Every update yields a new tree with a new unique name

Mindcastle .swap file example

```
uuid=5d16d5a2-5870-4cd0-8b2e-bd47babb4ee9
size=104857600
key=4390126266e2cf75724313595ca94dd76280eef0fb6b5dd05f20879cf98b01b9
snapshot=2ba195097e66dd4661077635b598ed2e1556cb2d6d27d338a9c8143def98e255
snaphash=0516e503544ed89ae271fea6095cd69b
fallback=http://my-bucket.s3-eu-west-1.amazonaws.com
```

- Enough to mount a disk from anywhere, rest gets demand-fetched over HTTP(S)
- Writes happen locally, chunks can be synced back with a tool like *rclone*

Structure of the Code

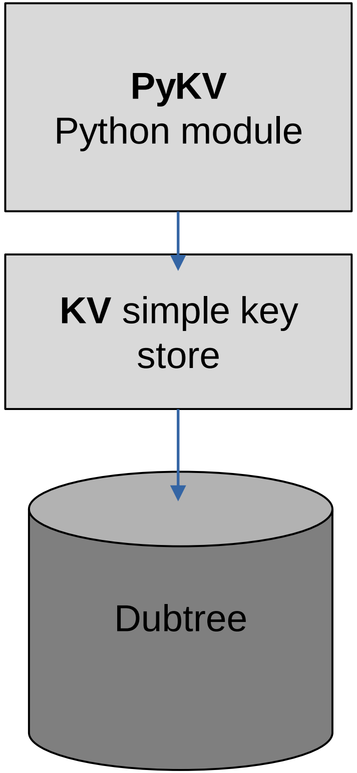


Network Block Device (**NBD**) front-end

KVM-tool front-end

Compression, Buffer cache

Dubtree



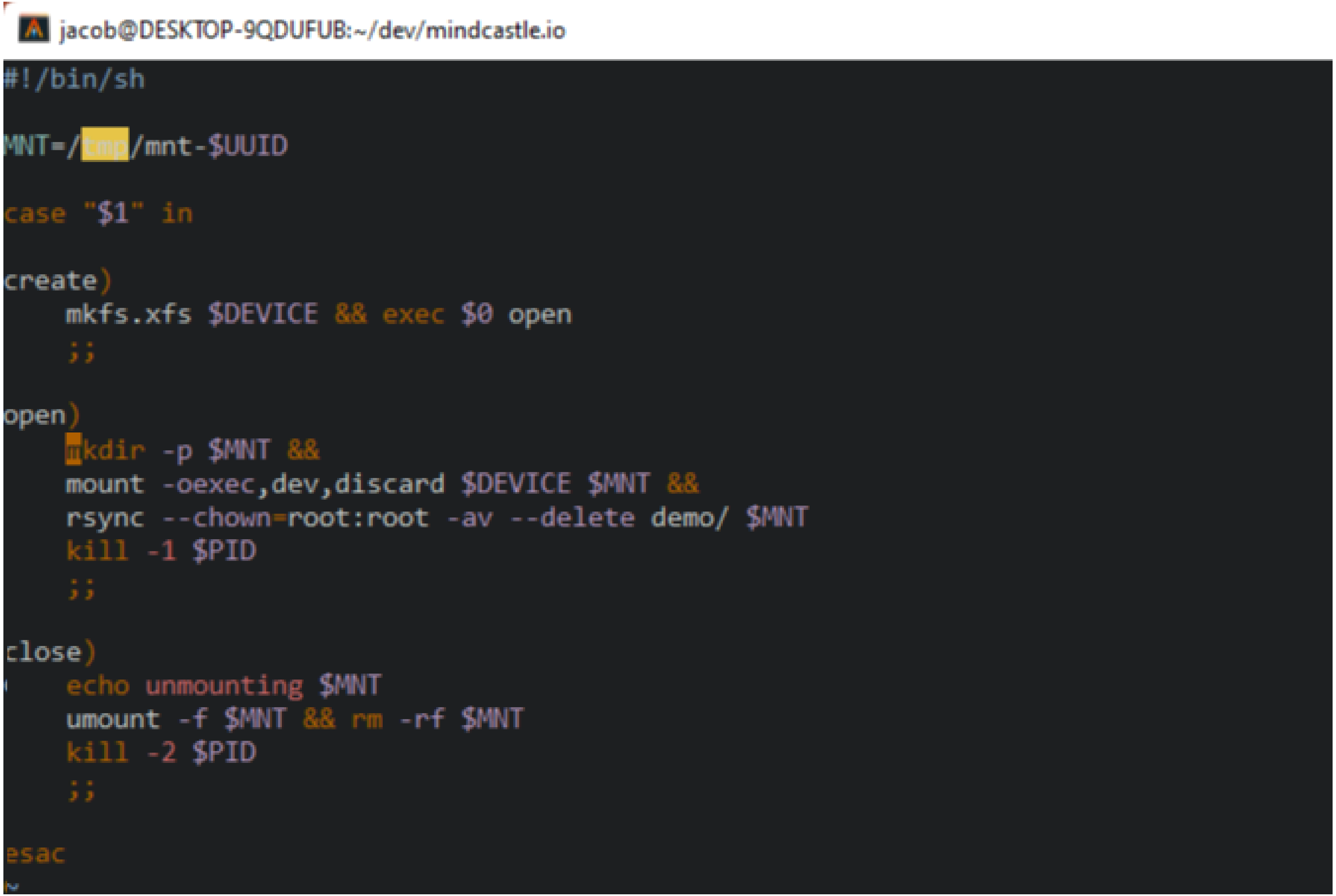
PyKV Python module

KV simple key store

Dubtree

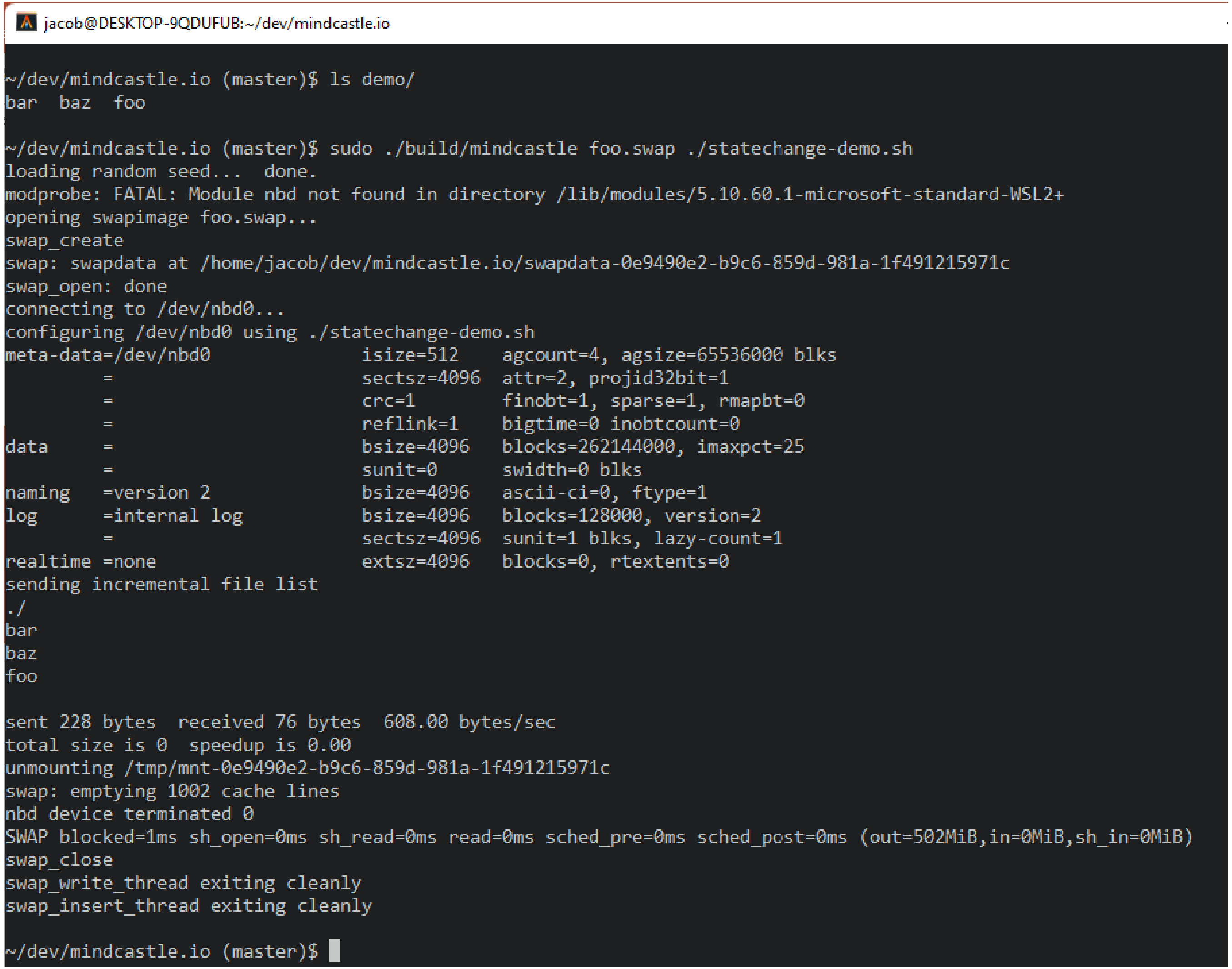
Mindcastle IO perf (i7-7700 M2 SSD)

Data transform	SHA512	SHA512 + AES256
100k random 4kiB writes	99676/s	94792/s
100k random 4kiB reads	56766/s	42554/s

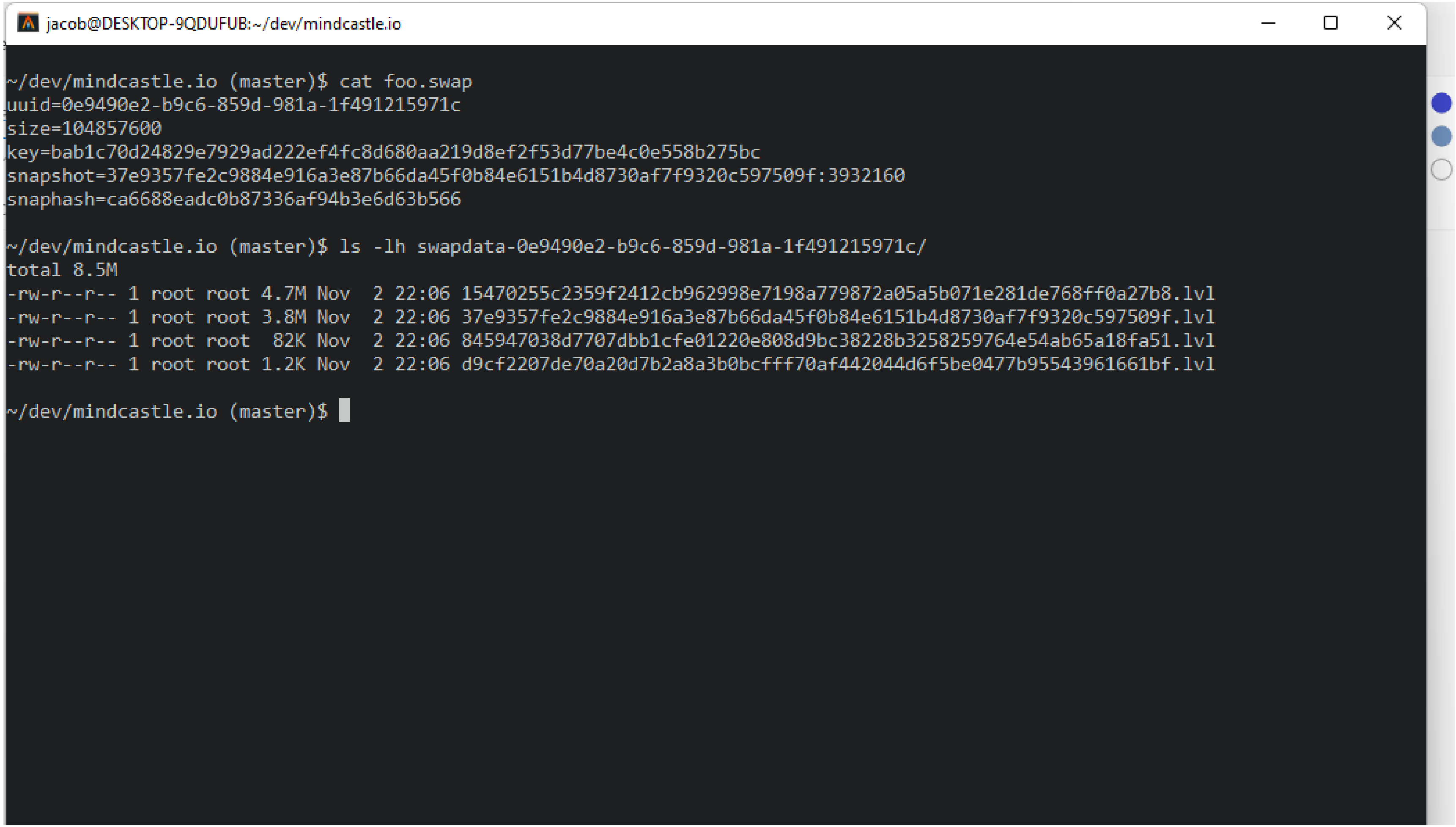


jacob@DESKTOP-9QDUFUB:~/dev/mindcastle.io

```
#!/bin/sh

MNT=/tmp/mnt-$UUID

case "$1" in

create)
    mkfs.xfs $DEVICE && exec $0 open
    ;;

open)
    mkdir -p $MNT &&
    mount -oexec,dev,discard $DEVICE $MNT &&
    rsync --chown=root:root -av --delete demo/ $MNT
    kill -1 $PID
    ;;

close)
    echo unmounting $MNT
    umount -f $MNT && rm -rf $MNT
    kill -2 $PID
    ;;

esac
~
```



```
~/dev/mindcastle.io (master)$ ls demo/
bar  baz  foo

~/dev/mindcastle.io (master)$ sudo ./build/mindcastle foo.swap ./statechange-demo.sh
loading random seed...  done.
modprobe: FATAL: Module nbd not found in directory /lib/modules/5.10.60.1-microsoft-standard-WSL2+
opening swapimage foo.swap...
swap_create
swap: swapdata at /home/jacob/dev/mindcastle.io/swapdata-0e9490e2-b9c6-859d-981a-1f491215971c
swap_open: done
connecting to /dev/nbd0...
configuring /dev/nbd0 using ./statechange-demo.sh
meta-data=/dev/nbd0             isize=512    agcount=4, agsize=65536000 blks
         =                      sectsz=4096  attr=2, projid32bit=1
         =                      crc=1        finobt=1, sparse=1, rmapbt=0
         =                      reflink=1    bigtime=0 inobtcount=0
data     =                      bsize=4096   blocks=262144000, imaxpct=25
         =                      sunit=0      swidth=0 blks
naming   =version 2             bsize=4096   ascii-ci=0, ftype=1
log      =internal log          bsize=4096   blocks=128000, version=2
         =                      sectsz=4096  sunit=1 blks, lazy-count=1
realtime =none                  extsz=4096   blocks=0, rtextents=0
sending incremental file list
./
bar
baz
foo

sent 228 bytes  received 76 bytes  608.00 bytes/sec
total size is 0  speedup is 0.00
unmounting /tmp/mnt-0e9490e2-b9c6-859d-981a-1f491215971c
swap: emptying 1002 cache lines
nbd device terminated 0
SWAP blocked=1ms sh_open=0ms sh_read=0ms read=0ms sched_pre=0ms sched_post=0ms (out=502MiB,in=0MiB,sh_in=0MiB)
swap_close
swap_write_thread exiting cleanly
swap_insert_thread exiting cleanly

~/dev/mindcastle.io (master)$ 
```



```
~/dev/mindcastle.io (master)$ cat foo.swap
uuid=0e9490e2-b9c6-859d-981a-1f491215971c
size=104857600
key=bab1c70d24829e7929ad222ef4fc8d680aa219d8ef2f53d77be4c0e558b275bc
snapshot=37e9357fe2c9884e916a3e87b66da45f0b84e6151b4d8730af7f9320c597509f:3932160
snaphash=ca6688eadc0b87336af94b3e6d63b566

~/dev/mindcastle.io (master)$ ls -lh swapdata-0e9490e2-b9c6-859d-981a-1f491215971c/
total 8.5M
-rw-r--r-- 1 root root 4.7M Nov  2 22:06 15470255c2359f2412cb962998e7198a779872a05a5b071e281de768ff0a27b8.lvl
-rw-r--r-- 1 root root 3.8M Nov  2 22:06 37e9357fe2c9884e916a3e87b66da45f0b84e6151b4d8730af7f9320c597509f.lvl
-rw-r--r-- 1 root root  82K Nov  2 22:06 845947038d7707dbb1cfe01220e808d9bc38228b3258259764e54ab65a18fa51.lvl
-rw-r--r-- 1 root root 1.2K Nov  2 22:06 d9cf2207de70a20d7b2a8a3b0bcfff70af442044d6f5be0477b95543961661bf.lvl

~/dev/mindcastle.io (master)$
```

Summary

- Mindcastle is high-performance, encrypted virtual disk accessible from anywhere
- Use it to quickly and reliably “broadcast” file system images to many nodes
  - Edge sensors
  - Cloud compute workloads
  - Containers and VMs, possibly stateful
- Other uses:
  - Versioning and “broadcasting” of large datasets
- Looking for more users & contributors

VERTIGO.AI

Questions?

(Learn more at http://mindcastle.io)