Joyent

ZFS in the Manta storage system

OpenZFS Developer Summit

David Pacheco (@dapsays)
Joyent

What is Manta?

Joyent

- Scaling the Unix philosophy to “Big Data”

What is the Unix Philosophy?

Joyent

- 1986: Jon Bentley to Don Knuth: write a program that demonstrates Literate Programming:

  “Given a text file and an integer k, print the k most common words in the file (and the number of their occurrences) in decreasing frequency.”

- Knuth’s solution: 10 pages

What is the Unix Philosophy?

Joyent

- McIlroy's solution:

```
tr -cs A-Za-z '\n' | tr A-Z a-z | \
    sort | uniq -c | sort -rn | sed ${1}q
```

The Unix philosophy

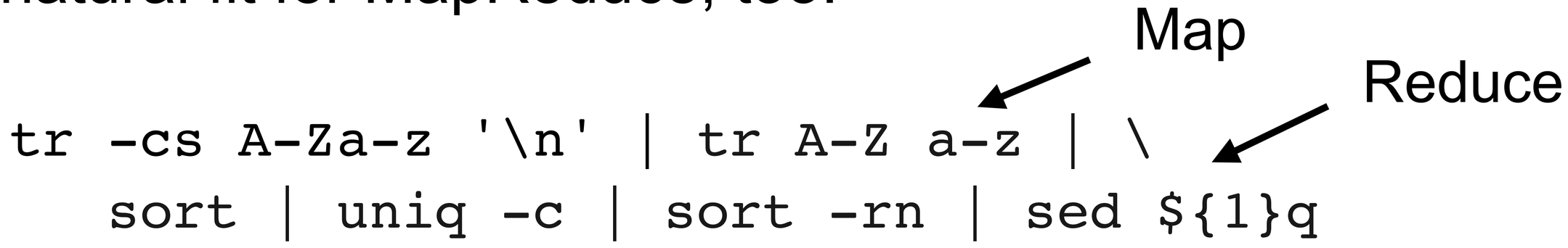
Joyent

- Small programs that do one thing and do it well
- Plus several conventions:
  - standardized input/output
  - stream processing,
  - newline-separated records, often with fields separated by whitespace (or some other character) conventions
- Not just the tools, but an **approach** to building programs

Big Data: 1986 all over again?

Joyent

- Google's MapReduce paper sets up the same problem: "*Count of URL Access Frequency: The map function processes logs of web page requests and outputs (URL; 1). The reduce function adds together all values for the same URL and emits a {URL; total count} pair.*"
- A natural fit for MapReduce, too:



Challenges bringing Unix to Big Data

Joyent

- Arbitrary-size data
- Arbitrary programs: **OS** is the abstraction
- Parallelization abstractions: map-reduce
- Multi-tenant

Storage

Joyent

- NAS is nice, but H/A NAS is a challenge for **CAP**.
- Object stores look like a file system, but:
  - No partial updates => consistency only affects metadata
  - No volume management
  - Universal protocol (HTTP)
- Internally: store objects as files for computation

OS Containers

Joyent

- One kernel on bare metal, many virtual OS containers (“zones”), each with its own root filesystem
- Much more efficient than hardware-based virtualization
- “root” in the zone does not compromise the rest of the system
- Rich interface between “global zone” and individual tenants’ zones

Putting it altogether: Manta

Joyent

- Scalable, durable HTTP Object Store
- Namespace looks like a POSIX filesystem
- *In situ* compute as a first-class operation
- Quick demo

Bentley's challenge, scaled up

Joyent

- Arbitrarily scalable variant of McIlroy's solution to Bentley's challenge:

```
mfind /manta/public/examples/shakespeare | \
mjob create -o -m "tr -cs A-Za-z '\n' | \
tr A-Z a-z | sort | uniq -c" -r \
"awk '{ x[\$2] += \$1 }
END { for (w in x) {
    print x[w] \" \" w } }' | \
sort -rn | sed ${1}q"
```

Manta overview

Joyent

- Frontend: SSL, LB, API servers
- Metadata: postgres databases
- Storage: dedicated servers, nginx over ZFS + zones
- (plus a bunch of other stuff)

ZFS in Manta

Joyent

- Deployment
- Storage
- Compute
- Metadata replication

ZFS for deployment

Joyent

- Components deployed as zone **images** (similar to what Docker is popularizing)
- Images are just zfs datasets
  - Components are delivered as incremental changes from a base image
  - SDC takes care of distributing images
- Dynamic sizing: zones can be granted more disk space with “zfs set quota”

ZFS for storage

Joyent

- Durability: kind of important
  - COW
  - RAIDZ2
  - Checksums
- Fast ZIL device (that's client latency!)
- No volume management
- Well-known to this crowd, but these pieces are **critical**.

ZFS for compute

Joyent

- User tasks get their own zones, with their own filesystem
- 128 zones per system
- Base image has 9000 packages installed,
  using 36GB of disk space, so clones are clutch
- Users can ask for extra disk space: we just “zfs set quota”
- Filesystem is writable: users can do whatever they want.
  When they’re done, we **zfs rollback**.

On zfs rollback ...

Joyent

- Since launch, Manta has done ~16 million rollbacks (almost two per minute per system, on average)

illumos#4504: space map corruption

Joyent

- July, 2013: saw first panic in space_map_sync()
- December, 2013: first machine entered panic loop (space map corruption on disk)
- At least one other machine entered a panic loop; ~10 others showed signs of on-disk space map corruption
- Matt Ahrens and George Wilson diagnosed the problem, and Keith Wesolowski built tools for analyzing the space maps on disk and developed procedures for booting corrupted machines
- It was a dark time.

illumos#3821 rollback + zil race

Joyent

- We continue to see this issue periodically
- Two failure modes: panic, hang
- Manta survives panics quite well
- Manta does not deal well with OS hangs
  (typically all zone rollbacks hang on that system, eventually causing most jobs to hang)
- This issue seems streaky, and the D script for gathering more data seems to make it less likely to happen.

ZFS for metadata replication

Joyent

- Metadata tier: postgres shards
- Synchronous replication for durability
- When new peer shows up, use zfs send/receive to bootstrap replication

Key takeaways

Joyent

- Unix loves Big Data
- ZFS enables us to build a multi-tenant distributed storage system without having to worry about the storing-bits-on-disk problem
- ZFS's pooled storage model is a key enabler for transient OS containers

What we'd love from ZFS

Joyent

- zdb enhancements
- hardening for "zfs receive" and "zpool import"
- continued stability: FCS quality all the time

References

Joyent

- “Programming Pearls: a literate program”: http://dl.acm.org/citation.cfm?id=315654
- “MapReduce: Simplified Data Processing on Large Clusters” http://research.google.com/archive/mapreduce.html
- More on Manta: https://github.com/joyent/manta

Joyent

ZFS in the Manta storage system

OpenZFS Developer Summit

David Pacheco (@dapsays)
Joyent