

SFD8 - Day 1 - Coho Data

Andy Warfield (@andywarfield)

1. The private cloud is not dead

rack-scale data platform for private clouds

storage and nw switching

big challenge is locality. You should be able to scale storage and compute as required, and make sure traffic goes between them as efficiently as possible

enterprise it as insulation against change

- the rise of commodity hw
- all sorts of new storage media
- More vendors, more “evolving” vendors
- public vs private cloud debate
- new applications. really.

Coho customer profile

Private cloud environments (ITaaS Provider)

risk averse on capex

traditional budget being challenged

trying to build something that competes with an AWS style storage model

Our customers have a run-rate requirement (and budget) for storage, desire to reduce hardware mgmt overhead

need to support data analytics, app services, corp app development

Old IT - boring. necessary. funded

New - emerging but aspirational

Coho wants to be:

a data-centric bridge for evolving enterprise infrastructure,
“storage transformation without disruption for the modern enterprise”

Since last year

- Shipped an AFA (NVME + SSDs - 50 useable TB in 2RU) - compression, thin provisioning, thin clones. No “dedupe” per se.
- Added support for OpenStack and XenServer
- added a pile of boring but important enterprise features (snmp mibs, AD)
- spent a lot of time maturing scale, stability, support and performance
- incorporated docker into the storage platform
- big data reference architecture

shout out to support and engineering

- challenges of building storage with diamonds
- workload numerology
- docker in your (big) data

2. the architectural anxiety of medium-sized storage

use a hierarchy of different types of flash and integrate with the network to expose its performance

- can mix NVMe flash, SSDs and spinning disks
- enterprise NVMe is used for primary storage
- lot of effort goes into managing placement and using newer, faster memories

the switch is completely integral to the delivery process

a ton of work to get the integration right when customers plug their stuff into the Arista switches

building for 1000 or 10000 nodes - you do things that don't make sense for 10 or 100 nodes (FB and google)

Legacy OS / application code that I can't change
inconvenient storage protocols (NFSv3, iSCSI, HDFS)

Virtualisation was the financial argument ...

diamonds are expensive, precious and hard to divide - pic.twitter.com/Xv0DVo560I

server-side work - struggling to drive the load to make it worthwhile

the memory hierarchy - pic.twitter.com/ZcflzOe5MB, pic.twitter.com/EUfgWpP7ca

forwarding decision times are becoming shorter (request processing)
placement decision times are becoming longer.

you need a pile of cores to work with this stuff now

what happens when you add contention? Drop in performance - this is a big design concern

"Storage data paths are like network data paths. In hell."

- everything is address translations
- request forwarding semantics are ...

Summary

-building "cloud" storage at scale is completely different than building storage at FB, google or AWS

- storage designers today have a harder job than they have ever had in the past. traditional layout, core-level performance, service-model of delivery
- Customers don't care, by and large, about this pain. They just want to scale on demand.

[whiteboard]

on-disk layout has been revised at least 20 times ...

3. Workload numerology understanding workloads

“it’s not just a matter of speeds and feeds”

manage the placement of bot data and network traffic based on detailed workload analysis

- use the notion of the SDN “controller”
- Coho FlashFit
- Coho OnStream - send detailed data back to Coho

traditional storage worried about durability, and getting high performance off crappy hardware

modern storage design is about solving a connectivity and locality problem, rather than a durability problem

[OnStream dashboard demo]

Described as “analytics backhaul”

performance based placement decisions benefit from workload characterisations, but characterising things like working sets are expensive in time and space

counter stacks efficiently encode the cardinality of uniqueness over time

Hyper Log Logs “HLL” - <https://en.wikipedia.org/wiki/HyperLogLog>

All of the log data is available via REST

Summary

Understand working set characteristics :)

4. Docker in your (big) data

Moving extensible compute into the storage system

allow the storage system to be extended using container-based computation

- challenge idea that storage system should provide (1 of 4) protocols
- quickly analyse stored data by launching Apache spark clusters
- incorporate 3rd-party appliances adjacent to data
- allow dev teams to push code to run directly within the platform

containers run on dedicated cores, memory ... very careful not to mess with the corp competency

thread - process - container - virtual machine

micro service - load-balanced collection of containers that do one thing very well

big data environments as a container use case

1. decide on distort (cloudera/horton/mapreduce/etc)
2. procure a bunch of hw (even VM-based deployments need different hw)
3. stand it up
4. figure out how to move data in and out

FRICTION

5. application team can start writing code

[Coho System Architecture diagram]

Use case with UBS and Intel

23 nodes

Open vswitch style switching (<http://openvswitch.org>)

micro services running on containers on the storage system - this is pretty cool
(transcoding example, etc)

Config API: Google's Kubernetes - <http://kubernetes.io>

www.cohodata.com/bigdata/

Summary

Docker itself is just a tool

- pretty fantastic way to move compute closer to data
- relatively easy to integrate with Coho
- integrating 3rd-party tools allows customers to reduce time-to-insight
- exciting stuff is coming with this.