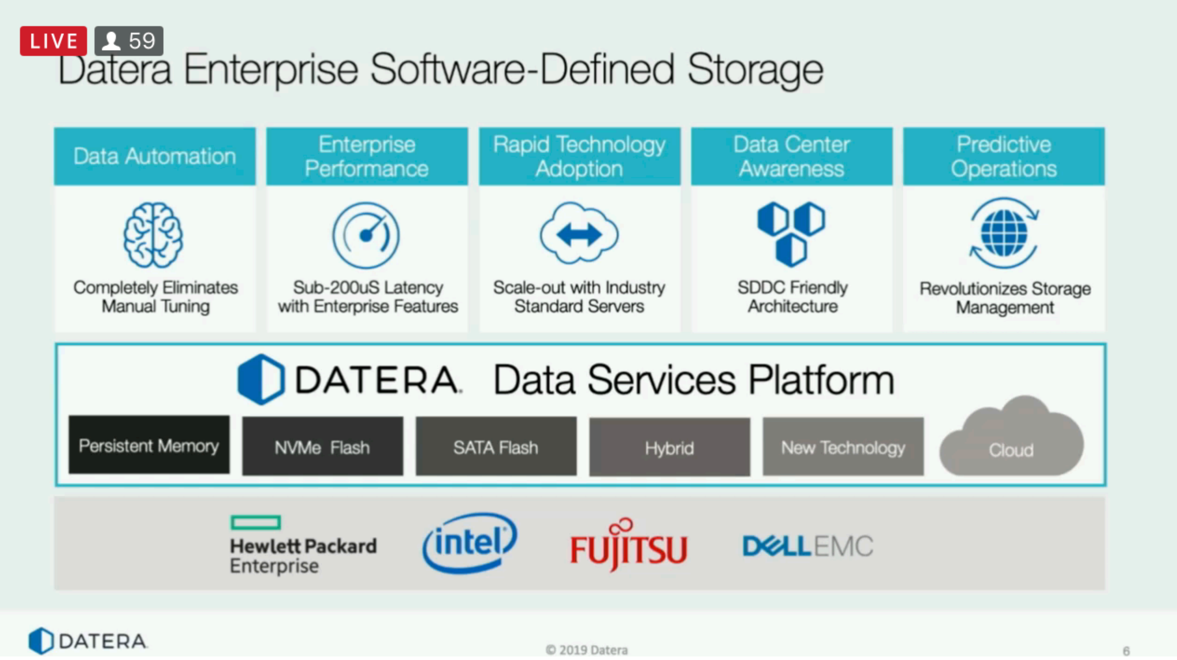
SFD18 - Datera @DateraInc

*Bill Borsari
Enterprise Software-Defined Storage

Enterprise gives you:
High Performance
Enterprise Features
- QoS
- Fault Domains
- Stretched Cluster
- L3 Networking
- Deduplication
- Replication
HA
Resiliency

Software-defined gives you:
Automation
DC Awareness Agility
Continuous Availability
Targeted Data Placement
Continuous Optimisation
Rapid technology adoption

Combine both of these and you get “Enterprise Software-Defined Storage”



Datera Enterprise Software-Defined Storage

Data Automation	Enterprise Performance	Rapid Technology Adoption	Data Center Awareness	Predictive Operations
Completely Eliminates Manual Tuning	Sub-200uS Latency with Enterprise Features	Scale-out with Industry Standard Servers	SDDC Friendly Architecture	Revolutionizes Storage Management

DATERA Data Services Platform

Persistent Memory | NVMe Flash | SATA Flash | Hybrid | New Technology | Cloud

DATERA
© 2019 Datera
6

With Datera you can choose to stay on the bleeding edge rather than having to wait.

Technology is designed around standard servers, with media, federated as a standard storage array

Support a number of industry standard servers

“Any server could be a storage server, but not every server is a good storage server”

Top Use Cases
- Enterprise DCs
- DB Acceleration
- VMware
- Containers
- Service Providers

Datera Data Services PLatform
- Persistent Memory
- NVMe Flash
- SATA Flash
- Hybrid
- New Technology
- Cloud

VMware - not a workload, it’s an environment in which we run stuff. The stuff you get the day-to-day done on

Why Modernise DC storage?
Scale
- PODs giving way to rack or row scale
Automation
- dedicated systems move to software solutions
New Apps - New Networks
- massive shift in applications and their traffic patterns
Agility
- get faster or get “Ubered”

Don’t support FC. Not that they couldn’t do it, they chose not to. Opportunity for education. Ethernet can be an organisational channel - packets are owned by the networking team.

Driving Agility with Availability
- traditional storage systems traded agility to achieve availability
- just SDS does not magically fix this
- to maintain availability with agility
-- step 1 - scale out
-- step 2 - Flexibility
-- step 3 - Self-Healing
-- step 4 - policy based

Why Datera?
L3 Networking
- Datera brings standard protocols with modern networking to DC storage. In the Datera solution resources are designed to float to allow for agility, availability, and scalability

Policy based Operations
- Datera was built from day 1 with policy controls and policy templates to easy operations at scale while maintaining agility and availability
Targeted Data Placement
- ensure data is distributed correctly across the physical infrastructure to meet policies around perfromance, availability, data protection while controlling cost

*Shalesh (?)

Applications and Portability
- break monoliths into micro-services
- Have application control its environment
- abstract resources and their bindings to its consumers
- have related services packaged as single entity that keeps them together
- App can recover / resume from complete or partial failures

Why Containers?
- Modern applications are moving away from monolithic to micro-services
- Modern apps like Cassandra, MongoDB, Spark etc, are built for scale and do not want to have platform affinity
- Efficient - better resource utilisation compared to VMs
- Agility - Faster to provision, deploy, scale

“Bare metal’s back, we just don’t call them physical servers anymore”

Containers and Kubernetes
- Applications are looking at K8s to run their modern applications
- Container-centric management platform
- Orchestrates compute, network, and storage infrastructure for workloads
- Platform for building an ecosystem of components and tools - make it easier to deploy, scale, and manage applications
- Defines applications using Pods and Services

What is a Kubernetes Pod?
- Smallest unit of deployment, representing an application using one or many containers
- Encapsulates storage resources and networking configuration
- Binds containers and storage resources as single manageable entity
- Failures, deployments are handled at this granularity

Modern Applications and Labels
- Application provisioning is based on intents, AKA labels / hints
-- performance (faster access)
-- Protection (snapshots, backup)
-- Isolation (resources separation)
-- Persistence (keep it forever or forget soon)
- Datera is built ground up to handle this
- Defines labels internally for legacy applications
- Consume labels / hint from modern applications

Datera is built for today and tomorrow
- Datera defines resources (storage, applications, configurations and pools) and behaviours (policies)
- Policies define the app-resource(s) mapping
- Applications are allocated resources to match the intent
- System is constantly optimising resources to achieve the best efficiency, placement, and performance

Apps and resources - under the hood
- Apps are running in public clouds
- Apps have requirements for compute and storage
- Apps are allocated resources matching their requirements under the hood
- Datera automates this process by having a resource manager and optimiser

Choose Your Difficulty Level
- Templates and policies
  - Define behaviours
  - Intent driven blueprint (skeleton)
  - Allows to package multiple policies together
  - Tenancy based control of behaviours (resources are assigned to tenants)
- Users are exposed to details based on their role
- Advance users can define the policies / templates and others can consume them
- Brings simplicity and consistency to multi-user and scale environments

Modern Applications Work Great With Us!
- Consistency and standardisation across scale deployments
- Deployment velocity, when you need containers fast!
- Framework defined namespaces are extended into Datera tenants, keeps it consistent
- Enables graduation of applications from QA / Dev to production without customising the app for the environment
- Future proof deployments by hiding advance Datera features behind templates

*Demo