Google Developers

Supercharge Your Google Compute Engine App with Persistent Disk

Jay Judkowitz, Sr. Product Manager
judkowitz@google.com, @Jay_Judkowitz
Andrew Kadatch, Staff Software Engineer,
akad@google.com

Agenda

1. What is Persistent Disk (PD)
2. What is PD used for
3. How does PD work
4. What are PD's main beneficial properties to developers
5. How to get the most out of PD
6. How to protect PD volumes

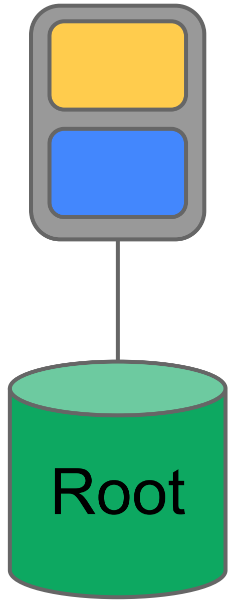
PD Overview

Context, Architecture, Properties, And Developer Impact

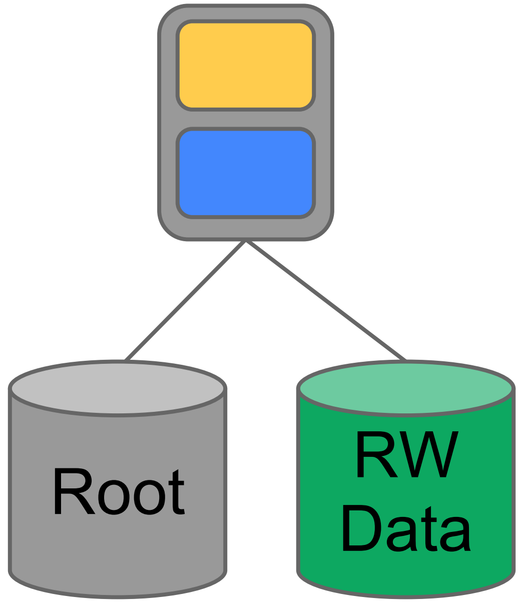
Overview: PD In The Context Of Cloud Platform Storage

Google Cloud Storage Option	Interface	Use Cases
Google Cloud Storage	Blob. Get/put semantics.	Static content served globally Media files Code artifacts
Google Cloud Datastore	NearSQL	Non-relational, schemaless data for highly scalable applications
Cloud SQL	SQL	Managed MySQL for small-medium OLTP and LAMP applications
BigQuery	File. Get put semantics.	Uploading data for BigQuery queries
Scratch Disk (formerly Ephemeral Disk)	Block. Virtual SCSI device. Destroyed with VM	OS boot volumes for stateless GCE VMs Non-shared scratch space for GCE VMs
Persistent Disk	**Block. Virtual SCSI device. Persistent until deleted.**	**OS boot volumes for stateful GCE VMs** **RW data volumes for customer managed datastores** **RO sharing of static content across many GCE VMs in one region with high performance**

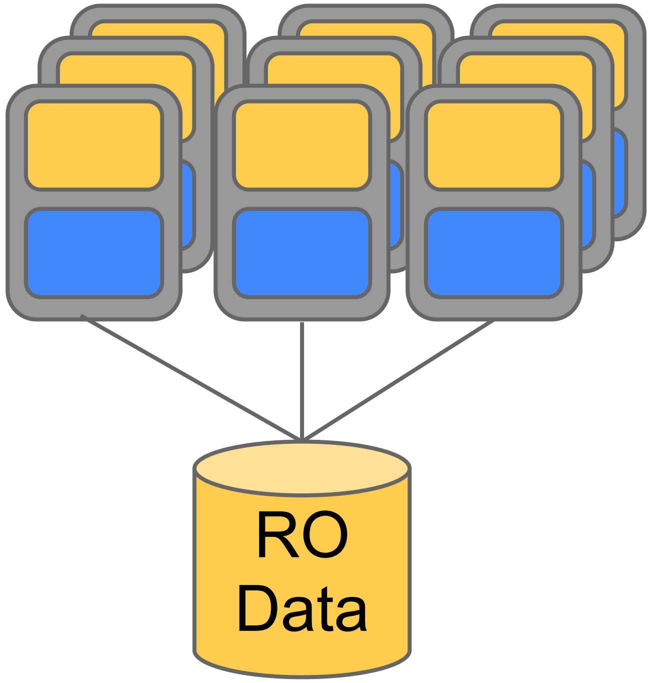
PD Use Cases



Stateful root volume



User managed datastore data volume



Instant distribution of static content

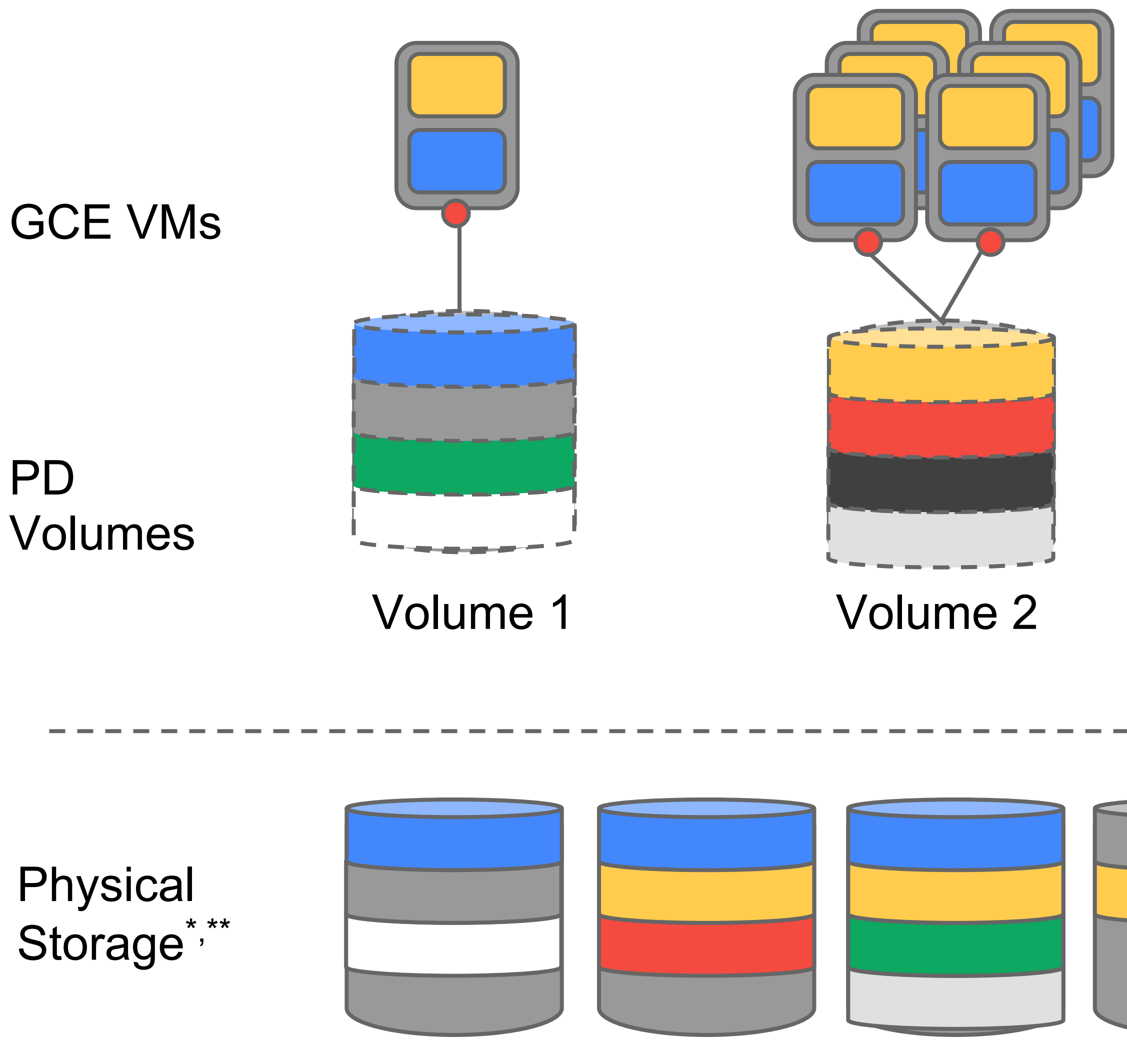
Example Use - MapR

"We have been running the MapR Distribution in the cloud for some time. We have been pleased with the raw speed and consistent performance of Google Cloud Platform's Persistent Disk. For customers who need Hadoop data to persist and be available to Elastic Map Reduce at a moment's notice, Persistent Disk is a great option."

Jack Norris

Chief Marketing Officer, MapR Technologies

PD Architecture (Simplified)



Security:

- Encryption of data in flight and at rest (AES-128-CBC)

Resiliency & Integrity:

- Multiple copies of all data
- End to end HMAC checksums

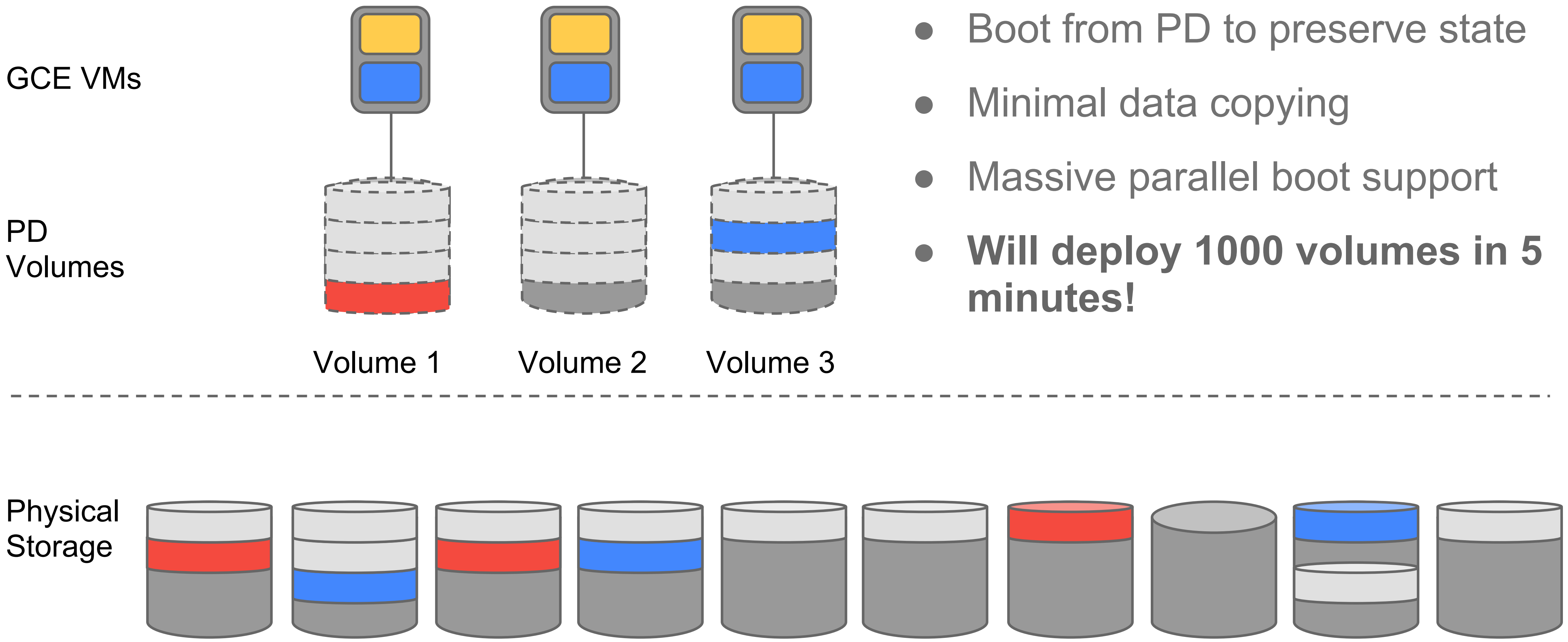
Performance characteristics:

- Random write optimizations
- Read from fastest copy

* On-disk layout may change over time taking into account price and availability and durability requirements

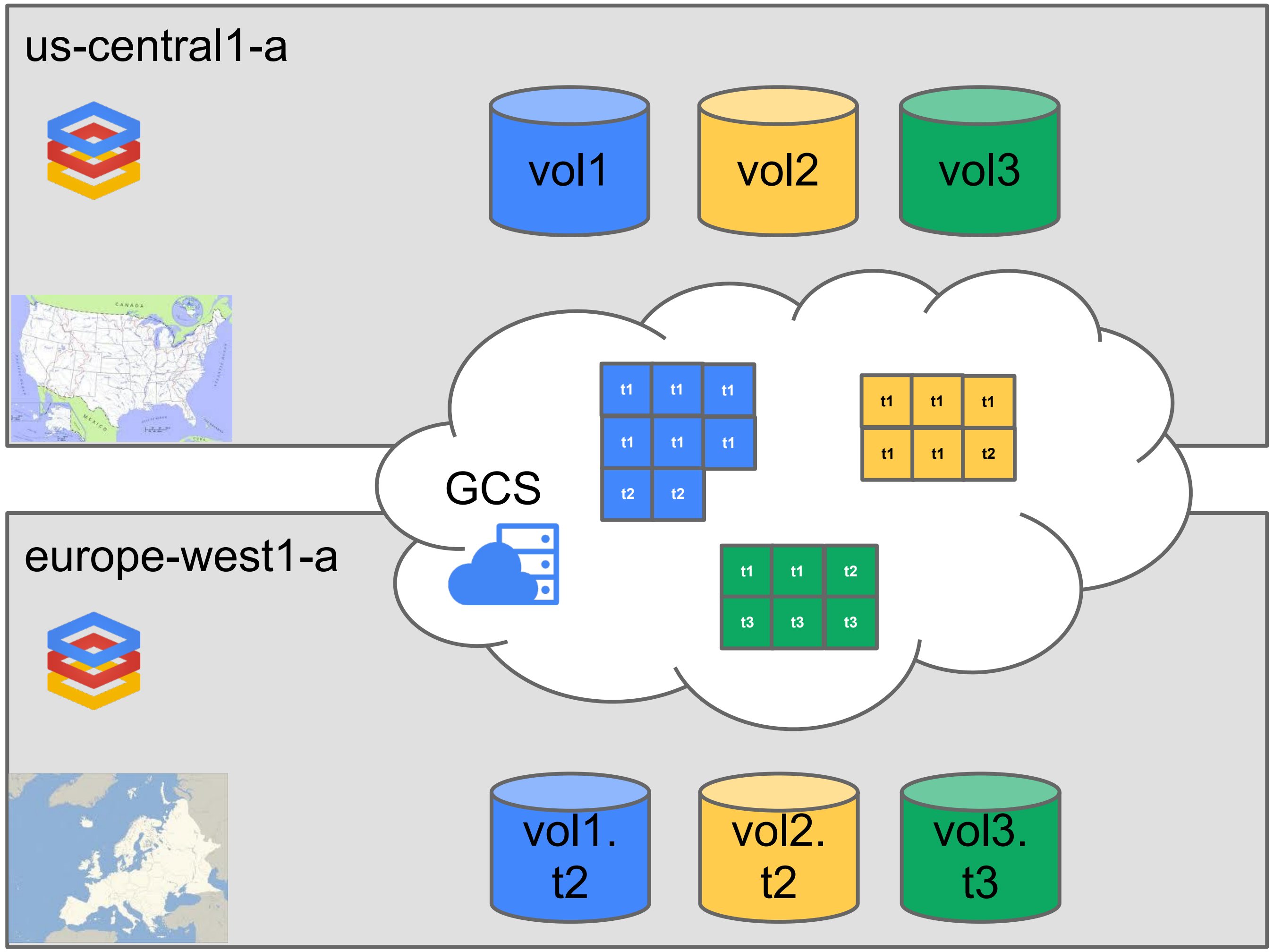
** Specifics of replication/encoding vary as needed for performance and durability

Feature: Boot From PD (Simplified Diagram)



- Boot from PD to preserve state
- Minimal data copying
- Massive parallel boot support
- **Will deploy 1000 volumes in 5 minutes!**

Feature: Snapshot And Restore With GCS



- Point in time snapshot to Google Cloud Storage (GCS)
- Differential snapshots
- ***GCS global replication!***
- Restore from snapshot anywhere in the world

Using PD

Simple Command Line Tool

Create 100GB disk in us-central2-a:

```
 $ gcutil adddisk --size_gb=100 --cell=us-central2-a disk_name
```

Create a new virtual machine with 2 extra disks, one of them read-only:

```
 $ gcutil addinstance --disk=disk_name1 --disk=disk_name2,mode=read_only
vm_name
```

Attach disk to a running virtual machine:

```
 $ gcutil attachdisk --disk=disk_name vm_name
```

Create new disk snapshot and copy to another disk:

```
 $ gcutil addsnapshot --source_disk=disk_name snapshot_name
 $ gcutil adddisk --source_snapshot=snapshot_name disk_name
```

Full documentation: http://goo.gl/v4wbB

PD Volumes In The VM

PD volumes will show up in the VM as VirtioSCSI disks.

- Name of device inside VM's made from Google provided images will be `/dev/disk/by-id/google-<disk_name_at_create_time>`

Formatting and mounting data volumes: http://goo.gl/bv8dM

Making PD data volumes mount every time you launch

- Option 1: Boot from PD, any changes you make the fstab file will persist
- Option 2: Booting from scratch disk
    - Use startup scripts and custom attributes to make the VM do what you need at deploy time (http://goo.gl/Ha97f and http://goo.gl/E82me)

Current PD Limits

The limits are as 5/15/2013

Volume Size

- PD volumes now can be up to 10TB.
- VM size determines what can be mounted

Instance type	Max attached PD space	Max attached devices
f1-micro, g1-small	3TB New!	4
n1-*-1, n1-*-2, n1-*-4, n1-*-8	10TB	16

IO Rates: For now, to help consistency of performance, PD IO is limited to the lowest of

- 123 MB/s
- 239 read IOPs
- 601 write IOPs

Billing

The prices are as 5/15/2013

Pricing

- $0.10 / GB / month
- $0.10 / million IOs

Billing granularity: Sub 5 minutes

How to see the current bill for the month

- Go to the cloud console: http://goo.gl/10uwD
  - Click on the project, click on the gear, select billing, see line item for PD
- Console is updated daily

PD Best Practices

Background: What We Do At Google

We strive to be great at:

- Durability
- Security
- High availability
- Consistency of performance
- Speed

Some results still depend on how the product is used, hence the best practices...

Protection: Backups & DR

Backups will always be required...

- Differential snapshots = efficient backup and restore
- Snapshot data is geo-distributed
- Volume performance is not affected when snapshot is taken
- Automate the process and schedule regularly
- Journaling file systems ensure most reliable recovery

Protection: Snapshots

Steps to make snapshots safely:

1. Do not use root volume for application data
2. Flush and freeze* the application
3. Flush and freeze* the file system (freezefs for ext3/ext4)
4. Make a snapshot and wait until it is READY or UPLOADING
5. Unfreeze* the file system
6. Unfreeze* the application

*Freezing/unfreezing may be not required for some applications and filesystems.

Protection: Google Planned Maintenance

Best response is no response: Architect "Cloudy" Applications:

- Geo-distribute of compute and data, load balance & auto-scale
- See "Building Robust Systems" talk by Marc Cohen (May 17, 2:00PM - 2:40PM, Room 2)

For more traditional applications, when maintenance warnings come, schedule migration of VMs to another zone

- Use the "gcutil moveinstances" command
- Use your own migration script

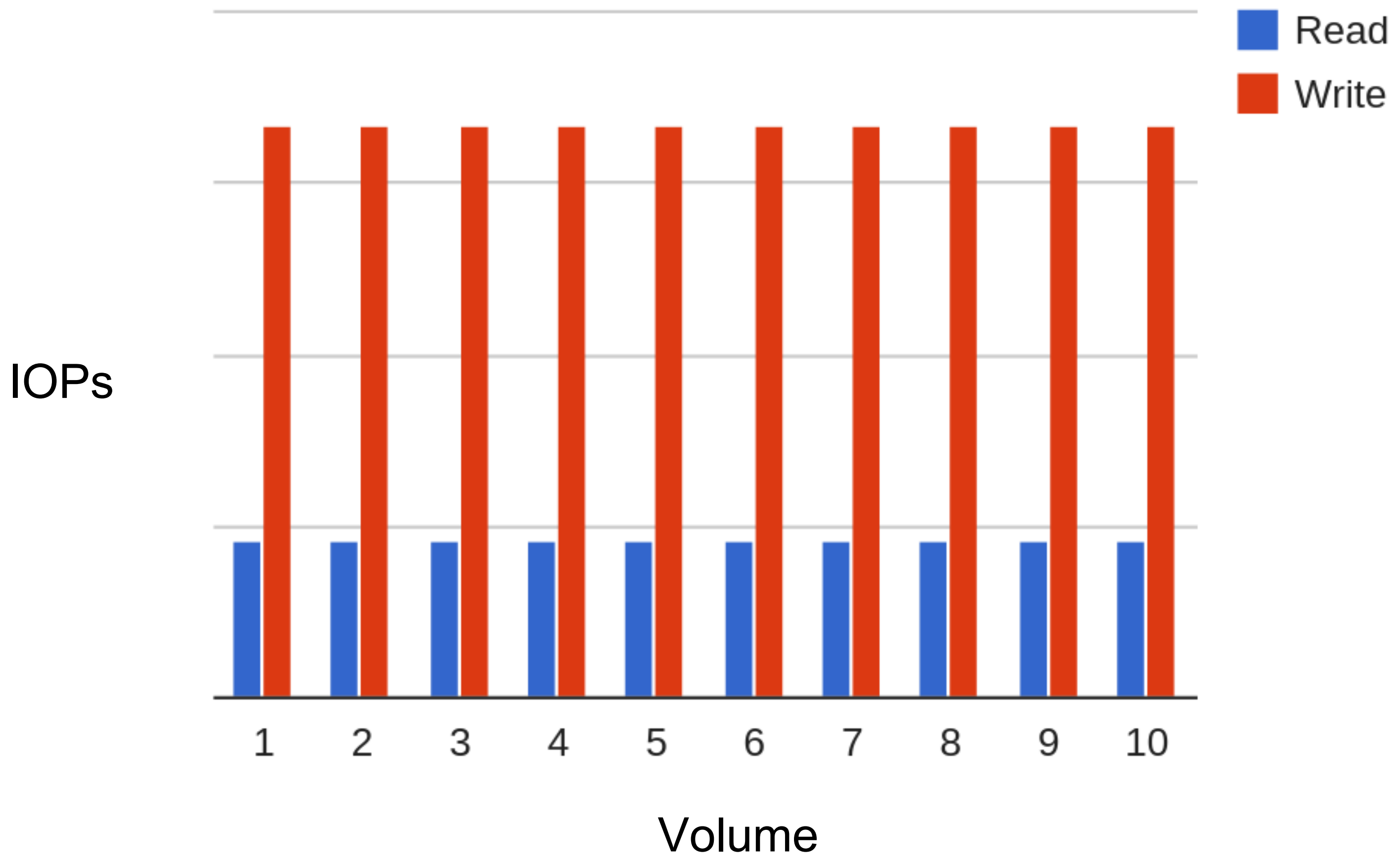
Protection: Backup, DR And Planned Maintenance

Test Regularly!!!

"Amateurs back up. Professionals restore."

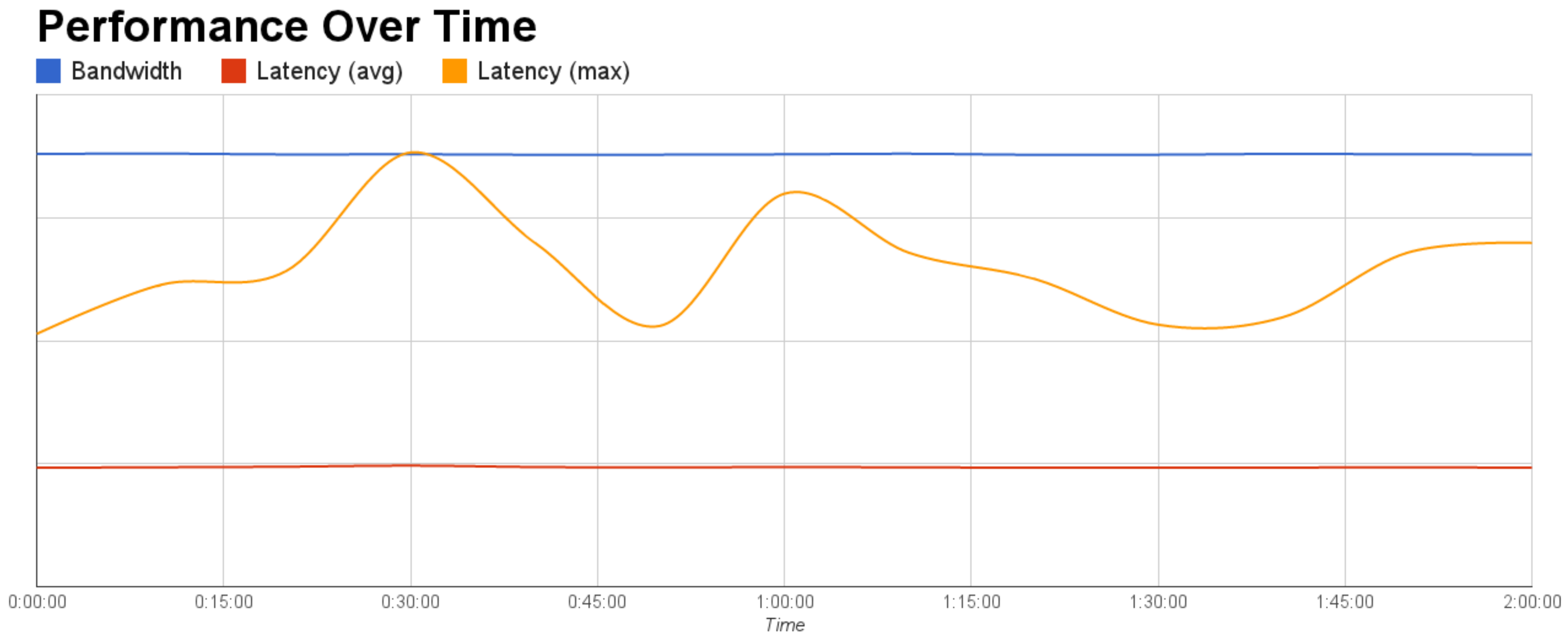
Performance: Creating Volumes - Part 1

Creating many volumes, testing their performance, and throwing away slow volumes ***will not*** yield better results



Performance: Creating Volumes - Part 2

No need to "warm" a volume before use by writing to all blocks



Performance: RAID'ing & Striping Volumes

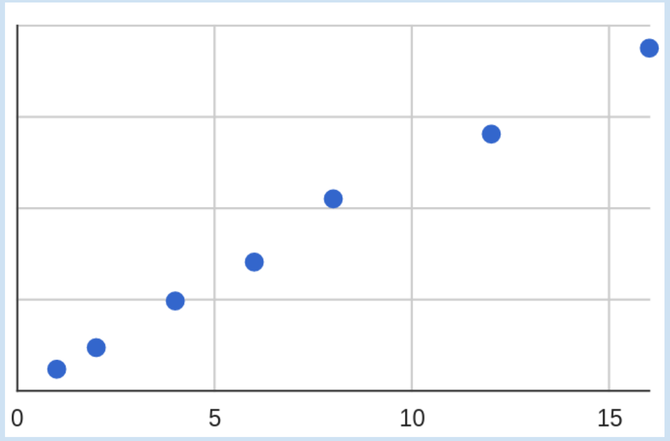
RAID across PD volumes is not needed to ensure reliability

RAID 0 striping PD volumes helps performance. To a point. For Now...

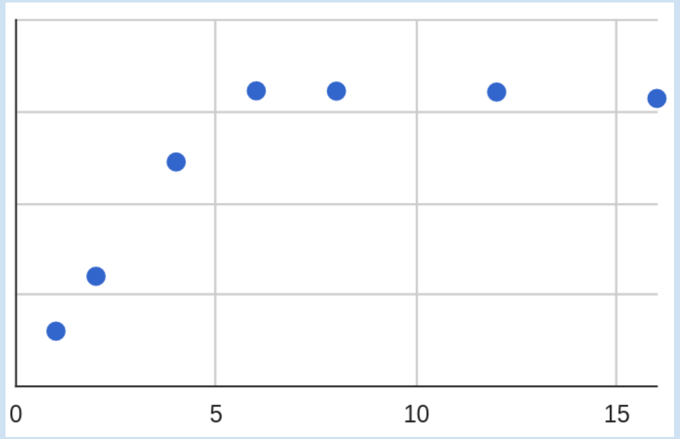
- Long term goal is to make performance independent of # of volumes to simplify planning

Database Type Workload

4KB Random Reads

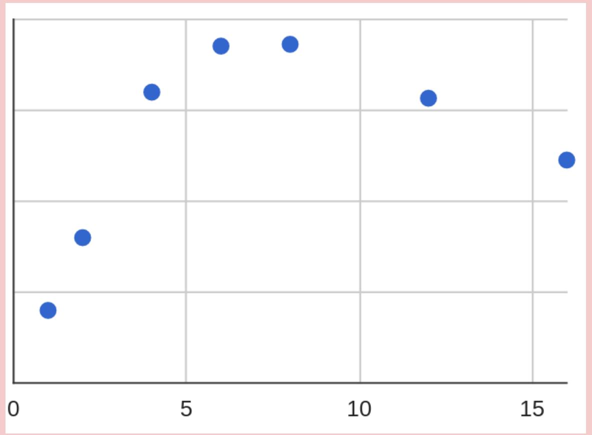


4KB Random Writes

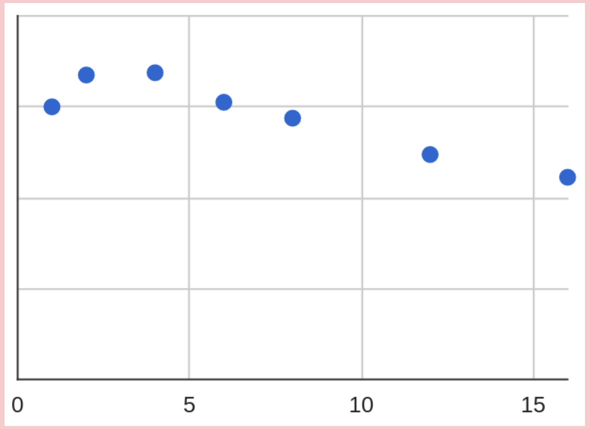


Bulk Data Type Workload

1MB Sequential Reads

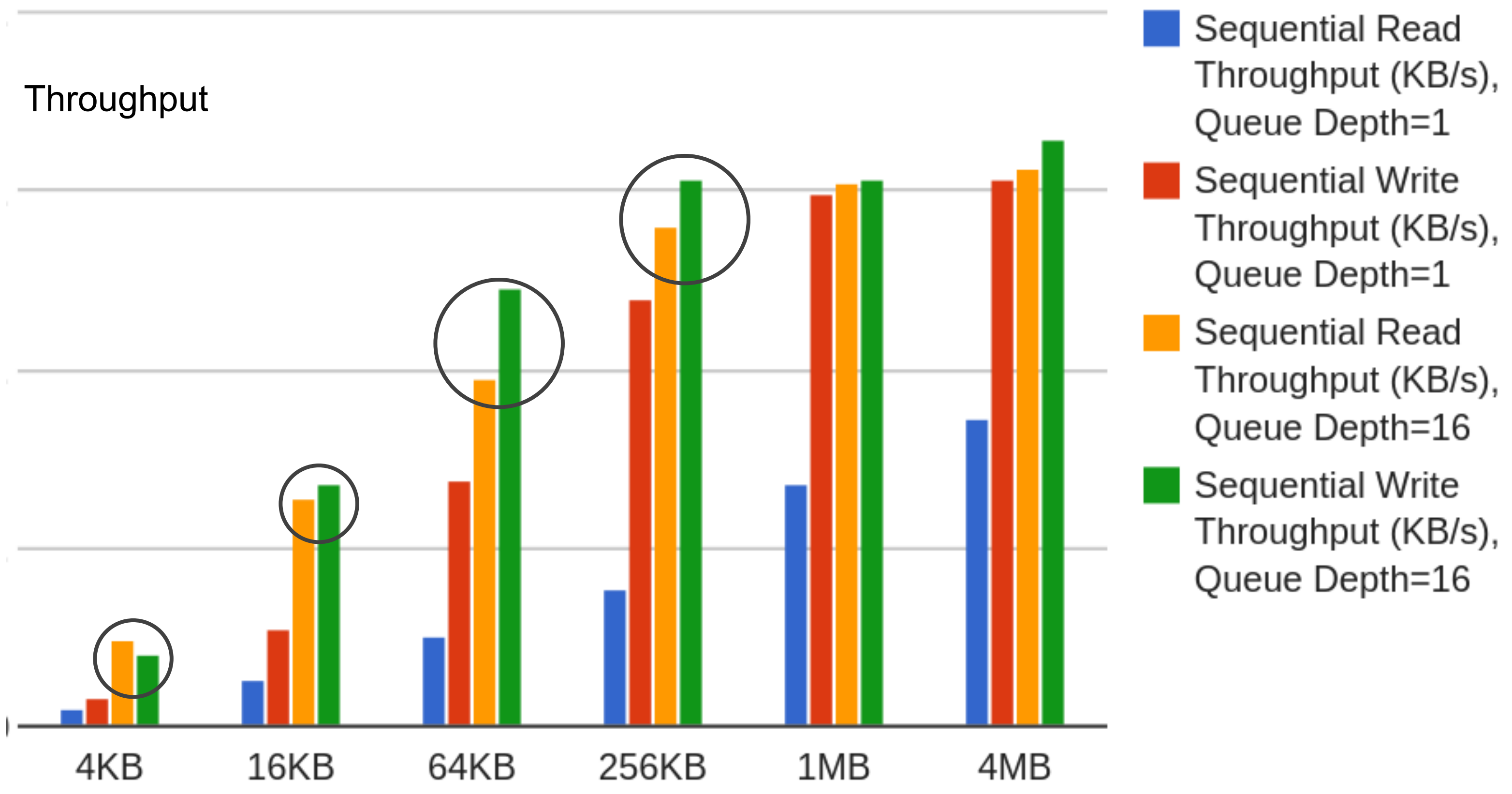


1MB Sequential Writes



Performance: IO Size & Queue Depth

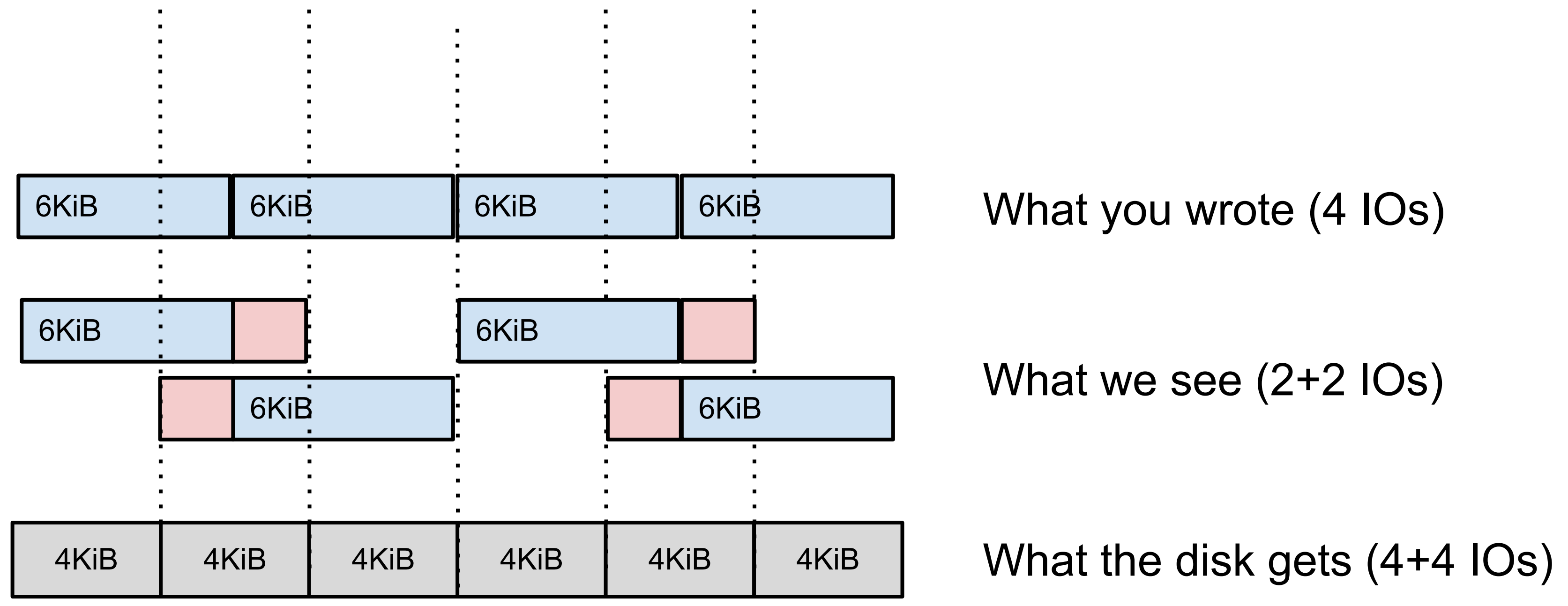
- Use multiples of 4KiB
- Increasing IO size up to 1-2 MiB generally improves performance
- Larger IO queue depth helps performance, especially at lower block sizes



Performance: IO Size & Alignment

Align IOs

- 4096 byte physical block size, 512 byte logical block size
- Misaligned writes are slow



Performance: Disk Partitioning

- Use separate PD volumes instead of partitioning the disk
- Use fdisk >= 12.17.2 to partition the disk

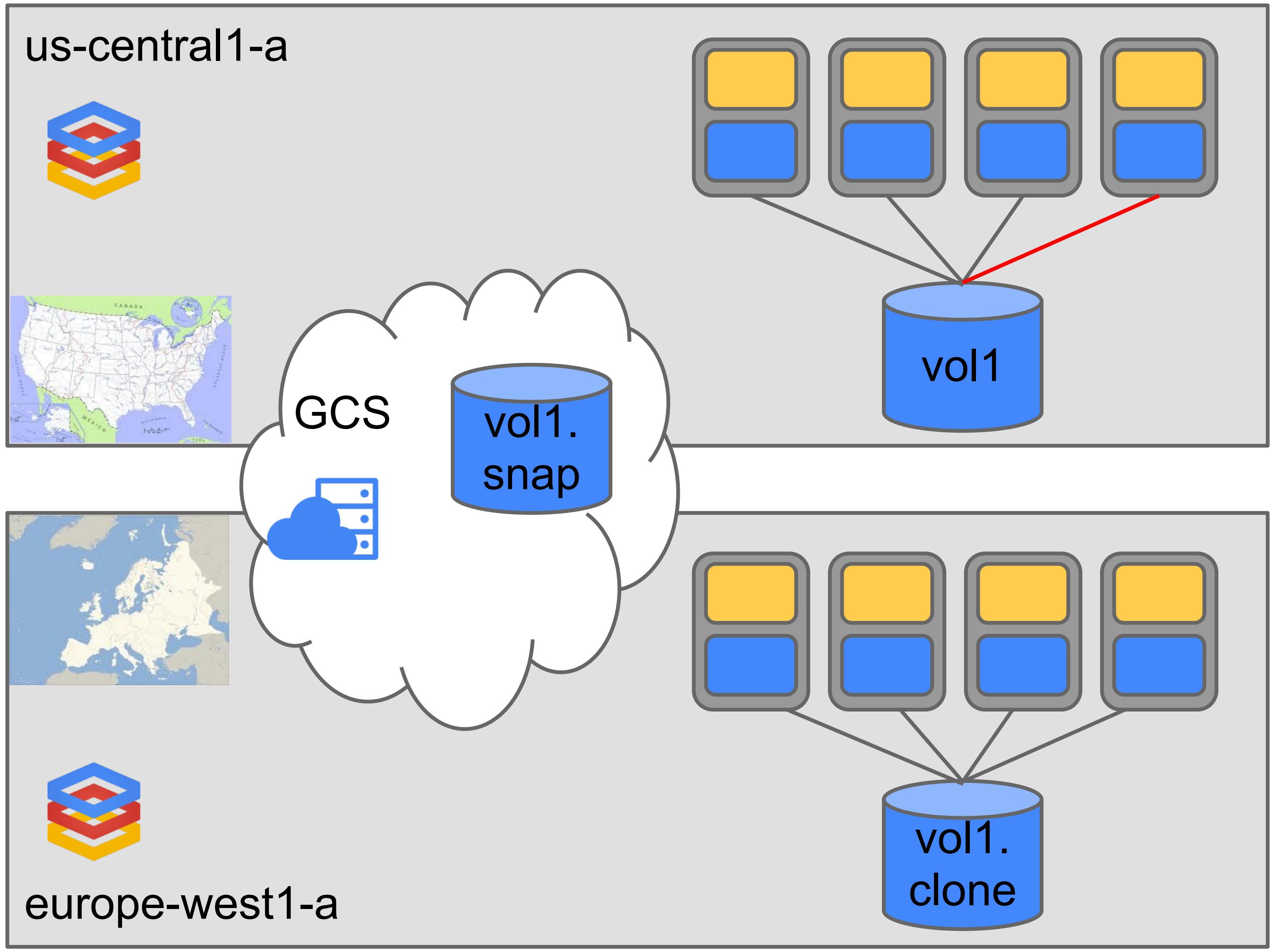
```
> fdisk -lu /dev/sda

Disk /dev/sda: 1000.2 GB, 1000204886016 bytes
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes

  Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *        2048      499711      248832   83  Linux
/dev/sda2          501758  1953523711   976510977    5  Extended
Partition 2 does not start on physical sector boundary.
```

Performance: Filesystem Formatting

- Align partitions on 1MB boundary
- Format disk with 4KB blocks
- ext4 and ext3 work really well
- See http://goo.gl/RxOA3 for instructions on having Google take care of this for you to avoid issues

Using PD & GCS To Distribute Data

- Distributing data to many VMs within a zone
- Distributing data geographically
- Use in conjunction!



Importing Data To PD

From on-prem SAN or NAS or block stores residing in other clouds

- Mount the storage to a physical or virtual machine
- scp or "gcutil push" the data to a GCE VM with a PD volume mounted*

From GCS or other cloud object stores:

- Use gsutil in a GCE VM to download the data to a filesystem on a mounted PD volume
- Some object stores have bittorrent support for faster download

* For accelerated copy, you may want to invest in something like Aspera

Summary Of Key Points

1. PD performance is optimized for building developer managed data services for cloud applications

2. PD scale will allow the deployment of thousands of GCE VM boot devices

3. PD + GCS is a phenomenal way to distribute static content globally

4. Follow Google best practices for:
   - Performance
   - Data movement
   - Avoiding downtime
   - Backup & DR

Questions?

Resources

Google Compute Engine: http://goo.gl/pG09E

Persistent Disk: http://goo.gl/pKMrF

Sign-up and try for yourself: http://goo.gl/Wqrcj

Google Cloud Blog: http://goo.gl/kKaGa

GCE discussion groups gce-discussion@googlegroups.com

Thank You!

Jay Judkowitz

judkowitz@google.com

@Jay_Judkowitz

Andrew Kadatch

akad@google.com

Google Developers