

Let's Talk Storage!

Thursday, 12:00-1:00pm

Anthony E. Nocentino

anocentino@purestorage.com

SQL Server & Azure SQL Conference

PRODUCED BY

Azure Data Community, Microsoft & NextGen

Get *Whova*

Official Event App

The event invitation code is: **DEVSpring22**

- Access **links** to join all sessions and keynotes virtually
- Explore the **professional profiles** of event speakers and attendees
- Send **in-app messages** and exchange contact info
- **Network and find attendees** with common affiliations, educations, shared networks, and social profiles
- Receive **update notifications** from organizers
- Access the **event agenda**, GPS guidance, maps, and parking directions at your fingertips

Download Whova and take your event mobile.

Get Whova from the App Store or Google Play.



Please sign up for the app with your social media account or email.

The event invitation code is:
DEVSpring22

You will be asked for an event invitation code after installing Whova.



Agenda

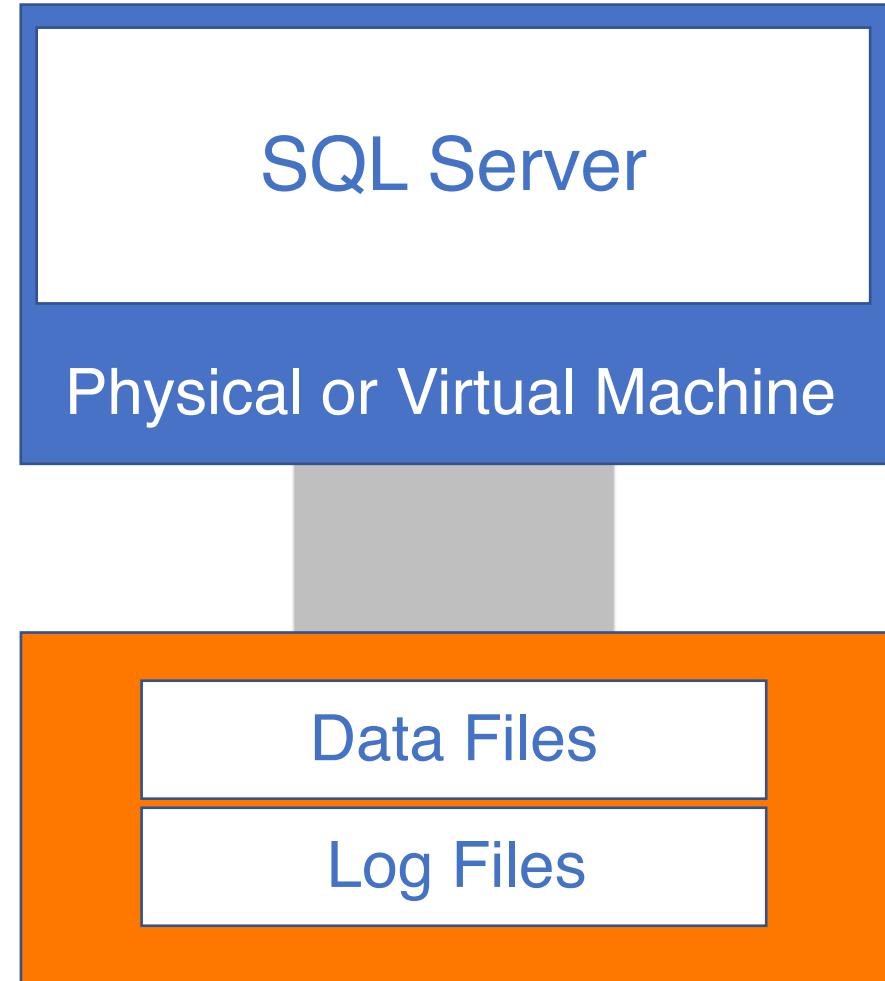
- Storage is Where Data Lives
- Protocols and Interconnects
- Key Performance Metrics and Why Database Care
- Database Files, Transaction Log Files, and File Systems
- A Closer Look at Latency
- Hypervisors
- Story Time

The Basics of Storage Subsystems

- Latency
- IO Block Size, aka Request Size
- IOPS
- Throughput, aka Bandwidth

Storage is where data lives...

- SQL Server stores data...
- On disks
 - In files
 - Local
 - SCSI
 - PCIe
 - **Remote**
 - Fibre Channel
 - iSCSI
 - NVMe-oF



Protocols, Interconnects and Storage Devices

- SCSI
- SATA
- Fibre Channel
- iSCSI
- NVMe
- NVMe-oF



Key Performance Metrics

Latency

How Long a Request Takes

Transaction Log I/Os

Saturation Leads to Queuing

Throughput

Amount of data moved

Function of the storage interconnect

Type of storage used

IOPs

Number of Requests

Size Depends on the Application

Why Database Systems Care About These...

Latency

Throughput

IOPs

Does your system sell things?

Does users need reports?

Larger Requests Take Longer

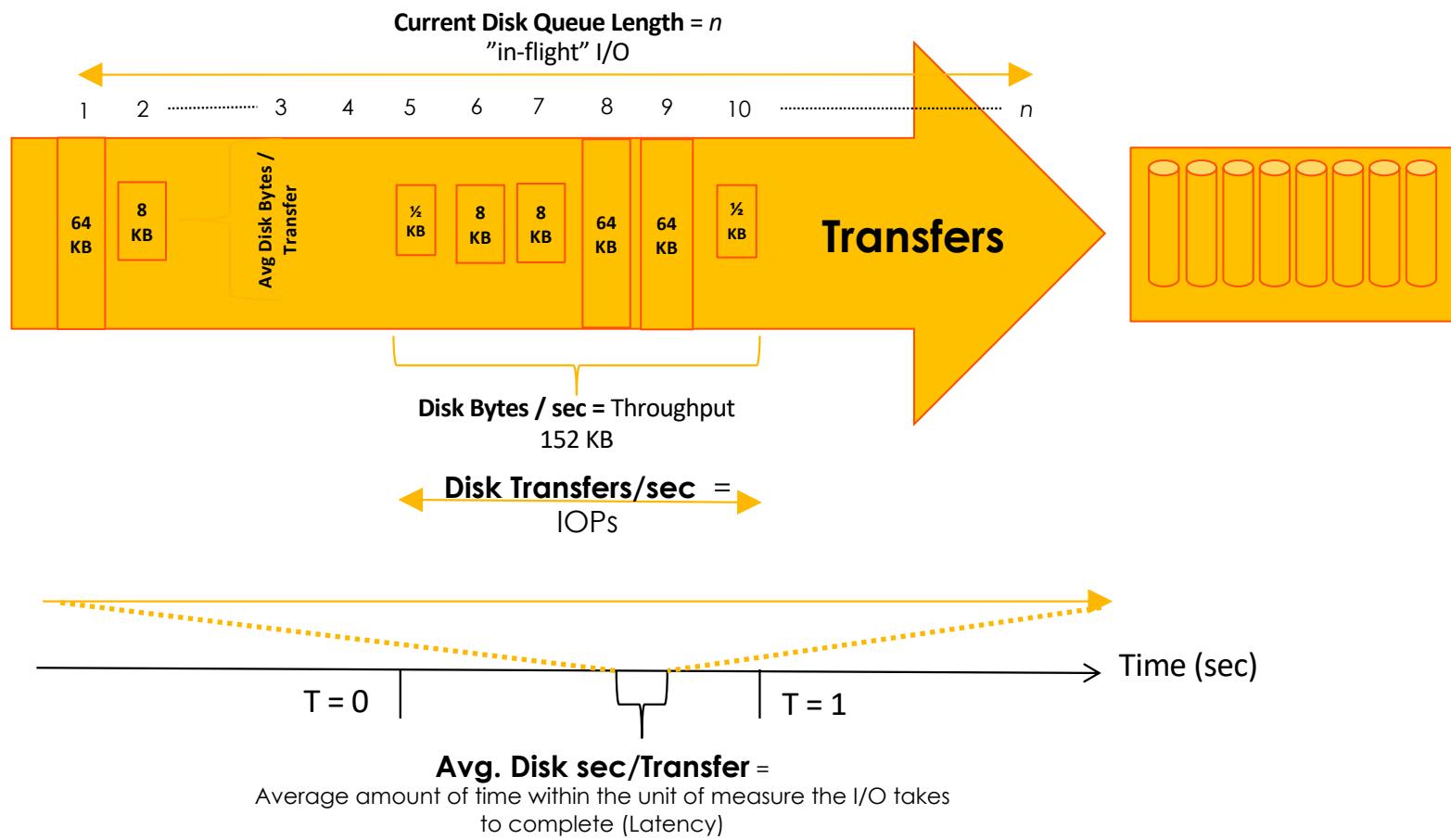
Is your system time sensitive?

Backups?

Often governed in Cloud and On Premises

OLTP

OLAP



Basics of SQL Server I/O

Operation	IO Block Size
Transaction log write	512 bytes – 60 KB
Checkpoint/Lazywriter	8KB – 1MB
Read-Ahead Scans	128KB – 512KB
Bulk Loads	256KB
Backup/Restore	1MB
ColumnStore Read-Ahead	8MB
File Initialization	8MB
In-Memory OLTP Checkpoint	1MB

<https://www.nocentino.com/posts/2021-12-10-sqlserver-io-size/>

Basics of SQL Server I/O

- Write Ahead Logging
- Checkpoint / Lazy Writer
- Bulk Operations
- Asynchronous I/O



Data Files, Transaction Log Files and Filesystems

- **Database files**

- Read and write, varying IO sizes and latencies
- Multiple files, potentially across multiple volumes

- **Transaction log files**

- Low latency, only one (sometimes)
- 2TB max

- **TempDB**

- Read and write, varying IO sizes and latencies
- Multiple files, local storage or remote?

But What about
64KB NTFS AUs?

Let's Look Closer...at Latency

- Monitor Latency
- **sys.dm_io_virtual_file_stats**
- Average per file since instance startup
- Have a monitoring tool...

<https://www.nocentino.com/posts/2021-10-06-sql-server-file-latency/>

Let's Look Closer...at Latency

```
SELECT
    DB_NAME(mf.database_id) AS [DBName],
    mf.name AS [FileName],
    mf.type_desc AS [FileType],
    vfs.num_of_reads AS [NumReads], --Number of reads issued on the file.
    vfs.num_of_writes AS [NumWrites], --Number of writes made on this file.
    vfs.num_of_bytes_read AS [ReadBytes], --Total number of bytes read on this file.
    vfs.num_of_bytes_written AS [WriteBytes], --Total number of bytes written to the file.

    --Calculate the percentage of bytes read or written to the file
    vfs.num_of_bytes_read * 100 / (( vfs.num_of_bytes_read + vfs.num_of_bytes_written )) AS [PercentBytesRead],
    vfs.num_of_bytes_written * 100 / (( vfs.num_of_bytes_read + vfs.num_of_bytes_written )) AS [PercentBytesWrite],

    --Calculate the average read latency and the average read IO size
    CASE WHEN vfs.num_of_reads = 0 THEN 0 ELSE vfs.io_stall_read_ms / vfs.num_of_reads END AS [AvgReadLatency_(ms)],
    CASE WHEN vfs.num_of_reads = 0 THEN 0 ELSE ( vfs.num_of_bytes_read / vfs.num_of_reads ) / 1024 END AS [AvgReadSize_(KB)],

    --Calculate the average write latency and the average write IO size
    CASE WHEN vfs.num_of_writes = 0 THEN 0 ELSE vfs.io_stall_write_ms / vfs.num_of_writes END AS [AvgWriteLatency_(ms)],
    CASE WHEN vfs.num_of_writes = 0 THEN 0 ELSE ( vfs.num_of_bytes_written / vfs.num_of_writes ) / 1024 END AS [AvgWriteSize_(KB)],

    --Calculate the average total latency and the average IO size
    CASE WHEN vfs.num_of_reads + vfs.num_of_writes = 0 THEN 0 ELSE vfs.io_stall / ( vfs.num_of_reads + vfs.num_of_writes ) END AS [AvgLatency_(ms)],
    CASE WHEN vfs.num_of_reads + vfs.num_of_writes = 0 THEN 0
    ELSE ( vfs.num_of_bytes_read + vfs.num_of_bytes_written ) / ( vfs.num_of_reads + vfs.num_of_writes ) / 1024 END AS [AvgIOSize_(KB)],

    --The physical file name
    mf.physical_name AS [PhysicalFileName]

FROM
    sys.dm_io_virtual_file_stats(NULL, NULL) as [vfs]
    inner join sys.master_files as [mf] ON [vfs].[database_id] = [mf].[database_id]
    AND [vfs].[file_id] = [mf].[file_id]
ORDER BY
    [AvgLatency_(ms)] DESC
--  [AvgReadLatency_(ms)]
--  [AvgWriteLatency_(ms)]
```

<https://www.nocentino.com/posts/2021-10-06-sql-server-file-latency/>



Let's Look Closer...at Latency



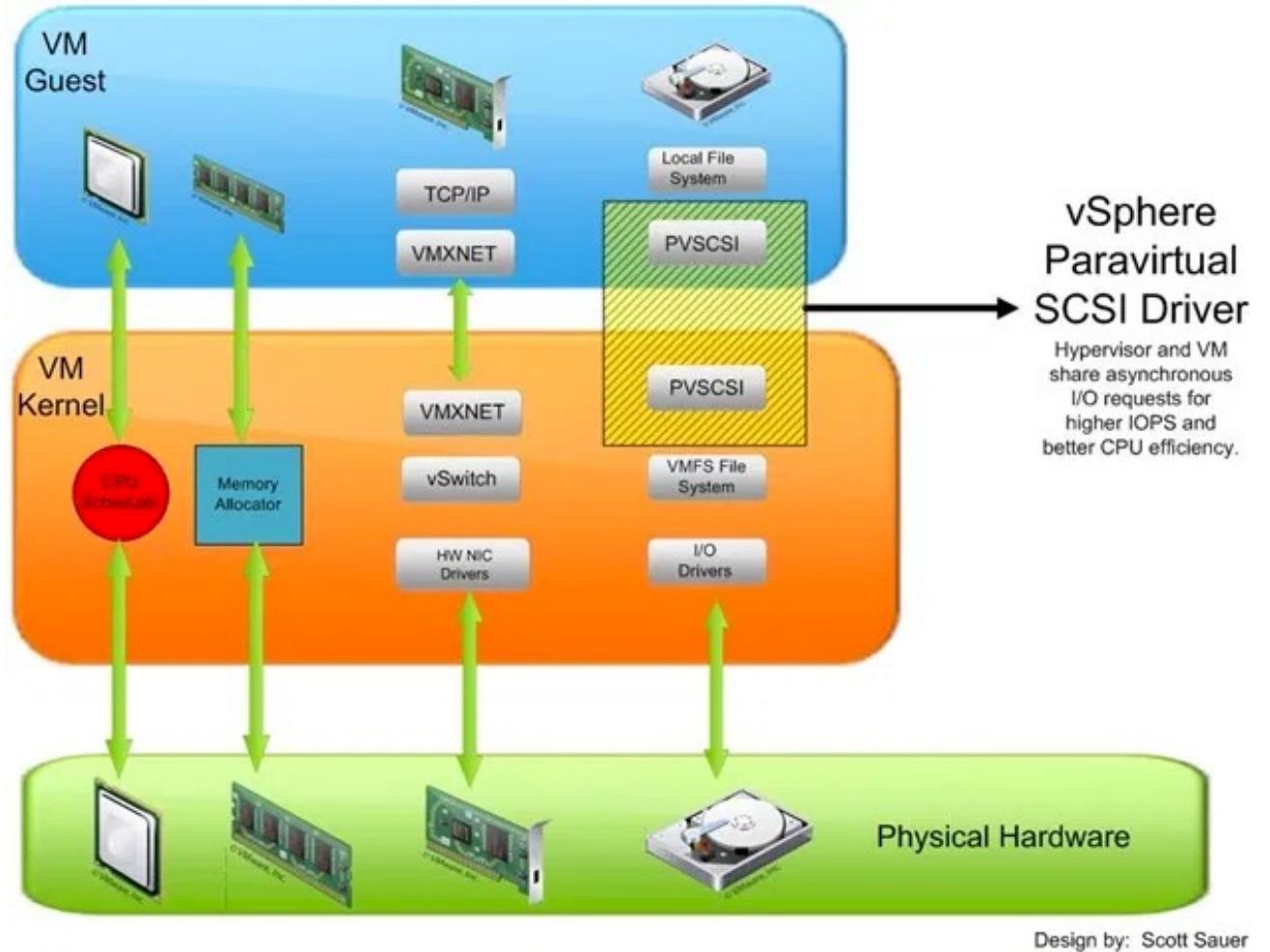
DBName	FileName	FileType	NumReads	NumWrites	ReadBytes	WriteBytes	PercentBytesRead	PercentBytesWrite	AvgReadLatency_(ms)	AvgReadSize_(KB)	AvgWriteLatency_(ms)	AvgWriteSize_(KB)	AvgLatency_(ms)	AvgIOSize_(KB)	PhysicalFileName
tempdb	temp3	ROWS	23	53261	1277952	530538496	0	99	7	54	41	9	41	9	T:\TEMPDB\tempdb.mdf
tempdb	temp8	ROWS	43	53981	2129920	546938880	0	99	1	48	41	9	41	9	T:\TEMPDB\tempdb.mdf
tempdb	tempdev	ROWS	503	53694	19636224	535355392	3	96	4	38	39	9	39	10	T:\TEMPDB\tempdb.mdf
tempdb	temp4	ROWS	26	53329	1417216	547823616	0	99	0	53	38	10	38	10	T:\TEMPDB\tempdb.mdf
tempdb	temp5	ROWS	33	53974	1761280	536707072	0	99	0	52	38	9	38	9	T:\TEMPDB\tempdb.mdf
tempdb	temp2	ROWS	25	53815	1351680	546734080	0	99	0	52	38	9	38	9	T:\TEMPDB\tempdb.mdf
tempdb	temp7	ROWS	32	53307	1695744	531218432	0	99	0	51	35	9	35	9	T:\TEMPDB\tempdb.mdf
tempdb	temp6	ROWS	531	53372	7135232	540041216	1	98	2	13	31	9	30	9	T:\TEMPDB\tempdb.mdf
FT_Demo	FT_Demo	ROWS	564	2112	23461888	17309696	57	42	1	40	14	8	11	14	X:\FT_Demo.mdf
model	modellog	LOG	9	17	1015808	40960	96	3	17	110	0	2	6	39	S:\SYSTEM\MSSQL15.MDF
tpch100	tpch100	ROWS	33850	1759903	971161600	999181459456	0	99	1	28	5	554	5	544	X:\TPCH\tpch100.mdf
TestDB1_LS	TestDB1_log	LOG	7	0	1007616	0	100	0	4	140	0	0	4	140	S:\SYSTEM\MSSQL15.MDF
TestDB1	TestDB1_log	LOG	7	5	1007616	11264	98	1	4	140	0	2	2	82	L:\LOG\TestDB1_log.ldf
tpcc100	tpcc_log	LOG	11	5	1024000	14848	98	1	3	90	0	2	2	63	L:\LOG\tpcc_log100.ldf
TestDB1	TestDB1	ROWS	334	1	13926400	8192	99	0	1	40	0	8	1	40	D:\DATA\TestDB1.mdf
msdb	MSDBData	ROWS	7122	40371	233316352	423354368	35	64	1	31	1	10	1	13	S:\SYSTEM\MSSQL15.MDF
msdb	MSDBLog	LOG	52	191943	350720	279290368	0	99	2	6	1	1	1	1	S:\SYSTEM\MSSQL15.MDF
model	modeldev	ROWS	78	4	35872768	32768	99	0	1	449	0	8	1	427	S:\SYSTEM\MSSQL15.MDF
TestDB1_LS	TestDB1	ROWS	29	0	1671168	0	100	0	1	56	0	0	1	56	S:\SYSTEM\MSSQL15.MDF
tpcc100	tpcc	ROWS	665	1	37715968	8192	99	0	1	55	0	8	1	55	D:\DATA\tpcc100.mdf
tpch100	tpch100_log	LOG	15	47682006	1040384	2010226025984	0	99	0	67	1	41	1	41	X:\TPCH\tpch100_log.ldf
tempdb	templog	LOG	44	85605	1294336	5233082368	0	99	4	28	1	59	1	59	T:\TEMPDB\templog.ldf

<https://www.nocentino.com/posts/2021-10-06-sql-server-file-latency/>



Hypervisors

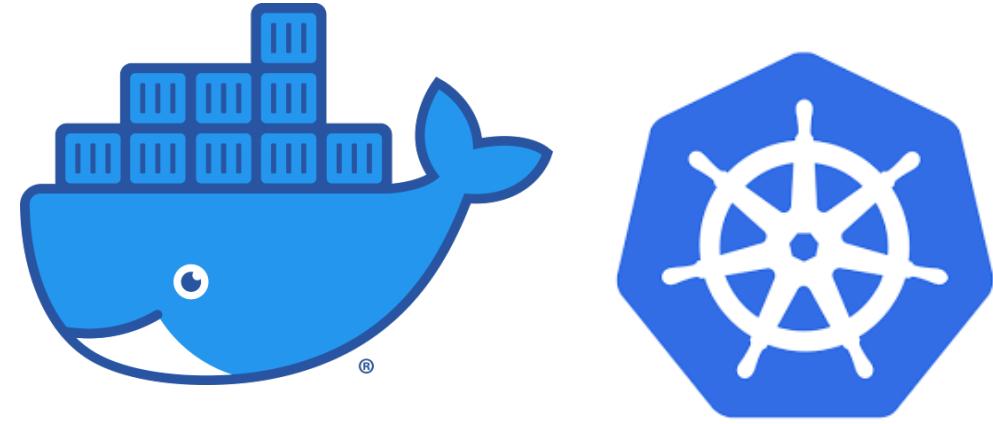
- VMware
- Hyper-V
- Red Hat
- Others



<https://www.nocentino.com/posts/2021-09-27-sqlserver-vms-best-practices/>

What About Containers?

- Docker
 - Kubernetes
 - Azure Arc enabled Data Services
-
- Node Level Storage
 - Local or Remote storage
 - Cloud or On-Prem

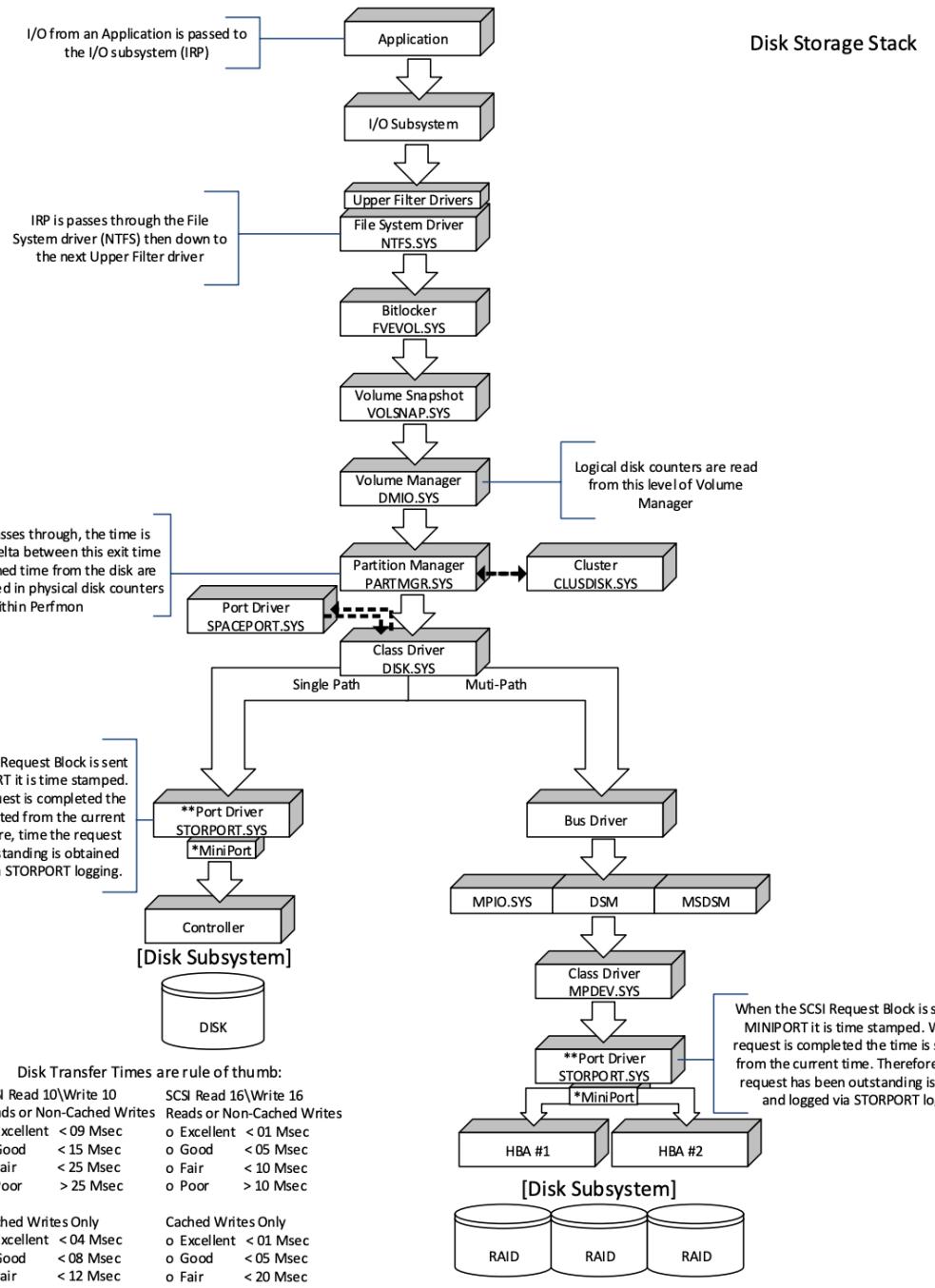


File Systems

- NTFS
- ReFS
- SMB
- Linux: EXT4, XFS and NFS (4.2+)



Think This is Complex?



Devices

The List is Long



Block Replication vs. Logical Replication

- Transaction Replication
- Log Shipping
- Availability Groups
- Storage Replication



Story Time...

Line of Business Application

App Vendor

Web Servers

37 Second Page Load Times

DBAs

SQL Servers

High File and Disk Latency

VM Admins

Hypervisor

High Disk Latency

Network Team

Storage Interconnect

No Meaningful Metrics

Storage Team

Storage

Super Low Latencies

Review

- Storage is Where Data Lives
- Protocols and Interconnects
- Key Performance Metrics and Why Database Care
- Database Files, Transaction Log Files, and File Systems
- A Closer Look at Latency
- Hypervisors
- Story Time

Need more data?

- **Contact me!**

- **email:** anocentino@purestorage.com
- **Twitter:** @nocentino
- **GitHub:**
<https://github.com/nocentino/Presentations>
- **Blog:** www.nocentino.com

- **Pluralsight** - Hit me up to get free access

- Linux
- Kubernetes
- Azure

Questions?

Don't forget to complete an online

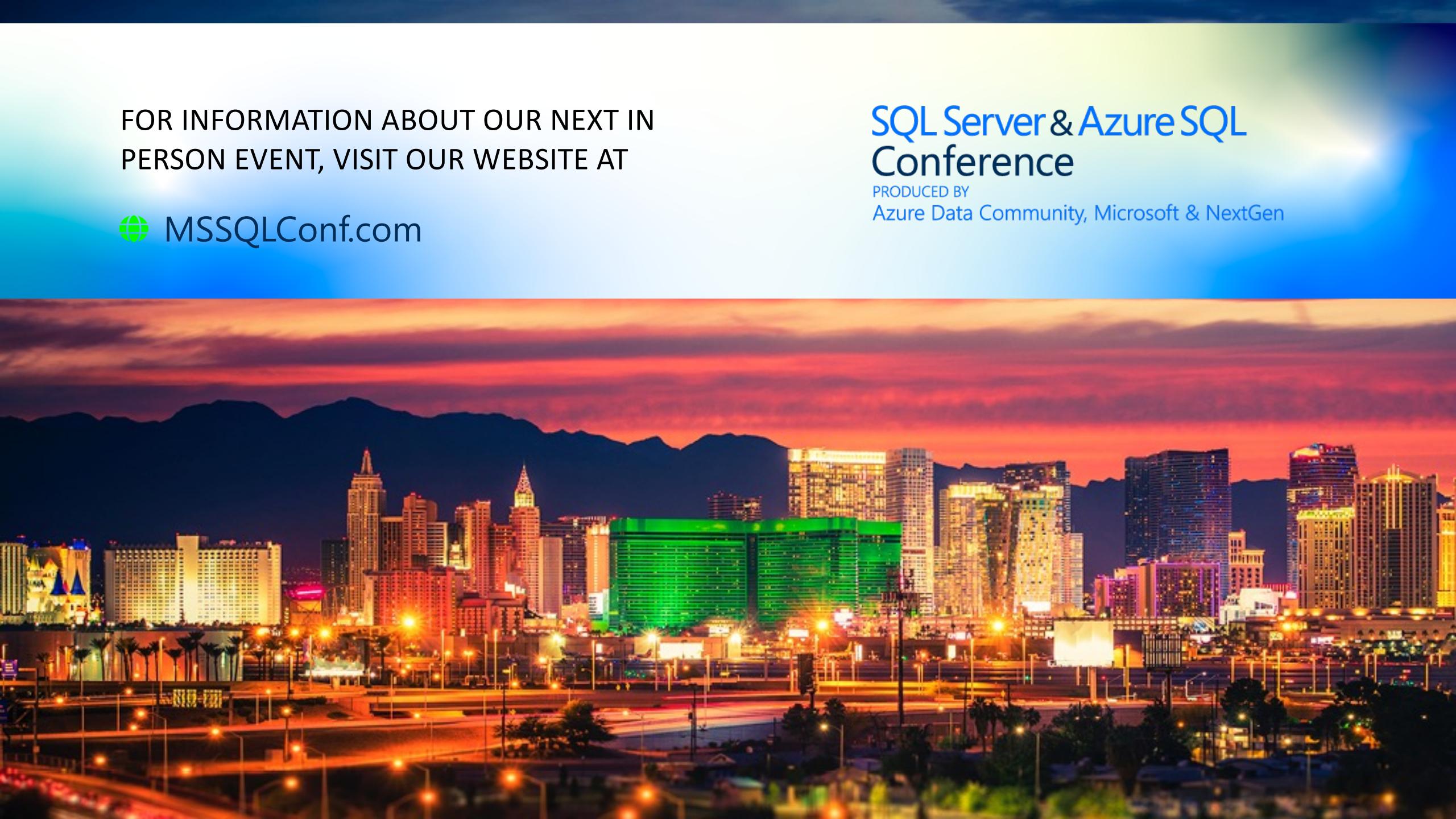
Let's Talk Storage

Your evaluation helps organizers build better conferences
and helps speakers improve their sessions.

**SQL Server & Azure SQL
Conference**

PRODUCED BY

Azure Data Community, Microsoft & NextGen



FOR INFORMATION ABOUT OUR NEXT IN PERSON EVENT, VISIT OUR WEBSITE AT

 MSSQLConf.com

SQL Server & Azure SQL Conference

PRODUCED BY

Azure Data Community, Microsoft & NextGen