
Microsoft
DevDays ASIA 2020
online
亞太技術年會

Microsoft

DevDays ASIA 2020

online

亞太技術年會

Azure SQL Database – Hyperscale Tier

Balmukund Lakhani

Program Manager – Azure SQL Database

Agenda

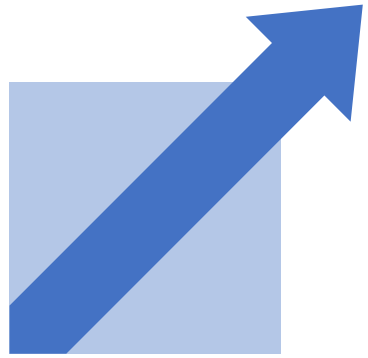
- Hyperscale Architecture.
- Migration to Hyperscale.
- Performance & Best Practices.

Challenges with managing VLDBs

Size of data

- Operations take a LONG time (days in some cases)
- Ongoing operations degrade database performance.
- Higher probability of failure for longer operations.
- Provisioning more storage to expand the database can be painful.

Scaling Compute



- Logistics of moving to larger box.
- Economics of sizing for max peaks.

Imagine a world where..

Limitless database

A cloud database supports 100TB and more

Size of data

There are few to no size-of-data operations

Backup

Backups have zero performance impact

Restore

Restores are extremely fast and constant time

Compute

Scale up/down happens in minutes, not hours for both storage and compute

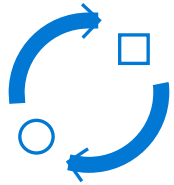
What is Azure SQL Database Hyperscale?

Storage

Scalable new storage architecture

Cloud native

Architected for cloud

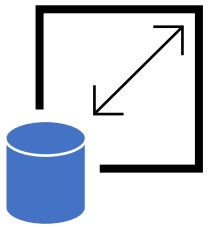


Seamless compatibility

Fully compatible with Azure SQL Database

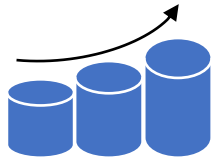
Performance

VLDB operations without VLDB headaches



No limits

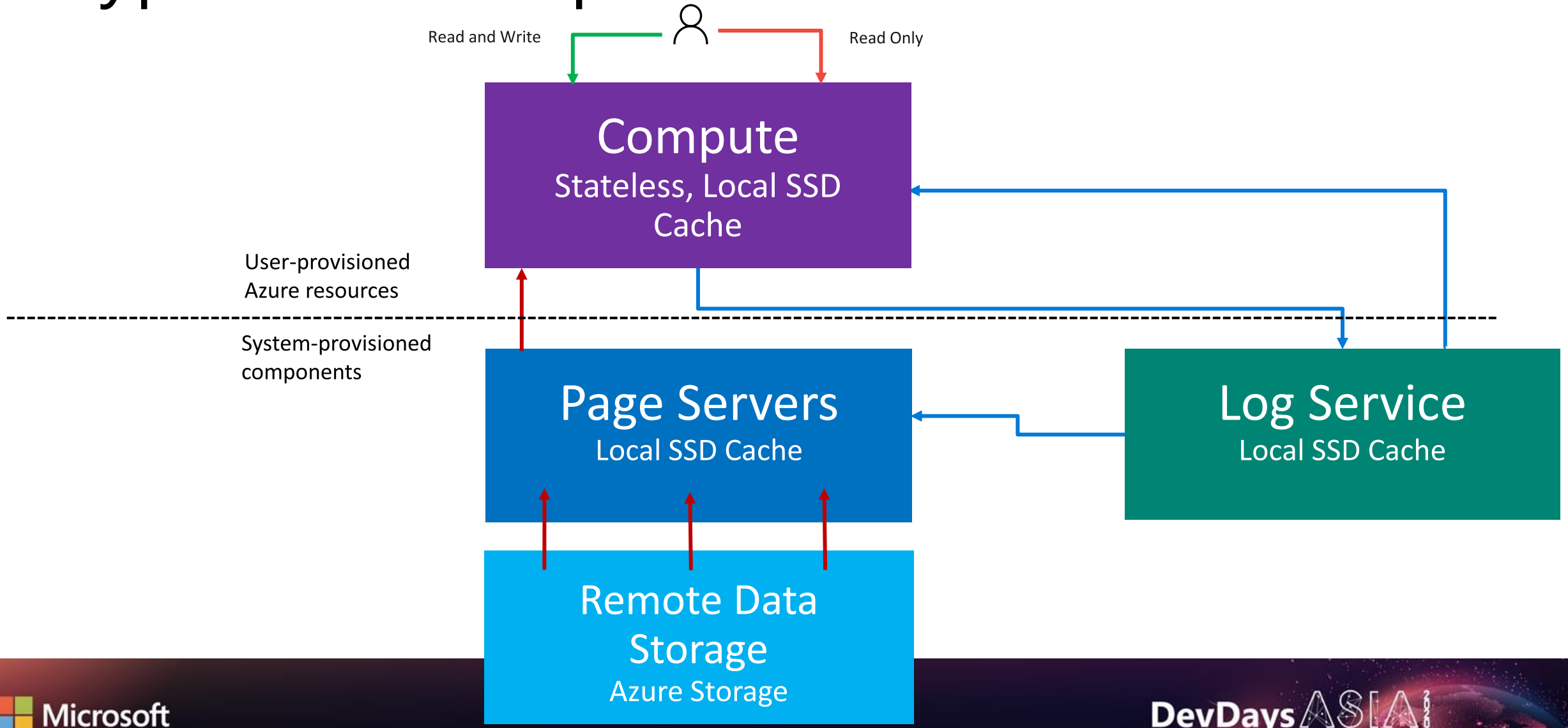
Scale compute and storage

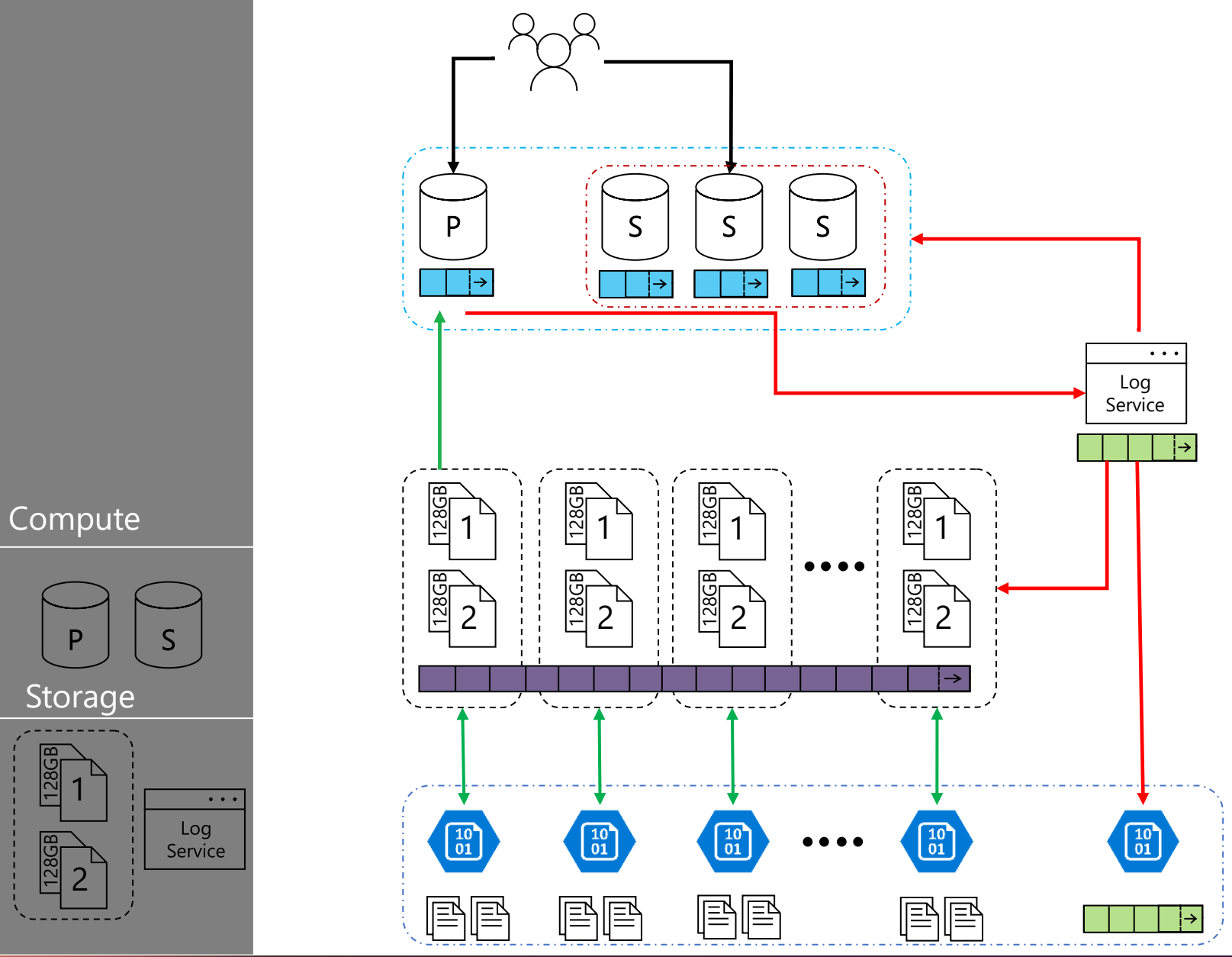


Large database

Support for 100TB+

Hyperscale components

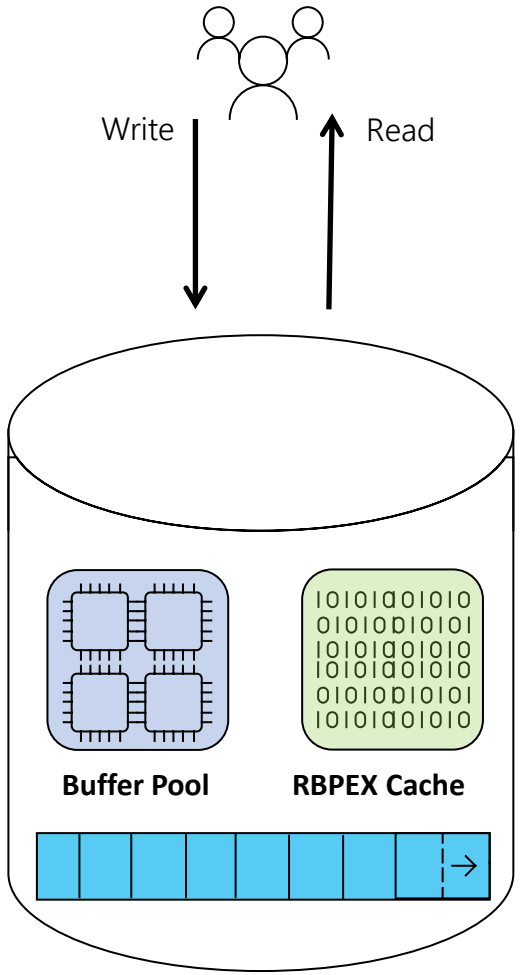




Distributed system

- Separate compute and storage
- Externalized log service (Xlog)
- Data file = page server
- Tiered storage
  - Redundancy at all tiers
  - Storage can be encrypted at rest
  - 100 TB supported limit

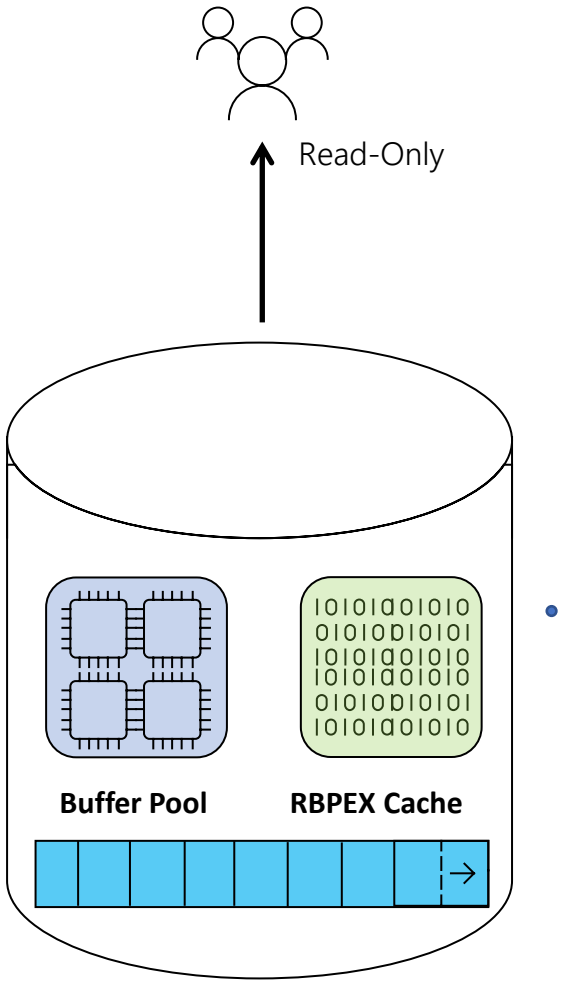
Primary



- Configurable vCores 2 – 80
- Memory and directly attached SSD proportional to number of cores
  - Memory dependent on hardware generation
  - **Non-covering** RBPEX
  - 1-2 ms data access latency
- Read-Write Workloads
- Seamless end user interactions

Compute

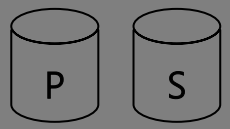
Readable Secondary



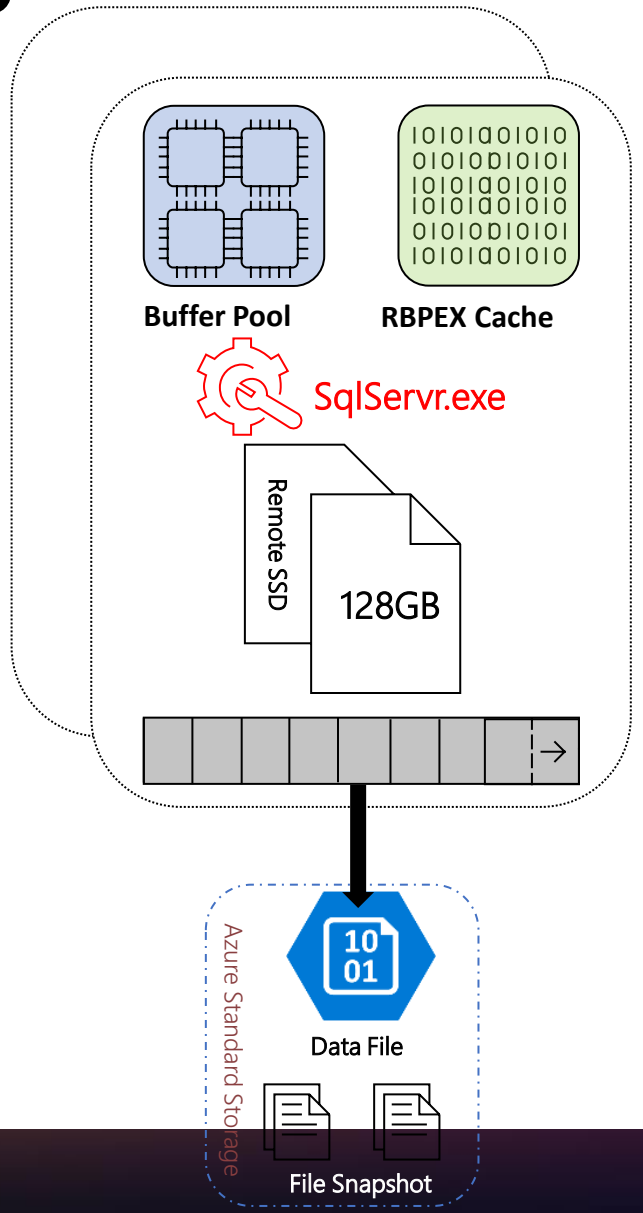
- Currently same size as the Primary
- Up to 4 readable secondary replicas
- Read-only workloads
  - ApplicationIntent = READONLY
  - Round robin distribution
- Hot failover target

Higher SLA with > 0 replicas

Compute



Page Server



- "Remote" SSD - 128GB per Page Server
- **Covering** RBPEx
- Each page server has its data file on Azure Standard storage
- Offload operations from compute
  - Checkpoints executed on Page Servers
  - File snapshots for backup operations are isolated from compute
- Built-in high availability for page servers and storage
- Allocations are 10 GB increments

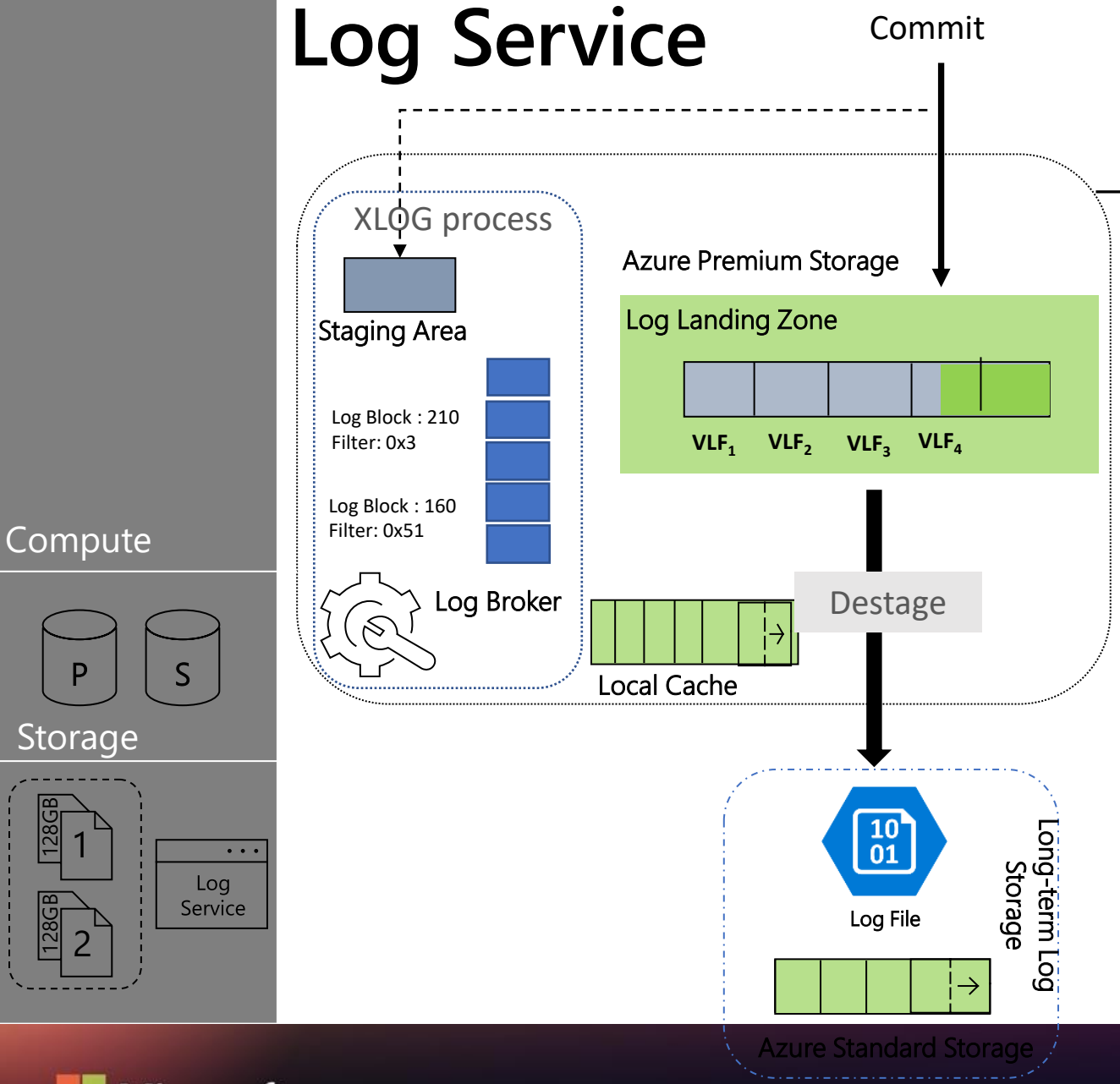
Compute

P S

Storage

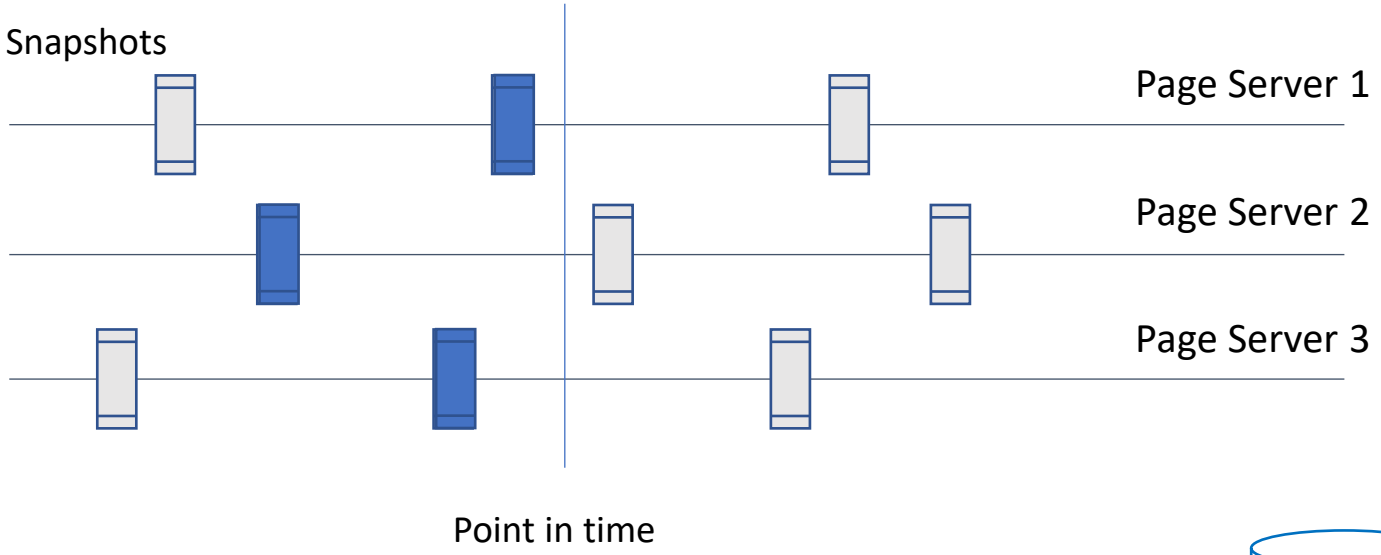
128GB

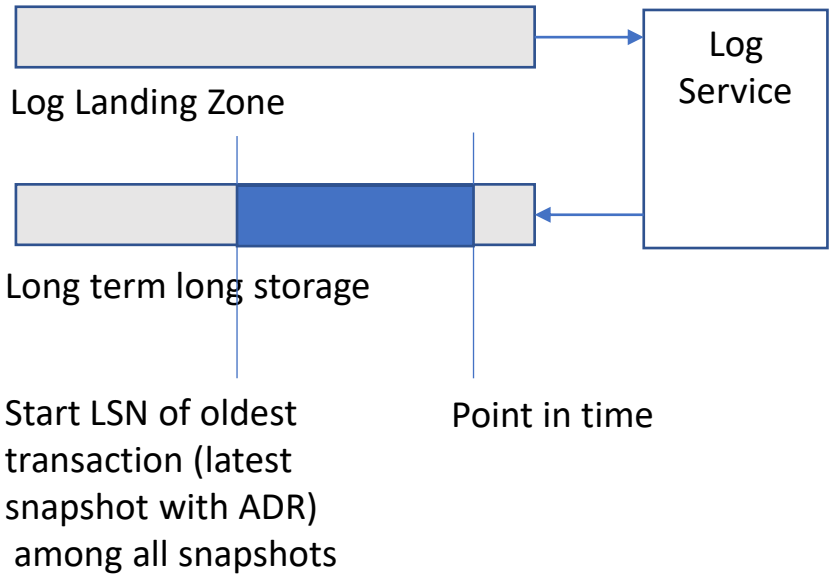
Log Service



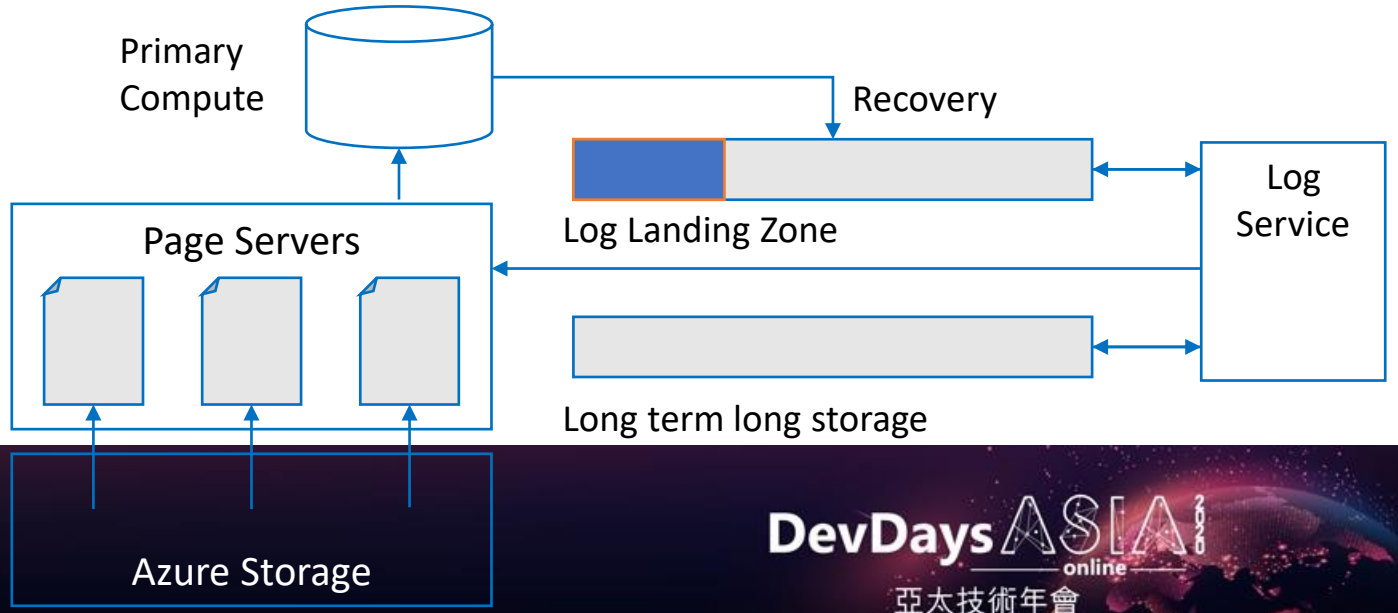
- Provides "infinite log" abstraction.
- Log Service consists of 3 components
  - Xlog process
  - Landing Zone
  - Long-Term Log Storage
- Landing Zone (Premium storage, 1 TB, P30)
  - Durable commit point, ~2-4 ms latency
  - Broken into VLFs which contain Log Blocks
- Xlog Process
  - Filtration of log blocks to page servers/secondaries.
  - Broker services pull requests for log blocks.
  - Responsible for de-staging to long-term storage.
- Long Term Storage :
  - Availability and Resiliency built-in

Backup & Restore





- Backup has no impact to apps
- Parallelized copy (metadata)
- Constant time PITR
- Accelerated database recovery



Performance and scale best practices

- SQL Server database engine, common SQL best practices apply
- Max log generation rate: 100 MB/s irrespective of compute size
  - May require sufficient compute on primary to generate 100 MB/s
- Scale up/down/add replica latency: ~60-90 seconds irrespective of data size
- Tempdb, RBPEX: finite resources on local SSD
  - Size is proportional to compute size (number of cores)
- Use memory optimized table variables to alleviate tempdb contention
- Use resumable indexes to create/rebuild indexes on very large tables
- Use MAXDOP hint for index create/rebuild offline
  - Database-scoped MAXDOP used for the rest of workload may or may not be optimal
  - Up to MAXDOP 16
- Secondary replicas are asynchronous
  - If workload cannot tolerate any data latency, read on the primary
  - Multiple readable replicas could be at different points of redo, thus have varying data latency
- Some operations are size-of-data (e.g. cross-region copy, geo-restore)

Migration to Hyperscale

- Azure Data Migration Service

- Bulk load/Bulk copy
- Using Azure Data Factory
- Using Spark connector for SQL
- Using a .Net app. Sample: smart bulkcopy
- Bulk load into heaps for larger tables, create indexes later
- If using clustered columnstore, bulk load directly into the table with CCI

- Transactional replication
  - Use concurrent snapshots
  - Requires primary keys


Microsoft
DevDays ASIA 2020
online
亞太技術年會
Thank you