Socrates:
The New SQL Server in the Cloud

Spencer Schoenberg

Goals

- Highly available
- Elastic
  - Shared-disk
- High performance
  - Low log latency
- Large-scale databases
  - Eliminate O(database size) operations
- Backwards compatibility

Existing Solutions (HADR, shared-disk)

- Azure SQL DB
- Google Spanner
- Amazon Aurora
- Oracle

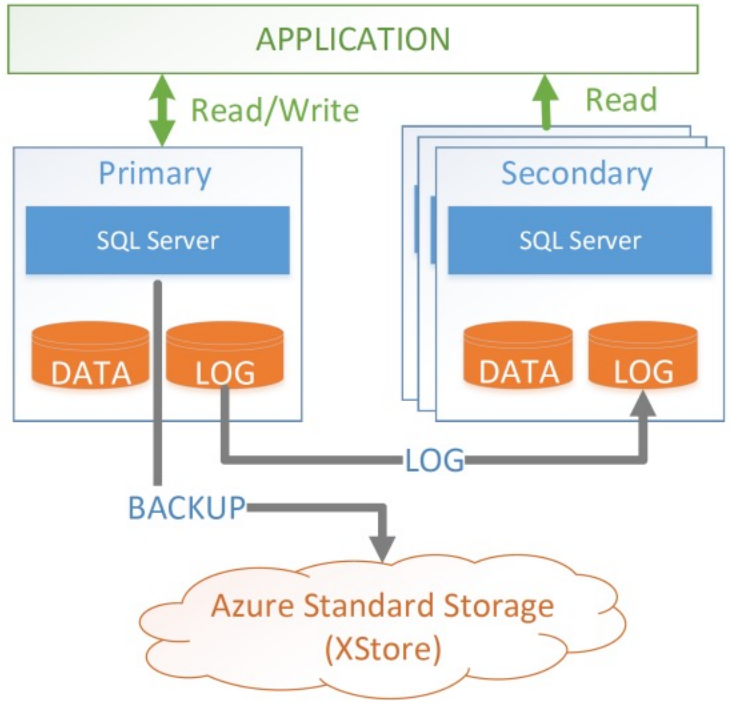

Figure 1: HADR Arch. (Replicated State Machines)

Architecture

- Compute Tier
  - Compute Nodes
- Logging Tier
  - XLOG Service
- Storage Tier
  - Page Servers
- Persistence Tier
  - XStore

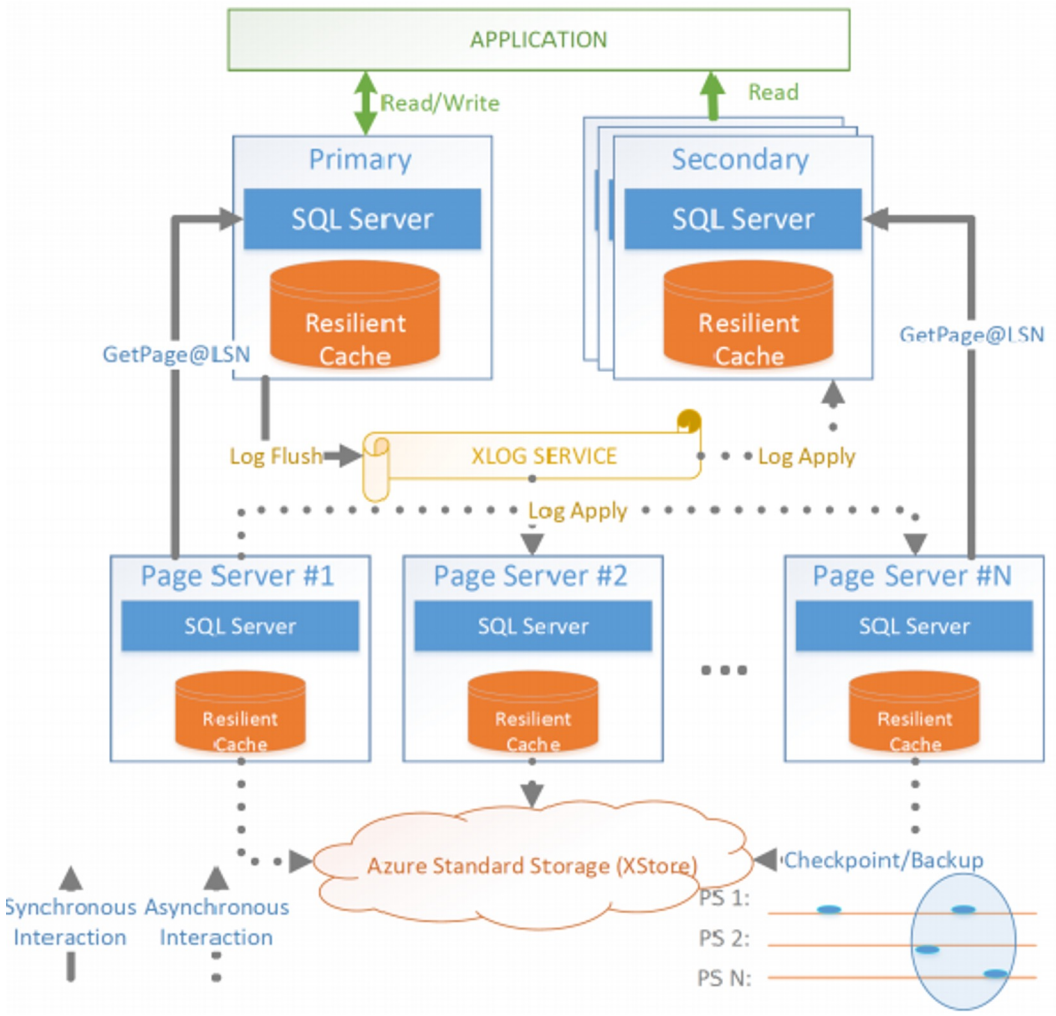

Figure 2: Socrates Architecture

Compute Tier

- Stateless
- Single primary
  - Read + Write
- Multiple secondaries
  - Read
  - Can promote from failover
- Unaware of network separation
  - All IO is transparently virtualized
  - Unaware of other replicas

Compute Tier - Primary

- Pushdown
  - Backups
  - Log durability
- Resilient Buffer Pool Extension (RBPEX)
  - Cache spills into local SSD
  - Resilient
  - Only need to apply log updates after reboot
- `getPage(pageId, LSN)`
  - Page Server returns page at least as new as Log Sequence Number
  - Keep track of newest LSN per page when writing
  - Read back if evicted from cache

Compute Tier - Secondaries

- Log blocks are not persisted
- Only store hot data pages
- New log records (from XLOG) applied to cached pages

XLOG (Logging Tier)

- Primary to Landing Zone (LZ)
  - Synchronous
  - 3 replicas
  - Small storage
- Primary to XLOG
  - Asynchronous
- Pending Area to LogBroker
  - Only after in LZ can logs progress to replicas and archive
- Destaging
  - Written to XStore
  - Written to XLOG local cache
  - Removal from LZ
- Replicas + Pager Servers from LogBroker
  - Pulled from LogBroker following memory hierarchy

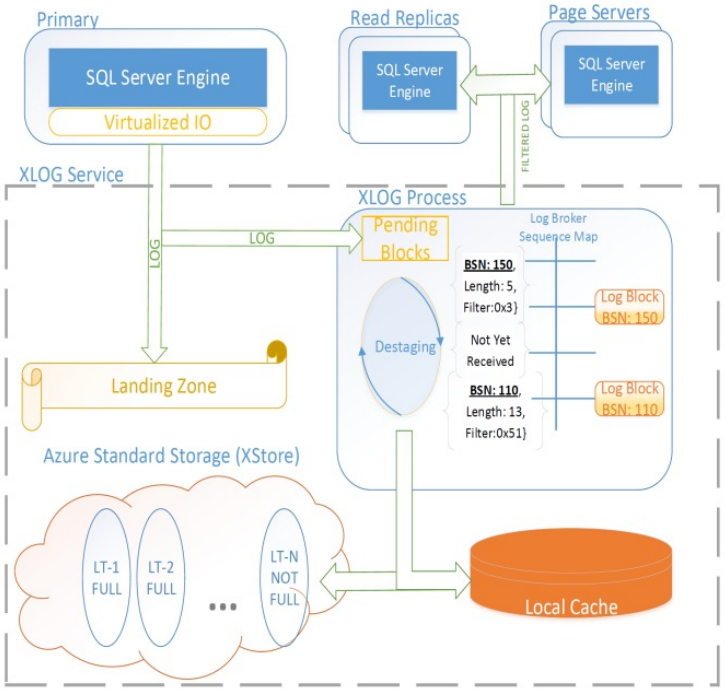

Figure 3: XLOG Service

Page Servers (Storage Tier)

- Stateless
- New log records (from XLOG) applied to a partition of the database
- All data stored in RBPEX cache
- Dense cache organization
  - Stride-preserving cache layout
  - Avoids read amplification
- Can asynchronously seed a new Page Server

XStore (Persistence Tier)

- Durability
  - Replicated to 6 nodes across 3 AZs
- Snapshots
  - Full copy sent per week
  - Daily delta
  - Log backup per 5 minutes
- Supports Backup and Recovery
  - Point-in-time restore (requires snapshot and small-ish delta log)

Performance - Throughput

- 5% less CPU utilization vs HADR
  - On CDB default mix
  - More time in Ios
- In Update-heavy CDB
  - Log becomes bottleneck
  - Still higher as backups to XStore are handled downstream

	CPU %	Write TPS	Read TPS	Total TPS
HADR	99.1	347	1055	1402
Socrates	96.4	330	1005	1335

Table 2: CDB Throughput: HADR vs. Socrates (1TB)

	SF	Log MB/s	CPU %
HADR	30000	56.9	46.2
Socrates	30000	89.8	73.2

Table 5: CDB Log Throughput: HADR vs. Socrates

Performance - Caching

- HADR 100% hit-rate, but limits database size
- Still large cache hit rate despite
  - Cache only 15%, 1% database size

Data Size	Scale Factor	Memory Size	RBPEX Size	Local cache hit %
1TB	20000	56GB	168GB	52

Table 3: Socrates Cache Hit Rate (CDB)

Data Size	Customers	Memory Size	RBPEX Size	Local cache hit %
30TB	3.1M	88GB	320GB	32

Table 4: Socrates Cache Hit Rate (TPC-E)

Questions?