Introduction to Sterling

Last updated by | Vitor Tomaz | Aug 5, 2020 at 12:34 PM PDT

Contents

Introduction

Sterling is the evolution of WA SQL DB architecture targeted primarily to improve availability, reliability, performance, predictability and compatibility to provide a premium experience and premium SLA to our top tier customers.

Secondary objectives are the provision of a stateless query service that supports HDInsight and Azure DB data sources, and the reduction of COGs (Cost of Goods Sold) for cold databases.

There are common customer request we will integrate into the product like: Automatic Management, On-Demand capacity, lower entry point price, scale-up or scale-out option, greater T-SQL compatibility, large database sizes, better application compatibility.

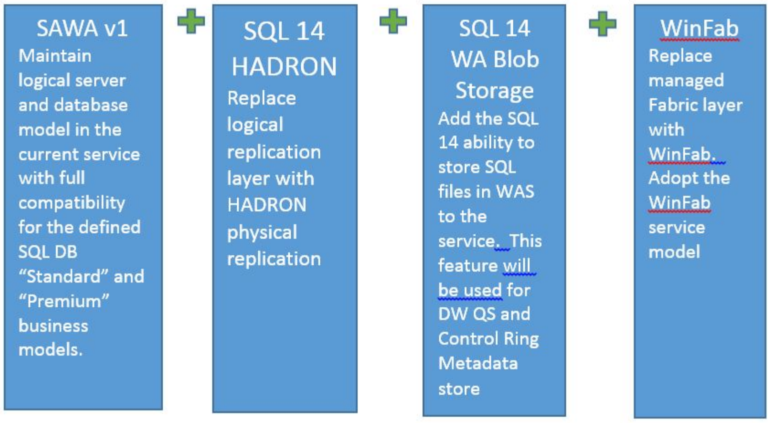
The new architecture looks to improve internal aspects like availability and reliability, deployment robustness and speed, replication performance, storage and database density, security isolation, performance isolation, and reduce single points of failure.

What is Sterling

The Goal: As we introduce the Azure SQL DB Premium offering we will need to improve the reliability and performance predictability of the service to match the expectation of the premium customers. In addition, we need to light up some key SQL features to create more value in our premium offering and enable Data Warehousing Query Service to build on the platform.

The Problem: The current SAWA managed Fabric, logical replication and service model are at their architectural limits and the fragility of the system prevents us from attaining the above goals.

The Deliverable: The Sterling project will deliver a platform with 4-9s availability and predictable performance to power the premium Azure SQL DB offering and provide the platform for DW QS preview.



SAWA v1
Maintain logical server and database model in the current service with full compatibility for the defined SQL DB "Standard" and "Premium" business models.

SQL 14 HADRON
Replace logical replication layer with HADRON physical replication

SQL 14 WA Blob Storage
Add the SQL 14 ability to store SQL files in WAS to the service. This feature will be used for DW QS and Control Ring Metadata store

WinFab
Replace managed Fabric layer with WinFab. Adopt the WinFab service model

Picture 1

Platform Benefits

- Solid foundation for Azure SQL DB for the long term
- Robustness, Performance, Scale, Density – Overall COGs
- Foundation for next gen HW
- Foundation for SQL Data Warehousing service
- Raising abstraction layer between Azure SQL DB and WA
- Big steps towards making Azure SQL DB "just another WA tenant"

Similarities and Differences between SAWA v1 and SAWA v2 (Sterling)

Microsoft Windows Azure SQL Database is a cloud-based relational database service that is built on SQL Server technologies and runs in Microsoft data centers on hardware that is owned, hosted, and maintained by Microsoft.

Similar to an instance of SQL Server on your premises, SQL Database exposes a tabular data stream (TDS) interface for Transact-SQL-based database access. This allows your database applications to use SQL Database in the same way that they use SQL Server. Because SQL Database is a service, administration in SQL Database is slightly different.

Unlike administration for an on-premise instance of SQL Server, WA SQL DB abstracts the logical administration from the physical administration; you continue to administer databases, logins, users, and roles, but Microsoft administers the physical hardware such as hard drives, servers, and storage. This approach helps SQL Database provide a large-scale multi-tenant database service that offers enterprise-class availability, scalability, security, and self-healing.

Because Microsoft handles all of the physical administration, there are some differences between SQL Database and an on-premise instance of SQL Server in terms of administration, provisioning, Transact-SQL support, programming model, and features. For more information, see General Guidelines and Limitations (Windows Azure SQL Database).

This document describes the re-architecture of WA SQL DB called Sterling or Project Sterling. This document does not intent to serve as an introduction to WA SQL DB. It is written with an eye to explaining the salient points of the architecture to a reader familiar with current Azure SQL DB system.

For an introduction to Windows Azure SQL Database please to this page.

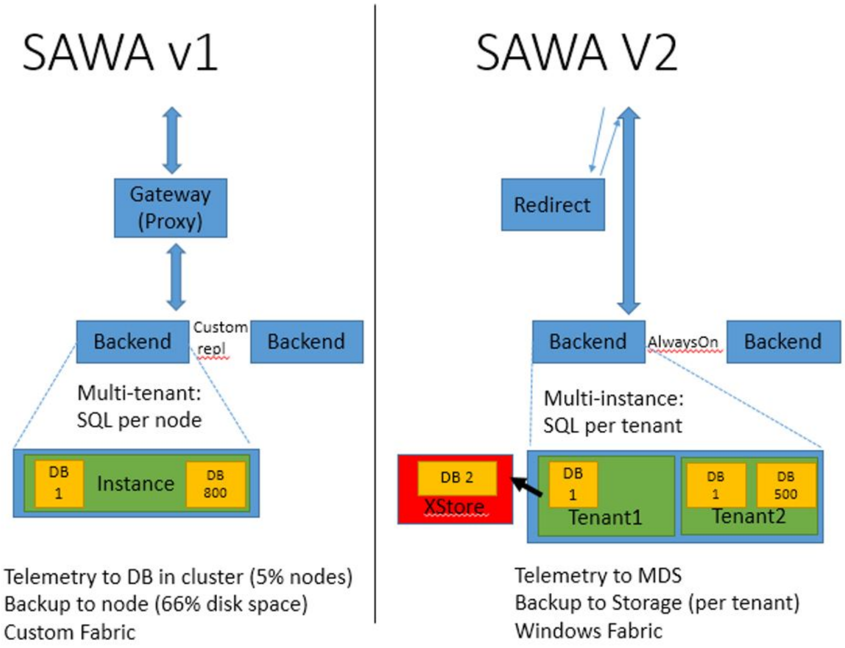
Insights

SQL DB is not a software rental business. We sell a database service; the language of performance for databases is transactions per unit-of-time. We can incorporate predictability into the performance promise.

Principles

We will not expose details about the underlying hardware resources. We will describe performance as a measure of database throughput. The customer experience of performance should be approximately what they would get from a dedicated computer.

List of Differences between SAWA v1 and Sterling



Picture 2

SAWAv1	SAWAv2 (Sterling)
Logical Replication (SERepl)	Physical Replication (HADRON)
Limited SQL compatibility	Almost full programming surface area (Hekaton, CCI)
All state on local storage (LS)	Local and remote storage, eventually tiered
Logging, auditing, backups local	Azure storage (XStore)
Managed Fabric	WinFab
Gateway proxy in all query paths	Redirect to direct connect
Multi-tenant per SQL instance	Single-tenant per SQL instance
AD joined, single security context for entire ring	Cert based auth, per-tenant security context
Resource governed inside SQL instance	Governed outside where possible; inside for some cases
Telemetry in Opstore	Telemetry in MDS via XEvents

Difference 1: No More Gateway

- Client connects to a redirector that uses TDS redirects to a back-end node. Old clients make use of a proxy service to redirect at the SNI packet level to the back end node. Login processing is driven from each backend node.

- Regular TDS commands no longer proxy faster RTT latency

- No gateway No GSQL Parser (duplication removed)

- No gateway No GW Metadata (will be replaced)

- No extra hardware nodes to route connections, so our COGS should go down

Difference 2: Service Isolation

- We want single-service deployments to become normal/common. Services will be loose coupled to guarantee simple unit deployment. This means that components need to take fewer dependencies on each other

- WinFab gives us ability to isolate and deploy services more formally. Services have a stronger definition. Deployments get easier/more discrete. WinFab takes over state around replica primaries (GPM).

Difference 3: Control Ring

- A cluster will not have one big ring, we will have at least 2 rings.
- Front-End services move into a control ring. Back-end services are in one or more tenant rings

Difference 4: No Sharing of SQL Instances

- We will not share SQL Instances across users
- Move isolation boundary to OS + SQL Server process boundary
- Gives us better security isolation
- We don’t have to build perfect RG in all cases
- We will have containers for each SQL Instance (to achieve the same objective of t-shirt sizes in SAWA v1)
- Isolated DB + Log files
- We also can imagine turning on features within an instance (Fulltext, TDE, Traceflags)
- This gets us much closer to SQL Server internally (though initially our exposed surface changes are limited)

Difference 5: Enable Remote Storage

- Possibility to have databases stored “locally” on the node and use Hadron for HA or stored “Remotely” on XStore (like SQL in a VM does) and use XStore replication for availability.
- Today we can’t get the same IOPS from WA Storage, but we expect this to improve in the future.

Difference 6: Re-do metadata

- No more GPM
- No more Gateway Metadata
- We centralize metadata into a new component called CCM (Central Cluster Metadata) and in a single database called CMS (Central Metadata Storage).

Difference 7: Data Exhaust Model

- We will move to a much more data telemetry-driven model
- All features generate data exhaust (That means all features will exhaust telemetry data for troubleshoot)
- We copy it all off the clusters into MDS, then to a DW (Telemetry exhaust exist in the for of MDS logs. MDS logs can be leveraged for alerting. MDS logs are then imported into a Data warehouse for analytical purposes)
- You can alert in MDS and you can write queries in the DW
- Testers write tests over the telemetry, not over the cluster itself

- We will stop using on-cluster tooling
- We eliminated MSDB
- We eliminated Opstore
- PII data generally scrubbed out

Difference 8: Gen3 Hardware

- We cannot have our own custom HW SKU anymore, MS consolidates and saves money, can get more hw more regularly.
- We are using Gen3 WA Compute SKU, 4-5 SSDs (about 500GB of primary space total), 1-4TB rotating disk
- No space for anything except databases
- Telemetry, backups, dumps, ... - all go out to WA Storage instead

Feature comparison

The following table compares some features in SQL 14 Box, WA SQL DB SAWA v1 and Sterling

Program/Feature	Box (SQL 14)	Support in SAWA v1	Support in Sterling
.Net providers	Supported	Supported	Supported TDS >= 7.4 desired
JDBC, ODBC, PHP, Node.js ODBC for Linus	Supported	Supported	Supported TDS >= 7.4 desired
Session Tracing ID	Not supported	Supported	Supported
Back –Up & Restore	Supported	Supported through applications such as SSIS or BCP.	Restore Supported per offering. Premium/Standard capacity to restore varies. Backup is automatic.
Import/Export	Supported	Supported	Supported
Copy Database as	Not supported	Supported	Not Supported. Customer can use Database Restore Instead.
Database Cloning	Supported	Not Supported	Not Supported
Database Collation	Supported	Can set database collation	Can set database collation
Database Mirroring	Supported	Not Supported	Not Supported
Database Size	524,272 terabytes	Web 100MB, 1GB, 10GB Business 10GB, 20GB, 30GB, 40GB, 50GB, 100GB, 150GB Basic 100MB, 500MB, 1G, 5G	Up to 500GB

Program/Feature	Box (SQL 14)	Support in SAWA v1	Support in Sterling
		Standard: From 100MB to 250GB Premium: From 100MB to 500GB	
Database Tuning Advisor	Supported	Not Supported	Not supported
Distributed Queries and Transactions	Supported	Not Supported	Undefined
Entity Framework	Supported	Supported	Supported
File Stream	Supported	Not Supported	Not Supported
Full Text Support	Supported	Not Supported	Supported but no ETA
Linked Server	Supported	Supported as target	Undefined
Log Shipping	Supported	Not Supported	Not Supported
MARS	Supported	Not Supported	Undefined
Number of databases that can be created in one server	32,767	500	Undefined
Policy Based Management	Supported	Not Supported	Undefined
Recovery from logical corruption	Backup / Restore is supported	Backup / Restore is supported	Backup / Restore is supported
Recovery from logical data loss	Backup / Restore is supported	Backup / Restore is supported	Backup / Restore is supported
Recovery from Physical corruption	Backup / Restore is supported	Backup / Restore is supported	Backup / Restore is supported
Recovery from physical data loss	Backup / Restore is	Backup / Restore is supported	Backup / Restore is supported

Program/Feature	Box (SQL 14)	Support in SAWA v1	Support in Sterling
	supported		
Replication	Supported	Not Supported	Not Supported
Resource Governance	Supported	Service implements Resource governance internally for distributing resources among workloads.	Service implements Resource governance internally for distributing resources among workloads. *users might be able to create their own pools
Service Broker	Supported	Not Supported	Not Supported
Show plan/Statistics	Supported	Supported	Supported
Snapshot a Database	Supported	Not Supported	Not Supported
Spatial Data Types	Supported	Supported: geography, geometry	Undefined
SQL Agent	Supported	Not Supported	Not Supported
SQL Profiler	Supported	Not Supported	Undefined Potentially will be xevents based
SQL Server error logs	Supported	Not Available	Not Available
SQL Server startup parameters	Supported	Not Supported	Not Supported
SQLDumper	Supported	Not Supported	Not Supported
T-SQL Security	Supported	Supported	Supported
XEvent	Supported	Not Supported	Supported
Table Partitioning	Supported	Not Supported	Supported
SAFE Common Language Runtime (CLR) and CLR User-	Supported	Not Supported	Supported

Program/Feature	Box (SQL 14)	Support in SAWA v1	Support in Sterling
Defined Types, Aggs, Functions, Procs			
Hekaton	Supported	Not Supported	Supported not ETA
Parallel Queries	Supported	Not Supported	Supported
Buffer Pool Extension	Supported	Not Supported	Not Supported
New cardinality estimation engine	Supported	Not supported	Supported
Column Store Indexes	Supported	Not Supported	Supported
Sequence Objects	Supported	Not Supported	Supported
Create/drop application role	Supported	Not Supported	Supported
Katmai Functions (TRY_CONVERT, CUME_DIST, FIRST_VALUE, LAG, LAST_VALUE, LEAD, PERCENTILE_CONT, PERCENTILE_DISC, PERCENT_RANK)	Supported	Not Supported	Supported
Data Compression	Supported	Not Supported	Supported
Online Clustered Index Rebuild w/LOB Column	Supported	Not Supported	Supported
Large Index Rebuild	Supported	Has issues with tlog size	Supported
SELECT INTO	Supported	Not Supported	Supported
CHECKPOINT	Supported	Not Supported	Supported
non-sysadmin DBCCs	Supported	Not Supported	Supported
read/write/updatetext	Supported	Not Supported	Supported
Alter database	Supported	Supported but limited	Supported

Program/Feature	Box (SQL 14)	Support in SAWA v1	Support in Sterling
CREATE/DROP RULE/DEFAULT	Supported	Not Supported	Supported
Change tracking	Supported	Not Supported	Supported
KILL @spid	Supported	Not Supported	Supported

SAWA v1 and SAWA v2 (Sterling) contrasted and compared

This table summarizes differences between SAWA v1 and SAWA v2 architecture. For a detailed SAWA v2 architecture investigation please read SAWA v2 aka Project Sterling - Architecture Overview

	SAWA V1	SAWA v2
Architecture	Composed of a single ring, and many state full tightly coupled components.	Has an Isolated Tenant ring and a Control ring and loosely coupled components that can be deployed/updated individually. Most components are stateless. No single points of failure and loose coupling of component services. Control ring has services to control the service state and database catalog. Tenant ring host databases and provide high availability.
Service State	Stored in diverse databases like GatewayMetadata, GPM and LPM. Some databases exist on the nodes while others exist a central location. GPM database can be recreated from information on the LPM databases and works as a cache.	Service metadata and state is stored in the Cluster Metadata Store (CMS) database. CMS uses a remote storage database as HA is provided by WinFab storage replication. RTO is guaranteed by Hadron replication. Cluster Metadata Store (CMS): provides a single logical point of metadata storage for a SAWA v2 cluster. The store provides the source of 'truth' regarding the state of the cluster and its resources. CMS also provides the persistence for state-machine driven workflows driven from within the Management Service that control all updates to resources and which ensures that the metadata either reflects a known stable state of a resource or that a workflows is in-progress currently modifying the resource.
Historical Telemetry	Stored in the OPSStore repository. Requires specific software for access.	Imported to a central data warehouse. Regular SQL tools can be used to access the data.
Telemetry stored on the node	MSDB database on every node has node global data stored for all databases running on the node. We have history of using this data to	There is no MSDB database on nodes. All telemetry data is saved to a MDS repository.

	SAWA V1	SAWA v2
	troubleshoot recent performance issues.	
Feature Surface	Limited by design.	Potential to expose almost all SQL on-premises surface.
HA	Uses proprietary replication at the SQL Engine Level.	Can use Windows Storage replication or Hadron depending on Edition and desired features. The single HA model (replacing SEREPL) and make this technology work for both LS and RS with minimal differences. HADRON physical replication is required for box surface area, security isolation as well as higher perf and more robustness. In LS HADRON is used for persistence while in RS is it used for availability (low RTO), but fundamental infrastructure is not different and indeed both cases can be used in a single replication topology.
Login	Gateway component running on a pool of machines is responsible to proxy all connectivity to the back end nodes. This proxy is done at the TDS layer.	Has lower latency as newer application client drivers accepting redirection can be redirect to back end tenant ring, hence connecting directly to the SQL Server instance hosting the user database. Clients that do not support redirection are proxy at the SNI-Level
Fabric	Proprietary Fabric	Uses WinFabric leveraging as much of Azure platform as possible.
Database location	Databases are stored on disk drivers located on the nodes.	Databases can use local or remote storage.
Security	Because many databases use the same SQL Server Instance, all databases in a node can be compromised.	Clients might not share SQL Instances for isolation. SQL instances each running with low privilege and each running with different accounts, leveraging Windows mechanisms wherever possible for resource isolation.

	SAWA V1	SAWA v2
Federations	Supported	Not-Supported
Create DB as Copy	Supported	Not-Supported
Client Library		Support the same clients as SAWA v1. Clients using TDS version 7.4 or above will make use of redirection to connect to the back end instances. Pre 7.4 clients will use a SNI level proxy component.
Client Redirection	State full TDS level gateway	Stateless redirector or SNI level gateway service for older clients. Stateless Alias service is used to redirect clients to databases running on the tenant ring.

How good have you found this content?

🙂 🙁