Database Design and Scalability

Ext.prc.002.04 | 01.06.2024

UPM 24.x

ZINFI Confidential & Proprietary
Shared Under NDA

Contents

Details of Database Type and Scalability

ZINFI uses Microsoft SQL Server's relational database management system (RDBMS) (version2017) to structure and process the data at the back end.

Hosted in the MS Azure Infrastructure - the Azure SQL Database offers the DTU-based model and the vCore-based model. The DTU-based model offers a blend of compute, memory, and I/O resources in three service tiers to support lightweight to heavyweight database workloads: Basic, Standard, and Premium. Performance levels within each tier provide a different mix of these resources to which we can add additional storage.

The service tier, computer tier, and resource limits for a database, elastic pool, or managed instance can be changed anytime. For example, you can build your first app on a single database using the serverless computer tier and then change its service tier manually or programmatically at any time to the provisioned compute tier to meet the needs of your solution.

Capabilities of the Database Infrastructure

- Our database server has high-end query performance, which can be boosted by 10Xto 100X with the xVelocity in-memory column-store index for data warehousing.
- The infrastructure performs more consistently for concurrent and mixed workloads by defining resource usage with the Resource Governor.
- We can monitor specific events and capture server events for near-real-time diagnosis using SQL Server Profiler.
- We can visualize resource utilization with multi-server management to proactively make changes before trouble arises.
- The infrastructure can collect data from T-SQL queries and bundle it into a collection set for analysis using the Performance Data Collector.
- I/O intensive workloads can be accelerated by cutting the growing volumes of data by approximately 50%–60% with high-grade compression capabilities.

Scale Out (Horizontal Scaling)

- The infrastructure can grow beyond the constraints of any deployment environment with hybrid IT opportunities across traditional and cloud servers.
- We can expand data warehouses with database features like Remote Blob Storage and partitioned tables that scale to 15,000 partitions.
- We can scale out heavily accessed databases while maintaining consistency with peer-to-peer replication.
- The infrastructure can leverage Service Broker to build highly scalable, service-oriented solutions.

Scale Up (Vertical Scaling)

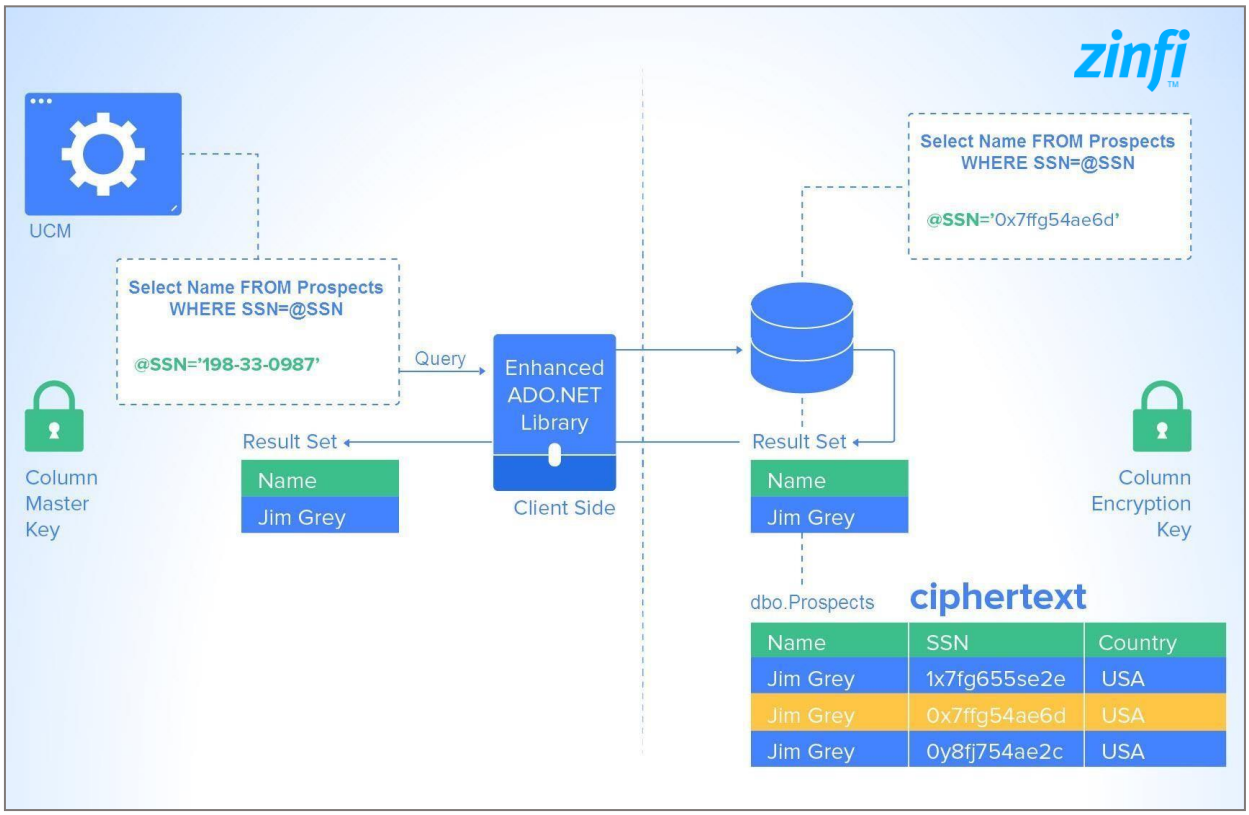
- We can improve scalability and performance of large workloads and consolidation scenarios with up to 320 logical processors and 4TB of memory.

- Our SQL Server deployment in virtual machines can use up to 64 virtual processors and 1TB of memory.

Features of the Updated Database Architecture

- **Always Encrypted**

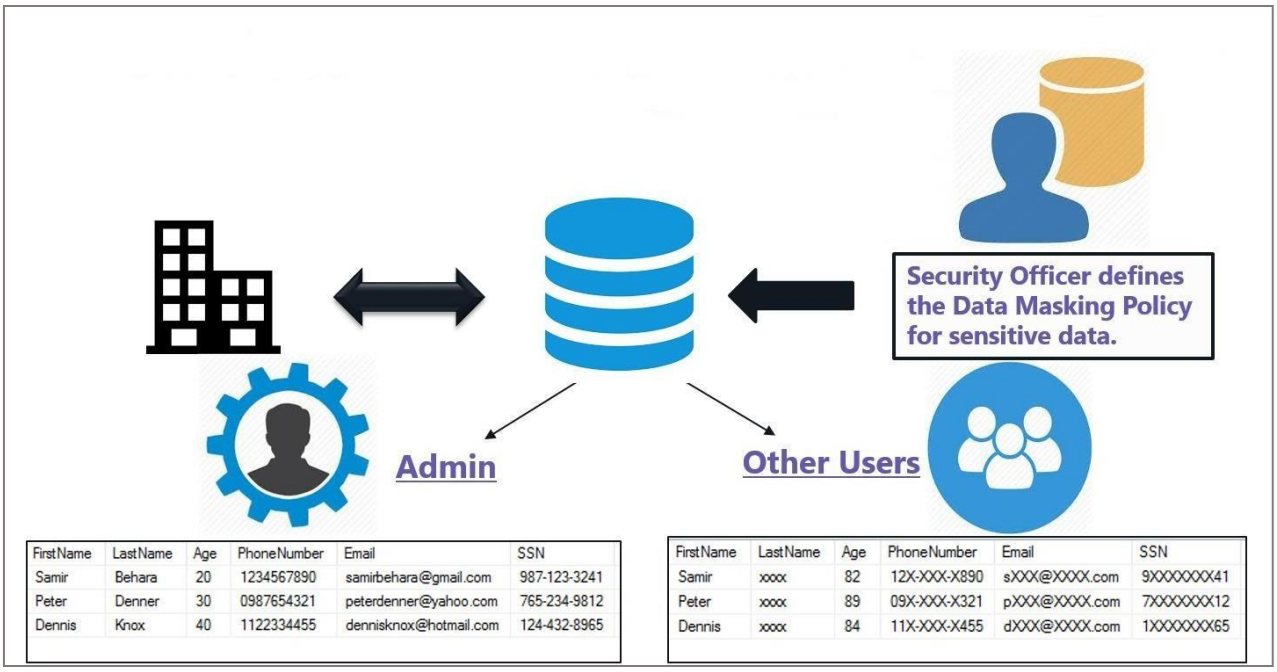
  The Always Encrypted feature protects data and enables SQL Server to perform encrypted data operations so that owners can protect their confidential data by using an encryption key. This feature ensures that your important data stored in the managed database remains encrypted and protected.



 |

- **Dynamic Data Masking**

  This feature gives an obscured version of your confidential data to selected partners and allows only authorized users to view it by defining masking rules.

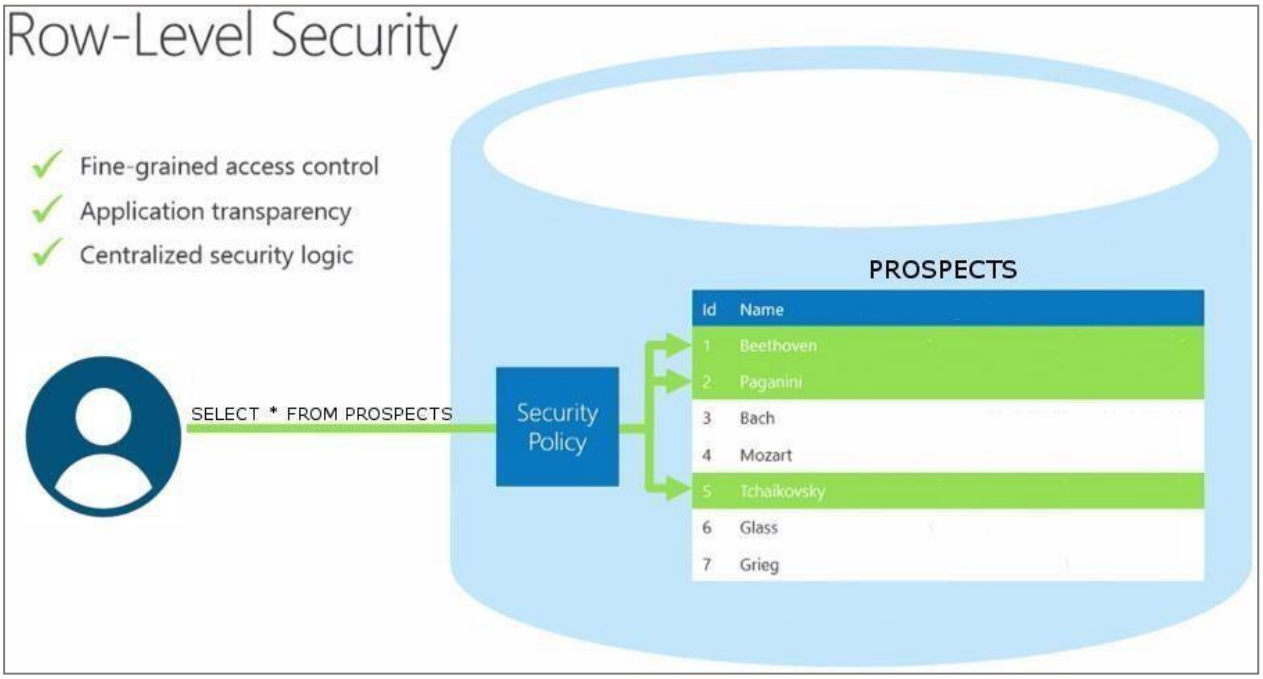


- **Real-time Operational Database**

  The real-time operational database prepares ZINFI's Unified Partner Management (UPM) platform for optimal transactional performance and helps to increase workload consistency by combining in-memory online transaction processing (OLTP) with in-memory column-store.

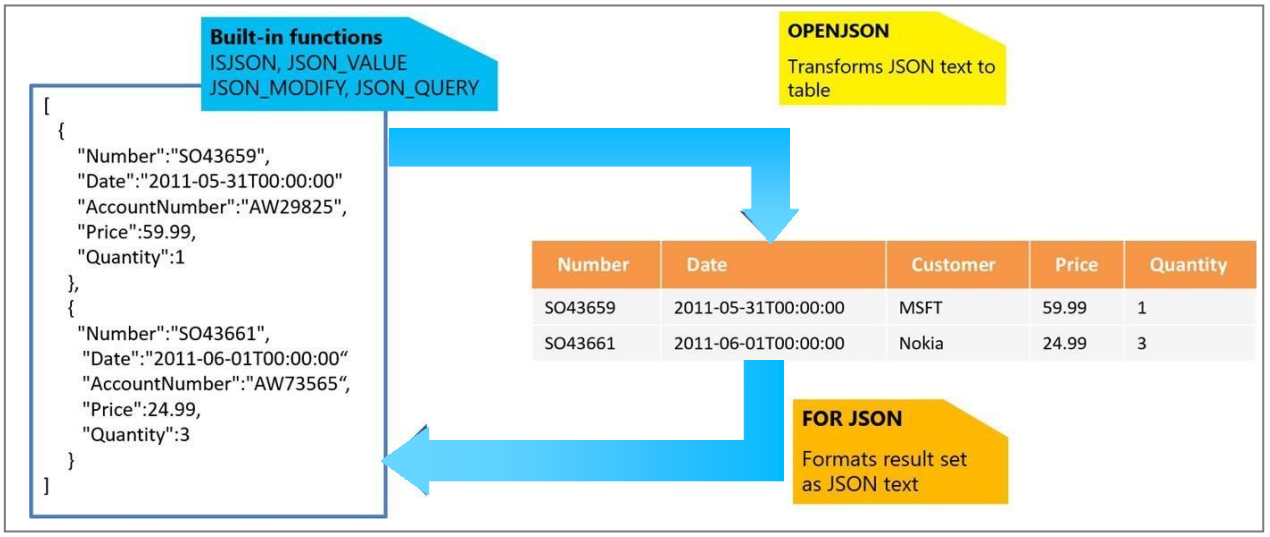
- **Row-Level Security**

  The Row-Level Security feature is a major development in this database management system. It restricts some users from viewing data in tables by requiring an SQL Server login, and allows you to implement row-level security to ensure that new users will be unable to detect whether rows of data were filtered for restricting data.



- **JSON Support**

  The Java Script Object Notation (JSON) feature allows you to interchange JSON data between the SQL Server database engine and various other applications. It can analyze JSON formatted data and even convert relational data into JSON format for convenient storage.



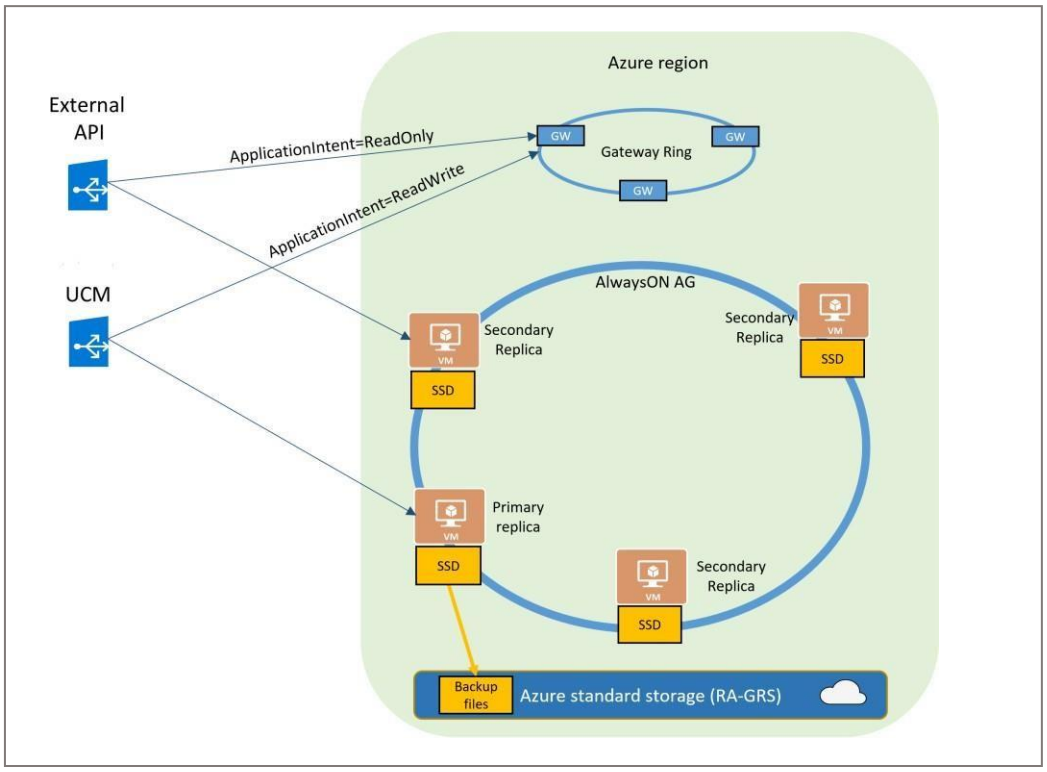
- **Database Connection - read only connection for read operation from External API**

  Read-only replicas are implemented to offload read-only query workloads. Read Scale-Out is a feature that allows you to load balance Azure SQL Database read-only workloads using the capacity of read-only replicas. This capability redirects the read-only client connections to one of the automatically provisioned HA replicas and effectively doubles the compute capacity of the database or elastic pool at no additional charge. This is ideal for load balancing of complex analytical workloads without affecting the primary OLTP workload.

  The Read Scale-Out feature allows to load balance SQL Database read-only workloads using the capacity of one of the read-only replicas instead of all queries hitting the read-write replica. This way the read-only workload will be isolated from the main read-write workload and will not affect its performance.

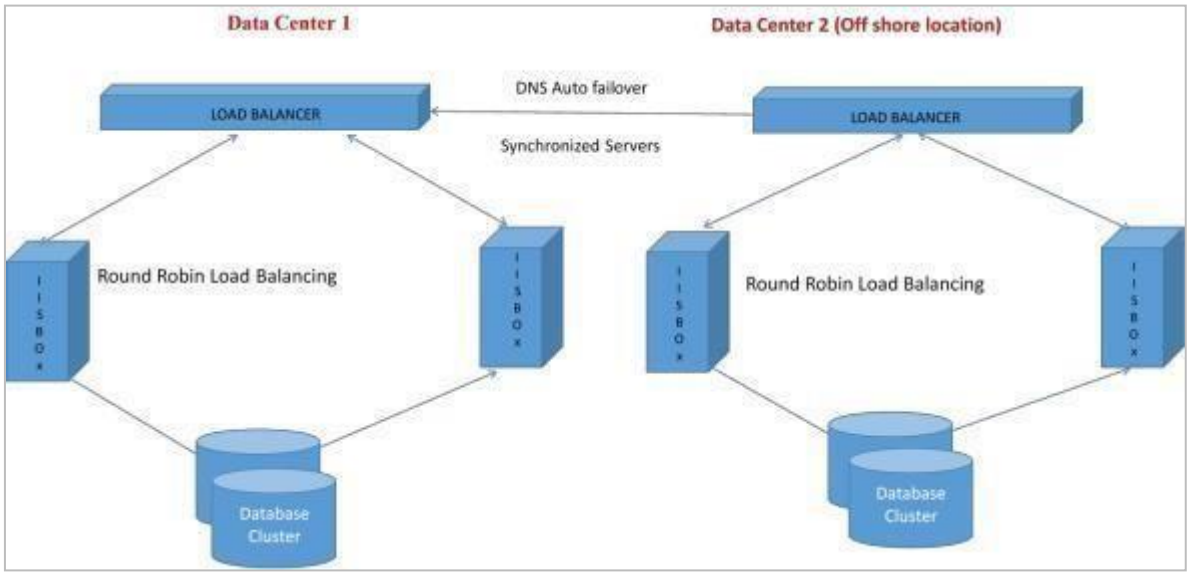
  As part of High Availability architecture, each database, is automatically provisioned with a primary read-write replica and several secondary read-only replicas. The secondary replicas are provisioned with the same compute size as the primary replica. The read scale-out feature allows us to offload read-only workloads using the compute capacity of one of the read-only replicas, instead of running them on the read-write replica. This way, some read-only workloads can be isolated from the read-write workloads and will not affect their performance.

  When we conduct a read operation from External API we enable read scale-out for a database, the ApplicationIntent option in the connection string is updated dictating the connection is routed to the read-only replica. When connected to a read-only replica, Dynamic Management Views (DMVs) reflect the state of the replica and are queried for monitoring and troubleshooting purposes.



Details of Hardware and Scalability

The following is an overview of our hardware design to support scalability along with load balancing and failover.



Server Hardware Capabilities

- **CPU and memory:** The CPU processes the instructions received by the computer. Faster clock speeds and multiple processors increase the performance scalability of a web server, particularly for sites that rely on dynamic content. 64-bit versions of Windows run on the x64 family of processors from Intel and can address a large amount of primary memory. ZINFI uses multiple CPUs with many gigabytes of memory, which is further upgradable.

- **SMP:** Internet Information Services (IIS) supports symmetric multiprocessors (SMPs) and can use additional processors to improve performance. If the system is running only IIS and doesn't handle dynamic content or encryption, a single processor with multiple cores suffices. We always use multiple processors if IIS is running alongside other services.

- **Disk drives:** The data storage capacity we need depends entirely on the size of content files and the number of sites supported. Enough disk space is required to store all data plus workspace, system files and virtual memory. I/O throughput is just as important as drive capacity. However, disk I/O is rarely a bottleneck for websites on the public Internet; generally, bandwidth limits throughput. High-bandwidth sites consider a hardware-based redundant array of independent disks (RAID) solutions using copper or fiber channel–based small computer system

interface (SCSI) devices. ZINFI has configured multiple hard disks in failover array in the web servers, which are further upgradable in hot swappable mode.

- **Data protection:** Hardware RAID implementations are always preferred over software RAID implementations. RAID 5 (disk striping with parity) offers good protection against single-drive failure, but exhibits lesser write performance. With that in mind, we've configured redundant hardware load-balanced servers. With load balancing, the additional servers offer the necessary fault tolerance. The load balancers also support remote data synchronization with another, similar setup to provide ultimate redundancy.

- **UPS:** Sudden power loss and power spikes can seriously damage hardware. To prevent this, our data centers are backed by an uninterruptible power supply (UPS). A properly configured UPS system allows the operating system to automatically shut down the server gracefully in the event of an extended power outage, and it's also important in maintaining system integrity when the server uses write-back caching controllers that do not have on-board battery backups.

Hardware Scalability Design

To focus system design on hardware scalability rather than on capacity, it is typically cheaper to add a new node to a system in to achieve improved performance than to undertake performance tuning to improve the capacity that each node can handle. But this approach can have diminishing returns.

For example: suppose 70% of a program can be sped up if parallelized and run on multiple CPUs instead of one. If α is the fraction of a calculation that is sequential, and $1-\alpha$ is the fraction that can be parallelized, the maximum speed that can be achieved by using P processors can be calculated according to Amdahl's law:

$$\frac{1}{\alpha + \frac{1-\alpha}{P}}$$

Substituting the value for this example, using 4 processors we get:

$$\frac{1}{0.3 + \frac{1-0.3}{4}} = 2.105$$

If we double the compute power to 8 processors we get:

$$\frac{1}{0.3 + \frac{1-0.3}{8}} = 2.581$$

Doubling the processing power has improved the speed by only roughly one-fifth. If the whole problem was parallelizable, we would, of course, expect the speed to double also. Therefore, throwing in more hardware is not necessarily the optimal approach. ZINFI performs regular server performance tuning based upon requirements in ideal use conditions. This process greatly emphasizes the overall server performance apart from hardware upgrades.

