Solution Brief

ScaleArc for SQL Server

For Deployment with SQL Server 2016

Summary

SQL Server 2016 offers a number of benefits, including AlwaysOn Availability Groups, but migrating to this version of SQL Server presents a number of challenges. ScaleArc offers a seamless and transparent path for migration, along with an essential set of features that take fault tolerance, performance, scalability, and visibility to the next level.

This solution brief outlines the benefits that ScaleArc provides when migrating to, and operating in, a Microsoft SQL Server 2016 database environment.

ScaleArc can be transparently deployed into SQL Server environments, including SQL Server 2005, 2008/2008 R2, and 2012/2014/2016. For those looking to migrate to SQL Server 2016 from an older version, ScaleArc can facilitate a faster migration at a lower operational cost. ScaleArc can be deployed in minutes and does not require any changes to existing applications or databases. The ScaleArc software can be deployed on bare metal, on VM, or in the cloud.

ScaleArc's patented technology provides the following benefits:

- enables automated higher availability with dynamic SQL query-level load balancing
- immediately improves performance – speeds response times up to 60X with the world's first transparent, query-level cache
- delivers real-time SQL analytics for instant troubleshooting and capacity planning
- enables scaling of your existing database infrastructure 10x or more, without making changes to databases or applications
- provides an elegant, simple SQL firewall as an added layer of protection

Overview

SQL Server 2016 introduces substantial improvements over SQL Server 2008 R2. Foremost among these improvements is the concept of AlwaysOn Availability Groups, which allows for enterprise-level high availability and disaster recovery compared to mirroring functionality.

A SQL 2016 AlwaysOn Availability Group supports a failover environment for a discrete set of user databases, known as availability databases, which are grouped and can fail over together. Availability Groups consist of a primary database and from one to eight sets of corresponding secondary databases, known as replicas. These can be comprised of 3 synchronous and 5 asynchronous replicas, which can be utilized for servicing read queries, while the primary replica can service all queries. The recommended deployment for SQL Server 2016 Availability Groups consists of deploying the databases as standalone instances on top of Windows Server Failover Clustering (WSFC) services, which are responsible for failover and fault tolerance. A WSFC group must be created for every Availability Group.

Challenges with Migrating to SQL Server 2016

SQL Server 2016 includes many compelling capabilities, but taking advantage of these features requires significant investment. The overall architectures are quite different (e.g., use of WSFC), demanding significant planning and design before migrating. Applications must also support the concept of ‘read intent’ strings to fully take advantage of the scale out environment offered by the usable secondary replicas in SQL Server 2016.

SQL Server 2016 includes compelling benefits, especially in the area of auto failover. But migrating to these versions requires significant application modifications to take advantage of the database’s new capabilities.

Load balancing is not aware of replication lag – SQL 2016 now supports round robin load balancing for readable secondaries. The round robin mechanism routes read-only requests through the Availability Group Listener, an improvement over 2012/2014’s approach where requests always go to the first secondary in the list. Load balancing is aware of replication health, but NOT replication lag. So if replication is suspended or broken, then that node will not receive traffic, but if replication is only lagging, the node will continue to receive traffic. This can introduce data integrity and consistency issues, which are anathema for mission-critical OLTP applications.

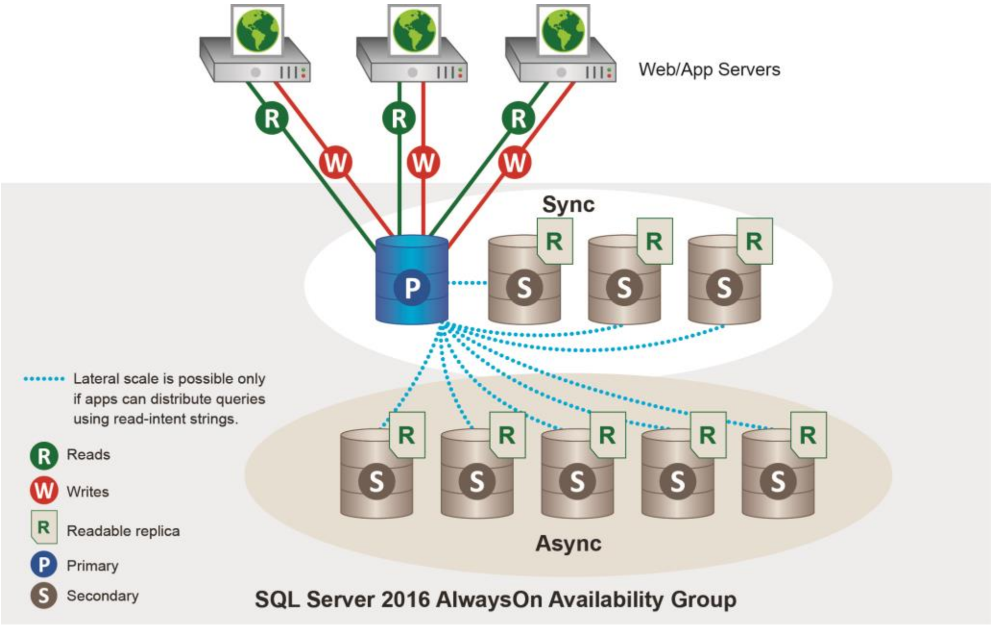
Connecting to the database environment via the Availability Group Listener leads to slow failure detection – A SQL 2016 Availability Group deployment relies on the concept of deploying standalone database instances on top of Windows Server Failover Cluster. Each Availability Group has what is known as the Availability Group Listener (AGL), which consists of a Virtual Network Name (VNN) that maps to one or more IP addresses and port combinations. The AGL has no intelligence above Layer 4 (TCP), and a client (application server) uses the VNN to establish connections to the primary replica. The connections can then be re-routed to one of the available secondary replicas based on the read-intent connection string. During failover, queries can be dropped or timed out until the failure is detected and connections actually time out.

Migration to SQL 2016 requires changes to applications that are difficult to support – Significant challenges exist around application-level support for read-only scale out using the Availability Group secondary replicas. For the intelligent read-only vs. read-write connections to be routed appropriately (via the AG Primary), the client-side connections must contain read-intent connection strings (Application Intent=ReadOnly). If a connection does not contain these parameters, only the primary replica will see the connection. Also, note that in this model, all connections go to the primary replica first via the VNN. The primary then takes care of routing the connections to the secondaries based on the connection strings.

Limited real-time visibility – While numerous Microsoft toolsets exist that support predictive analysis and database-level reporting, no current toolset allows for real-time logging and query-level analysis at a single aggregation point without installing server-side agents or resorting to things like periodic sampling. In addition, sorting through logs at the database level can be very time consuming, and logs cannot offer true performance characteristics from the client/application perspective.

Failover modes can lead to downtime – Within the SQL 2016 Availability Group model, the roles of primary and secondary replicas are dynamic in nature in that they can potentially change roles at any given point in time. All along this process, data synchronization is happening in the background across both synchronous and asynchronous replicas, but replication can affect the various forms of failover, depending on the replica. Overall, three forms of failover exist – automatic, manual, and forced (with possible data loss). Most of the time, automatic failovers are preferred, but in some instances a forced failover may be needed, which can result in application timeouts and queries being dropped while the roles are in flux. If all the synchronous replicas in an Availability Group are down, auto failover will not be triggered. Even when the failover mode is automatic for a given Availability Group set, there can be a potential lag in detecting a failure condition and promoting the new primary replica.

Performance still lags – Beginning with SQL Server 2012/2014, a myriad of performance enhancements focused around Microsoft xVelocity technology was introduced, allowing for better scale than previous SQL Server versions. There is still room for improvement, particularly when it comes to general offload of database traffic via a query-level cache. Applications with a high amount of read-only workload typically benefit from the use of query caching, especially if this cache is deployed at an aggregation point and can be leveraged across multiple database servers. Some caching mechanisms on SQL Server allow for tasks such as adhoc query caching, but controlling and purging cache at a granular query level, even across stored procedures, for example, is not possible. For example, a single insert into a table with a million rows that was cached will invalidate the entire cache associated with that table. Also, on-board SQL Server caching lacks built-in granular instrumentation that allows for quickly creating cache policies or viewing items such as real-time cache usage statistics. The cache is relevant only to that single server and cannot be leveraged across processes such as READ queries that may be going to other secondary replicas.


Web/App Servers
R
W
Sync
P
S
R
Lateral scale is possible only if apps can distribute queries using read-intent strings.
R Reads
W Writes
R Readable replica
P Primary
S Secondary
Async
SQL Server 2016 AlwaysOn Availability Group

A typical SQL Server 2016 deployment can lead to bottlenecks as the always-on primary database might be overloaded. Lateral scale is possible only if apps can distribute queries using read-intent strings.

Connection management faces constraints that impact performance – Since connection pooling is generally done at the application server in most Microsoft environments (ADO.net, JBDC driver for SQL, etc.), it is constrained to the single application server itself. Additional application servers creating new connections will have their own sets of connection pools for the SQL Server back end.

In the Availability Group model, when the application servers initiate their connections to SQL Server, they point directly to the VNN, which maps to the primary replica. The primary replica then is responsible for re-routing the queries to the secondaries based on the read-intent string, but the primary still has to do work in fielding the initial TCP connection. Database servers are not optimized for generic TCP connection handling, especially in high-volume scenarios, and surges can occur at any point, overwhelming the primary or secondary replicas. Since all queries are sent initially to the primary replica, the primary replica can become a performance bottleneck under heavy load.

ScaleArc for SQL Server – With 2016: Higher Availability, Instant Scalability, and Better Performance with No App Changes

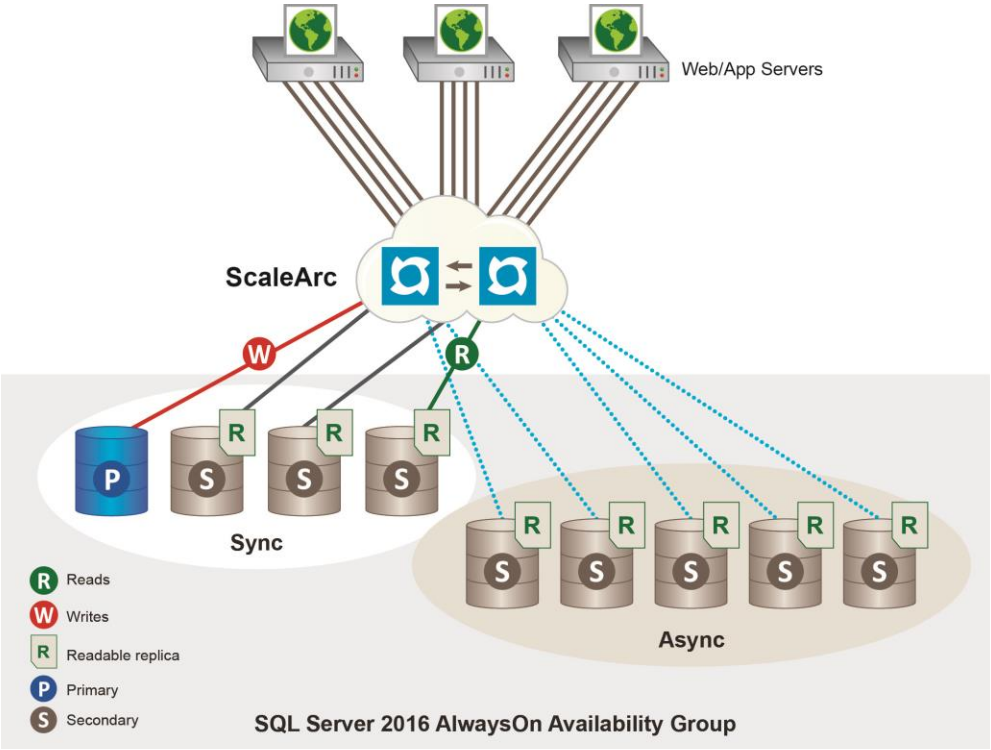
The ScaleArc database load balancing software abstracts application servers from SQL Server database servers to break the forced 1:1 dependency between the app and database tiers. ScaleArc enables zero downtime for apps – it provides auto failover, instant scale up, and transparent scale out. ScaleArc for SQL Server provides the following benefits:

- Scales your database 10x or more, and increase uptime, without any changes to your applications
- Boosts performance up to 60x with transparent caching
- Reduces troubleshooting time with real-time visibility into all queries

ScaleArc is a high-performance SQL proxy that can be deployed on bare metal, on hypervisors, or in a cloud environment. Additionally, it has been fully tested and deployed in Azure.

HA and Auto Failover

ScaleArc is a Tabular Data Stream (TDS) protocol-level proxy that enables automatic read/write split and provides dynamic load balancing, allowing for seamless deployments in SQL Server 2016 setups without requiring app/driver or database changes. Application servers do not have to insert any ‘intent’ connection strings to fully utilize the replicas for scale out of

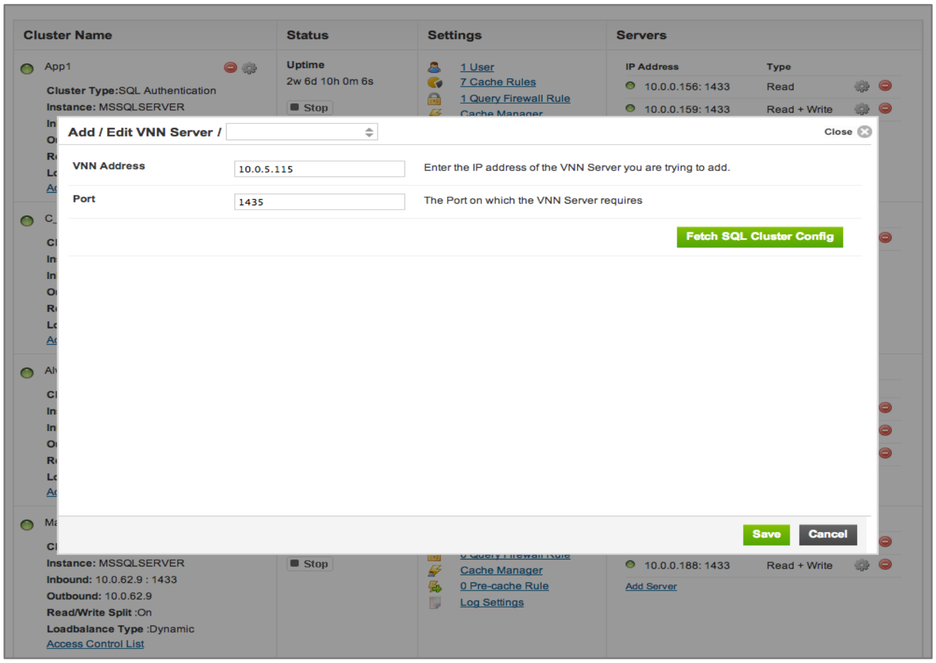


ScaleArc is a Layer 7 SQL proxy that deploys transparently, without requiring changes to applications or databases.

SQL 2016 functionality and of read-level connections. ScaleArc will automatically inspect each query and map it to connections on the back end established with the primary and secondary replicas, inserting the appropriate connection string on the fly. In performing query-level load balancing to all the replicas, using patented query-level connection management and load balancing, ScaleArc will select the best-performing database instance using characteristics such as Time To First Byte. ScaleArc measures database performance, end to end, from the application server perspective.

ScaleArc tracks replication lag between the primary and secondary replicas, allowing the administrator to dictate the tolerance of the lag behind the primary. ScaleArc monitors the lag by inserting tracer data and reading from all the replicas in parallel – this method is much more accurate than the generic stored procedure leveraged in SQL Server today.

ScaleArc achieves SQL Server 2016 availability by advertising and managing a floating virtual IP for the application servers to connect to at all times, thus abstracting the individual SQL Server servers/ Availability Groups from the applications. In a SQL Server 2016 environment, the ScaleArc floating Virtual IP takes the place of the VNN associated with WSFC, and it in turn will map to its own connections mapped to the back-end primary and secondary replicas. Connections are managed much more effectively at a query level, and all connections no longer have to be seen by the primary replica via the VNN. ScaleArc monitors the health of the databases using its own health checks and tracks the status of the Availability Group through monitoring the VNN.



App1

Cluster Type:SQL Authentication
Instance: MSSQLSERVER

Uptime
2w 6d 10h 0m 6s

1 User
7 Cache Rules
1 Query Firewall Rule
Cache Manager

IP Address	Type
10.0.0.156: 1433	Read
10.0.0.159: 1433	Read + Write

VNN Address: 10.0.5.115 — Enter the IP address of the VNN Server you are trying to add.

Port: 1435 — The Port on which the VNN Server requires

Instance: MSSQLSERVER
Inbound: 10.0.62.9 : 1433
Outbound: 10.0.62.9
Read/Write Split :On
Loadbalance Type :Dynamic

Cache Manager
0 Pre-cache Rule
Log Settings

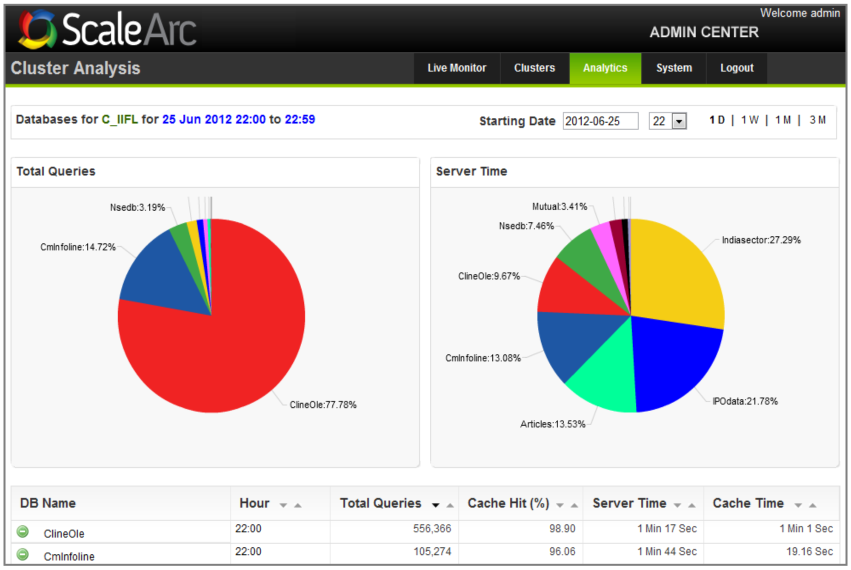
IP Address	Type
10.0.0.188: 1433	Read + Write

ScaleArc can be configured quickly and easily using its SQL Server 2016 auto-config via the VNN server.

ScaleArc is a full SQL proxy, so the application servers need not be notified of a SQL Server failure. All application servers continue to point to the Virtual IP address configured on ScaleArc. The ScaleArc software provides surge protection, which offers protection against flash events and denial of service attacks, along with a surge queue which automatically kicks in when a role change happens across the Availability Group, allowing for non-serviceable read-write queries to be intelligently queued until the primary replica is back on line. When that occurs, the queries are fed in a FIFO order to the primary replica, thus protecting the application server from dropped queries or timeouts. ScaleArc provides all these capabilities with 100% transparency to the applications. Failover times are much faster than when using the VNN in a standard SQL 2016 Availability Group deployment and prevent lost or dropped queries. Configuring ScaleArc with Availability Groups is a snap, as it allows you to simply point to the VNN. ScaleArc will automatically determine the replicas and their states on the fly. Servers can easily be taken out of rotation for routine maintenance without requiring changes to apps – enabling zero-downtime maintenance.

Real-time SQL Visibility

ScaleArc provides for unparalleled real-time visibility of all SQL traffic traversing through production database servers. SQL analytics are derived from de-duplicating granular log data as it's being centrally logged by ScaleArc. This approach does not require any sampling, as all queries are logged and all data is utilized. Nor does it add any performance overhead to the app or database servers. ScaleArc charts the comprehensive query/stored procedure performance data in a simple graph and highlights all frequent-but-slow queries for instant troubleshooting. These queries can be immediately added to the ScaleArc cache with a single click, to instantly accelerate database and application performance. Also, with ScaleArc's real-time SQL instrumentation, application developers and DBAs now have a non-intrusive, performance-centric view of the SQL query load, all with a simple click of the mouse. The ScaleArc SQL analytics can also be used for auditing the SQL traffic and analyzing performance bottlenecks.



Databases for C_IIFL for 25 Jun 2012 22:00 to 22:59

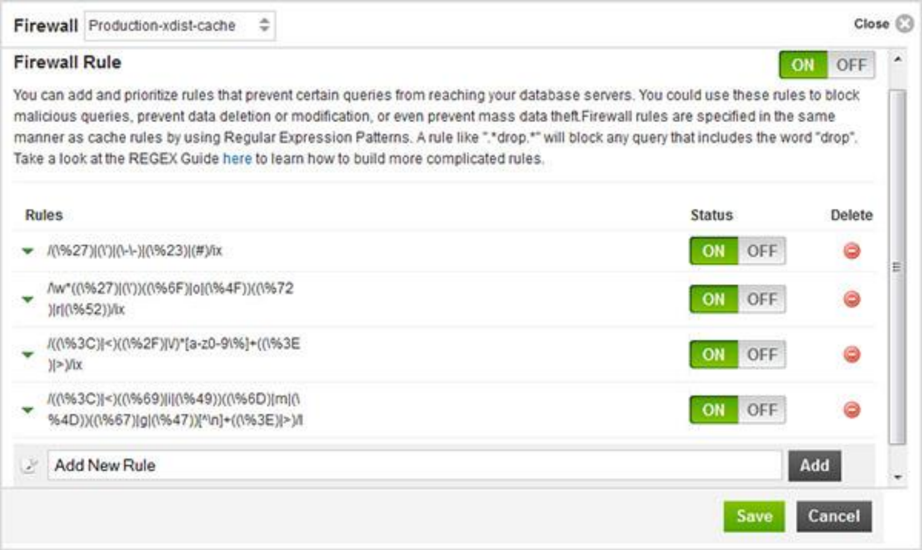
Starting Date 2012-06-25 22

DB Name	Hour	Total Queries	Cache Hit (%)	Server Time	Cache Time
ClineOle	22:00	556,366	98.90	1 Min 17 Sec	1 Min 1 Sec
CmInfoline	22:00	105,274	96.06	1 Min 44 Sec	19.16 Sec

The ScaleArc query analytics display provides insight into all SQL traffic, reducing troubleshooting time and helping with capacity planning.

Database Security

ScaleArc creates an "air gap" between the applications and databases. Applications connect to ScaleArc, not to the databases, and ScaleArc provides granular access controls on a per-tenant (i.e., per app) basis to provide much tighter lockdown of the database environment. By doing this, ScaleArc is the only technology that can provide full, granular database access logging for all connections, reads, writes, and transactions without any additional performance overhead on the database stack. This means that every single query is logged in great detail and there are tools for historical or real-time analysis on those logs, enabling audit-quality analysis and other capabilities. Query-filtering capabilities are available to transparently block specific query patterns at ScaleArc without having to touch the application code. You can block queries with a single click of the mouse button or through the ScaleArc API. With ScaleArc, you can eliminate downtime while patching



Firewall Rule ON OFF

You can add and prioritize rules that prevent certain queries from reaching your database servers. You could use these rules to block malicious queries, prevent data deletion or modification, or even prevent mass data theft.Firewall rules are specified in the same manner as cache rules by using Regular Expression Patterns. A rule like ".*drop.*" will block any query that includes the word "drop". Take a look at the REGEX Guide here to learn how to build more complicated rules.

Rules / Status / Delete

```
/(\%27)|(\')|(\-\-)|(\%23)|(#)/ix                                         ON OFF
/\w*((\%27)|(\'))((\%6F)|o|(\%4F))((\%72
)|r|(\%52))/ix                                                            ON OFF
/((\%3C)|<)((\%2F)|\/)*[a-z0-9\%]+((\%3E
)|>)/ix                                                                   ON OFF
/((\%3C)|<)((\%69)|i|(\%49))((\%6D)|m|(\
%4D))((\%67)|g|(\%47))[^\n]+((\%3E)|>)/I                                  ON OFF
```

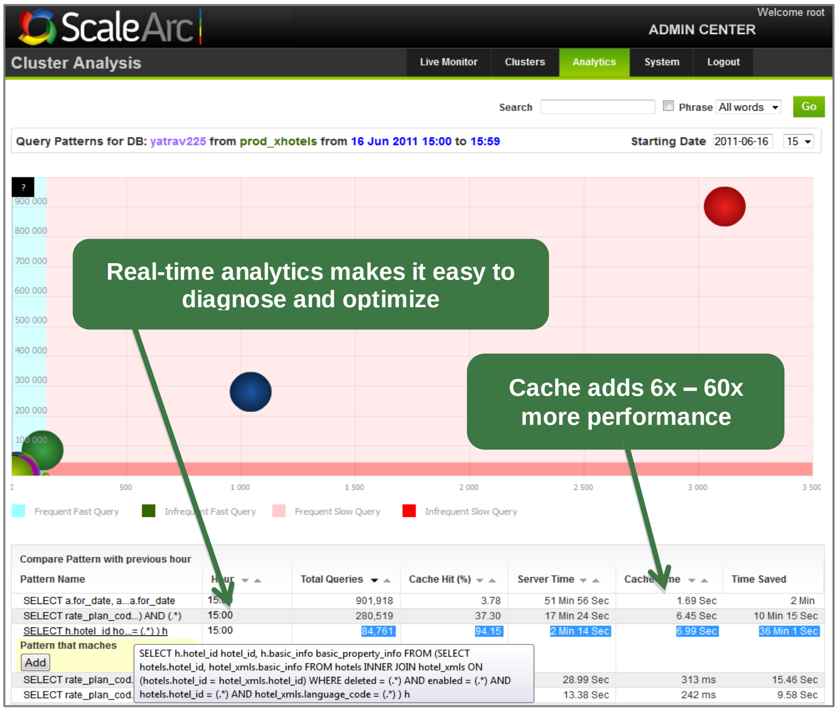
ScaleArc's SQL query firewall lets you simply block any unwanted SQL queries with simple Regex patterns that can be automatically generated from ScaleArc's analytics or custom crafted to suit certain specific use cases.

vulnerable databases without affecting your applications. Now you can simplify the process of keeping your database stack on the latest security patches.

Transparent Query-level Caching for Better Performance

ScaleArc's SQL caching technology is unique in the marketplace. It's an agentless approach that uses a NoSQL database to store repetitive query responses, thus enabling blazingly fast responses to subsequent matching queries. Caching with ScaleArc requires no application changes, can be deployed in minutes, and has finer granularity than other solutions. ScaleArc caching uses SQL query patterns generated from wire-speed de-duplication of all SQL queries. This approach speeds up application performance and reduces load on the database servers. Application response times improve significantly because the responses are served from memory rather than disk. ScaleArc has been shown to increase response times up to 60x.

The ScaleArc software is also the only product to support transparent caching of stored procedures, allowing an administrator to drill into the stored procedure and determine which components can, and should, be served out of the optimized ScaleArc in-memory cache. Cache can be purged in a variety of ways including pre-setting time to live using the fully RESTful ScaleArc API or by inserting application-level strings for various write operations. Cache can be cleared down to an object level, allowing for granular and efficient cache management.



Query Patterns for DB: yatrav225 from prod_xhotels from 16 Jun 2011 15:00 to 15:59

Real-time analytics makes it easy to diagnose and optimize

Cache adds 6x – 60x more performance

Compare Pattern with previous hour

Pattern Name	Hour	Total Queries	Cache Hit (%)	Server Time	Cache Time	Time Saved
SELECT a.for_date, a...a.for_date	15:00	901,918	3.78	51 Min 56 Sec	1.69 Sec	2 Min
SELECT rate_plan_cod...) AND (.*)	15:00	280,519	37.30	17 Min 24 Sec	6.45 Sec	10 Min 15 Sec
SELECT h.hotel_id ho...= (.*)) h	15:00	84,761	94.15	2 Min 14 Sec	6.99 Sec	36 Min 1 Sec
SELECT rate_plan_cod.				28.99 Sec	313 ms	15.46 Sec
SELECT rate_plan_cod.				13.38 Sec	242 ms	9.58 Sec

SELECT h.hotel_id hotel_id, h.basic_info basic_property_info FROM (SELECT hotels.hotel_id, hotel_xmls.basic_info FROM hotels INNER JOIN hotel_xmls ON (hotels.hotel_id = hotel_xmls.hotel_id) WHERE deleted = (.*) AND enabled = (.*) AND hotels.hotel_id = (.*) AND hotel_xmls.language_code = (.*)) h

ScaleArc highlights frequent-but-slow queries.

Connection Pooling and Multiplexing

ScaleArc provides immediate benefits with SQL connection offload, pooling, and management. Since ScaleArc is SQL-protocol aware, it can terminate SQL connections. ScaleArc provides the ability to isolate the client and server SQL connection stack and can thus maintain persistent SQL connections to the database servers across all the Availability Groups, reusing them for multiple clients as required. An integrated, tunable surge queue in ScaleArc manages concurrent connection bursts, providing protection to database servers from excessive load.

FEATURE COMPARISON

Feature	ScaleArc and SQL Server 2016	SQL Server 2016 only
CONTINUOUS AVAILABILITY		
Auto failover (not Zero Downtime)	Y	Limited
Zero downtime transparent auto failover	Y	N
Zero downtime maintenance	Y	N
Replication monitoring	Y	Limited
Surge queue	Y	N
DATABASE SECURITY		
Query firewall	Y	N
Zero downtime patching	Y	N
Traffic segregation / user access controls	Y	N
Centralized access logging, analytics, auditing	Y	N
PERFORMANCE AND SCALE		
Authentication offload	Y	N
Read/write split	Y	N
Dynamic load balancing	Y	Limited
Connection pooling	Y	N
Query response caching	Y	N
ANALYTICS		
Consolidated SQL analytics (real-time and historical)	Y	N
Live monitors	Y	N
RESTful API	Y	N
Historical statistical analysis	Y	N

Y = Yes **N** = Does not exist

RESTful API Makes Integration Easy

ScaleArc is easy to configure, automate, and manage using its developer-friendly RESTful API. Functions including cache management and server provisioning are managed via the RESTful API, and the RESTful API makes it easy to integrate ScaleArc into your existing applications.

> The ScaleArc RESTful API makes it easy to integrate ScaleArc into your existing applications.

Summary

SQL Server 2016 represents a new platform for the enterprise with many enticing new features and benefits. Still, end users and administrators face a number of challenges in migrating applications to SQL Server 2016 to fully take advantage of all the benefits. ScaleArc offers a seamless and transparent path for migration, along with an essential set of features that take fault tolerance, scalability, and visibility to the next level.

For more information about ScaleArc for SQL Server on SQL Server 2016, visit **www.scalearc.com**, or contact us at **sales@scalearc.com**.

ScaleArc™

2901 Tasman Drive, Suite 205
Santa Clara, CA 95054
Phone: 1-408-780-2040
Fax: 1-408-427-3748
www.scalearc.com

ScaleArc is the leading provider of database load balancing software. The ScaleArc software inserts transparently between applications and databases, creating an agile data tier that provides continuous availability and increased performance for all apps. With ScaleArc, enterprises also gain instant database scalability and a new level of real-time visibility for their application environments, both on prem and in the cloud. Learn more about ScaleArc, our customers, and our partners at **www.ScaleArc.com**.

10/21/15