



# Whitepaper

## **Building High Availability Data Infrastructure With Microsoft SQL Server**

September 2006



## **Growing Need for Information Availability**

Information is quickly becoming a competitive differentiator. An organization's ability to access actionable information at any time under any circumstance can determine its' ability to succeed in an on demand marketplace. The need for data access continuity is driving organizations to focus on building computing infrastructure that will provide persistent information availability across the enterprise.

Data access continuity requires uninterrupted availability of both the application and its underlying data. Network traffic management technology has addressed the issue of application availability however; the marketplace has yet to offer a similar cost-effective approach to enabling persistent data availability.

## **Current Approaches**

### *Cold Standby*

Backup and restore is a simple, low-cost, and reliable approach for managing data availability. Organizations that can tolerate an inability to access data for extended periods continue to rely on this proven methodology.

### *Warm Standby*

Replication and log shipping techniques enable complete content synchronization with redundant database instances. Depending on the technology, fail-over to standby resources can be a manual process or happen automatically.

One of the primary concerns with the warm standby approach is that it requires investment in deploying and maintaining redundant database computing capacity that will intentionally sit idle during normal operation.

### *Database Mirroring*

With SQL Server 2005, Microsoft introduced a mirroring feature that enables the database to toggle between primary (principal) and secondary (mirror) instances with minimal opportunity for data loss. Microsoft's mirroring technology is another iteration of the Warm Standby approach with the same inherent concerns and cost with idling redundant database computing capacity.

Whitepaper



### ***Database Clustering***

Microsoft SQL Server Clustering Services is a robust high availability solution that provides data access continuity with real time content replication and automated fail-over capabilities. However, database clustering is a sophisticated solution that has stringent infrastructure requirements, which, for many organizations, result in significant upfront investment as well as considerable ongoing costs.

### **Revolutionary Approach**

Resonate has developed a revolutionary approach to database high availability that does not require sophisticated and expensive clustering technologies or idling redundant computing capacity. With over 10 years of experience in developing network traffic management software, Resonate has adopted a breakthrough methodology to building real-time high availability infrastructure for Microsoft SQL Server.

### **Introducing dbDispatch**

dbDispatch is a simple network appliance that enables persistent data availability by managing database traffic.

### ***Database Traffic Management***

Unlike traditional high availability technologies, database traffic management is a fundamentally different approach to data access continuity. By intercepting, analyzing, and distributing database traffic originating from the application tier across multiple databases, dbDispatch enables completely uninterrupted data access.

### ***Key Features***

#### **Non-Intrusive**

The network appliance is deployed between the application and database tiers. By presenting itself as a single database instance to the application, dbDispatch is completely non-intrusive to the infrastructure and shields applications from the complexity of interacting with multiple databases.

Whitepaper

### Easy to Deploy

dbDispatch does not require any special configuration or modification of the application, database or storage tiers.

### No Single Point of Failure

dbDispatch is deployed in pairs and dynamically passes database traffic management responsibilities between appliances to ensure that it does not become a single point of failure.

### Shared Nothing Environment

Because dbDispatch manages the database traffic and not the database itself, each database can be completely independent.

### How dbDispatch Works

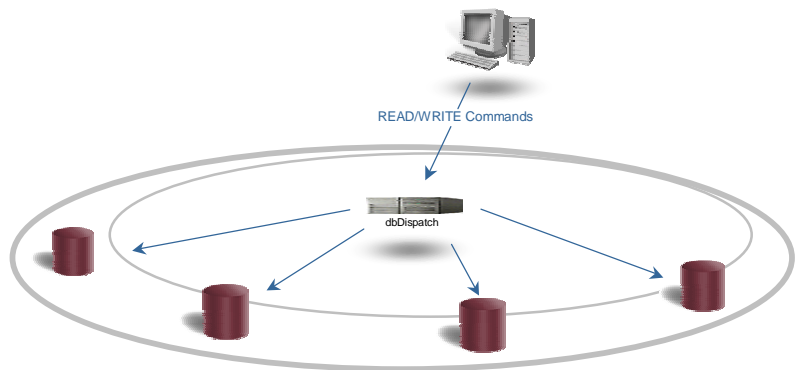
dbDispatch presents a single, virtual representation of multiple databases to an application. This allows the application to interact with dbDispatch as though it is the database and enables dbDispatch to manage the database traffic across multiple physical databases.

#### *Real-Time Synchronization*

Commands to write to the database are simultaneously sent to all available databases to ensure data synchronization.

#### *Intelligent Load Balancing*

By distributing read requests, dbDispatch eliminates the idling of redundant database resources by utilizing every available database to meet the information demands of the organization.

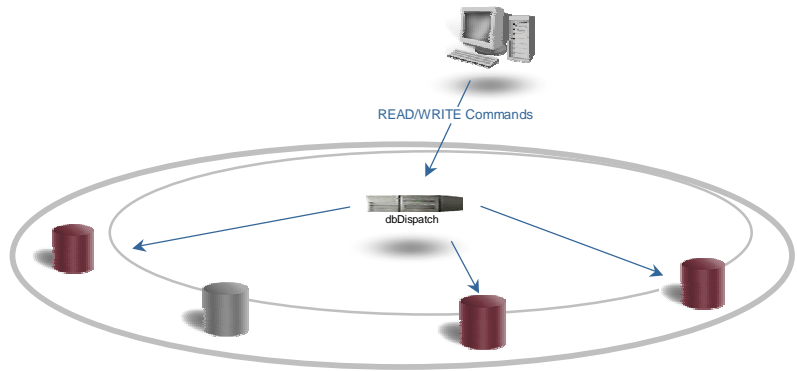


Real-Time Data Synchronization and Intelligent Load Balancing

### ***Eliminating the Need for Fail Over***

dbDispatch constantly monitors the availability of each database. If a database becomes unavailable for any reason, planned or unplanned, the appliance continues to route traffic to the remaining available databases. Because the application is not aware of the individual databases - it sees dbDispatch as a single database instance - there is no fail-over.

A database becoming unavailable is completely transparent to the application layer because dbDispatch continues to intelligently route traffic to the remaining available database instances with no service interruption to the application or error messages presented to the users.

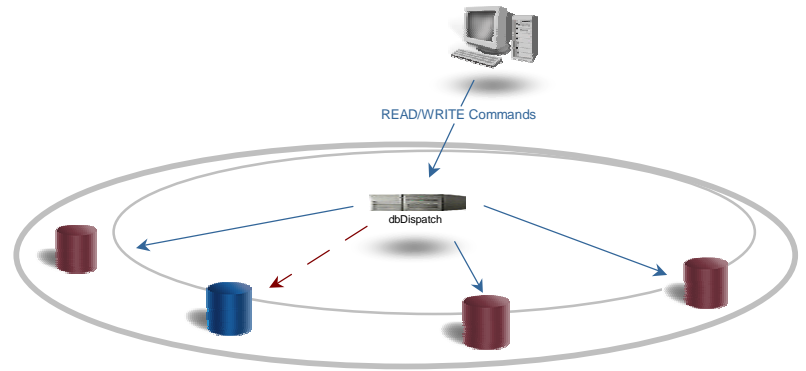


Persistent Data Access with an Unavailable Database

### ***Automatic Resynchronization***

When a database becomes unavailable, dbDispatch not only removes it from the traffic flow, the appliance also begins caching all database write instructions. As soon as the database comes back online, dbDispatch begins sequentially flushing the cache to resynchronize the content. During this process, dbDispatch continues to isolate the unsynchronized database from the real-time traffic.

Once the previously unavailable database is resynchronized with the content from the cache, dbDispatch automatically resumes full utilization of the resource.



Synchronizing a Previously Unavailable Database

### **Conclusion**

Having persistent access to critical information can make or break a company in today's marketplace. However, building robust data infrastructure can be costly and time-consuming.

dbDispatch enables organizations to build and maintain a resilient data infrastructure for mission critical applications quickly and easily using a simple, low-cost network appliance.

Whitepaper



**KEEPING YOUR BUSINESS OPEN FOR BUSINESS**

# Whitepaper

To learn more, call 1.408.545.5522  
or email [info@resonate.com](mailto:info@resonate.com).

This document is not warranted to be free of error or subject to any other warranties or conditions. Resonate disclaims any liability with respect to the contents herein. This document may not be reproduced for any purpose without prior written permission.

dbDispatch is a registered trademark of Resonate, Inc. Other names may be trademarks of their respective owners.

© 2006 Resonate, Inc. All rights reserved.