

Microsoft SQL Database Clustering and Load Balancing

One of the misconceptions I often run into during client engagements is what SQL clustering really is. More often, than not, people new to SQL server administration and/or SQL clustering are under the impression that SQL Server clustering results in a “load balanced” database service.

I want us clear up this misconception. SQL Server 2000 doesn't support out-of-the-box solutions for load balancing. As far as I have seen SQL Server 2005 does not either.

There are several resources that explain the options for a sort of load balancing for SQL server. However, all of the built-in SQL solutions do not provide real database load balancing. The built-in options are really designed to keep a cold standby copy of the data in a read only database, or updating a read only reporting SQL server in ‘near real-time’. There is a technique to “load balance” a SQL database by using “Federated Databases” in which a DBA manually splits the tables in a database between multiple servers and then creates Views for applications to access the data.

Another option to “load balance” a SQL server database, is by using a 3rd party application to manage/maintain a “read/write” database on two servers. There is a bit of information on this below and in the references found at the bottom of this entry.

One thing to keep in mind is, if you plan to implement any of these solutions, make sure the application you want to use with these SQL database load balancing option, supports it! Not all of them will. For example, a Microsoft SharePoint portal does not support this type of load balancing.

If you are interested in more information, keep reading.

Microsoft Clustering Services (MSCS) Microsoft Cluster Service (MSCS) does not do anything to help performance or scalability. Despite its misleading name, it does not enable you to scale out or distribute traffic. Instead, if one server fails, MSCS will fail over to a backup server, which will resume processing

Federated Database Design SQL Server Enterprise Edition comes with a feature called Distributed Partitioned Views (DPV), which enables you to create a federated database, to divide processing of queries between multiple separate database servers. With DPV, the database is partitioned into many member tables, storing each of those tables on separate database servers, and then creating a partitioned view over the set of member tables.

Continuous Replication The power in this approach is that it is relatively easy to configure, and it allows you a quick solution to load balance your database read access. The obvious drawback of this approach is that it does not use live data. It must be acceptable for clients accessing this database to use slightly delayed data.

Database Load Balancing Load balancing web servers and application servers is easy and straightforward because there's no data to persist. You can not use traditional load balancing for database servers, because the databases would get out of sync as soon as a write is made to one of them.

This kind of guaranteed synchronization can only be provided when load balancing is built into the database itself, as is done with Oracle's 9i Real Application Cluster, or provided through a middleware transaction server, such as Database Scattering, a new product offered by [Metaverse](#).

Detailed Information Resources:

- [Compare load balancing options for SQL Server 2000](#) By Baya Pavliashvili
- [Achieving Massive Scalability with SQL Server](#) by Doug Kerwin