

Re: Building high available and high performance web sites with MS

Source:

<http://www.tech-archive.net/Archive/SQL-Server/microsoft.public.sqlserver.clustering/2006-11/msg00098.html>

- *From:* Linchi Shea <LinchiShea@xxxxxxxxxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Thu, 16 Nov 2006 18:38:01 -0800
-

What is the purpose of adding extra nodes past the 2 node cluster if clustering is just for failover? I read about adding extra nodes when

One reason is cost. If you can live with the assumption that only one node may fail at a time, you can have three 'active' nodes sharing a common 'passive' node, i.e. 3+1, and save two servers compared to three sets of two-node clusters. In case two or more nodes fail at the same time, performance would suffer on a 3+1 cluster. As long as this is made clear to the business, it is a small(er) probability sceanrio, and it is clearly spelled out in the SLA, a 3+1 cluster may be attractive to the business, or whoever writes the check.

Linchi

"Flaxen" wrote:

What is the purpose of adding extra nodes past the 2 node cluster if clustering is just for failover? I read about adding extra nodes when researching and didnt see why that would be advantageous? In case multiple servers malfunction at the same time?

Ok, now that I know Clustering wont solve my problem, is there another method of doing database load balancing? Would have lets say 3 servers, each acting as a publisher/subscriber and then a 2-node cluster acting as a distributor work well in this situation? I feel in this solution, i would have 3 sql servers with the same info and then a high available distributor.

Then I assume I would need a way to make the 3 sql servers appear as one server somehow so the code only has to reference one sql server. Maybe round robin dns?

Am i making this too difficult? Is there an easier solution? I cant imagine I am the only one who needs a solution where you can scale Microsoft SQL servers effectively.

Re: Building high available and high performance web sites with MS

Thanks in advance.

"Geoff N. Hiten" wrote:

You are asking about two goals. First let me address the availability issue. Microsoft clustering is a failover technology where a SQL instance can move between host computers in case of hardware or systems failure. The basic single-instance, two-node cluster (sometimes incorrectly called Active/Passive) uses only one host at a time. You can add additional nodes (host computers) and instances, but the instances do not share information and are not "brick-type" scalable.

Right now, the way to scale SQL is to replace the host computer with a larger one. With failover clustering, you can do this with minimal downtime by replacing one node at a time.

—
Geoff N. Hiten
Senior Database Administrator
Microsoft SQL Server MVP

"Flaxen" <Flaxen@xxxxxxxxxxxxxxxxxxxxxxxxxxxx> wrote in message
news:7A300D7D-7E43-49DF-A3A7-9A134C55AA22@xxxxxxxxxxxxxxxxxxxxx

Hi all. My first time posting to this newsgroup, but I think I am in the right place. I am in the discovery stage of building a web site that we anticipate will have a large amount of traffic. Our design platform is Windows and Microsoft SQL and the goal is to have web sites and database servers that are not only high available but also scalable that I can put in a new server as needed to increase performance.

For the web sites, I know that I can use the NLB services of Windows 2003 standard edition. My lack of knowledge, though, is on the MS SQL side. I have been reading a lot of info on Microsoft SQL server failover, but from what I have read this typically entails having two MS SQL servers running in an active/passive mode.

Re: Building high available and high performance web sites with MS

Does Microsoft SQL Server support the ability to have multiple MS SQL servers using the same database and appearing as one physically database?

If one fails, the other servers pick up the load, and if we need to add a new server we simply install it into the cluster?

I understand that I will need enterprise edition to accomplish the goals above, but are these goals even doable?

I know I can setup a bunch of SQL servers all doing transaction replication, but that makes each server appear as its own database server and I feel is not the proper solution for this.

Hope my questions make sense and looking forward to the answers.