

# Re: High availability

---

*Source:*

<http://www.tech-archive.net/Archive/SQL-Server/microsoft.public.sqlserver.clustering/2007-07/msg00029.html>

---

- *From:* "Geoff N. Hiten" <[SQLCraftsman@xxxxxxxxxx](mailto:SQLCraftsman@xxxxxxxxxx)>
  - *Date:* Wed, 11 Jul 2007 13:39:43 -0400
- 

I agree with Andrew. Having someone who can help you down the clustering and high availability path is a small price compared to getting it wrong. The best consultants teach and guide while designing and building.

And yes, I am also a consultant, but I spent over fifteen years on the full-time side of the fence.

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

"Andrew J. Kelly" <[sqlmvpnoospam@xxxxxxxxxxxxxx](mailto:sqlmvpnoospam@xxxxxxxxxxxxxx)> wrote in message  
[news:eTrBJC%23wHHA.2432@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx](mailto:news:eTrBJC%23wHHA.2432@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx)

I would start here: <http://www.microsoft.com/sql/technologies/highavailability/default.msp>  
But if you are serious about a system of this size and want to do it right I would definitely recommend bringing in a good consultant to get you started on the right path. And this isn't just because I am a consultant:).

—  
Andrew J. Kelly SQL MVP

"Suri Nagarajan" <[SuriNagarajan@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx](mailto:SuriNagarajan@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx)> wrote in message  
[news:79FA8260-CDB5-4E99-AB1E-58DFF7008D6F@xxxxxxxxxxxxxxxxxxxxxx](mailto:news:79FA8260-CDB5-4E99-AB1E-58DFF7008D6F@xxxxxxxxxxxxxxxxxxxxxx)

Thanks Geoff and Andrew for the good info, I guess my best bet is to scale up rather than scale out and make sure the database is tuned right.

Is there any good book you guys might suggest which could walk me thru on setting up sql server failover clustering, replication, High availability, etc., ?

Suri.

"Andrew J. Kelly" wrote:

Re: High availability

Just to clarify that some. Geoff is 100% correct in that SQL Server does not have the implementation of multiple dbs such as Oracle Rac. But as I eluded to earlier there are two types of replication that you may have heard of that some people relate to this. This is Merge and bi-directional replication. Each allows multiple SQL Servers to have a full or partial copy of the db and has the ability to update the others with changes. But this is not typically a scale out solution for performance so I am not advocating it just making you aware of what SQL Server has to offer and hopefully avoid some confusion.

—  
Andrew J. Kelly SQL MVP

"Geoff N. Hiten" <SQLCraftsman@xxxxxxxx> wrote in message  
[news:%23q0Z6N8wHHA.3756@xxxxxxxxxxxxxxxxxxxxxxxxxxxxx](mailto:news:%23q0Z6N8wHHA.3756@xxxxxxxxxxxxxxxxxxxxxxxxxxxxx)

> There are no native multiple-master database  
> implementation > technologies  
> for SQL Server.

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

>  
>  
>  
>  
>  
>  
> "Suri Nagarajan"

<SuriNagarajan@xxxxxxxxxxxxxxxxxxxxxxxxxxxx> wrote in  
> message

[news:91A0A7A4-536D-45AF-8C7E-CCFAE8C43CB9@xxxxxxxxxxxxxxxxxxxxx](mailto:news:91A0A7A4-536D-45AF-8C7E-CCFAE8C43CB9@xxxxxxxxxxxxxxxxxxxxx)

>>I think you are right regarding scaling up, now I have to  
>>convince my

>>client  
>> to invest in bigger and better hardware for hosting the  
>>database, not >> one

>> but  
>> two or more of similar hardware for fail over server.

Thanks for the  
>> info,

Re: High availability

>> it was really useful.  
>>  
>> But just for my understanding – in SQL server is there an  
>> implementation  
>> like Oracle to have multiple master database in different  
>> servers >> with  
>> each  
>> database mimicking the same tables,structure and data  
>> which gets  
>> automatically synchronized among themselves, so that  
>> different users >> can  
>> connect to different servers and still be looking at the  
>> same data?  
>>  
>> "Andrew J. Kelly" wrote:  
>>  
>>> I have worked on many systems with thousands of users  
>>> on a single >>> server  
>>> with no problem given the right configuration. Both  
>>> scaling up and >>> out  
>>> have  
>>> pros and cons but unless the system is read only scaling  
>>> out is much  
>>> more  
>>> difficult than scaling up in most cases with SQL Server.  
>>> These days >>> you  
>>> can  
>>> scale up to a 64 processor system with 1TB of memory  
>>> so hitting the >>> max  
>>> performance limits are pretty hard to do if done  
>>> correctly.  
>>>  
>>> -- >>> Andrew J. Kelly SQL MVP  
>>>  
>>> "Suri Nagarajan"  
>>> <SuriNagarajan@xxxxxxxxxxxxxxxxxxxxxxxxxxxx> wrote in  
>>> message  
>>>  
>>> [news:5AA822BD-FD7A-47A5-8D01-9455D22C2149@xxxxxxxxxxxxxxxxxxxx](mailto:news:5AA822BD-FD7A-47A5-8D01-9455D22C2149@xxxxxxxxxxxxxxxxxxxx)  
>>> > Yes, I did think of scaling up as an option. I have been  
>>> dealing >>> > with  
>>> > DB2  
>>> > and Oracle based systems in the past , I recently took  
>>> over this >>> > SQL  
>>> > server  
>>> > based system, I am trying to understand/explore all  
>>> options >>> > available  
>>> > (since  
>>> > I am not sure about the options available in SQL  
>>> server >>> > environment).  
>>> >

Re: High availability

>>>> > Currently we are dealing with user environment with  
>>>> > couple of >>>> > hundred  
>>>> > users  
>>>> > which is not a problem, the application is running  
>>>> > smooth. But in >>>> > the  
>>>> > near  
>>>> > future this system has to be deployed in an  
>>>> > environment where >>>> > couple  
>>>> > of  
>>>> > thousand users will be accessing the system (huge  
>>>> > jump in volume >>>> > of  
>>>> > data  
>>>> > and  
>>>> > number of transactions).  
>>>> >  
>>>> > I feel if I scale up, we will be hitting the max  
>>>> > performance >>>> > limits  
>>>> > soon  
>>>> > due  
>>>> > to sudden increase in number of users. I think scaling  
>>>> > out will >>>> > be  
>>>> > more  
>>>> > flexible to add more power in the future if needed.  
Both High  
>>>> > availability  
>>>> > and performance are critical requirements since its an  
>>>> > on-line  
>>>> > transactional  
>>>> > system.  
>>>> >  
>>>> >  
>>>> >  
>>>> > "Andrew J. Kelly" wrote:  
>>>> >  
>>>> >> FYI, Replication can be used for non-read only  
>>>> >> servers but is >>>> >> usually  
>>>> >> not  
>>>> >> a  
>>>> >> good choice for scaling out in order to get  
>>>> >> performance gains. >>>> >> You  
>>>> >> need  
>>>> >> to  
>>>> >> separate the two requirements (Performance & High  
>>>> >> availability)  
>>>> >> somewhat  
>>>> >> as  
>>>> >> they are typically two completely different solutions  
>>>> >> or >>>> >> approaches.  
>>>> >> What  
>>>> >> is wrong with scaling up? How large is the server  
>>>> >> now and what >>>> >> do

Re: High availability

>>> >> you  
>>> >> need  
>>> >> to support? And are you sure it has been tuned properly? You >>> >> might  
>>> >> not  
>>> >> even  
>>> >> have the need to scale up or out.  
>>> >>  
>>> >> -- >>> >> Andrew J. Kelly SQL MVP  
>>> >>  
>>> >> "Suri Nagarajan" <Suri  
Nagarajan@xxxxxxxxxxxxxxxxxxxxxxxxxxxx> wrote >>> >>  
in  
>>> >> message  
>>> >>  
news:BEFF51E6-28E0-4784-A24C-0DBB2C4B497F@xxxxxxxxxxxxxxxxxxxx  
>>> >> >I am new to SQL server, I would appreciate any  
help on following  
>>> >> >question.  
>>> >> >I  
>>> >> > tried researching and found few options (detailed  
below after >>> >> > the  
>>> >> > question)  
>>> >> > which didn't really help what I am trying to do.  
Any suggestion >>> >> > for  
>>> >> > the  
>>> >> > Gurus  
>>> >> > will be helpful.  
>>> >> >  
>>> >> > I am trying to implement a high availability system  
which will >>> >> > have  
>>> >> > very  
>>> >> > high database access traffic – potentially couple of  
thousand >>> >> > users  
>>> >> > trying  
>>> >> > to  
>>> >> > access a transaction system (read and/or update by  
most users).  
>>> >> > Unfortunately this is not a new system, its an  
existing system >>> >> > with  
>>> >> > around  
>>> >> > 200 tables implemented in a single server (Using  
SQL server as  
>>> >> > database),  
>>> >> > due to increase in number of users and traffic I  
need to find >>> >> > ways  
>>> >> > to  
>>> >> > increase availability and performance.  
>>> >> >  
>>> >> > My final system should be implemented on  
multiple servers and >>> >> > each



Re: High availability

>>>  
>