Re: SQL 2005 servers with multiple instances

Source:
http://www.tech−archive.net/Archive/SQL−Server/microsoft.public.sqlserver.setup/2006−03/msg00223.html

- *From*: "Geoff N. Hiten" <SQLCraftsman@xxxxxxxxx>
- *Date*: Fri, 17 Mar 2006 11:11:37 −0500

I have architected similar solutions where the failover instance had a minimal footprint while it was just maintaining synch. I found the ability to partition resources very beneficial.

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

"Chris Williamson" <cwilliamson13@xxxxxxxxxxxxxxxx> wrote in message
news:%23sTJIWYSGHA.4740@xxxxxxxxxxxxxxxxxxxxxx

> Well, log shipping is the current plan. This brings me back to the instances... I understand running two instances uses more resources than just one, given the same number of user databases. But given that one instance is solely for receiving transaction logs, does the ability to limit memory and set affinity outweigh any additional overhead? Or should I just forgo that and just use the one instance?
>
> Thank you, I appreciate your responses...
>
> Chris
>
> "Geoff N. Hiten" <SQLCraftsman@xxxxxxxxx> wrote in message
> news:%23$DplsTSGHA.5116@xxxxxxxxxxxxxxxxxxxxxx
>
> > Reverse order answers.
> >
> > The systems I was talking about have mirrored controllers with redundant cache. I don't recommend junk. :)
> >
> > As for your situation, I would use log shipping unless I had a very large number of databases. Then I would look for a third−party synchronization solution.

GNH

"Chris Williamson" <cwilliamson13@xxxxxxxxxxxxxxxxx> wrote in message
news:ue%23qvJTSGHA.4976@xxxxxxxxxxxxxxxxxxxxxxx

> You know, I didn't realize SQL Standard supported
> Fail−Over clustering.
> Unfortunately, we've already purchased our non−cluster hardware (one new
> server, one old server, many new drives) and we spec'ed it for SQL2000's
> features. Now that we're implementing, we've decided to use SQL 2005.
> Our plan is to use log shipping until database mirroring becomes
> available for production.
>
> One of our goes is to have two complete sets of data. Unless something
> has changed since I worked with it, clustering only has one copy of the
> data, shared between the servers. A hardware failure ont eh quarum
> could cause disaster on both servers. This happened to our Exchange
> 2000 cluster. We had to restore from tape because the quarum controller
> locked up, losing all data in its hardware write cache.
>
> Thank you,
>
> Chris
>
> "Geoff N. Hiten" <SQLCraftsman@xxxxxxxxx> wrote in message
> news:OK4q$3RSGHA.1572@xxxxxxxxxxxxxxxxxxxxxxx
>
> > Have you checked out the latest possible
> > hardware configurations for
> > SQL 2005 clustering?
> >
> > I have built two−node clusters with
> > dual−proc dual core brand name
> > servers for between $35K and $40K,
> > including all licensing.. Similar
> > quad−proc systems go for around $100K,
> > again including all licenses.

Clustering is no longer restricted to big
dollar implementations.

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

"Chris Williamson"
<cwilliamson13@xxxxxxxxxxxxxxxxx>
wrote in message
news:O1Kgb7LSGHA.5156@xxxxxxxxxxxxxxxxxxxxxxx

> I have an interesting
> configuration that I was
> hoping others might
> comment on. We have 2
> SQL 2005 standard servers
> that need to provide
> redundancy for each other,
> without using clustering.
> We'll be
> replicating the databases
> from one server to the other,
> and vice−versa.
>
> Our big question is, should
> we use multiple instances?
> The knee−jerk
> reaction is to not use
> instances because of the
> extra overhead and the
> inefficient sharing of
> memory between the
> instances. However, I have
> a few reasons why it would
> be beneficial...
>
> First, consider the two
> servers with many small
> drives for the 'Live'
> data (4 logs, 7 data all 15k),
> but a small number of large
> drives for
> the 'Redundant' data (4 for
> Data and logs, only 10k).
> The same RAID
> controller is being shared
> for the Live data and the

Redundant
data/logs. Also, we're using
processor licensing, so the
additional
instances won't cost any
more.

My concern is the load
affecting the Live databases
while applying
transaction logs to the
Redundant databases, which
would be done at
least every 30 minutes. To
my knowledge, SQL
provides no
prioritization, other than
setting proc affinity and
boosting
priority.

I'm proposing that we use
two instances on each server
− Live and
Redundant. Server1\Live
would replicate to
Server2\Redundant, and
Server2\Live would
replicated to
Server1\Redundant. The
Backup
instances would be limited
to 256MB memory and have
its cpu & io
affinity set to one of the
dual hyperthreaded
processors. This would
ensure the Live instance
always takes precedence
over the backup
instance. Even if everything
is at 100%, theoretically,
Live would
have 87% of the resources
(3 processors + 1/2 of the
fourth). The
backup instance wouldn't be
doing anything except
restoring logs, so
there shouldn't be much loss
due to overhead.

Obviously, during a
(manual) failover, the
memory and affinity would
be adjusted. Not as fancy
Clustering, but doesn't have
the price tag
either.

I totally understand the extra
overhead if dealing with two
instances
that are very active. But in
this situation, I don't think
there'd be
a problem. Does anyone
have any comments or
helpful hints with our
configuration?

Thank you,

Chris

.