

# Re: Is there a best practise to implement a warm standby-solution

---

*Source:*

<http://www.tech-archive.net/Archive/Windows/microsoft.public.windows.server.clustering/2007-07/msg00187.html>

---

- *From:* "Edwin vMierlo [MVP]" <EdwinvMierlo@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx>
  - *Date:* Thu, 26 Jul 2007 17:02:07 +0100
- 

I don't think that MSCS is the right platform for this.  
Doing all this on a failover cluster is just adding complexity.

you are talking about some sort of "Application Failover" from within an Application, monitored and decision making by a third app (your app Y)... I fail to see why you cannot run this on standalone hosts.

Also, the below does not take care of "state", you need to code that into your appX

"RonC" <RonC@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx> wrote in message  
<news:F658334A-165A-4B3A-8996-338763D666BA@xxxxxxxxxxxxxxxxxxxx>

Thanks all, I do appreciate your attention:

I suspected that some of what I need is not provided by MSCS. My concern is

"am I trying to do something that is beyond support". I'll try to articulate

the solution in a way that focuses on my areas of concern:

- 1) In a two node cluster I am planning to run an instance of the same application (ApplicationX) on each node. ApplicationX is cluster aware.

Each

instance will be managed by a separate resource group. These resource groups

will be configured to not failover (i.e. only run on one node). Is this

Re: Is there a best practise to implement a warm standby-solution

type

of configuration supported?

2) In the same two node cluster, I am planning to run one instance of an application (ApplicationY, different from the above). ApplicationY will be cluster aware. Using the eventing mechanism in MSCS, ApplicationY will listen to cluster events initiated from the resource groups managing ApplicationX instances. ApplicationY will be managed by a third resource group. This resource group will failover, between the two nodes. The

failover

criteria will be based on the online/offline status of ApplicationX

instances

collect through MSCS events. Is this type of implementation supported?

My plan is to have ApplicationY tell the local instance of ApplicationX to be the active one. Note even though both instances of X are online, only

one

can be "active".

ApplicationX is not stateless so NLB won't fly.

I need both instances of ApplicationX past initialization to reduce the failover time.

I don't want to fail to an instance of ApplicationX if it's not healthy.

"Chuck [MSFT]" wrote:

We still do not failover 'state' information.

--

Chuck Timon, Jr.

Microsoft Corporation

Windows Server 2008 Readiness Team

This posting is provided 'AS IS' with no warranties, and confers no

rights.

"RonC" <RonC@xxxxxxxxxxxxxxxxxxxxxxxxxxxx> wrote in message  
[news:828842FC-A0BA-428C-9C44-9F68CBC2EC46@xxxxxxxxxxxxxxxxxxxx](mailto:news:828842FC-A0BA-428C-9C44-9F68CBC2EC46@xxxxxxxxxxxxxxxxxxxx)

Hi Chuck. Thanks, but I realize there will be a disruption to the

Re: Is there a best practise to implement a warm standby-solution

Re: Is there a best practise to implement a warm standby–solution

clients

during failover.

All I want to do is reduce the failover time. I can achieve this with

a

second instance of the application running in a warm state.

The

application

is complex with multiple services. If a service fails in the

active

instance,

I want to know I am failing to an instance of the application

that is

in

good

health (having a better service level than the failed from side).

Failing

to

the warm instance would only involve a failover of an I P address.

Thanks again for your response.

Hi Chuck. Thanks, but I realize there will be a disruption to the

clients

during failover.

"Chuck [MSFT]" wrote:

Microsoft Clustering services cannot provide what you need. We

provide

'high' availability, not '100%' availability.

Failing over

application

Re: Is there a best practise to implement a warm standby–solution

Re: Is there a best practise to implement a warm standby-solution

'state' is not possible. We disconnect clients during a failover and

the

clients must be able to reconnect. If your applications use

Distributed

Transactions, then the ACID process should help deal with that.

--

Chuck Timon, Jr.  
Microsoft Corporation  
Windows Server 2008 Readiness Team  
This posting is provided 'AS IS" with no warranties, and confers no rights.

"RonC"

<RonC@xxxxxxxxxxxxxxxxxxxxxxxxxxxx>

wrote in message

[news:7922DEA4-AE78-4726-819C-29DC3A2B6CCB@xxxxxxxxxxxxxxxxxxxx](mailto:news:7922DEA4-AE78-4726-819C-29DC3A2B6CCB@xxxxxxxxxxxxxxxxxxxx)

Ryan, thanks for your response.

My application is not a problem. Rather I am looking for a best practice from a MSCS perspective to achieve the following:

1) The ability to monitor the state of multiple instances of the

same

application, while only one instance is in an active mode.

(Probably

using multiple resource groups).

Re: Is there a best practise to implement a warm standby-solution

Re: Is there a best practise to implement a warm standby–solution

2) The ability to failover active state. The ability to monitor the state of all application instances and make failover decisions based on application state. (Probably using another resource group and custom code with

the

MSCS eventing mechanism).

So my questions:

Is there a best practice or out of the box mechanism to achieve

some /

all of the above.

If a custom solution is required, do you think I am moving towards

the

land of unintended / unsupported.

"Ryan Hanisco" wrote:

Hi RonC,

This will really depend on the application.

In most cluster–aware apps,

Re: Is there a best practise to implement a warm standby–solution

all  
nodes stay  
running  
with the app  
ready to go.  
On failure,  
the

other

takes  
over with  
only the  
time for the  
failure  
detect and  
session

reconnect.

If the app is  
not built for  
this, your  
options may  
be far  
fewer.

--

Ryan  
Hanisco  
MCSE,  
MCTS:  
SQL 2005,  
Project+  
Chicago, IL

Remember:  
Marking  
helpful  
answers  
helps  
everyone  
find the info

they

need  
quickly.

"RonC"

Re: Is there a best practise to implement a warm standby–solution

Re: Is there a best practise to implement a warm standby–solution

wrote:

Hello:

I  
am  
planning  
to  
integrate  
an  
existing  
application  
with  
MSCS.  
However,  
in  
a  
failover  
scenario  
a  
full  
application  
startup  
will  
take  
too  
long.  
Therefore  
I  
am  
planning  
to  
implement  
a  
warm  
standby  
instance  
of  
the

application.

I  
would  
like  
that  
status  
of  
the  
warm

Re: Is there a best practise to implement a warm standby–solution

Re: Is there a best practise to implement a warm standby-solution

standby  
instance  
to  
be  
a

consideration

in  
a  
failover  
decision.

Do  
you  
have  
any  
suggestions  
on  
how  
I  
might  
accomplish  
this  
with

MSCS

as  
opposed  
to  
building  
a  
custom  
cluster  
aware  
component?