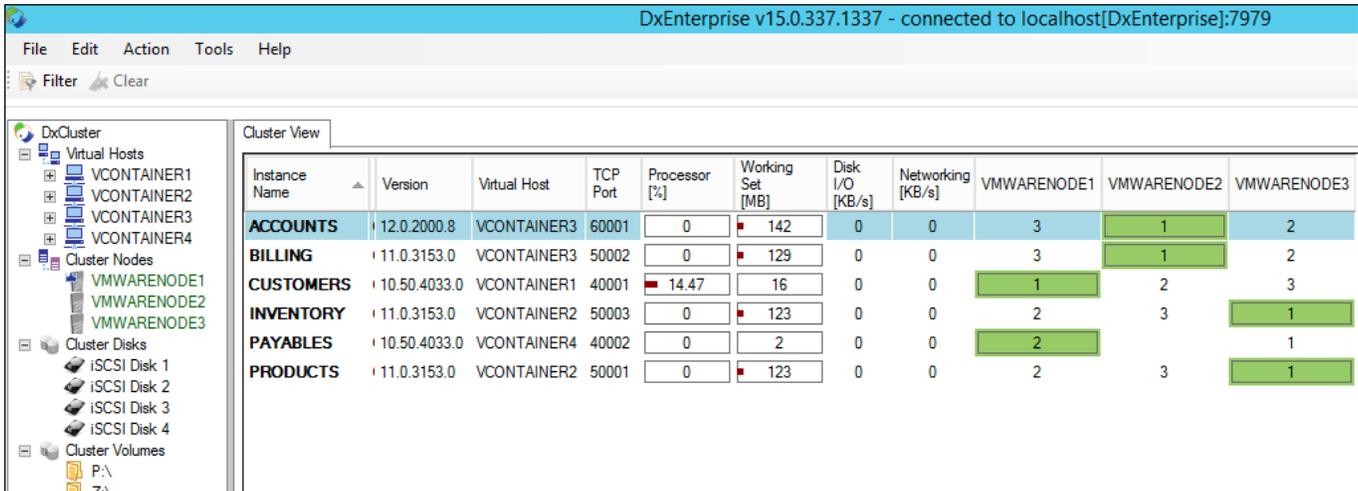# DH2i DxEnterprise & Microsoft SQL Server

A Hands-On Review of DH2i's High Availability & Multi-Subnet Disaster Recovery Software for Windows

by Denny Cherry, Consultant & MVP, Denny Cherry & Associates

DxEnterprise: High Availability Software for Windows

DxEnterprise is high availability software for Windows from DH2i that can be used to solve a range of well-documented problems in the Microsoft SQL Server world. Not every feature was tested during the process of putting together this whitepaper; however the features that were reviewed lead to the scenarios that are described. I'll review how to use DxEnterprise to create simple-to-manage, simple-to-deploy, high availability and disaster recovery solutions. In the process, I will show how it supports datacenter orchestration, letting you quickly and easily migrate SQL Server instances from one hardware platform to another with little downtime, while upgrading the operating system (OS), SQL Server, or server infrastructure. DxEnterprise can also be used to balance workloads of multiple SQL Server instances across hardware resources, either automatically or manually. Overall, based on what I've seen, I expect that DxEnterprise can significantly help with datacenter costs and uptime.

DxEnterprise for SQL Server Cluster View

DxEnterprise for SQL Server High Availability & Multi-Subnet Disaster Recovery

DxEnterprise expands on the native Microsoft SQL Server high availability configuration options by introducing instance-level mobility with its Virtual Host (Vhost) technology. While Microsoft's native high availability options require a very strict configuration, such as the same operating system for all members of the Windows cluster, DxEnterprise has no such requirement. Nodes of the cluster can run a variety of operating systems spanning any mix of physical, virtual, or cloud hosts. This allows for an easy upgrade path for the SQL Server, Windows, or infrastructure version that is running within the cluster.

DH2i's InstanceMobility concept means that only the IP address, network name, and storage move from one Windows Server to another—not the entire operating system configuration. The Vhost lets SQL Server instances run wherever they're moved to, on the active host's local OS and local application installation. This allows for a lightweight, quick failover process that moves with the speed of an application restart. This has the additional benefit of allowing for operating system upgrades where the native Windows Server Failover Clustering (WSFC) does not. WSFC's lack of flexibility locks a lot of applications onto a specific version of Windows Server as well as SQL Server, as the cost and process to upgrade a failover cluster is simply too high for most applications to justify.

DxEnterprise also removes the licensing limitations that Microsoft has put in place for Windows clusters running Microsoft SQL Server. Using the native features, SQL Server supports only a two-node cluster when using Microsoft SQL Server Standard edition, requiring Enterprise edition for any clusters that will have three or more nodes in the cluster. DxEnterprise allows you to use any edition of Microsoft SQL Server from Express up to Enterprise, while still having the same level of high availability within your configuration.

	DxEnterprise	Windows Clustering
Failover Clustering	Yes	Yes
Mixed-OS Clusters	Yes	No
Mixed Physical, Virtual, and Cloud Hosts	Yes	No
Easy OS and/or SQL Server Upgrades	Yes	No
>2 Node Clusters on Standard Edition	Yes	No
Multi-Subnet Clustering on Any Version/Edition	Yes	No

With DxEnterprise, disaster recovery becomes even easier for older versions of SQL Server because multi-subnet failover configurations are possible on stretch clusters for all versions and editions of SQL Server. Using the native failover clustering features, multi-subnet failover is only available in SQL Server 2012 and newer and is an Enterprise edition feature in SQL Server 2012. Multi-subnet failover is supported in all versions and editions of Microsoft SQL Server because DxEnterprise uses standalone SQL Server instances. As far as SQL Server is concerned, it is just an IP address change upon SQL Server restart. This may require careful configuration of any additional TCP endpoints, which are configured within the instance, but that is the only caveat.

DxEnterprise for Orchestration, Hardware Migrations & Upgrades

Moving a SQL Server instance from one hardware platform to another is often a complex task, especially if you want to attempt an operating system upgrade at the same time. Worse than planning the migration is planning the rollback if there is a problem. This becomes very complex as you have to potentially address renaming servers, turning servers back on, changing IP addresses, all while dealing with a management team and business unit that want the system back up and running now. Using DxEnterprise, we can migrate a SQL Server instance using just seven steps:

1. Install DxEnterprise
2. Move databases to shared storage (if needed)
3. Rename physical server and change its IP address
4. Configure DxEnterprise to use the old name and IP address
5. Install DxEnterprise on the new server and present the same SAN storage to the new server
6. Install SQL Server on the new server
7. When ready, use DxEnterprise to switch the SQL instance from running on the old server to the new server

We start by installing the DxEnterprise software on our existing server, which is already running our production workload. If the databases are currently on shared storage (such as SAN storage) then the databases don't need to be moved. If they are on local storage, they'll need to be moved to SAN storage or some other shared storage.

As we want to keep the network name and IP address that we are currently using for our applications, we need to take a short outage—basically just longer than it takes to reboot our current production server to change its network name and IP address. After the server comes back online, we configure DxEnterprise to use that network name and IP address. (Alternatively, we can quickly create a temporary staging host and move the instance there. Once the original source is turned off, we create a host with the old source IP and name, and rehost from the temporary staging host to the original source information. It's simple, quick, and requires no outage to change host name.) Either way, DxEnterprise is now managing the SQL Server instance startup. We restart the SQL Server instance and all of our users are now connected using the same name and IP address.

All of these steps so far can be done at any time—even months prior to the actual migration.

When the new server is built, patched, and ready for use, we install DxEnterprise and present the shared storage to the server. We then install SQL Server as normal, and place that SQL Server instance under the control of DxEnterprise. On the night of the migration, we simply use DxEnterprise to failover from our old production server to our new production server.

As DxEnterprise does not require that the operating systems on both servers are the same, we can rehost the workload to the node with a newer version of the OS as a part of the failover. If there's a problem, we can simply fail back. Because DxEnterprise allows for different operating systems to be running within the same cluster, if there is a problem we can roll back in seconds, hours, or even days later if needed.

DxEnterprise allows for additional control logic when services are started and stopped; for instance, to start additional software on the node of the cluster running the SQL Server instance when the SQL Server instance starts.
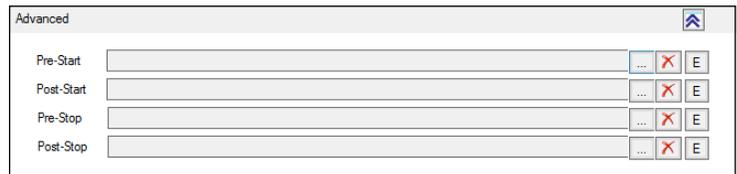

Logic Insertion Points for a SQL Server Instance

DxEnterprise to Balance Workloads Across Servers

One of the biggest problems with running multiple SQL Server instances on servers is how to move one of those instances to another server if an instance starts to use too much CPU or memory resources. While all of the native tools within SQL Server are great for monitoring and alerting, there's nothing in the native failover clustering stack that allows you to automatically move instances from node to node of a SQL Server cluster automatically in order to solve a noisy neighbor problem. DxEnterprise solves this for you.

When a multi-instance high availability cluster is configured using DxEnterprise instead of using the native failover clustering technologies within Microsoft Windows, you can configure policies that tell the DxEnterprise software to move an instance from one node of the cluster to another.

As an example, let's look at a four-node cluster running 10 instances of SQL Server on a cluster in which one instance is the data warehouse while the other nine are OLTP applications. You could configure a rule so that when the CPU on the data warehouse goes over 50% of the node's abilities you would move the other instances off that node. DxEnterprise allows you to set policies such as moving lower-priority workloads off the node, load-balancing nodes, or moving the affected workload to another host.
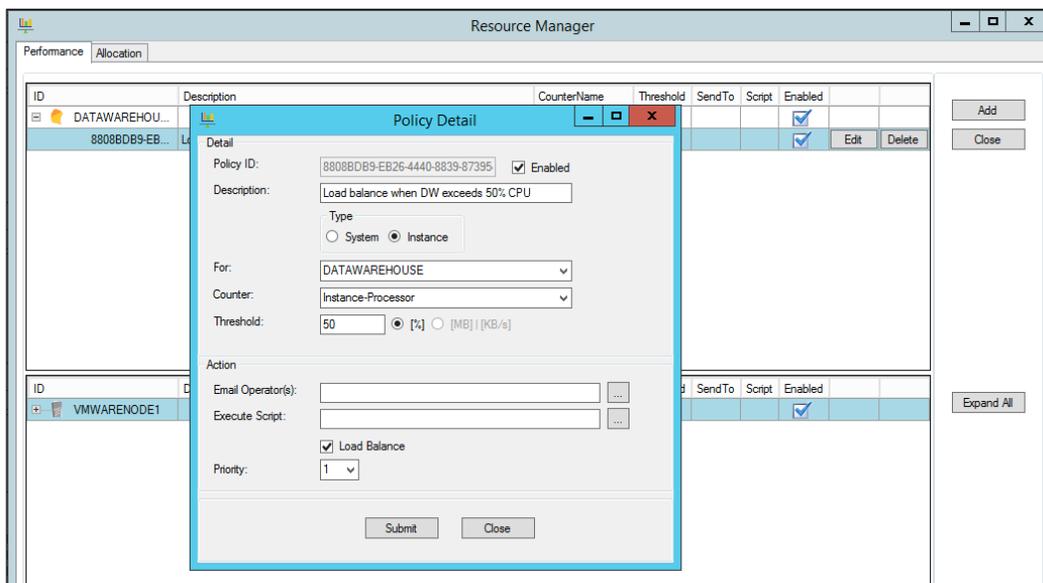
SQL Server Instance Policy for CPU

Because the SQL Server instances are not tied to any specific server environment, it becomes quite easy to determine if a workload's performance problems are OS-specific, since the workload can be moved from server to server without issue. This can assist with performance tuning, OS troubleshooting, configuration issues, etc.

DxEnterprise to Reduce Costs & Increase Uptime

Between Microsoft's rising software licensing costs and the continued push within IT organizations to do more with less, the ability to implement a high availability strategy is becoming harder and harder for companies to realize. With DxEnterprise, the cost of Microsoft SQL Server licenses can be reduced, the cost of the Windows infrastructure can be reduced, and uptime can be increased; all with the ability to reuse existing infrastructure and without the need to have specialized training in Microsoft's high availability technologies. With DxEnterprise, the costs and troubles of upgrading a cluster—or even a single node—disappear. In a native Windows clustering configuration, the only upgrade option for the Windows operating system is to build a new cluster, then go through the complex and expensive process of migrating the database instances to the new cluster. With DxEnterprise, each node of the cluster can be upgraded in place without rebuilding the cluster. This removes the need for extensive downtime while applications are migrated, and effectively removes the barriers to doing operating system upgrades of your clustered environments.

DxEnterprise also lets you configure a more dynamic cluster. This not only keeps our applications running thanks to the high availability technologies built into the product, but we can also keep applications healthy by monitoring the application's resource usage and proactively moving resources from one node of the cluster to another as needed. This dynamic load balancing allows our applications to grow and use the resources that they need when they need them, without adversely impacting other applications within the environment.

Conclusion

Throughout this paper we've reviewed some of the features that DxEnterprise makes available, including more flexible high availability solutions than the native Windows Server Failover Clustering solutions. Additionally, DxEnterprise provides easy-to-implement system migration options, all while helping to reduce SQL Server costs and increasing system uptime.

DH2i has also made DxEnterprise available as a free trial for users to test in their own datacenter or in DH2i's remote lab. You can sign up for a free trial at: http://dh2i.com/trial/