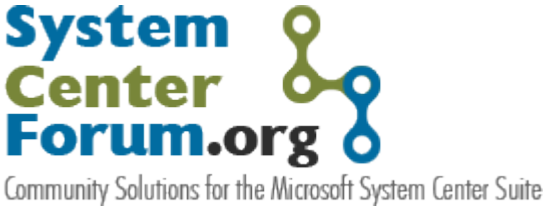


Deploying OpsMgr in Highly Available and Distributed Enterprise Environments

Guidance on planning for large, complex deployments of OpsMgr

Authors:
Andy Dominey, MVP – Operations Manager

Version: 1.0
December 2008

Contents

Purpose of this document

This document is designed to clarify the finer points of deploying OpsMgr into complex large environments. It seeks to document the requirements and necessary steps appropriate to these types of deployment scenarios and also aims to offer some design advice with regards to planning for an OpsMgr deployment in this scenario.

It will look in detail at the following:

- The use of gateway servers to span long distance WAN links
- The use of high availability including clustering and SQL database mirroring/log shipping.
- Business continuity as it relates to OpsMgr

This document is intended to assist with the planning of a complex and/or highly available OpsMgr infrastructure.

Introduction

Microsoft System Center Operations Manager 2007 (OpsMgr) is designed to monitor and manage Windows server estates ranging from the small (<100 servers) to the enterprise level (5000+ servers). With that fact in mind, the product is very scalable and can be deployed to very large and complex environments up to a maximum supported 6,000 server agents per management group (although management groups designed for 10,000 exist). However, whilst the product works well out of the box, there are a number of points to consider and a number of steps that must be taken to maximise the effectiveness of OpsMgr in large and complex environments.

The information on the points to consider and steps to take exists today but is somewhat fragmented and difficult to locate. This document seeks to bring together and rationalize the information that exists in various documents today (listed in the references section) to enable the reader to be confident in designing and implementing OpsMgr into some of the larger and more complex environments.

With that in mind, a number of key items will be addressed in this document including planning for a large enterprise deployment of OpsMgr, building a highly available solution and utilizing OpsMgr features to monitor truly global organizations.

Please be aware, however, that whilst this document seeks to simplify the planning and deployment of OpsMgr into large and complex environments, it should not be considered a substitute for experience. Therefore, if you are planning a large deployment and you do not have a great deal of experience with the product or with the technologies involved, it is always the recommendation of 1E and of the author to seek professional assistance.

Planning for a Successful OpsMgr Deployment

The most important aspect of any deployment is planning. This point cannot be stressed enough. The saying goes “Fail to prepare, prepare to fail” and this is absolutely true of an OpsMgr deployment and especially when you are preparing to deploy in a large and complex environment.

Before you can begin to architect the solution, there are a number of key factors to consider. These include:

- Business reasons for deploying/migrating to OpsMgr
- Business requirements
- Technical (functional) requirements
- Existing infrastructure configuration
- Available project resources
- Available budget
- Level of product skill/experience for both OpsMgr and the products in the current environment

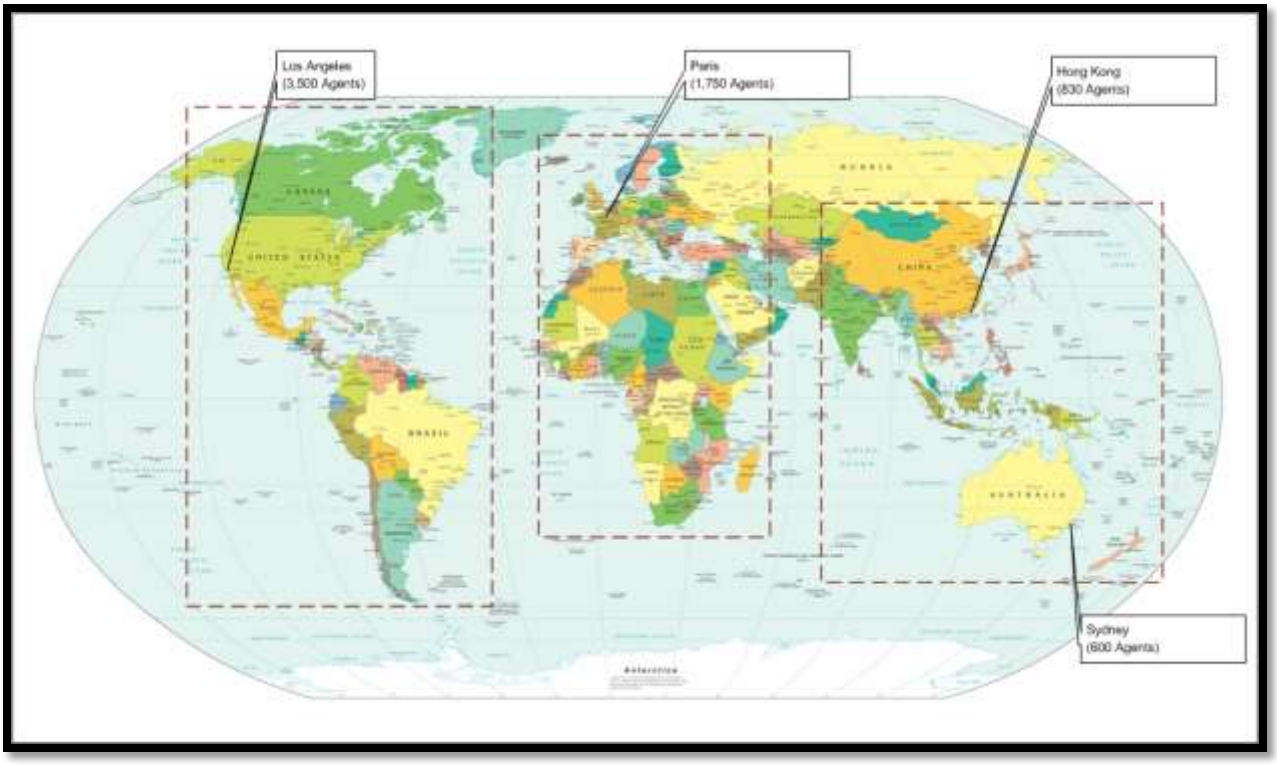
There may be additional items to consider in your environment. Each one of these items will have a significant impact on the time and cost of the deployment and also on the day to day management of the infrastructure. These items will not be examined in this document except to highlight the fact that they must be considered carefully. If any one of these factors is not considered, the project may become significantly more difficult.

Architecting the Solution

When considering the architecture for OpsMgr, the key factors will be as follows:

- Number of agents to monitor
- Applications to be monitored (which can also be defined as the management packs required)
- Physical and logical location of agents
- Any other additional components such as non-Windows systems network devices
- Extra features such as Audit Collection.

In this document, we will assume that we are dealing with a large number of agents (4000+) and that the organization is globally dispersed (i.e. devices to be monitored are geographically distributed across multiple continents/time zones). However, the principles laid down herein may be applied to deployments of any size and simply scaled up/down as appropriate. In addition, this document will not serve to offer any design recommendations, it will simply document the options available when planning a deployment of this size and type. Our assumed environment is shown below:



Before specifics can be decided such as number and location of servers, it is necessary to plan the basic architecture, devoid of any hardware specifications. When architecting an OpsMgr environment for a large, distributed company they will usually have two inherent requirements, **Consistent Global Monitoring** (that is monitoring all servers globally from a single pane of glass) and **High Availability/Disaster Recovery**. They may also be particularly interested in **Business Continuity** i.e. the process of recovering the business (not just the IT infrastructure) from a major failure or disaster with minimal loss of data and revenue. These requirements are by no means the only items that will be defined but are, from 1E's experience, common to the majority or enterprise environments.

Consistent Global Monitoring

Many large enterprises maintain an infrastructure that consists of a number of substantial offices, often around a hub site, but each remote office will often host a significant number of devices to be monitored. Often, these remote offices can contain more than 500 agents and are located at the end of long distance WAN links, sometimes even over intercontinental distances. In this case, bandwidth usually isn't a concern since even long distance WAN links tend to be large pipes of several Mb, but as we will document later, this is of limited concern.

In my experience, these companies often require monitoring to encompass all devices globally presenting them in a single pane of glass. With that in mind, we are considering the single management group design only in this document. In this type of infrastructure, the usual approach of adding additional management servers to service agents in different locations may not work as expected. Let's look at this in more detail.

Management Servers and Long Distance WAN Links

In a traditional OpsMgr deployment, agents report to management servers which update the OpsMgr database and sync with the Root Management Server. This is usually a perfectly valid way of structuring a management group. Unfortunately, when we are considering intercontinental distance WAN links, this is no longer a viable option. In order to understand this, we must look at the way a management server transfer's data to and from the database.

A traditional management server uses a simple SQL OLEDB connection to connect the database. This connection works (in simple terms) in the following way:

1. Connection establish message sent from management server to database server
2. Message received by database server and request for credentials sent to management server
3. Credentials sent from management server to database server
4. Credentials verified
5. Connection established
6. Data transmitted
7. Connection terminated (the common term for this is 'torn down' meaning that no part of the connection is left intact).

This method is acceptable for instances when a management server is on the same local network to the database server but is not appropriate when transferring data over long distance WAN links. This is due not to technology but to physics, - the physics of 'Speed of Light'. As we can see from the explanation of the SQL connection above, the connection is very active or 'chatty'. This is fine over a good, local LAN link but when we have data transferring in this way over a long distance WAN link, each step of this process is affected by latency caused by speed of light delay. The amount of latency which will cause alert delays will depend on many factors including number of alerts, frequency of alerts and bandwidth on the WAN link. However, from my experience, latency of more than 50ms could potentially cause delays. This further supports Microsoft's recommendation that a management server should **never** be located across a WAN link from the operational database. Table **1.1** shows how speed of light can affect the latency of a connection across a long distance WAN link.

Source – Destination	Distance (approx)	Roundtrip Latency (approx)[1]
Hong Kong – New York	16,500km	165ms
London – New York	5,500km	55ms

Table 1 - Illustrating data transfer across long distance WAN links

[1]Latency across fiber is calculated as follows:

- The distance from Hong Kong to New York is 16,500km.
- The speed of light in vacuum is 300 x 10^6 m/s.

- The speed of light in fibre is roughly 66% of the speed of light in vacuum.
- The speed of light in fibre is 300 x 10^6 m/s * 0.66 = 200 x 10^6 m/s.
- The one-way delay from Hong Kong to New York is 16,500 km / 200 x 10^6 m/s = 82.5ms.
- The round-trip delay is 165ms.

Note: The latency figure does not take into account latency caused by signal boosting and data conversion from Ethernet (electrical) signal to fibre (optical) signal.

This additional latency on the connection has been known to manifest as severe delays in the updating of alert and health service data.

To address this challenge, the gateway server role may be utilized in place of management servers in the remote locations. Gateway servers service agents in the same way as a management server but they differ in the way in which they send data to the database. Unlike management servers, gateways do not directly connect to the database. Instead, they forward (or proxy) data to another management server for transmission to the database. Unlike the very active OLEDB connection used by the management servers, gateways establish a TCP connection with the assigned management server and maintain this connection in an open state until either the connection fails with an error or the management server is no longer available. In addition, data sent from the gateway server is consolidated, resulting in a 20-30% reduction of total data sent to the database when compared to individual agents reporting directly to a management server across the WAN link.

With these factors in mind, the gateway offers a much more efficient method of monitoring large numbers of agents at the end of a long distance WAN link.

According to the OpsMgr Infrastructure Design and Planning Solution Accelerator, here are three key use case scenarios for the Gateway server role:

1. When agents located across trust boundaries necessitate administrative overhead because certificate authentication is required, requiring potentially substantial manual configuration.
2. Agents behind a firewall require multiple “allow” rules to permit agent traffic to pass through the firewall, raising potential security concerns.
3. Agents located across WAN links consume network bandwidth, potentially affecting service delivery to and from the remote location, and network latency between the management server and Operational database is probable.

It’s this third use case scenario that will be the focus in this document.

Using Gateways across WAN links to monitor large numbers of agents

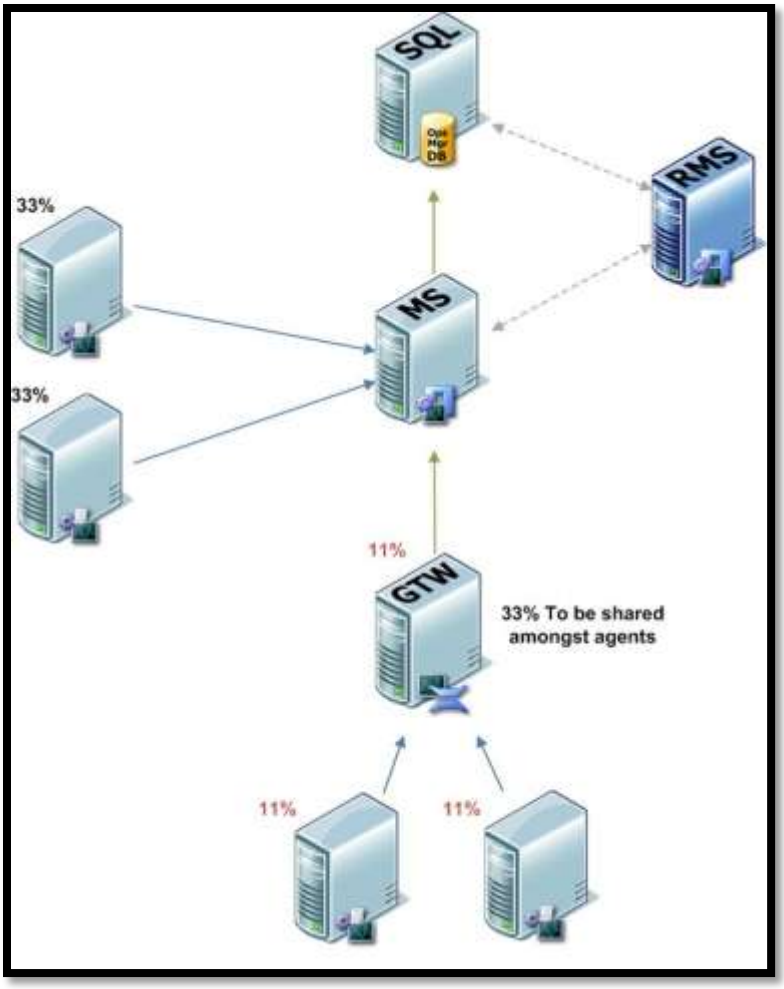
Now that we have established the advantages of using a gateway server to monitor agents across a long-distance WAN link, we’ll look at the impacts of making such a decision.

There are a number of factors to consider when using a gateway for any other purpose than to monitor agents in a DMZ. Typically when monitoring DMZ’s, the number of machines to be monitored will be

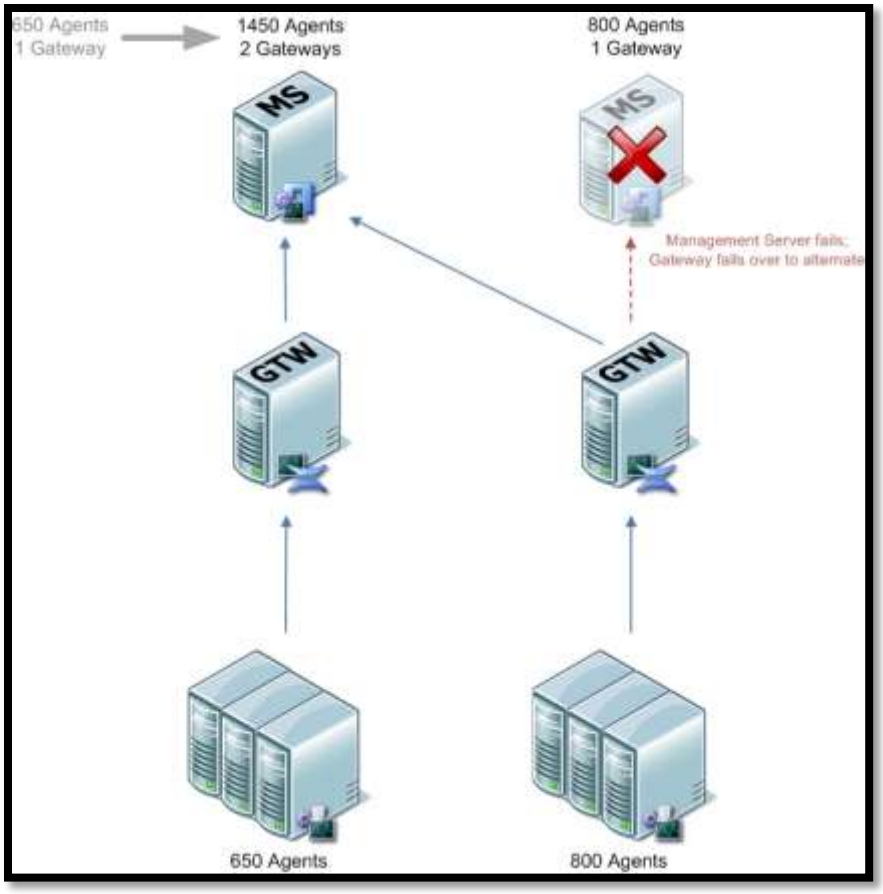
relatively small in comparison to monitoring the bulk of the corporate network. With that in mind, gateways have, by design been engineered to monitor less agents. Prior to OpsMgr Service Pack 1 (SP1), gateway serves were only able to accept in the region of 200 agents making them unsuitable for any other purpose than monitoring DMZ's and small un-trusted networks. However, with the addition of SP1, gateways have now been tested to 800 agents and, depending on the hardware configuration and WAN link specifics, could theoretically support many more than this. Compare this to the 2000 supported agents on a management server and the importance of a well designed architecture becomes apparent. To recap, the following list is a record of the design rules/best practises for OpsMgr:

1. No more than 2000 agents to a Management Server (MS)
2. No more than 800 agents to a Gateway (GTW)
3. No more than 5 Gateways to an upstream Management Server
4. Management Servers for Gateways should be dedicated
5. No agents to RMS if dedicated MS(s) exist
6. RMS and MS close to active db (LAN connectivity)
7. Agents and GTW in remote locations report to MS close to active db

In addition to the limited number of agents supported on a gateway when compared to a management server, another consideration is the fact that the gateway needs to forward its data to a management server for relaying to the database. This is where the second factor comes in. Whenever you connect a gateway server to a management server, that same management server should not also be used to host traditional OpsMgr agents. This point deserves some additional clarification. In terms of priority, a management server does not distinguish between a traditional agent and a gateway server that it hosts. Because of this, data from a gateway will be assigned the same level of priority as that from a standard agent. On the surface, that may seem reasonable but consider for a second that you have 100 agents reporting to the gateway and the challenge immediately becomes apparent. Each of the 100 agents reporting to the gateway will only be afforded 1/100 of the standard priority afforded to an agent. If the management server hosting the gateway also has 100 agents of its own, each receiving 1/100 of the total priority, the agents reporting to the gateway will ultimately only receive 1/100 of that 1/100 slice of priority, or in more understandable terms, 1/10,000 of the priority. The diagram below explains:



With this in mind, it is recommended (but strangely, undocumented) that you assign a dedicated management server with no traditional agents reporting to it (even in a failover scenario), for the sole purpose of collecting data from your gateway server(s). It is important to keep in mind that Microsoft's recommendation is no more than 5 gateways and no more than 2,000 agents reporting to a single management server (whichever comes first). It is not beyond comprehension that you could host 10 or more gateways with the right hardware but any more than 5 and there is the possibility to run into processing bottlenecks within the product. This number should also be considered when planning for failover so in a highly available architecture, you would need to account for failover. In other words, in most cases hosting no more than 3 gateways per management server during normal operation and controlling failover so than no more than 6 will exist on a single management server at any one time. The diagram below demonstrates this:



The diagram above, illustrates the fact that when you are architecting your OpsMgr infrastructure for high availability, you need to consider the 'worst case scenario'. In other words, how the system will react in the event of a failure and how a failure will affect the infrastructure. In this case, how many agents will be reporting to the failover management server in the event of a management server failure?

One point that should also be considered when architecting the use of gateways is that whilst Microsoft has not documented a maximum number of management servers and gateways for a management group, the maximum supported limit is 10 management servers. There is currently no unsupported limit on gateways but if you consider the numbers of gateways recommended on a management server, you will be looking somewhere in the region of 20-30 gateways maximum (although the maximum I have experimented with is 16).

It is also important to ensure that you spread agents equally across multiple gateway servers that are reporting to a single management server. Once again, this is due to the way in which the management server prioritizes data from the gateway servers. This can be seen in the gateway behaviour image above.

Sizing Gateway Servers

Finally, when planning to use gateway servers in this way, it is important to ensure that they are sufficiently powerful. Of course, this mindset should be taken throughout the design but it is particularly

important when you begin to push the limits of the product. When planning for the RMS and management servers, RAM is critical followed by CPU and finally disk I/O (which is largely unimportant). Gateways however cache far more data to disk than management servers and therefore the order of importance of resources shifts. In the case of gateways, disk I/O is the priority, followed directly by RAM and then by CPU. For this reason, gateways are great candidates for virtualization, except in scenarios where scale is a key factor.

Creating a Highly Available Infrastructure

Now that we have investigated the concept of using gateways to span long-distance WAN links, we will look at the other major factor when planning for a large enterprise deployment, Availability. The vast majority of large companies will maintain an unhealthy reliance on their IT infrastructure and will therefore, in many cases, require that infrastructure to be highly available i.e. can tolerate failures and disasters. We'll cover disaster recovery and business continuity later and for now will focus on high availability.

Inherently, OpsMgr offers some high availability support, though high availability for some aspects of the product is supported by other Microsoft technologies such as Cluster Services and SQL Server. In order to consider an infrastructure 'highly available', it should be able to withstand a single failure of software and/or hardware anywhere in the infrastructure with little or no loss of service. In terms of OpsMgr, this equates to planning for failure of a management server, gateway, the RMS or either of the databases. In an ideal situation, a highly available solution should be able to tolerate a simultaneous failure across multiple components of the product.

By design, the RMS and Operational database represent single points of failure for OpsMgr. As OpsMgr provides no native method of coping with a failure of these components, if either were to fail the OpsMgr architecture would effectively be dead in the water until that component is recovered.

The following list details the affects of a failed RMS:

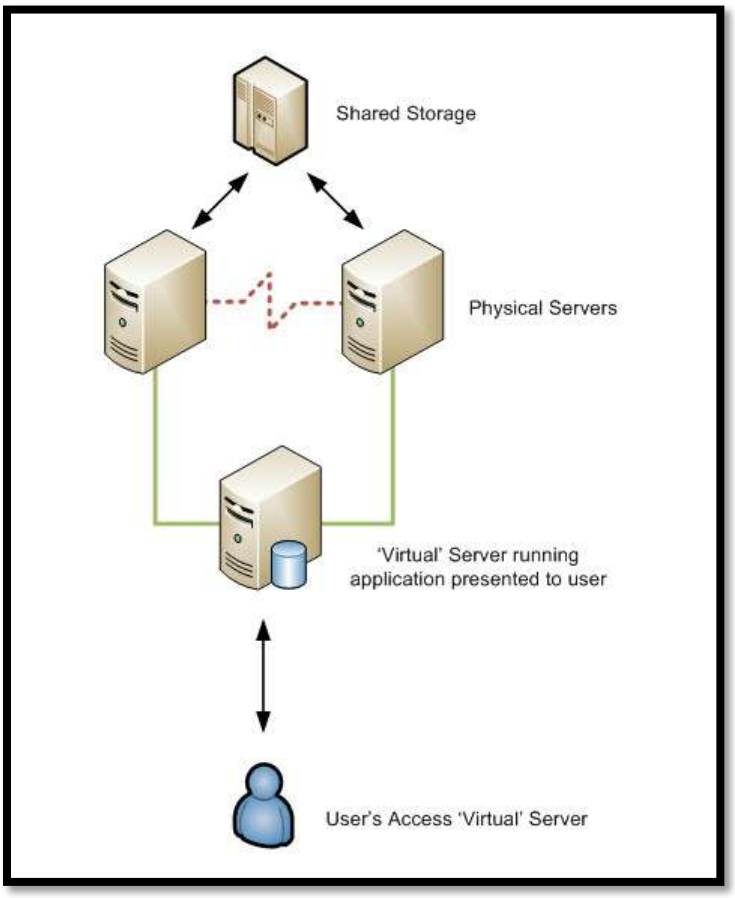
- Console access is lost
- Notifications are no longer processed
- State is no longer calculated (since this is controlled by the Config service on the RMS
- Connectors no longer function
- Connected Management Group connections would no longer function
- Any other SDK dependant components will fail

It should be noted that data processing continues and the management servers will continue to update the database so very little data should be lost. However, since state updates are no longer occurring, monitoring will not be up to date and once the RMS recovers, it may take some time to re-synchronize the monitoring states.

In addition, failure of the data warehouse in a large deployment will have a noticeable, adverse effect on the performance of the management group and should therefore be considered a single point of

failure in this case. With this in mind, these components should take advantage of supporting Microsoft technologies to facilitate high availability. The technology that will be most prevalent here is Microsoft Cluster Services.

Clusters function by presenting a 'virtual' server to the users whilst in fact hosting the data on two or more physical servers. These physical servers share the storage media on which the clustered application data is located and in the event of a server failing, another server can assume responsibility and continue to service user requests. The high level explanation of clustering is diagrammed below:



There are 2 main types of cluster configuration based on this idea. There are also some additional types but they are ultimately based on the 2 main types, usually simply with more physical servers (nodes).

Active-Passive Cluster

In an Active-Passive cluster, one node hosts the application and the other node sits idle. If the active server fails, the application is failed over and the passive node becomes the active node.

Active-Active

Both nodes host an application simultaneously. In the event that one server fails, the application it was hosting will fail over and the other node will now host both its own application and the failed over one.

In an OpsMgr environment, Active-Passive clustering is the only supported configuration. In addition, for reference, the supported cluster configurations for the OpsMgr components are detailed below:

Operations Manager 2007 SP1 supports the clustering configurations for Operations Manager server roles as shown in the following table:

Server Role	Cluster	Notes
Operations Manager 2007 SP1 Operations database	Single Active-Passive cluster	Setup needs to be run only once on the active node of the cluster. Other Operations Manager 2007 SP1 server roles must not be installed on the cluster or nodes of the cluster.
Root Management Server	Single Active-Passive cluster	Setup needs to be run on every node of the cluster. Other Operations Manager 2007 SP1 server roles must not be installed on the cluster or nodes of the cluster.
Operations Manager 2007 SP1 Reporting data warehouse	Single Active-Passive cluster	Other Operations Manager 2007 SP1 server roles must not be installed on the cluster or nodes of the cluster.
Audit collection database	Single Active-Passive cluster	Other Operations Manager 2007 SP1 server roles must not be installed on the cluster or nodes of the cluster.

Table 2 *- Supported Cluster Configurations in Operations Manager 2007*

For the databases, SQL Server may be clustered so that in the event of one of the nodes failing, the database will 'fail over' to the available node and continue to service the management group. In addition, the RMS role may also be clustered. I have found clustering of the RMS to be very reliable, though a little complex and fiddly to install and configure.

OpsMgr management servers and gateways are not supported in a cluster configuration mainly due to the fact that the application itself is designed to tolerate failures of management servers and gateways. Agents can failover to an available management server or gateway in the event of a server failure and gateways are able to automatically fail over to alternate hosting management servers in the event that the server they are connected to fails.

In addition to clustering the RMS, there is also another method of recovering the RMS in the event of a failure; promoting another management server to an RMS. Since the RMS should not host agents, it is recommended that a dedicated management server be provisioned to act as a backup RMS. In order to

ensure this server does not sit idle, it could be used to host the web console, reporting web components and the UI.

Agent Assignment in a Highly Available Infrastructure

When you are building a resilient infrastructure in OpsMgr, and also when you are using gateway servers to host machines across long-distance WAN links (often in the same Active Directory forest), it is critical to manage agent failover to ensure that in the event of a management server failure, the agents are distributed relatively equally across all the remaining servers. This is difficult to accomplish on a large scale without using Active Directory Integration.

Active Directory (AD) Integration allows agents to be assigned their management group and management server information directly from Active Directory. This is very convenient for large enterprises as typically, agents will not be deployed using the Operations Console but rather, using a software delivery mechanism such as Configuration Manager. This however, does not provide a strong enough argument for using AD integration. What really provides weight to this configuration is the fact that it is very difficult to manage agent failover without using AD integration. In other words, if you are using gateway servers and/or a backup RMS, there are going to be some decisions that you will have made about which management servers or gateways host which agents. AD integration is the only reliable and manageable method to facilitate these assignments.

AD integration works by creating a custom container in the directory and populating it with auto-created groups pertaining to the different management servers. These groups are identified by the agents using Service Connection Points (SCP's) and they are automatically populated by OpsMgr based on an expression specified in the Administration pane of the Operations Console. One important fact to point out at this point is that enabling AD integration **does not** require a schema update, something that the AD team will be happy to hear.

For more information on AD Integration, refer to the excellent document but Pete Zerger and Anders Bengtsson. This document can be found here.

Alternatives to Clustering

As we have already discussed, clustering is the recommended method to provide high availability for the OpsMgr databases and the RMS. However, if clustering is not an option due to hardware, software or skill limitations, there are some other options. As mentioned earlier, a backup RMS can be used to add an additional level of redundancy to a clustered RMS but in the event that clustering the RMS is not possible, the backup RMS strategy may be used as the primary method of providing high availability. One thing that should be considered though is the fact that promoting a dedicated management server to an RMS is, for the most part, a manual process. It is possible to script the process but this will involve custom development and is beyond the scope of this document.

Providing high availability to the SQL databases in the absence of clustering is possible in two ways; SQL **Log Shipping** or **Database Mirroring**.

SQL log shipping is the process of creating a second instance of SQL (usually on separate physical hardware) containing a copy of the OpsMgr databases. Logs are periodically transferred ('shipped') to the backup server and replayed against the copy databases in order to provide a relatively up to date copy of the data in the event of the primary database server failing.

Database Mirroring. The other method is to utilize new native technology present in SQL 2005 SP1 and above; database mirroring. Database mirroring achieves the same result as log shipping but unlike log shipping, is managed fully by SQL. It also provides an up to date 'synched' copy of the database as opposed to a copy which is updated on a schedule and therefore may be somewhat out of date.

On the surface, database mirroring would seem the obvious choice. However, since it has not been fully tested by the OpsMgr product team it is, as yet, unsupported. Log shipping however, is fully supported though less functional. In addition, before deciding on either one of these technologies, bear in mind the fact that the database copies will be hosted on physically unique SQL servers with different computer names and possibly different SQL instance names. This means that in the event of having to invoke disaster recovery and utilize the backup database, registry changes will need to be made to each management server and the OpsMgr databases will need to be updated to reflect the new database server name. However, this process can either be scripted or simplified by clever use of SQL aliases.

Business Continuity with OpsMgr

In addition to high availability, there is another factor to considering when planning deployments of all sizes, and that factor is **Business Continuity**.

Business continuity spans a broad range of technologies and processes but the basic concept is ensuring that data is both secure and accessible in the event of a failure of any kind. We use the term 'data' here in its broadest sense. Business continuity can be used to describe everything from the restoring of files from a tape through to the process of relocating an entire department or office in the event of a natural disaster, for example.

For the purposes of an OpsMgr deployment we will assume that high availability has already been considered and addressed and that data access and security is the major concern. Firstly we must consider data access in the event of a failure and the data we are specifically referring to is that which is contained within the databases since we already know that the management servers are redundant and, for reference, RMS data can easily be regenerated if necessary.

Delivering High Availability with Business Continuity

To ensure data continuity, in an ideal situation databases should be stored on two on-line servers in separate geographically remote locations to mitigate the risk of data loss in the event of damage caused by a disaster natural or otherwise. This can be achieved one of two ways. Firstly, as detailed earlier, log shipping or database mirroring may be used to ensure an up to date (or near up to date) copy of the databases is kept in another location. Another method is to utilize a somewhat unknown technology known as geographically remote clustering or 'stretch' clustering.

This is the process of hosting one node of a cluster in one location whilst hosting the other node in a separate location. To ensure true continuity, these locations should be in separate buildings on different utilities (power grid, water etc). Stretch clustering does require some additional hardware and skill but is not beyond the expertise of a number of large enterprises and as such, is a viable option in certain circumstances. At present, this configuration is also unsupported in the strict sense, although clustering the database is supported and stretch cluster is a supported configuration for Microsoft Cluster Services providing it is configured correctly and all hardware exists on the Windows cluster supported hardware list. The main advantage of stretch clustering is the fact that (in most cases) it encompasses both high availability and business continuity at the same time, as data is stored and accessible in both locations.

Backup and Recovery

Data accessibility is only one aspect of business continuity however. The other aspect to consider is backup and recovery. This is the process of ensuring that in the event of a complete failure where data is either no longer accessible or is corrupt, a recent backup copy of the data exists and can be restored and brought back online quickly and reliably. The first component of this strategy is ensuring you are performing regular backups to reliable media.

There are many methods for backing up data, each exhibiting their own pros and cons but suffice to say, data should be backed up as often as realistically possible, which is usually once in a 24 hour period at a minimum. OpsMgr best practices state that a daily full backup of the operational DB is recommended. In addition to a daily full backup, in environments where data integrity is critical, setting the operational database to Full Recovery Model and backing up the transaction logs regularly is also recommended. I find it beneficial to backup first to quick access storage such as NAS and then to tape where possible to enable very fast recovery of the latest backup and an archive of older backups in the event that the latest backup is damaged in some way. The period of time to archive backup tapes will vary depending on the company but at a minimum, backup tapes should be kept for seven days to allow for recovery of data up to a week old. This may of course differ for the reporting database since the data contained therein will remain valid for far longer than in the Operational database.

Backup is but one part of the process however. Once regular backups are being performed successfully, it is always prudent to create a solid **recovery plan** which should include regular tests of the recovery procedure. This will help to not only ensure that the backed up data can be recovered, but will also serve to familiarize the IT staff with the process for recovering data so that in the event of a major disaster, there is no confusion and data can be recovered with the minimum of fuss. I cannot emphasize this process enough. Far too often have I been called to recover data that simply isn't there because a company's recovery procedures and testing were lacking. When testing recovery of data, do not make any assumptions.

If tapes are taken offsite by an external company, invoke disaster recovery with them at least twice a year to ensure not only that the data on the tape is recoverable but that also, they are able to bring the tapes back to site in a timely and reliable manner. Always remember, once data is gone, it's gone.

Conclusion

Whilst I have endeavored to provide sound advice and recommendations based on experience throughout this document, there will be situations where the advice herein is not appropriate. It is always both the recommendation of myself and 1E as a company, that when attempting an OpsMgr deployment, especially a large or complex one, you seek the necessary expert advice to ensure that the investment in the technology is maximized.

However, by following the points documented herein and adhering to the basic principles of good planning, thorough research and documentation, attention to detail and appropriate training and skill transfer, the process of planning and implementing an OpsMgr deployment is possible.

Additional Resources Operations Manager Deployment Guide
http://technet.microsoft.com/en-us/library/bb419281.aspx

Configuring Active Directory Integration
http://systemcenterforum.org/wp-content/uploads/ADIntegration_final.pdf

Gateway Configuration in OpsMgr 2007
http://systemcenterforum.org/wp-content/uploads/OpsMgr2007_Gateway_Config_v1.2.zip

OpsMgr Infrastructure Design and Planning Solution Accelerator
http://www.microsoft.com/downloads/details.aspx?FamilyId=AD3921FB-8224-4681-9064-075FDF042B0C&displaylang=en#filelist

Feedback

I hope you find this document useful. Your feedback is always welcome and appreciated at andyd[AT]1e.com