Unicenter NSM Capacity Planning

In order to design the appropriate deployment architecture for Unicenter® Network and Systems Management (NSM) you need to consider:

- **Business Requirements – what and why.** What questions do you want NSM to answer? Do you want to know that business critical processes are running? Do you want to know that communications with remote sites are performing correctly? Are there Service Level Agreements (SLAs) and other monitoring requirements, which can help identify the deployment focus? What do you want NSM to do? Until these sorts of questions are asked and answered, further planning is nonproductive.

- **Who is going to use the information?** The information gathered and presented must fit the requirements of those who need it. Executives looking to satisfy their SLAs require a different perspective than operations. Information gathered must be capable of satisfying both needs.

- **Where do these people exist?** Are they in headquarters? Or, does it need to be available in regional operations centers? Do remote office managers need access to status information?

- **Deployment Environment.** This includes the existing infrastructure – network configuration, machine performance and distribution, existing software and procedures. You should also take into consideration any past technical issues and plans for future changes to the environment. For example, has a particular network connection been troublesome in the past? Are there plans to upgrade or relocate the servers?

- **Unicenter NSM Architecture.** This includes both the primary components and their dependencies and communication requirements. For example, the Distributed State Machine (DSM) monitors objects by polling and receiving traps messages through either SNMP or DIA. The number of objects that an individual DSM can reasonably monitor, as well as the timeliness of the status changes it reports, is directly related to the quality of the network connection between the DSM and its managed objects.

Capacity planning requires a careful review of all these elements in order to determine what ratio and placement of components will deliver optimum product performance and delivery. Although there are many other Unicenter NSM components, this document focuses on capacity planning for Agent Technology (AT). It identifies factors known to impact AT capacity and provides guidelines based on performance testing. Note that these guidelines are not a substitute for conducting careful pre-deployment testing in your own lab based on your particular environment and business requirements.
Planning Considerations for Agent Technology

Agent Technology is comprised of two primary components – the Distributed State Machine (DSM) and Agents. While these components provide valuable insight into the status of the environment, improper planning and tuning can lead to significant overhead or poor performance. Careful planning must be undertaken to assure that placement and tuning are consistent with operational requirements, infrastructure capacity and business objectives.

Distributed State Machine

Agent Technology monitors network infrastructure, application servers, applications and other components in order to detect and repair bottlenecks or outages as early as possible, thereby reducing costly down time. The DSM monitors objects by polling or receives trap messages (via SNMP or DIA) which it uses to compare an object’s current state with its previous state to determine whether a change of state has occurred.

The usefulness of DSM data, however, is affected by its timeliness, accuracy and impact on the network. For these reasons, you should keep the following guidelines in mind when determining the appropriate number and distribution of DSMs for your environment:

- **DSM placement should be relevant to achieve business goals and client specific objectives**
  
  When selecting which machines to monitor, you should consider their relevance in regards to achieving business goals and objectives. For example, while business critical servers and infrastructure should be included in the complement of monitored machines, other devices, such as desktop and laptop computers or printers, while important, may not require the level of 24x7 monitoring provided by Unicenter.

- **DSMs should be placed as electronically close to the objects they are monitoring as possible (generally within 3 network hops)**
  
  When a DSM monitors distant objects, particularly objects separated by a WAN, time-outs, lost traps and other inaccuracies may occur, causing delays and false alerts. While DIA is superior to SNMP in this regard, monitoring large numbers of remote objects is not recommended. Neither is it always logical to install a DSM in a remote location to monitor only a couple of machines. The network and workload must be carefully analyzed and considered in the context of operational requirements, infrastructure capacity and business objectives when determining how remote objects should be monitored. One should consider all solutions including
installation of a small DSM in the remote location or using other means to confirm state changes, such a Unicenter Remote Monitoring.

- **A DSM’s anticipated workload should not be such that it will exceed the capacity of the machine on which it resides**
  
The footprint of a DSM is largely dependent on the number of objects it monitors and the stability of the environment. The DSM process (aws_dsm) is most CPU-intensive during start up, but, once it stabilizes, CPU consumption shifts to the Distributed Services Bus (aws_orb), SNMP Gateway (aws_snmp), and DIA (aws_agtgate). Large monitored object populations prolong startup and require that DSM allocate greater memory resources.

- **DSM Polling Interval affects DSM capacity**
  
  DSM’s main impact on the network is from polling. While a DSM can poll its agents and other objects at very short intervals this is discouraged because it is generally not necessary. Polling, instead, acts as a back up to SNMP or DIA trap messages that agents issue at the time of a detected state change. Whereas a trap is issued immediately, there will always be some lag time during the polling interval before the next poll is received. It is preferable, therefore, to not clutter the DSM and the network with superfluous polls indicating no state change and permit it to process traps as fast as possible.

  Short polling intervals also affect the number of objects DSM can monitor. First, because excessive polling consumes the MAXPOLLRATE. In general, to allow ample capacity for other traffic, projected polling should not exceed 50% of MAXPOLLRATE. For more details on MAXPOLLRATE, see the Unicenter NSM Inside Systems Management guide. Secondly, short polling intervals induce polling latency which further delays polling and reduces DSM capacity. When the number of polled objects grows and the polling interval is too short, a condition could be created wherein the polling cycle cannot complete before the next begins. This bottleneck and additional overhead causes DSM to be unable to manage the workload.

  Reducing the poll interval resolves the situation.

- **DSM should have fast reliable access to the MDB**
  
  Ideally, DSM should also have fast reliable access to its MDB, particularly during startup when the full inventory of monitored objects must be transferred and also when processing status updates. It is generally more important, however, that the DSMs be located closer to their managed objects.

  It is **not** recommended that DSMs used for primary monitoring be located on the MDB server. If the MDB server supports other applications, the rate at which objects are transferred to the DSM during startup may impact those applications. The transfer rate can be tuned to reduce DSM startup impact (see “WorldView Gateway initialization configuration section” in Inside Systems Management) although this may affect DSM startup performance.
Note: Further discussion of DSM placement in relation to the MDB is provided later in this document.

Agents

Although Unicenter Agents provide a rich spectrum of data on many aspects of the infrastructure, you should exercise care in determining which agents to install and on which systems. The OS System Agent can be reasonably installed on most server machines and is often the most widely deployed agent.

Whether you are using Adaptive Configuration or classic configuration files to define agent configuration, careful consideration should be given to ensure that the metrics collected are pertinent to the health and well being of the host, the applications it supports and the enterprise. The default configuration for the Windows OS system agent (caiWinA3), for instance, can generate over 50 managed DSM objects. Multiply that by several hundred Windows hosts and the DSM population can grow rapidly to critical levels.

MDB Considerations

MDB placement is of strategic importance in planning an Agent Technology architecture. The MDB must be accessible by the DSMs it serves, but it must also be accessible to the end users using its data. Where it is not practical for a DSM to access a remote MDB due to network constraints or business or organizational requirements, then additional MDBs may also be placed in these locations.

Consideration should also be given to the location of the end users of MDB data. If, for instance, there are regional operations centers that will be accessing the MDB, then consideration should be given to locating additional MDBs there to provide ready access to system status. In these cases, critical alerts should be identified at the regional level and forwarded to the central MDB via the MDB Bridge Service to provide a consolidated display of overall system status.

Additional information regarding the MDB placement can be found in the “Local or Remote MDB” document.

Data Isolation

When there are multiple DSMs monitoring objects in a single MDB, it is vital that they be segregated so that only a single DSM monitors each object. This can be done in several ways – the most direct of which is through IP Address scoping. Using the DSM Wizard or MCC Tools, IP address scoping can be implemented to restrict which IP addresses a particular DSM monitors. The
DSM will automatically update the MDB and DSM objects once the new configuration is activated.

**Note:** This does not apply to “secondary” DSMs which are deployed for fault tolerance/failover purposes.

### Capacity Considerations for Agent Technology

The term “capacity,” when used in the context of planning a solution, should be understood to mean not simply the maximum load that can be applied but also the ability to deliver the results required in compliance with a specified service level under a given load. Significant improvements in performance for Unicenter NSM r11 components make it possible, in general, to reduce the number DSMs deployed when compared to previous versions. Reducing the number of DSMs can reduce both hardware and administrative costs driving down the total cost of ownership while still keeping DSMs electronically close to its agents.

Although capacity discussion for AT centers around the DSM, you should first be aware of how AT scaling decisions impact MDB capacity.

### Impact on MDB Capacity

MDB utilization is based on the number of objects it holds and other factors, such as agent activity and network alerts. Utilization will also be impacted if other Unicenter applications will use the MDB or if the host system will support other applications or databases.
Consider the following graph:

![Graph: Supported MDB Objects Based on Utilization Percent](image)

The capacity of the MDB is a linear scale varying from 75,000 to 25,000 objects based on the percent of MDB utilization anticipated where the lower numbers are due to higher anticipated activity.

**Note:** MDB managed objects should not be confused with DSM objects. Although DSM objects are associated with managed objects, they reflect what is found by the DSM 2nd level discover (which is driven by policy). There is no direct relationship between the two object types and no simple rule for converting the number of managed objects into the number of DSM objects. DSM object counts, however, are generally much larger than managed object counts.

Database servers hosting MDBs with low utilization should typically have at least the following minimum specifications (for more information on server configuration please refer to "Unicenter NSM Infrastructure Sizing Tool"):

- CPU Count = 1
- CPU Speed = 2800 MHz
- RAM = 2048 Mb

**Note:** All recommended hardware configurations provided in this paper are best practice guidelines not requirements. Local conditions and configurations should be carefully considered in determining optimal hardware configuration.

Database servers hosting MDBs with medium to high utilization should typically have at least the following minimum specifications:

- CPU Count = 2
- CPU Speed = 2800 MHz
- RAM = 4096 Mb

When considering enhancements to these hardware specifications, priority should be given to including additional memory and upgrading disk systems (e.g., 1 or more RAID 10 arrays).

**DSM Capacity**

DSM capacity estimates are based on the number of objects a DSM may monitor adjusted for other factors such as agent and network alert activity. Consideration should also be given to available network capacity between the DSM and both the objects it will monitor and the MDB.

DSM activity has a profound effect on the DSM capacity. The chart below shows DSM capacity as a linear scale, varying from 50,000 to 30,000 objects, based on the percent of DSM activity anticipated - where a DSM in an extremely stable environment could manage up to 50,000 objects but a DSM in an extremely unstable environment might only handle 30,000 objects. Note, these are not hard limits.

Three factors that most impact DSM capacity are:
- Location in regards to MDB (local or remote)
- Host machine capacity
- Network capacity
MDB Location and Cold Startup

DSM startup is one of the busiest times for the MDB during a DSM run cycle. It is also a critical time as the environment is not properly or completely monitored until startup is completed and therefore must be done as quickly as possible. As a Cold Start on a large DSM can take thirty to sixty minutes or more, the DSM Warm Start option is usually preferred as it significantly reduces startup by preloading DSM with the previous object inventory and state saved in the object store.

Second level discovery occurs during DSM start up and is comprised of 4 important operations. First, DSM must receive its inventory of hosts to manage from the MDB and determine their existence. Then, it discovers any agents active on the hosts. Next, agent instantiation occurs wherein configured monitored metrics are discovered. And finally DSM updates the MDB with the new status.

The charts below demonstrate the impact on a cold startup of a local versus a remote MDB. In the first chart below, a workload of simulated devices is evenly divided between two DSMs, one local and one remote. The chart shows that the remote DSM completed its startup in about 30 minutes while the local DSM started well but took nearly twice as long to finish due in part to heavy MDB processing.
In the chart below, the same workload was split between two DSMs using a common remote MDB. Here the startup performance was almost parallel and, in fact, it was a little faster than the remote DSM performance above. Note that while “remote”, these DSMs and the MDB were located on the same high performance subnet.

As mentioned earlier, the DSM should also be as electronically close to the MDB as possible but it remains more important that the DSM be close to its monitored objects.

Generally, the DSM and MDB may be co-located in smaller installations as startup implications are smaller. In fact, best practice is to always install a DSM on the MDB server for at least two reasons. First, it becomes the default DSM for newly discovered machines so that they will be monitored even before they have been moved to their desired DSM. Second, this DSM should not be used to monitor a large number of objects; it is a good option for monitoring the other DSMs in the environment.
Host Machine Capacity

Naturally, the most prominent constraint is the capability of the host on which the DSM runs. All host performance management comes down to 3 constraints – memory, CPU and I/O – and DSM is no different. DSM consumes a lot of memory depending on its workload – not just for the aws_dsm.exe process but for its other components, such as aws_snmp, as well. When DSM shares the machine with a DBMS server or other applications, it is important to make certain there is enough available RAM to handle all memory requests without excessive paging.

The next critical element is the CPU(s). Aws_dsm, by itself, does not take advantage of a multiprocessor system. However, various DSM processes can be run on different processors so that a multiprocessor system can be used to good advantage. Aws_dsm.exe is extremely CPU intensive during startup and during periods of high status volatility. Aws_snmp and aws_orb are always quite active as they handle all of the message traffic to and from the DSM. Therefore, multiple CPUs are desirable in an active large environment.

Below are general server hardware guidelines based on DSM utilization as defined above. (Refer to “Unicenter NSM Infrastructure Sizing Formulas” for more specific guidelines.)

**Lightly loaded DSM servers:**
- CPU Count = 1
- CPU Speed = 1600 MHz
- RAM = 1024 Mb

**Moderately loaded DSM servers:**
- CPU Count = 2
- CPU Speed = 2800 MHz
- RAM = 2048 Mb

**Heavily loaded DSM servers:**
- CPU Count = 2
- CPU Speed = 2800 MHz
- RAM = 4096 Mb

When enhancing DSM server hardware configuration, priority should first be placed on memory and increased CPU speed.
Best practice is to have a DSM as the primary user of a server. However, when the workload is very light, such as in a small remote office, then the DSM (and MDB if need be) can be co-located with other applications on a server.

**Configuration 1:** Assume a headquarters building or campus environment where NSM is to manage 100 to 300 machines and network gear. All monitoring is considered local with no WAN links. An environment such as this may produce 20 to 25,000 DSM objects based on each server having an OS agent, some of the servers having a log agent, and some of the servers having some other agent. Depending on ratios, this number could vary considerably but assume that each OS agent generates about 50 to 60 objects, while the log and other agents generate 20 objects each. The object per agent number is critical as it can quickly explode the DSM population. Where do these objects per agent numbers come from? Examine the default Windows OS System Agent for example.

The Node View depicted above is of a Windows 2003 Server with only a Windows OS agent installed using the default configuration. Each object in the figure represents a DSM object. Here, 36 DSM objects are shown as follows:
This configuration does not include any services, processes, files, etc. There could easily be 20 additional metrics are monitored per agent. Some agents produce a very large number of managed objects; an Active Directory agent, for instance may represent 120 or more DSM managed objects. So, it is not only the type of agent but the metrics monitored that must be considered when sizing the environment.

### Network Considerations

While a DSM may handle 30,000 objects or more, the constraining factor is as likely to be the network architecture as server hardware or other factors. Deciding whether to monitor a group of machines locally or remotely is governed by a combination of a DSM’s utilization and network configuration. (Refer to “Unicenter NSM Infrastructure Sizing Formulas” for more specific guidelines.) For example, if a DSM's workload would be 34,000 objects (or 80% utilization in the previous chart) or more and the network connection to the target subnet is 10Mbps or less, then an additional DSM monitoring the subnet would be recommended. However, a local DSM is not recommended where the network is 100 Mbps or better and DSM capacity is available. In considering these decisions, be certain to allow for unexpected DSM workload as well as network congestion and reliability by applying these parameters conservatively.

Consider how changing the network affects the recommended architecture. For example, assume a retail architecture with headquarters, three regional offices and one hundred local offices with 100 managed servers in the headquarters, 20 in each of the regional offices and 10 in each local office. All told, DSMs would be required to manage about 150,000 objects requiring 5-7 DSM servers with good network infrastructure. The constraining factor, though, is the network which can significantly change the architecture.
Configuration 1: Assume the architecture has network connections of 10 Mbps or better within the headquarters and to the regional offices with 256 Kbps to the local offices. In this case, the environment could be monitored with 7 centrally located DSM servers monitoring an average of about 20-25,000 objects each. The recommended configuration of the DSM servers in this case is dual CPUs of about 2.8 GHz or better and 2 GB of memory.

Configuration 2: Assume 2 Mbps or slower network connections from headquarters to the regional offices and 1 Mbps to the local offices. Here, the network performance and capacity constrains DSM traffic with the regional offices and the recommended architecture places DSMs at each of the regional offices instead. The headquarters and local offices could be monitored from headquarters by 5 DSM servers with dual CPUs of 2.8 GHz or better and 2 GB of memory with 4 of those used for the local offices. Each regional office will have a DSM server of the same configuration. So, the slower network to the regional offices requires one additional DSM server to manage the environment when compared to configuration 1.

Configuration 3: Assume 2 Mbps or slower network connections from headquarters to the regional offices and 128 Kbps or slower connections to the local offices. Here, the network performance and capacity constrains DSM traffic with the local offices so that it is no longer practical to manage them from headquarters or the regional offices. In this case, a DSM installed on a single CPU machine of about 1.6 GHz and 1 GB of memory should be installed in each local office or since DSM is expected to be lightly loaded co-locating it on a server handling other applications might be considered.

These examples, while valid, are somewhat simplistic in that they do not consider other factors that require consideration when designing an architecture. Consider the following:

Configuration 4: Assume the same configuration as in Configuration 2 above but the organization has decentralized systems operations to the regional offices which maintain their own network operations staff. It may, therefore, be considered more consistent with business practices and management structure to install a second DSM server in each regional office to share the load of monitoring the regional office and its subordinate local offices. Consideration may also be given to implementing an MDB in each regional center to provide better access to network management data to the local staff. The machine configuration would be in the range of dual 2.8 GHz or better CPUs with 2 GB of memory. Depending on network architecture, this arrangement may provide more reliable monitoring of the local offices because the connections are more direct. The total number of machines required is, again, 7.
Organizational Considerations

Many enterprises do not fit neatly into a single architectural type but rather are a hybrid of one sort or another. Perhaps there is a large headquarters operation but also local offices and perhaps a subsidiary within the corporate structure. Or, perhaps there are several peer level locations with redundancy to support disaster recovery operations. And, of course there are many others.

**Configuration 1:** Assume a headquarters campus with a remote subsidiary operation that is monitored and controlled from headquarters. Assume headquarters has 200 Windows servers and 50 Unix server all of which have OS agents generating 50 DSM objects each and 10% have log agents generating 10 DSM objects each. This would generate about 22,000 DSM objects.

Assume the subsidiary location has 50 Windows servers each with OS agents generating about 50 DSM objects each and 10% have log agents generating 10 DSM objects each. Assume the network connection to the subsidiary is 2 Mbps. The subsidiary would generate about 4,500 DSM objects.

Since the headquarters staff is responsible for monitoring the subsidiary, the recommended configuration would be to place the MDB and DSMs at headquarters. It would be recommended that the MDB be installed on its own machine. This machine would require WorldView Manager to be installed and best practice would be to install a DSM, Event Management and providers here as well. As described above, this DSM would be to catch unassigned objects and monitor the other DSMs. This machine could be a single CPU processor of about 2.8 GHz or better and 2 GB of RAM.

Two DSM machines would be recommended and these should be 2x2.8 GHz or better and 2 GB of RAM.

Managing DSM Workload to Optimize Capacity

There are several tools available to manage the DSM workload and, therefore, to optimize the DSM’s capacity.

- **IP Address Scoping**
- **Class Scoping**
- **Polling**

**IP Address Scoping**

IP Address Scoping is the most obvious choice for managing DSM workload. Using either the DSM Wizard or the MCC DSM Configuration tool the scope of IP addresses managed by DSM can be controlled thereby limiting the number...
of hosts DSM manages. One can assume that each managed host discovered by DSM will result in about 50-75 managed objects. A couple of subnets of managed servers can quickly become a significant number of managed objects and exhaust DSM resources.

**Note:** This is a very loose estimate. Consider that several agents, including system and log agents, could have dozens of watchers each.

### Class Scoping

By default, DSM will manage every class of object defined to the MDB. This could be desirable or it could be wasteful. For instance, perhaps there are agent classes that do not exist in the site’s inventory. DSM will still interrogate each object of a host class that could potentially be a parent of the agent class during startup delaying DSM settling. For instance, policy for both the caiW2kOs and caiWinA3 agent classes declare their parents as the same group of machines. This means that every machine of the 5 classes comprising the group will be interrogated for both agent classes. If caiW2kOs is not active in the environment, remove it from the scope.

Parent classes can also be reduced by editing the gwclass.dat and gwclassii.dat files. For instance, if monitoring Workstation class machines is not required, removing the “Workstation” entry in these files will prevent aws_wvgate from retrieving them from the MDB. Also, all subclasses of the classes listed in the policy files will be retrieved which could exacerbate the condition.

### Polling

Polling is another important factor affecting DSM performance and the number of objects it can manage. The first thing to understand is that polling whether using DIA or SNMP is actually controlled by the aws_snmp gateway. DSM passes polling requests to aws_snmp gateway and it issues as many poll requests as required up to the value defined in MaxPollRate in the atservices.ini file each second. By default MaxPollRate is 300 polls per second in NSM r11.x.

The second thing to consider is the polling interval required by your site. CA Best Practices state that polling intervals should normally be 10 to 20 minutes as polling is considered a backup for traps. In those environments where more intense polling is required the number of objects managed by a DSM may need to be reduced.

Consider the following when calculating projected polling rates:

- Network infrastructure
- You should target no more than 50% and no less than 20% of the MaxPollRate utilization
• Required polling interval

• Suppose you determine to limit polling to 200 per second. For a 5 minute polling interval (300 seconds) do not attempt more than 30,000 polls during the interval (300 polls × .5 × 200 polls/second = 30,000 polls every 5 minutes)

• Assuming 50 monitored objects per host, the DSM in the above example should monitor no more than 1,500 hosts