Veritas Enterprise Vault™ Discovery Accelerator

Best Practices for Implementation

14.2

VERITAS

Veritas Enterprise Vault™ Discovery Accelerator: Best Practices for Implementation

Last updated: 2021-12-13.

Legal Notice

Veritas Technologies LLC
2625 Augustine Drive
Santa Clara, CA 95054

http://www.veritas.com

Contents

Chapter 1

Introduction

Purpose

Sizing and implementing Discovery Accelerator requires careful planning to ensure that the product can perform to expectations, scale as the customer requirements grow, and ensure that the underlying Enterprise Vault infrastructure is configured to support the required activity.

Discovery Accelerator requires the proper consideration in the following areas:

- Discovery database server
- Discovery application server
- Enterprise Vault infrastructure, including indexing and storage services

The purpose of this guide is to discuss the different aspects that you need to consider during sizing and recommend best practices for implementation.

The Discovery Accelerator business users and the team designing a solution need to discuss the Discovery Accelerator model, paying attention to the anticipated number of active cases, simultaneous users, potential volume of review, and any other litigation support applications with which data must be exchanged. Veritas publishes several white papers that explore the different options and use models.

This guide assumes that are you are familiar with how to configure and administer Discovery Accelerator and associated products. You can obtain more detailed installation and configuration information from the Discovery Accelerator documentation. Veritas also publishes several white papers that explore specific details such as effective searching, security, and more.

What's new in this guide

This guide has been updated from the previous version as the result of further performance investigations and feedback.

Search performance of Discovery Accelerator 14.2 is improved due to the introduction of Elasticsearch as the new indexing engine in Enterprise Vault 14.2.

In cases other than search, the performance of Discovery Accelerator 14.2 is equivalent to that of previous versions.

Supporting documents

Use this guide in conjunction with the following documents:

- *Enterprise Vault Performance Guide*, which is available from the following page of the Veritas Support website:

  http://www.veritas.com/docs/000005725
- *Enterprise Vault Compatibility Charts*, which is available from the following page of the Veritas Support website:

  http://www.veritas.com/docs/000097605
- *Enterprise Vault 14.2: SQL Best Practices Guide*, which is available at the following location:

  http://www.veritas.com/docs/000021697
- Enterprise Vault Best Practice Guide - Implementing Enterprise Vault on VMware, which is available at the following location:

  http://www.veritas.com/docs/000081275
- *Enterprise Vault Indexing Best Practices Guide*, which covers details of the indexing system. The document is available from the following page of the Veritas Support website:

  https://www.veritas.com/support/en_US/doc/EV_Indexing_BP_142

Customer profiles

Small customer

- 100 mailbox users with both journal and mailbox archiving.
- Requires a single server for Enterprise Vault and Discovery Accelerator.
- Server storage needs to be local or directly attached device.
- One legal Discovery user (case administrator).
- 10–20 active cases per year.
- Low search and production load, and single online reviewer.
- Likely to have one or two cases in analytics at any time.

Medium customer

- 10,000+ mailbox users with both journal and mailbox archiving.
- Large volume of historical data to archive.
- Environment spread over multiple sites.
- Rapid and large or critical responses required.
- Legal Discovery team likely to have 4–10 members.
- 50–100 active cases per year.
- High search and production load, and 10 online reviewers.
- Likely to have up to 25 cases in analytics at any time.

Large customer

- 30,000+ mailbox users with both journal and mailbox archiving.
- Large volume of historical data to archive.
- Environment spread over multiple sites.
- Rapid and large or critical responses required in high pressure environment.
- Legal Discovery team likely to have 5+ full-time members.
- 1000+ active cases per year.
- Very high search and production load, and 20 – 30 online reviewers.
- Likely to have up to 50 cases in analytics at any time.

General hardware considerations

Apply these hardware considerations to all servers deployed as part of the Discovery Accelerator environment.

CPU

The power of a server is not necessarily determined by the CPU speed in terms of cycles per second. Factors such as the server architecture and number and type of processors and cores can provide a far greater benefit over increasing CPU speed.

Hyper-threading technology is claimed to provide up to 30% improvement in performance. These processors contain two architectural states on a single processor core, making each physical processor act as two logical processors. However, the two logical processors must share the execution resources of the processor core, so performance gains may not be attained. The operating system, and potentially the application software, needs to support hyper-threading to avoid inefficient use of the logical processors. Windows server operating systems support hyper-threading.

Multi-core technology provides similar performance to a comparable multi-CPU server. These processors contain multiple complete processor cores, which act as complete physical processors. Each physical core has its own architectural state and its own execution resources, so the performance gains are reliable.

With the ever-increasing number of processor core combinations and high clock speeds, the traditional x86 Front Side Bus architecture can start to become a bottleneck beyond eight processor cores. A popular and cost-effective method of scaling up the x86 architecture is to use an architecture that supports non-uniform memory access (NUMA). Processors and memory are grouped into nodes that have high-speed local access. However, access to memory co-located with other processor nodes is slower. Therefore, the operating system (and potentially application software) needs to be NUMA-aware and optimized to make the best use of processors, cores, and their associated resources. Windows server operating systems support NUMA.

Memory

A 64-bit platform should be considered for memory-intensive applications, if supported. Using a 64-bit platform can provide more efficient memory utilization, which can bring performance benefits.

Storage

In most cases, you require RAID-based storage to achieve your storage requirements. To maintain performance and reliability, consider hardware-based RAID rather than software-based RAID. To achieve redundancy on striped arrays while maintaining performance, consider the RAID scheme carefully.

RAID levels 5 and 6 are popular, cost-effective methods of achieving redundancy while maintaining striped disk read performance. However, writing incurs a cost of four to six physical operations per write. A poorly sized RAID-5/6 implementation can significantly reduce the performance of write-intensive activity. Correctly sizing a RAID-5/6 implementation to maintain write performance may become more costly than RAID-10. Therefore, in some cases discussed below, a RAID-10 scheme should be considered.

In the case of local or direct attached storage, use multiple controllers supporting multiple channels to distribute the load between the multiple storage locations and provide sufficient throughput. The controllers should also provide a battery-backed read and write cache to aid performance.

Before you use partitions on a storage area network (SAN), consider the I/O load together with any other applications that are already using the SAN to ensure that the performance can be maintained. Ideally, discuss the implementation with your SAN hardware vendor to ensure that you achieve optimum performance. Typically, you should create LUNs across as many suitable disks as possible, using entire disks rather than partial disks to prevent multiple I/O-intensive applications from using the same disks. When you configure HBAs on the host, ensure that the Queue Depth is set to an optimal value. This should be discussed with the storage vendor.

When you create a volume on a storage device (basic NTFS or virtual datastore), it is very important to align the volume with the device sector or stripe unit boundaries to prevent unnecessary disk operations that can significantly impact performance. Most current operating systems and hypervisors create new volumes with an offset of 1 MB (2048 sectors). If manual alignment is necessary, you can achieve this through the Windows command-line tool diskpart. See the Windows Help or Microsoft TechNet for more information.

Virtualized infrastructure

There are important aspects to consider when installing Discovery Accelerator or SQL Server in a virtualized infrastructure. Follow the recommendations of your hypervisor vendor and Microsoft when you size and configure the environment.

The primary objective is to ensure that the resource requirements described in this guide are dedicated to the virtual machine to ensure minimum impact to the performance from intermediate layers or co-existing guests.

The hypervisor should be type-1 (native) to ensure the minimum impact on hardware resource requirements. Note the following general guidelines:

- In a typical virtualized infrastructure, local disks might be used for the hypervisor and SAN-based storage for the guest operating system images and data file locations. The guest operating system and data storage partitions should be independent, dedicated locations, as described above.
- Disk partitions should be aligned with the device sector or stripe unit boundaries to prevent unnecessary disk operations that can significantly impact performance. By default, most current hypervisor versions align new partitions to 2048 sectors, which should be sufficient for most devices.

  The disk partitions to be used for the database log files should be created as recommended by the hypervisor vendor for sequential access.

  The disk partitions to be used for the database data files should be created as recommended by the hypervisor vendor for random access (most likely virtual hard disks).
- Virtual hard disks should be created as fixed size and not dynamic.
- Hyper-threading can be left enabled and can be used by the hypervisor to improve efficiency; however, the hyper-threaded logical cores must not be counted in the resource allocation calculations. For example, a 32-core server with hyper-threading may show 64-logical cores. However, all calculations and resource reservations should still be based on 32 cores (do not allocate the additional 32 logical cores). It may appear that there is spare capacity, but these additional cores do not provide the capacity of real cores, and allocating them will be detrimental to performance.
- Avoid using virtual machine snapshots, which can impact performance. If snapshots are used for maintenance and upgrade purposes, the snapshots should be rolled-up before commissioning the server for production use.

- The memory requirements recommended above should be dedicated and prioritized to the virtual machine to prevent dynamic allocation or sharing.
- The number of processor cores as recommended above should be exclusively dedicated to the virtual machine, and the processor priority and bandwidth set to provide the virtual machine with full utilization.
- The server power management scheme and all associated hypervisor and operating system settings should be carefully considered to ensure any performance implications are understood or minimized. Using C-states is very likely to impact performance.

If you want to install the SQL Server instance on a virtualized machine, avoid installing multiple instances on the same virtual machine.

Chapter 2

Database platform

The Discovery Accelerator database servers should be sized, tuned and maintained according to Microsoft best practice advice for SQL Server. See the TechNet article "SQL Server best practices". In addition, it is recommended to review all associated Microsoft tuning and update guidelines including the following Microsoft article:

https://support.microsoft.com/en-us/kb/2964518

This guide discusses some of those best practices from a Discovery Accelerator perspective and should be used in conjunction with Microsoft advice.

The Discovery Accelerator database performs the majority of data processing, and this tends to be very resource-intensive. These typically high loads will struggle to co-exist with any other database application, and, during Discovery Accelerator implementation, you must pay special attention to the database server and its configuration.

We recommend that the database server runs a single SQL Server instance only, with no other applications or services running on the database server. SQL Server needs to fully utilize the server resources, and another application may introduce contention issues that results in undesirable performance.

Hardware requirements

SQL Server benefits from multiple processors, particularly with the parallel activity generated by the Discovery Accelerator service. The flow of data within the Discovery Accelerator service, and the manipulation of data at the Discovery Accelerator database, can cause high memory utilization. Most Discovery Accelerator actions result in updates to the database, and the volume of manipulation means that the I/O subsystem has to handle a high level of activity.

CPU considerations

A Discovery Accelerator customer database generally fully utilizes at least four CPUs during Discovery search, review, and export activity, which also depends on the database server I/O subsystem throughput. In addition, collecting analytics data for a single case can require the use of more CPUs.

Therefore, a SQL Server instance that is hosting a Discovery Accelerator customer database should have a minimum of four CPUs dedicated to the SQL Server instance — either four physical CPUs or similar combination of multi-core CPUs, but not based on hyper-threading. Medium and large customers may wish to consider at least eight CPUs.

Note: Hyper-threading is not recommended to improve database performance due to potential performance problems when the database places a load on the memory. For more information, see Microsoft Knowledge Base article 322385 (http://support.microsoft.com/kb/322385). If hyper-threading is to be used, pay particular attention to the MAXDOP setting as described in Knowledge Base article 322385.

The Discovery Accelerator database needs to be hosted on an x64-based platform. The 64-bit platform provides performance benefits due to memory enhancements.

In most cases, the SQL Server instance should manage the CPU resources. Do not set the CPU affinity mask unless absolutely necessary, as this can significantly impact the performance. When you run multiple SQL Server instances, the most common reason for setting the CPU affinity mask is to prevent an instance being starved of resources (see “Multiple SQL Server instances” on page 34).

Memory considerations

The Discovery Accelerator stored procedures employ methods that are memory-intensive. In addition, the flow of data between many tables and their indexes causes a large number of pages to be required in memory at any time, consuming large volumes of memory. The Discovery Accelerator analytics feature can potentially insert large text documents into tables, which results in very high memory use and corresponding I/O activity.

The SQL Server instance that hosts the Discovery customer database requires enough memory to ensure that the data manipulation does not cause excessive paging to disk both at the Discovery database and tempdb, which will quickly degrade the performance. A small environment should have a minimum of 8 GB of RAM, whereas a medium or large environment

should have at least 16 GB of RAM available to the SQL Server instance for the Discovery Accelerator databases.

The use of analytics places particular pressure on the memory. If users enable multiple cases for analytics simultaneously, the collection of analytics data in each case applies very high pressure on memory. This is reflected in very high I/O activity, high CPU, and potentially a reduction in throughput. To reduce the I/O load and maintain the throughput, you can scale up the database server memory by 4 GB per anticipated concurrent analytics data collection.

Using an x64-based 64-bit platform provides more efficient memory utilization and brings performance benefits. Install the appropriate edition of Windows Server and SQL Server to support the capacity of memory that you have installed.

The memory maximum value should be tuned to ensure sufficient physical memory is available for the operating system and any co-existing instances, reporting services, or other services. A simple rule of thumb for a single instance would be to set the maximum memory to 75% of the available RAM.

It may also be necessary to change the SQL Server minimum and maximum memory to ensure the memory is used appropriately between SQL Server instances, Reporting services or other co-located services.

Network considerations

We recommend that the Discovery Accelerator database server, Discovery Accelerator server and Enterprise Vault servers are connected via gigabit network technology. The database servers may require multiple network interface cards to support the anticipated loads.

It is also recommended that the TCP Chimney Offload, TCP/IP Offload Engine (TOE) and TCP Segmentation Offload (TSO) are disabled. See Veritas technical article http://www.veritas.com/docs/000081332.

Storage considerations

Take steps to ensure that the storage does not become a bottleneck. By following Microsoft SQL Server best practices, you can ensure that the database server is suitably sized. Try to avoid using network-based storage for the database files.

Each database requires the disks to be arranged for two different purposes: the database data files, and the log files. The data files require good random access or high number of IOPS, and therefore a striped array of many disks

should be used (using hardware-based RAID rather than software-based RAID). The log files require good sequential write performance, so each log file should be placed on its own high speed array with good transfer rates.

To achieve redundancy on the sequential write-intensive disks (log), use a RAID-10 scheme with high speed, 15k rpm disks.

In the case of local or direct attached storage, use multiple controllers that support multiple channels to distribute the load and provide sufficient throughput. The controllers should provide a large capacity battery-backed read and write cache. A 512 MB controller cache is recommended for local or direct attached storage.

When you select storage, consider the advice in "Storage" on page 11, and create and align partitions as advised. See the TechNet article "SQL Server best practices". This article also recommends that you format both log and data partitions with 64 KB allocation unit sizes.

Arrange the database server storage to accommodate the different types of data. Typically, database servers should have the following partitions.

Recommended partitions for database servers

Partition	RAID array
System drive	RAID-1 array
Tempdb log partition	RAID-1 or 10 array
Tempdb data partition	RAID-10 striped array of several drives
Configuration database and Custodian Manager log file partition	RAID-1 or 10 array
Configuration database and Custodian Manager database data file partition	RAID-10 striped array of many drives
Each customer database log file partition	RAID-1 or 10 array
Each customer database data file partition	RAID-10 striped array of many drives
One or more analytics data file partitions, which can also serve as Full Text Index file locations	RAID-10 striped array of many drives
One or more analytics full-text index file partitions (optional)	RAID-10 striped array of many drives

Ensure that the partitions only contain the related database files so that they do not become fragmented on disk. If multiple database files are located on

one partition, it is likely to require regular file defragmentation to maintain performance.

When the Discovery Accelerator customer database is created, alter the default data file size to a large value, perhaps representing the next year's activity. This prevents file fragmentation and wasted I/O and waits while growing the files.

Consider the storage capacity of the Discovery Accelerator database server carefully.

Sizing the configuration database

The Discovery Accelerator configuration database stores details of the following:

- All registered customer databases
- The Custodian Manager database
- All customer configuration options
- Errors that the service has logged

The database is also used to manage analytics data collection tasks across all customers and cases. In general, the configuration database remains less than 10 MB. However, error conditions can quickly grow the related table. A good rule of thumb is to allow at least 100 MB for the configuration database data file.

Sizing the Custodian Manager database

The Discovery Accelerator Custodian Manager database stores details of the following:

- All custodians, including their attributes, email addresses, and group membership.
- Historical record of changes to custodian attributes, addresses, or groups over time.

The Custodian Manager synchronizes custodian details with the corporate directory infrastructure through Active Directory, Domino Directory, or LDAP queries.

Use the following rules of thumb to size the Custodian Manager database:

Base capacity (MB) = ((3.54e)+(0.23ea)+(2.52g)+(0.23ag)+(0.17en)+(0.12et))/1000

Yearly capacity (MB) = ((0.238ace)+(0.24acg)+(0.05m)+(0.05ect))/1000

Where:

e Total employees in directory sources.

a	Average number of email addresses per employee/group.
c	Average number of address changes per employee per year.
g	Total number of groups.
n	Average group membership per employee.
m	Typical total number of group movements per year.
t	Average number of custom attributes (in addition to standard attributes).

Note: This calculation provides a high-level estimate only. It does not take into account the operating system, paging, and log file devices. It also does not include any additional capacity that may be required during product version upgrades.

When the Custodian Manager is synchronizing custodian details, the database typically requires 100 to 300 IOPS.

The following is an example medium scenario. 30,000 employees are managed within the corporate directory, each with three email addresses and two custom attributes. There are a total of 20,000 groups (based upon distribution lists), and each employee is on average a member of 10 groups. Across all groups there are typically 6,000 changes to group membership per year. The email addresses of users and groups tend to change on average once every 5 years (therefore, 0.2 times per year).

Base capacity (MB) = ((3.54e)+(0.23ea)+(2.52g)+(0.23ag)+(0.17en)+(0.12et))/1000

= (3.54*30,000)
+(0.23*30,000*3)
+(2.52*20,000)
+(0.23*20,000*3)
+(0.17*30,000*10)
+(0.12*30,000*2)
/1000

= 249.30 MB

Yearly capacity (MB) = ((0.238ace)+(0.24acg)+(0.05m)+(0.05ect))/1000

= (0.238*3*0.2*30,000)
+(0.24*3*0.2*20,000)
+(0.05*6,000)
+(0.05*30,000*0.2*2)
/1000

= 8.07 MB

Therefore, the Custodian Manager database needs a base of 249.3 MB and an additional 8.07 MB for the first year, plus 20% extra headroom. The total is 308.85 MB.

Sizing the customer database

Use the following rule of thumb to size the core Discovery customer database (excluding analytics tables):

Capacity per year (MB) = ((2.53cis)+(0.44acs)+(1.82is)+(11.6ce)+(0.361cism))/1000

Where:

c	Average cases per year
s	Average number of searches per case
i	Average number of items captured per search
a	Average number of archives searched per case
m	Average number of review marks per captured item
e	Average number of items exported or produced per case per year

Note: This calculation provides a high-level estimate only. It does not take into account the operating system, paging, and log file devices. It also does not include any additional capacity that may be required during product version upgrades. However, this calculation does allow for a small proportion of transient storage required for unaccepted searches.

On a server with 8 GB of RAM, a Discovery customer database data file typically requires 250 to 1,800 IOPS and log file around 500 to 1,200 IOPS. However, this varies depending upon concurrent activities and server specification.

In the following example, there is an average of 100 cases per year, with five searches per case normally searching 50 journal archives. Each search identifies 3,000 items on average (a total of 15,000 per case or 1.5 million items over all cases, per year). Each item has on average a single review mark, and 33% of the items are exported or produced (5,000 items per case).

The calculation to apply is as follows:

Capacity per year (MB) = ((2.53cis)+(0.44acs)+(1.82is)+(11.6ce)+(0.361cism))/1000

= (2.53*100*3,000*5) [Captured item storage]

+(0.44*50*100*5) [Index volume search records]

+(1.82*3,000*5) [Search headroom]

+(11.6*100*5,000) [Export/production storage]

+(0.361*100*3,000*5*1) [Review & marking storage]

/1,000

= 10,174.80 MB

Therefore, the core capacity of the customer database for the first year needs to be 10.175 GB, plus 20% extra headroom. The total is 12.21 GB.

Analytics table considerations

The analytics tables within the customer database require special consideration. When you enable a case for analytics, Discovery Accelerator creates a set of new tables specifically for the case. These new tables contain the original item metadata and content, together with the analysis details. The analytics feature collects the original item content and other metadata from Enterprise Vault and inserts it into the tables. Once the data collection is complete, the additional features introduced by analytics are available to end-users, and these tables are then subject to a much-reduced load.

The analytics tables are broken out into separate file groups, which are created ad-hoc when each case is enabled for analytics. Each file is created on a partition selected in a round robin fashion from the list of available partitions configured for each customer database. In addition, the analytics tables require full-text indexes, which are also created ad-hoc when each case is enabled for analytics. Each set of full-text index files are created on a partition selected in a round robin fashion from the list of available partitions configured for each customer database.

We recommend that the table file groups and full-text indexes are located on the same partitions, which will then be co-located with the table file group.

The Discovery Accelerator analytics tables grow to a similar size as the total converted-to-HTML size of all included original items and their attachments. This also affects the Discovery Accelerator database log file size in a similar manner. Analytics operates on the item converted content provided by Enterprise Vault, so binary data such as image files will not consume space.

The converted content is encapsulated in XML that contains other metadata. Many of the analytics table columns are full-text indexed, which adds further overhead.

Estimating the size of the analytics tables for each enabled case can become complicated when you take into account all the different characteristics of the source data: volume, size distribution, recipient volume and distribution, attachment volume, types and distribution, and number of unique conversations. The Discovery Accelerator source table tblIntDiscoveredItems contains some relevant details, but these are not sufficient to provide estimated sizes.

You must size the file group partitions to encapsulate multiple enabled cases of varied size with potentially varied characteristics. So, you must make a high-level estimate that is based on the maximum number of cases and items in analytics at any time, combined with the average size, and overheads accounting for conversations and row sizes.

So, the Discovery customer database also requires the following additional storage for the Analytics file groups:

Total capacity (MB) = 0.0166097i+0.00025ai

Where:

i	Total number of items (including attachments) in analytics at any time.
a	Average original item size.

Note: This calculation provides a high-level estimate. It does not take into account the operating system, paging, and log file devices. It also does not include any additional capacity that may be required during product version upgrades.

The calculation also assumes the average converted content ratio, based on typical email and office attachments. If the source data contains more text-concentrated documents such as log or text files then the ratio is likely to be higher, or if the data contains more binary files such as images then the ratio is likely to be lower. This could be accommodated by changing the value of 0.00025 by either increasing or decreasing the value by 0.0001.

The following is an example medium scenario. Up to 25 cases that contain an average of 15,000 items per case (375,000 items in total) are expected to be in analytics across all Discovery cases at any time. The original items have an average size of 160 KB including attachments, with 20% of messages containing one or two attachments—therefore making up an extra 30% of items (approximately 487,500 items in total).

Total capacity (MB) = 0.0166098i+0.00025ai

= (0.0166098*487,500)+(0.00025*487,500*160)

= 8,097.28 +19,500

= 27,597.28 MB

So, the total capacity of all file group partitions needs to be 27.6 GB plus 20% extra headroom, which gives 33.12 GB.

Collecting analytics data for a single case is likely to produce sustained periods of very high I/O activity. This varies according to the available physical memory and other concurrent activities. On a database server that provides a throughput of up to 150,000 items per hour, the I/O activity might typically be 1,200 to 2,900 IOPS at the analytics case data file and 100 to 200 IOPS at the customer database log file. The process may take several hours to complete.

You can distribute the load from multiple analytics data collections by using multiple file group locations, which are defined when creating the Discovery customer database. Divide the estimated storage between at least as many independent partitions as there are expected to be simultaneous cases enabled for analytics. So, if multiple cases are concurrently performing analytics data collection, they should be using different partitions.

Note: Concurrent analytics data collection applies very high memory pressure, which is reflected in even higher I/O activity per case. See the memory requirements above.

Once the analytics data collection has completed, the normal I/O activity is much lower. The partition will be selected again for subsequent cases, but this should not have any impact on other cases that are still in analytics and located on the partition.

Security product considerations

The use of security products may be necessary to protect company assets. However, without tuning, some products can be very invasive, and they can considerably impact performance. It is vital to ensure that any security product in use is tuned accordingly, and key disk locations are excluded from real-time scanning.

See the following article for more information:

http://www.veritas.com/docs/000015469

Database tuning

The database server requires additional tuning to ensure that the best performance is achieved. In addition, as the Discovery Accelerator database grows, it requires regular maintenance, monitoring, and potentially tuning according to the usage patterns.

SQL permissions

The facility to create configuration and customer databases with Discovery Accelerator is dependent on the Vault Service account having the SQL Server role of database creator (dbcreator).

The Discovery Accelerator databases are created with the Vault Service account as the DBO. This provides sufficient permissions for normal operation, but certain operations may require additional permissions.

Enterprise Vault, Compliance Accelerator, and Discovery Accelerator provide database roles with which you can control SQL database security.

You can use these roles to grant the Vault Service account only the permissions needed for normal daily operations, and additional permissions when they are required. See the following article for more information:

http://www.veritas.com/docs/000070503

Server settings

In most cases the majority of server settings should be left at their default values. In some cases specific tuning may be required. Particular care should be taken when changing these settings to ensure changes do not severely affect performance.

Memory and processor

The memory maximum value should be tuned to ensure sufficient physical memory is available for the operating system and any co-existing instances, reporting services, or other services. A simple rule of thumb for a single instance would be to set the maximum memory to 75% of the available RAM.

Processor and I/O affinity should only be tuned according to Microsoft advice when absolutely necessary, for example when using multiple SQL Server instances.

Advanced settings

It is very important that the Maximum Degree of Parallelism should be tuned according to the recommendations in the following Microsoft article, taking into account the number of cores, hyperthreading and the number of cores per NUMA node:

http://support.microsoft.com/kb/2806535

See also the following article for more information on hyper-threading:

http://support.microsoft.com/kb/322385

Deploying databases and the impact of model database

SQL Server creates new databases based on the model database, and therefore any changes to the model database will also be present in the Discovery Accelerator databases subsequently created. In addition, some of the default model database values may not appropriate for Discovery Accelerator databases, such as the options for file autogrowth values.

Therefore, after creating any Discovery Accelerator databases, some of the options will need to be checked. The following settings and options can be examined and changed by either opening the database properties in SQL Management Studio or through the use of database views, sp_dboption and ALTER DATABASE (see SQL Server Books Online).

- The database transaction log file autogrowth should be set at around 200 MB – 1 GB for each Discovery Accelerator database. The database data file autogrowth value should be set according to the recommended values in the table below. This can be viewed using SQL Management Studio database properties, or the following SQL statement could be used to gather the sizes (in 8 KB pages):

```
SELECT name, type_desc, physical_name, size, growth,
max_size, is_percent_growth from sys.database_files
```

Recommended auto growth values for databases

Database	Data file autogrowth value
Configuration	50 MB, unlimited growth.
Custodian	Approximately 1 – 2 days growth as per sizing guide. For example 50 MB, unlimited growth.
Customer	Approximately 1 – 2 days growth as per sizing guide. For example 100 MB, unlimited growth.

- The database recovery model should be set according to your backup or high availability strategy. The recommended default is FULL. The current value can be viewed using SQL Management Studio database properties or using the following SQL statement:

```
SELECT name,(CASE recovery_model WHEN 1 THEN 'FULL' WHEN 2
THEN 'BULK_LOGGED' WHEN 3 THEN 'SIMPLE' END) from
sys.databases
```

- The database options should be checked with the expected default values. The current options can be viewed using the SQL Management Studio database properties dialog, or the following SQL statement will show the options set:

```
EXEC sp_dboption <database name>
```

Or for SQL Server 2014 SP3 onwards use:

```
SELECT * FROM sys.databases where name = '<database name>'
```

The model database normally has only the AUTO_CREATE_STATISTICS and AUTO_UPDATE_STATISTICS options set. The Discovery Accelerator databases should have these two options plus Discovery Accelerator will add ARITHABORT. The sys.databases view will also show the Discovery databases have the is_fulltext_enabled and is_broker_enabled set to 1.

Any other options set should be returned to their default using either the SQL Server Management Studio database properties dialog or ALTER DATABASE (see SQL Books Online).

The tempdb database

The tempdb database is a shared resource that is used to store the following:

- User objects (user defined objects, temporary tables and indexes, table variables, tables returned in functions, and system tables).
- Internal objects (work tables for cursors, spool operations, LOB storage, hash join storage, hash aggregate storage, and intermediate sort results).
- Version stores (row versions from updates).

The tempdb database is a temporary database containing transient data that is recreated on each restart of SQL Server, so it will not need to be recovered.

The tempdb database pages move quickly between memory and disk, which becomes a bottleneck if the disks are inappropriate and the configuration is not tuned. You can take the following steps to avoid problems:

- To ensure that tempdb does not become a bottleneck, create one data file per CPU core, taking into account CPU core affinity mask settings. Make each file the same size as below.
- To ensure that tempdb has sufficient capacity to execute expected loads and to prevent tempdb file growth from causing unnecessary I/O or fragmentation, create each data file with minimum sizes of 200 MB and set them to grow by 200 MB.
- Move the tempdb data files to a dedicated striped disk array (RAID-1 or 10) and the log file to a dedicated high speed drive with high transfer rates (RAID-1 or RAID-10). You can use the following SQL statement to move the file location, which will take effect on the next SQL Server restart:

```
USE master;
GO
ALTER DATABASE tempdb
MODIFY FILE (NAME = tempdev, FILENAME =
'E:\SQLData\tempdb.mdf');
GO
ALTER DATABASE tempdb
MODIFY FILE (NAME = templog, FILENAME =
'F:\SQLData\templog.ldf');
GO
```

Warning: The tempdb database is recreated each time SQL Server is started. If the new filenames are invalid, SQL Server will fail to start. When this happens, you must start SQL Server from the console with the –f flag to enter single user mode. Then you can repeat the above procedure with a valid location.

- Do not replicate the tempdb storage device or configure SQL Server replication on the tempdb database. The tempdb database is a temporary database containing transient data that is recreated on each restart of SQL Server, so it will not need to be recovered (use the SIMPLE recovery model). Any form of replication can significantly impact the performance.

Database maintenance

It is recommended that regular backups of all Discovery Accelerator databases are performed. When performing the backups, you may need to stop the Discovery Accelerator services and back up all the other associated Enterprise Vault data to ensure consistency across the backups.

When using Availability Groups, backups should ideally be taken from secondary synchronous replicas.

After the backup has completed, the database log files could be shrunk.

As the databases are used, crucial tables and their indexes fill and can become fragmented. This reduces overall performance and can lead to additional I/O. The solution to fragmented indexes is to reorganize or rebuild very active indexes regularly. Some index maintenance may need to be executed while Discovery Accelerator servers are stopped.

In addition, as some tables grow very large, the frequency of statistics update may become less than desirable. This can result in out-of-date statistics and potentially cause inefficient query plans. Therefore, it is advisable to update the statistics on a regular basis.

Create a regular maintenance plan that includes the following features:

- Performing the Discovery Accelerator database backup
- Reorganizing indexes
- Updating statistics

See the following article on the Veritas Support website for more information:

http://www.veritas.com/docs/000040169

When setting up maintenance plans for databases within Availability Groups the agent job must take account of the local server replica state, and must

also be manually copied to all SQL Servers participating in the AG. For more information see “SQL Agent and maintenance jobs” on page 41.

The plan should keep the database in good condition. However, these procedures are based on a change ratio that, as the tables grow, can gradually degrade performance. Therefore, it is important to perform an additional inspection on a monthly basis to ensure that the databases are in optimal condition (fragmentation and statistics).

The following SQL statement can be used to identify all tables and indexes where the external fragmentation exceeds 5% and the table consists of at least 1,000 pages.

```
SELECT OBJECT_NAME(i.object_id) AS TableName,

i.name AS TableIndexName, phystat.avg_fragmentation_in_percent

FROM sys.dm_db_index_physical_stats(DB_ID(), NULL, NULL, NULL,
'DETAILED') phystat

JOIN sys.indexes i ON i.object_id = phystat.object_id AND
i.index_id = phystat.index_id

WHERE phystat.avg_fragmentation_in_percent > 5 AND
phystat.page_count > 1000
```

Rebuilding indexes can be time-consuming, so low levels of fragmentation should be addressed with a reorganize instead. The following table recommends the most appropriate action to take according to the level of fragmentation.

Fragmentation	Corrective approach
5% – 30%	Reorganize
More than 30%	Rebuild (Can be achieved as online rebuild)

The system stored procedure sp_updatestats (available in Maintenance Plans) should be executed on a regular basis to ensure that all statistics are up to date. However, this may not update all the desired statistics in some circumstances. The best way to perform these optimizations is through monitoring to identify any statistics that may require updating.

A method of identifying whether statistics should be manually updated using UPDATE STATISTICS might be to execute the following query:

```
SELECT object_name(i.object_id) as 'table_name',

        i.name as 'index_name',

        STATS_DATE(i.object_id, i.index_id) as
'last_stat_update',

        (case WHEN rowcnt>rowmodctr THEN abs((1-(cast(rowmodctr
AS float)/cast(rowcnt AS float)))*100) WHEN rowcnt<rowmodctr
THEN abs((1-(cast(rowcnt AS float)/cast(rowmodctr AS
float)))*100) ELSE 0.0 END) as 'percent_change'

FROM sys.indexes i

LEFT JOIN sys.objects o ON o.object_id = i.object_id

LEFT JOIN sys.sysindexes si ON o.object_id = si.id

WHERE o.type='U' AND rowmodctr>0 AND rowmodctr<rowcnt

AND (case WHEN rowcnt>rowmodctr THEN abs((1-(cast(rowmodctr AS
float)/cast(rowcnt AS float)))*100) WHEN rowcnt<rowmodctr THEN
abs((1-(cast(rowcnt AS float)/cast(rowmodctr AS float)))*100)
ELSE 0.0 END) > 5
```

Note: As of SQL 2005, the sysindexes view is provided for compatibility purposes, and the rowmodctr value is a calculated value that may not be accurate. However, it should still be sufficient to indicate outdated statistics.

For example, the following tables (and in particular their indexes or statistics) benefit from specific maintenance. These tables are very active and have a number of indexes to maintain.

Table	Notes
tblSearchItems	Used during searching to store captured items before acceptance. Likely to have an equal or higher write-to-read ratio.
tblIntDiscoveredItems	Used to store all accepted items. Likely to have a higher read-to-write ratio.
tblIntAnalysedItems_<case#>	Used to store all original item content and other item metadata. During analytics case data collection, the related tables have a higher write-to-read ratio. Once this process is completed, the tables have a higher read-to-write ratio.

The statistics for the indexes on the following key tables are automatically updated on a regular basis to ensure optimum query plans, so they do not require any specific statistic maintenance:

- tblIntDiscoveredItems
- tblSearchItems
- tblAddressUser
- tblDiscoveredItemToSearch
- tblHistCasePermission
- tblIntSearches
- tblProductionToDiscoveredItem
- tblItemSetItems
- tblIntMarkHistory
- tblCase
- tblConversations_<case#>
- tblContacts_<case#>
- tblPrincipal
- tblAddress
- tblVaults
- tblHistCaseAddressUser
- tblIntMessageCaptureSummary
- tblIntSecurity
- tblSearchSchedule
- tblSearchResults
- tblItemPolicy
- tblIntAnalysedItems_<case#>
- tblIntConversationItems_<case#>

The database file placement on disk can also lead to file fragmentation, which can degrade performance. If multiple database files reside on a single partition, the database data and log file disks should also be regularly file defragmented to maintain performance.

Note: The database maintenance plan should not include a database or data file shrink. As for the above storage advice above, the database should have pre-allocated storage to avoid file growths and therefore any data file shrinking will be counterproductive.

Database upgrades

The version of SQL Server running Discovery Accelerator databases may need to be upgraded. It is recommended that the following steps are included in the upgrade:

- Perform a backup of all the Discovery Accelerator databases.
- Perform the SQL Server upgrade.
- Check that the Discovery Accelerator databases' compatibility level matches the supported server version (maximum supported level 120).
- Change the Discovery Accelerator databases' recovery models to "SIMPLE".
- Rebuild indexes.
- Update statistics with full scan.
- Return the recovery models to their original settings (recommended "FULL").
- Perform a backup of all the Discovery Accelerator databases.

Note: Upgrading SQL Servers that are part of failover cluster instances or hosting availability groups will require additional planning to ensure the SQL environment remains consistent at all times. Please see Microsoft upgrade instructions and also best practices detailed in MSDN article "Upgrade and Update of Availability Group Server with Minimal Downtime and Data Loss".

Discovery Accelerator product upgrades will most likely require upgrading the database schemas. The upgrade process may require significant additional storage capacity during the upgrade, most notably at the transaction logs and tempdb database. It is recommended the following steps are included in the upgrade:

- Perform a backup of all the Discovery Accelerator databases.
- Temporarily increase the tempdb database storage or enable auto-growth.
- Perform the Discovery Accelerator database upgrades.
- Rebuild indexes.
- Update statistics.

- Perform a backup of all the Discovery Accelerator databases.

Advanced database tuning

There are certain circumstances in which the SQL Server may benefit from additional tuning, and there are also some more advanced approaches to improving database performance which may or may not be beneficial.

Multiple SQL Server instances

Running multiple SQL Server instances is not recommended because of the load profile of Discovery Accelerator. However, if this is unavoidable, you should tune the instances appropriately to reduce any impact between them and ensure that they are not starved of resources.

The SQL Server instance that hosts the Discovery Accelerator customer database should have the minimum memory option configured to ensure that the memory recommended in the “Memory considerations” section on page 16 is dedicated to it.

To ensure that the SQL Server instance is not starved of CPU resource, you may need to consider balancing the available CPUs between the installed SQL Server instances. There are several ways to do this, including the following:

- Using the Windows resource manager to allocate resources to each instance.
- Setting the CPU affinity mask of each SQL Server instance to bind each SQL Server scheduler to specific CPUs.

You can also limit a SQL Server instance to a subset of CPUs without binding the SQL Server scheduler to specific CPUs by using trace flag 8002 with CPU affinity. This should counteract any performance problems that are associated with using CPU affinity. For more information, see the Microsoft Press book on SQL Server tuning.

The hardware architecture needs to be considered to ensure CPUs are not inefficiently allocated to the SQL Server instances, which can be significantly detrimental to performance.

In the case of NUMA architectures, the processors, cores, and memory are arranged in nodes that should be carefully allocated to the SQL Server instances to avoid creating a memory or I/O bottleneck. It may be worth considering using Soft-NUMA to divide the resources with both non-NUMA and NUMA-based hardware. See the Microsoft Press books for more information.

Distributing I/O with multiple database files

Using multiple database files should not be necessary if the Discovery customer database files are placed on an adequately sized array.

Medium and large environments might consider creating multiple data files on separate striped disk arrays (RAID-10). This may improve query performance in large databases by enabling parallel queries in the different files.

Creating multiple data files causes SQL Server to distribute the database tables and indexes between the files, as determined by SQL Server. The use of multiple files lets SQL Server use multiple threads to access data distributed over those files, which may help improve performance by splitting sequential work into parallel activity.

SQL Server controls how the data is distributed between the files. Adding files to an existing large database may not necessarily produce any benefits. The number of files and their location also affects the performance. This depends on the I/O capabilities of the disk array on which the data is located, and placing multiple data files on a single array may cause performance problems. Two main problems need to be avoided: I/O contention and file fragmentation.

I/O contention

Placing too many data files on a single disk array can cause a greater I/O load than the array can manage, which may slow performance. To ensure the throughput can be maintained, you may need to balance the data files between multiple disk arrays.

File fragmentation

File fragmentation can occur when multiple data files reside on a single partition, particularly if the data files do not have a suitable minimum size and so have to grow regularly. Each file growth ends up interleaved between the other data files and causes a significant performance impact. Therefore, each data file should be created with a large minimum size, with perhaps the combined total equivalent of one year’s expected growth. Also ensure that the database is grown by a large fixed size rather than by percentage.

A general rule may be to create as many data files as there are CPUs, taking into account CPU affinity mask settings, and place them on separate disk arrays (RAID-10).

Moving Discovery Accelerator tables or indexes into file groups

Splitting tables and indexes into separate file groups can add considerable complexity to the database management. Any errors during implementation can lead to data loss, damage to the schema, and problems during product upgrades. Therefore, this approach is not recommended.

Rolling over customer databases

You may need to roll over customer databases to remain within the available storage capacity. The database storage requirements are high. To reduce storage costs, you may want to roll over the database periodically to a new customer database (that is, create all new cases in a new database), and then archive the old database when it is no longer active. Alternatively, you can move the old customer database to slower storage and disable the associated Discovery Accelerator customer tasks. This would allow the customer database to be brought online quickly.

Note: You cannot consolidate customer databases. So, after you have created a case in a new customer database, you cannot merge it back into the old customer database.

Co-locating multiple customer databases

Multiple customer databases may be used within a Discovery Accelerator installation as an alternative method of logically separating the data, distributing the I/O load, keeping the database within acceptable sizes for performance, and enabling more concurrent activity between legal discovery users.

It is also possible that multiple Discovery and Compliance Accelerator installations may want to use the same database server.

Hosting more than one very active database on a single database server can become detrimental to performance, particularly if each Discovery customer database is expected to use analytics. You must size the database server appropriately to host the additional load, and the cost of scaling up a single server can become less cost-effective than using multiple database servers. (Scaling up will hit limitations, as well). In this case, consider multiple Discovery Accelerator installations with independent servers.

The customer databases may be divided by factors such as function (Litigation/HR), location, or even by individual legal discovery users. Typically, Litigation and HR customer databases are candidates for hosting on the same SQL Server due to the different use cases.

For example, litigation tends to be very active with large searches of journal archives, whereas HR-style customer databases tend to have low activity with very specific searches of individual mailbox archives.

Note: Additional customer databases may create an impact at the Discovery Accelerator server and the underlying Enterprise Vault infrastructure, which needs to be considered before implementation. For more information, see the chapter “Discovery Accelerator server” on page 57.

High Availability

Discovery Accelerator 14.1 or later supports SQL Server 2014 SP3 and onwards Always On Availability Groups and Failover Cluster Instances, which makes use of Windows Server Failover Clustering. Using these technologies Discovery Accelerator automatically handles failover situations.

This section details the recommended practices for deploying Accelerator databases for High Availability using Always On technologies. This section does not consider any previous SQL Server HA methods such as database mirroring or log shipping or any other server clustering technologies such as Microsoft Cluster Services.

This section assumes that you are familiar with installing and configuring Failover Cluster Instances or Availability Groups, and associated Microsoft recommended best practices.

Purpose

Failover Cluster Instances provide SQL Server instance-level continuity over multiple servers through shared database access. Accelerator databases can be added to Failover Cluster Instances to provide highly available SQL Server and associated database access.

In most cases Failover Cluster Instances provide the most efficient solution for providing high availability.

Availability Groups provide database-level continuity over multiple SQL Servers through database replication. Accelerator databases can be added to Availability Groups to provide highly available database access, and various replica scenarios.

However, Availability Groups introduce complexity, increased storage requirements and many additional configuration considerations to ensure the database reliability and performance are maintained.

Preparation

When preparing the SQL Server environment for Accelerator databases, it is vital to ensure every SQL Server instance participating in the infrastructure is consistent. In particular, ensure the following aspects are identical between each instance:

- Ensure each SQL Server version, service pack and patch level are the same. It is important to take into consideration Microsoft updates to ensure the FCI or AG is reliable. There are a number of critical known issues in Availability Groups with several Microsoft Cumulative Updates. For more information, see the following article:

  https://support.microsoft.com/en-us/kb/3033492
- Ensure each instance is configured with the same collation.
- Ensure each instance is configured with the same logins and permissions for Enterprise Vault.
- Ensure each instance is tuned according to Microsoft recommendations and the section “Deployment” on page 39.
- If using a non-default SQL Server port (not recommended), ensure each instance uses the same port.
- Each SQL Server instance should use the same drive letters, particularly for locations such as database files and backup locations.
- Each SQL Server participating in the HA infrastructure should be the same specification.
- Do not employ multi-subnet failover clustering; this is currently not supported.

Availability Groups

When configuring Availability Groups, there are a number of considerations in respect of Discovery Accelerator databases and their maintenance. The following sections describe these considerations.

Discovery Accelerator 14.1 or later does not support Availability Group read-only routing and does not provide additional scalability options utilizing multiple secondary replicas.

However, read-only replicas can be created for purposes such as offline reporting queries and other HA/DR purposes. If this is required, in most

cases it is recommended to create asynchronous read-only replicas to minimize the impact to Enterprise Vault database performance.

The server sizing will need to accommodate the storage requirements for each database replica.

As with any replication technology, backups are still required to prevent data loss.

It is recommended to use a static IP address for the AG listener.

Deployment

Creating new Discovery Accelerator 14.1 or later databases in a Failover Cluster Instance is simply a matter of creating the new databases using the relevant virtual IP address and ensuring the database files are created on the appropriate storage device.

Creating new Discovery Accelerator 14.1 or later databases in an Availability Group requires initially creating the new databases using the relevant AG listener address and then manually adding the databases to the Availability Group using SQL Server management tools or using PowerShell via the Add-SqlAvailabilityDatabase cmdlet.

Migrating existing Discovery Accelerator databases to a Failover Cluster Instance or Availability Group requires planning to ensure an appropriate migration sequence is developed and database connection configuration settings changed using the Accelerator Manager website. See the Discovery Accelerator *Installation Guide* for more information.

Once databases are created or migrated to Availability Group SQL servers, it is crucial to add them to the Availability Group as soon as possible to ensure the databases are available on failover.

Availability Group considerations

To ensure the performance of the databases is sufficient, the synchronization mode and locality of each replica must be carefully considered.

It is recommended to create primary and synchronous replica databases with automatic fail-over locally to the Discovery Accelerator infrastructure to ensure performance is not degraded.

Monitoring will report on the most recent database backup from all replicas in an Availability Group. The reporting does not distinguish between whether the replica is synchronous or asynchronous.

Discovery Accelerator should always be configured to connect to the Availability Group listener to prevent downtime on failover.

SQL Permissions

The Discovery Accelerator login on each SQL Server hosting the Availability Group will need an additional server level permission, “View Any Definition”.

Analytics tables

The customer database will create additional analytics tables within new data files using the location specified in the Accelerator Manager website. This drive letter location will need to exist on each SQL server in the Availability Group.

Synchronous replica considerations

Each synchronous secondary replica must have committed each transaction log entry before the primary replica can commit the transaction. This can introduce considerable performance impacts, particularly if replicas are located over slow network connections or on lower specification hardware than the primary.

This could also have significant performance impacts in case of automatic fail-over.

To minimize the performance impact of synchronous replicas and ensure automatic failover provides consistent performance, consider the following:

- Ensure the primary and synchronous replicas are co-located on a high-speed network. Do not create synchronous replicas over slow connections.
- Ensure each SQL Server is the same specification.
- Keep the number of synchronous replicas to the absolute minimum required.

As with any replication technology, the replica will also contain any errors introduced in the primary, so a backup strategy is essential to prevent data-loss.

Asynchronous replica considerations

An asynchronous replica will not prevent the primary replica from committing transactions. However, the following should be considered:

- Each additional replica will generally add an overhead, which can increase network and server loads. Therefore, keep the number of replicas to the minimum.
- The asynchronous replica is not guaranteed to be up to date, so should not be relied upon for high availability, which could lead to data loss. However, this might form part of a disaster recovery strategy, ensuring you take into consideration that errors might be replicated.

As with any replication technology, the replica will also contain any errors introduced in the primary, so a backup strategy is essential to prevent data-loss.

SQL Agent and maintenance jobs

SQL Agent jobs are not part of Availability Groups and are not replicated, automatically synchronized or failed over.

Therefore, you must ensure any maintenance agent jobs created are manually copied to each SQL Server that is part of the availability group and set up to be appropriately synchronized.

Each Agent job must first identify whether the local replica is the current primary before performing any maintenance or other tasks. This can be achieved using the SQL function `sys.fn_hadr_is_primary_replica` as a condition in the maintenance job.

Any Agent jobs created by Discovery Accelerator will be automatically created on each replica server and synchronized. However, if a Discovery Accelerator agent job scheduling is manually modified on the primary using SQL Management tools, it will not be replicated by SQL Server to the other agents and therefore must be manually adjusted on each replica.

Backup jobs

An appropriate backup strategy is essential to prevent data loss, in particular, where errors, such as dropping a table, would be replicated.

Typically, an Availability Group should be configured to prefer backups on a secondary replica.

SQL Agent jobs are not part of availability groups and are not replicated, automatically synchronized or failed over. Therefore, you must ensure any agent jobs are manually copied to each SQL Server that is part of the availability group and set up to be appropriately synchronized.

Backup jobs created using the Maintenance Plan Wizard natively use `sys.fn_hadr_is_preferred_backup_replica` function to determine which server to perform backup on. For other backup jobs Microsoft recommends that you use this function as a condition in your backup jobs, so they execute only on the preferred replica.

In addition, the backup job will need to take account of any local server differences such as differing backup locations. This can be simplified by ensuring every SQL server has the same resource configuration.

Deleting or moving databases

Before Accelerator databases can be deleted or moved they must first be removed from the Availability Group.

If a customer database is removed from an Availability Group with cases in analytics, then you may experience issues when you subsequently try to re-enable analytics in the case. For guidelines on how to resolve these issues, see the following article on the Veritas Support website:

http://www.veritas.com/docs/000040498

Note: Reorganizing replicas within an availability group does not require removing the database from the group and can be achieved as per Microsoft best practice advice.

Monitoring the Discovery database server

Regular monitoring will enable a baseline performance profile to be measured and used for comparison over time to identify any trends in performance.

The SQL Server Data Collector can be used to gather and store performance metrics in order to monitor trends over time. Dynamic Management Views can also be used to provide current performance metrics.

Monitor the Discovery Accelerator database server during particular activities to ensure that the environment is performing correctly and allow appropriate database tuning.

Two activities to monitor for benchmark purposes are the following:

- When you accept a large search in an existing database with a large volume of results.
- When you enable a large case for analytics.

Remember that, in isolation, these activities do not represent peak load. In production use, the database is also under load from different activities such as search result processing, export processing, and various update activities.

You can also use Windows Performance Monitor to obtain system and SQL Server statistics. Typically, you should monitor the following counters.

Object	Counters	Instances
PhysicalDisk (and potentially LogicalDisk as well)	Avg. Disk Read Queue Length Avg. Disk Write Queue Length Disk Transfers/sec Avg. Disk Bytes/Transfer Avg. Disk sec/Transfer Disk Bytes/sec Split IO/sec	SQL Data and log file drives
Memory	Page Faults/sec Pages/sec Available Bytes	
Processor	% Processor Time	_Total & All processors
System	Processor Queue Length Context Switches/sec	
SQLServer:Buffer Manager	Buffer cache hit ratio Page life expectancy Procedure cache pages Lazy writes/sec Checkpoint pages/sec	
SQLServer:Access Methods	Page Splits/sec Full Scans/sec	
SQLServer:Memory Manager	Total Server Memory (KB) Target Server Memory (KB)	
SQLServer:Databases	Transactions/sec	_Total & Discovery Accelerator Database
SQLServer:SQL Statistics	Batch Requests/sec SQL Compilations/sec SQL Re-Compilations/sec	
SQLServer:Locks	Average Wait Time(ms) Lock Timeouts/sec	_Total

Object	Counters	Instances
	Lock Waits/sec Number of Deadlocks/sec	
SQLServer:Latches	Average Latch Wait Time(ms) Latch Waits/sec	

CPU and memory

The % Processor Time for the _Total counter indicates overall system activity, but it is worth monitoring the individual processor counters to see if any particular processors are heavily loaded for sustained periods.

If the % Processor Time is generally above 80%, and the Processor Queue length is generally above twice the number of CPUs, then the CPUs are likely to be a bottleneck. However, in SQL Server, high CPU use can be an indication of other factors such as ineffective indexes.

In addition, if the context switches/sec are above 15,000 per CPU when you experience high CPU, it is possible that the server is spending too much time switching between threads of equal priority (but only if the CPU time is above 80%). This may occur for various reasons, as described in the Microsoft books. However, this is most likely to occur with other co-existing software such as multiple SQL Server instances. In this situation, see the section “Multiple SQL Server instances” on page 34.

SQL Server should normally manage memory allocation automatically and avoid situations where memory paging can occur. However, it would be worth monitoring the memory counter Pages/sec, which records the number of hard faults to disk. If there are significant or sustained hard faults, trace the problem to the source. Watching the other SQL Server metrics listed below should also help to indicate if memory capacity is a bottleneck.

Disk

Typically, the disk read/write queue length counters are monitored for evidence of a disk bottleneck. The queues should not normally rise above twice the number of disks in the array. Monitor the average disk sec/transfer to measure the latency of the device accesses. Ideally, this should be approximately 5ms. However, anything in excess of 20ms is of concern.

The use of these counters may not be appropriate when using a SAN, and the hardware vendor’s tools should be used instead.

The Split IO/sec counter can indicate evidence of file fragmentation, and high levels should be addressed with file defragmentation. The remaining counters can be used to measure transfer rates and throughput.

Note: The physical disk counters represent all activity to a physical disk or array, which may contain several partitions (logical drives). The logical disk counters monitor activity to individual logical drives, so they can be used to identify which logical partition is utilizing the disks.

SQL Server

You can monitor the SQL Server performance counters to indicate workload and performance problems. Typically, the following counters should be monitored.

Counter	Notes
Buffer Cache Hit Ratio	Should be above 90% to avoid too much I/O. A lower value may indicate too little server memory.
Total Server Memory, Target Server Memory	If the total server memory exceeds or is equal to the target server memory, there may be an issue with memory pressure. The server may require more memory.
Page Life Expectancy	Indicates how long pages remain in memory. Values that are regularly less than 300 seconds may indicate insufficient memory.
Lazy Writes/sec	Indicates how many times the lazy writer is moving changed pages to disk to free buffer space. This should be quite low. High values indicate high I/O, which more memory will help to reduce.
Page splits/sec	Ideally should be around or below 80 – 100 per second. Index fill factors can be examined to improve this situation.
Batch Requests/sec, Transactions/sec(_Total)	Can indicate the number of SQL requests, and therefore the overall load the SQL Server is handling.

As well as monitoring the system counters, you can extract more detailed information from SQL Server to identify potential performance issues and enable specific tuning.

You can measure the amount of time that is spent waiting for I/O operations using a SQL Server system table-valued function, fn_virtualfilestats. The following query displays the database files and the average time spent waiting on I/O (both read and write):

```
SELECT db_name(database_id),
file_id,io_stall/(num_of_reads+num_of_writes) as 'Avg IO wait
(ms)'

FROM sys.dm_io_virtual_file_stats(db_id('<database_name>'),
NULL)
```

Where <database_name> is the name of a Discovery Accelerator database, or replace the db_id() with NULL to view all databases.

An average value above 20ms suggests that the I/O subsystem could be the source of a bottleneck.

Note: This displays an average since the database was created, and therefore any changes in hardware will not reflect an accurate change in this query. Instead, the io_stall column should be measured at intervals over several hours and the deltas used to determine improvements.

It is essential to measure the index fragmentation for particular key tables, as described in “Database maintenance” on page 29.

You can execute the following SQL statement, which outputs statistics on all tables and indexes where the external fragmentation exceeds 10% and the table consists of at least 1,000 pages.

```
SELECT OBJECT_NAME(i.object_id) AS TableName,

i.name AS TableIndexName, phystat.avg_fragmentation_in_percent

FROM sys.dm_db_index_physical_stats(DB_ID(), NULL, NULL, NULL,
'LIMITED') phystat

JOIN sys.indexes i with (nolock) ON i.object_id =
phystat.object_id AND i.index_id = phystat.index_id

WHERE phystat.avg_fragmentation_in_percent > 10 AND
phystat.page_count > 1000
```

Note: This query may take several minutes to complete, depending on the size of the database.

Useful queries

The following queries can be used on the Discovery customer database to identify search, acceptance, analytics, review marking, and export throughputs.

Identifying overall search throughput per hour

Execute the following query against each Discovery customer database. It outputs the number of hits retrieved, an estimated number of items searched, and the number of indexes searched per hour for the past 24 hours.

```
DECLARE @32bitExchMbx int,@64bitExchMbx int,@32bitExchJnl
int,@64bitExchJnl int,@32bitFSA int,@64bitFSA
int,@32bitDominoMbx int,@32bitDominoJnl int,@64bitDominoMbx
int,@64bitDominoJnl int

SELECT
@32bitExchMbx=50000,@64bitExchMbx=50000,@32bitExchJnl=400000,@6
4bitExchJnl=5000000,@32bitFSA=60000,@64bitFSA=5000000,@32bitDom
inoMbx=50000,@64bitDominoMbx=50000,@32bitDominoJnl=400000,@64bi
tDominoJnl=5000000

select left (convert (varchar, tblRate.enddate,20),14) as
'finished date',count(distinct(searchid)) as 'searches
running',

SUM(sitems) as 'estimated items searched',

sum(tblrate.numhits) as 'hits retrieved and processed',

count(*) as 'index volumes processed',

avg(tblRate.duration) as 'avg search and processing
duration(sec) per index processed',

count(distinct indexserverid) as 'EV index services searched',

sum(distinct(totvaults)) as 'total index volumes included in
running searches'

from

(

select
a.searchid,y.totvaults,a.searchvaultid,a.startdate,a.enddate,a.
numhits,a.statusid,
```

```
"duration" = DATEDIFF(ss,a.startdate,a.EndDate),

"hitrate" = (Cast(a.NumHits as
float)/DATEDIFF(ms,a.startdate,a.EndDate))*1000*3600,

"sitems" = SUM(CASE

WHEN c.Type=9 AND ivs.VolumeType=0 THEN @32bitExchMbx

WHEN c.Type=9  AND ivs.VolumeType=1 THEN @64bitExchMbx

WHEN c.Type=17 AND ivs.VolumeType=1 THEN @64bitExchJnl

WHEN c.Type=17 AND ivs.VolumeType=0 THEN @32bitExchJnl

WHEN c.Type=129 AND ivs.VolumeType=1 THEN @64bitFSA

WHEN c.Type=129 AND ivs.VolumeType=0 THEN @32bitFSA

WHEN c.Type=1025 AND ivs.VolumeType=0 THEN @32bitDominoMbx

WHEN c.Type=1025 AND ivs.VolumeType=1 THEN @64bitDominoMbx

WHEN c.Type=513 AND ivs.VolumeType=1 THEN @64bitDominoJnl

WHEN c.Type=513 AND ivs.VolumeType=0 THEN @32bitDominoJnl

ELSE 0

END),

c.indexserverid

from

        (select
searchid,searchvaultid,startdate,enddate,numhits,indexvolumeset
id,vaultid,statusid from tblSearchVault z with (nolock)

        union all

        select
searchid,searchvaultid,startdate,enddate,numhits,indexvolumeset
id,vaultid,statusid from tblSearchVaultsArchived x with
(nolock)) a

        inner join tblVaults c with (nolock) on
a.vaultid=c.vaultid

        inner join tblIndexVolumeSet ivs with (nolock) on
a.VaultId=ivs.VaultID and
a.IndexVolumeSetID=ivs.IndexVolumeSetID
```

```
        left outer join (SELECT searchid,count(*) as 'totvaults'
from tblSearchVault with (nolock) group by searchid union all
SELECT searchid,count(*) as 'totvaults' from
tblSearchVaultsArchived with (nolock) group by searchid) y on
a.searchid=y.searchid

where a.enddate is not null

and a.enddate > dateadd("hh", -24, getutcdate ())

group by
a.searchid,y.totvaults,a.searchvaultid,a.startdate,a.enddate,a.
numhits,a.statusid,c.indexserverid

) as tblRate

group by left (convert (varchar, tblRate.enddate,20),14)

order by "Finished Date" asc
```

Note the following:

- The query provides an estimated number of items searched based upon the average number of items in each type of archive. This may vary between environments and therefore should be tuned by changing the variable definitions at the start (if a value is unknown, the default should be used). Appropriate values can be obtained from the Enterprise Vault directory database using the following query:

  ```
  SELECT (CASE WHEN indexvolumetype=0 THEN '32-bit' WHEN
  indexvolumetype=1 THEN '64-bit' END) AS 'VolumeType', [9] AS
  'Exchange Mailbox', [17] AS 'Exchange Journal', [33] AS
  'Public Folders', [65] AS 'SharePoint', [129] AS
  'FileSystem',[257] AS 'WSS', [513] AS 'Domino Journal',
  [1025] AS 'Domino Mailbox'

  FROM (select indexvolumetype, Type,IndexedItems from
  IndexVolume iv inner join root a on
  iv.RootIdentity=a.RootIdentity) AS SourceTable

  PIVOT (AVG(IndexedItems) FOR Type IN
  ([9],[17],[33],[65],[129],[257],[513],[1025])) AS PivotTable
  ```

  Note: The query is designed to be executed on Enterprise Vault 10.0 and newer directory databases.

- This query outputs the throughput per hour across all searches rather than an average throughput per search. This is because multiple searches can be executed concurrently, and the overall throughput is shared between the searches. Therefore, the individual search throughputs could be very misleading.
- The query only outputs rows for time periods in which index searches completed, so the results may exclude time periods.
- The number of hits retrieved per hour may vary considerably, as this depends on whether the searches conducted during this period actually found any hits. The number of indexes searched per hour varies depending on how many searches have been conducted in this time period, how many results were retrieved and how many archives (and their indexes) are included in each search.

The following is sample output from searches across thousands of archives.

Finished date: 2011-10-07 10						
Searches running	**Estimated items searched**	**Hits retrieved and processed**	**Index volumes processed**	**Avg search and processing duration (sec) per index processed**	**EV index services searched**	**Total index volumes included in running searches**
1	2,931,104	146,927	6,056	7	8	15,992

Finished date: 2011-10-07 11						
Searches running	**Estimated items searched**	**Hits retrieved and processed**	**Index volumes processed**	**Avg search and processing duration (sec) per index processed**	**EV index services searched**	**Total index volumes included in running searches**
2	12,547,700	1,015,456	25,925	8	8	31,992

Finished date: 2011-10-07 12						
Searches running	**Estimated items searched**	**Hits retrieved and processed**	**Index volumes processed**	**Avg search and processing duration (sec) per index processed**	**EV index services searched**	**Total index volumes included in running searches**
2	10,164	5,370	21	7	7	16,010

Identifying overall search acceptance throughput per hour

Execute the following query against each Discovery customer database. It outputs the number of items accepted per hour for the past 24 hours.

```
select left (convert (varchar, capturedate,20),14) as 'Accept Date',
    count(*) as 'Items Accepted in hr'
from tblIntDiscoveredItems with (nolock)
where capturedate > dateadd("hh", -24, getutcdate ())
group by left (convert (varchar, capturedate,20),14)
order by "Accept Date"
```

Note the following:

- This query only outputs rows for time periods in which items were accepted, so the results may exclude time periods.
- The number of items accepted per hour may vary depending on how many searches were accepted in a given hour and how many items were included in each search.

The following is sample output:

Accept Date	Items Accepted/hr
12/03/2009 15:	199991
12/03/2009 16:	137817
12/03/2009 17:	1462274
13/03/2009 09:	200029

Identifying export throughput per hour

Execute the following query against each Discovery customer database. It outputs the number of items exported per hour for the past 24 hours.

```
select left (convert (varchar, tblRate.completiondate,20),14)
as 'Finished Date',

sum(tblrate.numitemsproduced) as 'items exported by exports
completed within hour',

avg(tblrate.numitemsproduced) as 'average items exported per
export completed within hour',

avg(tblRate.duration) as 'avg duration (s)/export',

avg(tblRate.exprate) as 'avg items exported per export/hr',

avg(tblRate.overlapping) as 'avg overlapping exports',

avg(tblRate.exprate)*(avg(tblRate.overlapping)+1) as 'estimated
overall export throughput/hr'

from

(

select a.productionid,a.numitemsproduced,b.name,

"duration"=datediff(ss,a.createdate,a.completiondate),

"exprate"=(Cast(a.numitemsproduced as
float)/DATEDIFF(ms,a.createdate,a.completiondate))*1000*60*60,

a.CreateDate,
```

```
"overlapping" = (select count(*) from (SELECT b.productionid
from tblProduction b with (nolock) where
(b.completiondate>a.createdate AND
b.completiondate<a.completiondate) union select d.productionid
from tblProduction d with (nolock) WHERE
(d.createdate>a.createdate AND d.createdate<a.completiondate) )
as tblSubQ),

a.completiondate

from tblProduction a with (nolock)

inner join tblstatus b on a.statusid=b.statusid

where a.completiondate is not null

and a.completiondate > dateadd("hh", -24, getutcdate ())

) as tblRate

group by left (convert (varchar, tblRate.completiondate,20),14)

order by "Finished Date" asc
```

Note the following:

- This query only outputs rows for time periods in which items were exported, so the results may exclude time periods.
- The number of items exported per hour may vary depending on how many exports have been conducted in a given hour and how many items were included in each export.

The following is sample output:

Finished Date	Items exported by exports completed within hour	Average items exported per export completed within hour	Avg duration (s)/export	Avg items exported per export/hr	Avg overlapping exports	Estimated overall export throughput/hr
13/03/2009 12:	20000	20000	190	378901	0	378901
13/03/2009 20:	20000	5000	393	46704	2	140114

Identifying review marking throughput per hour

Execute the following query against each Discovery customer database. It outputs the number of items marked per hour for the past 24 hours.

```
select left (convert (varchar, markdate,20),14) as 'Mark Date',
count(*) as 'Total Items Marked in hr'
from tblIntMarkHistory with (nolock)
where markdate > dateadd("hh", -24, getutcdate ())
group by left (convert (varchar, markdate,20),14)
```

Note the following:

- This query only outputs rows for time periods in which items were marked, so the results may exclude time periods.
- The number of items marked per hour may vary depending on how many reviewers were working in a given hour, their working practices, and therefore how many items were marked.

The following is sample output:

Mark Date	Total Items Marked/hr
27/03/2009 08:	26238
27/03/2009 09:	26357

Identifying overall analytics ingestion throughput per hour

Execute the following query against each Discovery customer database. It outputs the number of items retrieved from Enterprise Vault and inserted into the analytics tables per hour for the past 24 hours.

```
DECLARE @sCmd nvarchar(1000)
DECLARE @curTable int, @LastTable int
DECLARE @tbls TABLE (rownum int IDENTITY(1,1) NOT NULL,
TableName nvarchar(255) NOT NULL,OwnerName nvarchar(255) NOT
NULL )
```

```
CREATE TABLE #Results (UpdatedDate varchar(255), MsgAttHr int,
MsgHr int, MsgAttSizeHr int)

INSERT @tbls (TableName, OwnerName)  SELECT object_name(o.id),
u.[name]     FROM sysobjects o INNER JOIN sysusers u ON u.uid =
o.uid   WHERE o.type='U'  and o.[name] like
'tblIntAnalysedItems%'

SET @LastTable = @@ROWCOUNT

SET @curTable = 0

WHILE (@curTable < @LastTable )

BEGIN   SET @curTable = @curTable + 1

                      SELECT @sCmd = N'select "UpdatedDate" =
left (convert (varchar, lastupdated,20),14),"MsgAttHr" = count
(*),"MsgHr"=sum(case when parentrowid is null then 1 else 0
end),"MsgAttSizeHr" = sum (size) from ' + TableName + N' with
(nolock) where lastupdated is not null and lastupdated >
dateadd("hh", -24, getutcdate ()) group by left (convert
(varchar, lastupdated,20),14) order by "UpdatedDate" asc'

                       FROM @tbls

                       WHERE rownum = @curTable

           INSERT #Results   EXEC sp_executesql @sCmd

END

SELECT      [Ingest Date]=UpdatedDate,

            [Messages and attachments/hr]=SUM(MsgAttHr),

            [Top level messages/hr]=Sum(MsgHr),

            [Messages and attachments
size(kb)/hr]=Sum(MsgAttSizeHr)

FROM #Results a

Group By a.UpdatedDate

ORDER BY a.UpdatedDate ASC

DROP TABLE #Results

GO
```

Note the following:

- This query only outputs rows for time periods in which items were ingested, so the results may exclude time periods.
- The number of items ingested per hour may vary depending on how many cases were concurrently enabled for analytics, and the size of each case.

The following is sample output:

Ingest Date	Messages and attachments/hr	Top level messages/hr	Messages and attachments size(kb)/hr
14/04/2009 13:	56788	43715	10032449
14/04/2009 14:	166727	127982	29599214
14/04/2009 15:	139046	106917	25096562
14/04/2009 16:	129148	99256	22630495

Chapter 3

Discovery Accelerator server

The Discovery Accelerator server directs the flow of data between Enterprise Vault and the Discovery Accelerator database, and manages the interaction with end-users. End-user actions usually result in Discovery Accelerator database activity, but may also involve Enterprise Vault.

The Discovery Accelerator server can potentially be very resource-intensive, and so it may not best co-exist with other applications or services. In a small environment, the server can potentially co-exist with the Enterprise Vault services, but you need to pay careful attention to the hardware specification.

In most situations, you should not host other applications or services with Discovery Accelerator. The Discovery Accelerator server may need to be scaled out, depending on the expected use.

Hardware requirements

The Discovery Accelerator service concurrently coordinates significant activities and therefore benefits from multi-processor servers. The flow of data through the Discovery Accelerator service can cause high memory utilization, and a multi-user environment can cause the customers' service to consume resources. In addition, the Discovery Accelerator pre-fetch cache and export features can result in very high I/O loads.

We recommend that the Discovery Accelerator database server and application server are connected through gigabit network technology. It would also be highly beneficial if the Enterprise Vault infrastructure can made be available to Discovery Accelerator through gigabit technology.

Server considerations

The Discovery Accelerator service creates many threads of execution to manage different concurrent activities. A multiprocessor server is essential to ensure that all these concurrent activities can function with reasonable performance. A very small environment may require Discovery Accelerator to co-exist with Enterprise Vault. If co-existing with Enterprise Vault, at least eight physical cores are recommended.

All other installations should have a dedicated Discovery Accelerator server with at least four processor cores available to the operating system. Either four physical CPUs or a combination of multi-core and multi-CPU can be used, but server sizing must not be based upon hyper-threading. If analytics or more than one customer database is to be used then at least eight processor cores should be installed.

The Discovery Accelerator service can require a high memory capacity, particularly during searching and analytics data collection.

A Discovery Accelerator server should have 16 GB of RAM available to the operating system to handle the high memory requirements of Discovery Accelerator. If multiple customer databases are to be used, the memory should be increased by at least 4 GB per active customer database.

The Discovery Accelerator server may need to be scaled out, as discussed in the following sections, to balance the load between customer tasks, reviewing, and analytics data collection. Additional servers for customer tasks and Analytics purposes should be sized as above.

If the online preview and reviewing facilities are expected to host more than 300 concurrent reviewers, the Discovery Accelerator customers' service should be scaled out to additional servers in a network load-balanced arrangement. Each additional server can consist of a 2-core server with 8 GB of RAM, which should typically support up to 300 lightweight reviewers. However, this depends on the working practices of those users, and more active users may require 16 GB of RAM and a 4-core server.

When scaling out Discovery Accelerator, each customer database requires a customer tasks service, each of which can only reside on one server.

Storage considerations

The Discovery Accelerator service that runs the customer tasks requires storage to be arranged for two different purposes: the pre-fetch cache, and the export location.

By default, the pre-fetch cache is limited to 1 GB disk capacity, but you may need to increase this to make the best use of it. For more information, see “Pre-fetch cache tuning” on page 63.

The export location storage requirements vary depending on the total number of anticipated exports at any one time and the total size of all the original items for each export. Additional space is required for overheads during processing of Exchange message production to PST. The storage needs to be 25% larger than the total export size during processing. However, once complete, the additional space is released.

A single export to native format generally produces 200 – 300 IOPS. However, exporting to Exchange PST incurs overheads which can produce between 900 and 1,600 IOPS. In addition, concurrent exports can increase this load considerably. Therefore, it may be worth providing multiple export locations to distribute the I/O load. For more information, see “Export and production tuning” on page 67.

The type of storage and interfaces that you use must ensure that the storage for either of these purposes does not become a bottleneck. LAN-based storage should not be used for the pre-fetch cache and may not be suitable for the export location. The best devices are local storage, direct attached storage, or partitions on appropriately sized storage area network (SAN).

Both the pre-fetch cache and export location require good random access or a high number of IOPS, so a striped array of many disks should be used (using hardware-based RAID rather than software-based RAID). Due to the size and volume of data involved with these activities, the RAID also requires good transfer rates and therefore high-speed disks should be used. These two sets of files need to be located on different arrays to ensure that the export throughput and pre-fetch cache benefits can be maintained.

To achieve redundancy on the striped arrays while maintaining performance, consider the RAID scheme carefully. RAID levels 5 and 6 are popular and cost-effective methods of achieving redundancy while maintaining striped disk read performance. However, writing incurs a cost of four to six physical writes per operation. This may be acceptable for the pre-fetch cache location, which should have a higher proportion of reads, but a poorly sized RAID-5/6 implementation can significantly reduce the performance of write-intensive exports. Correctly sizing a RAID-5/6 implementation to maintain write performance may become more costly than RAID-10, and therefore a RAID-10 scheme must be considered for the export location.

Redundancy may not be an issue for the pre-fetch cache, which contains a copy of the original data. Therefore, no data loss occurs if the cache is lost. However, only new items are added; the missing items are not repopulated unless Discovery Accelerator is instructed to rebuild the cache.

In the case of local or direct attached storage, multiple controllers supporting multiple channels should be used to distribute the load between the pre-fetch cache and export locations and provide sufficient throughput. The controllers

should also provide a battery-backed read and write cache to aid performance.

When you select storage, consider the advice in "Storage" on page 11, and create and align partitions as advised.

A Discovery Accelerator server should typically have the following partitions:

- System drive (RAID-1)
- Pre-fetch cache (RAID-5 or RAID-10 striped array with many disks)
- One or more export locations (RAID-10 striped array with many disks)

Note: Exclude the export locations from anti-virus scanning. Otherwise, file locking issues may prevent the export from completing.

Custodian Manager tuning

You can install a single Custodian Manager database in a Discovery Accelerator installation. However, all Discovery Accelerator customer databases in the installation share the Custodian Manager database.

The Custodian Manager is managed through an IIS Web site, which is automatically installed on the server on which the Custodian Manager is created. This Web site has a low user load and should not have any impact on other activities on the server.

The Custodian Manager customer tasks service typically has little impact on the other Discovery Accelerator services and requires little specific tuning. However, a large customer that expects any co-existing customer databases to be under heavy load may need to scale out the Custodian Manager to a dedicated server. In this situation, it is likely that the customers' service and analytics' service need to be scaled out as well, and therefore the Custodian Manager could be co-located on one of those servers.

Discovery Accelerator customers service tuning

The Discovery Accelerator customers service provides the client access service. Client/server communications are encrypted and digitally signed, by default using port 8086. The user token from the client workstation can only be used by the server for identification and not impersonation.

The customers service benefits from tuning to the specific environment requirements. If the online preview and reviewing facilities are to be used in

large environments, the customers service should be hosted on a separate server.

Each customers service should support up to 300 concurrent users. If you need to support more than 300 users, add additional servers with appropriate Network Load Balancing implemented to direct the traffic between the servers.

Network sockets

You must adjust the Windows TCP/IP socket parameters to provide the .NET environment with adequate network sockets at a sufficient reusable rate. To do this:

- Locate the following key in the Windows registry:

  ```
  HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\
  Parameters
  ```

- Update the following values, or create them if they do not already exist:

Name	Type	Default	Recommended (decimal)
TCPTimedWaitDelay	DWORD	120	60

- Extend the dynamic port range using the following netsh commands to change the start port to 1025 (default 49152) and number of ports to 64510 (default 16384):
  - `Netsh int ipv4 dynamicport tcp start=1025 num=64510`
  - `Netsh int ipv4 dynamicport udp start=1025 num=64510`

Note: Setting the MaxUserPort registry key to 65535 will revert to pre-Windows 2008 functionality and set the start port to 1025 and number of ports to 64,510. However, it is recommended to use the netsh commands.

Discovery Accelerator analytics service tuning

The Discovery Accelerator analytics service benefits from tuning to the customer's specific environment. Host this service on one or more additional servers if one or both of the following applies:

- You are likely to enable cases for analytics on a regular basis.
- You anticipate that you may want to enable several cases for analytics concurrently (such as medium or large customers).

Scaling out the Discovery analytics service improves performance and prevents contention with core Discovery activity. However, concurrent analytics data collection should be avoided where possible due to the impact at the database server. For more information, see the “Database platform” chapter on page 15.

Retrieval thread tuning

Even with a well-specified database server, more than six active Enterprise Vault Content Management (ECM) retrieval threads are ineffective due to the performance of inserting the data into the database. In addition, by default, an individual source Enterprise Vault server is limited to using up to two available threads. So, if the case search results to be ingested are from fewer than three source Enterprise Vault servers, analytics may not take advantage of the potential throughput. The per-server thread limit prevents an individual Enterprise Vault server from becoming overloaded with retrievals.

If the additional load upon individual Enterprise Vault servers is of no concern, it is beneficial to set both the maximum number of ECM threads and the thread limit per Enterprise Vault server to six. This ensures that however many source Enterprise Vault servers are required to service the ingestion, the optimum throughput is achieved.

However, a poorly specified database server impacts the performance such that six threads may not meet expectations.

Change the following Analytics Data Collections settings in the System Configuration:

Name	Default	Recommended
Maximum number of threads for fetching data from Enterprise Vault servers (hidden)	10	6
Maximum number of threads per vault server to retrieve data from Enterprise Vault (hidden)	2	6

Discovery Accelerator customer tasks tuning

The Discovery Accelerator service benefits from tuning to the customer's specific environment. The underlying Enterprise Vault infrastructure and expected patterns of activity can impact the overall Discovery Accelerator performance.

Each customer database and the Custodian Manager database has its own customer tasks. You may need to scale these out if the individual customer databases are each under load from a high volume of searches and exports.

Pre-fetch cache tuning

The pre-fetch cache runs according to a nightly schedule to download original items from the Enterprise Vault storage service for all recently accepted search results to the server running the customer tasks. This can impact the performance of the Enterprise Vault services during the scheduled period, so it needs to be carefully considered before implementation.

The pre-fetch cache is used by the online preview and reviewing facilities and the export and production features. Retrieving original items from the cache can perform significantly better than from the Enterprise Vault storage service, and it can prevent the storage service from being heavily utilized during peak times of activity.

The pre-fetch cache mechanism can download two parts of the original item: the HTML version displayed for reviewing, and the native format used for exporting. By default, the service only downloads the HTML component for reviewing purposes.

Normally, it is not worth enabling the native format download for export purposes. This is because the cache only retains items for five days, and exports are unlikely to occur within this timeframe. Furthermore, the additional storage overhead results in less capacity being available for the HTML items for review.

During the pre-fetch cache update window, the pre-fetch mechanism downloads 200 items per 30 seconds by default. This prevents overloading on smaller systems, but on larger systems the batch size may need to be increased. This depends on how many items can be downloaded within 30 seconds. If the download takes just over 30 seconds, the software waits until the next 30-second interval before starting to download again. Therefore, the batch size may need to be adjusted to between 200 and 1000 to gain the best throughput.

Change the following Item Pre-fetch Cache settings in the System Configuration:

Setting	Default	Recommended
Cache enabled	Off	On
Cache location	—	Dedicated striped array.
Start prefetching time of day	20:00	Set time to avoid other scheduled activity.
End prefetching time of day	05:00	Set end time to prevent overrunning.
Cache maximum size (Mbytes)	1000	Sufficient to include 5 days of search results. This can be determined by monitoring the event log which will indicate when the cache is full.

Change the following Item Pre-fetch Cache (Advanced) settings in the System Configuration:

Setting	Default	Recommended
Prefetch Native format	Off	Off
Item batch size (hidden)	200	Value to match throughput during 30 seconds.

Search tuning

Discovery Accelerator searches by default up to ten index volumes concurrently per Enterprise Vault indexing service. Discovery Accelerator has a customer-wide soft-limit of 100 search threads, at which point the threads are then allocated equally between all Enterprise Vault index servers searched up to 100 index servers. Beyond this, each Enterprise Vault server is then allocated one thread.

Discovery Accelerator 14.2 and later supports searching 64-bit Elasticsearch indexes and legacy 32-bit and 64-bit non-Elasticsearch indexes available in Enterprise Vault 14.2 and later.

Depending on the Enterprise Vault server specifications and type of indexes, the default search thread settings may not be optimal. An improper value can potentially overload the indexing servers and storage devices, and the associated contention can significantly degrade the performance.

The 32-bit and 64-bit indexing engines have different performance characteristics and may require different tuning. The Discovery Accelerator search threads are a customer-wide setting regardless of index type or location. Therefore, if the Enterprise Vault servers contain a mixture of mix of Elasticsearch index volume and legacy 32-bit and 64-bit non-Elasticsearch index volumes., it may be beneficial to consider upgrading the 32-bit index volumes that will be searched to take full advantage of the 64-bit Elasticsearch indexing engine.

See "Enterprise Vault indexing service considerations" on page 75 for more information.

You may need to adjust the number of search threads to balance the load evenly or take better advantage of well specified servers and dedicated index groups.

Increasing the number of search threads per Enterprise Vault server might help increase the index volume iteration rate, provided that at each Enterprise Vault server there is sufficient memory and the indexing storage does not become a bottleneck.

The optimum value for the number of threads depends on the specification of all Enterprise Vault index servers. An environment containing a mixture of different specification Enterprise Vault servers may need to be tuned to the average specification.

To determine the best value, make several benchmark searches representing typical anticipated searches. This may consist of a few large journal archives, or a large number of mailbox archives, or a combination of the two. Repeat the same search with different numbers of threads, starting with a low value such as two threads and increasing in steps of five. Use the SQL query in “Identifying overall search throughput per hour” on page 47 to measure the performance.

When an index volume is searched Discovery Accelerator will conduct any other searches queued for that index volume to make efficient use of the index whilst in memory. The number of consecutive searches needs to be balanced to ensure reasonable sharing between searches of other index volumes and to prevent resource contention. Environments containing 32-bit index volumes should limit the number of consecutive searches.

Change the following Search setting in the System Configuration, and then restart the customer tasks.

Setting	Default	Recommended
Number of Vault search Threads	10	64-bit and mixed 32-bit/64-bit: 10 64-bit in dedicated index groups: Start at 10 and tune to environment
Maximum Number of consecutive searches on same index	0	32-bit and mixed 32-bit/64-bit: 100 64-bit only: 0

If you increase the number of search threads above 10, you may need to increase the Enterprise Vault 32-bit index server process limits on every Enterprise Vault server. However, you must take the server specification into consideration, as described in "Enterprise Vault indexing service considerations" on page 75.

Add the following registry keys to each Enterprise Vault server hosting 32-bit index volumes, and then restart the indexing service:

HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\KVS\Enterprise Vault\Indexing

Setting	Default	Recommended
ThresholdIndexServers	20	10 + Number of vault search threads (but no fewer than 20).
MaxIndexServers	30	20 + Number of vault search threads (but no fewer than 30).

To optimize search result processing, ensure that the following Search setting in the System Configuration is enabled:

Setting	Default	Recommended
Use sequence numbers for searches	On	On

If you can use date ranges within searches, you can further improve the efficiency of the searches by enabling more selective searches that exclude indexes that do not contain documents within the given date range. To take advantage of this feature, enable the setting below. This setting does not have any impact on searches that you conduct without specifying a date range. Such searches continue to search all related index volumes.

This feature relies upon directory synchronization with Enterprise Vault to update the Discovery Accelerator database with date range details for each index. This occurs by default every six hours. However, if directory synchronization is disabled, or the update interval is increased beyond one day, there is a risk that a search may miss results if an index contains newer entries than the date range records in the Discovery database.

To enable date range optimization, change the following Search settings in the System Configuration. However, ensure that the directory synchronization is enabled and the update interval is at least once per day.

Setting	Default	Recommended
Optimise searches based on oldest and youngest items	Off	On

The Discovery Accelerator search page provides a feature-rich search that lets you find documents that are based on a large number of criteria. However, the search allows an open search that will retrieve every item in every index searched. Therefore, end-users should be advised to ensure searches are as specific as possible, particularly using date ranges if possible. Mandating fields can be an effective method of ensuring that no open searches are triggered (see the Search settings called “Require <field> to be specified” in the System Configuration).

Export and production tuning

Each export or production is allocated multiple threads. Each of the threads downloads original items either from the pre-fetch cache, if available, or directly from the Enterprise Vault storage services. The use of multiple threads prevents Discovery Accelerator from waiting for any slower Enterprise Vault storage services, and provides optimal throughput for a single export across multiple Enterprise Vault servers.

Discovery Accelerator has a customer-wide default limit of 4 concurrent exports and 100 export threads. The limit can help prevent the Discovery Accelerator server from being overloaded by multiple concurrent exports. By default, each export starts 25 threads.

The thread limit and threads per export can be adjusted to improve throughput or concurrency and/or reduce the effects of export on other Discovery Accelerator customer functions or the Enterprise Vault storage services.

The defaults may be to be too high for many environments, and may be worth considering lowering to 10 threads per export and then tuning according to the factors below.

To determine the best thread values for your devices, make several benchmark exports representing typical anticipated exports in order to do the following:

- Ensure that the export destination disks can sustain the load. Adjust the thread limit and potentially adjust the threads per export to allow the desired number of threads within the 4 concurrent export limit.
- Ensure that the system does not spend most of its time idle waiting for the Enterprise Vault storage services. If many threads are waiting then the threads per export may need to be increased. The thread limit may also need to be adjusted to accommodate the threads per export.
- Ensure better coexistence with searching and end-user activity by reducing CPU utilization. Adjust the thread limit and potentially adjust the threads per export to allow the desired number of threads within the 4 concurrent export limit.

Take care to ensure that the system is not tuned for infrequent peak loads, which could result in reducing the throughput during normal usage.

Change the following setting in the System Configuration:

Setting	Default	Recommended
Number of Production Threads Per Production Run	25	10 or based upon hardware and activity profile.
Total Number of Production Threads Per Customer	100	40 or based upon hardware and activity profile; should be a multiple of the above production threads.

Architecture considerations

Medium and large customers may have distributed their Enterprise Vault infrastructure over multiple sites with varying levels of interconnectivity. In addition, Discovery Accelerator may need to be used by different parts of an organization located across different sites.

The legal discovery functionality is available through the Discovery Accelerator customers' service, which only generates low levels of traffic between the server and Discovery client, and therefore should allow the server to be utilized throughout the organization.

However, any exported data will tend to be very large, and in most situations it will be located on the Discovery Accelerator server. Transporting these large volumes of data across inter-site network links may become an issue. Multiple Discovery Accelerator installations may therefore be needed with additional Discovery servers behind slow or expensive network links.

The majority of work occurs between the Discovery Accelerator server and database server (which should be co-located within a site) and the Enterprise Vault infrastructure, which may be distributed.

The traffic between Discovery Accelerator and Enterprise Vault varies depending on the size of the Enterprise Vault implementation and the activity patterns of the legal discovery team. Background activities such as Enterprise Vault and Active Directory synchronization generate short and infrequent periods of high traffic, depending on the number of mailboxes.

The main source of traffic is during export, when original archived data is downloaded from storage services, and during searching, when results are returned from indexing services. The search result data is transported as XML, but the export data is transported as large volumes of binary data.

These activities may cause issues between sites with slow or expensive network links, and therefore may warrant local Discovery Accelerator servers to split the load. This would mean that a search of the entire organization may require the use of several different installations, but it is likely that most searches would be local to the site in which it resides.

Note: Discovery Accelerator can search for and retrieve data (for review, export, and analytics) across independent Enterprise Vault installations.

Using multiple customer databases

You can use multiple customer databases to logically separate the data. This is an alternative method of distributing the I/O load, keeping the database within acceptable sizes for performance, and enabling more concurrent activity between legal discovery users. This approach also has the advantage of sharing the Custodian Manager service between all of the customer databases. There is no need to duplicate custodian information between each customer database.

Small or medium environments may consider this approach to be a more manageable method of scaling the Discovery Accelerator database. Larger environments may want to consider multiple Discovery Accelerator installations due to the impact at the database server.

The customer databases can be divided by factors such as function (Litigation/HR), location, or even by individual legal discovery users. When considering hosting multiple databases on a single SQL Server, pay careful attention to the expected level of activity. Hosting two very active databases on a single database server can become detrimental to performance, especially when analytics is used. In this case, you should consider multiple Discovery Accelerator installations.

Typically, Litigation and HR customer databases are candidates for hosting on the same SQL Server due to the different use cases. For example, litigation tends to be very active with large searches of journal archives, whereas HR-style customer databases tend to have low activity with very specific searches of individual mailbox archives.

However, adding customer databases increases the load at the Discovery Accelerator server, particularly if analytics is to be used. Each customer database has its own dedicated customer tasks service and analytics service. This can result in contention at the Discovery Accelerator server if the customer databases are heavily loaded with searches, exports and analytics data collection. In this situation, the customer task and analytics service for each customer database may need to be scaled out to separate servers.

You also need to pay careful attention to which Enterprise Vault servers are used by the separate customer databases. In the situation where multiple customer databases can access the same set of Enterprise Vault servers, the infrastructure must be capable of supporting the potentially high-load overlapping activity.

Each customer database can independently search and retrieve large volumes of data for review, export, and analytics. This may cause excessive resource utilization at the Enterprise Vault servers. It is essential that the customer databases balance the load by reducing the search and export threads, as described previously. The number of ECM retrieval threads may need to be reduced in each analytics service.

Using multiple Discovery installations

You may require multiple Discovery Accelerator installations in instances where multiple Enterprise Vault sites are installed in physically disparate locations, perhaps with slow or no network links or across boundaries of legal jurisdiction. Also, as the demands of the Discovery Accelerator environment grow, the volume of data and levels of activity between Discovery users may affect performance.

You may be able to achieve the following by installing multiple independent Discovery Accelerator installations on independent hardware:

- Meeting topological or legal boundaries.
- Distributing the load.
- Keeping each environment within acceptable sizes for performance.
- Enabling more concurrent activity between legal discovery users.

However, this has the disadvantage that each installation requires its own Custodian Manager. This decentralizes custodian management and increases the hardware requirements.

This approach may be required by medium or larger environments to scale out Discovery Accelerator, or to meet geological and legal boundary demands.

You can divide the separate installations by factors such as department function (Litigation/HR), geographical location, or even by individual legal discovery users. Each environment requires its own set of dedicated servers, including a database server and Discovery Accelerator server.

Pay careful attention to which Enterprise Vault servers can be used by the separate Discovery Accelerator installations. In the situation where multiple Discovery Accelerator installations are able to access the same set of Enterprise Vault servers, the infrastructure must be capable of supporting the potentially overlapping high activity.

Each Discovery Accelerator environment can independently search and retrieve large volumes of data for review, export, and analytics. This may cause excessive resource utilization on the Enterprise Vault servers. It is essential that the Discovery Accelerator services balance the load by reducing the number of search and export threads as described previously. The number of ECM retrieval threads may need to be reduced in each analytics service.

Using multiple eDiscovery products

Using Discovery Accelerator alongside other eDiscovery products that can independently search Enterprise Vault index volumes and retrieve large volumes of data for review, export, and analytics, may cause excessive resource utilization on the Enterprise Vault servers.

It is essential that each eDiscovery product, including Discovery Accelerator, is tuned to balance the load by reducing the search and export threads as described previously. The number of ECM retrieval threads may need to be reduced in each analytics service.

For example, if ADSS and Discovery Accelerator are both deployed, it may be necessary to reduce the number of concurrent ADSS searches to 5 and Discovery Accelerator search threads to 5 to balance the overall load.

Monitoring Discovery Accelerator

Monitor the Discovery Accelerator server during the following activities to ensure that the environment is performing correctly:

- During a large export to native format (for example 100,000 results).
- During a search that includes many different journal style vaults (a large volume).
- During analytics case data collection.

Use Windows Performance Monitor to obtain system statistics. You should typically monitor the following counters.

Object	Counters	Instances
PhysicalDisk (and potentially LogicalDisk as well)	Avg. Disk Queue Length Disk Transfers/sec Avg. Disk Bytes/Transfer Avg. Disk sec/Transfer Disk Bytes/sec Split IO/sec	Export location drives
Server	Files Open	
Memory	Page Faults/sec Pages/sec Available Bytes	

Object	Counters	Instances
	Free System Page Table Entries	
Processor	% Processor Time	_Total
Network Interface	Output Queue Length Packets/sec	All instances
System	Processor Queue Length Context Switches/sec File Data Operations/sec Processes	

CPU and memory

The % Processor Time for the _Total counter indicates overall system activity. If the CPU is consistently above 80%, the server may start to indicate performance issues.

If the % Processor Time is generally above 80%, and the Processor Queue length is generally above twice the number of CPUs, then the CPUs are likely to be a bottleneck.

In addition, if the context switches/sec are above 15,000 per CPU when you experience high CPU, it is possible that the server is spending too much time switching between threads of equal priority (but only if the CPU time is above 80%). This may occur for various reasons, but it is most likely to occur when there are too many threads running concurrently. For example, the server may be overloaded with many concurrent exports with too many threads (see “Export and production tuning” on page 67), combined with other activities such as searching and on-line reviewing (consider scaling out the customers service).

It would be worth monitoring the memory counter Pages/sec, which records the number of hard faults to disk. If there are significant or sustained hard faults, trace the problem to the source.

Disk

The disk counters are only relevant if the pre-fetch cache and export file locations are locally attached. If the export file locations are on a remote server, monitor that server for disk activity.

Typically, the disk read/write queue length counters are monitored for evidence of a disk bottleneck. The queues should not normally rise above twice the number of disks in the array.

Monitor the average disk sec/transfer to measure the latency of the device accesses. Ideally, this should be approximately 5ms. However, anything in excess of 20ms is of concern.

The use of these counters may not be appropriate when using a SAN. Use the hardware vendor's tools instead.

The Split IO/sec counter can indicate evidence of file fragmentation, and high levels should be addressed with file defragmentation. You can use the remaining counters to measure transfer rates and throughput.

Note: The physical disk counters represent all activity to a physical disk or array, which may contain several partitions (logical drives). The logical disk counters monitor activity to individual logical drives, so they can be used to identify which logical partition is utilizing the disks.

The Server counter Files Open can also be useful to indicate how many files are in use at any time. This can help to indicate how many concurrent exports are occurring.

Chapter 4

Enterprise Vault infrastructure

The introduction of Discovery Accelerator can have a significant impact on the existing Enterprise Vault architecture. The Enterprise Vault performance also influences Discovery Accelerator considerably.

If the Enterprise Vault environment is being sized along with the Discovery Accelerator implementation, the sizing can take into account the impact of Discovery Accelerator. However, it is likely that you are adding Discovery Accelerator to an existing Enterprise Vault environment, and therefore you may need to upgrade the existing servers or add further ones.

Enterprise Vault indexing service considerations

Discovery Accelerator search functionality can have a significant effect on the Enterprise Vault indexing services. Tuning these services can greatly improve Discovery Accelerator search efficiency and time to complete.

Enterprise Vault 14.2 and later, Elasticsearch is the new 64-bit indexing engine and Enterprise Vault continues to support any legacy 64-bit and 32-bit non-Elasticsearch indexes. In addition, Enterprise Vault also offers index groups to provide a distributed and scalable indexing architecture. Index groups can offer a method of scaling-out and distributing index volumes to achieve a greater search throughput.

Discovery Accelerator supports searching 64-bit Elasticsearch indexes, legacy 64-bit and 32-bit non-Elasticsearch indexes that are available in Enterprise Vault.

Server considerations

You may need to upgrade an existing environment to meet the higher specifications that the eDiscovery services demand. In addition, you may want to consider the use of index groups to provide dedicated index servers. See the *Enterprise Vault Indexing Best Practices Guide* at https://www.veritas.com/support/en_US/doc/EV_Indexing_BP_142.

The servers that are hosting the Enterprise Vault indexing service may need to handle sustained periods of CPU, memory, and I/O-intensive activity during searches with, by default, ten concurrent searches running until all indexes have been searched. This activity is likely to be repeated throughout the working day as the legal discovery team adds search requests.

This activity needs to run alongside any normal journaling, archiving, retrieval, or end-user loads. A multi-processor server is essential to ensure that all these concurrent activities can function with reasonable performance.

The Enterprise Vault indexing services can require a high memory capacity, particularly when complex searches may be submitted to multiple indexes concurrently.

The minimum number of processor cores and RAM required per Enterprise Vault indexing server is 16 cores and 32 GB.

Either physical CPUs or a combination of multi-core and multi-CPU can be used, but server sizing must not be based upon hyper-threading.

Note: If an existing Enterprise Vault indexing server is changed, the existing 64-bit index metadata in the Repository.xml file may need to be rebuilt to make best use of the changed hardware. This can be achieved by using the solution procedure described in the Veritas article http://www.veritas.com/docs/000088150.

32-bit and 64-bit index volumes

In upgraded environments, Discovery Accelerator may need to search a large number of 64-bit Elasticsearch index volumes and legacy 64-bit and 32-bit non-Elasticsearch index volumes.

To take advantage of the features and performance offered by the 64-bit Elasticsearch indexing engine introduced in Enterprise Vault 14.2 and later, it would be beneficial to upgrade all search target 32-bit and 64-bit non-Elasticsearch index volumes to 64-bit Elasticsearch index volumes. See the *Enterprise Vault Indexing Best Practices Guide*: https://www.veritas.com/support/en_US/doc/EV_Indexing_BP_142

Index detail level

The Enterprise Vault indexing service offers several levels of indexing detail.

Indexing levels for Enterprise Vault indexing service

Level	Notes
Brief	The index created by Enterprise Vault enables searches on the following attributes of each item: Author, Subject, Recipient, Created Date, Expiry Date, File Extension, Retention Category, and Original Location. If a search matches an attachment to an item, the search result contains both the main item and the attachment.
Medium (32-bit only)	As for Brief, and in addition enables searches on the content of each item, excluding phrase searches.
Full	As for Brief, and in addition enables searches on the content of each item, including phrase searches.

You can specify the default indexing detail level for archive indexes in various places. Once items have been stored in a particular archive and the index has been created, you cannot change the index detail level without rebuilding that index.

To take advantage of Discovery Accelerator's content phrase-based searching, ensure that all the indexes that will be searched use the Full level. We strongly recommend that you set the index detail level for all archives to Full.

64-bit index volume tuning

If an Enterprise Vault indexing service is hosted on a server that is less than the recommended specification, and it is not a dedicated indexing server, then the server could become overloaded.

The following Enterprise Vault Server extended setting can be modified to reduce contention on resources. This setting should be tuned depending upon server specification and workload through experimentation to ensure an indexing backlog does not start to form.

Setting	Default	Recommended
Maximum concurrent indexing capacity	30	5 – 30 depending on resource availability.

32-bit index volume tuning

Index granularity

Each 32-bit index is created using a schema that defines how the index should be structured. Part of the schema is the document granularity, which defines whether items with attachments are stored separately (attachment granularity) or merged together as a single item (item granularity). The schema was determined at creation time and cannot be changed for any legacy 32-bit index volumes.

Discovery Accelerator searches of 32-bit attachment granularity index volumes can be slower and potentially encounter resource issues.

It is therefore recommended to upgrade existing 32-bit index volumes to 64-bit. As this could potentially take considerable time, it would be worth targeting the most beneficial archives first, such as journal archives.

Maximum locations per index

The legacy 32-bit indexing service used internal structures known as “locations” to determine when to split an index into multiple index volumes, and this therefore determined the number and size of legacy 32-bit index volumes created for large archives. The number of locations used depended on the version of Enterprise Vault installed when the archive and its index volumes were first created.

The performance will vary depending on the size and volume of legacy 32-bit index volumes, and therefore it is recommended to upgrade all relevant index volumes to 64-bit Elasticsearch index.

Index file fragmentation

The index files quickly become fragmented on disk, even if there is a large volume of free storage capacity. This file fragmentation can cause severe performance problems which need to be managed on any index storage device.

Either an automated background file defragmentation product or scheduled device defragmentation must be employed. Any scheduled defragmentation should be performed with the indexing service stopped to prevent the potential for corruption.

Note: For certified automated background defragmentation tools, see the Enterprise Vault *Compatibility Charts* at http://www.veritas.com/docs/000097605.

System tuning

Disable Windows file indexing on the drives that contain Enterprise Vault indexes.

If you have installed anti-virus software to scan file and network accesses, disable it on the index servers. Anti-virus software should exclude the index locations and the Enterprise Vault server cache from its file scan due to potential issues with anti-virus software corrupting indexes.

The opportunistic file locking mechanism is known to cause problems when storing index files on NAS storage devices. Therefore, the current Support advice is to disable opportunistic locking at the NAS head and in Windows.

Co-existing eDiscovery solutions

If multiple eDiscovery solutions need to co-exist, the Discovery Accelerator may need to be tuned to ensure the Enterprise Vault indexing servers are not overloaded.

It is essential that the Discovery Accelerator services balance the load by reducing the search and export threads as described previously. The number of ECM retrieval threads may need to be reduced in each analytics service.

Co-existing eDiscovery solutions should also be tuned as described in the associated documentation. This ensures that all products use Enterprise Vault services in a balanced way.

Enterprise Vault storage service considerations

The Discovery Accelerator pre-fetch cache, export feature, online reviewing, and analytics data collection can create an unexpected load at the Enterprise Vault storage services.

Server considerations

You may need to upgrade an existing environment to meet the high demands of the Discovery Accelerator services.

The server that is hosting the Enterprise Vault storage service may need to handle sustained periods of CPU, memory, and I/O-intensive activity during exports and analytics original item retrieval.

By default, each Discovery Accelerator export starts 25 concurrent retrieval threads, up to a maximum of 100 threads, which run until all the items have been downloaded. This activity is likely to be repeated throughout the working day as the legal discovery team initiate export requests.

Each Discovery case that is enabled for analytics needs to download all the associated messages. By default, it starts two concurrent retrieval threads per Enterprise Vault server, although this might be increased to improve collection. These threads run until all the items have been downloaded—by default, up to 10 threads across all Enterprise Vault servers per concurrently enabled case.

The Discovery Accelerator case review features can also generate on-demand downloads from the Enterprise Vault storage service. The Discovery Accelerator pre-fetch cache discussed previously can help minimize the potential impact of online reviewers by downloading the most recently searched items during a predefined time window.

All this activity needs to run alongside any normal Enterprise Vault journaling, archiving or end-user tasks. A multi-processor server is essential to ensure that all these concurrent activities can function with reasonable performance.

The minimum number of processor cores and RAM needed per Enterprise Vault storage server is 8 cores and 16 GB. If Enterprise Vault indexing services are co-located with storage, then use the recommendations as 16 cores and 32 GB.

Either physical CPUs or a combination of multi-core and multi-CPUs can be used, but server sizing must not be based upon hyper-threading.

Note: If an existing Enterprise Vault indexing server is changed, the existing 64-bit index metadata in the Repository.xml file may need to be rebuilt to make best use of the changed hardware. This can be achieved by using the solution procedure described in Veritas article http://www.veritas.com/docs/000088150.

Storage considerations

Under normal circumstances, the archived data that the Enterprise Vault storage services manage is likely to be stored on slower, archive-level storage or off-line storage devices. This is likely to continue to be acceptable and preferable for the majority of cases.

However, larger customers who demand high-speed responses may want to store archived data on higher specification devices to ensure that online reviewing, analytics, and export activities are not delayed. Many devices that

are targeted at archive data or for compliance are already capable of providing a high throughput of retrievals, and so should remain suitable for medium and large customers.

When Discovery Accelerator online reviewing, analytics data collection, and exports are running, the Enterprise Vault storage services are queried to obtain the original archived data. This creates a higher I/O load than normal, which may cause contention with normal journal, archiving, and end-user tasks. If multiple Enterprise Vault servers share a single storage device for all vault stores, the load generated (particularly by analytics data collection) can result in a significant bottleneck at the storage device.

Ensuring that the Discovery Accelerator pre-fetch cache is operational at an appropriate time window should reduce activity at undesirable times. The Discovery Accelerator pre-fetch cache downloads original data at out-of-hours times.

Scaling Enterprise Vault for Discovery Accelerator

The Enterprise Vault architecture is likely to need scaling up to meet the demands of Discovery Accelerator and the archiving load expected. However, scaling up individual Enterprise Vault servers does have limited benefit. Therefore, to meet service-level expectations, the Enterprise Vault infrastructure may need scaling out by adding additional Enterprise Vault indexing and storage servers.

Scaling out

Each Enterprise Vault service can be hosted on different physical servers, but distributing every service on a separate server does not necessarily provide any benefit.

Enterprise Vault index groups enable a set of dedicated and scalable servers to manage a set of index volumes. See the *Enterprise Vault Indexing Best Practices Guide*:
https://www.veritas.com/support/en_US/doc/EV_Indexing_BP_142

The addition of more storage services and vault stores should not affect the SIS ratio because of the enhanced sharing-group single instance storage.

When you use an EMC Centera device, the SIS is handled differently than NTFS partitions or other network-based storage, and it is shared across vault stores located on the same Centera. Therefore, the addition of servers and vault stores to the same Centera does not affect the SIS.

Rolling over journal vaults

The journal archives are likely to grow quickly, resulting in very large archives with large indexes. To keep journal archives and their indexes within manageable sizes, you can roll over the journal archives on a regular basis (for example, monthly).

Rolling over journal indexes also provides more control as to where the data and indexes are located. To prevent the Enterprise Vault servers from becoming oversubscribed with search requests, you can add further Enterprise Vault servers and add all new journal archives to them. This effectively makes the previous Enterprise Vault servers read-only servers, improving the search performance on them.

Monitoring Enterprise Vault

The Enterprise Vault documentation recommends methods for monitoring the Enterprise Vault environment. You should also monitor the indexing and storage services to measure the impact of Discovery Accelerator as follows:

- The impact of searches to allow the Discovery search threads to be correctly tuned.
- The impact of the pre-fetch cache or production upon the storage service.
- The impact of Discovery online reviewing, if the pre-fetch cache is not used.

Use Windows Performance Monitor to obtain system statistics. You should typically monitor the following counters:

Object	Counters	Instances
LogicalDisk	Avg. Disk Queue Length Disk Transfers/sec Avg. Disk Bytes/Transfer Avg. Disk sec/Transfer Disk Bytes/sec Split IO/sec	Index location drives
Server	Files Open	
Memory	Page Faults/sec Pages/sec	

Object	Counters	Instances
	Available Bytes Free System Page Table Entries	
Network Interface	Output Queue Length Packets/sec Bytes Total/sec	All instances
Processor	% Processor Time	_Total
Process	% Processor Time	All Indexserver.exe processes
System	Processor Queue Length Context Switches/sec File Data Operations/sec Processes	

CPU and memory

The % Processor Time for the _Total counter indicates overall system activity. If the CPU is consistently 80%, the server may start to indicate performance issues.

However, consistently high CPU is not the only indicator of performance problems. If the index server processes tend to use very little CPU for long durations, this could indicate an I/O bottleneck. Monitoring the index server processes for low CPU is useful when the index file locations are not located on the index server itself, but on a remote storage server.

It would be worth monitoring the memory counter Pages/sec, which records the number of hard faults to disk. If there are significant or sustained hard faults, the problem should be traced to the source.

Disk

The disk counters are only relevant if the index file locations are locally attached. If the index file locations are on a remote server, monitor that server for disk activity.

Typically, the disk read/write queue length counters are monitored for evidence of a disk bottleneck. The queues should not normally rise above twice the number of disks in the array.

Monitor the average disk sec/transfer to measure the latency of the device accesses. Ideally, this should be approximately 5ms. However, anything averaging in excess of 20ms is of concern.

The use of these counters may not be appropriate when using a SAN. Use the hardware vendor's tools instead.

The Split IO/sec counter can indicate evidence of file fragmentation, and high levels should be addressed with file defragmentation. The remaining counters can be used to measure transfer rates and throughput.

Note: The physical disk counters represent all activity to a physical disk or array, which may contain several partitions (logical drives). The logical disk counters monitor activity to individual logical drives, so you can use them to identify which logical partition is using the disks.

The Server counter Files Open can also be useful to indicate how many files are in use at any time, perhaps suggesting that the server is oversubscribed with search requests.

Chapter 5

End-user advice

The methods that end-users employ can influence the overall Discovery Accelerator system performance. We recommend that you distribute the following advice to users of Discovery Accelerator.

Legal holds

Legal holds can affect the performance of search acceptance. Therefore, it is best to create a case initially without legal holds and then, once the bulk of searches have been accepted, enable legal holds on the case.

Searching

Searching can potentially be time-consuming, but you can improve performance by ensuring that your searches are appropriately created and focused. A poorly constructed search can inadvertently cause thousands of indexes to be searched for large volumes of data, perhaps unnecessarily. Following these simple practices helps to improve overall search performance.

- Ensure that the case is searching only the archives that are relevant to the case in the case properties.
- Do not start a search without criteria. This retrieves every item from all indexes.
- Do not use wildcards unless necessary. This can severely impact performance.
  Due to limitations on the wild card searches from the Elasticsearch indexing engine introduced in Enterprise Vault 14.2 and later, it is recommended to provide a minimum of three leading characters along with the wildcard when dealing with searches involving phrases.
  For example, search criteria “Enterprise Vault indexing engine migrated

to ela*" is better than specifying "Enterprise Vault indexing migrated to e*"

- Make searches specific, and try to include author or recipient details.
- Specify date ranges. This can reduce the number of indexes searched.
- Avoid overusing search terms. Thousands of terms can cause iterative searches.
- Ensure that scheduled searches do not run at the same time as system backups.
- Quickly accept or reject searches to avoid filling and slowing the database.
- Test new searches in research folders, and then delete the folders as necessary.

Hotword Analysis

Compliance Accelerator 12.5 introduced the Hotword Analysis feature within Search result Accept functionality. This feature is integrated into the Accept task and will be triggered as part of same.

It enhances the review procedure in Compliance Accelerator by enabling the reviewer with the Hotword hits information as well as enables them to navigate between them.

- Hotword Analysis doesn't impact the performance of Sampling or Search throughput.
- The performance for analysis is degraded if there is an increase in the number of hotwords in a Search, hence it is suggested that you use an appropriate list of hotwords only.
- Increase in the number of wild card characters in words or phrases may also result in slow analysis.
  Due to limitations on the wild card searches from the Elasticsearch indexing engine introduced in Enterprise Vault 14.2 and later, it is recommended to provide a minimum of three leading characters along with the wildcard when dealing with searches involving phrases.
  For example, search criteria "Enterprise Vault indexing engine migrated to ela*" is better than specifying "Enterprise Vault indexing migrated to e*".

Analytics

When you enable a case for analytics, the original items are downloaded from Enterprise Vault into Discovery Accelerator. This can place a high load on the Discovery services, but you can improve performance by following some simple practices:

- Try to avoid enabling a case for analytics during office hours. If possible, enable the case at the end of the day to retrieve message content overnight.
- Avoid enabling several cases for analytics simultaneously. Enable the cases one at a time and wait for the previous case to be marked as completed before you enable the next.
- Define rules for automatic categorization before you enable a case for analytics.

Reviewing

The online reviewing facility provides an integrated, feature-rich environment with which you can review large volumes of documents before you produce them. An appreciation of the processes triggered during reviewing can prevent unnecessary load on the service.

The message preview pane may initially take several seconds to display. This is expected. During this time, the service fetches the archived item from Enterprise Vault, which may be located on slower archive equipment. Do not attempt to speed up the display by clicking several different items.

Export and production

When you export or produce search results, Discovery Accelerator starts to download the original data from Enterprise Vault to the desired export location. If you select a poor destination, such as a location over the network, the export can take significantly longer.

The Discovery Accelerator administrator should provide several different high-speed locations on the Discovery Accelerator server in which to place exports.

- Export to a high-speed drive located on the Discovery Accelerator server.
- Try to avoid running simultaneous exports. If this happens, try to ensure that they are output to different drives.

- Exporting Exchange email to native format enables easier tracking and authentication through the production reports.
- If an export fails for any reason, use the export retry feature rather than re-export.
- Only export to HTML if necessary.

Note: You do not need to break a large export into several smaller exports, as required in previous versions. However, splitting a large export into several smaller exports may help you to administer larger volumes of data.