Redpaper

Wei-Dong Zhu
Markus Lorch

IBM Content Collector Deployment and Performance Tuning

Overview

This IBM® Redpaper™ publication explains the key areas that need to be considered when planning for IBM Content Collector deployment and performance tuning.

This paper is divided into two major parts. In the first part, we present the questions that need to be asked and answered for a proper system design and the archiving policies, and provide other useful supporting information for the implementation. We discuss workload that is created at the various stages and the archiving process on the front end, on Content Collector, and on the back-end system components to help you understand the effects of design choices.

In the second part of this paper, we deal with operational topics, such as monitoring and tuning the archival system for optimal performance in conjunction with Content Collector. We provide commands and explanations so you can analyze logging output and use this information to tune archiving policies and system settings. We explain techniques to measure system throughput and provide simple mechanisms to diagnose common bottlenecks.

Specifically, we cover the following topics in the paper:

- Planning compliance and archiving scenarios
- Monitoring and health checks
- Performance and scalability
- Planning for high availability and load balancing
- Planning for backup and disaster recovery

Planning compliance and archiving scenarios

The primary aim of most Content Collector deployments is to archive content from email servers, shared file systems, and groupware solutions, such as Microsoft® SharePoint. Content Collector is not designed to replace the source system (for example, the email server), but to allow the system operator to define policies on how less frequently used data that can be archived and removed from these systems.

It is crucial that the user's requirements are properly elicited and understood. The expectations about the services the Content Collector solution can deliver also must be set realistically. The single most important topic to consider here is that content that is still in active use in the source system *should not* be removed from this system. Removing content that is in active use would cripple the original system, hurt productivity and reduce acceptance of the archival system by the users. A second topic is understanding that archiving content produces additional load on the source system, as the data has to be read from, marked, and potentially replaced with a place holder on the source system.

Archiving solutions are often technical projects with a focus on reducing storage cost and improving system performance of the system originally containing the data. With all the focus on technical aspects, it is easy to forget the implications of certain design choices on the users that need to work with the data that is to be archived. Users often do not like having their content archived, as it can add a layer of indirection and access delay when accessing this data after archiving. It is therefore important that the system, besides its technical aims, also caters to the user's requirements. A prominent example where the user must not be forgotten are mailbox management scenarios. It is important to design the mailbox management system and archiving policies in a way that does not hamper the user's productivity by archiving content too early. Also, users who often work disconnected from the enterprise archive should not be forgotten, so offline repository support should be included in the planning. Properly designed archiving policies greatly reduce storage space while retaining direct access to frequently used data.

Keep it simple

This general design principle is one that we find important for archiving systems. The policies created, especially if users see the effects of archiving, should be easy to understand and support the acceptance of the system's behavior. Less complex archiving selection and life cycle rules also often reduce the load impact of archiving. Fewer archiving policies make the system easier to manage.

Our advice is to start with a basic setup. As you gain experience, add the complexity as needed. Select the subset of functionality required by the specific customer environment. Implement simple to understand and simple to execute archiving policies. Keep the number of life cycle management stages low.

Gathering requirements

It is crucial to understand the requirements that the archiving system must address and the environment in which it needs to operate before beginning to plan a deployment. To help with gathering these requirements and to better understand the implications for the archiving system architecture and operation, we put together the following set of questions and explanations. We cannot cover every possible scenario and combination of requirements, but the subset of topics covered here addresses the most important requirements that influence the system design:

- What are the primary goals of the archiving solution?
- What are the characteristics and volume of data to be archived?
- What is the amount of time available for archiving?

What are the primary goals of the archiving solution

First, we need to understand the primary need for archiving. The often encountered key requirements are:

- Archiving to fulfill legal requirements and provide full-text search capabilities on the archived contents to designated personnel (typically referred to as compliance archiving and eDiscovery scenarios)
- Archiving to reduce storage requirements of file, email, or groupware servers and provide full-text search capabilities on the archived contents for users (for email archiving scenarios; these cases are also often referred to as mailbox management scenarios)
- Archiving content to make it available for the use in business process and records management scenarios

Compliance archiving scenarios typically do not interfere with the way users work with their content and have a very limited number of system users, which makes

them simpler to plan and deploy. For other content archiving scenarios, the user use-cases and requirements need to be taken into consideration. In “Mailbox management planning considerations” on page 5, we describe things to look out for when planning an email mailbox management scenario. Similar considerations apply to non-email content archiving scenarios.

What are the characteristics and volume of data to be archived

It is of key importance to know the volume and characteristics of the data to be archived. Ideally, you should understand:

- Daily archiving volume: The number of items (emails or files) that need to be archived on a daily basis (for example, number of items from email journals and number of items from user mailboxes to be archived during a day)
- Initial archiving volume: The number of items (emails or files) that need to be archived at the beginning of the project as part of the existing data backlog
- Percentage of duplicates: The expected percentage of duplicate items (duplicate emails, duplicate files, and duplicate email attachments).

  For example, if an email has an average of five recipients, we expect to see six copies of this message archived in mailbox management scenarios (one sent copy and five received copies). If the email journal is also archived, then a seventh copy will be processed. Knowing this number of expected duplicates (number of total copies - 1) helps greatly with more accurate storage requirement predictions.
- The average size of items to be archived, especially the expected size of files or email attachments, as these typically make up the majority of the required storage

One way of gathering the above details is to perform a proof of concept archiving run using a representative sample of real world data. Content Collector audit logs can be used to record information about email attributes, such as recipients, email, file size, and email attachment count. In email archiving scenarios, most of the information can be gathered using a virtual machine running Content Collector and a task route that only collects email and extracts metadata and attachments but does not archive the data. Besides the information about characteristics, such a small scale archiving test (if performed on physical hardware) can also be used to determine the possible throughput from the existing email server or file server system. In “Content Collector logs” on page 20, we explain how to work with the Content Collector audit logs.

What is the amount of time available for archiving

The amount of time available for archiving is an important factor in system design and architecture. You need to discover:

- The hours per working day available for archiving
- The time frame planned for initial back log archiving
- Time for other uses of the email system and the enterprise content management archive system (for backup, reorganization, and other activities)

If high volume archiving (backlog archiving or mailbox management) is to be performed, then this workload may need to be scheduled during non-working hours. Experience shows that company's email systems often have little or no time periods of low utilization on a daily basis, as backup and database reorganization times consume much of the time during non-working hours. For file servers, there is typically more time for archiving available.

Tips: A good practice is to put a time table together that lists all systems involved and the utilization of these systems over time to plan for an appropriate archiving window.

Consider using weekends to do these activities. Do not forget to plan for periodic maintenance slots.

Mailbox management planning considerations

The business driver behind mailbox management scenarios is often storage cost reduction. A second driver frequently encountered is the consolidation of user managed local message archives (local PST or NSF files) and improved search mechanisms for email content. This section provides a collection of suggestions and experiences to consider when planning a mailbox management scenario.

Archiving all emails can be counterproductive if storage cost reduction (by moving data from the high-performance email server storage to less expensive archive storage) is one of the main objectives of the archiving implementation. The impact of storing a message in the archive can increase the total storage requirement. This is especially true for very small messages. The archive database storage requirement is, to some degree, independent of the message size. and the email server storage may not be reduced significantly if message stubs are kept for small messages.

However, if not all messages are archived, then the archive search cannot provide complete results. Often, the benefit of having all messages searchable in

the archive outweighs the additional storage impact incurred by also archiving small messages.

All mailboxes using single instance attachment storage mechanisms should be considered for archiving to achieve an email server storage reduction. If not all messages holding a reference to a specific attachment in the single instance storage are archived, then the attachment will remain on the email server storage.

Content Collector always archives the complete email (there is no option to only archive attachments), and the different email stubbing options pertain only to what Content Collector does with the original copy of the email. For example, Content Collector may be configured to remove the attachments from the source email after it has created an archive copy in the enterprise archive of the full message. In the archive, the email and the attachments are stored separately. This action enables Content Collector, in conjunction with an enterprise content management archive, to not only store duplicated messages *only once,* but also to store only one copy of each unique attachment, no matter in how many emails this same attachment is used.

Companies often ask for a quick reduction of mailbox storage and therefore for the deletion of archived content from the mailboxes. Deleting emails from the user's mailbox must be *planned carefully*, as you do not want to cripple the email system and reduce user efficiency. Users typically must be able to continue to work with their email in the email system for the duration of frequent access. The email archive is not intended to replace the email system.

For most mailbox management scenarios, it is beneficial to keep the message bodies of archived messages intact for the duration of frequent access. The time frame of frequent access varies by businesses. We believe a duration of three to six months is typical. Nevertheless, early archiving of emails can take place. It is merely the stubbing process that operates with an intentional delay.

Immediate space saving can be accomplished by replacing email attachments with references to the archived version. This stubbing option retains the full fidelity of emails and can be applied directly during initial archiving (this option does not require a second modification to the emails, but can be applied directly in the archiving task route). Users have the full email body in the original formatting in their mailbox and can reply to, forward, and otherwise work with their email without needing to restore the original. The original attachments are accessible through a link. The link is accessible to anybody that is part of the user’s organization. The archived attachments remain accessible to users who receive forwarded copies or replies as long as the recipient has network access to the enterprise archive. Email stubs that are to be forwarded to external recipients must be restored to include the attachment data, as the stub URLs are typically not accessible from outside the company’s network.

In real world deployments, only removing attachments from archived emails can reduce email storage on the email server by up to 80% with virtually no impact on the way users work with their emails.

If the offline repository function is needed, then emails must not be immediately stubbed during archiving to give the offline repository function time to make a local copy of archived messages before content is removed from these messages.

Leaving email bodies intact allows users to work with their local search functionality of the email client when searching for text in email bodies. This significantly reduces the search load on the email archive. The system requirements for the email archive search functionality can be reduced if email bodies are kept in the user's mailboxes for the duration of frequent and semi-frequent access.

Figure 1 shows a sample of 3400 emails and a average size share versus storage size share histogram from a real world deployment.

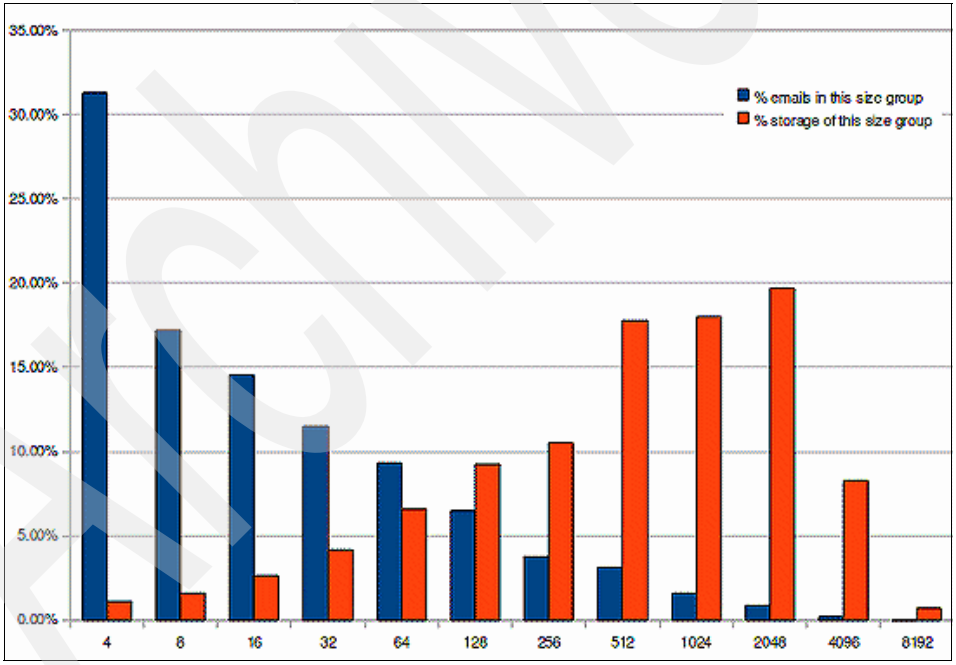

Figure 1 Histogram of emails by their body size and size groups (from 4 KB to 8 MB)

The x-axis divides the sample data set into different size bins (4 KB to 8 MB) and the x-axis shows the percentage of emails that belong to each group and the percentage of email server storage that is consumed by this group. We can see

that about 80% of all emails are smaller than 64 KB, but account for only about 15% of all storage. This means that most of the storage is consumed by the remaining 20% of emails.

The example analysis is based on email size data from the standard Content Collector audit logs of a few days worth of archiving. It does not show the number of attachments. Based on other experiences, we can assume that virtually all of the large emails contain attachments and that most storage is used up by these attachments (for example, it is not typical that email bodies without attachments exceed 64 KB in most industries). We can draw the conclusion that by removing the attachments from the email bodies, the majority of storage required by the existing email system can be freed. Note that on email server systems that use duplicate detection and single instance storage, such as IBM Lotus® Domino® Attachment and Object Storage, the actual storage consumption of these large emails can be significantly less.

When emails are no longer needed for active use, you can abbreviate the email stubs or completely remove them. This process of moving emails through different archiving stages (archived, archived and attachments removed, archived and body shortened, and archived and deleted) is referred to as *life cycle stubbing*. When using email life cycle stubbing, it is important to keep in mind that every stage of the life cycle adds additional processing impact. The Content Collector server must scan all applicable mailboxes on every scheduled life cycle collector run for messages that need to transition into a new life cycle state and modify the corresponding emails in the user's mailbox. This action creates a significant load on the email server and should be planned carefully. The premiere reason for deleting email stubs is to keep the item count in user's inbox folder below the vendor recommended maximum values (in particular, review Microsoft Exchange recommendations, which vary depending on the version of Microsoft Exchange used).

For many usage scenarios, we recommend retaining the complete email bodies. Abbreviating email bodies may seem like a natural next step in the email life cycle after archiving and removing attachments. However, the advantage of abbreviating or even removing email bodies must be carefully weighed against the disadvantages. For example, abbreviating the body reduces email storage requirements further, but only to a relative small amount (a few KB per average email, and virtually nothing for very small messages, which in some scenarios are the most frequent type of messages encountered). On the other side, abbreviating the body hurts email fidelity to a large degree, as formatting information is lost and the local email client search can no longer work reliably for abbreviated emails.

From a performance perspective, the least amount of impact on the email system and the archiving system is incurred in a setup that archives emails only after the

initial active use period has passed. The user then has time to delete unnecessary needed emails that might reduce the volume of messages to be processed. Furthermore in this scenario, the first archiving step includes a task to immediately stub the email by removing attachments from the email system, leaving the remainder of the email bodies intact. The remaining (relatively small) text bodies are kept for the typical lifetime of the email in the user's mailboxes. A second and final step (called *stub delete*) deletes the email completely from the user's email boxes when it becomes unlikely that this email will be used again (for example, after six months). This scheme creates a minimal impact on the email servers, as it requires that the email be retrieved from and modified on the email server once during archiving. The second and final operation on the email server is the stub delete at the end of the expected use period. The user has to use the archive search to locate and work with email for which stubs have been deleted.

Recommendation

In summary, we recommend the following items for mailbox archiving:

- Archive messages with a minimum delay of 30 days after they have been received to allow for deletion of unnecessary emails and avoid archive access for frequently used data.
- Remove attachments and replace them with a hyperlink during the initial archiving step to reduce email server disk usage while keeping archiving load on the email server to a minimum.
- Keep message bodies intact for the full time that the emails are to be kept in the email box.
- Delete message bodies after they are typically not needed anymore (usually 3 to 12 months after receipt).
- Consider if offline repository functions are needed and use alternative archiving task routes for offline repository users to split archiving and stubbing.

Content Collector server workload components

In order to plan for and later tune Content Collector deployments, you have to understand the distinct workloads that make up the total workload of an email archiving deployment.

Workload on the Content Collector servers can be categorized as follows:

- Archiving workload

  This workload covers scanning for and reading email from the source email servers, processing these emails on the Content Collector servers, saving the

emails in the enterprise archive, completing this transaction by marking the email as archived in the email system, and potentially deleting attachments or the complete message from the source system.

This workload is usually the workload that contributes the most to Content Collector and enterprise archive system requirements in terms of CPU and storage requirements, but also creates significant load on the email and the archive and text indexing system. The most important sizing metric for this workload is the number of emails that have to be processed by Content Collector in a defined time frame followed by the number of mailboxes that need to be visited.

- Mailbox scanning workload

  This workload is created whenever Content Collector has to check for archival candidates in a specific mailbox. Proper design of the archiving schedules should keep these scans to a minimum. If mailboxes have many items in them, if scans for messages to be archived are frequent (for example, if Content Collector is set to an “Always” schedule), and if Content Collector have complex filter constraints, then this part of the workload can create a significant disk subsystem load on the email server.

- Stubbing life cycle workload

  This workload encompasses scanning the email system for email that has been previously archived and modifying the already archived emails according to the defined stubbing life cycle. For example, 30 days after an email has been archived, the attachments are deleted from the email, leaving only references to the archived version of the attachments in the enterprise archive. This workload can require that the remaining email stub be retrieved again from the email server, modified by Content Collector, and saved back to the email server.

  The email server and the Content Collector server are affected by this workload. The enterprise archive is not involved in these operations. The number of mailboxes that have to be processed and the frequency at which emails have to be moved from one life cycle state into the next are the key sizing matrix.

- Interactive workload (search, view, and restore)

  In mailbox management scenarios, the users issues requests to search for, preview, and restore archived content to the Content Collector servers. This workload requires the Content Collector server to retrieve content from the enterprise content management backend, parse and process it (for example, for preview operations), and forward the content on to the user’s computers. To ensure that sufficient resources are available to fulfill these requests with low latency, other non-critical archiving tasks (such as automatic archiving of

old messages from user's mailboxes) should be reduced during peak working hours.

Planning your workload and archiving schedules

There are two important facts that help you understand and plan archiving schedules for email archiving scenarios:

- Selection of emails to be archived versus actual archiving operation
- Avoid setting Content Collector schedule to Always

Selection of emails to be archived versus actual archiving operation

The Email Server Connector performs the selection of emails to be archived and the actual archiving of the selected messages in disjunct operations. The system uses the following three step approach to reduce email server load when a collector run is scheduled:

1. The Email Server Connector resolves the configured group members and identifies mailboxes that have to be archived.
2. It then starts processing the mailboxes and issues a query for archival candidates on one mailbox at a time. The result of this query is a list of items to be processed (a to-do list).
3. This to-do list of emails (to be archived) is then processed asynchronously and in parallel.

Unnecessary Content Collector runs are avoided. If the internal work queue still has pending emails to be archived or stubbed from a previous Content Collector run, then all mailboxes with a backlog of messages pending to be archived will not be scanned for messages to be archived.

In general, the intervals for Content Collector runs should be scheduled with enough time in between them to ensure that the system is able to archive all already identified emails before the next run is scheduled. To derive an adequate schedule time, the system should be run with a manual Content Collector schedule (for example, run once) to archive the initial back log of emails. After the system reaches a steady state (the back log has been processed), you can time how long a typical archiving run takes to complete and, based on this information, derive an adequate schedule.

To ensure that all mailboxes are, over time, processed with the same probability, the Email Server Connector changes the order by which mailboxes identified in step 1 are processed for each Content Collector run.

Avoid setting Content Collector schedule to Always

Avoid running any production task routes with a Content Collector schedule setting of Always. This setting can lead to frequent scans on the source email server for emails to be archived on all configured mailboxes. The only exception where we think this setting may be considered is for the interactive archiving task routes. In that use case, the trigger mailbox is frequently polled by the Email Server Connector to determine from which users' mailboxes archiving requests exist. These frequent checks on the trigger mailbox do not have a significant impact due to the small size of this mailbox. If it is acceptable that the interactive archiving requests can wait for several minutes before being processed, then an interval setting of 15 minutes for the Content Collector that serves the trigger mailbox should be considered.

For compliance (journal) archiving scenarios, a best practice is to keep the journal mailbox size small. Consult your email server documentation to discover how many emails can be stored in a journal before you experience performance degradation problems. We recommend setting the Content Collector interval (the time it takes to populate the journal) to 25% of the recommended maximum size or less. This action ensures that archiving starts long before the journal mailbox develops performance problem while also avoiding unnecessary scans on almost empty journal mailboxes. It might be helpful to understand that Content Collector does not guarantee that all messages currently eligible for archiving are selected in a single Content Collector run. A collection interval that is too large might lead to an archiving backlog and increase in journal size.

Email systems benefit when you keep the number of items located in the same folder of a mailbox low. A frequent archiving approach is usually a good starting point.

High volume journaling setups typically use multiple journal mailboxes and email databases to hold the journals. These setups have multiple advantages over a single large journal mailbox:

- They allow spreading the load of journaling and archiving from journals to multiple disk subsystems.
- They prevent a single mailbox from growing too large and becoming difficult to manage and access.
- They allow for increased parallelism in journaling and archiving.
- They allow for different levels of service for specific user groups.

Be aware of potentially long query times when Content Collector is querying the email system for messages to be archived from mailboxes holding a huge number of messages. The number of items in the queried mailbox, the state of how well it is organized internally, and how much of it is potentially cached (has

been recently used), and the speed of the disk subsystem holding this mailbox greatly influence the query time.

Delayed archiving is a symptom of a long running query for archival candidates. You can discover if delayed archiving is due to a long running query by temporarily turning on informational logging on the Email Server Connector, which will show the delay between when the query for archival candidates is sent and when the first message is archived (see “Understanding Email Server Connector operation” on page 21 in “Content Collector logs” on page 20). If the delay is more than a couple of minutes, then you likely have a bottleneck that needs to be addressed.

The email server disk utilization of the disks that contain the queried mailbox database will peak during the time the email system executes the query. If the disk subsystem is highly utilized for a long period of time and this storage is shared with other mailboxes, then all other operations, such as regular user email operations on the same disk storage, might be slowed significantly during the time of the query.

If the query time cannot be improved, then consider moving large mailboxes to a dedicated storage unit to avoid impacting other operations, and strive to reduce the schedule frequency (querying for emails to archive) to a minimum.

For IBM Lotus Domino based systems, we recommend using the journal roll-over functionality and archive all but the current journal. This action prevents concurrent access to the currently active journal database from the Lotus Domino and Content Collector servers and yields better performance. Typically, the number of emails that can be stored in a Lotus Domino mailbox or journal can be very large before a performance impact becomes significant. However, very large mailboxes (for example, those close to the maximum capacity) tend to slow the archiving process. Using the roll-over mechanism helps keep the size of journal mailboxes at a lower file size (we recommend a few gigabytes) while at the same time offers plenty of buffering, which allows a low archiving interval (for example, collecting emails every hour).

In mailbox management and other archiving scenarios, you must define the frequency that user mailboxes or other source system containers, such as file system folders, are checked for items that need to be archived. Scanning for archival candidates creates load on the source system (the email server or the file server) and the frequency of scanning should be balanced with the number of expected archival candidates. For example, in an environment where the average user receives 20 emails per day, an interval that defines a weekly collection of messages from user should suffice, as it will likely turn up less than 100 messages per collection. If more frequent Content Collector runs are configured, then the time it takes to complete a scan for archival candidates can be a significant part the processing time, which creates unnecessary email

server load and hurts the total throughput. Complex filter expressions may also have an impact on the time it takes to perform the scan for archival candidates.

For deployments with large user bases, consider how you should group the mailboxes to work with archival and maintenance schedules and with the necessary archiving frequency. For example, if we consider a deployment for 10,000 users, where each user should be visited once a week, we can come up with five groups of roughly 2,000 users, where each group will be visited once a week. Furthermore, there might be special groups for accounts that have to be archived more or less frequently. If the systems are resource constrained, for example, due to maintenance or reorganization intervals, a further separation based on email server clusters might allow high throughput archiving from one cluster while another cluster is skipped so that you can do maintenance, backup, and reorganization.

Email stubbing workloads can typically be scheduled on weekends (run only once per week). In most cases, it does not matter if the scheduled state transition for archived messages is delayed by a few days until the next run of this workload due to a once a week interval setting.

Interactive archiving is typically done during peak working times. It might be beneficial to keep the Content Collector system at a low load during these times to ensure that interactive archiving requests are fulfilled quickly and to lessen the archiving load on both the email server and the enterprise content management backend during peak working times. This configuration might also ensure quick response times on the interactive retrieve (search, view, and restore) requests.

If email or file back logs have to be archived in the initial phase of a new deployment, then the time available for this archiving phase must be defined. The more time available, the less additional resources are needed, and the load generated by back log archiving can be given priority when the interactive and daily archiving workload is low. For example, this configuration can initially be scheduled to run on the weekends. The additional processing capacity needed in the first year for back log archiving is often consumed in subsequent years of production by an increase in archiving volume.

Email, file, or groupware server and network load must also be considered when planning the archiving schedules to avoid overlapping workloads that cause a high system load (an example of two workloads that suffer great performance reductions if run in parallel is email server database reorganization paired with email archiving).

Content Collector archiving can create significant work for the email servers because mailboxes and emails are archived at a much higher rate than typical user interactions access and modify this data. Typical bottleneck areas are the disk subsystem hosting the email server databases, and the total number of

requests. This scenario encompasses regular user requests and archiving requests from the Content Collector servers, if they are scheduled to perform archiving during work hours. The network bandwidth and network latency can become an limiting factor if WAN connections are in place between the email server and the Content Collector servers. The RPC style protocols that are used to access the email servers perform best in low latency and direct connections. Content Collector cannot benefit from some of the performance improvements, such as local message caches, due to the nature of archiving. Locate the Content Collector server next to the email server if you have a distributed environment and use direct gigabit Ethernet connections to keep latency low and avoid throughput limitations.

General recommendations for system planning

In this section, we provide a number of recommendations that we believe are best practices for archiving setups. Consider using these recommendation before deployment takes place, as making changes later is sometimes difficult or impossible.

Host name alias

We recommend using a host name alias for the archiving solution under which the Content Collector Web Service is available to *all* users of the enterprise. This host name alias is the name that is used in all stub links (the URLs put in place of archived content). The host name must be carefully chosen, because changing the stubs to point to a different host name is expensive and sometimes even impossible. Using an alias provides a large degree of flexibility in assigning and later possibly changing the server where the archiving and restore services actually run. In addition, it is the host name used in issued HTTPS certificates.

If direct links to the back-end enterprise content management archive are used (for example, in older file archiving scenarios), then the same recommendation for a host name alias also applies to the back-end archive.

Finally, if multiple classes of services are offered by the Content Collector Web Service (such as retrieve for archived files and retrieve for archived email), then consider using multiple host name aliases. This configuration preserves the option to route to specific Content Collector instances based on the class of requests, which may be helpful to implement differentiated service levels.

Storage planning

A Content Collector archiving infrastructure requires a number of storage units that are allocated to the different servers. Make sure none of the performance critical storage locations reside on the standard (operating system disk) location. Place performance critical storage on specific storage devices based on the

workload you expect for these storage areas and the performance characteristics of the storage devices. It is a best practice to provide explicit and easy to understand names for mount points and drive labels to avoid misunderstandings. Clearly denote what the use of a specific drive or mount point is. Using the same type of storage (for example, expensive Fibre Channel (FC) drives with RAID 10 configuration) for all uses might not be cost effective.

We provide high-level guidelines to plan the following type of storage:

- Archival storage for the binary and text documents
- Temporary storage
- Text index storage
- Database storage
- Log storage

Archival storage for the binary and text documents

This storage might be, in some cases, a fixed storage (WORM) device, but in other cases might be regular disk storage. Initially, the system only writes to this storage during archiving by using infrequent reads (retrieves). In later stages, migration of older content to lower tier storage or deletion of content (expiration management) is added to the workload. Storage deduplication is handled by a combination of mechanisms that the Content Collector system provides together with support form the enterprise content management backends. If fixed storage devices are used, then these devices usually provide data deduplication.

We recommend using RAID 5 arrays that use inexpensive, large drives (for example, SATA drives). Depending on your environment and setup, a SAN or NAS solution can be used. Farmed content engine setups (FileNet® P8) require the use of a distributed file system. Here, an NAS solution is typically the preferred mechanism.

Temporary storage

Various components of the system require medium sized temporary storage locations (for example, the Content Collector working directory for the Email Server Connector or the Text Extract Connector, and the Content Collector Indexer for the text version of emails to be indexed). This storage location contains highly volatile documents that do not need protection against data loss, as the information stored there can be recomputed if lost. We typically use fast local disks (for example, SAS drives that form a RAID 0 array) for this type of storage. This is the key reason for using two internal disks in the Content Collector servers. The temporary storage locations are then striped across the two drives.

Text index storage

The full text search indexes have a special usage pattern. Data is typically written in large sequential writes during index creation or reorganization, but read at potentially high transaction rates in small, random accesses. To achieve high search performance, one or multiple large disk arrays must be able to provide the necessary I/O operations per second (IOPS) for the search load. We recommend the use of RAID 10 arrays of enterprise level SAS or FC drives with 15 K rpm. The number of drives is not determined by the needed capacity but by the expected search load. The use of date partitioning features for text indexes (see “Performance and scalability” on page 29) is strongly recommended and can reduce the number of needed drives significantly. The latest Solid® State Disk (SSD) technology appears to be a perfect match for this use case, but little practical experience has been gained so far using SSD arrays.

Database storage

Fast database storage is key for good system performance. We recommend the use of RAID 10 arrays of enterprise level SAS or FC drives with 15 K rpm for the enterprise content management archive database and the Content Collector configuration database.

Log storage

Logs are written sequentially and typically are not read. The disk load is relatively low if logs are written to dedicated drives. The drive mechanisms can work without repositioning if only sequential writes are performed. If the logs are written to disks that are also used for other purposes, then the constant repositioning can cause a much higher total disk load and slow the other disk workloads. We recommend using dedicated drives for the system and transaction logs.

Close attention should be paid to the application server logs of FileNet P8 repositories. The Content Collector duplicate detection mechanism can create significant logging activity when many duplicates are encountered because uniqueness constraint violation messages from the database.

Layered detection of duplicate content and impact on storage

Detecting duplicate content typically provides an enormous reduction of the storage space required in the enterprise content management archive. Examples of duplicate content are identical emails that have been stored in and are archived from multiple mailboxes, identical attachments that are used in otherwise different emails, or identical files stored in multiple locations. Content Collector can detect if it encounters a copy of an already archived file, email, or attachment, and only creates a new reference for this new instance of the already archived item in the archive. Besides the tremendous storage reduction,

this action also reduces load on the archive server and email server, because the content does not need to be completely processed again.

In scenarios using both mailbox management and compliance journal archiving, Content Collector also recognizes duplicates between the email archived from the mailboxes and journals. This feature enables companies that must do compliance archiving to also perform mailbox management with virtually no additional storage requirements if all messages processed during mailbox management have already been archived as part of the compliance journal archiving workload or vice versa.

Duplicate emails are, in all scenarios, detected by Content Collector by hashing algorithms. The implementation of the detection and management of other duplicate content (such as email attachments and files) differs depending on the version of the back-end enterprise content management archive used and if storage subsystems are used that provide duplicate detection on the storage layer.

Storage subsystems that offer duplicate detection at the storage layer might benefit from a customized configuration, which helps you avoid the load of checking for duplicates of content for which Content Collector already has performed the duplicate detection (in this example, the email body). Some devices offer a threshold for the file size to determine on which files stored on the device a duplicate detection should be performed. Files smaller than the threshold will not be checked, as the expected benefit of the storage reduction is small. This functionality is often used to increase storage subsystem performance.

We can use this mechanism to avoid duplicate checks on email bodies by setting the threshold to a value that is larger than the majority of the typical email body size. A sample typical maximum email body size is 32 KB for an IBM Lotus Notes® email server and 64 KB if Microsoft Exchange email servers are used.

Text indexer workload

The creation of a text index of archived content is performed by an indexing component attached to the ECM repository. The specific flow of data differs significantly depending on which version of the enterprise content management archive is used. We explain here the data flow that exists depending on the archive used.

If IBM Content Manager is used as archive repository, data (both email or files) are ingested in their native format into the repository. The Content Collector Indexer component can run asynchronously on the Content Manager Library Server to process archived content in batches. Specifically, it retrieves a batch of

newly archived content items from the repository, uses filters to convert text data embedded in the archived items (for example, in email bodies and attachments or in archived files) together with its metadata to an interim XML representation, and submits the batch of XML documents to IBM Net Search Extender for indexing. Net Search Extender then creates a text index using the text content from these temporary XML documents. The process of extracting the text content from binary files is process-intensive and requires the majority of a CPU's resources on the Library Server. The Content Collector Indexer by default spawns a large number of parallel threads (the number of threads is automatically adapted to the number of CPUs present on the system). If CPU resources are to be reserved for other tasks while the indexer is running, then the number of concurrent indexer threads can be configured to a lower number.

Scheduling the Content Collector Indexer in a way such that it lags behind archiving for several hours or maybe even days has a positive impact on indexer performance. In mailbox management email archiving scenarios, there is a good chance that a significant number of duplicates of a specific document have been ingested before the document is selected for indexing if the delay is set to the time it takes for Content Collector to visit all mailboxes once. For all duplicate email instances of a specific message that have been archived as part of one batch of emails that need to be indexed, only a single XML document is created and needs to be added to the text search index. This greatly reduces the number of items that need to be indexed.

In all archiving scenarios, a large batch size of emails to be indexed in one indexer run reduces the number of index updates and keeps the index better organized and requires less index reorganization runs. The down side of scheduling the indexer with a delay is that archived content is not immediately available for search. However, if the archiving policy does not immediately remove content from the source system, then typically the search functionality is not needed immediately and a delay in indexing for text search can be accepted. We recommend setting a long delay to try and capture the majority of duplicate instances of an email in a single index run.

If the FileNet P8 Content Manager is used as the enterprise archive, text indexing differs depending on the type of content archived. For standard file archiving, Content Collector archives binary data to the repository, and the FileNet P8 Content Based Retrieval (CBR) functionality submits this content frequently for text conversion and indexing to the Content Search Engine.

For emails, the process is different. The CPU intensive text conversion takes place as a Text Extract task on the Content Collector server that is running on commodity CPU resources. An XML document (ICCEmailSearch document class) is created on the Content Collector server that holds all the text content of the email, including the attachments. This XML document (called the XML

Instance Text (XIT) document) is stored in the enterprise archive. The Content Search Engine is thus alleviated of the CPU intensive text extraction and merely needs to add the XML document's content to the text index. The XIT document is also stored in the archive for later re-indexing or updates. If a duplicate instance of an email is detected, the Content Collector task route updates the XIT document with information about the duplicate message and stores this information as a new version of the XIT document to the archive. The Content Search Engine indexes the new version and deletes the previous version from the index.

Monitoring and health checks

Content Collector logs provide useful information about the performance and health of the system. Monitoring these log files and performing health checks should be done on a periodic basis. In this section, we describe various Content Collector logs, how to monitor the system using a performance counter, and provide sample queries you can use to get system performance and health information.

Content Collector logs

We recommend configuring Content Collector to store its logs in a single location. If two independent disk drives are available on the Content Collector server, then use one disk drive for the operating system and the Content Collector log files, and the other disk drive for the Content Collector working and temporary directories. The best results are achieved if a local RAID 0 disk array consisting of two enterprise SAS disks are used for the working directory (Email Server Connector) and temporary directory (Text Extract Connector). Monitor the utilization of these disk drives during operation. Utilization of more than 60% may indicate that this disk drive is limiting Content Collector throughput.

The Content Collector logging level should be set to informational, warning, or error logging for a production system. Informational logging provides additional insight into the operation of the system but has an impact associated with it. Informational logging is a good start when a new system is set up. After the system goes into production, use the warning and error log levels as well. Trace-level logging is usually too detailed to be useful for ordinary monitoring, but may be needed to solve a specific issue.

Another general recommendation is to set the log retention period to several days and, particularly for audit logs, have a mechanism configured that archives these logs. Be careful not to archive the currently active log file or otherwise block access to this file, as it may prevent additional log entries from being created.

Be aware that the Content Collector Web Service also writes a log file at a location that differs from the configured location of the other log files (by default, this location is in the `ContentCollector\AFUWeb\afu\logs` directory). You might want to screen this directory for potential problems on a regular basis.

You can also have the audit logging feature enabled for your task routes (enabled by default). The audit logs are in a column format and allow an automated monitoring of the system for successful throughput and possible failures. Closer examination of audit logs (for example, to gather the average size of archived emails) can be performed through a spreadsheet program.

If not already included, we recommend adding the following metadata items to the audit log configuration of your task routes:

- File – File Extension
- File – File Size
- Email – Attachment Flag
- Email – Attachment Count

These values help you understand the size and attachment distribution of content that is being archived and can be used for storage requirement projections and planning. They can also help you understand the throughput numbers of the system.

If you want to have meaningful audit log data on stubbing task routes, then you must add an EC Extract Metadata task (one of the tasks offered by the Email Server Collector) to the task route. This task allows you to have metadata available for logging in stubbing task routes. One metadata item that might be of particular interest in stubbing task routes is “Email – Processing State”, which provides insight into the state of an email when it was selected for stubbing (for example, ARCHIVED, STUB_NOTHING_ADD_TEXT, or STUB_ATTACHMENTS), which may help you understand when analyzing how email moves from one stubbing state to another stubbing state.

Finally, you need to also consider monitoring the logs created by auxiliary components, such as the Content Collector Indexer component (on an IBM Content Manager archive repository) and possibly use the CBR tracing option to monitor the text index creation on the FileNet P8 archive repository.

Understanding Email Server Connector operation

On newly configured systems, it is important to understand in more detail when the Email Server Connector is “crawling” mailboxes to process and when it is processing which mailboxes. The audit log can provide some of this information. A higher level of insight can be gained from the `sysout` log file of the Email Server Connector when the informational logging level is set.

The following two types of operations and related messages are of particular interest:

- Scanning for mailboxes to process

  A message that contains the string "`crawling store`" is listed every time the system finds a mailbox (based on the configured collector data sources, such as user groups to process). At the time of this message, the mailbox is added to the internal list of mailboxes that needs to be processed.

- Opening and processing a mailbox

  A message that contains the string "`Processing store`" is listed every time a mailbox is opened for processing and a query for messages to be archived is issued to the email system.

  By searching for these messages and aggregating them on a per minute basis, a helpful statistic can be created that reports how many mailboxes for archiving were discovered and when ("`crawling store`" message) and when the previously discovered mailboxes are processed and at what rate ("`Processing store`" messages).

Other messages that are frequently seen when the Email Server Connector is configured with informational logging level are "`pruning stores`" messages, which can be ignored. To understand the reason for these messages, you must know that for efficiency reasons the Email Server Connector does not close an email box or email store after it finished processing a message, but rather keeps the store open in case additional messages have to be processed from the same store. Frequently unused stores are detected and closed; this activity is reported by the "`pruning stores`" messages.

Monitoring using performance counters

For a Microsoft Windows®-based server systems archiving solution, the operating system component Windows Performance Monitor, perfmon, (also called Reliability and Performance Monitor) can be used to collect and analyze performance data of all servers in a common place. A single instance of Windows Performance Monitor can be used to collect statistics on system operation from every Content Collector server and from the enterprise content management backend servers and email servers, if they run in the Microsoft Windows operating system. For this setup to work, all servers should belong to the same Windows domain. This setup ensures that the domain user account used to run perfmon can access the necessary counters on all servers. Also, there is no significant time drift between the system clocks of the different machines in the same domain. A 30 second interval is typically a good compromise between fine granular data and acceptable disk space consumption. For monitoring that exceeds a 24 hour period, we recommend

stopping and restarting the perfmon counter log once a day to keep the file size of the log file down (perfmon has a built-in scheduler that can be used to set up this schedule).

Running performance monitor in the 32-bit version: Content Collector performance counters are provided as 32-bit counters. If you run on a 64-bit version of Microsoft Windows, start the performance monitor by running the following command to make sure you use the 32-bit version of performance monitor when monitoring these counters:

```
mmc /32 perfmon.msc
```

This action allows you to view the current values of the counters in perfmon. To be able to record the 32-bit Content Collector counters, you must ensure that the system service responsible for collecting the counter values and storing them in a counter log file are also using the 32-bit version:

- For Windows Server 2003 64-bit, we must modify and restart the Performance Logs and Alerts service. Using the registry editor, open the ImagePath property of the registry key:

  `HKLM\System\CurrentControlSet\Services\Sysmonlog`

  Change the standard value (that points to the 64-bit version) to:

  `%SystemRoot%\syswow64\smlogsvc.exe`

- For Windows Server 2008 64-bit, no registry changes are necessary, as the new Performance Counter DLL Host service was introduced to make 32-bit counters available to 64-bit processes. You simply need to ensure that both the Performance Logs and Alerts service and the Performance Counter DLL Host service run.

In the following sections, we provide a recommended list of performance counters for the individual servers. The first category is a set of counters that should be there for use in following step-by step instructions on which counters to add to your counter log.

Generic counters that should be collected on all servers

The following generic counters should be collected on all servers:

- Processor - % Processor Time (select “All instances”)
- Physical Disk - % Idle Time (select “All instances”)
- Physical Disk - Disk Transfers/sec (select “All instances”)
- Network Interface - Bytes Total/sec (select instances used, if in doubt “All instances”)

- Memory – Available MB
- Process - Percent of processor time (select “All instances”)

Additional counters for the Content Collector servers

On the Content Collector servers, in addition to the above counters, collect all CTMS counters for all instances. Be aware that some instances can only be added if Content Collector is running. In addition, task route specific instances of counters are created the first time a specific task route is loaded. To update the performance monitor counter log, you can:

1. Start the task route service with at least one task route existing.
2. Add the CTMS counters to the counter log, making sure all instances of all counters are added.

This way, instances for task routes created at a later time automatically become part of the counter log and no information is lost due to a missing instance.

Counters for Microsoft Exchange email servers

On the Microsoft Exchange email server, in addition to the generic counters for CPU and disk utilization mentioned earlier, collect the following counters for all instances from the Microsoft Exchange objects:

- MSExchangeIS objects:
  - Active connection count
  - Active user count
  - RPC operations/sec
  - RPC packets/sec
  - RPC requests
  - RPC averaged latency
  - Client: Latency >2, >5, and >10 (three counters)
- MSExchageIS Mailbox objects:
  - Folder opens/sec
  - Message opens/sec

The list of available and important counters may differ based on the Microsoft Exchange version used. To interpret the values, we recommend reviewing Microsoft’s technical documentation on Microsoft Exchange performance topics.

After you have configured all counters (again, it is helpful if all counters exist in a single perfmon counter log), start the counter log. The perfmon creates a log similar to one with the name `c:/perfmon/Content Collector_00001.blg`. Make sure you have about 250 MB available at the log location for each day of logs you want to keep. Verify that it records data for all servers by letting it run for

5 minutes, then open it and graph the CPU and physical disk - % idle time for all machines. This graph gives you a good overview of how busy the systems are.

After you have the graph and counter configuration that you want, you can use the copy-to-clipboard icon to get a copy of your perfmon configuration (this is only for the visualization part). Paste this copy to Notepad and save it to a file. After you restart perfmon, the graph and system monitor configuration is gone and you have to import it again. You can open the text file and import the configuration into perfmon. To perform the import, use the **Properties** button or the Ctrl-Q key combination.

Figure 2 shows the monitor of the journal archive activities of Lotus Domino emails to the IBM Content Manager archive repository.

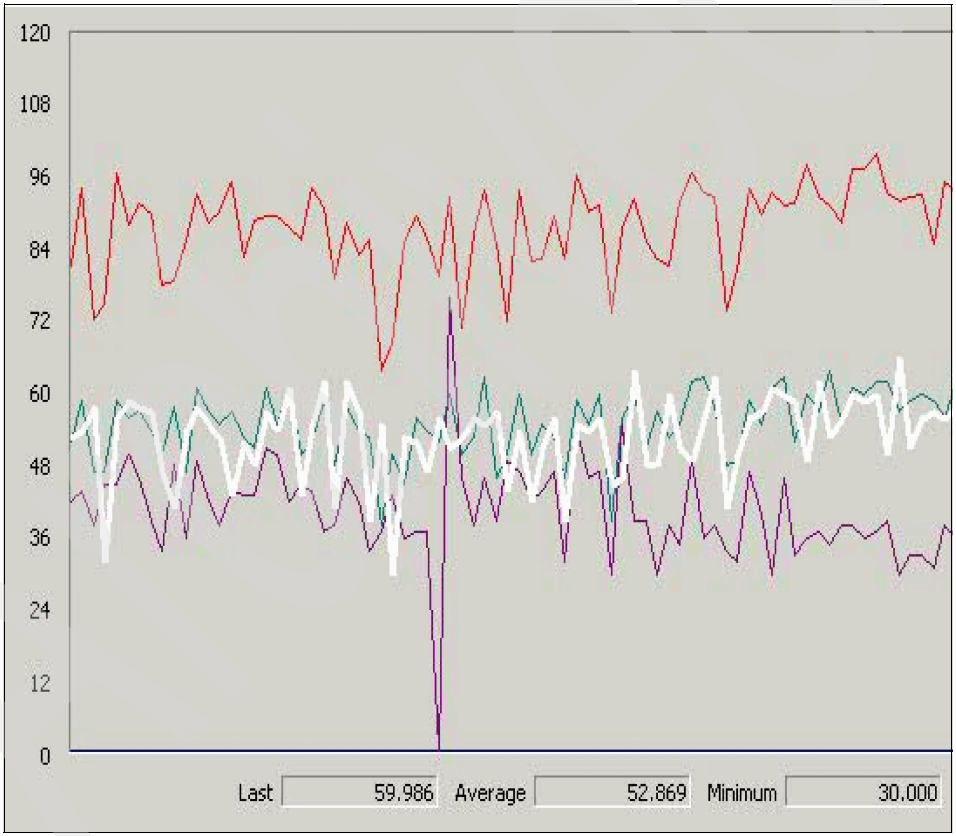

Figure 2 Monitoring the archive of Lotus Domino to IBM Content Manager

Figure 2 includes the counters for the % of Processor Time, the number of Accessed Entities/sec, the Item Backlog, and the Documents Created/sec. With one look at the graph, you can see that total processor utilization of this 2-way

AMD Opteron dual-core machine is about 90%. The number of Accessed Entities/sec and Documents Created/sec tracks each other closely at an average of 53 emails per second, and the Item Backlog, which is the internal processing queue of the Task Route service, is constantly filled with 35 to 40 items that must be processed.

Querying the archive and the full text indexing engines

The enterprise content management archive repository stores metadata on archived contents in underlying databases. In this subsection, we provide you with a set of sample queries for these databases to check how many emails have been archived, and what the number of duplicate emails, the number of attachments, and duplicate attachments are. If content other than email is archived, the queries are similar to the queries used in email archiving to find the attachment count. The sample queries shown here are for the IBM DB2® database system and need to be adapted to work on other database systems.

To check how many items you currently have in the archive, the following queries may be helpful.

Queries for an IBM Content Manager based archive repository

In the Content Collector 2.1.1 data model, you can perform the following queries:

- To get the number of distinct email instances (without duplicates), run the following query:

  ```
  select count(*) from ICCEmailCmpLD001 with ur;
  ```

- To get the number of total email instances (with duplicates), run the following query:

  ```
  select count(*) from AFUEChild001 with ur;
  ```

- To get the number of distinct attachments (without duplicates), run the following query:

  ```
  select count(*) from ICCAttachments001 with ur;
  ```

- To get the number of total attachment instances (with duplicates), run the following query:

  ```
  select count(*) from AFUAChild001 with ur;
  ```

In the Content Collector 2.1.0 data model, you can perform the following queries:

- To get the number of distinct email instances (without duplicates), run the following query:

  ```
  select count(*) from ICCEmailLD001 with ur;
  ```

- To get the number of total email instances (with duplicates), run the following query:

  ```
  select count(*) from AFUEChildLD001 with ur;
  ```

To discover the views (for example, ICCEmailLD001) that are easier to use than the base table names, you can use the following query:

```
select COMPONENTVIEWID, COMPONENTTYPEID, ITEMTYPEID, COMPONENTVIEWNAME,
CONCAT (CONCAT('ICMUTO',CAST(COMPONENTTYPEID as char(4))),'001') TABLE
from icmstcompv iewdefs where itemtypeid in (select keywordcode from
icmstnlskeywords where keywordclass=2 and keywordname in
('ICCEmailCmpLD', 'ICCAttachments')) with ur
```

To see how many items were ingested or modified on a particular day, you need to adapt the items table to match your system's table names by running the following query:

```
select date(changed) date, count(*) count from ICMSTITEMS001001 group
by date(changed) with ur;
```

The name of the database table of open tasks is formed by the combination of AFU1FTIOPEN<*item type ID*>, where <*item type ID*> is the numeric item type ID. Add preceding zeros if necessary to make the ID five digits long.

Monitoring the backlog of items that the Content Collector Indexer has to process can be done by running a query similar to the following (adapt the last digits based on the internal number of the item type you want to monitor):

```
db2 "select count (*) from ICMADMIN.AFU1FTIOPEN01007"
```

The Content Collector Indexer component has built-in performance statistics that it can display on the console output. Review the product documentation for details.

To capture and persist the console performance statistics output, run the indexer from within a script and redirect the console output to a file. On AIX®, we use the **nohup** command when running the indexer component manually. This command allows the operation to run detached from the console and hence avoid process interruptions if the console is closed on accident or loses the connection.

Also, the indexer configuration can be extended to run additional scripts before or after the actual update of the text index by using DB2 Net Search Extender. This

function can be used to monitor index disk utilization before and after the index updates to gather information about the typical increase in index size for each batch of documents added to the index.

To see how many items the text index currently holds, query the db2ext.textindexes table for the column number_docs:

```
db2 "select number_docs from db2ext.textindexes"
```

If you have more than one text index, you have to show additional columns to identify which text index belongs to which item type.

To see how many items DB2 Net Search Extender has currently processed during an index update operation (if the monitoring provided by the Content Collector indexer component is not activated), run the following command to retrieve the current statistics from the index update process:

```
db2text control list all locks for database icmnlsdb
```

Queries for FileNet P8 archive repository

To discover the object ID by its symbolic name, use the following query:

```
db2 => SELECT object_id,symbolic_name FROM ClassDefinition
x'9B06EAE70C99F944B9CA279435EA42E2' ICCMailInstance
x'974F613023895F45A804E28A65E97FBA' ICCMailJournal
x'F890E9972A0123469EA9C3ED141DBE72' ICCMail
x'BE22769A46A02041AE1B19779OB9B5F4' ICCMailSearch
```

Where:

- *ICCMailInstance* (also called the email instance (EI)) is a custom object. Each email that has been archived creates one such instance objects. It holds only metadata.
- *ICCMail* (also called the distinct email instance (DEI)) is a regular object derived from the Document class. It holds mostly the content elements for the email body and the attachments.
- *ICCMailSearch* (also called the XML instance text (XIT)) is a regular object derived from the Document class. It has versioning and Content Based Retrieval (CBR) enabled and holds metadata for search result display, and a content element representing the text content of the email plus attachments in XML format for indexing.

You can use the object_id values in subsequent queries.

To determine the item count ICCMail (distinct email instances) and ICCMailSearch (text-only version of emails for CBR) using the docversion table, run the following command:

```
db2 => SELECT count(*) FROM OSFMD.DocVersion WHERE
object_class_id=x'BE22769A46A02041AE1B197790B9B5F4' with ur;
```

To determine the item count ICCMail (distinct email instances) and ICCMailSearch by date range, run the following command:

```
db2 => SELECT count(*) FROM OSFMD.DocVersion WHERE create_date between
'2009-10-21-15.10.00.000000' and '2009-10-21-16.12.00.000000' and
object_class_id=x'F890E9972A0123469EA9C3ED141DBE72' with ur;
```

The different create dates of the distinct email instances in ICCMail and the last date the text-only version in ICCMailSearch was updated due to duplicate messages can lead to differences in the reported numbers of items created in a specific date range.

You can query the ICCMailInstance objects (the total number of emails archived including duplicate items) in the same way, but as these are custom objects, they are part of the generics table, not the docversion table. Run the following query to query objects:

```
select count(*) from generic where
object_class_id='E7EA069B-990C-44F9-B9CA-279435EA42E2' with ur;
```

Performance and scalability

A newly built system requires initial tuning for maximum performance before going into production. Performance tuning is a complex and iterative process. This tuning involves establishing quantitative objectives, constant system monitoring, and selective and careful tuning to ensure that the objectives are met over time.

Depending on the results of monitoring, you and your performance team should adjust the configuration of the system's resources and applications to maximize system performance, reduce down time, and avoid potential problems or a crisis.

When you are working on improving system performance, we recommend that you follow this set of general guidelines for performance tuning:

- Establish quantitative, measurable, and realistic objectives.
- Understand and consider the entire system.
- Change one parameter at a time.
- Measure and reconfigure by levels.

- Consider design and redesign.
- Remember the law of diminishing returns.
- Recognize performance tuning limitations.
- Understand the configuration choices and trade-offs.
- Do not tune just for the sake of tuning.
- Check for hardware and software problems.
- Put fallback procedures in place before you start tuning.

Tuning Content Collector for desired throughput

Content Collector has two key parameters to control system throughput:

- Thread count
- Queue size

These parameters can both be set by using the Configuration Manager by selecting **Task Route Service Configuration** in the Tools menu.

Thread count

The Thread count parameter determines the maximum number of concurrently processed items. As a rule of thumb, a number of two to four threads for each physical CPU core should be configured for high throughput scenarios. Content Collector ships with a default setting of 16 threads, which is adequate for most deployment scenarios. If a high number of CPU cores are available, the number of threads may be increased. If the throughput of the Content Collector server should be throttled, the number can be decreased. A typical scenario for throttling is to reserve CPU resources for the interactive search, view, and restore workload or to prevent an overloading of the email or file server. You should not configure a thread count of less than two threads.

Queue size

The Queue size parameter defines the number of items that a collector can submit. It acts as a buffer between the submitting connector and the processing tasks. In practice, values between 64 items (file archiving) and 128 items (high-throughput email archiving deployments) have been used with good results. You can use a maximum value of up to 256 items for eight core servers in Lotus Domino based systems and when scale-out setups are used.

Measuring raw throughput from the email system

Figure 3 on page 31 shows a sample task route that can be used to ingest emails from email servers to determine the maximum throughput at which data can be read from the email servers without archiving or modifying the content. Actual

archiving throughput will often be significantly lower than for archiving, as the messages must also be written to the enterprise content management archive and be marked as archived on the email system. Despite this uncertainty, much information can be gained from a sample processing run with such a task route (the same principle can be used for file archiving scenarios):

- What is the maximum throughput that emails can be read from the current email server system?

  If this number is already below or only slightly above the required throughput, then the reasons for this limitation (in this example, email-server disk IO limitations or WAN network latency issues) must be resolved or the system architecture should be adapted for example by utilizing multiple email servers in parallel.

- What are the characteristics of the data to be archived, for example with respect to size?

  This information can be obtained from the audit logs, which can be imported into a spread sheet software and analyzed.

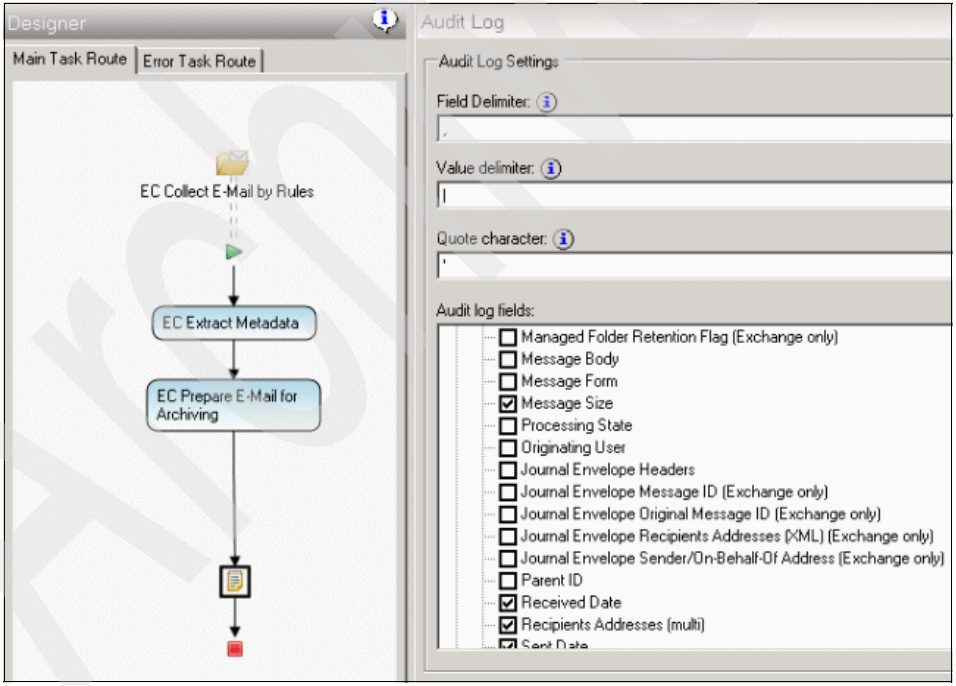

Figure 3 Task route for collect email by rules, extract metadata, and pre for archive

Figure 3 on page 31 shows a task route that consists of the following collector and tasks:

- EC Collect Email by Rules collector.
- EC Extract Metadata task: This task extracts metadata, such as recipients and message size, that is available for later usage by the Audit task.
- EC Prepare Email for Archiving task: This task writes a temporary file to the Content Collector working directory.
- Audit Log task: The right pane of Figure 3 on page 31 shows the configuration of data that should go into the audit log. As shown in this example, the metadata fields for Message Size and Recipients Addresses and Attachment Count might be of interest to you if you want to determine average message size, the average number of recipients, and average number of attachments to expect.

A similar approach can be used to understand the potential archiving throughput that can be achieved with the email server system deployed. To force the system to also retrieve the email attachment data from the email server, the EC Extract Attachments task should be added after the EC Prepare Email for Archiving task. If the throughput achieved differs greatly between a task route with and without attachment extraction, then an email server disk or network bandwidth constraint is likely the bottleneck.

Scaling Content Collector deployments

Content Collector can be scaled both vertically (by using a larger computer with multiple CPU cores) or horizontally (by using additional Content Collector server instances). The choice of how to scale depends greatly on which email server system is deployed. For IBM Lotus Domino based email systems, a vertical scaling using 2-way or 4-way systems with current multi-core CPUs machines provides good results. A scale-out approach using multiple smaller machines is also possible. The most economical choice in terms of hardware cost is often to use multiple servers, but this approach has to be weighed against the administrative cost of managing these systems.

For Microsoft Exchange based systems, we usually require the use of multiple servers to scale Content Collector deployments, because a single server is limited by the email interface in regards to throughput. A larger number of smaller servers is typically more beneficial in these environments.

In scale-out setups, we have had the best results if all Content Collector servers are of the same hardware configuration. The acting primary server has a slightly higher CPU requirement (it manages the workload distribution) than the extension nodes in a scale-out setup. However, the built-in high-availability

mechanism makes it hard to predict which server takes on the role of the primary server and may cause this workload to shift to another server during operation.

Content Collector is by default optimized to achieve the highest possible throughput ratios for archiving large mailboxes, such as journal mailboxes. In a scale-out setup, this means that all Content Collector servers combine their processing power to work on the currently selected mailbox. The servers process one mailbox at a time and the set of currently selected items from this mailbox is finished before they move on to the next mailbox. In the standard configuration, this means that the email server load created by Content Collector is always focused on one email server at a time. If an email server is at its processing limit with the load that a single Content Collector server creates or if the bottleneck is in the network connection between the Content Collector server and the email server, then adding more Content Collector servers in a scale-out setup will put more load on the existing bottleneck. This situation in turn leads to an overload situation in which the total work performed by the setup declines. In such situations, the configuration needs to be changed to archive content from multiple servers in parallel.

Tuning Content Collector to process multiple email sources in parallel can be achieved through assigning particular Content Collector instances to specific email user groups that reside on different servers. If IBM Lotus Domino is used, multiple Email Server Connector crawler threads can be configured to achieve a similar effect without the need for multiple Content Collector instances. The instructions about how to adapt the crawler threads differ based on the version of Content Collector used and should be requested from your IBM support contact.

Ensuring scalability for text search

Searching content on large text indexes is an expensive operation. To be able to provide full text search to users or for high volumes of regulatory searches in an economical fashion, it is important to minimize the amount of work that needs to be done by the search component to fulfill a single search request. The largest reduction on search load can be achieved by educating users to always specify a time range for their full text search.

Tip: Always specify a default date range for user searches in mailbox management scenarios.

A default date range can be suggested to the user in the Content Collector search user interface by setting the environment variable AFU_DATE_RANGE_IN_MONTHS to a value greater than zero. This results in the date fields of the search form being populated by default with the number of specified months plus the current month. If a user enters a simple query, for

example, a specific word in the subject field, and submits this query without changing the date range, then only the number of specified recent months are searched, not all of the archive. The default behavior (if the environment variable is not present in the environment in which the Content Collector Web applications are started) is not to populate the date fields with any values and therefore relying on the user to specify a meaningful date range.

Using date range constraint searches in conjunction with date partitioned text search indexes has the following two advantages:

- It reduces the number of potential search results that need to be processed and provided back to the user client.
- If data segmentation based on date ranges can be configured in IBM Content Manager (through the use of multiple item types with date-range segmentation) and in FileNet P8 (through archive date segmentation for index areas), the date range specified with a particular query reduces the search server load by limiting the query to those item type text search indexes or index collections respectively that correspond to the specified date range. For example, if emails for the last four years have been archived with a date range segmentation of one index for every 6 months, then a query for emails from year one will only search the two full text indexes associated with year one (one quarter of the total text index size). This action reduces search server disk IO load considerably. A storage architecture that moves older text search indexes (that are typically used less frequently) to less expensive storage can reduce costs.

More information about how to enable date range segmentation can be found in the documentation of FileNet P8 by selecting **System Administration** → **Content Engine Administration** → **Content-based retrieval**. This feature is supported starting with FileNet P8 Version 4.5.1. It is important that the Content Collector style set ICC_FileSystem_PushAPI_2.1_p8cse_4.5.1 or later is used with this feature and the Content Collector Search Application configuration is modified to query for date ranges using the same database property (ICCMailDate) that the Content Engine index segmentation has been configured to use. For more information, refer to the following address:

`http://publib.boulder.ibm.com/infocenter/p8docs/v4r5m1/index.jsp?topic=/com.ibm.p8.doc/ce_help/cbr/cb_about_verity_partitions.htm`

The selection of proper date ranges depends mostly on the daily archiving load. One index partition can hold between a few millions to tens of millions of documents depending on a variety of parameters and the specific usage scenario. For email archiving, a general rule of thumb at the time of this writing is to put less than 50 million emails into an IBM Content Manager Item Type and use FileNet P8 to configure your index areas in a way that a new collection is

being created for every 5 to 8 million emails. These numbers are not hard limits; they are provided as a guideline for system design planning.

Proper configuration of virus scanners

Virus scanners should be configured to exclude the Content Collector temporary and working directory for two reasons:

- Performance. Content that is being archived has typically been processed at an earlier stage by a virus scanner and the impact of checking the temporary files created during archiving may be a waste of resources and slow archiving operation.
- In the case of a suspected virus, finding the locks on Content Collector temporary files leads to errors and in Content Collector processing. These errors need to be analyzed and the reason determined.

Carefully plan how virus scanners are to be run on enterprise archives (if at all), including the text indexing engine. The deeply nested file directory structures created in file storage areas and in the temporary areas used during text indexing may cause a virus scanner to put significant disk load on the server when analyzing these directories. Similar rules apply to desktop text indexing products, for example, Microsoft Windows Search. These services should be disabled to prevent them from creating unnecessary system load.

We strongly recommend not running a virus scanner on the text index files created by the text indexing engines. We have seen scenarios where a virus scanner, due to erroneously detecting false positives in the text index, blocked access to the index files in critical index update phases, resulting in corrupt files that then need to be regenerated.

Planning for high availability and load balancing

A Content Collector server provides two classes of services:

- Archiving services
- Interactive services for search, view, and restore operations

Archiving services

Content Collector provides high availability for archive services if more than one Content Collector server is installed in a scale-out setup.

The archiving functionality is governed by the task routing service. It has an internal concept of a primary node that, besides performing archiving tasks, is

managing the work distribution and extension nodes that act as secondary of the primary and are only archiving items that the primary node assigned to them.

In case of a scale-out Content Collector installation, the task routing service on the primary node distributes work evenly to all Content Collector server instances (itself and all extension nodes). If the task route service on the primary node detects that it lost communication to a secondary node, it will avoid submitting additional work to this node and treat all work that was currently assigned to this node as lost. In the case of emails, these emails will not be marked as archived and they will be picked up by Content Collector again the next time the originating mailbox is checked for emails to be archived. If the current primary node ceases operation, then the remaining secondary nodes detect this situation and one of these nodes assumes the responsibilities of the primary to ensure continued operation.

Interactive services for search, view, and restore operations

An external mechanism is required to provide for failover if the Content Collector server providing these interactive services fails.

The interactive services for search, view, and restore operations are called directly by the Content Collector extensions in the user's search client or by the email web access server. The services are provided by a web services that in standard installations run in an embedded IBM WebSphere® Application Server on each Content Collector server. The Content Collector server that is the designated primary server during installation will by default be the server that provides the interactive services to the clients, so the host name of this machine is used as the host name (or alias) in the stub links and for the user email search application.

If this server becomes unavailable, a mechanism must be in place that makes one of the extension nodes accessible to the email client extensions under the configured host name or host name alias. Interruptions in the interactive services are typically much more noticeable (as they impact users) than, for example, interruptions in the archiving process.

Content Collector relies on a configuration database that requires its own high availability mechanism. All Content Collector servers stop processing if this database becomes unavailable.

To provide high availability for the interactive services, there are a number of possible solutions. Two examples that have been used in the field are:

- Use of cluster services

  A stand-by server will be activated when there is a failure of the server providing the web services. If such a setup is chosen, then a possible

architecture would consist of two servers in a scale-out setup (active-active for archiving services) and cluster services are used only for high availability of the interactive services (active-passive for interactive services).

- Use of a load balancer in front of the web services components

  The use of a load balancer is a good way of achieving high availability. At the same time, enable Content Collector to deal with a high number of interactive requests by using the web services components of all Content Collector servers in parallel.

In a load balancer setup, the system is assigned the archiving host name alias that is used in the Content Collector configuration as the single alias for all Content Collector servers (for example, archive.example.com). The email client extensions then contact this machine with all their interactive requests (using https). The load balancer in turn forwards these requests to the cluster of Content Collector servers.

The load balancer itself introduces a new single point of failure into the system that also needs to have a backup to provide for high availability.

In addition, the load balancer can be used to terminate the https connection and forward the connection to the Content Collector servers using http. This releases the Content Collector server from the expensive https handshake operations.

Figure 4 shows the Content Collector system setup using load balancer.

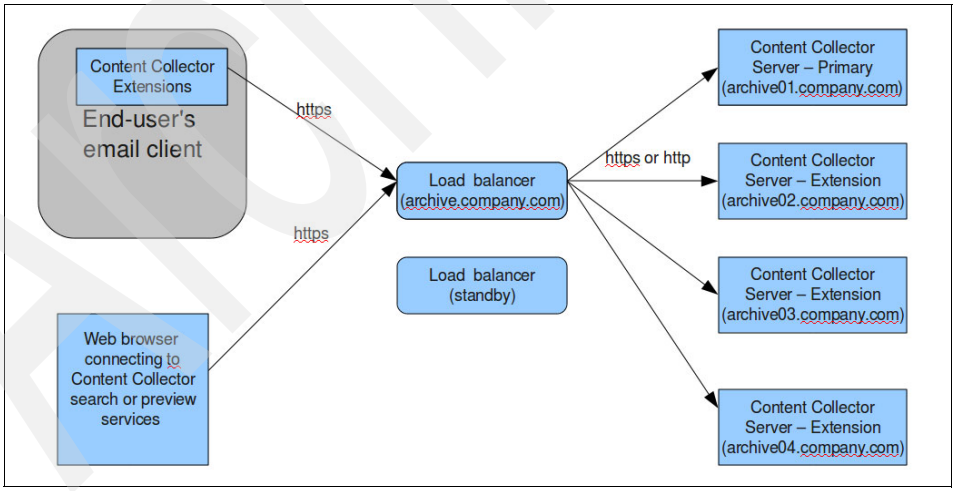

Figure 4 Using Load Balancer for Content Collector system

In this configuration (Figure 4), the load balancer in front of the Content Collector servers receive all requests to the Content Collector Web Application and to the

Configuration Web Service. The requests are for the host name archive.company.com. All requests are received over SSL on port 11443. The SSL tunnel is terminated at the load balancer. The requests are then forwarded to the Content Collector servers over HTTP to port 11080.

The sessions must be *sticky based on the negotiated session ID (a cookie)* to ensure that a user who opened a session and authenticated to the Content Collector server is continuing the session with the same server.

To discover if the Content Collector Web Application is still alive and operational on each Content Collector server, the load balancer can periodically invoke the following URL of the Content Collector Web Application:

`http://<Content CollectorServer>:11080/AFUConfig/status`

The configuration web service needs to be running on every Content Collector server. It uses JDBC to access the configuration database. To configure the JDBC access on all nodes, perform the following steps:

1. After successfully setting up the primary node, copy over the `jdbcInstall.cmd` file from ….\ContentCollector\ctms\scripts.
2. Edit the password, path, and cell name to match the archiving user.
3. Copy the `jdbcInstall.cmd` file to all secondary nodes, in the same directory.
4. Run **`jdbcInstall.cmd`** on every node.

Now all nodes will be enabled for configuration access. This situation can be verified by accessing the virtual status URL on every node.

To establish this setup, the easiest approach is to set up the operation of all servers independently, test them independently by redirecting all traffic to one server at a time, then changing the load balancer configuration before activating load-balancing across all instances.

Use of Configuration Manager in a failover scenario

The Configuration Manager component is by default installed to provide write access on the primary node of a Content Collector deployment. A table in configuration database governs write access to the configuration.

If there is a primary node physical or operating system failure, it may become necessary to modify the configuration using one of the extension nodes of the Content Collector setup. Perform the following steps:

1. In the system registry under HKEY_LOCAL_MACHINE\SOFTWARE\IBM\Content Collector\Server, the value for the installtype type must be changed from `NODE_B` (for an extension node) to a value of `NODE_A` (indicating a primary node).
2. The “IBM Content Collector GUI Components” service must be started after the above key value is changed.

After these steps are completed, the Configuration Manager can be started, and it allows write access to the configuration database. It is dangerous to have multiple servers enabled for write access at the same time. If multiple Configuration Manager instances open the configuration database for write access simultaneously, the configuration can become invalid.

Planning for backup and disaster recovery

Although the backup of Content Collector components, the enterprise content management archive, and text search indexes can be performed independently, ideally an all inclusive backup and restore strategy should be developed for the entire archiving environment. This scenario is essential for preserving state between applications and therefore preventing possible data loss.

The Content Collector servers only hold transient data that does not need to be backed up. A single backup after installation and upgrades of the servers may be sufficient. The only component that requires backup is the configuration database for Content Collector. We recommend placing this database on the enterprise content management archive database system to take advantage of the archive database backup mechanism.

For disaster recovery purposes, we recommend hardware level replication, which maintains write order fidelity, across the key data sets (see below). However, replication to a disaster recovery site is not a substitute for conventional backups. If an error occurs that corrupts data at the primary site, that corruption will be replicated to the disaster recovery site. If replication is the only backup, recovery will not be possible.

The data maintained by each building block relies on the referential integrity of other building blocks. If the archiving environment no longer conforms to cross application relationships, inconsistencies can occur. Some of these inconsistencies may be difficult to resolve; other may be impossible to resolve. A backup strategy should maintain this relationship and complete the backup of the

environment while the data is at rest. Achieving this daily within a backup window is complicated due to the performance characteristics of tape.

By using snapshot capabilities of SAN and NAS storage or a logical volume manager, the downtime required for backups can be minimized. Assuming all of the mutable data (the database for metadata, the full-text index collections, and the file storage areas) for the enterprise content management application is stored on a SAN, a snapshot can be used to capture a point-in-time copy of the application in a state that is suitable for a file system backup. While application components are shut down, a snapshot of the mutable data is taken. Most snapshot technologies are near instantaneous, allowing the application components to be brought back up only a short time after being shut down, without having to wait for the transfer to tape. The snapshot is then mounted to an alternate location, allowing the backup software to back up the file system.

The use of fixed content devices may require a continuous differential backup in addition to a snapshot strategy. Having orphan data (that is, content without corresponding metadata) is preferred over the opposite situation, that is, metadata with the corresponding content missing, because it will not cause any user-visible errors. For this reason, the fixed storage backups should be run after the snapshots of the mutable data have been taken. Again, this approach minimizes the length of the backup window, in comparison to shutting down all services, backing up all the devices, and then restoring the services.

This backup strategy is appropriate if any of the following circumstances apply:

- The groups supported by the application and the technology group cannot tolerate large backup windows or the operational hours of the business make the prolonged backup windows untenable.
- The groups supported by the application cannot tolerate the financial liability of a large recovery point objective and expects to lose data of one day or less.
- The groups supported by the application can tolerate extended periods of time in the order of days where the application is not available for general use.

The use of snapshots reduces the backup window. They also ensure that the backup is a referentially accurate across all application components, thus greatly improving the chances of a successful restore.

Making a coordinated snapshot of all the mutable data requires either storing all of the data on a single storage array or using storage virtualization products such as IBM SAN Volume Controller. Additionally, a snapshot does not need to be backed up at the file system level. Greater performance can often be achieved by backing up the volume at the block level; this creates a bit for bit copy of the original content.

The team who wrote this paper

This paper was produced by a team of specialists from around the world working at the International Technical Support Organization, Rochester Center.

Wei-Dong Zhu (Jackie) is an Enterprise Content Management Project Manager with the International Technical Support Organization. She has more than 10 years of software development experience in accounting, image workflow processing, and digital media distribution. Jackie holds a Master of Science degree in Computer Science from the University of the Southern California. Jackie joined IBM in 1996. She is a Certified Solution Designer for IBM Content Manager. Jackie has managed the production of numerous Enterprise Content Management IBM Redbooks® publications.

Markus Lorch is an Advisory Software Engineer at IBM Germany. He has more than five years of experience in the field of enterprise content management and content discovery software development. Markus leads the performance and scalability engineering activities for IBM Content Collector. He holds a Ph.D. in Computer Science from Virginia Tech. His areas of expertise include software performance, content archiving, text analytics, enterprise search, and distributed system security. He has more than 20 peer-reviewed publications on authorization and security topics, Grid and cluster computing, and semantic search mechanisms.

Thanks to the following people for their contributions to this project:

Dieter Schieber
Silke Wastl
Thorsten Hammerling
IBM Software Group, IBM Germany

Now you can become a published author, too!

Here's an opportunity to spotlight your skills, grow your career, and become a published author—all at the same time! Join an ITSO residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks in length, and you can participate either in person or as a remote resident working from your home base.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Stay connected to IBM Redbooks

- Find us on Facebook:

  http://www.facebook.com/IBMRedbooks

- Follow us on Twitter:

  http://twitter.com/ibmredbooks

- Look for us on LinkedIn:

  http://www.linkedin.com/groups?home=&gid=2130806

- Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:

  https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm

- Stay current on recent Redbooks publications with RSS Feeds:

  http://www.redbooks.ibm.com/rss.html

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:
This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

This document REDP-4704-00 was created or updated on October 13, 2010.

Send us your comments in one of the following ways:
- Use the online **Contact us** review Redbooks form found at:
  ibm.com/redbooks
- Send your comments in an email to:
  redbook@us.ibm.com
- Mail your comments to:
  IBM Corporation, International Technical Support Organization
  Dept. HYTD Mail Station P099, 2455 South Road
  Poughkeepsie, NY 12601-5400 U.S.A.

Trademarks