Benchmark Testing Results: OpenText Email Monitoring and Records Management Running on SQL Server 2012

Running OpenText Email Monitoring and Records Management on Microsoft SQL Server 2012 provides excellent performance and scalability

Technical White Paper
Published: July 2012
Applies to: Microsoft SQL Server 2012

Abstract

In February 2012, OpenText and Microsoft conducted performance and scalability testing on the Email Monitoring and Records Management components of the OpenText ECM offering running on Microsoft SQL Server 2012 data-management software.

The benchmark testing was very successful, with a peak ingestion of 995,000 email messages in a single hour, 14.8 million messages in a 24-hour period, and 171 messages per second—up to 15 times the typical ingestion volume.

This white paper presents the details of the benchmark testing and the additional benefits that SQL Server 2012 brings to the OpenText ECM Email Monitoring and Records Management components—confirming that SQL Server 2012 is an excellent choice for OpenText Email Monitoring and Records Management solutions.

Table of Contents

Introduction

Email has transformed the way that companies do business. Yet despite its unparalleled utility, the potential for mismanagement of email, paired with its rampant and often insurmountable growth, can make it a staggering liability for corporations. Email messages and attachments can represent business records that organizations must retain and manage securely to support regulatory compliance, avoid legal fines or litigation costs, and satisfy auditing requirements.

> "OpenText is one of our premier content-management partners, and together we are delivering solutions that enable organizations to use business information to achieve strategic advantage. I expect that OpenText will continue to play an important role in our product development efforts aimed at creating new content-management technologies that will help our customers optimize business processes, reduce costs, and utilize information in increasingly powerful ways."
>
> *Steve Ballmer*
> *Chief Executive Officer*
> *Microsoft*

It is predicted that the number of regulations governing email—along with the sheer number of emails transmitted throughout organizations—will continue to increase. Organizations need a solution for capturing, managing the lifecycle, and streamlining discovery processes for all regulated email communications.

The OpenText Email Monitoring solution helps organizations archive, retrieve, and classify all email content that is sent or received by specific accounts within an organization, helping to ensure that the complete range of electronic communications of specific users and groups is fully archived and auditable. OpenText Email Monitoring and other OpenText applications are built on a shared-services foundation, which provides a common integration layer and access to content that originates from other systems such as email, enterprise resource planning (ERP), file system, and productivity applications, such as Microsoft SharePoint.

> "By furthering our close collaboration, OpenText and Microsoft are creating a new dynamic in the marketplace and raising the bar for effective Enterprise Information Management in major industries such as utilities, oil and gas, government, financial services, and legal. In each case, we're demonstrating the power of combining productivity applications from Microsoft with information governance solutions and industry-specific ECM expertise from OpenText. Our relationship with Microsoft is a critical part of our strategy and we will continue to be a leader in delivering new solutions for our joint customers."
>
> *Mark Barrenechea*
> *Chief Executive Officer*
> *OpenText*

OpenText and Microsoft work together to deliver solutions that extend the Microsoft platform, including Microsoft Office, SharePoint, Microsoft Exchange, Windows Azure, and Microsoft SQL Server, with industry-specific content applications, combining the power of SQL Server with OpenText's deep understanding of information governance and business-process management. Running OpenText solutions on SQL Server helps organizations scale their database operations with confidence and improve IT and developer efficiency—all at a lower total cost of ownership (TCO) than competitive solutions.

In February 2012, OpenText and Microsoft conducted performance and scalability testing on the email monitoring and records management components of the OpenText ECM offering running on the SQL Server 2012 data-management software. This white paper details the results of this benchmark testing and provides links for further information.

Introducing the OpenText Email Monitoring and Records Management Functionality

OpenText helps customers address challenges related to their information—to gain better business insight, to create a positive business impact, to increase process velocity, to reduce risks related to information governance, and to address information security concerns. The OpenText ECM offerings, including Email Monitoring and Records Management, let organizations easily and rapidly access relevant content and leverage it to create business value without compromising their compliance and security needs.

The key benefit of the OpenText Email Monitoring solution is to manage incoming and outgoing email messages; more importantly, the OpenText Email Monitoring solution effectively classifies the email messages and provides defensible disposition of those messages immediately when required. The system therefore needs to remain available while ingesting and defensibly deleting high volumes of content.

For more information about OpenText ECM offerings, visit www.opentext.com/ecm.

Benefits of SQL Server 2012

SQL Server 2012 is a comprehensive, integrated, and enterprise-ready data-management software solution. It provides a reliable, cost-effective, low-maintenance database framework for OpenText that can support the largest and most process-intensive deployments. Hundreds of enterprises are currently running 10 terabyte (TB) and larger transactional databases on SQL Server.

Running OpenText Email Monitoring and Records Management functionality on SQL Server 2012 provides customers with many benefits:

- *Six nines (99.9999 percent) uptime availability for SQL Server 2012*
  SQL Server customers can protect their mission-critical databases from downtime and data loss with six nines uptime availability.[1,2]
- *Faster to deploy*
  On average, SQL Server database administrators (DBAs) can install and configure new database servers in 1.5 hours, while the largest competitor's DBAs can take 6 hours.[3]
- *Lower hardware cost*
  SQL Server can run on standard commodity server hardware, which can dramatically lower the TCO for customers.

---

[1] http://www.microsoft.com/sqlserver/en/us/product-info/why-sql-server.aspx
[2] http://www.microsoft.com/casestudies/Microsoft-SQL-Server-2008-R2-Enterprise/Stratus-Technologies/Protect-your-mission-critical-databases-from-downtime-and-data-loss-with-six-nines-uptime-availability/4000007136
[3] http://www.alinean.com/PDFs/Microsoft_SQL_Server_and_Oracle-Alinean_TCA_Study_2010.pdf

- *Lower software costs*
  The list price of SQL Server is a third of the largest competitors' cost;[4] in addition, SQL Server includes major database-related features, such as high availability; remote disaster recovery; partitioning; data compression; transparent data encryption; spatial; master data management; complex event processing; extract, transform, and load (ETL); online analytical processing (OLAP); data mining; reporting services; and self-service business intelligence (BI) tools. Competitors' licensing models add costs for options and add-ins.[5]
- *Simpler systems management and lower staffing costs*
  SQL Server database administrators can typically manage four times as many physical databases as a competitor's DBAs, leading to an estimated annual savings of $5,779 in administrative costs per database, a 460 percent difference in annual cost of administration per database.[6,7]
- *Most secure of any of the major database platforms*
  Since 2002, SQL Server has recorded the fewest reported vulnerabilities as compiled by the National Institute of Standards and Technology (NIST).[8]

With SQL Server, OpenText customers can save with reduced licensing, hardware, administration, and support fees, which translate into substantially lower costs over the life of the system.

For more information about SQL Server 2012, visit www.microsoft.com/sql.

Benchmark Testing Overview

As part of their strategic alliance, OpenText and Microsoft ran "real-world" benchmarking tests at the Microsoft Enterprise Engineering Center (EEC) in Redmond, Washington. The EEC is dedicated to strategic partnerships with industry leaders such as OpenText, providing the hardware, software, and services required by Microsoft customers and product groups to build and validate end-to-end solutions.

The benchmarking tests were designed to evaluate the performance and scalability of the Email Monitoring 10.2 and Records Management 10.2 components of the OpenText ECM offering running on SQL Server 2012. The benchmark tests measured the peak ingestion of email messages, in addition to the throughput when ingestion and dispositions of email messages were running concurrently, a typical real-world use scenario.

---

[4] http://www.microsoft.com/sqlserver/en/us/tools/cost-savings-calclator.aspx
[5] http://www.microsoft.com/sqlserver/en/us/product-info/competitor-compare.aspx
[6] http://www.microsoft.com/sqlserver/2008/en/us/compare-oracle.aspx
[7] http://www.alinean.com/PDFs/Microsoft_SQL_Server_and_Oracle-Alinean_TCA_Study_2010.pdf
[8] http://itic-corp.com/blog/2010/09/sql-server-most-secure-database-oracle-least-secure-database-since-2002

Test Environment Details

The test environment consisted of 26 servers of varying configuration, all running the Windows Server 2008 R2 Enterprise operating system. The core server environment consisted of a database server running SQL Server 2012, a server running the Archive Server component of the OpenText ECM offering, a search and index server, a server running the Content Server component of the OpenText ECM offering, two servers running Microsoft Exchange Server, and 10 email journal servers. Additionally, 10 journal test client servers were used to generate more than 150 million emails, simulating the load of a large production email environment.

Details of the servers are listed in Table 1.

Table 1. Servers used in benchmark testing

Server	Number used	Make/model (CPU speed)	Number of cores per server	RAM	Additional notes
Journal test clients	10	HP ProLiant DL380 G5 (2.66 GHz)	4	16 GB	20 GB free disk space
Servers running Exchange Server	2	HP ProLiant DL580 G7	32	64 GB	Eight 256 GB data stores
Journal bridge servers	10	HP ProLiant DL380 G6 (2.13 GHz)	8	48 GB	80 journal services
Server running Content Server	1	IBM System x3850	80	128 GB	Intel Hyper-Threading (HT) Technology enabled 1.8 TB database
Server running Archive Server	1	HP ProLiant DL580 G7	32	32 GB	10 TB email 200 GB database
Database server running SQL Server	1	NEC Express5800/A1080a	80	768 GB	2 TB database 15 TB storage Intel HT Technology enabled
Search/index server	1	HP ProLiant DL980 G7 (2.6 GHz)	64	1024 GB	1 TB index

Figure 1 shows the benchmark testing configuration.

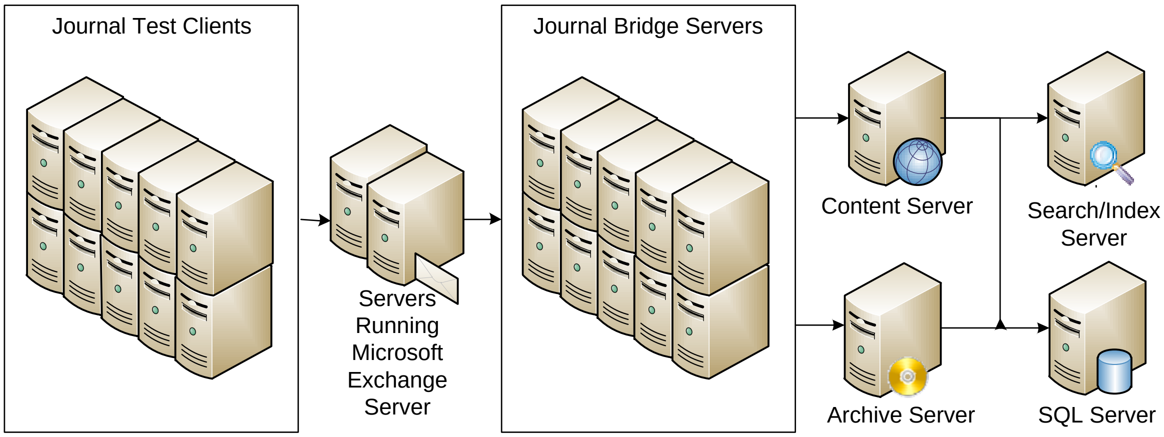

Figure 1. Benchmark testing configuration

Table 2 shows details of the software used. Note that all required software updates were applied.

Table 2. Software versions used in benchmark testing

Server	Software
Journal test clients	• OpenText test client • Microsoft Outlook 2012
Servers running Exchange Server	• Exchange Server 2012
Journal bridge servers	• OpenText Email Journaling Bridge for Microsoft Exchange 10.2.0 • Microsoft Outlook 2010
Server hosting Content Server and search/index server	• OpenText Content Server 10.0.0 with modules: ○ Classifications 10.2.0 ○ Content Archiving 10.0.1 ○ Email Management 10.0.0 ○ Records Management 10.2.0 ○ Security Clearance 10.1.0 ○ Search Engine 10.0
Server hosting Archive Server	• OpenText Archive Server 10.1.1 • OpenText Runtime and Core Services 10.2.1 • OpenText Archive Services 10.2.1 • OpenText Administration Client 10.2.1 • Apache Tomcat 6.0.33 with Java SE 6
Database server running SQL Server	• SQL Server 2012 Release Candidate 0 (RC0)

Email Monitoring Process

In the email monitoring process used for the benchmarking tests, the journal test client servers ran a custom service that checked the number of messages in a journal mailbox on Exchange Server and generated additional journal emails if the number fell below a configurable threshold. This let the journal test clients simulate a production email environment while only generating as much email as can be ingested. Each journal test client server ran eight test client services (total of 80 test client services).

The two servers running Exchange Server were configured with four mailbox stores each (total of eight mailbox stores). The 80 journal mailboxes were spread across the mailbox stores (10 mailboxes per store).

The journal bridge servers read email from their journal mailboxes and then sent the body of the emails to Archive Server and the meta-data to Content Server. Each journal bridge server was configured with eight journal services, each reading mail from a separate journal mailbox. Each journal service ran eight worker threads (total of 640 journal service worker threads). These worker threads took a block of emails from the inbox and moved them into a sub-folder for processing.

For each email processed, a journal worker thread first made a request to Content Server to see if another copy of the email had been stored previously, then stored the body of the message in Archive Server and stored the meta-data in Content Server.

Each of the journal mailboxes for a given test and bridge server was stored in a separate mailbox store, so that the journal test clients and the journal bridges distributed load across the Exchange Server mailbox stores and Content Server. This configuration ensures that the failure of a single journal test client did not dramatically affect the load that could be delivered for the test.

Server Configurations

This section describes in more detail some of the server configurations used in the benchmark testing.

Content Server Configuration

Content Server is typically a CPU-intensive and memory-intensive application, and the number of threads it needs is typically the main focus of tuning. To alleviate memory bottlenecks and provide the ability to support larger numbers of CPUs, a Non-Uniform Memory Architecture (NUMA) was used, with multiple connected memory controllers and CPUs and with memory spread across them.

The server used to host the Content Server front end included eight of the Intel Xeon processor E7-8870 processors. Each of these was a 10-core processor with a 30 MB cache and an integrated memory controller, resulting in an 8-node NUMA configuration with a total of 80 cores (160 logical processors with Intel Hyper-Theading [HT] Technology enabled).

For the benchmark tests, it was found that the best performance was obtained by running eight instances of Content Server 2010: one for each NUMA node (each node is a set of CPUs and memory connected to a memory controller and cache). Each Content Server instance was configured so that it would be scheduled on a separate NUMA node.

Archive Server Configuration

For large-scale deployments of Archive Server such as the benchmark testing, OpenText recommends using the concurrent mark-sweep collector[9] and increasing the initial and maximum memory pools from their default values.

Following are the Java Virtual Machine (JVM) configuration settings used it the benchmarking tests:

```
In the Java Options box, set:
-XX:MaxPermSize=256m
-XX:+UseConcMarkSweepGC
-XX:+CMSClassUnloadingEnabled
-XX:+CMSIncrementalPacing
-Dorg.apache.commons.logging.Log=com.opentext.ecm.
components.logging.wrappers.JCLBootLogger
-Drcs.console.loglevel=WARN

In the Initial memory pool box, specify 256 MB.
In the Maximum memory pool box, specify 1024 MB.
```

In the benchmarking tests, there were 10 journal servers, each running eight journal services. Each journal service was configured with eight worker threads (total of 640 journal service worker threads). The journal worker threads alternated between requests to Content Server and requests to Archive Server, so not all of the threads had active Archive Server requests at any given time.

Records Management 10.2 uses distributed agent infrastructure of Content Server to perform disposition actions. Each of the eight Content Server front-end instances was configured with 64 distributed agent worker threads (total of 512 distributed agent worker threads). There could therefore be up to 512 connections from Content Server to Archive Server when performing disposition actions.

The data flow on the search and index server was configured with 120 extractor threads. Each of these could connect to Archive Server to retrieve emails for indexing at any given time.

To accommodate the number of clients, the number of connector threads in the Apache Tomcat server hosting Archive Server was increased from the default (200) to 1,024 in the Tomcat server.xml file.

---

[9] A detailed description of the operation of this garbage collector can be found in "Memory Management in the Java HotSpot Virtual Machine" at https://java.sun.com/j2se/reference/whitepapers/memorymanagement_whitepaper.pdf

Finally, to accommodate the large number of Tomcat connector threads, the default number of database connections (20 connections) was increased to the maximum (120 connections).

Search and Index Server Configuration

The search and index server uses the following processes: extraction, extractor merges, document conversion, document conversion merges, update distributor merge, and update distribution.

To accommodate the expected ingestion rate, it was anticipated that approximately 120 extractors would be needed. After extraction, 15 merge processes each read input from eight extractors and fed the input into one of 15 document conversion processes. Finally, the output of the document conversion processes was merged before it was sent to the update distributor.

The merge processes between the extractors and the document conversion processes had no trouble keeping up with the volume of messages. However, the default configuration of the merge processes between the document conversion processes and the update distributor was tuned to handle the large volume of messages generated by 120 extractors.

Content Server on the search and index server was configured for 128 threads to handle the 120 extractors, in addition to administrative requests.

Database Server Configuration

This section describes the SQL Server 2012 configuration used in the benchmark testing.

- *Maximum memory configuration was set to 480 GB*.
  SQL Server 2012 can dynamically manage memory without administrator intervention. While in most cases this is adequate, servers that host large databases with heavy user traffic might need some limits on memory usage to prevent performance degradation. If the size of the database (actual data space used) is significantly greater than the amount of RAM on the database server, conflicting demands on the memory can develop over time. SQL Server 2012 can continue growing its in-memory data cache to avoid reading from disk, but the operating system might push back, trying to allocate memory for itself and other processes. The result can be excessive memory paging and a negative effect on performance. For this reason, the maximum memory configuration was set to 480 GB. For more information, see http://msdn.microsoft.com/en-us/library/ms178067.aspx.

- *"Max degree of parallelism" was set to 8*.
  The SQL Server 2012 configuration option "max degree of parallelism" controls the number of processors used for the execution of a query with a parallel plan. This option determines the computing and thread resources used for the query plan operators that can perform the work in parallel. There are several factors that affect the "max degree of parallelism" setting, including the number of processors used, if Intel HT Technology is enabled or not, and whether the server has NUMA configured. For the benchmark testing, the "max degree of parallelism" was set to 8. For more information, see http://support.microsoft.com/kb/2023536/en-us.

- *TempDB was created with one data file per core*.
  The configuration of the SQL Server tempDB database is critical to the performance of SQL Server. In the benchmark testing, tempDB was created with one data file per server core (for a total of 80 files), with each file set to the same size. This enabled the proportional fill algorithm to distribute the allocation load uniformly with minimal contention. For more information, see http://support.microsoft.com/kb/2154845/en-us.

- *"Locked pages in memory" privilege was granted to the SQL Server service account so that large memory pages could be used*.
  Applications based on Windows can use Address Windowing Extensions (Windows AWE) APIs to allocate and to map physical memory into the process address space. Memory that is allocated by using this method is never paged out by the operating system and is locked down until the application explicitly frees it or exits. The use of Windows AWE APIs for memory management is also frequently referred to as "locked pages." The "locked pages in memory" user right for the startup account on the instance of SQL Server in Windows Server was granted so that SQL Server could use large memory pages. For more information, see http://support.microsoft.com/kb/2659143/en-us.

- *SE_MANAGE_VOLUME_NAME right was granted to the SQL Server service account to permit instant file initialization*.
  Data and log files are initialized to overwrite any existing data left on the disk from previously deleted files. Data and log files are first initialized by filling the files with zeros when, for example, a database is created or when files, logs, or data files are added to an existing database. In SQL Server 2012, data files can be initialized instantaneously. This allows for fast execution of these file operations. Instant file initialization reclaims used disk space without filling that space with zeros; instead, disk content is overwritten as new data is written to the files. In the benchmark testing, the SQL Server service account was granted SE_MANAGE_VOLUME_NAME (members of the Windows Administrator group have this right and can grant it to other users by adding them to the Perform Volume Maintenance Tasks security policy). This allowed for instant file initialization when creating or expanding data files. For more information, see http://technet.microsoft.com/en-us/library/ms175935(v=sql.105).aspx.

- *RSS Queues were set to a maximum of 8, and the RSS base CPU was set to 16.*
  Receive-Side Scaling (RSS) improves the ability of Windows Server to handle heavy network traffic. To maximize SQL Server throughout in the benchmark testing, RSS Queues (MaxNumRssCpus) were set to the maximum of 8, and the RSS base CPU was set to 16. For more information, see http://sqlcat.com/sqlcat/b/msdnmirror/archive/2011/09/26/maximizing-sql-server-throughput-with-rss-tuning.aspx.

Test Data Set

The data set used was the Enron Email Data Set, the largest and richest set of publically available, general-purpose corporate email to date. This data set was publicly released as part of Federal Energy Regulatory Commission (FERC) Western Energy Markets investigation, was converted to industry-standard formats by Electronic Discovery Reference Model (EDRM), and consists of 1,227,255 emails with 493,384 attachments covering 151 custodians.[10]

Benchmark Testing Results

Results obtained from the benchmarking tests include observations from the initial period of building up the data set, from isolated tests of Records Management 10.2 on the distributed agent framework, and from testing concurrent ingestion and disposition of emails, representing a real-world usage scenario.

Initial Data Set Construction

Figure 2 shows the peak ingestion rate obtained during the period used to build up the data set for running disposition tests. Note that during the test period, the journal bridge servers were configured so as not to ingest email between 1:00 and 5:00 A.M. to correspond to the Exchange Server maintenance window. Additionally, the journal services and Content Server instances were stopped between 6:00 and 7:00 A.M. to collect logs.

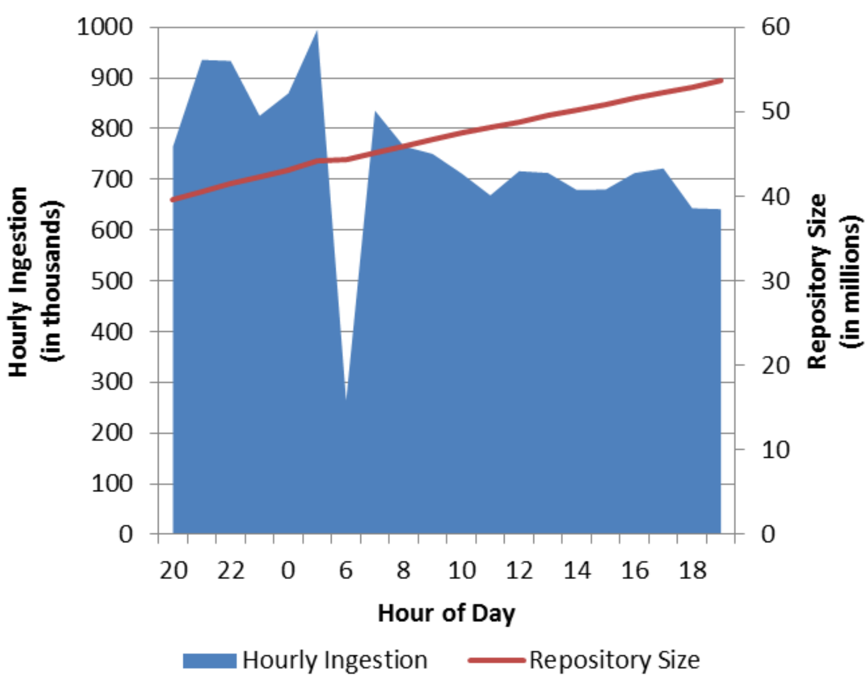

Figure 2. Peak ingestion rate (ingestion only)

[10] http://aws.amazon.com/datasets/917205

These results show that OpenText Email Monitoring is able to ingest very high volumes of email and to sustain high ingestion rates over a 24-hour period. During the 24-hour period, the software ingested a total 14.8 million emails (roughly 10 to 15 times what existing customers typically ingest) and was able to sustain ingestion of more than 600,000 emails for each full hour of ingestion.

Records Management and Distributed Agents

The disposition processing in Records Management 10.2 takes advantage of the distributed agent infrastructure available in Content Server 2010. The distributed agent framework spreads the work of a set of independent tasks among multiple worker-agent threads. For many agent processes, the distributed agent infrastructure allows a significant increase in throughput compared to a single-threaded agent process.

To determine if the distributed agent infrastructure is able to handle the high volume requirements of Records Management dispositions, a test task was created to generate distributed agent tasks with the action “none.” These tasks perform all the usual steps of a disposition action but do not “action” the item.

Table 3 shows the results of this test on a set of 581,470 objects with 16 distributed agent worker threads and performing no action.

Table 3. Disposition performance (with action: none)

Metric	Result
Worker threads	16
Action	None
Result set size	581,470
Disposition search time	0:00:40
Disposition action time	0:18:03
Extrapolated search throughput	872,205 emails/minute
Extrapolated action throughput	46,388,742 emails/day

These results show that Records Management 10.2 is able to return more than 580,000 results from a disposition search in 40 seconds. They also show that Records Management 10.2 is able to perform all the task and Records Management overhead for the actions in 18 minutes.

These metrics represent a search rate of approximately 872,000 results per minute and an action rate of more than 46 million emails per day—confirming that the Content Server distributed agent infrastructure running on SQL Server 2012 is able to keep up with even the most demanding Records Management environments.

Concurrent Ingestion and Disposition

Concurrent ingestion and disposition represents the real-world scenario of an operational OpenText Email Monitoring system using Records Management to dispose of email that does not need to be retained. Table 4 shows the result of performing a disposition search and action on approximately 8.65 million emails concurrent with email ingestion.

Table 4. Concurrent disposition performance (with action: destroy)

Metric	Result
Worker threads	512
Action	Destroy
Result set size	8,650,429
Disposition search time	0:10:38
Disposition action time	1:00:54:40
Extrapolated search throughput	813,520 emails/minute
Extrapolated action throughput	8,344,044 emails/day

These results show that Records Management 10.2 is able to return more than 8.6 million results from a disposition search in less than 11 minutes and is able to destroy those items in slightly less than 25 hours. These results represent a search rate of more than 813,000 email results per minute and an action rate of 8.3 million emails per day.

Figure 3 shows ingestion rates achieved while performing the disposition actions.

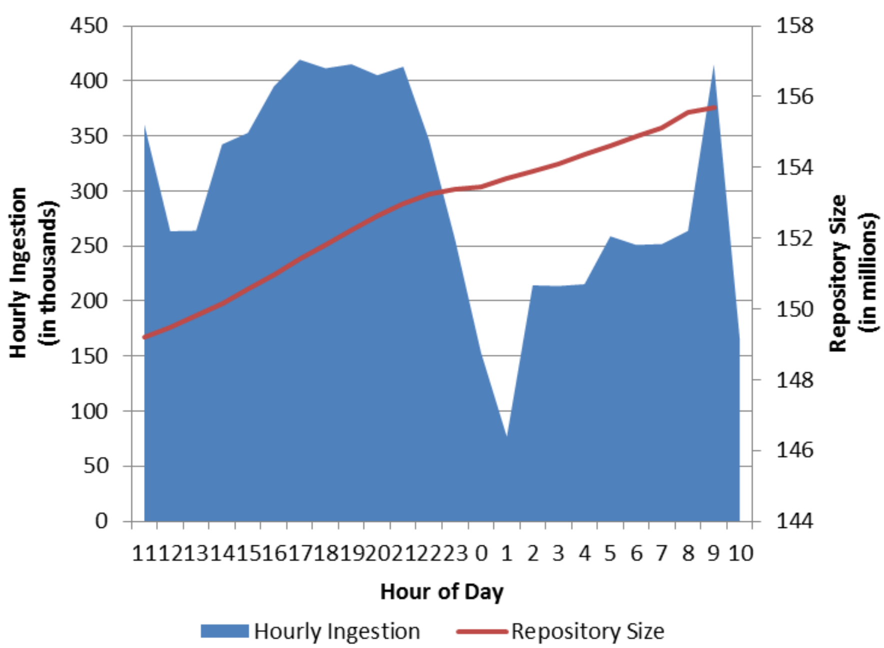

Figure 3. Peak ingestion rate (concurrent ingestion)

These results show that OpenText Email Monitoring 10.2 is able to maintain very high ingestion rates, even while processing disposition actions. Overall, Records Management was able to destroy 8.65 million emails in the same period that the bridge servers ingested another 7.2 million emails—confirming that Records Management 10.2 is able to keep up with even the busiest email monitoring environments.

Resource Usage

Resource-usage results were obtained by running the system performance report from Windows Performance Monitor on each of the servers. Overall, the resource usage on the servers shows that hardware was not a significant bottleneck during the benchmarking tests; the results of the tests therefore demonstrate the maximum throughput of the software.

Table 5 shows the resource overview from Content Server during the concurrent test.

Table 5. Resource use for server hosting OpenText Content Server

Component	Status	Utilization	Details
CPU	Normal	23 percent	Normal CPU load
Network	Normal	16 percent	Busiest network adapter is between 15 percent and 60 percent
Disk	Idle	86 per second	Disk input/output (I/O) is less than 100 read/write per second on disk 0
Memory	Normal	32 percent	89,712 MB available

Table 6 shows the resource overview from Archive Server host during the concurrent test.

Table 6. Resource use for server hosting OpenText Archive Server

Component	Status	Utilization	Details
CPU	Idle	6 percent	Low CPU load
Network	Idle	1 percent	Busiest network adapter is less than 15 percent
Disk	Normal	409 per second	Disk I/O is between 100 and 500 read/write per second on disk 2
Memory	Normal	30 percent	44,643 MB available

Table 7 shows the resource overview from the server hosting Exchange Server during the concurrent test.

Table 7. Resource use for server hosting Microsoft Exchange Server

Component	Status	Utilization	Details
CPU	Idle	13 percent	Low CPU load
Network	Idle	1 percent	Busiest network adapter is less than 15 percent
Disk	Normal	119 per second	Disk I/O is between 100 and 500 read/write per second on disk 7
Memory	Normal	54 percent	60,845 MB available

These results show that the hardware used to host the Content Server front-end instances, the Archive Server instance, and Exchange Server proved more than sufficient to execute on the tasks required.

Table 8 shows the resource overview from the database server hosting SQL Server during the concurrent test.

Table 8. Resource use for server hosting Microsoft SQL Server

Component	Status	Utilization	Details
CPU	Idle	14 percent	Low CPU load
Network	Normal	16 percent	Busiest network adapter is between 15 percent and 60 percent
Disk	Busy	1,699 per second	Disk I/O is more than 500 read/write per second on disk 1
Memory	Busy	94 percent	33,594 MB available

While these results show that both memory use and disk I/O were high on the SQL Server host during the tests, the higher disk I/O was observed during index rebuilds, so disk I/O is not believed to have had an impact on the tests.

Summary

The benchmark testing results were impressive, with a peak ingestion of 995,000 email messages in a single hour, 14.8 million messages in a 24-hour period, or 171 messages per second. These throughputs show that the OpenText ECM offering, when supported by the SQL Server 2012 data-management software, can support a massive number of concurrent transactions.

Table 9 shows highlights of the benchmark testing results.

Table 9. Highlights of benchmark testing results

Metric	Description
Peak email ingestion	• 171 messages/second • 995,000 messages/hour • 14.8 million messages in 24 hours
Concurrent email dispositions and ingestion	• 82 messages/second sustained ingestion • 96 messages/second sustained disposition • 7.1 million messages ingested and 8.3 million messages destroyed in 24 hours

Figure 4 shows the sustained throughput for a 24-hour period in messages per second.

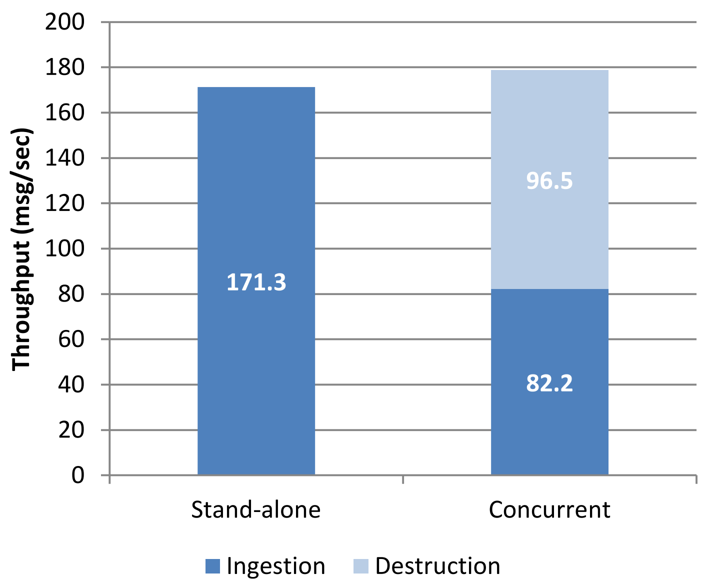

Figure 4. Sustained throughput for 24 hours (in messages/second)

A typical message-ingestion volume for large organizations is between 1 and 1.5 million messages per day; the benchmark testing results are up to 15 times higher.

In addition, a typical email-disposition rate for large organizations is approximately 170,000 per 24-hour period; the test results from this study are 48 times higher.

Conclusion

OpenText and Microsoft ran “real-world” benchmarking tests to evaluate the performance and scalability of the Email Monitoring and Records Management components of the OpenText ECM offering running on SQL Server 2012. The tests measured the peak ingestion of email messages, in addition to the throughput when ingestion and dispositions of email messages were running concurrently.

The results showed that the OpenText solutions, when supported by the SQL Server 2012 database platform, can support extremely high numbers of concurrent transactions, ingesting and dispositioning at rates that satisfy the most demanding email environments.

The benchmark testing confirms that the Email Monitoring and Records Management components of the OpenText ECM offering running on SQL Server 2012 can provide performance and scalability to meet the needs of even the largest enterprise customers. OpenText customers can select SQL Server 2012 for their database platform with confidence, and can be assured that their email requirements can be easily met with OpenText Email Monitoring and Records Management.

Additional Information

The following references provide more information about OpenText and Microsoft.

About OpenText

OpenText, an enterprise software company and leader in enterprise content management, helps organizations manage and gain the true value of their business content. OpenText brings two decades of expertise supporting millions of users in 114 countries. Working with customers and partners, OpenText brings together leading content experts to help organizations capture and preserve corporate memory, increase brand equity, automate processes, mitigate risk, manage compliance, and improve competitiveness.

For more information about OpenText products and services, visit www.opentext.com.

About Microsoft

Founded in 1975, Microsoft (Nasdaq “MSFT”) is the worldwide leader in software, services, and solutions that help people and businesses realize their full potential.

For more information about Microsoft products and services, visit www.microsoft.com.