Metalogix

metalogix.com

WHITE PAPER

Metalogix Replicator for SharePoint

ENTERPRISE PERFORMANCE AND SCALABILITY

Version 4.1

June 2012

CONTENTS

INTRODUCTION

Metalogix Replicator is an enterprise-class solution for replicating content, metadata, structure, security, look & feel and business processes from one SharePoint web application to anther web application. Replicator version 4 supports SharePoint Foundation 2010, SharePoint Server 2010, Office SharePoint Server 2007 and Windows SharePoint Services 3.0.

Metalogix Replicator supports the replication needs of Microsoft's largest (and smallest) SharePoint customers. Version 4 provides additional performance and scalability capabilities these customers will need as they continue to grow their Distributed SharePoint environments.

The release of Metalogix Replicator version 4 represents a new performance and scalability milestone for SharePoint replication solutions. Version 4 includes new features for supporting massively scalable SharePoint replication networks:

- Shared Replication Packages,
- Package Database Caching,
- Improved Initial Replication Support, and
- Improved Remote Differential Compression (RDC).

Together, these features build on the performance features that have been built into Replicator over the previous 3 major releases and 7 years of development effort. The Replicator version 4 performance feature set now includes the following components:

1. Multiple Replication Engine Deployment
2. Multi-threaded Replication Queue Architecture
3. Customizable Replication Interval
4. Configurable Package Event Count
5. Configurable Package Event Processing Duration
6. Replication Package Compression
7. Hardware Network Compression Support
8. Software Package Compression
9. Remote Differential Compression (RDC)
10. Independent Controls for Inbound Processing and Outbound Event Processing
11. Selective Structure Replication

12. Rule-based Item-level Content Replication
13. Selectable Replication Events
14. Shared Replication Packages
15. Package Database Caching
16. Configurable Replication Monitor Update Level

This document provides additional detailed information on the Replicator performance feature set. The first section describes each component of the Replicator performance feature set. The second section includes in-depth discussions of each feature as well as new benchmark testing performed at the Microsoft Technology Center in New York, NY and the Metalogix Network Test Lab. The final section of the document analyses Replicator from an operational point-of-view.

BENCHMARKING AT THE MICROSOFT TECHNOLOGY CENTER

In August 2010, with the support of the Microsoft Technology Center in New York, NY, the Metalogix Massive Scalability Team executed several performance tests to exercise a pre-release version of Metalogix Replicator version 4. The Scalability Team configured the largest known Distributed SharePoint 2010 environment in the world for these tests:

- 61 SharePoint farms connected into a single replication network
- Microsoft Hyper-V server virtualization technology
- 65 SharePoint 2010 Windows 2008 virtual servers
- 16 Windows 2008 R2 host physical servers
- 128 GB of physical RAM
- 1.2 TB of physical disk storage

A pre-release version of Metalogix Connect for Replicator was used to visualize and report on this configuration. The fully deployed 61 SharePoint farm environment is depicted in Figure 1 .

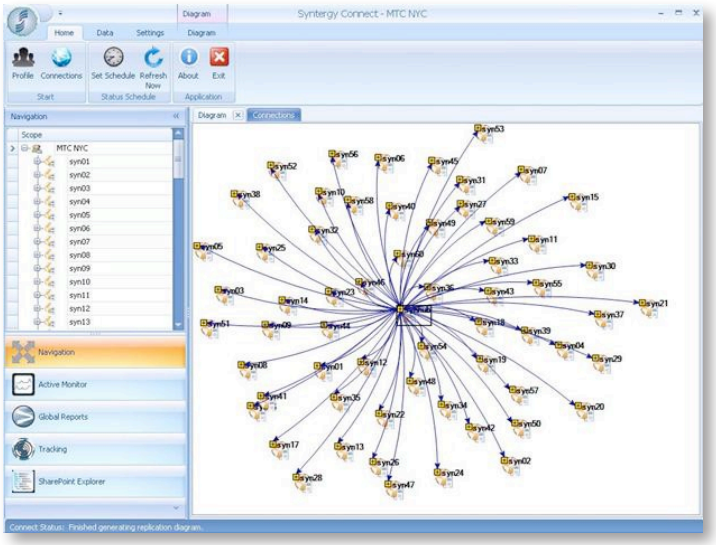

Figure 1. Microsoft Technology Center, New York Benchmark: Metalogix Connect for Replicator Visualization

REPLICATOR PERFORMANCE AND SCALABILITY FEATURES

The Replicator version 4 performance feature set builds upon many components that have been built into Replicator over the previous 3 major releases and 7 years of development effort. The new or significantly improved version 4 features are marked with an asterisk.

Feature	Description
Replicator Multi-Engine Deployment	Replicator Enterprise Edition enables the Replicator Engine to be deployed on multiple web front-ends in each farm. With multiple web front-ends hosting Replicator Engine in a farm, Replicator provides a higher availability and higher performance replication solution; in addition to supporting increased scalability.
Multi-Threaded Replication Queue Architecture	Replicator's Replication Engine architecture is based on a multi-threaded replication queue architecture that supports parallel Replication Package processing.
Customizable Replication Interval	Replicator supports several replication scheduling policies: Immediately, Interval, Daily, Weekly, Monthly and Manual.
Configurable Package Event Count	By default, Replicator includes the ability to group or batch multiple Replication Events into a single Replication Package for processing and transfer to the Target Web Application. This Configurable Package Event Count determines the maximum number of Events that can be packaged into a single Replication Package. This enables Replicator to be tuned for different levels of real-time replication, wide area network performance characteristics and available Inbound and Outbound Event Processing memory and CPU processing resources.
Configurable Package Event Processing Duration	Similar to the Configurable Package Event Count feature, the Configurable Package Event Processing Duration allows control over how many Replication Events are grouped or batched together into a single Replication Package based on elapsed Outbound Event Processing time. This setting enables Replicator local server resource requirements to be tuned for optimal performance.
Replication Package Compression	Replicator supports several forms of software compression in addition to support for and compatibility with variety of hardware network compression devices. Replicator can use either software or hardware compression to reduce the amount of Replication Package data transferred over a wide area network.
Software Package Compression	Replicator supports ZIP software package compression and a custom implementation of Microsoft's Remote Differential Compression (RDC) that is highly optimized for the one-way and bi-directional replication of SharePoint data.
Hardware Network Compression Support	When hardware network compression devices, bandwidth optimization appliances, or network accelerator solutions are available (such as the Riverbed® Steelhead® Appliance), Replicator can be configured to minimize the server resources used for software compression and maximize the effective of the network compression device or application.
Remote Differential Compression (RDC) *	Metalogix Replicator supports a custom implementation of Microsoft's Remote Differential Compression (RDC) technology that Metalogix has optimized to reduce the total amount of Replication Package data that needs to be transferred between the Source Web Application to the Target Web Application during replication.
Independent Outbound Event and Inbound Event Processing Controls	Replicator supports independent controls for Outbound Event Processing and Inbound Event Processing at the Web Application and Replication Connection levels for a Source Web Application and Target Web Application. In addition, replication for a particular Web Application can be configured as one-way or bi-directional.

Feature	Description
Selective Structure Replication	By default, many customers use web application-to-web application replication – replicating the entire content of each site collection in one web application to the second web application. Selective structure replication enables the SharePoint Administrator to select a specific subset of the Source Web Application structure to be replicated to the Target Web Application.
Rule-based Item-level Content Replication	Rule-based item-level content replication uses the Metalogix Replicator Rules Engine to process custom rule sets to determine if an item in a particular list or document library should be replicated.
Selectable Replication Events	Replicator supports selectable replication of changes that occur in a SharePoint Web Application, Site Collection, Web Site, List or Document Library. The different types of individual changes are called Replication Events. Replication Events are categorized into higher-level Event Groups. By selecting which Events need to be replicated and which Events don't need to be replicated, the Selectable Replication Events feature provides the SharePoint Administrator with fine-grained control over the Events processing during Inbound Event Processing, Package Transfer and Outbound Event Processing; which in turn help improve overall Replicator performance.
Shared Replication Packages *	In Replicator version 3, a separate Replication Package was created for each outbound Replication Connection. The Shared Replication Package feature in Replicator version 4 eliminates the time required to create a separate Package for each outbound Replication Connection – a single shared Package containing the batch of Replication Events is created. Depending on the number of Events in the package, the size and type of the SharePoint change, and the number of outbound Replication Connections, this can significantly reduce Outbound Event Processing time and resources.
Package Database Caching *	In Replicator version 4, upon acceptance of the transfer of an Inbound Replication Package, Replicator immediately caches the metadata for each Replication Event in the Replicator Configuration database. This improves the performance of the Replicator Engine by eliminating the need to repeatedly access the serialized Package data; especially in SharePoint farms that host multiple Replicator Engines.
Configurable Replication Monitor Update Level *	During normal Inbound Event Processing, Replicator, running on a Target Web Application, will make several calls to the Replicator Web Service running on the Source Web Application to keep the Monitor Replication reports on the Source Web Application as current as possible. To help minimize the Queued Item update network traffic between the Source and Target Web Applications, Replicator version 4.1 supports a new Replication Connection property called "quiet mode".
Improved Initial Replication Support *	The new Backup Mode feature provides a more efficient method of replicating a large site collection or hierarchy of web sites. Using Backup Mode, Replicator uses the SharePoint import and export operations to create a single archive of the entire group of web sites, and queues the replication of this archive as a single Replication Event.

* New or significantly improved Replicator version 4 feature

REPLICATOR MULTI-ENGINE DEPLOYMENT

Replicator Enterprise Edition enables the Replicator Engine to be deployed on multiple web front-ends in each farm. With multiple web front-ends hosting Replicator Engine in a farm, Replicator provides a higher availability and higher performance replication solution; in addition to supporting increased scalability. If one engine is stopped, the other engines are still operational.

The following is based on a version 4 benchmark performed at the Microsoft Technology Center in New York using a hub-and-spoke topology connecting 61 SharePoint 2010 farms. The figure shows:

- Multiple replication engines processing and transferring multiple packages simultaneously
- Multiple threads per engine supporting parallel Replication Package processing and transfer
- Smooth scaling of the replication processing under heavy load

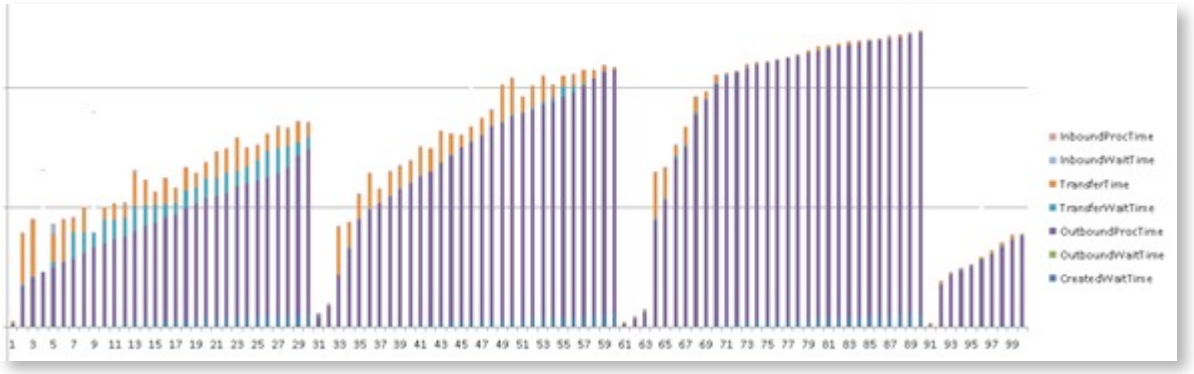

Figure 2. Multiple Engine, Multiple Thread Performance and Scalability

MULTI-THREADED REPLICATION QUEUE ARCHITECTURE

Replicator's Replication Engine architecture is based on a multi-threaded replication queue architecture that supports parallel Replication Package processing. Replicator Standard Edition supports 4 threads per replication engine with a single active engine running on a single web front-end per SharePoint farm. Replicator Enterprise Edition supports 10 to an unlimited number of threads running on a single replication engine running on an unlimited number of web front-ends per SharePoint farm.

Figure 2. Multiple Engine, Multiple Thread Performance and Scalability shows Replicator Enterprise Edition running on 4 web front-ends configured to perform parallel Replication Package processing and transfer.

CUSTOMIZED REPLICATION INTERVAL

Replicator supports several different replication scheduling policies:

- Immediately (default real-time, event-driven replication policy)
- Interval (seconds, minutes, hours)
- Daily
- Weekly
- Monthly
- Manual

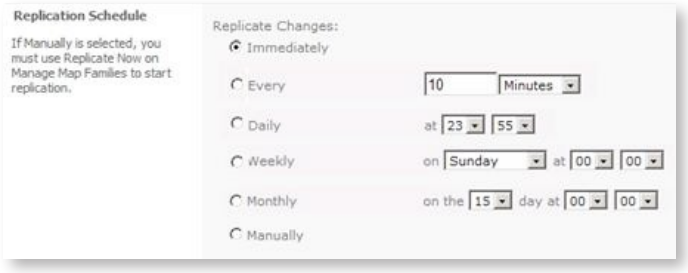

Figure 3. Map Family Replication Schedule

These settings enable fine grained control over when Outbound Event Processing and Package Transfer will occur. This in turns allows the SharePoint Administrator to better manage SharePoint server and wide area network resources.

CONFIGURABLE PACKAGE EVENT COUNT

By default, Replicator includes the ability to group or batch multiple Replication Events into a single Replication Package for processing and transfer to the Target Web Application. This Configurable Package Event Count determines the maximum number of Events that can be packaged into a single Replication Package. This enables Replicator to be tuned for different levels of real-time replication, wide area network performance characteristics and available Inbound and Outbound Event Processing memory and CPU processing resources.

A higher Package Event Count will result in fewer, larger Packages being created which can increase overall performance. The Windows Background Intelligent Transfer Process (BITS) protocol used by Replicator for Package Transfer is designed for high performance and reliability (restartable) with large file downloads as well as smaller downloads. Larger packages, somewhat surprisingly, can result in better throughput and performance in slow and unreliable wide area network scenarios.

CONFIGURABLE PACKAGE EVENT PROCESSING DURATION

Similar to the Configurable Package Event Count feature, the Configurable Package Event Processing Duration allows control over how many Replication Events are grouped or batched together into a single Replication Package based on elapsed Outbound Event Processing time. This setting enables Replicator local server resource requirements to be tuned for optimal performance.

REPLICATION PACKAGE COMPRESSION

Replicator supports several forms of software compression in addition to support for and compatibility with variety of hardware network compression devices. Replicator can use either software or hardware compression to reduce the amount of Replication Package data transferred over a wide area network.

The following sections describe the benefits of Replicator's support for hardware network compression and software package compression. Replicator's highly optimized implementation of Microsoft's Remote Differential Compression (RDC) is also described.

HARDWARE NETWORK COMPRESSION SUPPORT

When hardware network compression devices, bandwidth optimization appliances, or network accelerator solutions are available (such as the Riverbed® Steelhead® Appliance), Replicator can be configured to minimize the server resources used for software compression and maximize the effectiveness of the network compression device or application. A sample deployment is illustrated in Figure 4. Metalogix Replicator and Riverbed® Steelhead® Appliance Compressed Replication Package Solution.

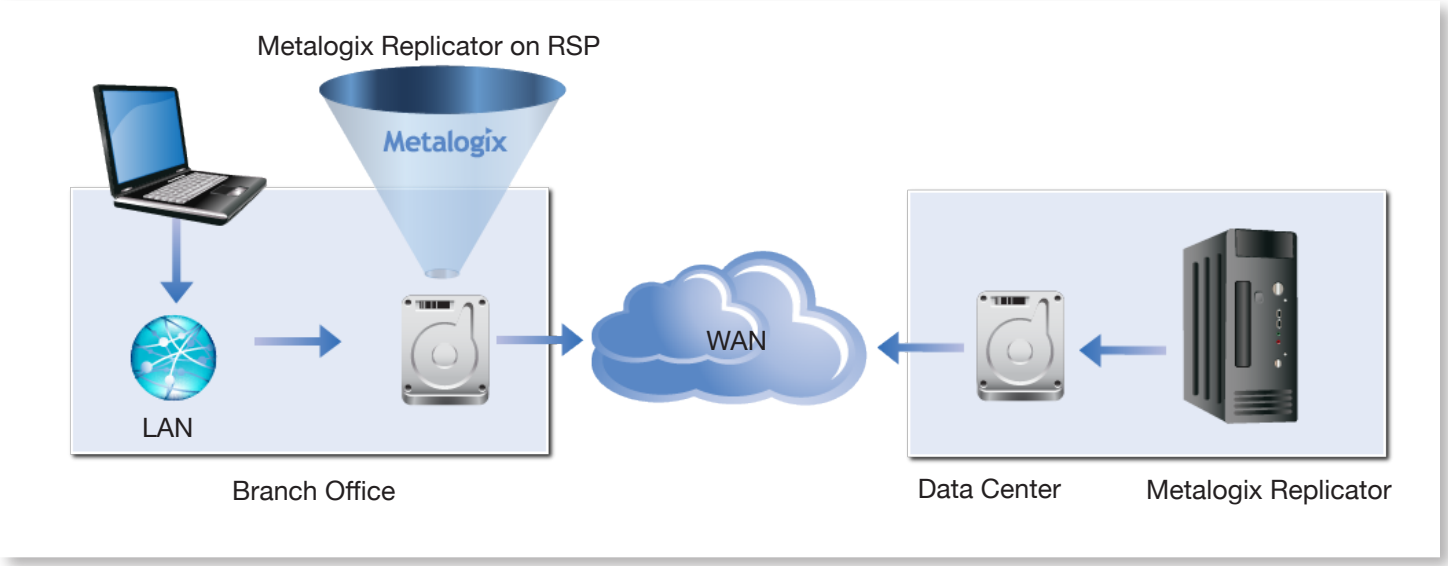

Figure 4. Metalogix Replicator and Riverbed® Steelhead® Appliance Compressed Replication Package Solution

Configuring Replicator to support hardware network compression devices avoids the redundant and potentially expensive overhead of compressing packages in software when hardware compression is available.

SOFTWARE PACKAGE COMPRESSION

Replicator supports ZIP software package compression and a custom implementation of Microsoft's Remote Differential Compression (RDC) that is highly optimized for the one-way and bi-directional replication of SharePoint data.

REMOTE DIFFERENTIAL COMPRESSION (RDC)

Metalogix Replicator supports a custom implementation of Microsoft's Remote Differential Compression (RDC) technology that Metalogix has optimized to reduce the total amount of Replication Package data that needs to be transferred between the Source Web Application to the Target Web Application during replication.

"Remote Differential Compression (RDC) allows data to be synchronized with a remote source using compression techniques to minimize the amount of data sent across the network. RDC is suitable for applications that move data across a wide area network (WAN) where the data transmission costs outweigh the CPU cost of signature computation. RDC can also be used on faster networks if the amount of data to be transferred is relatively large and the changes to the data are typically small." 1 More specific details on RDC can be found in Appendix A – About Windows Remote Differential Compression.

Metalogix's Dynamic Differential Cache improvements to Microsoft RDC use server-side pre-processing and post-processing to significantly reduce the amount of SharePoint data that is transferred over the wide area network.

During performance testing in the Metalogix Network Test Lab, using real-life and simulated test data (SharePoint tasks, multi-level document libraries, Office 2003 documents, Office 2007 documents, Office 2010 documents as well as other desktop file formats such as AutoCAD drawings, photos and images), Metalogix Replicator RDC reduced the total network traffic by more than 90% compared to ZIP compression.

INDEPENDENT OUTBOUND EVENT AND INBOUND EVENT PROCESSING CONTROLS

Replicator supports independent controls for Outbound Event Processing and Inbound Event Processing at the Web Application and Replication Connection levels for a Source Web Application and Target Web Application. In addition, replication for a particular Web Application can be configured as one-way or bi- directional.

When only one-way replication is required, performance is optimized when Event processing is disabled on the appropriate Web Application. In the bi-directional replication scenario, separate Replication Connections enable separate replication policies to be used for the outbound and inbound connections.

SELECTIVE STRUCTURE REPLICATION

By default, many customers use web application-to-web application replication – replicating the entire content of each site collection in one web application to the second web application. Selective structure replication enables the SharePoint Administrator to select a specific subset of the Source Web Application structure to be replicated to the Target Web Application.

For example:

- One or more site collections
- Root web site
- Any individual child web site
- Any child web site and subhierarchy of the child web site's child web sites
- Individual lists and/or document libraries in a web site.

Selective structure replication can be used to significantly reduce the volume of replication data by selecting the specific site collections and web sites to be replicated.

RULE-BASED ITEM-LEVEL CONTENT REPLICATION

Rule-based item-level content replication uses the Metalogix Replicator Rules Engine to optionally process custom rule sets to determine if an item in a particular list or document library should be replicated.

Although rules-based content replication is normally considered a security and content promotion feature, rule sets can also be used to block the replication of larger files or other files based on each item's metadata values and improve performance.

[1] Source: http://msdn.microsoft.com/en-us/library/aa372948(v=VS.85).aspx)

SELECTABLE REPLICATION EVENTS

Replicator supports selectable replication of changes that occur in a SharePoint Web Application, Site Collection, Web Site, List or Document Library. The different types of individual changes are called Replication Events. Replication Events are categorized into higher-level Event Groups. By selecting which Events need to be replicated and which Events don't need to be replicated, the Selectable Replication Events feature provides the SharePoint Administrator with fine-grained control over the Events processing during Inbound Event Processing, Package Transfer and Outbound Event Processing; which in turn help improve overall Replicator performance.

Figure 5. Metalogix Replicator Replication Events highlights the Replication Events section of the Configure Map Family web page from Replicator Central Administration. Entire Event Groups or individual Events can be enabled or disabled. This figure shows the default settings.

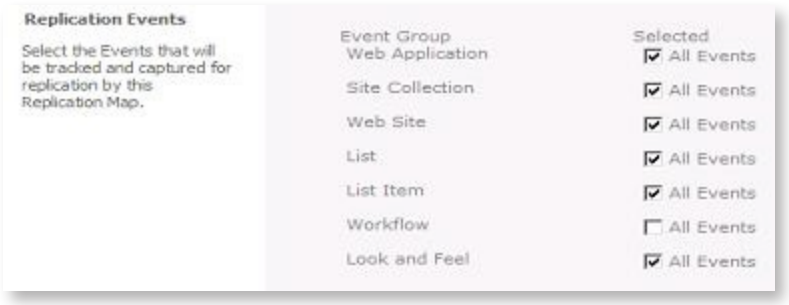

Figure 5. Metalogix Replicator Replication Events

Replication version 4.1 also includes new Web Site Replication Events for replicating SharePoint 2010's social networking features: ratings, comments and keyword tags.

SHARED REPLICATION PACKAGES

In Replicator version 3, a separate Replication Package was created for each outbound Replication Connection. The Shared Replication Package feature in Replicator version 4 eliminates the time required to create a separate Package for each outbound Replication Connection – a single shared Package containing the batch of Replication Events is created. Depending on the number of Events in the package, the size and type of the SharePoint change, and the number of outbound Replication Connections, this can significantly reduce Outbound Event Processing time and resources.

Figure 6. Microsoft Technology Center Benchmark: Shared Replication Package Outbound Processing Time illustrates how the elapsed Outbound Processing time for the first Package (green portion of first bar in the chart) was significantly reduced for the remaining 59 instances of the Package replicated to Target Web Application 2 through 60.

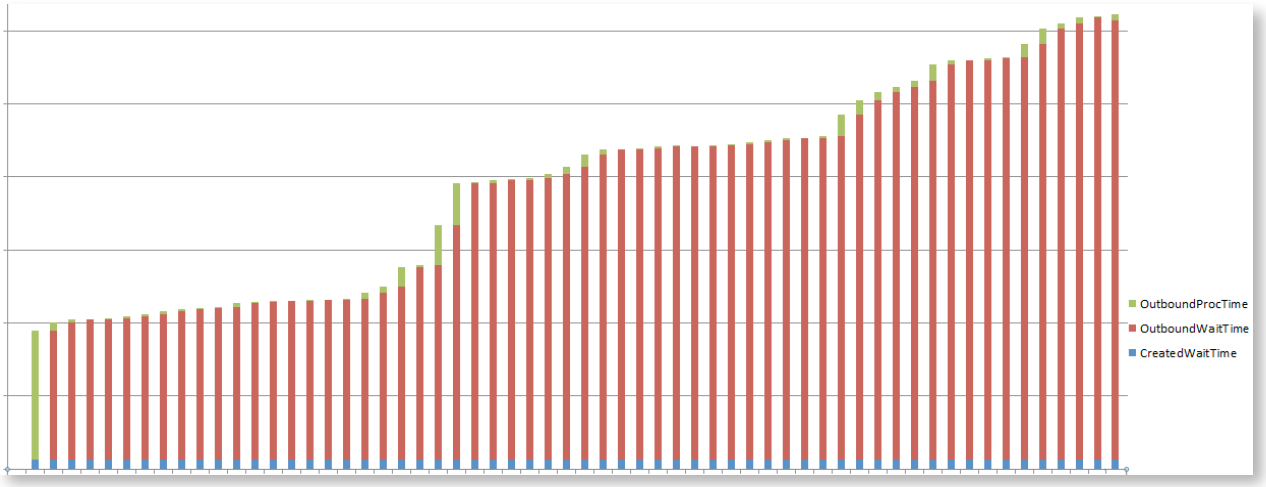

Figure 6. Microsoft Technology Center Benchmark: Shared Replication Package Outbound Processing Time

PACKAGE DATABASE CACHING

In Replicator version 4, upon acceptance of the transfer of an Inbound Replication Package, Replicator immediately caches the metadata for each Replication Event in the Replicator Configuration database. This improves the performance of the Replicator Engine by eliminating the need to repeatedly access the serialized Package data; especially in SharePoint farms that host multiple Replicator Engines.

Package Database Caching specifically applies to Inbound Event Processing on the Target Web Application Server and improves overall Inbound Event Processing by reducing the number of file system accesses, reducing the volume of data read from the file system and lowered file decompression CPU usage.

CONFIGURABLE REPLICATION MONITOR UPDATE LEVEL

During normal Inbound Event Processing, Replicator, running on a Target Web Application, will make several calls to the Replicator Web Service running on the Source Web Application to keep the Monitor Replication reports on the Source Web Application as current as possible.

To help minimize the Queued Item update network traffic between the Source and Target Web Applications, Replicator version 4.1 supports a new Replication Connection property called “quiet mode”. It can be configured through the Configure Replication Connection administration page where it is referred to as the Configurable Replication Monitor Update Level. This new setting has 2 values: Normal and Minimal (quite mode).

During performance testing in the Metalogix Network Test Lab replicating 1000 1KB files, the network traffic was reduced by 55% and the number of HTTP PUT operations was reduced by 60%.

IMPROVED INITIAL REPLICATION SUPPORT

The new Backup Mode feature provides a more efficient method of replicating a large site collection or hierarchy of web sites. Using Backup Mode, Replicator uses the SharePoint import and export operations to create a single archive of the entire group of web sites, and queues the replication of this archive as a single Replication Event.

Backup Mode significantly improves the performance and reliability of a large initial replication: instead of creating and transferring 10,000s of queued items and packages in a very large SharePoint environment, only a few (large) Packages need to be packaged and transferred.

In addition, if RDC compression is enabled, the amount of data transferred over a wide area network will be reduced dramatically. When replicating to multiple locations, only one Package is created for all destinations with the version 4 Shared Replication Package feature.

MICROSOFT TECHNOLOGY CENTER PERFORMANCE AND SCALABILITY BENCHMARK

In August 2010, Metalogix was invited to the Microsoft Technology Center (MTC) in New York, NY to measure the performance and scalability of an early release of Metalogix Replicator for SharePoint version 4.0.

Working closely with the MTC technical staff, the Metalogix Massive Scalability Team deployed and configured the world's largest SharePoint 2010 Distributed SharePoint environment, measured in terms of the number of independent SharePoint farms. Eleven large scale physical servers from Dell and HP running Microsoft Windows Server 2008 R2 were used to deploy and configure 85 Windows Server 2008

R2 virtual machines. Microsoft Windows Hyper-V was used for the operating system virtualization solution. The entire configuration was deployed and managed using Microsoft Windows Virtual Machine Manager.

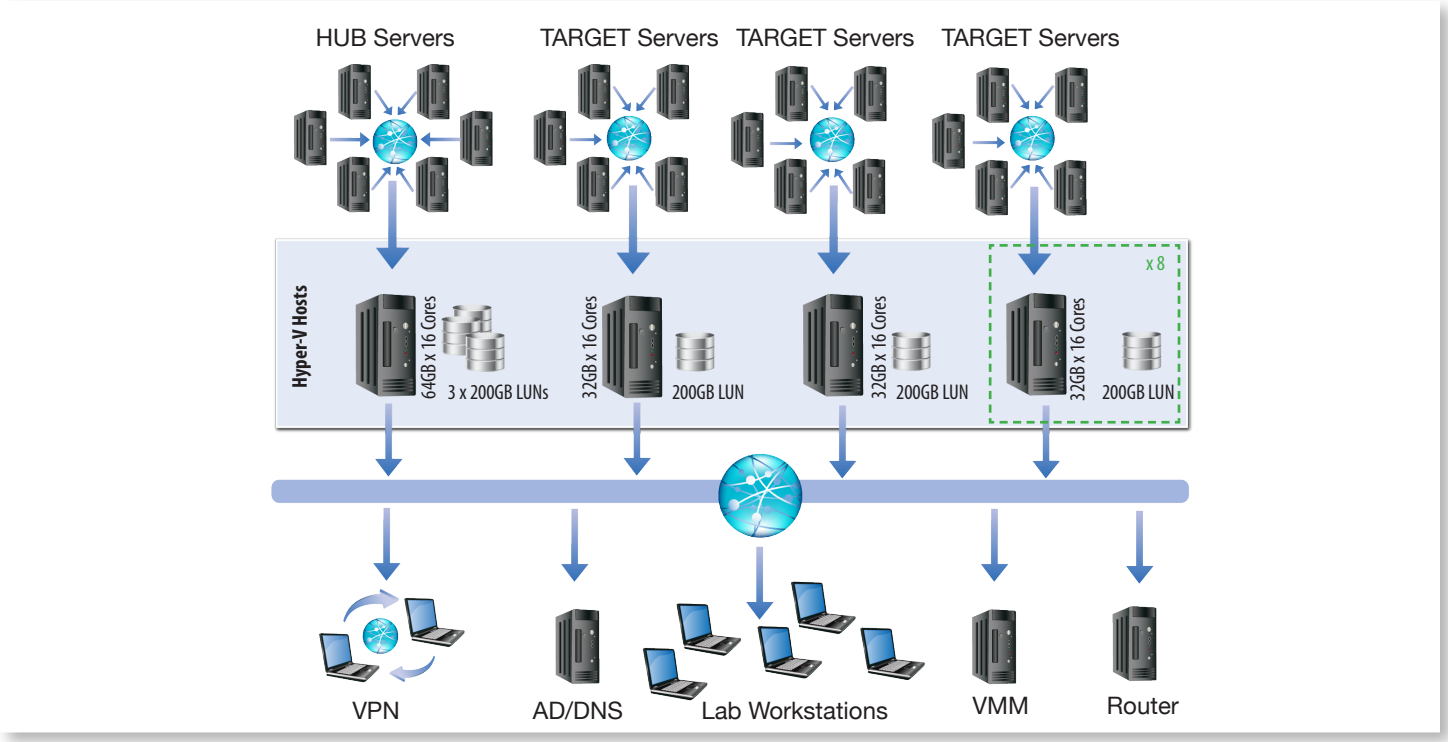

Figure 7. Microsoft Technology Center Windows Server 2008 R2 Hyper-V Host Environment: Metalogix Replicator Benchmark

Five virtual machines running on 16 processor cores with access to 64GB of RAM and 600 GB of disk storage were used to host 4 Web Front-End Servers and 1 SQL Server 2008 database server in the Hub SharePoint 2010 farm. Eighty (80) additional virtual machines running on 96 processor cores with access to 320 GM of RAM and 2 TB of disk storage were used to host 60 single-server SharePoint 2010 farms. The specific configuration of the Windows Server 2008 R2 physical hosts can be found in Table 1. Microsoft Technology Center Windows Server 2008 R2 Hyper-V Host Environment: Virtual Machine Configuration.

Host Type	No. Hosts	Cores	RAM (GB)	No. LUN	LUN Size (GB)	Total Cores	Total RAM	Total Storage (GB)	VM RAM	No. VM
Hub	1	16	64	3	200	16	64	600	-	5
Targets	1	16	32	1	200	16	32	200	4	8
Targets	1	16	32	1	200	16	32	200	4	8
Targets	8	8	32	8	200	64	256	1600	4	64
Totals	11					112	384	2600	12	85

Table 1. Microsoft Technology Center Windows Server 2008 R2 Hyper-V Host Environment: Virtual Machine Configuration

The following performance monitoring tools were used during the benchmark:

- Windows Performance Monitor
- Metalogix Replicator Replication Monitor
- Metalogix Replicator Replication Status
- Metalogix Connect for Replicator
- Custom Replication Data Collection PowerShell Scripts
- Custom Replicator Deployment, Configuration, and Management PowerShell Scripts
- Custom SharePoint Benchmarking PowerShell Scripts

METALOGIX NETWORK TEST LAB PERFORMANCE BENCHMARKING

During the design and development of Metalogix Replicator version 4, each new performance build was tested and measured along 40 dimensions, including:

- CPU usage
- Available memory
- Working set
- Disk access
- Network access
- Network utilization
- HTTP Web Server performance

The configuration of the Metalogix Network Test Lab is depicted in Figure 8. Metalogix Network Test Lab: Performance Benchmarking. In contrast with virtualized Microsoft Technology Center environment used for the massive scalability and performance benchmarking, the Metalogix Network Test Lab runs entirely on physical hardware.

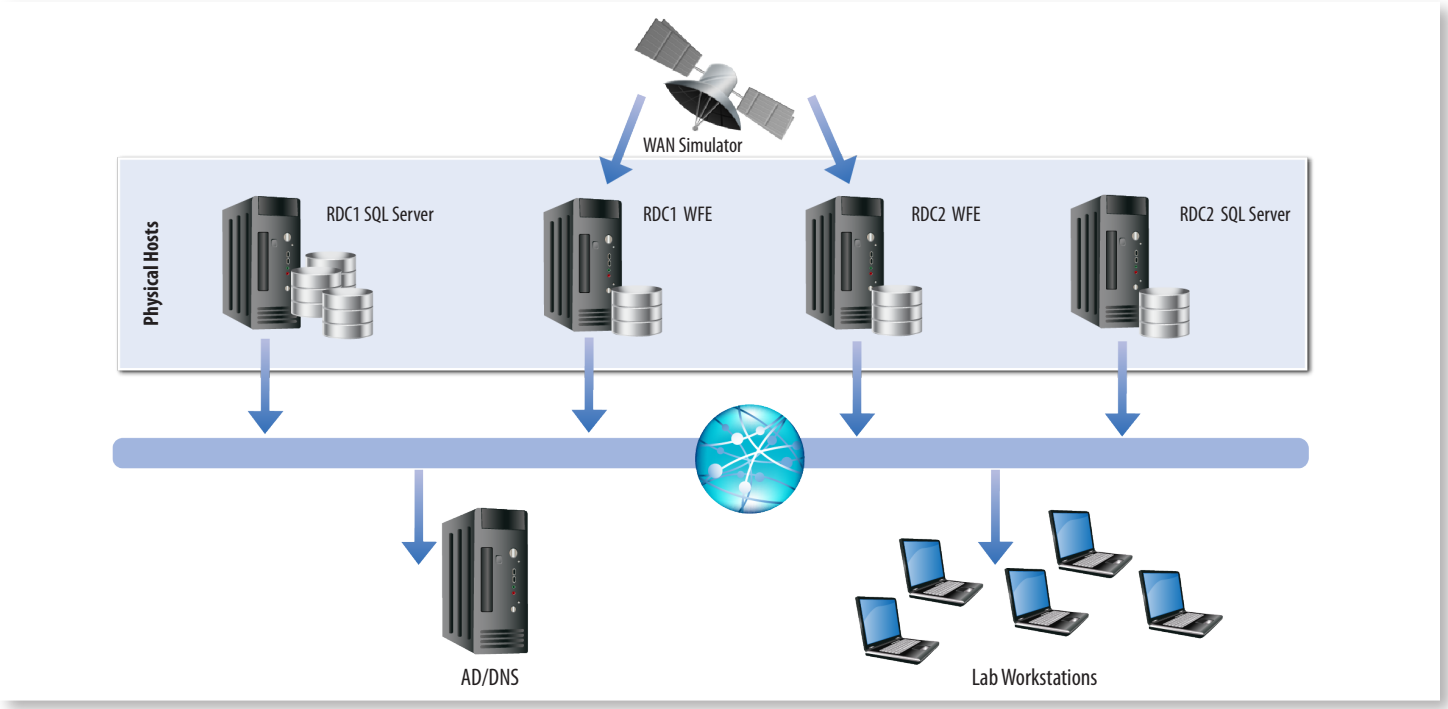

Figure 8. Metalogix Network Test Lab: Performance Benchmarking

UNDERSTANDING REPLICATOR PERFORMANCE AND SCALABILITY

The key to understanding Replicator performance and scalability from a day-to-day operational perspective involves understanding how the basic components of the Replicator Pipeline capture and package Replication Events, transfer packages from the Source Web Application to the Target Web Application and finally update the Target Web Application with the new or changed content.

METALOGIX REPLICATOR PIPELINE

The Replicator Pipeline is a detailed representation of how Replicator divides the overall replication process into 3 activities:

- Captures a change on a Source Web Application, packages a group of Replication Events and associated SharePoint content into a Queued Item and corresponding Replication Package,
- Transfers the Queued Item and Package from the Source to the Target Web Application, and
- When received and accepted on the Target Web Application, applies the changes represented by the Queued Item and Package to the Target Web Application.

These 3 activities, Outbound Event Processing, Package Transfer and Inbound Event Processing are shown in the following figure.

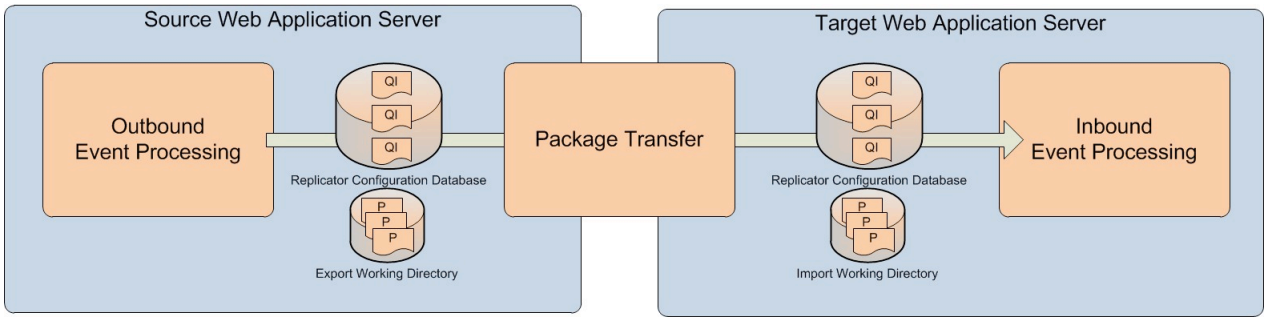

Figure 9. Metalogix Replicator Pipeline

Outbound Event Processing is responsible for capturing and recording Replication Events that occur in the Source Web Application. For example, when a user creates a new contact in a contact list in a SharePoint Web site or edits and saves a Microsoft Word document that is stored in a SharePoint document library, each of these is captured and recorded as a Replication Event. Replication Events are captured and stored in a set of SQL Server tables in a Replicator database. Outbound Event Processing is controlled by Replication Maps which determine what Events need to be captured, packaged and transferred to the Target Web Application. Groups of Replication Events are packaged into two types of messages or objects: Queued Items and Replication Packages.

A Queued Item is a unit of work to be transferred to a Target Web Application for remote execution. The Replicator Web Service on a Target Web Application is called to push a Queued Item from the Source Web Application to a Target Web Application.

A Replication Package is a collection of one or more Replication Events plus data about the SharePoint information that is packaged in a format specific to the Replication Transport being used. When an Event is being processed, Outbound Event Processing calls the SharePoint object model to extract the changed SharePoint information from the Source Web Application content database.

The Package Transfer activity is responsible for the transfer of Queued Items and Replication Packages (Packages) from the Source Web Application to the Target Web Application. Package Transfer is the process that sits between Outbound Event Processing (on the Source Web Application) and Inbound Event Processing (on the Target Web Applications). Package Transfer invokes Replicator Engine software components on both the Source and Target Web Applications.

Inbound Event Processing is responsible for processing the Queued Items and Packages received and accepted by the Target Web Application and applying them to the site collections, Web sites, lists and document libraries in the web application's content database. The Queued Items and Packages are applied to the Target Web Application content base by calling the SharePoint Object Model.

REPLICATOR PERFORMANCE

Replicator performance is best understood by analyzing the Replicator Pipeline activities depicted in Figure 9. Metalogix Replicator Pipeline. The same variables that affect performance of a SharePoint environment also affect Replicator performance. Replicator will perform optimally if the underlying SharePoint environment has been configured for optimal performance for a given user workload. In addition, there are Replicator-specific configuration settings and optimization that can further improve Replicator performance.

The performance profile of SharePoint, and each of the Replicator Pipeline activities, is based on a variety of factors including: user workload, document file formats, document size distributions, SharePoint server and farm configuration, database server configuration, local area network (LAN) characteristics, wide area network (WAN) characteristics, Replication data compression and replication monitor settings as well as other Replicator configuration settings.

User workload is defined by two factors: user content change rate and user content change size distribution. Qualitatively, the user content change rate is simply the rate at which users of a particular SharePoint environment make content changes: creating new list items, editing and saving existing

items, creating or uploading new documents, editing and saving existing documents and, less frequently,

creating new site collections, web sites, lists and document libraries or making metadata or security changes.

User content change size distribution refers to the size of the content changes and their distribution. For example, in a document collaboration solution, the average size of a Word document might be a few megabytes and the average size of a PowerPoint might be measured in 10's of megabytes. The amount of data needed to represent a new SharePoint contact or task item would be measured in 10s or 100s of kilobytes. The distribution of sizes vary from small 1K byte documents to media files measured in 10s of megabytes.

OUTBOUND EVENT PROCESSING

In Replicator version 4, each Map Family contains a Replication Schedule. A Replication Schedule controls how frequently Replication Events for a particular Replication Connection are processed and packaged into Queued Items and Replication Packages. Event Processing and Packaging can be scheduled to run:

- Immediately (real-time replication)
- After a timed interval (after a specified number of seconds, minutes, hours or days)
- Scheduled daily, weekly, monthly
- Manually initiated

With the exception of immediate replication, Outbound Event Processing performance is primarily determined by the Replication Schedule, user content size distribution and SharePoint system performance. For immediate replication, Outbound Event Processing performance is also affected by the user content change rate. From a system performance perspective, all Outbound Event Processing occurs locally in the SharePoint farm that is hosting a particular Source Web Application. On the local SharePoint farm, server RAM, disk organization, LAN network configuration and database server configuration are the greatest determinants of SharePoint and Replicator system performance.

When Replicator is configured for immediate replication, Outbound Event Processing (Event capture and processing) is performed in near real time.

PACKAGE TRANSFER

Package Transfer refers to the part of the Replicator Pipeline that is responsible for transferring Queued Items (and their corresponding Packages) from a Source Web Application to a Target Web Application. Package Transfer performance is determined by a number of different factors:

- User workload
- WAN network performance
- Source Web Application system performance
- Target Web Application system performance
- Replicator data compression

A Queued Item and corresponding Replication Package represents a batch of changes that users have made to the Source Web Application content. Queued Items are relatively small in size and are passed from the Source Web Application to the Target Web Application by calling the Target Web Application Replicator Web Service. Packages are either downloaded or uploaded to the Target Web Application using the BITS protocol running over HTTP/HTTPS. The majority of the Package Transfer time is consumed by the transfer of the Package. This is because the Package contains that changed SharePoint content that needs to be replicated from the Source Web Application to the Target Web Application and the changed content, in relative terms, may be quite large depending on whether the changed content is simple list item or a 10MB PowerPoint document.

Package Transfer performance be improved using Package compression settings that can be set using Replicator Central Administration on a Replication Connection by Connection basis. The following Package compression settings are supported in Replicator:

- None
- ZIP (default)
- RDC (based on Microsoft Remote Differential Compression technology)

Once the characteristics of the Queued Item and Package stream and the Package compression settings are understood, the next major determinant of Package Transfer performance are the characteristics of the WAN:

- Available bandwidth
- Round-trip latency
- Packet loss
- Reliability

For example, satellite links typically have lower bandwidth, high latency and lower reliability compared to terrestrial links; and exceptionally poorer performance compared to a fiber ring in a metropolitan area network (MAN) or LAN. Metalogix's advanced use of the Microsoft BITS protocol in the design of the Replicator Package Transfer process allows Replicator to function extremely well over high performance connections as well as over slow or unreliable satellite, ship-to-shore and battlefield networks.

If Replicator is installed and configured on multiple SharePoint farms connected by a high-performance network, the Package Transfer process will perform on a near real time basis between any pair of Source and Target Web Applications.

INBOUND EVENT PROCESSING

Inbound Event Processing begins once a Queued Item and its corresponding Package have been transferred to the Target Web Application by the Package Transfer process. Inbound Event Processing executes entirely within the SharePoint farm for the Target Web Application.

Inbound Event Processing performance is determined by the rate at which the Replicator Service can process the local Queued Items and their corresponding Packages, decompress the Packages and apply the content changes to the Target Web Application by calling the SharePoint Object Model. The Inbound Event Processing performance is affected by the following factors:

- User workload
- SharePoint server and farm configuration
- Database server configuration,
- Local area network (LAN) characteristics

Subject to the above constraints, Replicator Inbound Event Processing is able to operate at near real time performance.

PERFORMANCE MONITORING

Replicator Central Administration provides detailed replication monitoring, replication status and replication event processing history reports for monitoring the Replicator Pipeline. The following two figures are examples of the statistics available from the Replication Status report for a sample intranet Source Web Application that is being replicated to an extranet Target Web Application.

NOTE: The following figures are from a small-scale test performed on virtual single server SharePoint farm that is hosting both the Source and Target Web Applications. The statistics shown in these figures are for illustration purposes only.

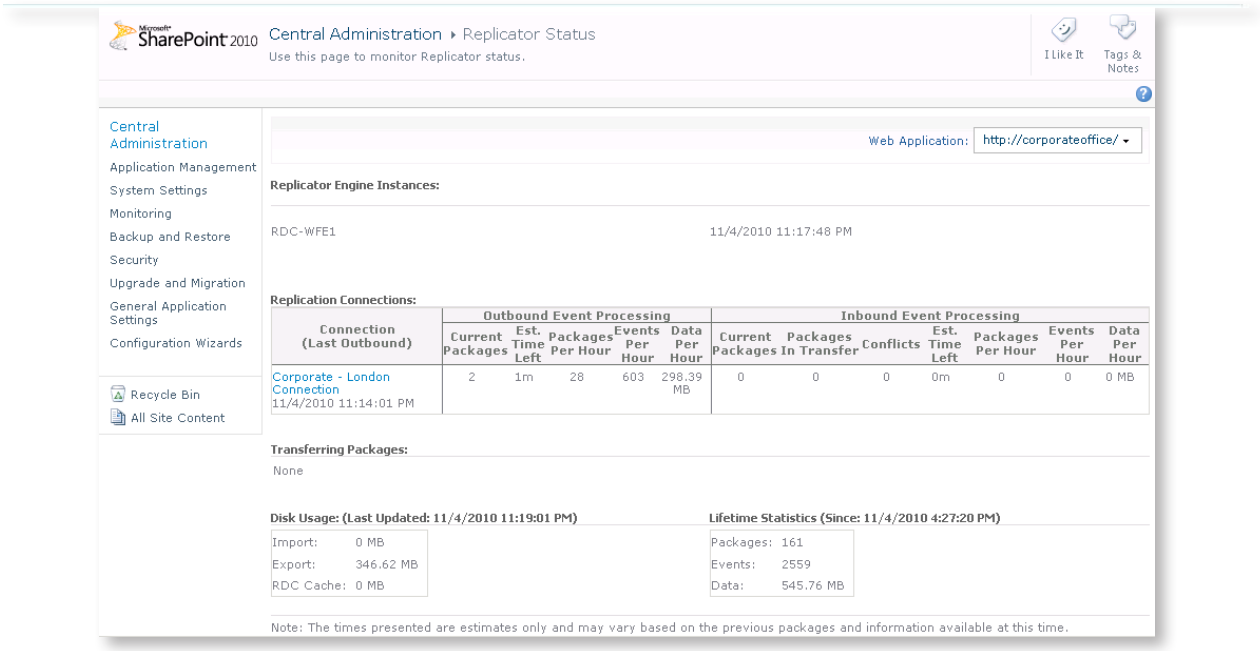

Figure 10. Sample Outbound Event Processing Statistics

In the above figure, the Outbound Event Processing statistics tells us:

- 13 Packages (and Queued Items) were processed representing 13 batches of Replication Events
- Total number of Replication Events = 212
- Total size of the 13 Packages (212 Events) = 13.8MB

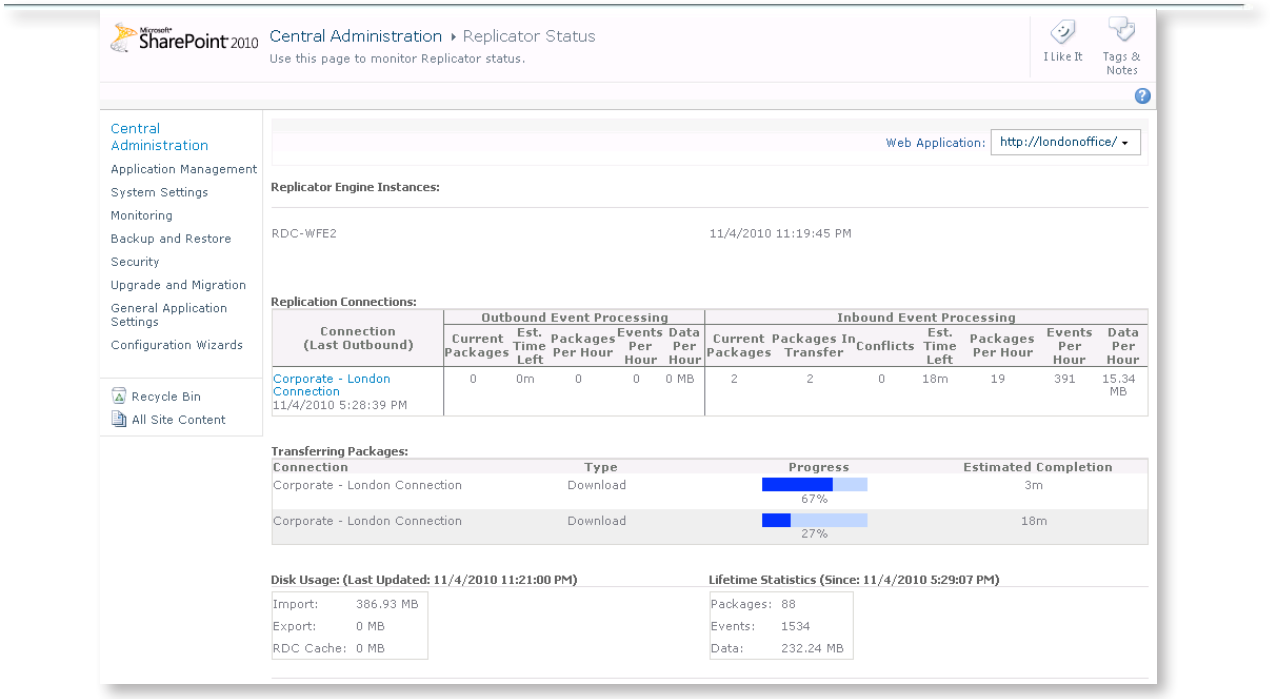

Figure 11. Sample Inbound Event Processing Statistics

In the above figure, the Inbound Event Processing statistics tells us:

- 14 Packages (and Queued Items) were processed representing 14 batches of Replication Events
- Total number of Replication Events = 213
- Total size of the 14 Packages (213 Events) = 11.82MB

NOTE: Inbound and Outbound Event Processing statistics are never expected to match, due to several factors, including system generated Queued Items and Packages, retries, filtered Packages and rejected Packages.

To monitor large Distributed SharePoint environments consisting of several farms, Metalogix Connect for SharePoint or Metalogix Connect for Replicator is the monitoring tool of choice for these environments.

REPLICATOR PIPELINE PERFORMANCE FEATURES

The following table maps the key Metalogix Replicator performance features to each phase of the Replicator Pipeline.

Feature	All Pipeline Activities	Outbound Processing	Package Transfer	Inbound Processing
Replicator Multi-Engine Deployment	YES			
Multi-Threaded Replication Queue Architecture	YES			
Configurable Package Event Count	YES			
Configurable Package Event Processing Duration	YES			
Independent Outbound Event Processing and Inbound Event Processing Controls	YES			
Selective Structure Replication	YES			
Rule-based Item-level Content Replication	YES			
Selectable Replication Events	YES			
Customizable Replication Interval		YES		
Shared Replication Packages		YES		
Replication Package Compression		YES	YES	
Hardware Network Compression Support		YES	YES	
Software Package Compression		YES	YES	
Remote Differential Compression (RDC)		YES	YES	
Package Database Caching				YES

Table 2. Metalogix Replicator Pipeline Performance Features

NOTE: The performance features that affect Outbound Event Processing also tend to enhance the performance of the downstream Replicator Pipeline activities: Package Transfer and Inbound Event Processing. There is only one feature, Package Database Caching, which only affects Inbound Event Processing.

REPLICATOR SCALABILITY

UNLIMITED MULTI-MASTER, POINT-TO-POINT REPLICATION NETWORK MODEL

The Metalogix Replicator architecture is based on an unlimited multi-master, point-to-point replication network model that does not require global sharing of any configuration information. Arbitrary network topologies are supported including:

- Serial
- Parallel
- Hub-and-spoke
- Cascading hub
- Federated hub
- Cross domain
- Mesh, and
- Offline.

METALOGIX CONNECT FOR REPLICATOR

Metalogix Connect is a desktop application that provides visualization, monitoring and management of the wide area SharePoint environment. It can be used to visualize the status of each farm and its associated connections. Connect is available as an add-on feature for the Basic and the Standard Edition (and is included in the Enterprise Edition).

Metalogix Connect for Replicator provides the highly scalable replication network management tools needed to monitor and maintain the largest (and smallest) distributed SharePoint environments.

APPENDIX A – ABOUT WINDOWS REMOTE DIFFERENTIAL COMPRESSION

Source: http://msdn.microsoft.com/en-us/library/aa372948(VS.85).aspx

Remote Differential Compression (RDC) allows data to be synchronized with a remote source using compression techniques to minimize the amount of data sent across the network.

RDC is different from patching-oriented differencing mechanisms, such as Binary Delta Compression (BDC), that are designed to operate only on known versions of a single file. BDC requires the server to have copies of all versions of the file, and differences between each pair of versions are pre-computed so that they can be distributed efficiently from a server to multiple clients.

RDC makes no assumptions about file similarity or versioning. Because differences between files are computed on the fly, RDC is ideally suited for synchronizing files that are different or have been updated independently.

RDC does not assume that the file data to be synchronized resides in physical files. Therefore, the RDC application is responsible for performing file I/O on behalf of the RDC library.

Because it is transport independent, RDC can be used with RPC, HTTP, or other desired transport mechanisms. The RDC application bears the responsibility for choosing the appropriate transport and performing any client or server authentication that is required to support the transport's security model.

RDC is suitable for applications that move data across a wide area network (WAN) where the data transmission costs outweigh the CPU cost of signature computation. RDC can also be used on faster networks if the amount of data to be transferred is relatively large and the changes to the data are typically small.

REMOTE DIFFERENTIAL COMPRESSION ALGORITHM OVERVIEW

RDC divides a file's data into chunks by computing the local maxima of a fingerprinting function that is

computed at every byte position in the file. A fingerprinting function is a hash function that can be computed incrementally. For example, if you compute the function F over a range of bytes from the file, Bi...Bj, it should then be possible to compute F(Bi+1...Bj+1) incrementally by adding the byte Bj+1 and subtracting the byte Bi. The range of bytes from the file, Bi...Bj, is called the hash window. The length of this window, in bytes, is called the hash window size.

The RDC library's FilterMax signature generator "slides" the hash window across the entire file by adding the byte at the leading edge and subtracting the byte at the trailing edge of the window. Meanwhile, the generator continually examines the sequence of fingerprint function values over a given range of bytes, called the horizon size. If a fingerprint function value is a local maximum within the range, its byte position is chosen as a "cut point," or chunk boundary.

After the file has been divided into chunks, the signature generator computes a strong hash value (an MD4 hash), called a signature, for each chunk. The signatures can be used to compare the contents of two arbitrarily different versions of a file.

Because the size of the signature file grows linearly with the size of the original file, comparing very large files can be expensive. This cost is reduced dramatically by applying the RDC algorithm recursively to the signature files. For example, if the original file size is 9 GB, the signature file size would typically be about

81 MB. If the RDC algorithm is applied to the signature file, the resulting second-level signature file size would be about 5.7 MB.

REMOTE DIFFERENTIAL COMPRESSION APPLICATION CONCEPTS

When developing an application that uses RDC, it is important to understand the following concepts and terminology.

In a typical RDC scenario, a server and a client have different versions of a file. (The terms client and server refer only to the computers' roles in this scenario, not their operating systems.) The client's copy of the file is called the seed file. The server's copy is called the source file. The objective of the RDC application is to download the file updates to the client, which uses them to construct a target file that combines the updates from the source file with the unchanged contents from the seed file.

The RDC client and server each use the RDC library's FilterMax signature generator to divide their copy of the file into chunks and compute a strong hash, called a signature, for each chunk of file data. Thus, the client has a list of signatures for the seed file, and the server has a list of signatures for the source file. These signature lists can be computed on the fly, or they can be pre-computed.

The client initiates the RDC protocol by requesting the source signature list from the server. Then the client compares each source signature against the signatures in its own seed signature list. If a source signature matches a seed signature, the client already has the file data for that signature. If a source signature does not appear in the client's list of seed signatures, the client must request the specified chunk (of file data) from the server.

The result of comparing the two signature lists is a needs list, which describes which chunks of file data, from where (seed or source file), are needed to construct the target file on the client computer. Each entry in the needs list is called a needs block.

The client iterates through each needs block and copies the specified chunk of the source or seed file data to the target file. Seed file data is copied locally. Source file data is downloaded from the server. The more similar the seed and source files are, the less network bandwidth is required to create the target file.

ABOUT METALOGIX

Metalogix is the trusted provider of innovative content lifecycle management solutions for Microsoft SharePoint, Exchange and Cloud platforms. We deliver high-performance solutions to scale and cost-effectively manage, move, store, archive and protect enterprise content. Metalogix provides global support to thousands of customers and strategic partners and is a Microsoft Gold Partner, a managed partner in Microsoft's High Potential ISV Group and GSA provider. Metalogix is a privately held company backed by Insight Venture Partners and Bessemer Venture Partners.

METALOGIX

5335 Wisconsin Ave NW, Suite 510, Washington DC 20015

sales@metalogix.com | www.metalogix.com | 1 202.609.9100