Metalogix® Content Matrix

Performance Optimization Guide

Contents

1

Introduction

This document covers a number of different topics to help you with optimizing the performance for migrating your content into SharePoint with Metalogix Content Matrix. These topics include:

- special considerations for migrating to a SharePoint on premises target, which includes:

    - hardware best practices and scaling

    - SharePoint connection adapters

- special considerations for migrating to SharePoint Online

- using PowerShell

- using Distributed Migration in SharePoint Edition to maximize resource utilization for large migration jobs

These migration optimizations are only a general set of guidelines to help you with your migration. For more information on any of the specific features and abilities of Metalogix Content Matrix, please see the appropriate Edition's Help file (located within the client application, or user documentation that can be found online at https://support.quest.com/metalogix-content-matrix/technical-documents).

Requirements

The only requirement that is necessary to use this document is to be in the process of installing, or have already installed, Metalogix Content Matrix Console into your environment, for migrating your content into SharePoint. Since appropriate installation of Metalogix Content Matrix can have a major impact on performance, it is preferable to modify your installation if necessary based on the guidance in this document.

Please refer to the *Metalogix Content Matrix Console Quick Start Guide* or *Metalogix Content Matrix Advanced Installation Guide* for details on installing Metalogix Content Matrix.

Special Considerations for Migrations to a SharePoint On Premises Target

NOTE: For performance considerations specific to migrating to a SharePoint Online target, see Special Considerations for SharePoint Online Targets.

Hardware Best Practices

In addition to the minimum requirements laid out in the *Metalogix Content Matrix Console Quick Start Guide* or *Metalogix Content Matrix Advanced Installation Guide*, migration servers where Metalogix Content Matrix will be installed will perform optimally with the following hardware configurations:

Non-WFE server/agent

- 4 core CPU or higher

- 8gb of RAM or higher

- 20gb of free disk space

WFE server/agent

- 8 core CPU

- 16gb of RAM or higher

- 50gb of free disk space

NOTE: Please avoid installation of Metalogix Content Matrix on a disk that also hosts the SQL server for the target SharePoint installation. This can result in significant resource utilization, and will eliminate many of the benefits of parallelization.

Scaling Hardware

Metalogix Content Matrix can take advantage of scaling hardware within the migration environment.

Scaling Out

Scaling out is the process of adding servers with Metalogix Content Matrix installed to perform migrations concurrently to different Web Front End (WFE) servers within the target on premises SharePoint farm.

Dedicated Migration Servers

Metalogix Content Matrix can be installed on dedicated migration servers to facilitate maximum resource utilization in migrations whenever possible. In this configuration, Metalogix Content Matrix can then be remotely pointed at the target SharePoint farm.

> NOTE: This deployment scenario will require the installation of the Metalogix Extensions Web Service adapter on the target SharePoint server (as defined in the Connection Adapters section of this document). This will allow for remote connections to the SharePoint Object Model (OM).

Load balancing can be effectively used in this case to ensure that migration requests from Metalogix Content Matrix will be properly balanced to WFE servers in the target farm based on utilization.
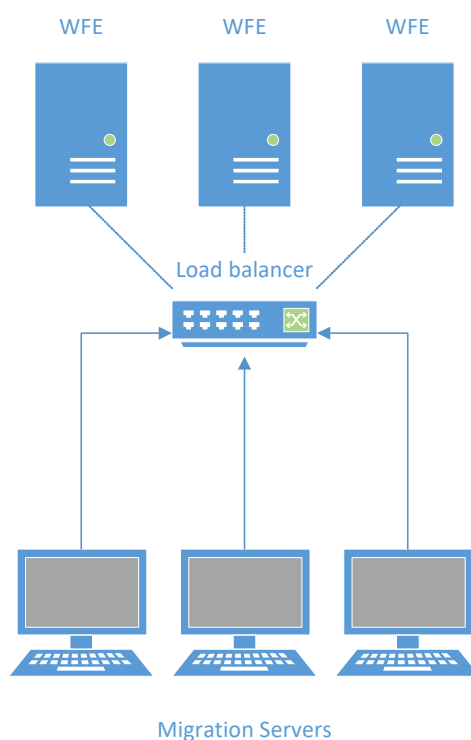
Migration Servers

The number of migration servers that can be utilized should be managed to ensure that the target SharePoint server can effectively handle all of the migration requests. Generally a one to one correlation with WFE servers is recommended.

If load balancing is not being used on the target farm, migration servers can be directly pointed at specific WFE servers to ensure maximum concurrent throughput. See below image:
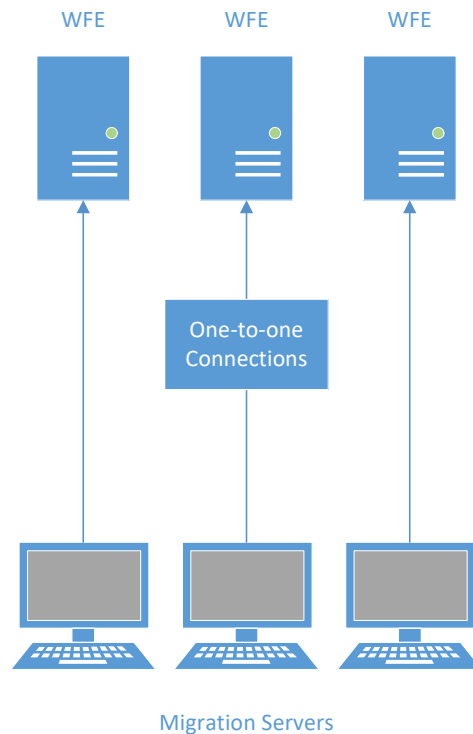
Migration Servers

Installation on WFE Servers

When dedicated migration servers are unavailable, Metalogix Content Matrix can be directly installed on WFE servers. In this configuration, Metalogix Content Matrix can use the local OM adapter for connections.

In this configuration, load balancing will be bypassed due to Metalogix Content Matrix utilizing only the local WFE OM for migrations.

It is recommended that Metalogix Content Matrix be installed on the maximum number of idle WFE servers in order to maximize throughput.

NOTE: If WFE servers are already servicing user requests in a production environment, extreme care should be taken to ensure that the user experience is not impacted. Due to the potential resource utilization of Metalogix Content Matrix on the host server, it is recommended that migrations via the application never be run on a production WFE outside of a maintenance window.

Scaling Up

Scaling up is the process of adding additional hardware to a single migration machine in order to facilitate faster copies.

In general, scaling up is only recommended to the point specified in the hardware best practices section below. It is more efficient to scale out the migration infrastructure and parallelize migration instead of overly increasing resources on a single machine.

Connection Adapters

There are four primary adapter types available in Metalogix Content Matrix for connections to SharePoint on premises environments. They are defined below:

Database (DB)

The DB adapter is a **read-only** adapter designed for source side connections. If possible, always use the database adapter as the source for migrations, as it is the fastest adapter available in the application. It is also particularly useful for WAN based migrations or migrations to Office 365. This is because you can take database backups of your Content DBs, and move the resulting .bak files to a SQL Server on the target LAN (or Azure VMs in the case of O365 migrations), and then do a local migration.

DB connections look like this in the Metalogix Content Matrix UI once connected:
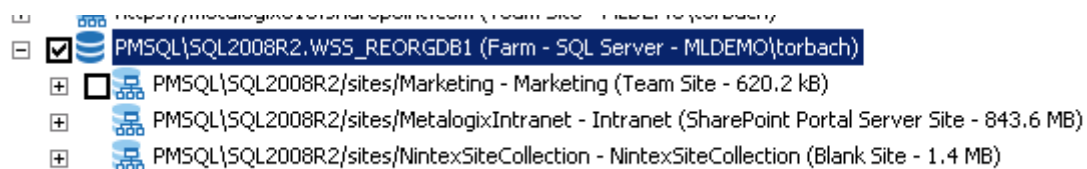
While the Database adapter does have some limitations, such as the inability to migrate Managed Metadata Services from one farm to another, you can use Metalogix Extensions Web Services (MEWS) for that part of the migration, and the DB Adapter for the rest of the migration for best performance when migrating between an on premises source and target.

Local Object Model (OM)

The OM adapter is the second fastest adapter available in the application, and the fastest adapter for writing content. It can only be accessed when Metalogix Content Matrix is directly installed on a WFE server in the SharePoint farm. This adapter is preferred if Metalogix Content Matrix is installed on a WFE server in the target SharePoint farm .

OM farm connections look like this in the Metalogix Content Matrix UI once connected:
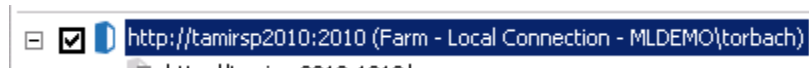
Metalogix Extensions Web Service (MEWS)

The MEWS adapter is a very fast adapter, but requires installation of a solution on the source or target SharePoint farm which it is being used in. The solution is included in the Metalogix Content Matrix installer, and can be installed by running the installer on any WFE in the target SharePoint farm and selecting the "Install Metalogix Content Matrix SharePoint WSPs" button, then checking the box to install the "SharePoint Extensions Web Service." If the target SharePoint farm will be connected to remotely, this adapter is preferred.

MEWS farm connections look like this in the Metalogix Content Matrix UI once connected:
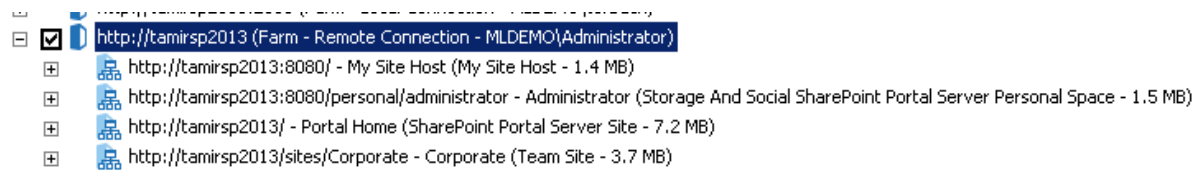
NOTE: The distinguishing factor between an OM connection and a MEWS connection is the "Local Connection" or "Remote Connection" string to the right of the connection.

Special Considerations for SharePoint Online Targets

Migrations from SharePoint on premises (on-prem) to SharePoint Online (referenced as SPO in this document), have a number of significant factors that make them more complex and time consuming than on premises to on premises SharePoint migrations. These factors include:

- Functional and Application Programming Interface (API) differences between SharePoint on-prem and SPO that result in some on-prem elements not being easy or possible to migrate to SPO.

- Performance-based limitations as a result of Microsoft's protection mechanisms to ensure that no single SPO tenant/client can cause an entire multi-tenant farm to be unstable.

- Performance impact of the physical distance between a client's on-prem SharePoint environment(s) and their SPO farm location.

This section explains why these limitations exist, and helps prepare Metalogix Content Matrix clients to plan a migration to SPO. This includes:

- Listing and explaining the functional and API limitations and their significance on migrations.

- Explaining the migration API leveraging Azure Storage to enhance the performance of content migrated to SPO.

- Ensuring that you allocate the appropriate time and consideration required when migrating their on premises farms to SPO.

The information in this section is essential reading for anyone migrating from SharePoint on-prem to SPO.

Functional and API Differences Between SPO and SharePoint that Impact Migrations

When migrating to an SPO target, factors to consider include:

- Different Application Programming Interfaces Available for SPO

AND

- OAuth and Migration Performance.

Different Application Programming Interfaces Available for SPO

For on-prem SharePoint, Microsoft offers a very rich API known as the SharePoint Server Object Model. This has been the most frequently used and richest API to access SharePoint on-prem.

However, due to the multi-tenant nature of SPO, Microsoft has to ensure that no single tenant can compromise other tenants' environments. In light of that, Microsoft cannot expose the full SharePoint Object Model in SPO. They currently expose five APIs that are useful for migrations:

- The Client Side Object Model (CSOM)

- The SharePoint REST API

- SharePoint Online Remote PowerShell (subset of CSOM)

- Office 365 Remote PowerShell (subset of CSOM)

- A migration-focused API that leverages Azure Blob Storage and batching to migrate list items and documents significantly more quickly to SPO (available both from CSOM and REST)

While the CSOM API is becoming close to the SharePoint Server Object Model in terms of feature parity, it is very slow for migrations. In light of that, Microsoft launched a Migration API using the Azure and SPO-Provided Containers, collectively known as the Import Pipeline. Today, the combination of the CSOM API and the Azure/SPO-Provided Containers is used by Metalogix Content Matrix to write content to SPO. SPO and the CSOM API has some limitations that make Metalogix Content Matrix and other migration tools unable to migrate all on-prem SharePoint content. For a full set of SPO limitations, see this MS web page listing. Another important page to look at is SharePoint 2013 deprecated functionality. This applies to both SharePoint on premises and SPO.

OAuth and Migration Performance

Quest recommends using OAuth Authentication when connecting to SharePoint Online for optimal migration performance. Using OAuth should result in reduced throttling of migration workloads.

For more details: https://www.quest.com/community/blogs/b/microsoft-platform-management/posts/impact-of-sharepoint-online-throttling-on-migrations.

Performance of Migrations Using the Import Pipeline

There are three primary factors that until now have impacted performance of migrations to SPO dramatically:

I. The physical distance between your on-prem SharePoint farm and your SPO tenant, and the fact that in most cases, this Wide Area Network (WAN), is across the Internet, making it very unpredictable.

Long distance between on-premises and Office 365

II. The protection mechanisms and throttling that Microsoft has in place to protect their multi-tenant clients from abuse by other tenants on the same farm.

- User and Farm based Throttling
- Virus Scanning
- Load Balancing of Web Server Connections
- Distributed Denial of Service Monitoring

III. The very chatty nature of the CSOM API that is the primary API used for migrations to SharePoint Online, compounded by the distance required for each API call to travel between on-prem environments and SPO.

On-Premises SharePoint

SharePoint Online/ OD4B Final Destination

Copy Site Collection/Site/List

1. Read List details from source
2. Write List to target

Copy/Paste Item

1. Copy Item
2. Paste Item
3. Copy Item Permissions
4. Paste Item Permissions
5. Copy Item Metadata
6. Paste Item Metadata
7. Etc.

Every single item is separately pushed to SPO

Microsoft developed the Import Pipeline to alleviate the throttling and protection mechanisms and the chattiness of the CSOM API:

I. Site collection structure, site structure, and list structure are still migrated using CSOM as they were before, and this happens prior to list items being migrated.

II. Instead of items being migrated with one API call at a time, they are now packaged locally into a batch with a corresponding manifest XML file to store permissions, metadata, etc. before being sent to SPO.

III. Instead of items/batches being sent directly to SPO, they are instead sent to an Azure Blob Storage location (provided by either Microsoft or the user).

IV. Once a batch has been successfully uploaded to the Azure Blob Storage location, Metalogix Content Matrix uses a CSOM request to ask SPO to pull content from the Azure Blob Storage location into the previously created list in SPO.

Please see image below:
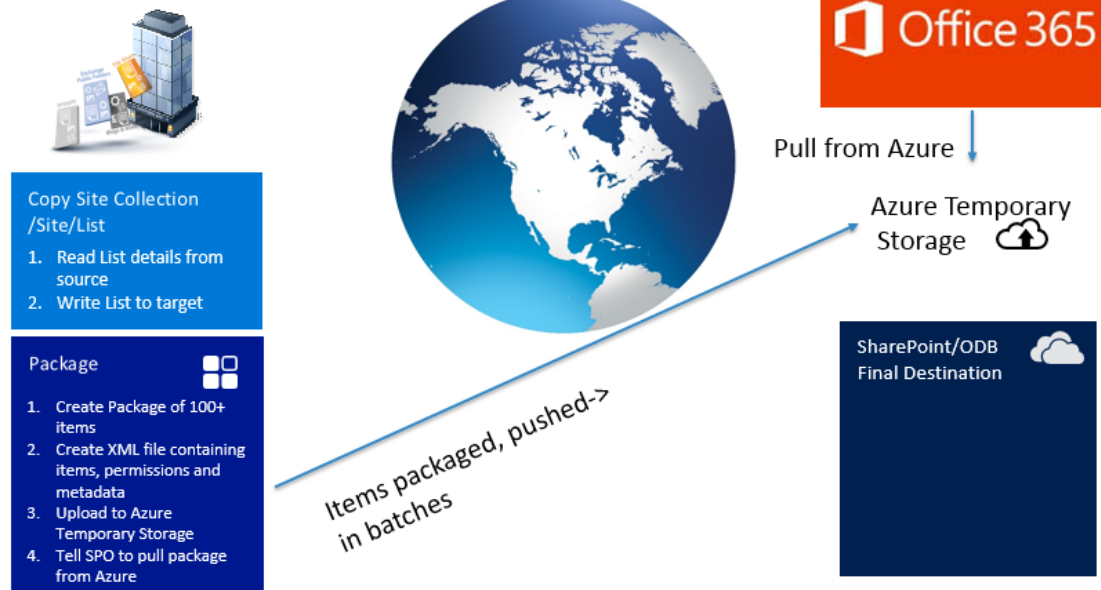
The net result is a significant increase in performance of list items, that can be anywhere between 5 to 30 times faster than these migrations were before depending on a number of factors.

In light of this, it is strongly recommended that our clients leverage the Import Pipeline for all SPO migrations. Users can either provide their own private (paid) Azure containers (as detailed later in this document), or SPO-provided (free) containers.

Refer to the Microsoft article Avoid getting throttled or blocked in SharePoint Online for tips on how to avoid throttling.

Configuring Content Matrix for Use with the Import Pipeline

Using Azure Private Containers

With Azure private (paid) containers you use your own Azure Storage location, which involves setting up an Azure Blob Storage account and configuring Content Matrix to work with it. Microsoft recommends the use of private containers for migrations to achieve faster performance.

For instructions for creating a storage account to manage Azure private containers, refer to the article Create a storage account - Azure Storage | Microsoft Learn. After the storage account has been created, you will also need to create an Access Key and copy the **Connection string** to use when configuring Metalogix Content Matrix migrations.

To specify a temporary storage location for processing of items and files:

1. On the machine(s) where you have installed Metalogix Content Matrix SharePoint Edition, open the file EnvironmentSettings.xml, which can be found in one of the following locations:

   - If you are required to be a local administrator on the machine where Content Matrix is installed: C:\ProgramData\Metalogix
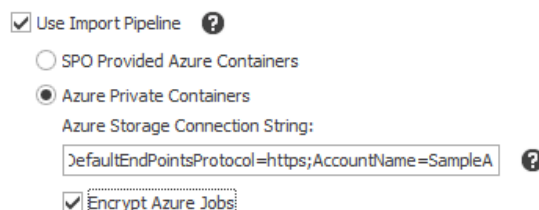
   OR

   - If you are not required to be a local administrator on the machine where Content Matrix is installed: C:\Users\<username>\AppData\Roaming\Metalogix\Common.

2. Find or add the parameter `UploadManagerLocalTemporaryStorageLocation` and set it to a location on your hard drive where processing of items and files can take place. It is important to have at least 40GB of free disk space for this processing. In the example below, I'm using C: \AzureAPI as the location for processing:

```
<XmlableEntry>
<Key>UploadManagerLocalTemporaryStorageLocation</Key>
<Value>C:\AzureAPI</Value>
</XmlableEntry>
```

To configure a migration to use Azure Private Containers:

On the **List Content Options**, tab select the **Use Import Pipeline** option, select **Azure Private Containers**, and enter your **Azure Storage Connection String** in the following format:

DefaultEndpointsProtocol=https;AccountName={your accountname};AccountKey={your account key}.

You may optionally choose to select **Encrypt Azure Jobs** for extra security at the cost of performance:

Removing Private Containers and Queues Created by Content Matrix from Your Azure Storage Account

After a particularly large migration using the Import Pipeline, you may want to remove private containers and queues created by Content Matrix to free up space for more migration actions. Using Azure PowerShell, you can simultaneously remove all containers or queues that were created on the same date. A prerequisite to removing multiple containers and queues is that Azure PowerShell must be installed and configured. Refer to the Microsoft Azure PowerShell documentation for installation instructions and general information about Azure PowerShell Cmdlets.

Naming Conventions

Beginning in version 9.3, Content Matrix uses naming conventions that allow you to easily identify containers and queues created by Content Matrix during migration using the Import Pipeline.

File Type	Naming Convention
Blob Container (per SharePoint list)	YYYYMMDD-<List Internal Name>-<GUID>-<blob>. EXAMPLE: 20200929-shreddocumentt-f0da9f001d3b450cb36ad9e9ecae219f-blob
Metadata Container (per Import Pipeline job)	YYYYMMDD-<List Internal Name>-<GUID>-<Batch number in XXX format> EXAMPLE: 20200929-shreddocumentt-f0da9f001d3b450cb36ad9e9ecae219f-001
Reporting Queue (per Import Pipeline job)	YYYYMMDD-<List Internal Name>-<GUID>-<Batch number in XXX format> EXAMPLE: 20200929-shreddocumentt-f0da9f001d3b450cb36ad9e9ecae219f-001

To remove items from your Azure blob storage account using Azure PowerShell:

1. Ensure there are no active migration actions running that use the Azure storage where the items are stored.

2. Open **Azure PowerShell**.

3. Connect to Azure using the **New-AzStorageContext** cmdlet. Your account name and account key are required parameters. Note that these are different from your Client ID and Application Secret.
   ```
   $context = New-AzStorageContext –StorageAccountName $accountName –StorageAccountKey $accountKey
   ```

4. Use the information in the following table to determine the appropriate command to use.

If you want to remove ...	Then use the command ...
all blob and metadata containers created on a specific date	```Get-AzStorageContainer -Prefix "<MMDDYY>" -Context <context>	Remove-AzStorageContainer -Force``` EXAMPLE: **Get-AzStorageContainer -Prefix "20200929" -Context $context02	Remove-AzStorageContainer -Force**
all reporting queues created on a specific date	```Get-AzStorageQueue -Prefix "<MMDDYY>" -Context <context>	Remove-AzStorageQueue -Force```

If you want to remove ...	Then use the command ...
	EXAMPLE: `Get-AzStorageQueue -Prefix "20200929" -Context $context02 | Remove-AzStorageQueue -Force`

Using SPO Provided Azure Containers

The SPO provided (free) Containers are a set of Azure storage containers that are associated with a user's tenant. This container is provided to a user at no cost from Microsoft and is associated with the infrastructure the user's tenancy exists within. Content can only be uploaded to this location when it is also encrypted, making encryption a requirement.

This action requires no work from the user outside of the selection of the appropriate checkboxes/toggles within Metalogix Content Matrix.

To use SPO provided Azure containers, make sure the **Use Import Pipeline** and **SPO Provided Azure Containers** are selected.

NOTE: The Encrypt Azure Jobs option is disabled for this option because Content Matrix always encrypts jobs using SPO Provided Containers.

Configuration Settings for Migrations Using the Import Pipeline

Metalogix Content Matrix has various XML properties that can be modified to fine-tune or help troubleshoot your Import Pipeline migrations. These properties can be found in the EnvironmentSettings.xml file. Refer to Content Matrix SharePoint Edition documentation for descriptions of Import Pipeline Configuration Settings.

Batch Size Settings

The following XML properties can optionally be modified. However, it is generally recommended to not modify these values unless you are encountering issues when migrating using the Import Pipeline.

UploadManagerBatchSizeItemThreshold

This key controls the maximum batch size in item count, including all folders and documents, to be used if Content Matrix will be submitting batches according to number of items. The default value is 200 and the value must be a positive integer.

```
<XmlableEntry>
<Key>UploadManagerBatchSizeItemThreshold</Key>
<Value>200</Value>
</XmlableEntry>
```

UploadManagerBatchSizeMBSizeThreshold

This key controls the maximum batch size to be used if Content Matrix will be submitting batches according to size. The default value is 1000 megabytes and the value must be a positive integer. The value should be set carefully, taking into consideration factors such as total upload bandwidth, speed of data retrieval from the source system, and so on.

```
<XmlableEntry>
<Key>UploadManagerBatchSizeMBSizeThreshold</Key>
<Value>1000</Value>
</XmlableEntry>
```

Batch Resubmission Settings

MaxAzureBatchRetryCount

This key controls the maximum number of times Content Matrix will resubmit the batch until it is successfully migrated. (The default value is 5.)

```
<XmlableEntry>
    <Key>MaxAzureBatchRetryCount</Key>
    <Value>5</Value>
</XmlableEntry>
```

ErrorMessagesForRetryAzureBatch

This key specifies error conditions for which you want Content Matrix to resubmit a batch, when it normally would not.

> NOTE: Separate multiple error messages with a pipe character (|), as shown in the example below.

```
<XmlableEntry>
    <Key>ErrorMessagesForRetryAzureBatch</Key>
    <Value>Item does not exist|Object Reference Not Set</Value>
</XmlableEntry>
```

Performance Settings

BufferSizeForPipelineMigrationInMb

This key controls the buffer size while uploading files to Azure Storage Account Containers using the Import Pipeline. The default value is 64, which means, for example a 128 MB file will be uploaded in two parts, 64 MBs at a time. The lower buffer size, the more quickly the computer processor handles the information. Keep in mind that the higher the value, the more system resources will be consumed.

```
<XmlableEntry>
    <Key>BufferSizeForPipelineMigrationInMb</Key>
    <Value>64</Value>
</XmlableEntry>
```

MaxParallelUploadFilesInPipeline

This key controls the number of files uploaded in parallel to Azure Storage Account Containers when using the Import Pipeline. The default value is 2, which means a maximum of two files can be uploaded in parallel. Keep in mind that the higher the value, the more system resources will be consumed.

```
<XmlableEntry>
    <Key>MaxParallelUploadFilesInPipeline</Key>
    <Value>2</Value>
</XmlableEntry>
```

UploadManagerMaxRetryCountThresholdForJobResubmission

> **WARNING**: This value should not be changed unless absolutely necessary.

This key controls the amount of time to wait for a response from the reporting queue before re-requesting a migration job. This value is specified in multiples of 15 seconds, meaning that the default value of 960 corresponds to 4 hours and the minimum value of 120 corresponds to 30 minutes. This value must be a positive integer greater than or equal to 120.

```
<XmlableEntry>
    <Key>UploadManagerMaxRetryCountThresholdForJobResubmission</Key>
    <Value>960</Value>
</XmlableEntry>
```

Temporary Storage Location

UploadManagerLocalTemporaryStorageLocation

This key indicates the directory in which the temporary binary files and manifest XML files for each batch are saved. If no filepath is specified, the default file path is used.

```
<XmlableEntry>
    <Key>UploadManagerLocalTemporaryStorageLocation</Key>
    <Value>C:\ProgramData\Metalogix\Temp folder sample</Value>
```

```
</XmlableEntry>
```

Performance Expectations

There are a significant number of factors that will impact the performance of migrations using the Import Pipeline. Many of those are beyond any migration vendor's control. For example, the SPO tenancy has a service that runs to pull data from Azure Blob Storage accounts. If that service is being called on by many clients at the same time, it could significantly slow down migrations.

Another example is the workflow. While the Import Pipeline is significantly faster for list items/documents, it is not faster for complex site collection/sub site structures with very complex and large Managed Metadata trees, workflows, content types, etc. With that in mind, we've seen migration speeds vary between 2GB/hour and 40GB/hour depending on workload when using reasonable hardware for the migration machines involved.

As compared to speeds of between 200MB and 550MB/hour on a single on premises machine, and 400MB/hour to 10GB/hour on Azure VMs, this is a very significant increase. When trying to determine how long a project will take, we still recommend a conservative approach. Test several workload types using the migration API, and preferable using Azure VMs, and see what you get. Run multiple jobs in parallel, and if you see no difference, and can afford the additional RAM/CPU, increase that, and then see what you get. Once you have those speeds, factor in that you will not maintain those speeds 24 hours/day due to variability in SPO. You will also have more remediation work to do post migration due to the differences between SP on premises and SPO.

Using Azure Based Virtual Machines

As mentioned above, we have found that migrations from Azure Virtual Machines (VM) in the same region as the SPO instance being migrated to, increase performance of the migration. The process that we recommend for any migration to SPO is:

1. If you do not know the region your SPO tenancy is in, please contact your MS Technical Account Manager (TAM) if your company has one, or contact MS Support. If MS Support does not give you this information, please contact Quest and we will get you the information.

2. Create an Azure VM in the same region as your O365 tenancy with a Microsoft-supported SQL Server (please note that if you have the bits and rights to use SQL Server, it is less expensive *to* create a Windows only VM, and then install SQL Server on it).

3. Using SQL Server Management Studio, back up your source Content DBs to .bak files.
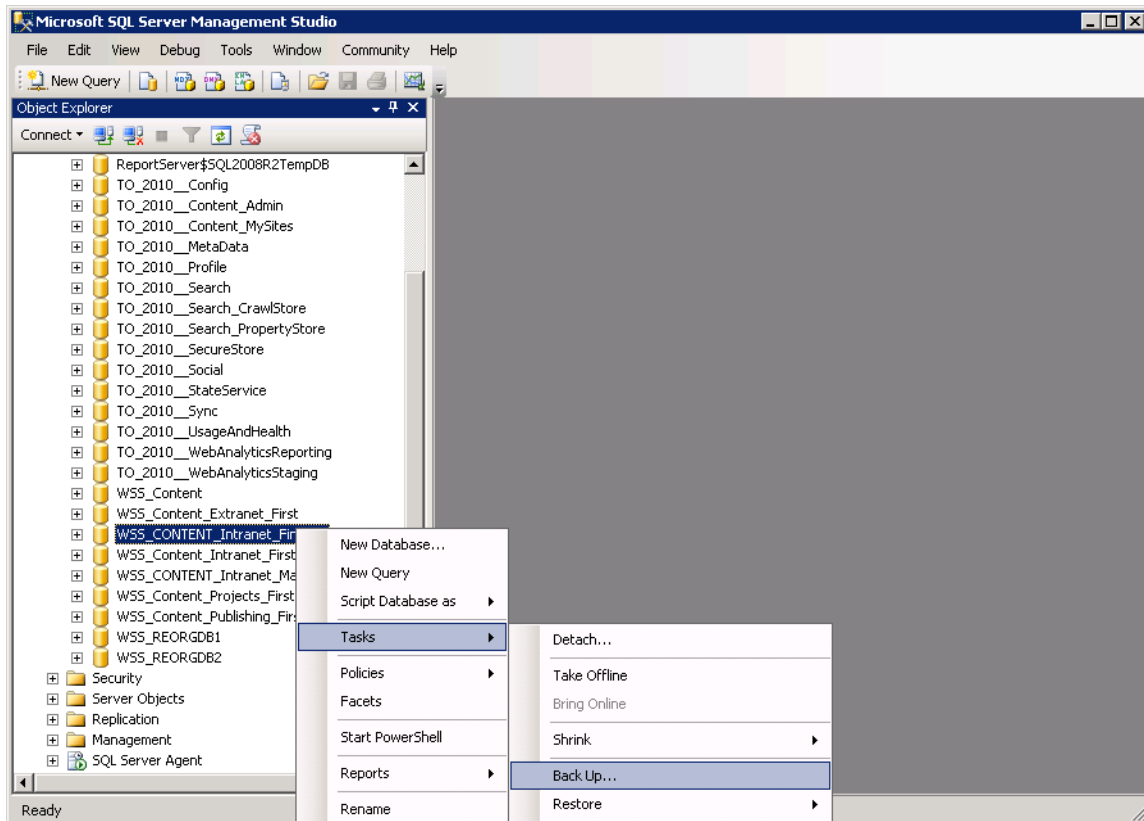
4. Use ftp, or other mechanism to copy the .bak file to your Azure VM. You can ship a physical hard disk to the Azure team if needed. For more information how to ship a physical hard disk, please see Microsoft's guidance here.

5. In your Azure VM, restore the databases from the .bak files.
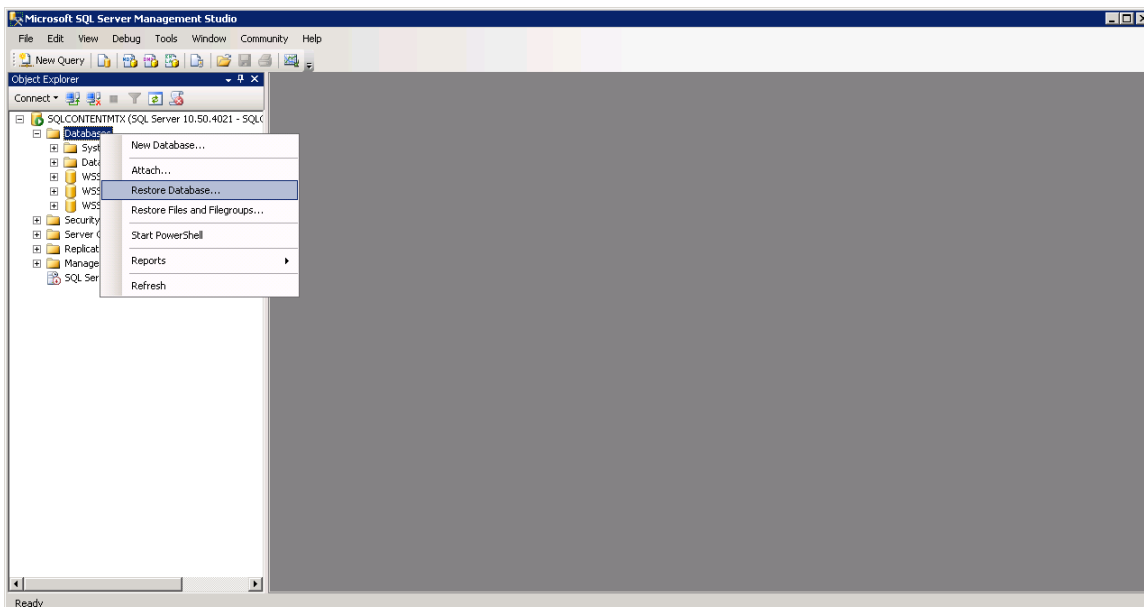
6. Once these databases have been restored, give your Azure VM admin dbo access to each of these databases

7. Install Metalogix Content Matrix SharePoint Edition Console on the Azure VM (you can install the console on multiple other Azure VMs that do not have SQL Server on them, but they must be in the same internal network in order to enable parallel, multi username actions).

8. In Metalogix Content Matrix, use the **Connect to SharePoint Database** option:
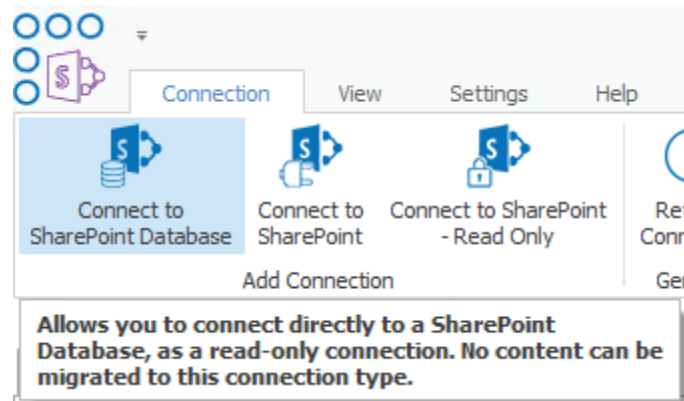
9. Enter the name of the database server (at this point most likely (local)).

10. Choose the Content DB you just restored, and the site collection within that Content DB that you would like to connect to.

11. Migrate content as you normally would with Metalogix Content Matrix.

12. For any operations that require farm level connections (such as copying the Managed Metadata Services Term Store), migrate that from the source farm instead of Azure.

13. On the **List Content Options** tab, make sure that **Preserve IDs on items/folders in Lists** option is selected.

    NOTE: There is no need to re-back up your Content DBs and restore them in Azure as long as you select this option.
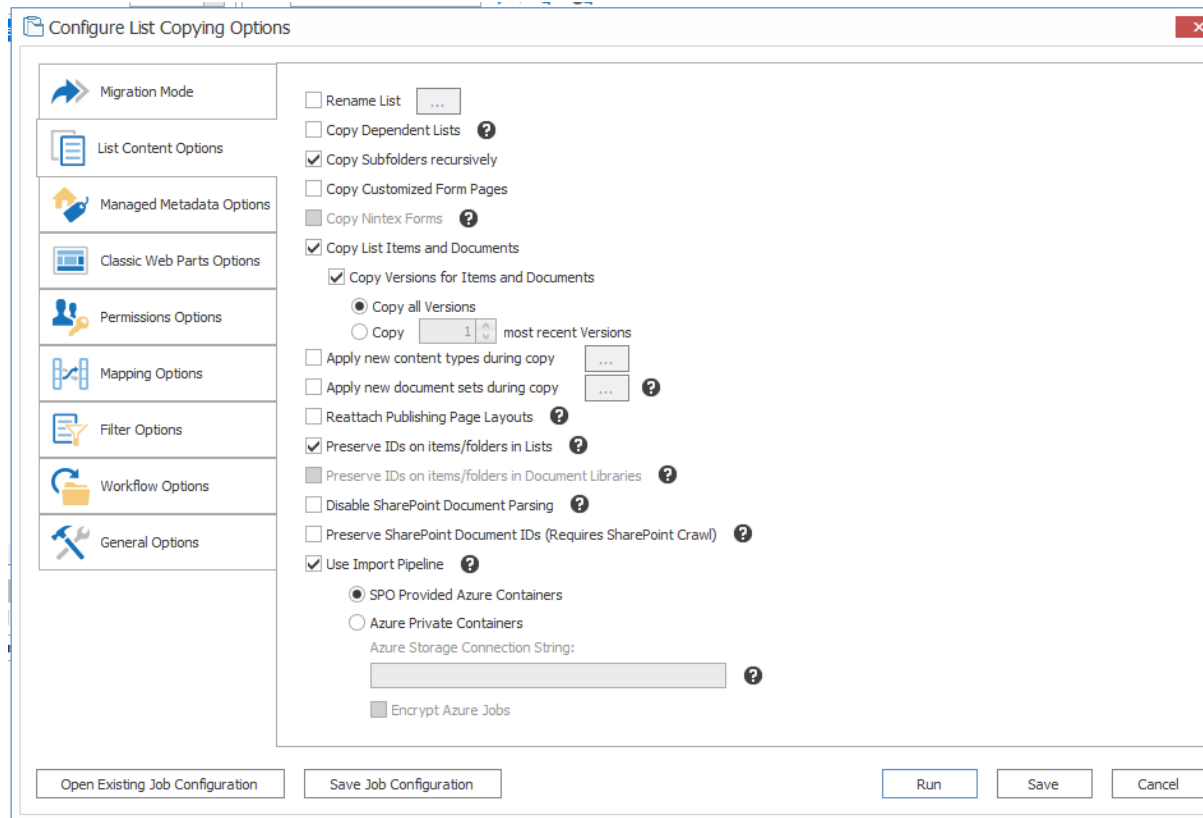
Additional Best Practices Unique to Content Matrix that Can Further Improve SPO Migration Performance

In addition to the above mentioned migration improvements, Metalogix Content Matrix has a number of unique capabilities that make migrations to SPO potentially much faster than they are with Azure/SPO Containers, and we encourage clients who are doing large migrations (1TB or more) to take advantage of these capabilities:

I.   Metalogix Content Matrix has the unique ability to connect directly to SharePoint Content Databases for 99% of the migration from SharePoint on-prem to SPO. This allows our clients to:

   a.   Backup their Content DBs on premises

   b.   Copy their Content DB .bak files to an Azure VM running SQL Server in the same geographical region as their SPO tenancy (if the content DBs involved are very large, you can also choose to ship physical disks to the Azure team, and that team will load them into your Azure VM).

   c.   Restore the Content DBs to SQL Server in Azure VM.

   d.   Use Metalogix Content Matrix' "Connect to SharePoint Database" functionality to migrate directly from the Content DBs in the Azure VM to SPO:

On-Premises SharePoint

SQL Server Backup/Restore

1. Backup Content DBs on-premises
2. Copy Content DBs to Azure VM
3. Restore Content DBs to SQL Server in Azure VM

II. As shown in the above diagram, by moving the Content DBs to an Azure VM in the same geographical region as your SPO tenancy, you have now moved the bulk of the objects involved in the migration to what is effectively a LAN in the SPO geographical region. The Azure VMs are frequently in the same data center as the SPO infrastructure.

III. In addition to migrating directly from the Content DBs, Metalogix Content Matrix allows you to:

   a. Migrate many items in parallel without a steep reduction in performance. Metalogix Content Matrix has significant parallelism and multi-threading capabilities that allow you for example to migrate three or four site collections at the same time from a single Azure VM, and then deploy additional Azure VMs in which you can migrate additional site collections in parallel.

   b. Use the unique Distributed Migration capabilities of Metalogix Content Matrix to distribute the load of migration across a multitude of different agents, thereby increasing the level of parallelism.

   c. Use PowerShell to control any operation available in the Metalogix Content Matrix UI programmatically. This opens up many possibilities including the ability to create a list or CSV file that contains source site collections, target site collections, site collection job configuration files, and status, and then have the migrating machine start the next available migration in this list. This is a recommended configuration in tandem with Distributed Migration to automate migration actions across a multitude of different servers.

   For example:

		Title	Source Site Collection		Target Site Collection	Job Configuration File	Status
☐	📎			▼			
		Finance ⊞ NEW	http://tamirsp2010:1818/		https://metalogixsoftware.sharepoint.com/finance	Standard	Not Started
		Marketing ⊞ NEW	http://tamirsp2010:200/sites/MetalogixIntranet		http://metalogixsoftware.sharepoint/com/sites/MetalogixIntranet	Incremental	In Progress
		Home ⊞ NEW	http://tamirsp2010:300/sites/home		https://metalogixsoftware.sharepoint.com/sites/home	Standard with 3 year item filter	Not Started

4

To achieve performance levels of 10GB/hour or higher

1. Create multiple migration users in your SPO tenancy, each with Global Admin permissions (MS recently rolled out SPO Admin roles. If those are available in your tenancy, you should be able to use those instead of Global Admin roles).

2. Deploy multiple Azure VMs in the same domain as the SQL Server mentioned above.

3. Ensure that the SQL Server VM has sufficient RAM to accommodate multiple simultaneous jobs.

4. Plan your migration so that each VM is migrating several jobs at a time, with larger Site Collections or Sites being migrated on separate machines (i.e., balance the workload across multiple VMs).

5. To achieve optimal performance, first migrate the structure of your SharePoint farm, and then the content.

6. Use Distributed Migration to distribute the load across multiple VMs.

7. Use PowerShell to control migration jobs. This allows you to programmatically control migrations, check against an Excel file, DB, or SharePoint list for a list of jobs that are still required, and perform migrations in the order you would like without monitoring the consoles frequently.

   NOTE: PowerShell also uses far fewer UI resources, and therefore results in faster migrations. PowerShell is also able to take advantage of the Distributed Migration features.

Disabling Unnecessary Features

The following options can significantly increase the time it takes to migrate content and should be turned off whenever possible:

- Verbose Logging

    - **Location**: In the configuration dialog, under the "General Options" tab. A check box labeled "Verbose".

    - **Function**: Turns on verbose logging. Is unnecessary unless full XML of all objects is required in the job listing.

- Item level permissions

    - **Location**: In the configuration dialog, under the "Permission Options" tab. A check box labeled "Copy Item Permissions"

    - **Function**: Copies item level permissions. As a best practice these objects should never be copied unless absolutely necessary.

- Copy form web parts

    - **Location**: Under the "Web Parts Options" tab, a check box labeled "Copy web parts on form pages".

    - **Function**: Copies web parts on form pages. In most cases these forms are not customized and this option will simply add checking overhead.

- Following Content (SPO targets only)

    - This feature is enabled by default when a new site (but not site collections) is created. When it is disabled, performance of Site copies improves by approximately 20% to 25% for document heavy workloads.

    - Since Metalogix Content Matrix does not currently give you the option of turning off only a particular feature on the target site, the only way to achieve this currently is to specify in the **Paste Configuration** - **Site Options** tab to **Clear Default Features on Target.**

Multithreading and Batch Jobs

Multithreading in Metalogix Content Matrix can be used in two places.

- Multithreading within a single job.

- Multithreading in the form of multiple jobs started concurrently.

Jobs can additionally be batched together to more efficiently migrate larger sets of data.

> NOTE: This is more efficient when using Distributed Migration and multithreading.

Single Jobs

Within a single job, Metalogix Content Matrix can utilize multithreading to copy site data concurrently.

Changing Single Job Multithreading

How many threads can be used simultaneously per job in Metalogix Content Matrix is controlled by the **Edit Resource Utilization Settings** option, which is accessible from the Settings ribbon at the top of the Console UI.

If the option is clicked, a dialog containing a slider will appear that will allow control of the number of threads available within a single action. If the slider is moved all the way to the left, this will turn off multithreading and only allow a single thread to be used during the action. (For more information on threading, refer to the Microsoft article About Processes and Threads.)

> NOTE: Turning off multithreading in this way can be a valuable way of troubleshooting whether multithreading is causing issues.

If the slider is left in the middle, it will allow twice the number of processors in threads to be created per action. For example, if the machine running Metalogix Content Matrix has a two core CPU, four threads will be able to be used if the slider is in the below state:
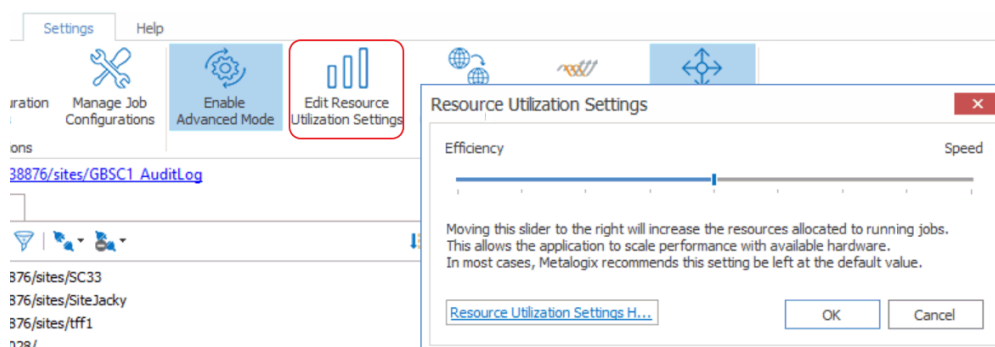
Moving the slider farther to the right will allow more threads to be used, but can potentially overwhelm

system resources. This could lead to potential errors if the system resources cannot handle the data being migrated. There is a chance that if the speed is too high, you would see a slowdown in the overall migration, because the migration is trying to run actions faster than the resources properly allow.

While this value can be set through the Content Matrix Console, you can also set it through the back end, if the UI setting does not seem to be working for you. Please contact Quest Support for more information on this back end setting.

Optimizing Single Job Multithreading

Metalogix Content Matrix will only create new threads when there are multiple SharePoint site nodes to be migrated in the current location of the job. This means that in order to optimize single job multithreading, multiple sites should be migrated within a single job.
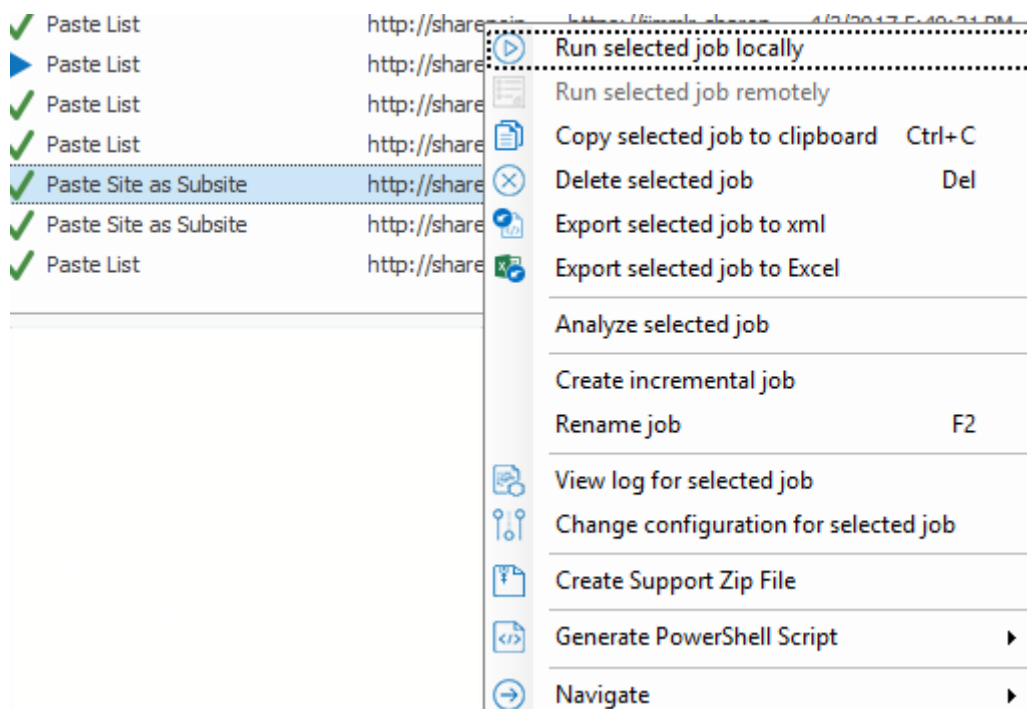
Concurrent Jobs

Metalogix Content Matrix has the ability to concurrently run jobs.

Running Concurrent Jobs

To achieve this, pre-configure two different migrations. In the job listing, there should be two separate entries, as shown below:

| Paste Site as Subsite | http://sharepoin... | https://jimmlr.sharep... | 4/7/2017 6:10:26 PM | Done | Lists Copied: 10, Items Copie... | Bytes Copied: 44.3 MB |
| Paste Site as Subsite | http://sharepoin... | https://jimmlr.sharep... | 4/8/2017 12:08:34 AM | Done | Lists Copied: 10, Items Copie... | Bytes Copied: 44.2 MB |

To run these jobs concurrently, for each job, right click and choose **Run selected job locally**.

After both jobs have been run separately, there will be two status dialogs tracking the status of their respective jobs.

NOTE: An alternate method of running two jobs concurrently would be to generate a PowerShell script for each job and run the scripts simultaneously, or to use Distributed Migration.
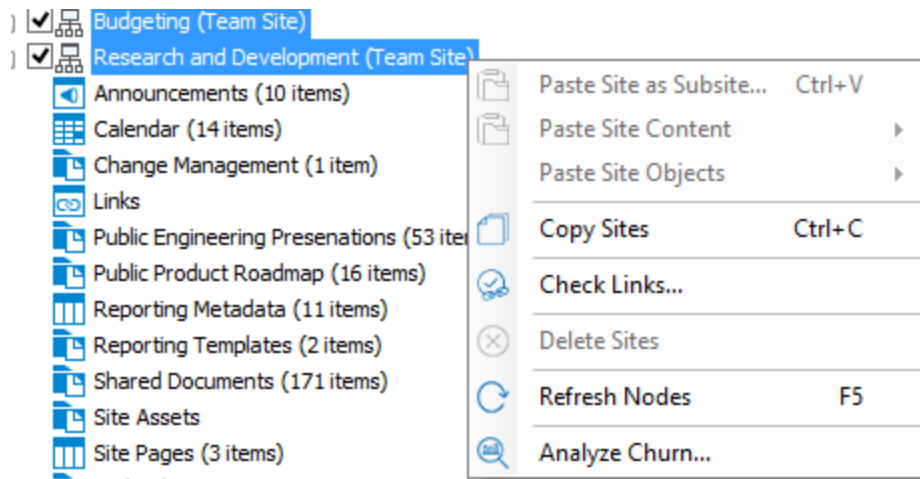
Optimizing Concurrent Jobs

To ensure data integrity, it is important to enforce the following rules when running concurrent jobs:

• Multiple jobs should never migrate data into the same list simultaneously

• Multiple jobs should take care to avoid migrating from the same source data simultaneously

Batched Jobs

Jobs can be batched into groups, and then run. There are two primary ways of batching data into runnable jobs.

• Multi-selecting same-type nodes (sites can only be multi selected with other sites, lists with other lists, etc.)

- Multi-selecting previously configured (or run) jobs

When batching data using either method, migration of nodes within the batch will be run sequentially. This means that if two sites are selected, they will be migrated in order. Similarly, selecting and running multiple jobs will run those jobs from top to bottom.

Batched jobs can be run concurrently with other batched jobs. Previously defined settings on multithreading and concurrent jobs also apply to all jobs run in a batch.

PowerShell

Metalogix Content Matrix has the ability to run jobs within the context of PowerShell. Running jobs in this way provides some notable advantages:

- Resource consumption is minimized (no requirements to render or cache the UI).

- Separate job lists can be used for each distinct job (job lists are the storage medium for jobs.

Configuring PowerShell

To take advantage of PowerShell functionality in Metalogix Content Matrix, simply configure a job via the UI in the normal fashion, and click the **[Save]** button at the bottom of the configuration dialog.

This will create a configuration entry for the job in the job listing that can be used to generate a PowerShell script. To generate a script, highlight the desired job and click the **Generate PowerShell Script** button at the top of the job listing, or right click and choose **Generate PowerShell**.

This will create a script of the configured job that can be copied and pasted into an existing script, or saved or run as desired.

NOTE: It is more efficient to pre-configure jobs in the UI instead of trying to write them initially in PowerShell. This will allow configuration of desired job functionality without having to learn all of the PowerShell parameters for the Content Matrix actions.

Optimizing PowerShell

Optimization of Metalogix PowerShell cmdlets:

- Use the **–quiet** parameter.

  This parameter will disable all visual logging to the PowerShell console (all logs are still tracked in the job listing), and will increase the speed of the migration significantly.

- Use the **–joblist** parameter to specify multiple job listings.

  This parameter will specify which job list to write log entries to when the PowerShell job is running. Providing multiple job lists for logging purposes will help to spread load and decrease bottlenecks during the migration.

Distributed Migration

Metalogix Content Matrix SharePoint Edition is able to use a unique feature called Distributed Migration in order to shift the load of multiple jobs to multiple machines.

In this model, jobs are submitted remotely to a Distributed Database, then pulled directly from the database by any available agent machine. The Metalogix Agent Service installed on each machine manages queuing and processing.

Distributed Migration Agents

Distributed Database
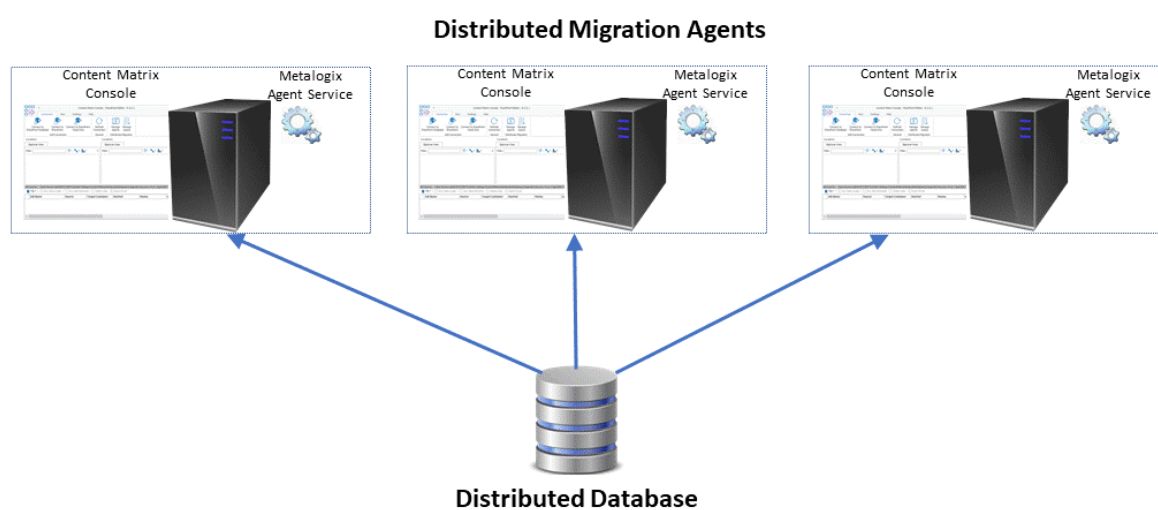
Complete instructions for configuring and using Distributed Migration can be found in the Metalogix Content Matrix SharePoint Edition Online Help.

About Us

Quest creates software solutions that make the benefits of new technology real in an increasingly complex IT landscape. From database and systems management, to Active Directory and Office 365 management, and cyber security resilience, Quest helps customers solve their next IT challenge now. Around the globe, more than 130,000 companies and 95% of the Fortune 500 count on Quest to deliver proactive management and monitoring for the next enterprise initiative, find the next solution for complex Microsoft challenges and stay ahead of the next threat. Quest Software. Where next meets now. For more information, visit www.quest.com.

Technical Support Resources

Technical support is available to Quest customers with a valid maintenance contract and customers who have trial versions. You can access the Quest Support Portal at https://support.quest.com

The Support Portal provides self-help tools you can use to solve problems quickly and independently, 24 hours a day, 365 days a year. The Support Portal enables you to:

- Submit and manage a Service Request

- View Knowledge Base articles

- Sign up for product notifications

- Download software and technical documentation

- View how-to-videos

- Engage in community discussions

- Chat with support engineers online

- View services to assist you with your product

Contacting Quest

For sales or other inquiries, visit www.quest.com/contact.