Ingesting Digital Content at Scale

CONFIDENTIAL INFORMATION

DISCLAIMER

TRADEMARKS

Document released: July 2016

Web address: http://www.exlibrisgroup.com

Table of Contents

Introduction

Digital content continues to proliferate at an unprecedented scale. From widespread digitization efforts to the fast-paced generation of born-digital content, many institutions are struggling to keep up with the deluge. Ingesting digital content is challenging in the best of circumstances; maintaining the quality of metadata, managing the storage and computational infrastructure to support the ingest workflows, and ensuring long-term access each pose unique challenges to those entrusted with the institution's digital treasures.

As an enterprise digital asset management and preservation system, Rosetta has been designed with high-throughput ingest scenarios in mind. While each institution has its own unique requirements, this document describes some considerations when designing a high throughput workflow and highlights some of the features Rosetta supports to optimize such flows.

Improving performance is never complete. Delivering high performance enterprise systems is a complex endeavor that involves software, infrastructure vendors, and internal IT staff. Improvements are made in every release in order to increase the throughput that Rosetta can support. Rosetta improvements can be based on the following:

- Internal testing
- Tuning projects/efforts
- Reports from customers

Scalability Considerations

When building a high throughput workflow, it is important to understand the building blocks of the workflow that affect the overall speed at which content can be ingested. Below are some considerations to keep in mind. Each aspect is discussed in further detail in a later section.

Parallelization

To increase throughput in a system, it is necessary to examine how long an individual process takes and identify opportunities to parallelize that processing. When ingesting digital content, Rosetta parallelizes at the level of the SIP (Submission Information Package). An individual worker thread operates on the package and moves it through the various processing stages. Depending on system resources, the number of worker threads can be increased to achieve higher levels of parallel processing.

Linear Scalability

Rosetta has proven near-linear scalability. This means that as new server instances are added to a Rosetta installation, the amount of material that can be ingested increases proportionally. Of course, the increased ingest rate can be affected by infrastructure bottlenecks such as network saturation, storage latency, etc. Rosetta supports a "go and grow" approach, allowing an institution to start small and add servers according to requirements.

File Handling

Rosetta maintains individual storage locations for three different modules of the system – Deposit, Operational, and Permanent. Combining those modules with the original staging location of the digital content results in three distinct file copy operations that Rosetta performs as it moves the files through the stages of processing.

As moving large numbers of files around the file system can have a big impact on processing time, how the copy operations are handled becomes significant in high throughput scenarios.

File Operations (I/O)

As part of ingesting digital content into the repository, Rosetta performs several operations on the files themselves. In the validation stack phase, Rosetta runs fixity (check sum), virus check, format identification, and metadata extraction operations. In the enrichment phase, Rosetta can be configured to create access copies of the ingested content. Since these activities must be performed for each file, how they are configured can have a great impact on the overall system throughput.

Hardware Planning

When planning infrastructure for a digital repository, there are many considerations that impact the resources required to achieve the desired throughput levels. Rosetta has published minimum system requirements, but to allow for mid-to-long-term planning of resources, below are some benchmarks that have been observed in the Ex Libris lab environment and at customer installations.

As with all benchmarks, an individual institution's throughput level varies depending on many factors. Optimization cycles and institution-specific benchmarking is required to make a more accurate estimate of the resources required to meet ingest goals.

Database

Benchmark Targets

Below are estimates based on standard configuration of Rosetta:

Number of Files	DB Storage
0.5M	175 GB
1M	275 GB
2.5M	570 GB
5M	1050 GB
10M	2000 GB

Considerations

Database storage requirements can be affected by usage patterns and configuration of Rosetta features. Some of the considerations that affect database storage include:

- Descriptive metadata per intellectual entity (IE), representation, and file
- Events
- Publishing
- Technical metadata extraction

Storage

Benchmark targets

Below are estimates based on standard configuration of Rosetta:

Storage	Factor
Deposit	1 x monthly ingest rate
Operational	2 x monthly ingest rate
Permanent	1.2 x repository size

Considerations

The biggest factor affecting required storage size for the Rosetta digital repository is ingest workflow file handling. If files are moved rather than copied, the storage requirements for the deposit and operational storage are reduced considerably.

Compute

Benchmark Targets

In lab and customer scenarios, Rosetta has comfortably ingested 50 GB of files per hour per machine, meeting the minimum system requirements. Some customers have successfully ingested at sustained rates significantly higher than this benchmark.

Considerations

There are many factors that influence ingest rates, the most significant being network bandwidth and disk IOps. Therefore, it is recommended to perform a series of optimization cycles to arrive at targets that accurately reflect the institution's infrastructure.

Scalability Features

Rosetta boasts many features that can be leveraged to increase the rate at which content can be ingested. Since there are trade-offs when implementing these features, institutions should consider the factors outlined below when making decisions about how to configure Rosetta.

This section is intended to address these features as they relate to scalability concerns. For more information and details about how these features are configured in Rosetta, refer to the Rosetta documentation in the Ex Libris Knowledge Center.

Parallel SIP Processing

Rosetta relies on a system of queues and worker threads to parallelize activities. Rosetta maintains queues for several discreet areas of system processing. For the purposes of optimizing ingest levels, the SIP processing queue is of greatest interest. Out of the box, Rosetta is configured to allocate five worker threads to the SIP processing queue.

Rosetta can perform parallel processing at the level of an individual SIP. In order to ensure maximum utilization of resources, it is important to plan the ingest workflow so that there are enough individual SIPs to occupy all of the allocated worker threads simultaneously. So while a SIP can contain more than one intellectual entity, it is often best to leave the ratio at one IE per SIP.

There are many factors which impact the decision of how many worker threads to allocate on an individual server. These factors include the server's resources (CPU cores, RAM, network bandwidth, storage IO rates) and the other work being performed by the server (maintenance jobs, user interface/API requests). Therefore, it is recommended to perform optimization cycles on the institution's actual infrastructure in order to determine the optimum number of worker threads. See below for more information on performing optimization cycles.

Worker Configuration

In order to fully utilize the resources available to it, Rosetta allows the number of workers to be configured. If more powerful servers are provisioned, the work level can be increased. Again, optimization cycles should be performed to determine the proper worker level for the provisioned hardware.

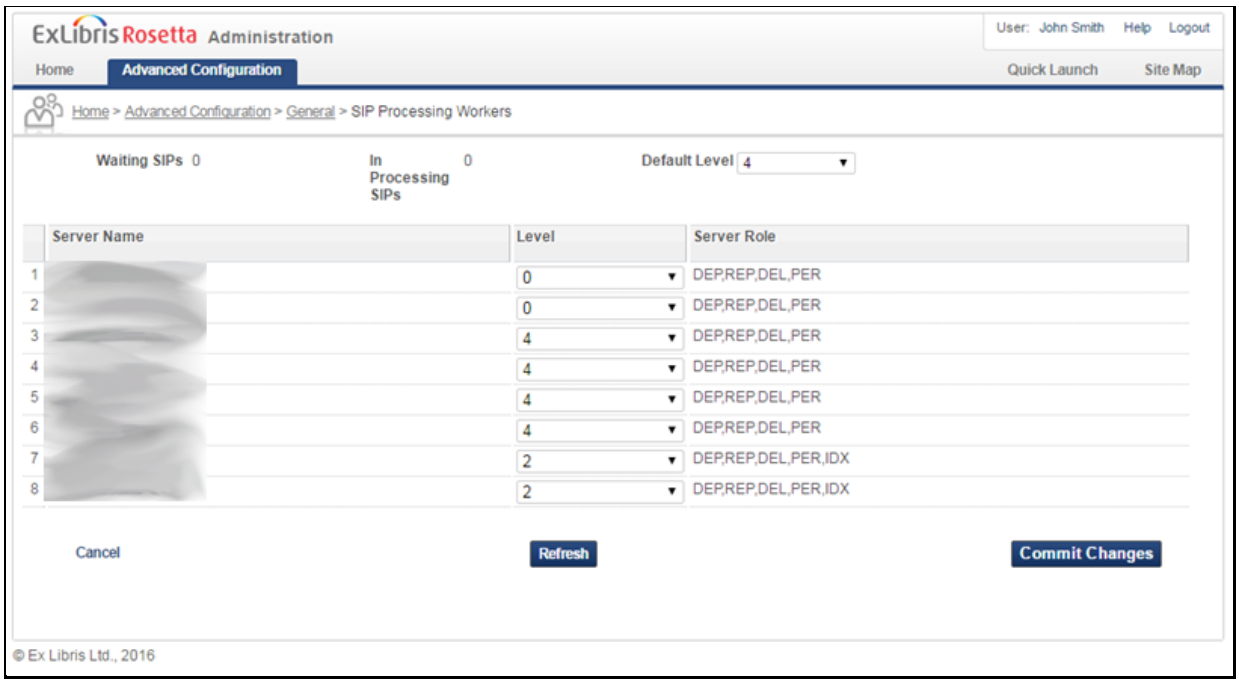

Figure 1: Worker Configuration

Topology

Rosetta supports multiple topologies to support both security and scalability requirements.

Rosetta server roles can be divided functionally for security concerns.

A system administrator may want to make delivery and deposit server roles available outside of the institution's firewall, while the back office roles are only available inside on the institution.

For scalability purposes, we can use all-in-one server roles but configure different levels for SIP processing workload. This ensures that there are enough system resources to handle user interface and API requests efficiently. The remaining servers' resources are fully dedicated to handling SIP processing workload.

In the sample topology below, two Rosetta servers are configured to handle UI requests. A load balancer routes UI requests to those two servers only. Those servers are configured with a lower SIP processing workload to ensure that resources are available for the UI requests. Another six servers are configured with a higher SIP processing workload in order to increase throughput. The worker configuration page above shows the configuration for this topology.

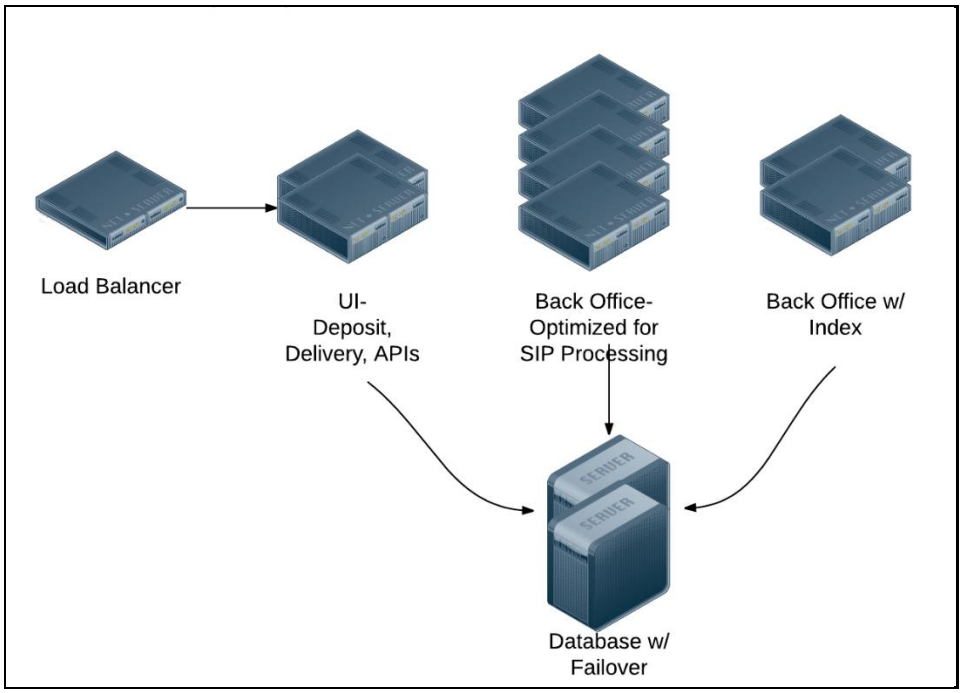

Figure 2: Rosetta Sample Topology

File Handling Method

When ingesting digital content into Rosetta, the files are logically moved through four stations:

- Original storage location

- Deposit storage
- Operational storage
- Permanent repository

Rosetta can be configured to move the files through these stations in different ways:

- **Copy** – The default file handling method is to copy the file from each location. This provides the maximum security and flexibility for the files, as individual copies are maintained throughout the entire workflow and can always be restored in case of error. However, this method requires the most time, network, and storage resources.
- **Move** – In some cases, storage locations are on the same physical mount point. In this case, it is more efficient to move the file to each location, as that requires only a logical update in the file system. However, this means that the file cannot be rolled back in case of corruption as the original file is being operated on.
- **Link** – The most efficient file handling method is to simply link to the original location of the file. This requires no updates to the storage location. Similar to moving a file, linking also limits the ability to rollback to re-process a file if something goes wrong.

File Handling Strategies

Below we highlight several strategies for setting the file handling method.

- Safe and Sound

  Advantages:

  - Files can be rolled back or re-processed at any time

  Disadvantages:

  - Requires more storage space
  - Takes time to copy file

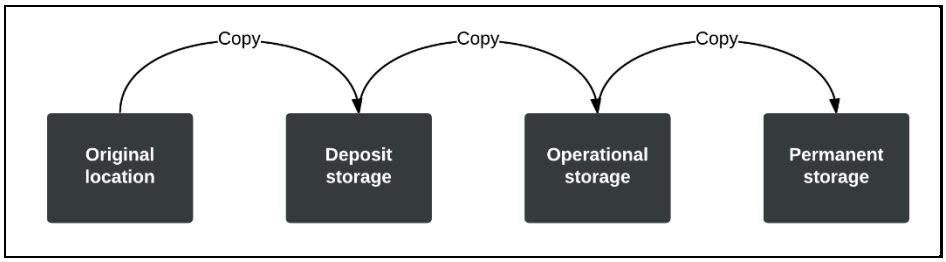

Figure 3: Safe and Sound File Handling

- Balanced

  Advantages:

  - Optimizes moving between operational and permanent

- Maintains a copy on the deposit storage for rollback purposes

Disadvantages:

- Additional space in deposit storage
- Time to copy file to operational

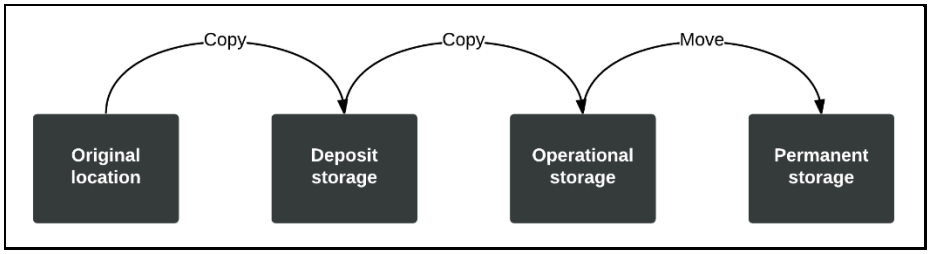

Figure 4: Balanced File Handling

- Throughput Optimized

  Advantages:

  - Requires least amount of storage space
  - Minimal time required for IO

  Disadvantages:

  - Limited options for rollback

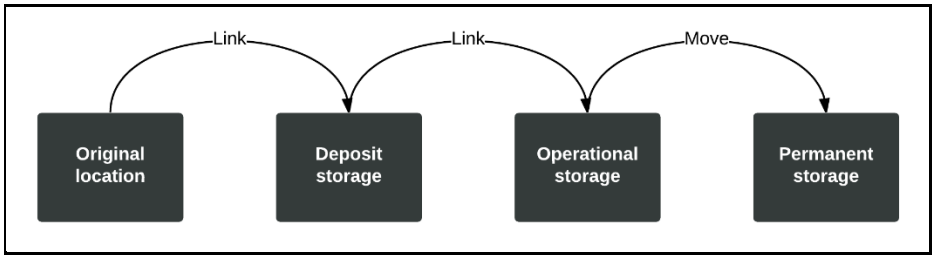

Figure 5: Throughput Optimized File Handling

- In-place / Migration

  This configuration is optimized for a special use case when migrating files from an existing system. In this flow, the files remain in their original storage location while Rosetta characterizes them and adds them to its permanent repository.

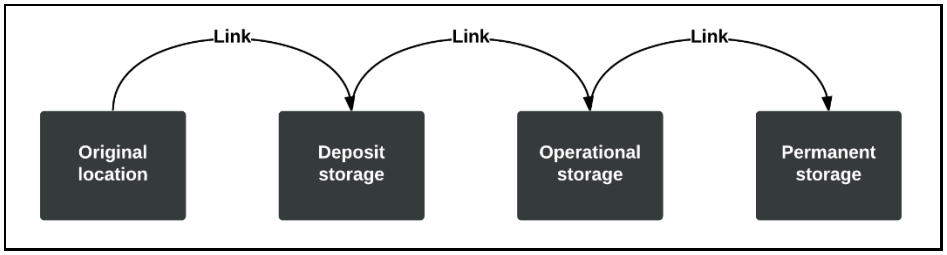

Figure 6: In-Place/Migration File Handling

Format Identification

As part of the validation stack, Rosetta runs a format identification task to evaluate the file and determine its format. The amount of the file that is scanned by the format identification task and provided to the tool can be configured. Out of the box, Rosetta scans the first 64KB of the file as part of this task. This number can be reduced for high throughput workflows, but that may reduce the effectiveness of the format ID process.

File Checksum

As part of the validation stack, Rosetta performs a checksum on the file and uses the checksum to ensure bit-level consistency of the file throughout the file processing workflow and into the permanent repository. Out of the box, Rosetta is configured to calculate the checksum using three algorithms- MD5, SHA256 and CRC32. The fixity algorithms can be limited in high throughput workflows, keeping in mind that this may slightly reduce the overall confidence in the consistency of the repository.

Workflow Configuration

The SIP processing workflow in Rosetta is configured out of the box to leverage all of the features of Rosetta. In some cases, an institution may want to perform some of the tasks pre- or post-ingest in order to optimize the workflow for higher ingest rates. Examples of tasks which are candidates to be removed from the ingest workflow include:

- **Virus check** Files are often scanned for viruses at an institution's edge making an additional virus check in Rosetta redundant. In this case, it is recommended that the virus check in Rosetta be removed from the workflow.
- **Derivative copies** – In some cases, derivative copies are created as a pre-ingest step and are provided to Rosetta along with the original. This reduces the amount of resources required during the ingest flow.
- **Metadata extraction** – Some institutions for whom ingest rates are of primary importance are willing to forgo the metadata extraction step in the ingest workflow. Files are stored in the permanent repository along with full provenance and descriptive metadata. Technical metadata can be extracted at a later time in a maintenance process.

Optimizing Rosetta

To achieve desired throughput levels from an end-to-end ingest workflow, it is recommended to perform a tuning exercise. The result of such an effort is an optimized workflow, configuration, and performance benchmark that will serve the institution in day-to-day operations and in planning for realistic ingest rates.

Rosetta should be configured according to the functional requirements, taking into consideration the issue discussed above.

Process

The process for the tuning effort includes setting resource utilization goals, running test ingests flows, monitoring results, analyzing for bottlenecks, making configuration changes, and rerunning the workflow.

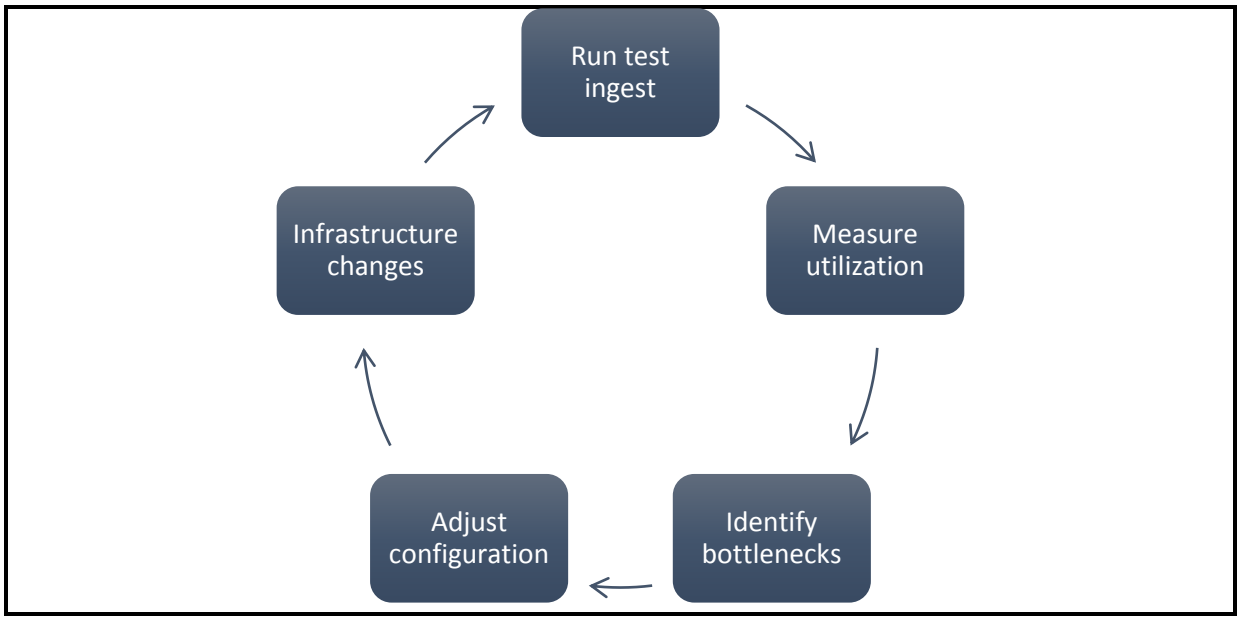

Figure 7: Optimization Flow

In general, we recommend setting a target of 80% CPU utilization. To begin the tuning exercise, representative content is prepared and staged for processing. An initial ingest cycle is executed with the out-of-the-box configuration while monitoring CPU utilization, memory usage, network saturation, and disk latency. Assuming resources are not fully utilized, the number of SIP processing workers is increased and another test run is executed. This is repeated until utilization no longer increases, indicating a resource bottleneck.

If the desired throughput levels have been reached, the process can end there. If not, the bottleneck must be analyzed and the appropriate steps taken to resolve the bottleneck revealed in the infrastructure.

Infrastructure Review

Before the optimization process is started, the infrastructure should be reviewed to ensure it complies with best practices. First, confirm that the environment complies with the current Rosetta system requirements for a new installation. The current system requirements are available in the Knowledge Center.

In addition, validate that all of the required ports are open and that the servers can communicate among themselves. Use the `top` command to check that the system is allocated with the proper resources. Virtual machines must have dedicated resource allocation. The full virtualization requirements are documented in the Knowledge Center.

Mount Points

The mount points for NFS storage should be configured with the following parameters:

```
rw,noatime,nodiratime,bg,nolock,hard,nointr,tcp,vers=3,timeo=6000,rsiz
e=32768,wsize=32768,actimeo=6000,retrans=6000,noacl,intr
```

Heap Size

Out of the box, Rosetta is configured with a Java heap size of 4 GB. In a high throughput environment, it is recommended to increase it to 8 GB. As a rule of thumb, between 25% to 33% of the total RAM in the server should be allocated to Rosetta. This can vary depending on the use of third-party components in the ingest workflow. For example, creation of derivative copies with Imagemagick requires more memory.

Tools

Several tools can be helpful during the optimization process.

Note: The scripts described below are available in the Rosetta Optimization Github repository.

`nmon` Script

The following script can be used to monitor the relevant system resources during the tuning exercise:

```
cd /exlibris
mkdir nmon
cd nmon
wget
http://sourceforge.net/projects/nmon/files/nmon_x86_rhel54/download
mv nmon_x86_rhel54 nmon
chmod +x nmon
```

`nmon` can be configured to run with various options (see the nmon documentation).

For regular monitoring, add the following to `crontab` (The results will be saved in a file under the home directory of the UNIX user adding the job):

```
00 00 * * * /exlibris/nmon/nmon -f -t -s60 -c 1440
```

The `nmon` log files can be loaded into an Excel template for analysis.

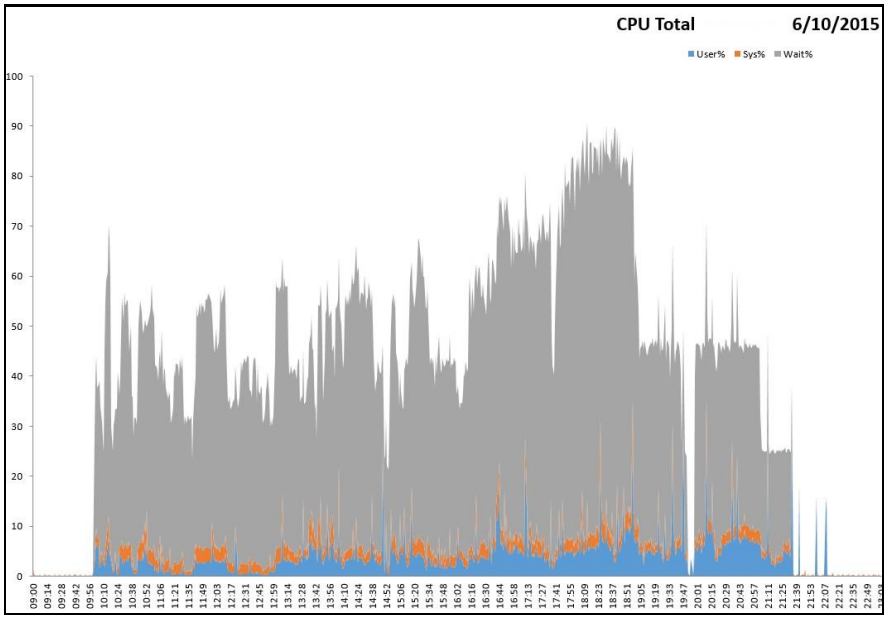

Figure 8: Output of `nmon` analysis

Log Mining

Rosetta prints the time it takes to execute each task in the SIP processing workflow to the log file. It can be helpful to analyze the log file to look for long running validation stack tasks. This often can point to infrastructure performance problems such as slow disk speeds or saturated network.

The following can be used to identify long running individual VS tasks in the Rosetta log file:

```
dps_log
grep VSSIP server.log | egrep -v "Dur: 0|Dur: 1 " | awk '{print
$12","$18","$21}' > <filename.csv>
```

The results can be loaded into Excel for sorting and analysis.

Queue Monitoring

To ensure robust operation, Rosetta utilizes several worker queues each with its own pool of worker threads. The default settings for the number of workers in each pool is sufficient in most cases. Some customers may have content or workflows which require customization of the default settings. For example, IEs with an especially large number of files.

In such cases, the queue backlog can be monitored during an ingest run to determine if there are sufficient worker threads allocated to handle the workload. A monitor queues script can be run which writes the number of SIPs waiting in each queue at that time.

```
csh -f monitor_queues.csh <duration(sec)> <frequency(sec)>
```

For example, you can execute the following, which logs the queue status every 15 seconds for an ingest that is expected to last for four hours

```
csh -f monitor_queues.csh <14400> <15>
```

The output is written to a file and can be loaded into Excel for analysis. A healthy system is represented below, with a bell curve for the SIP processing queue and a relatively flat backlog for the other queues:

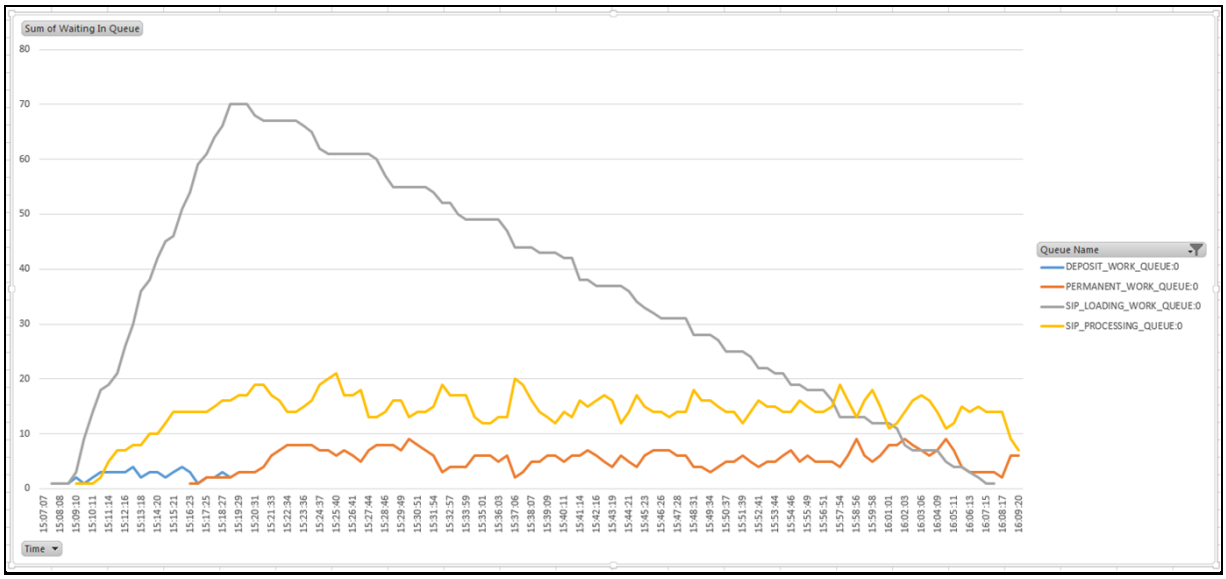

Figure 9: Worker Queue Monitor

Tuning

As a result of the findings of the optimization cycles, several adjustments can be made.

- Heap size can be changed based on the garbage collection logs and the memory utilization indicated by the `nmon` results.
- The work level can be adjusted based on the CPU, memory, and network utilization shown in the `nmon` output.
- Several environment-related issues may be revealed, including:
  - connection timeout
  - I/O latency or NFS configuration (such as no-lock)
  - network connectivity

- Worker tuning – If the system resources are optimally utilized, and the environment is configured correctly, the backlog in the different worker queues can be monitored with different ingest flows

After each adjustment, the cycle should be rerun to measure the impact of the change. This process is repeated until an acceptable ingest rate has been achieved. However, changes in infrastructure, workflows, or content will require a new optimization process.

Case Studies

Below are several case studies from institutions for which throughput rates are critical and which have undergone efforts in 2016 to optimize their workflow, configuration, and infrastructure. Each institution has unique needs and capabilities; however, these case studies are helpful as examples of the optimization process and its results.

National Library of Israel

The National Library of Israel (NLI) is responsible for the preservation or the cultural heritage of the State of Israel. The NLI is in the midst of a large scale digitization project, the results of which are made available to the public on the library's website.

Area	Size
Topology	▪ 3 REP servers ▪ Additional 2 DEL servers ▪ Each with 8 cores and 64 GB RAM
Content Profile	▪ Scanned daily newspapers (10%) ▪ Images (90%) ▪ Other: Audio files, digitized books, audio/video, etc.
Performance Goals	▪ Maximize the system and keep up with incoming digitization workload
Maximum Sustained Throughput Rate	▪ 3 TB and 180,000 files per day
Current Repository Size	▪ 15.5M files ▪ 2.8M IEs

Large Religious Institution in the US

Serving as the library and archive of a large religious institution in the US, this institution receives content from its branches all over the world. It currently operates two Rosetta instances – a dark archive for preservation purposes and an instance with content made available on its public website.

Area	Size

Topology	▪ Dark Archive
	  ▪ 7 all-in-one servers (6 deposit, 1 UI)
	  ▪ Each with 8 cores / 32 GB
	▪ DAM
	  ▪ 5 servers (2 delivery)

Content Profile	▪ Dark Archive – mostly images ▪ DAM- mixed- PDFs, images, etc.
Performance Goals	▪ 10 TB per day
Maximum Sustained Throughput Rate	▪ 5.2 TB and 25,000 files in 10 hours.
Current Repository Size	▪ 1.2 PB

Bavarian State Library

The Bavarian State Library is located in Munich, Germany. As a partner in a large book digitization project, it needs to ensure that Rosetta can keep up with the rate of incoming material.

Area	Size
Topology	▪ 8 all-in-one servers ▪ 2 UI ▪ 6 back office (2 with index) ▪ 16 GB RAM each
Content Profile	▪ Mostly digitized books
Performance Goals	▪ 200 books per day (2TB, ~ 200,000 files)
Maximum Sustained Throughput Rate	▪ 2 TB and 200,000 files in 10 hours.
Current Repository Size	▪ 160,000 IEs ▪ 900,000 files

Summary

Rosetta is a proven solution for the large digital repositories with significant throughput requirements. Rosetta has achieved ingest rates in the field that allow for the creation of a **4 petabyte repository within a year.** By following the advice provided in this white paper, your institution can optimize its workflows and infrastructure to take advantage of the scaling features provided by Rosetta. For consultation regarding optimizing your institution's digital ingest workflows, you can speak to your implementation project manager or open a support case.