Preparing Qlik® environments for Qlik NPrinting®

Prerequisites for QlikView® and Qlik Sense®

January, 2018

Table of Contents

Preamble

Qlik NPrinting is a server based reporting component of the Qlik platform that consumes data from Qlik servers to produce reports for end users. Qlik NPrinting can scale to perform high volume reporting and will place additional load on existing Qlik architectures.

This document drills into the interactions between QlikView, Qlik Sense and Qlik NPrinting and establishes best practices for preparing an architecture that meets the common reporting scenarios.
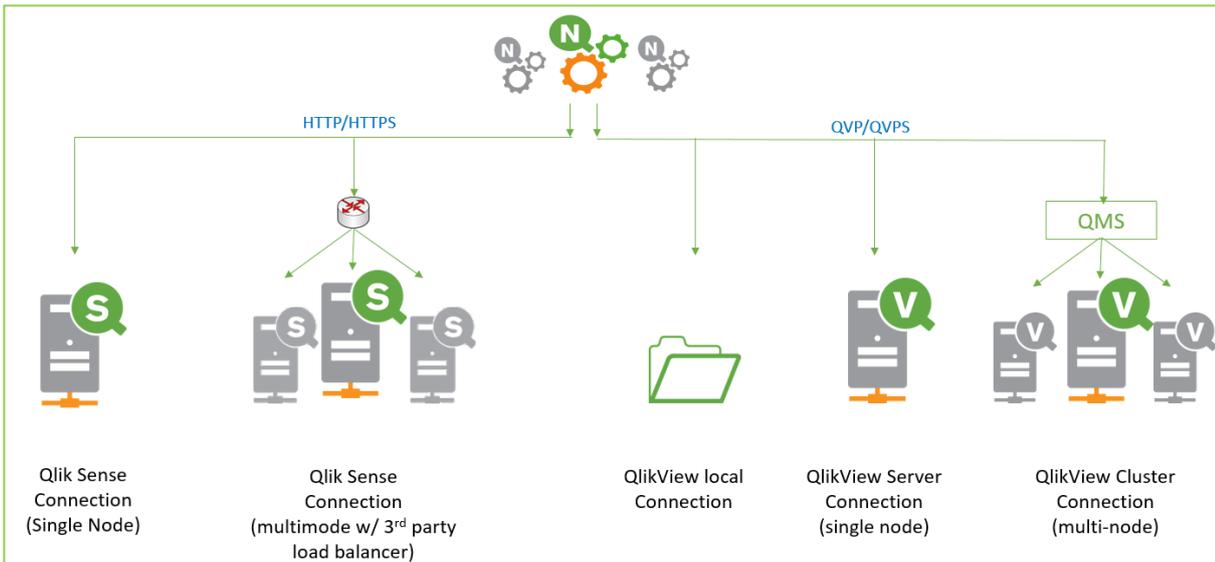
This guidance is intended solely for general informational purposes and its contents shall not be incorporated into, and do not form part of, the product documentation or any contract or agreement with Qlik. The information regarding the products in this document are subject to change without notice. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS DOCUMENT, ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. Qlik does not represent, warrant, or guarantee any particular outcome or result and disclaims liability for any direct or indirect losses resulting from any reliance upon information contained or omitted in this document. Qlik makes no commitment to deliver any future materials, code, or functionality and purchasing decisions should not be based upon any future expectation.

Connection Types

When Qlik NPrinting retrieves data from QlikView and Qlik Sense it opens a connection to a Qlik application. There are several connection types.

| Qlik Sense Connection (Single Node) | Qlik Sense Connection (multimode w/ 3rd party load balancer) | QlikView local Connection | QlikView Server Connection (single node) | QlikView Cluster Connection (multi-node) |

With QlikView:

Qlik NPrinting Engines require a licensed local copy of QlikView desktop to form a connection to a QlikView application. A Qlik NPrinting engine will spawn one or more QV.exe processes to handle these connections.

Qlik NPrinting connects to QlikView applications in one of three ways:

- Local connection:
  Uses a UNC file path to open a copy of the QVW using the hardware resources of the Qlik NPrinting engine.
- Server connection:
  Uses a QVP path to connect to a QVW hosted in RAM by a single QlikView server
- Cluster Connection:
  Also uses a QVP connection but first connects to the QlikView management service to obtain a QVS server name with which to form the connection

With Qlik Sense:

Qlik NPrinting Engines connects directly to Qlik Sense via a valid Qlik Sense proxy address over HTTP/HTTPS. No local resources are required to manage the connections.

It is also possible to connect to a multi-node Qlik Sense Site leveraging an existing 3rd party load balancer to improve connection resilience across multiple Qlik Sense proxy nodes.

Content Resolvers

Each live connection between a Qlik NPrinting Engine and a QlikView or Qlik Sense application is called a 'content resolver'.

A content resolver is needed whenever one of the following things happen in Qlik NPrinting:

- A report is run as part of a publish task
- A report is run through On-Demand
- Loading metadata in a connection
- Previewing a report from Qlik NPrinting Designer

By default, a single Qlik NPrinting engine can launch multiple content resolvers to one or more QlikView or Qlik Sense applications. Multiple content resolvers may connect to a single Qlik application, multiple Qlik applications, or a mix of both Qlik Sense and QlikView applications. The maximum number of content resolvers is determined by the number of logical processors on each of the Qlik NPrinting engines present in a Qlik NPrinting Site.

Each Qlik NPrinting engine can form the following number of content resolvers:

- QlikView = the sum of all the logical processors across each Qlik NPrinting engine
- Qlik Sense = the sum of all the logical processors across each Qlik NPrinting engine multiplied by 2

Example 1

	# Qlik NPrinting Engines	# Logical Processors per Engine:	Connection Multiplier	Maximum Content Resolvers	Formula
QlikView	2	4	1	8	2 x 4 x 1 = 8
Qlik Sense			2	16	2 x 8 x 2 = 16

In Example 1, the Qlik NPrinting environment will spawn up to 8 content resolvers to QlikView and at the same time spawn 16 additional content resolvers to Qlik Sense.

When applied against the following connections; (1 QlikView, 3 QLIK Sense) the content resolvers would open/close against all required applications as needed. A sample point in time snapshot of an active environment for example could yield the following map of content resolvers:
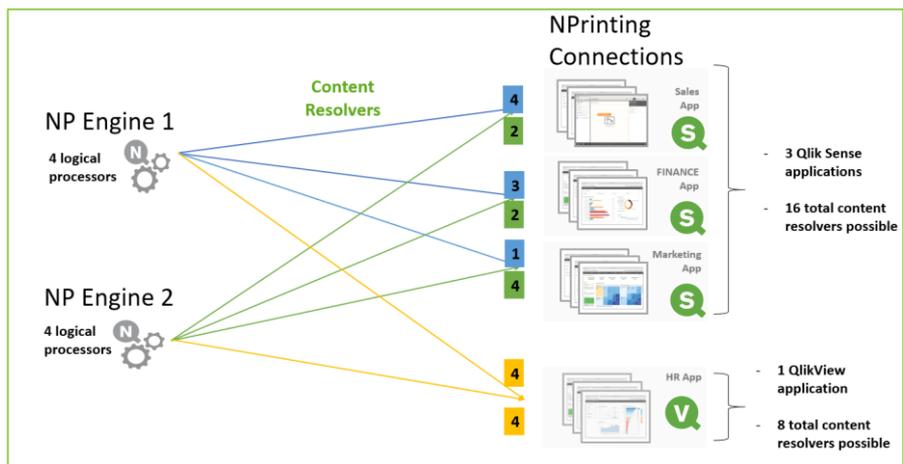
Example 2

	# Qlik NPrinting Engines	# Logical Processors per Engine:	Connection Multiplier	Maximum Content Resolvers	Formula
QlikView	3	12	1	36	3 x 12 x 1 = 36
Qlik Sense			2	72	3 x 12 x 2 = 72

In Example 2, the Qlik NPrinting environment will spawn up to 36 content resolvers to QlikView and at the same time spawn 72 additional content resolvers to Qlik Sense

Content Resolvers across multiple Qlik Servers

In Qlik NPrinting November 2017, a Qlik NPrinting site can connect to only one Qlik Sense site. The same Qlik NPrinting site can however connect to multiple QlikView Servers and/or multiple QlikView clusters. When connecting to multiple QlikView Servers or clusters however, the total number of content resolvers remains capped. The maximum is *by product.*

Simulating load from Qlik NPrinting

Qlik administrators can use the Qlik Sense Scalability tools to approximate Qlik NPrinting load on a Qlik system based on the maximum number of content resolvers. The difference with a simulated Qlik NPrinting content resolver and a simulated Qlik user is that the simulated Qlik NPrinting content resolver will have 0 "think time" between requests. The Qlik Sense scalability tools support the creation of automated routines that can be scaled to the maximum number of content resolvers with each step of the automated routing allowing for 0 "think time". Use of levels, pages and cycles with Qlik NPrinting reports tend to magnify and multiply the number of requests made by content resolvers upon Qlik increasing the load on QlikView and Qlik Sense.

Maximizing Performance

Avoid local QlikView connections

When using QlikView local connections, content resolvers open the QlikView app locally on the Qlik NPrinting engine. This drives up the need for additional RAM and CPUs on the Qlik NPrinting engine machine.

If multiple content resolvers are needed for the same connection, the QVW will be opened multiple times in RAM. The same application with a QlikView server connections or a QlikView cluster connections would result in the QlikView application opened on the QlikView server just once per QlikView server. Avoid local connections wherever possible.

Time to open a QlikView application

Content resolvers to Qlik Sense open relatively rapidly to Qlik Sense over HTTP(s) and no local client application is used to form the connection.

With QlikView, the content resolver must open the QVW in QV.exe either locally or over QVP. Opening a QlikView application through the client has a bearing on performance. For example, larger QVWs will take longer to open.

To help mitigate the time needed to open a QlikView application, Qlik NPrinting will try and leave one open content resolver per unique QVW open. If multiple content resolvers are opened to the same QlikView application, all content resolvers but one will eventually timeout. Leaving a QlikView application open through a content resolver helps to speed-up the performance of a follow-up request to the same QlikView application whether it is for the same report or a different report. A single content resolver will remain open to each unique QlikView connection until:

1. Its needed by another connection
2. A metadata reload is triggered.

When a metadata reload is performed on an Qlik NPrinting connection, all content resolvers against the underlying QlikView application are closed.

Reducing QlikView Connection "Thrashing"

Assuming load across all connections are even, it makes sense to have a greater number of logical processors on the Qlik NPrinting engine than there are unique QlikView connections. This allows Qlik NPrinting to maintain at least one live content resolver per QlikView connection and reduce the number of times Qlik NPrinting open/closes a QlikView application.

A busy system with many different connections and a low processor count will tend to "thrash" QlikView connections as the content resolvers frequently open/close connections to keep up with demand.

Please note that when using QlikView Connections, Qlik NPrinting Engines will scale in a linear fashion up to 12 logical processors per Qlik NPrinting engine. To add more logical processors (and content resolvers) to an environment beyond 12 logical processors it is recommended to add another Qlik NPrinting engine node to the deployment.

Systems using Qlik Sense exclusively can vertically scale linearly beyond 12 processors. The '12 processor limit' best practice applies only to QlikView connections.

Dedicating Qlik servers for use by Qlik NPrinting

For heavy and continuous Qlik NPrinting usage that occurs concurrently with regular user load of the existing Qlik Sense or QlikView environment, it is recommended to dedicate a QlikView and/or Qlik Sense server to handle the Qlik NPrinting load. This is called a "Reporting Node". Reporting Nodes are common when supporting On-Demand scenarios. They also occur when offering 24/7 subscription via NewsStand or intra-day scheduled reporting

Qlik NPrinting On-Demand

When a Qlik application is deployed with On-Demand, active users on the Qlik Server have the option of sending reporting requests to Qlik NPrinting. The filter context is sent to Qlik NPrinting along with the report request and Qlik NPrinting forms its own connection to the Qlik application to mine the data it needs to produce the report requested by the user.
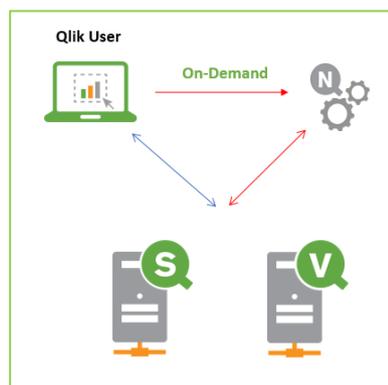

Since Qlik needs to respond to both the user and Qlik NPrinting requirements at the same time, On-Demand use cases tend to 'double up' or magnify the stress on QlikView and Qlik Sense.

On-Demand use cases guarantee concurrent stress by the end user population and Qlik NPrinting at the same time and for this reason it will often make sense to dedicate a Qlik Sense or QlikView node to act as a dedicated 'reporting node'

Reporting Nodes

With Qlik Sense or QlikView, you can either dedicate an independent Qlik Sense site to serve Qlik NPrinting or you can configure a reporting node within an existing Qlik Sense Site or QlikView cluster to serve Qlik NPrinting.

Reporting Node within a Qlik Sense Site / QlikView Cluster
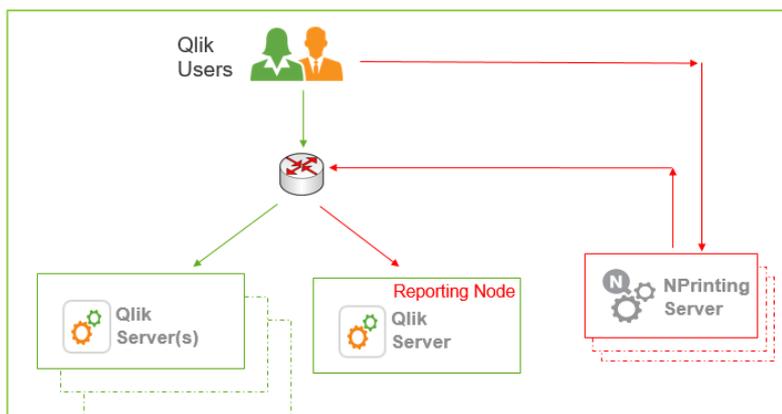
With Qlik Sense

Use **load balancing rules** to dedicate the reporting node within an existing Qlik sense site.

The load balancing rule would ensure content resolver requests sent by the Qlik NPrinting engine account are directed exclusively to dedicated nodes in the Qlik Sense site. This includes Qlik NPrinting requests initiated by any content resolver serving publish tasks, on-demand requests, and metadata reloads.

http://help.qlik.com/en-US/sense/Subsystems/ManagementConsole/Content/load-balancing-rules-overview.htm

With QlikView

Use the "**Document Available**" and select one or more of the **"Cluster Nodes"** to act as a reporting node.

This will ensure that a QlikView application with On-Demand enabled is deployed on dedicated QlikView server(s) that serve the users of the application as well as the Qlik NPrinting on-demand requests.

http://help.qlik.com/en-US/qlikview/Subsystems/QMC/Content/QMC_Documents_UserDocuments_Server_Performance.htm#Customize

Both Publisher and an additional QlikView server are needed for the configuration of a reporting node within an existing QlikView cluster.

When intra-day reporting needs increase beyond On-Demand scenarios it may make sense to host an independent QlikView server or cluster. This is because when intra-day reporting needs increase the concurrent demands of Qlik users and Qlik NPrinting on the same QVW and separating the workload of Qlik NPrinting and Qlik users in a meaningful way no longer becomes viable with the technique described above.

Summary

Best Practices for Preparing Qlik for Qlik NPrinting

- Determine the maximum of content resolvers

- Ensure that QlikView or Qlik Sense is sized to handle the maximum number of content resolvers.
  - Consult the Qlik NPrinting Architecture and Scalability white paper for benchmark test samples
  - Use the Qlik Sense Scalability tools to test the expected load.

- Dedicate an independent QlikView server or a new Qlik Sense "reporting node" when Qlik NPrinting reporting needs overlap with regular usage of QlikView and Qlik Sense. On-Demand is a scenario where needs magnify and overlap requiring a dedicated Qlik node to support.

Additional best practices for QlikView Connections:

- Avoid using QlikView local connections

- Where possible, avoid using very large QlikView applications as a source for Qlik NPrinting because they take longer for the content resolver to open and longer for a report to be produced.

- try to configure fewer QlikView Connections than the maximum content resolvers to mitigate connection thrashing.

Architectures for Common Reporting Scenarios

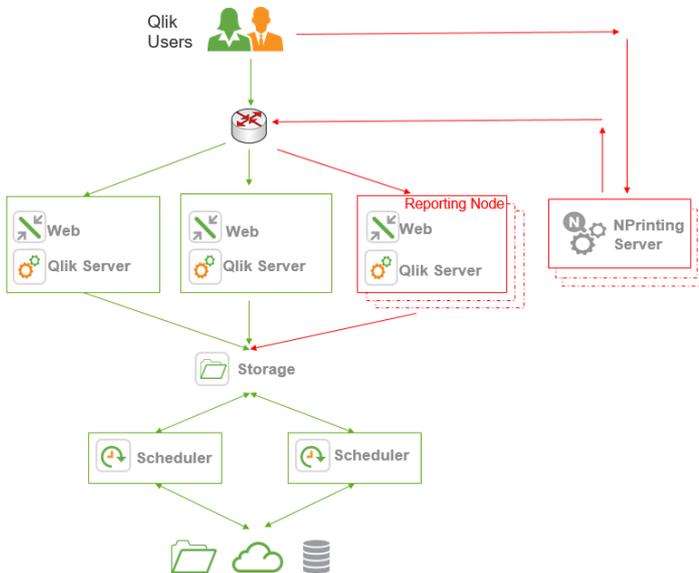## Scenario 1: Overnight Reporting
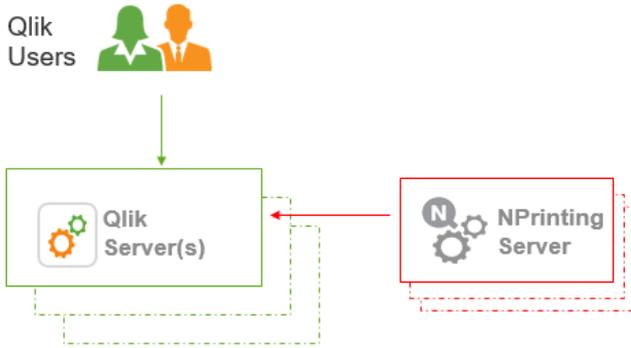
- Add an Qlik NPrinting server to existing QLIK architecture

- Qlik NPrinting reports are run overnight or "off hours"

- Usage of Qlik servers is reserved for users during the day and Qlik NPrinting overnight

- Scale up Qlik NPrinting servers to increase throughput and reduce the reporting production window

Scenario 2: On-Demand Reporting
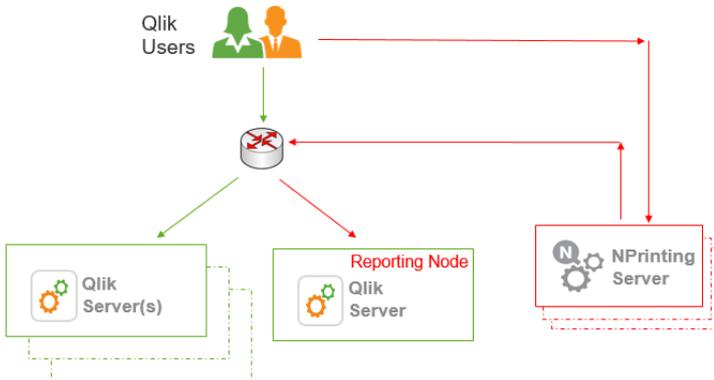
- Relevant for 24/7 NewsStand subscriptions
- Add an Qlik NPrinting server and a dedicated Qlik Server "Reporting Node" to existing Qlik Architecture.
- Leverage a load balancer to centralize the reporting node into an existing Qlik Sense Site or QlikView cluster
- Qlik NPrinting reports can run concurrently with active Qlik users
- Supports On-Demand and/or 24/7 subscriptions via NewsStand
- Scale up Qlik NPrinting servers and Qlik servers to increase throughput and concurrency.

Scenario 3: Enterprise Reporting

- Enterprise Architecture across Qlik and Qlik NPrinting

- Supports On-Demand reporting, 24/7 subscription reporting, and overnight reporting

- Leverages dedicated reporting nodes.

- Scale Qlik NPrinting servers and reporting nodes to increase throughput and concurrency

- Secure enterprise platform providing consistent data for visualization + reporting needs