

# **Qlik Sense Enterprise Client-Managed May 2021: Hybrid Data Load Performance Improvements**

With the Qlik Sense Enterprise Client-Managed<sup>®</sup> (QSECM) May 2021 release, Qlik has increased the maximum number of On-Demand App Generation (ODAG) requests which can be concurrently processed from 10 to 50. This technical brief will provide an overview of the change, benchmark tests demonstrating the improvement, and guidance on leveraging this change on a Qlik Sense Enterprise Client-Managed deployment.

## Contents

Background	1
What are the performance improvements in May 2021?	1
<b>Moderate Concurrency Test</b>	<b>2</b>
<b>High Concurrency Test</b>	<b>3</b>
Performance Tuning ODAG / Dynamic Views	5
Architectural and deployment considerations	7
<b>Compute Requirement Test</b>	<b>8</b>
Conclusion	10

## Background

This technical brief assumes a general understanding of On-Demand App Generation and/or Dynamic Views. For further background on these techniques refer to Qlik's Help:

- On-Demand App Generation: [https://help.qlik.com/en-US/sense/Subsystems/Hub/Content/Sense\\_Hub/DataSource/Manage-big-data.htm](https://help.qlik.com/en-US/sense/Subsystems/Hub/Content/Sense_Hub/DataSource/Manage-big-data.htm)
- Dynamic Views: [https://help.qlik.com/en-US/sense/Subsystems/Hub/Content/Sense\\_Hub/DynamicViews/dynamic-views.htm](https://help.qlik.com/en-US/sense/Subsystems/Hub/Content/Sense_Hub/DynamicViews/dynamic-views.htm)

Since both On-Demand App Generation and Dynamic Views on QSECM leverage the same underlying services for processing, this technical brief applies equally to each despite the differing presentation to the end consumer of the techniques.

## What are the performance improvements in May 2021?

In the May 2021 release of Qlik Sense Enterprise Client-Managed (QSECM), the maximum concurrent ODAG requests was increased from 10 to 50. In order to demonstrate the effective result of this increase, a series of benchmark tests were run, comparing fixed volumes of user activity across the February 2021 and May 2021 versions of QSECM. Concurrent user requests were simulated using the

Qlik Sense Enterprise Scalability tool to generate an ODAG application which required a fixed amount of time to reload from source data (60 seconds).

### Moderate Concurrency Test

#### Overview:

To simulate an ODAG app with a moderate amount of concurrency, a test was performed which initiated ODAG requests every 5 seconds for a number of users.

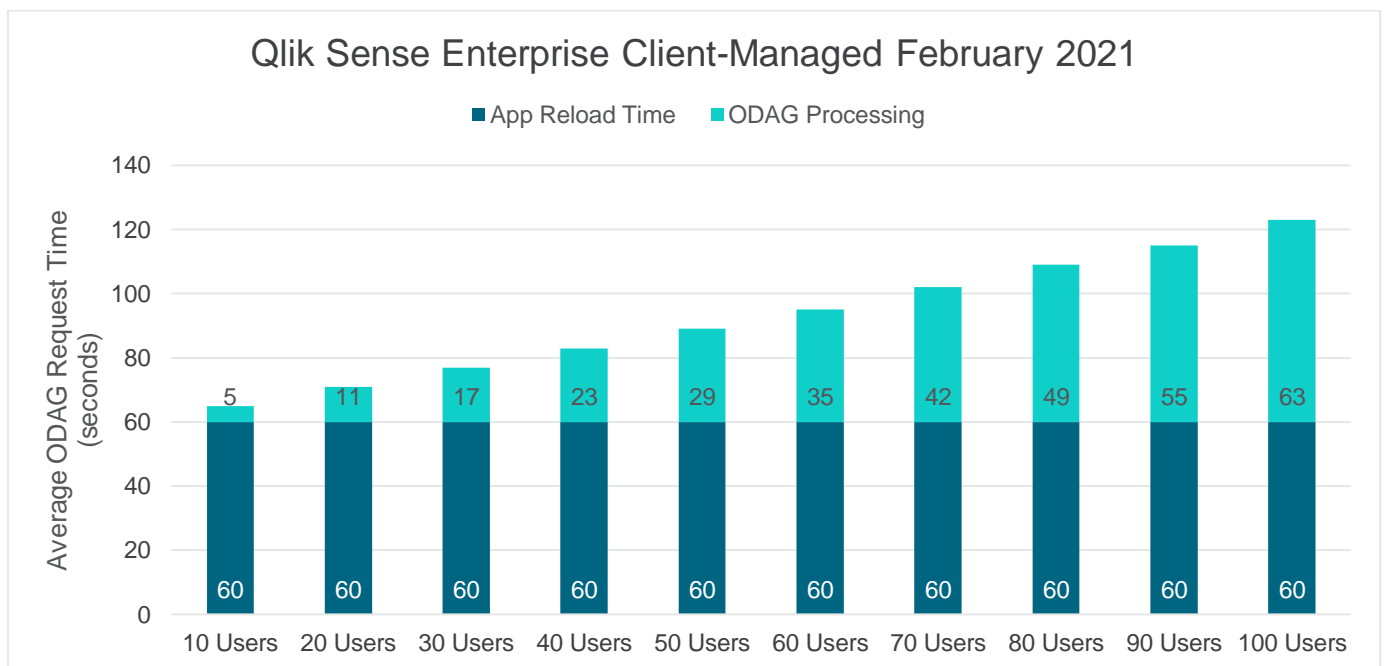
#### Summary:

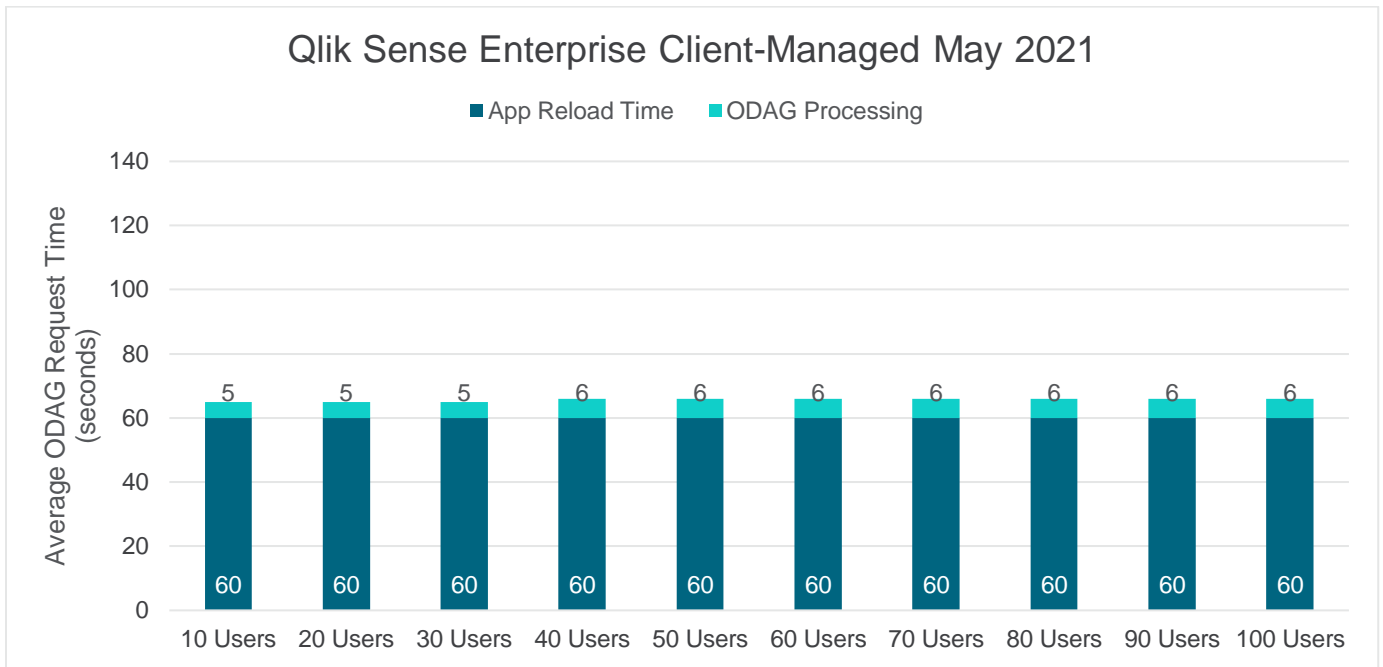
With a moderate concurrency of ODAG requests, May 2021 provides a consistently flat performance curve compared to a linear performance curve on February 2021. Given this improvement, drastically higher user volumes could be supported.

#### Test Environment:

- Server: Azure D32as\_v4 (32 vcpus, 128 GiB memory) / AMD EPYC 7452 processor
- Users: 10-100
- Time between user's ODAG requests: 5 seconds
- Maximum Number of ODAG requests permitted: 10 (February 2021), 50 (May 2021)

#### Results:





**Analysis:**

With the February 2021 (and earlier) release, the 10 maximum number of concurrent ODAG requests slowly became a bottleneck. To take the 80 concurrent user example, 55% of the average wait time for a requests ODAG app was due to the request being processed and queued by the ODAG Service.

With the May 2021 release, there were only minor upticks (1 second) in average ODAG processing in the moderate concurrency tests. Given the higher queue size (10 → 50), the server can support dramatically higher user volumes and provide a reliable user experience.

**High Concurrency Test**

**Overview:**

To simulate an ODAG app with a high amount of concurrency, a test was performed which initiated ODAG requests every 1 second for a number of users. This scenario is intended to simulate an extreme example of deploying Qlik’s hybrid data architecture.

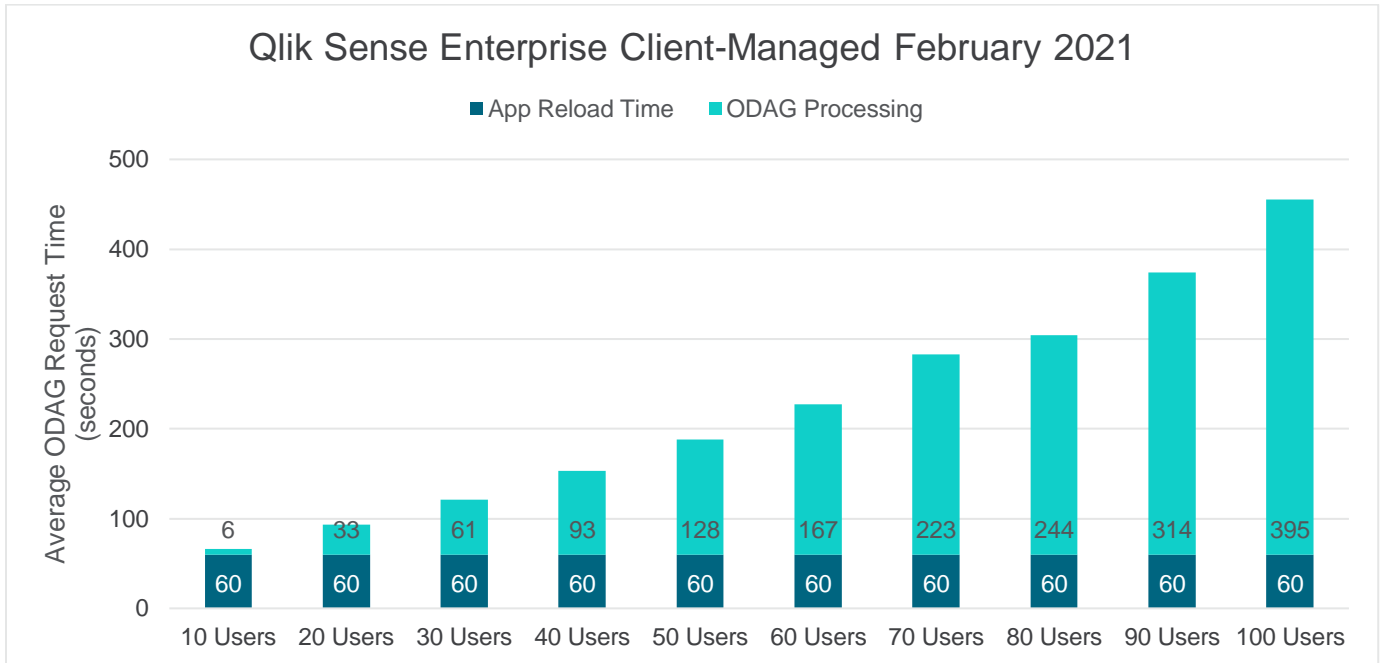
**Summary:**

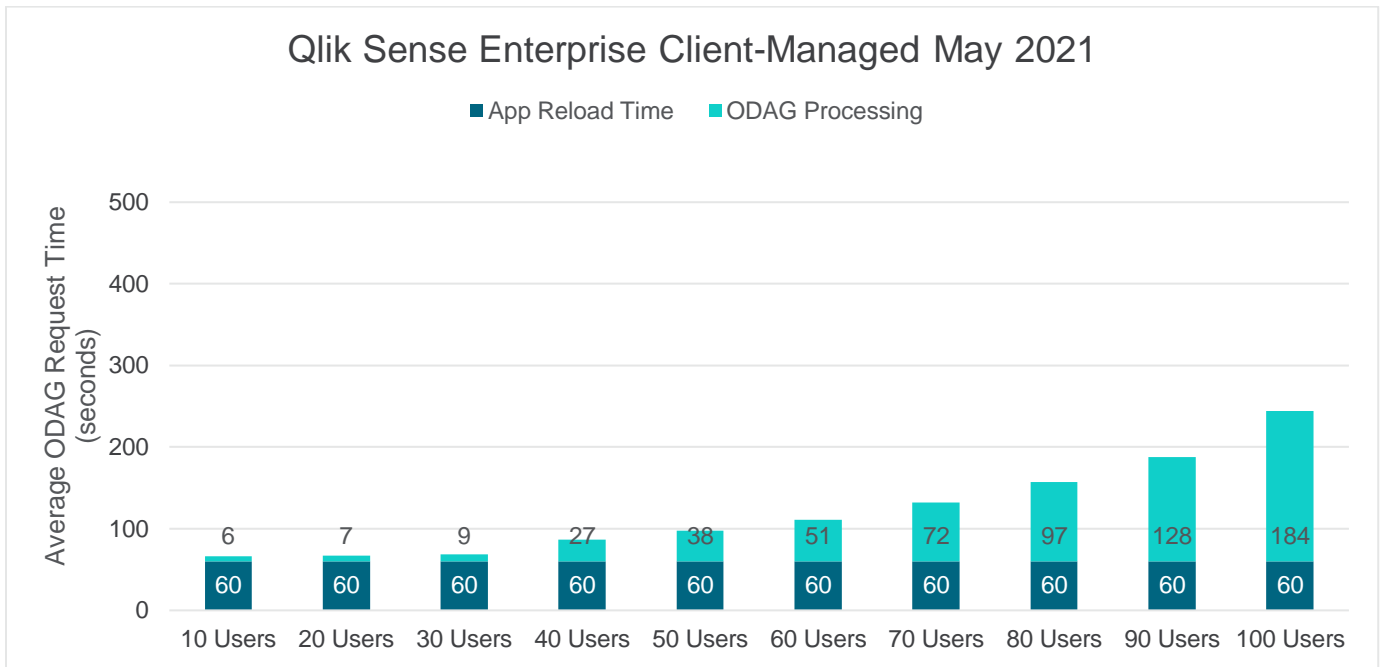
In high concurrency scenarios, wait-time will scale with the volume of concurrent requests. With a high concurrency of ODAG requests, May 2021 provides a dramatically lower performance curve compared to February 2021.

**Test Environment:**

- Server: Azure D32as\_v4 (32 vcpus, 128 GiB memory) / AMD EPYC 7452 processor
- Users: 10-100
- Time between user's ODAG requests: 1 second
- Maximum Number of ODAG requests permitted: 10 (February 2021), 50 (May 2021)

**Results:**





**Analysis:** With the February 2021 (and earlier) release, the 10 maximum number of concurrent ODAG requests rapidly became a bottleneck. To take the 80 concurrent user example, 80% of the wait time for a requests ODAG app was due to the request being processed and queued by the ODAG Service.

With the May 2021 release, wait-time for the ODAG request remained acceptable for a larger number of users. To take the 80 concurrent user example, the wait time was decreased from 244 seconds to 97 seconds; a 60% reduction.

## Performance Tuning ODAG / Dynamic Views

As depicted in the tests above, there are two components to end-user response time when using ODAG or Dynamic Views:

1. Server resources / performance
2. Data Load considerations

This brief will further elaborate specific server architecture considerations in a later section, but for high performance, a Qlik site should use:

- Recommended Chipsets: Reference [Qlik's Recommended Top-Performing Servers technical brief](#).
  - o Higher clock speed will entail faster app reloads
- Servers Tuned to high performance:

- Refer to your server's manufacture (if applicable) for high performance BIOS settings
- Ensure the Windows OS is configured for high performance:  
<https://support.qlik.com/articles/000031649>

For optimization of data loads, use:

- Optimized queries to relational databases
  - Consult administrators for those systems for guidance
- High-performing databases, where available
  - Highly scalable and performant databases like Snowflake, AWS Redshift, Google BigQuery, and Azure Synapse allow for higher scaling than many on-premise databases
- Optimized Loads when using QVDs
  - If necessary generate an `INLINE` table that contains the values passed to the ODAG / Dynamic View apps so that loads can use `WHERE EXISTS` or `WHERE MATCH` (which are optimized) as opposed to `WHERE` (which is unoptimized):

```
// optimized qvd loading

OriginCode:
LOAD * INLINE [
OriginCodeTemp
$(odag_OriginCode){"quote": "", "delimiter": "\n"}
];

// Load with optimized loading from qvd
OriginDetails:
LOAD
  OriginCodeTemp,
  FieldA,
  FieldB,
FROM [lib://NAME/QVDNAME.qvd](qvd) WHERE exists(OriginCodeTemp);

// Load without optimized loading from qvd
OriginDetails:
LOAD
  OriginCodeTemp,
  FieldA,
  FieldB,
FROM [lib://NAME/QVDNAME.qvd](qvd) WHERE OriginCodeTemp = 'ABC';

// → where match() with binding expression example
// where match(OriginCodeTemp, 'ABC', 'XYZ', 'DFG', 'KJL');
```

```
// if needed, drop table
```

- Disable unnecessary capabilities like global search indexing, insight generation and fields on the fly:

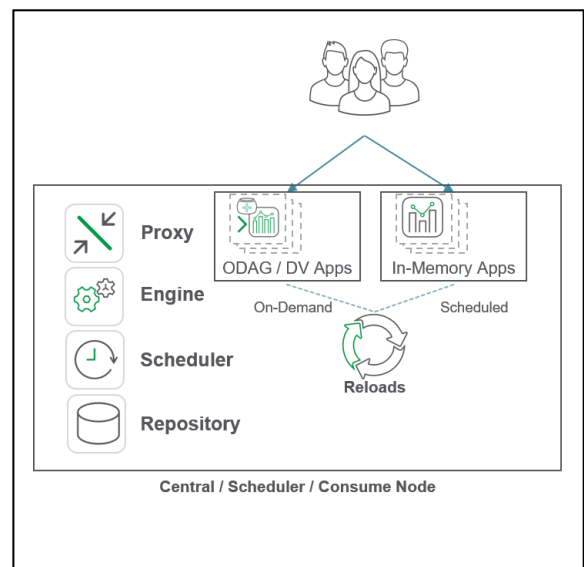
```
SET CreateSearchIndexOnReload=0; // Disable the global search index
SEARCH EXCLUDE *; // Excluding all fields from smart search
SET DISABLE_INSIGHTS = '1'; // Disable insight generation, '1' off, '' on
SET UseAutoFieldOnTheFly=0; // Disable fields on the fly 0 = off
```

## Architectural and deployment considerations

When leveraging ODAG or Dynamic Views, an architect or administrator should consider whether the current server architecture is properly configured to support the desired volumes and whether the existing architecture has sufficient compute resources to support the requested ODAG and/or Dynamic View applications.

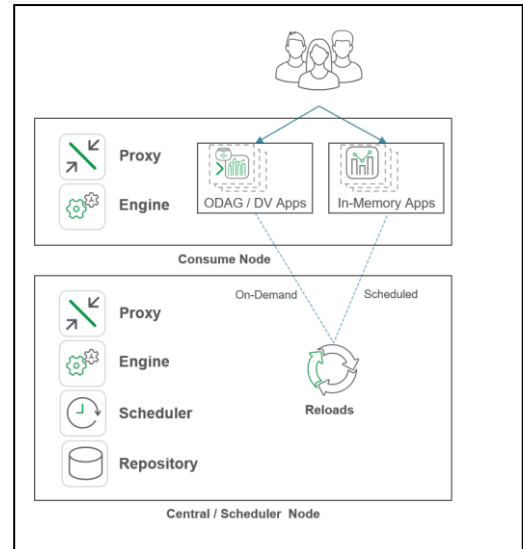
ODAG requests are treated like reload tasks in QSECM, thus the node which performs the reload of the generated application is a Scheduler node.

On single node deployments, this means the node which will perform the ODAG reload will be the same node which delivers end-user consumption activity. Example:





On multi-node deployments, an available slave (deprecated terminology) or worker Scheduler will perform the ODAG reload. Example:



In either deployment topology, sufficient resources need to be available to both adequately support the compute needed to perform reloads as well as to comfortably minimize or eliminate resource contention for end-user consumption activities.

When architecting for higher ODAG concurrencies, separation of user consumption and application reloads is strongly encouraged.

So, as to provide guidance for higher concurrency environments, a series of tests were run to benchmark performance curves on servers of varying compute. The shape of the results are not unique to the May 2021 release and are equally applicable to prior versions of QSECM. For deployments which do not need to support higher concurrencies, the following section can be skipped.

### Compute Requirement Test

**Overview:** To provide guidance on required compute by concurrency, a series of high concurrency tests were run using single-node servers which varied by available vCPUs.

#### Summary:

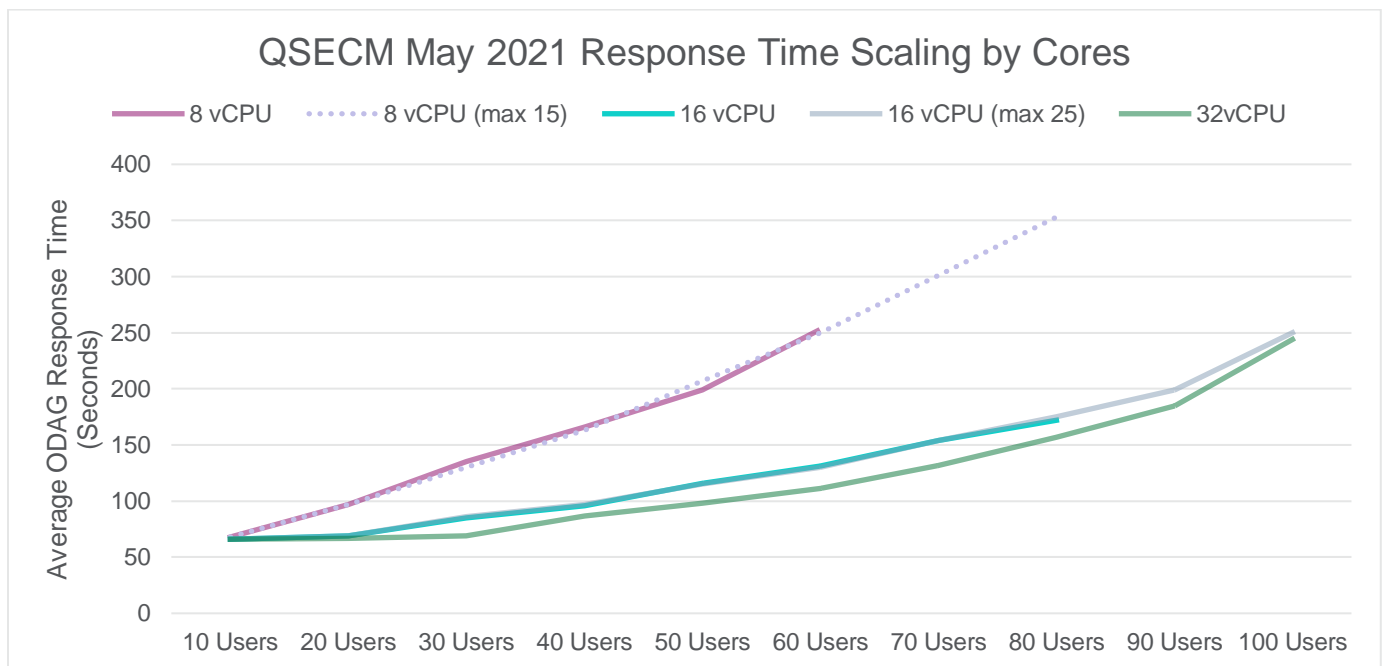
While high concurrencies can be supported on low compute deployments with the May 2021, sufficient available compute resources are required to provide an acceptable end-user experience and reliable performance curve. For lower compute deployments, it is recommended to lower the maximum number of concurrent ODAG requests to a value less than the absolute maximum (50).

#### Test Environment:

- Servers:

- 8 vCPU test: Azure D8as\_v4 (8 vcpus, 32 GiB memory) / AMD EPYC 7452 processor
  - 16 vCPU test: Azure D16as\_v4 (16 vcpus, 64 GiB memory) / AMD EPYC 7452 processor
  - 32 vCPU test: Azure D32as\_v4 (32 vcpus, 128 GiB memory) / AMD EPYC 7452 processor
- Users: 10-100
  - Time between user's ODAG requests: 1 second
  - Maximum Number of ODAG requests permitted: 15, 25, 50

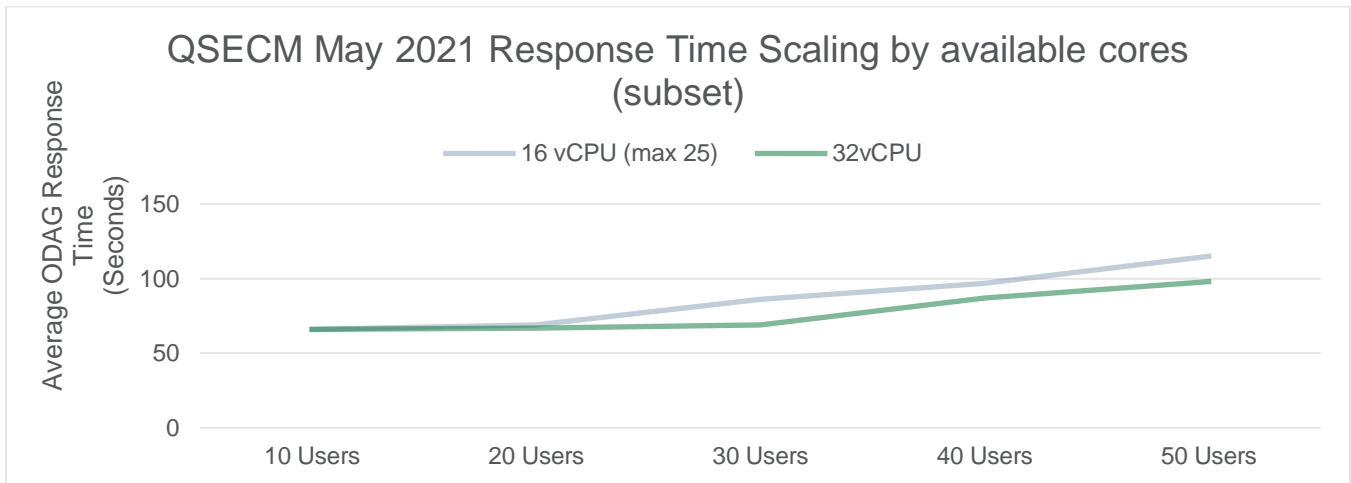
**Results:**



*Maximum number of concurrent ODAG requests enabled (50) unless stated otherwise; 1 second delay between requests; data points missing on 8 and 16 vCPU tests due to over-saturation of server*

**Analysis:** As shown, lower compute capacity will adequately support lower volumes of user requests. On the 8 and 16 vCPU tests with 50 maximum concurrent ODAG requests, higher volumes over-saturated the site such that some requested ODAG applications failed to be generated. With these lower compute sites, further reliable load could be attained by adjusting the maximum concurrent ODAG requests setting to a lower value.

When architecting a site for higher ODAG volume, careful consideration of the available vCPUs on Scheduler node(s) is encouraged to deliver consistently high volumes of ODAG requests. To highlight the performance curve, we can view a subset of the prior graph:



In this subset, we can see virtually flat performance curves so long as there is at least 1 vCPU available per user / ODAG request. Since ODAG requests generate an application reload, an architect should align the maximum number concurrent ODAG requests with available compute resources on Scheduler node(s) in their deployment.

To determine an appropriate setting for a deployment, an architect should reference the following formula as a rule of thumb:

$$(((Total\ vCPUs\ on\ Scheduler\ Nodes) - (Number\ of\ Scheduler\ Nodes * 2)) - (Average\ Concurrent\ Reloads\ performed\ per\ hour\ during\ business\ hours))$$

For a 2 Scheduler node deployment with 16 vCPUs and 12 hourly reloads, the calculation would be  $((16 * 2) - (2 * 2)) - 12 = 16$ . This rule of thumb provides an estimate of the number of available cores for ODAG reloads during any given business hour. Further analysis, tuning, and testing is encouraged prior to determining a compute architecture to support a deployment.

## Conclusion

- Qlik Sense Enterprise Client-Managed May 2021 has increased the maximum number of concurrent ODAG requests from 10 to 50.
- With this increase, higher user volumes can be supported. Which expands the use cases where ODAG or Dynamic Views can be deployed with consistent performance curves.

- As with prior versions of Qlik Sense Enterprise Client-Managed, optimization should occur to lower application reload times.
- 1 available CPU or vCPU per ODAG / Dynamic View request is recommended in high concurrency scenarios.



### About Qlik

Qlik's vision is a data-literate world, where everyone can use data and analytics to improve decision-making and solve their most challenging problems. Qlik provides an end-to-end, real-time data integration and analytics cloud platform to close the gaps between data, insights and action. By transforming data into active intelligence, businesses can drive better decisions, improve revenue and profitability, and optimize customer relationships. Qlik does business in more than 100 countries and serves over 50,000 customers around the world.

[qlik.com](https://www.qlik.com)