



Performance Benchmark

OnPremises and Private Cloud

AgilePoint NX v9.0, Software Update 1

January 16, 2026

Contents

- Legal Statements and Policies..... 4
- Performance Benchmark 6
 - Primary Objective..... 6
- Test Environment..... 7
 - AgilePoint Server Machine 7
 - Database Server 7
 - Client machine 7
- Scope for AgilePoint Server Performance..... 9
- Architecture 10
 - AgilePoint Server..... 10
 - Database Server 10
 - App Builder..... 10
- Impact Analysis on AgilePoint Server 11
 - Objective 11
 - Test Criterion..... 11
 - Test Method 11
 - Acceptable Performance 12
- Results 13
 - Results for time taken to complete 5,000 instances initiated in parallel 13

Results for average time taken to complete 1 activity 14

Contributing Factors 16

Test Process 17

Scope for AgilePoint API Performance 18

Impact analysis on AgilePoint API 19

 Test Method 19

 Test Criteria 19

 Test Result 20

 Contributing Factors 29

Impact Analysis on AgilePoint NX Portal 30

 Objective 30

 Test Method 30

 Test Criterion..... 30

 Test Results..... 31

 Contributing Factors 34

Legal Statements and Policies

This section provides legal statements and general policies for AgilePoint software and documentation.

Disclaimer of Warranty

AgilePoint, Inc. makes no representations or warranties, either express or implied, by or with respect to anything in this document, and shall not be liable for any implied warranties of merchantability or fitness for a particular purpose or for any indirect, special or consequential damages.

Copyright and Trademarks

© 2026 AgilePoint, Inc. All rights reserved.

AgilePoint is a registered trademark of AgilePoint, Inc. AgilePoint's products, including **NX**, are trademarks of AgilePoint, Inc. References to other companies and their products use trademarks owned by the respective companies and are for reference purpose only.

Government Rights Legend

Use, duplication or disclosure by the U.S. Government is subject to restrictions set forth in the applicable license agreement and as provided in DFARS 227.7202-1(a) and 227.7202-3(a) (1995), DFARS 252.227-7013(c)(1)(ii) (Oct 1988), FAR 12.212(a) (1995), FAR 52.227-19, or FAR 52.227-14, as applicable.

Virus-Free Software Policy

AgilePoint recognizes that viruses are a significant security consideration for our customers. AgilePoint takes the following measures to make sure our software is free of viruses upon delivery:

- AgilePoint is built on top of Microsoft .NET framework. The pre-compiled executable is a .NET Common Language Runtime (CLR) application, not a native machine binary. As far as is known at this time, there are no viruses that infect .NET CLR executables.
- The virtual environment for the product packaging process is fully isolated and protected, and anti-virus software is installed and running during packaging.
- The compiled and packaged software files are scanned by virus scanning software before they are released. An official disclosure document regarding the findings from our virus scanning activities is available upon request.

External Links in Documentation

AgilePoint, Inc. user resources, such as Product Documentation and community forums, sometimes include links or URLs that AgilePoint does not control. Third-party web sites are referenced for convenience only. AgilePoint has no control over these external web sites, so we cannot make any representations about their accuracy or guarantee their security, such as the potential presence of malware. You assume any and all risks associated with visiting non-AgilePoint URLs.

Performance Benchmark

The purpose of this document is to help businesses conduct a Benchmark Analysis on AgilePoint NX performance under load. This document provides guidance on how to make an assessment of the impact on AgilePoint NX performance based on different factors like:

- Number of process instances running concurrently.
- Number of users concurrently making an API call to AgilePoint NX.
- Performance analysis of the module to login to AgilePoint NX Portal.
- Render the task list.
- Render the task forms under load.

Primary Objective

The primary objective of the AgilePoint NX Benchmark Analysis is to provide stakeholders with information that supports an acceptable level of AgilePoint NX performance given the AgilePoint NX suite is installed.

Performance has been carried out under 3 categories:

- AgilePoint Server performance
- AgilePoint API performance
- AgilePoint NX Portal UI performance

Test Environment

The following hardware and software specifications were used to test the application. The hardware and software specifications are representative of a typical deployment.

AgilePoint Server Machine

- **Processor** - AMD EPYC-Genoa Processor, 3.25 GHz with 8 GB RAM, running on Vultr.
- **Operating system** - Windows Server 2022 Standard.
- **AgilePoint software installed** - AgilePoint NX v9.0 SU1
- **AgilePoint Server configuration** - AgilePoint Server thread capacity set to 50.
- **Other software installed** - .NET 4.8, Node.js.

Database Server

- **Processor** - AMD EPYC-Rome Processor 2.00 GHz with 8 GB RAM, running on Vultr.
- **Software installed** - Windows Server 2022 Standard, Microsoft SQL Server 2019 (RTM) - 15.0.2000.5 (X64)
- **Network Speed**: Ethernet 1 GB/s

Client machine

- **Processor** - Intel(R) Xeon(R) Platinum 8370C CPU @ 2.80GHz, 2.79 GHz with 16 GB RAM. running on Azure.

- **Operating System** - Windows Server 2022 Datacenter.
- **Browsers** - Microsoft Edge Version 131.0.2903.63 (64-bit), Google Chrome 131.0.6778.86 (64-bit), Mozilla Firefox 132.0.2 (64-bit).

Scope for AgilePoint Server Performance

The scope of the performance testing is to determine the following:

- **Time taken to complete 5,000 process instances of a workflow that has 8 eForms, 6 web services, 18 Database activities, 1 Condition activity, 1 Or activity, 1 Start activity, and 1 Stop activity.**

The started time of the first process instance and completed time of the 5,000 (last) process instance is calculated. The difference gives us the time taken to complete 5,000 process instances.

- **Average time taken to execute a single activity.**

The time difference between process instance started date and completed date is calculated for each of the 50 process instances. The average is then calculated. This is then divided by 36 (including the Start and Stop process activities) to get the average time taken to execute a single activity.

- **Number of Process Instances that can be completed in 1 hr.**

A large number of process instances are initiated. The number of process instances completed in 1 hr is calculated.

Architecture

This section describes the main software components used during the performance benchmark test.

AgilePoint Server

AgilePoint Server is the main server-side component for AgilePoint NX. The core technology for AgilePoint Server is the process engine, sometimes called the workflow engine. This engine runs process models that are created in the Process Builder in the AgilePoint NX Portal. Process models are, in turn, a composite of XML and JSON files. Built on Microsoft .NET, the workflow engine runs these process models as executable applications.

Database Server

The machine where Microsoft SQL Server is installed to support AgilePoint NX. This database stores the data required for AgilePoint NX.

App Builder

AgilePoint NX App Builder is the overall component that is used to create, manage, and deploy applications in AgilePoint NX. The App Builder includes the Process Builder and eForm Builder components.

Impact Analysis on AgilePoint Server

The following impact analysis results are based on activities performed in conjunction with initiating a huge number of process instances of a workflow that has 8 eForm, 6 Web Services 18 Database, 1 Condition, 1 Or and 2 Start and Stop process activities.

Objective

The objective of this test case is to measure the impact on the time taken to initiate and complete a very large number of process instances for a process (workflow) that has 8 eForm, 6 Web Service, 18 Database, 1 Condition, 1 Or and 2 Start and Stop process activities.

Test Criterion

The results are based on a test conducted by initiating a large number of process instances of a workflow that has 8 eForms, 6 Web Service, 18 Database, 1 Condition, 1 Or and 2 Start and Stop process activities. For example, in the sample test, the results are based on 5,000 process instances. Time will be measured for completing all 5,000 process instances.

Test Method

AgilePoint has developed a tool called Process Initiator to initiate process instances of a workflow processes in bulk. The test measures the time taken for completing all 5,000 process instances, number of process instances in 1 hr, and the average time taken to complete a single activity.

The workflow has been deployed to AgilePoint Server using AgilePoint NX Process Designer.

Acceptable Performance

A time of up to 1-2 hr is an acceptable time for completing 5,000 process instances. An average time of 1.5 s per activity is acceptable.

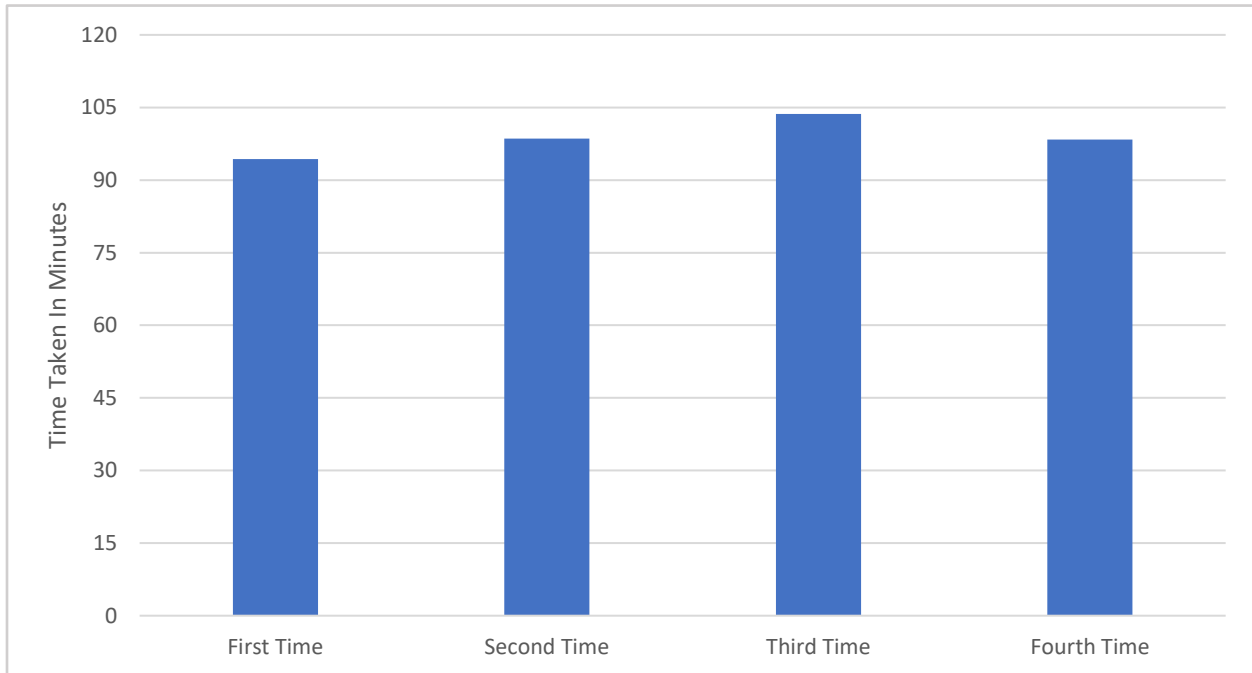
Results

Results for time taken to complete 5,000 instances initiated in parallel

* Number of test process instances initiated = 5,000

Number of Process Instances	Start Time of First Process Instance	End Time of Last Process Instance	Total Time to Complete 5,000 Process Instances
5,000	2024-11-13 02:41:59.937	2024-11-13 04:16:19.120	1hr 34 min 20 s
5,000	2024-11-13 04:27:13.837	2024-11-13 06:05:49.540	1hr 38 min 36 s
5,000	2024-11-13 06:08:31.447	2024-11-13 07:52:09.447	1hr 43 min 38 s
5,000	2024-11-14 20:42:24.640	2024-11-14 22:20:46.667	1hr 38 min 22 s

5,000 Parallel Instances



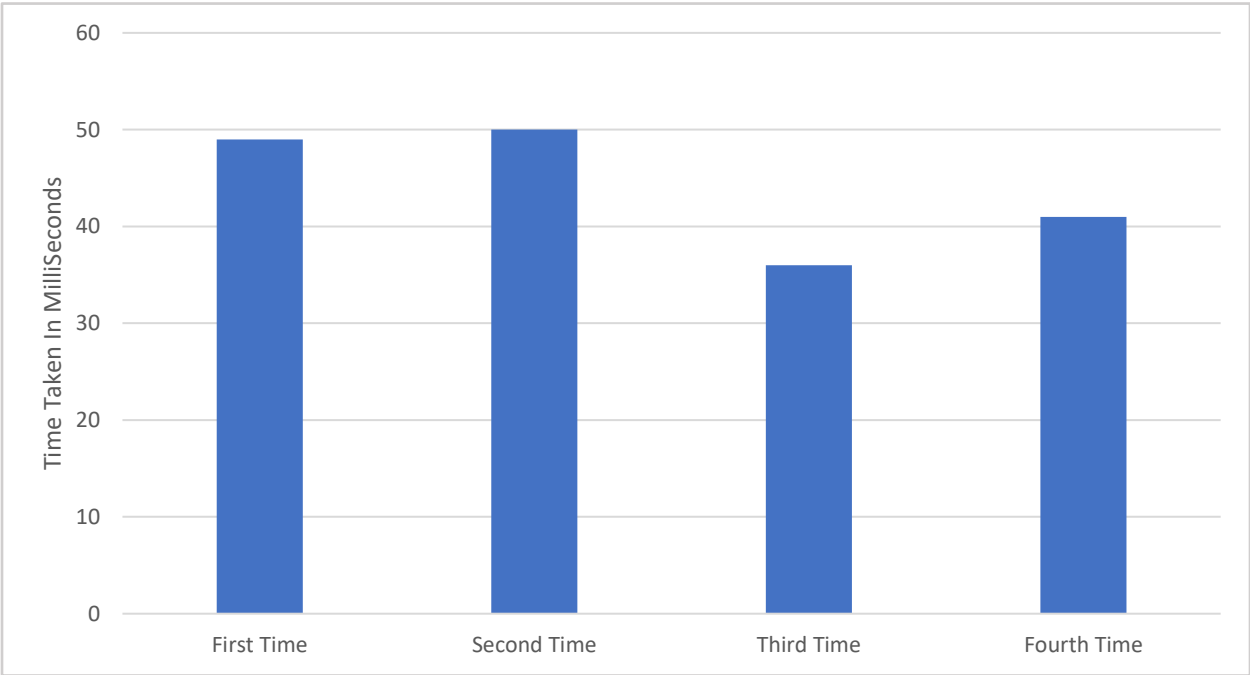
Results for average time taken to complete 1 activity

To calculate the average time per activity under reasonable load, the number of process instances initiated at same time has been kept to 50.

*** Number of process instances initiated = 50**

Number of Process Instances	Average Time Per Activity (In Milliseconds)
50	049 ms
50	050 ms
50	036 ms
50	041 ms

Time to Complete 1 Activity



Contributing Factors

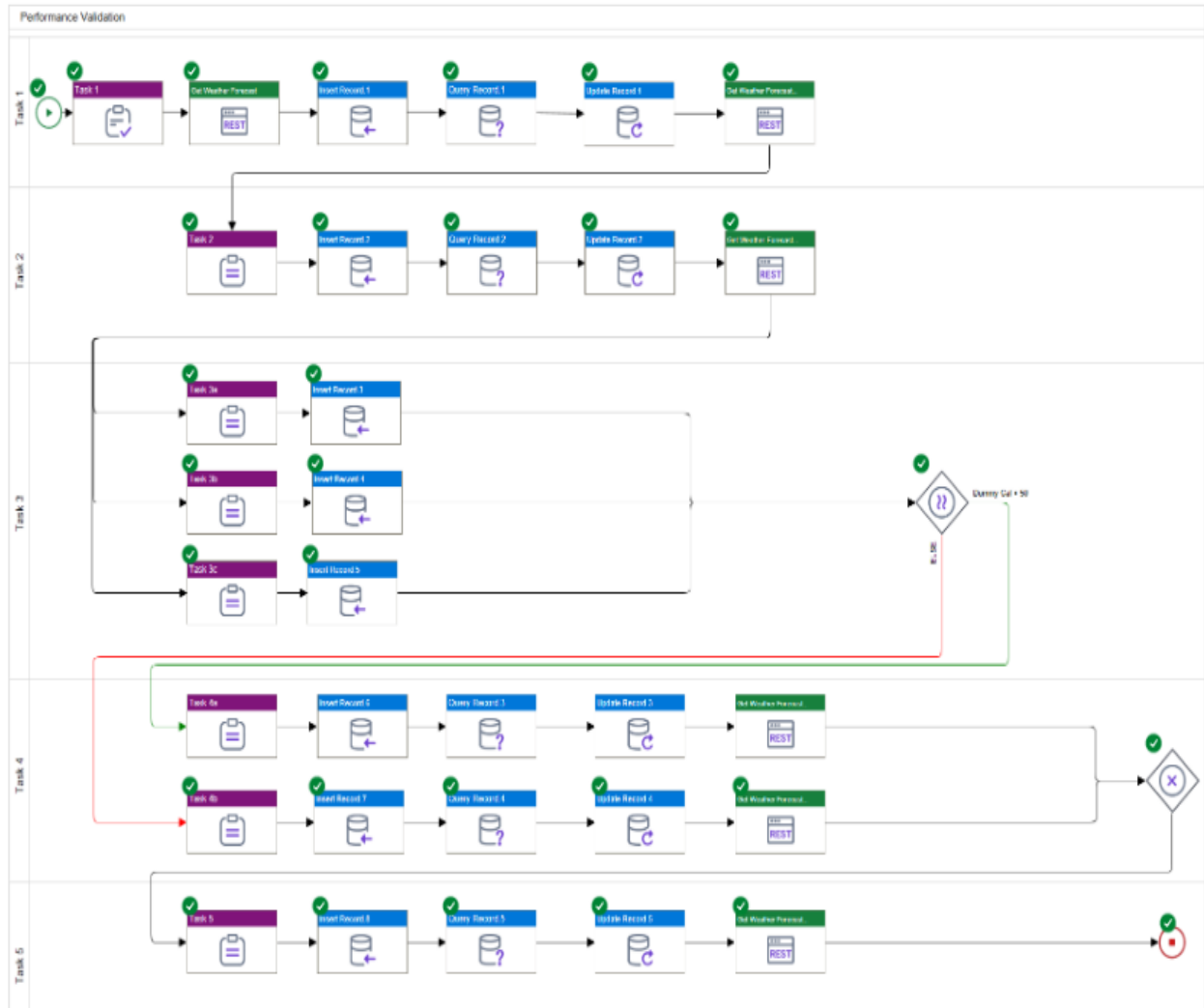
The results mentioned here are under simulated conditions as described throughout this document. Performance may vary based of the following factors:

- Hardware specifications for the systems used in a particular test.
- Network Bandwidth.
- Usage pattern (i.e., is the load distributed through the day and occurs in a random way or is it a frequent burst of high activity).
- This test has been performed in a virtual environment. Physical environments may result in better performance.

Test Process

This image shows the process model that is used for the performance benchmark test.

Performance Validation



Scope for AgilePoint API Performance

The scope of AgilePoint API performance is to determine the following:

1. Average time taken to execute AgilePoint APIs in a case where 25 concurrent users are making simultaneous API calls to a server with a load of 500 process instance running. APIs, which are frequently used from the Work Center and form rendering module, are considered for API performance testing.
2. Average time taken to execute a few of APIs in case of 1, 10, 25, 50 and 100 concurrent users.
3. API performance as the volume of the process instances in the database grows from 50,000 to 1,000,000.

Impact analysis on AgilePoint API

In this analysis, API performance is observed on how it varies along with number of process instances in database grows from 50,000 to 5,000,000.

The analysis is based on the set of APIs, which are frequently called from the AgilePoint NX Portal like Work Center, Manage Center, and for form rendering. API calls are simulated to mock the real case scenario where the server receives a number of requests concurrently and experiences load. Average time taken by an API is been calculated by considering time for successful request, time for AgilePoint Server to prepare the response, and time for receiving a complete response.

Next, API performance is analyzed based comparisons between the number of users making concurrent API calls, starting from 1 User and further sampling at 10, 25, 50 and 100 users with 500 running process instances.

Test Method

To do the multiple concurrent API performance testing, JMeter performance analysis testing tool is been used to capture the average time taken for making API calls.

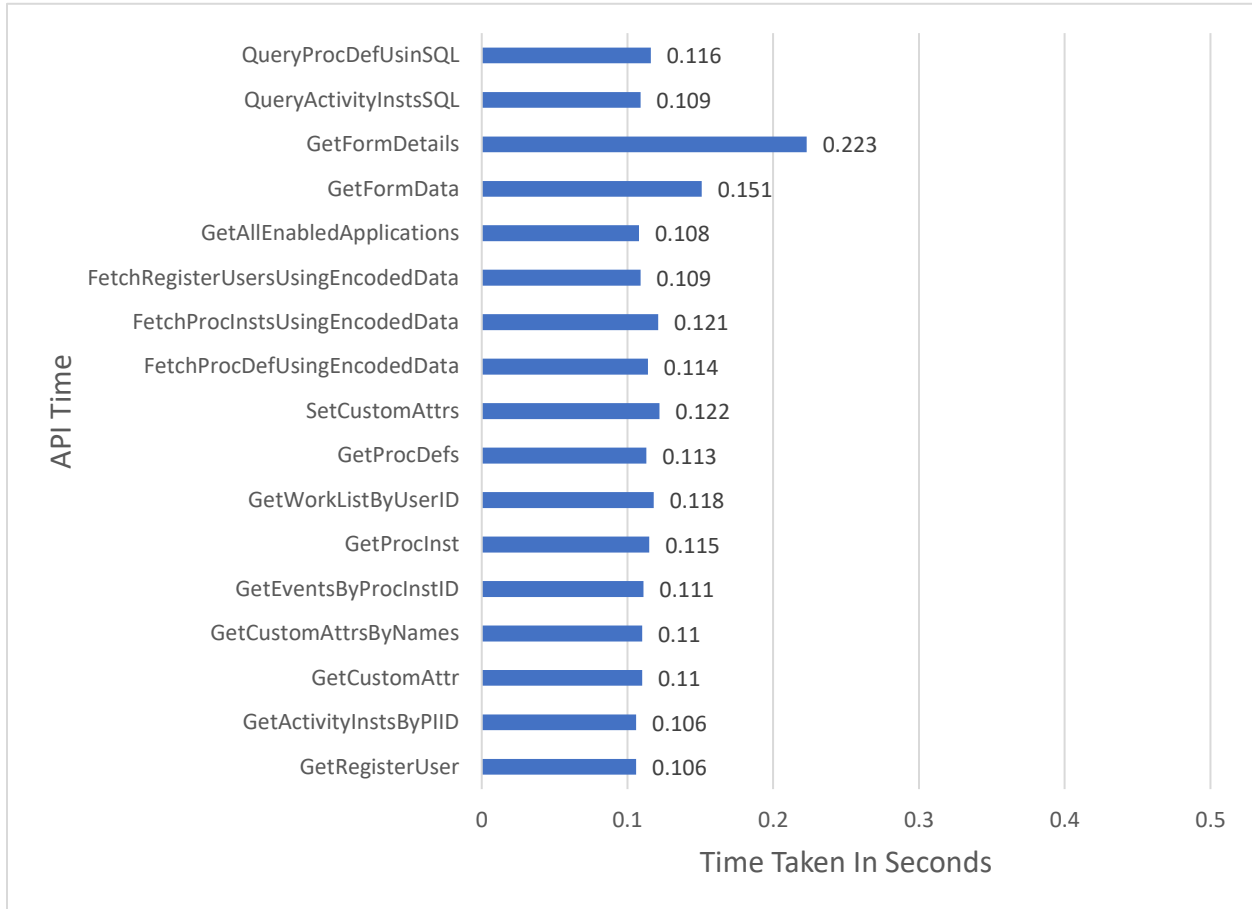
The Apache JMeter application is open source software, a 100% pure Java application designed to load test functional behavior and measure performance.

Test Criteria

The results are based on the JMeter reports, which provides average time taken by each API call based on number of concurrent users and variation in the API performance as the process instances grows.

Test Result

Average API time taken for 25 users



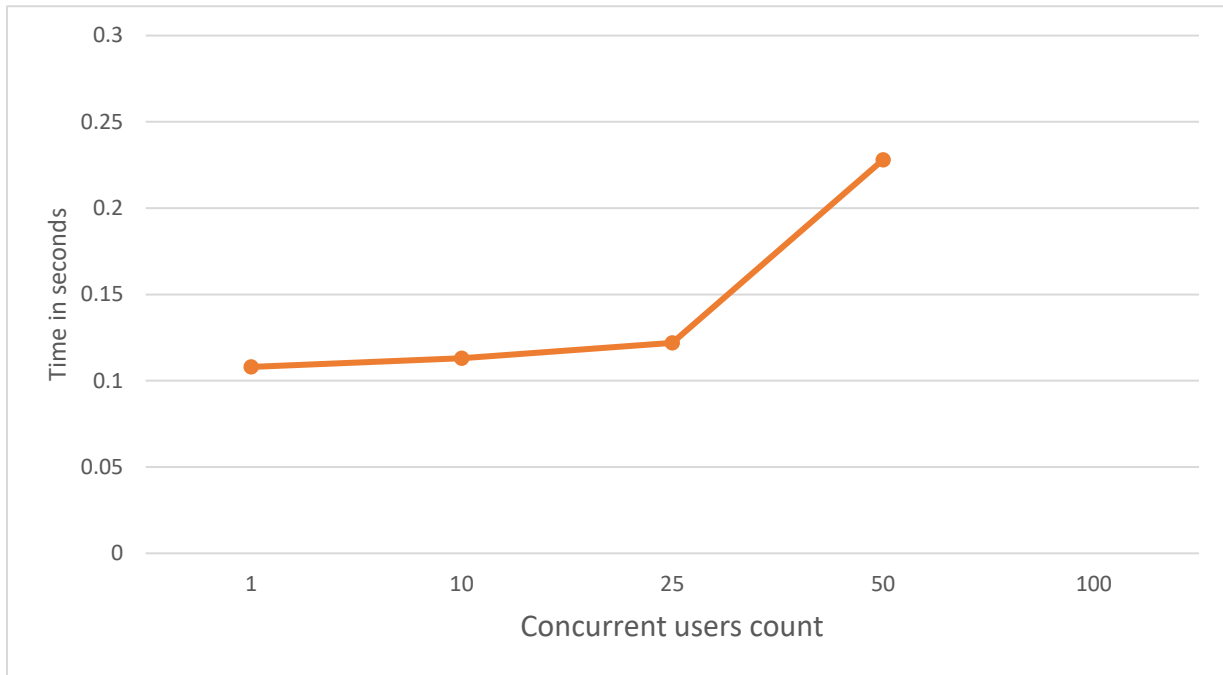
API Name	API Description	Average in Seconds
GetRegisterUser	Retrieve AgilePoint registered user details	0.106
GetActivityInstsByPIID	Get complete activity instance details based on process instance ID	0.106

API Name	API Description	Average in Seconds
GetCustomAttr	Retrieve workflow data for a single process instance	0.110
GetCustomAttrsByNames	Retrieve process variable for multiple process instances	0.110
GetEventsByProcInstID	Get all the events associated with a process instance	0.111
GetProcInst	Get process instance details based on the process instance ID	0.115
GetWorkListByUserID	Query to get all the manual work items/tasks for a specified user	0.118
GetProcDefs	Get all process models	0.113
SetCustomAttrs	Update workflow data for single process instance	0.122
FetchProcDefUsingEncodedData	Retrieve process model data based on filter criteria	0.114
FetchProcInstsUsingEncodedData	Retrieve process instance data based on filter criteria	0.121
FetchRegisterUsersUsingEncodedData	Retrieve AgilePoint registered user details based on filter criteria	0.109

API Name	API Description	Average in Seconds
GetAllEnabledApplications	Retrieve all application metadata details	0.108
GetFormData	Get the required data for rendering the eForms	0.151
GetFormDetails	Get eForms related files details for the current form rendered	0.223
QueryActivityInstsSQL	Retrieve activity instance data based on filter criteria	0.109
QueryProcDefUsingSQL	Retrieve process model data based on SQL query	0.116

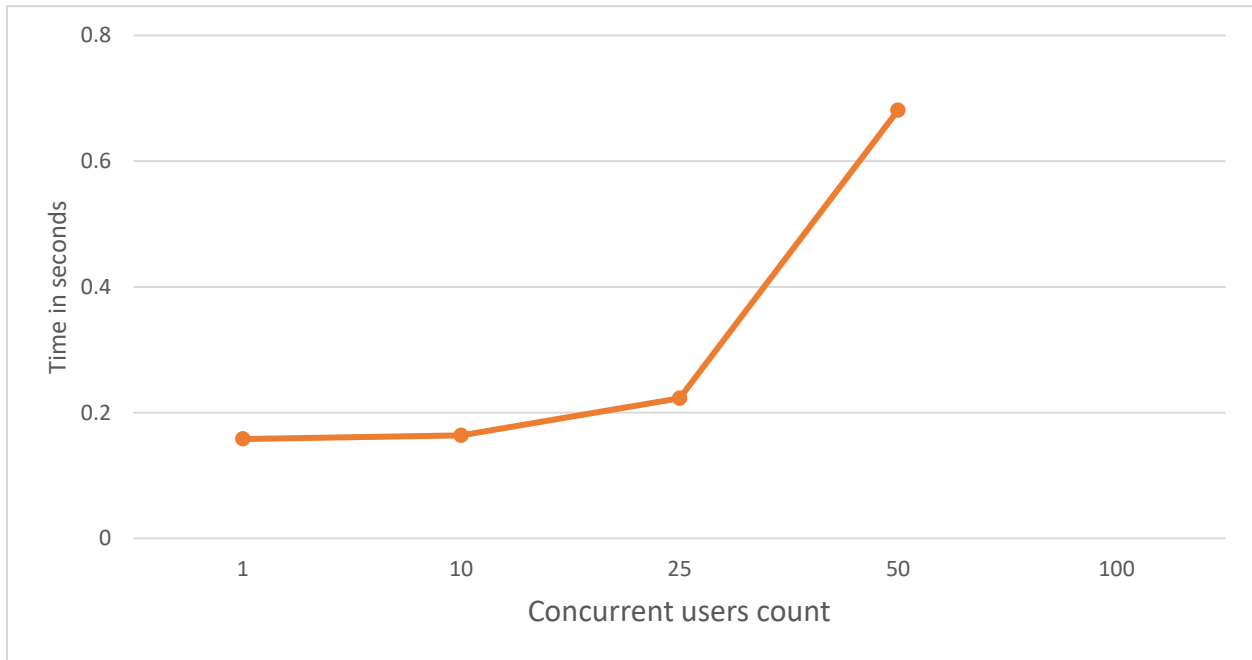
Growth in Average API time taken as the number of concurrent users increases.

FetchProcInstsUsingEncodedData: Retrieve process instance data based on filter criteria



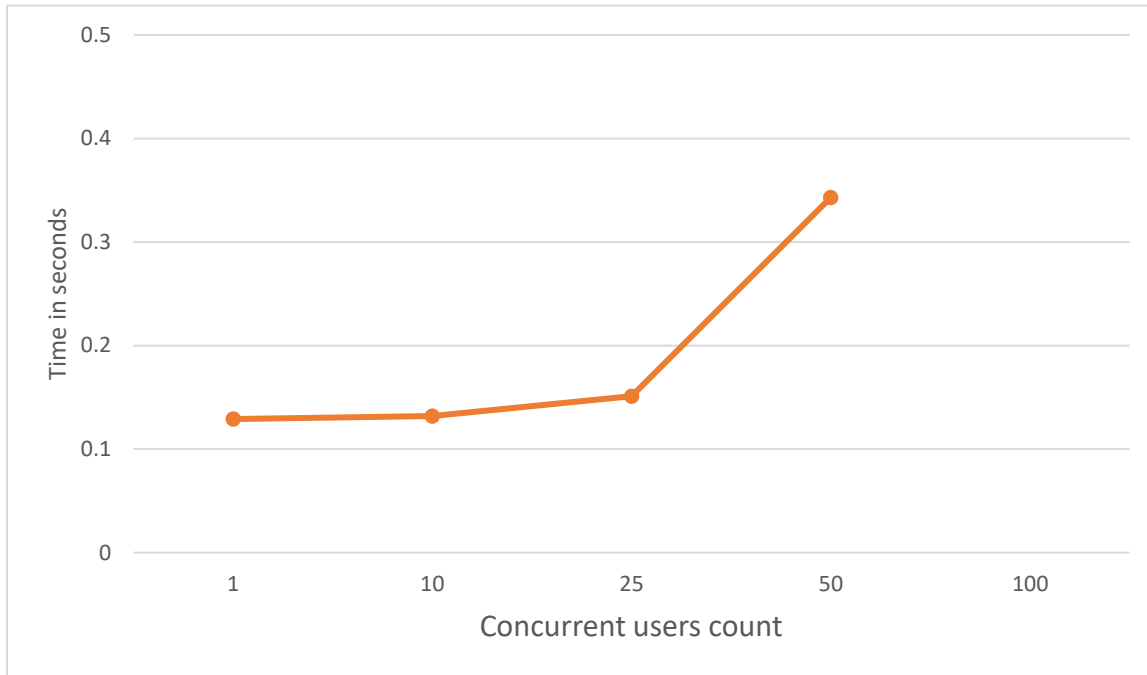
User count	Average time (seconds)
1	0.108
10	0.113
25	0.122
50	0.228

GetFormDetails API: Get eForms related files details for the current form rendered



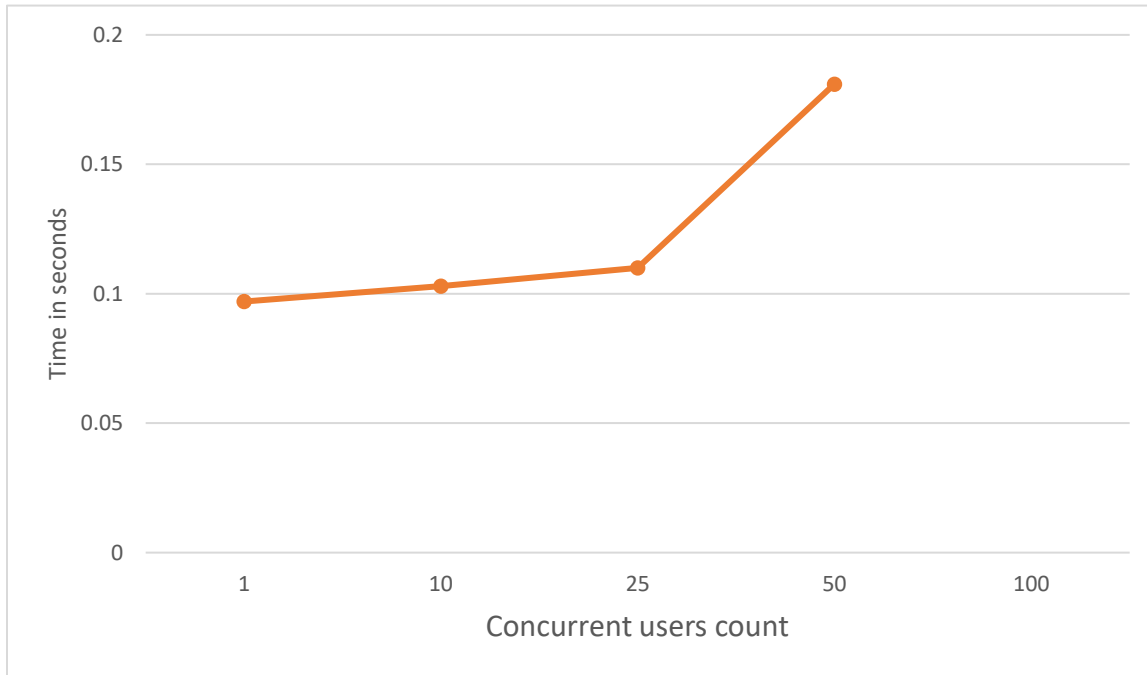
User count	Average time (seconds)
1	0.158
10	0.164
25	0.223
50	0.681

GetFormData: Get the required data for rendering the eForms



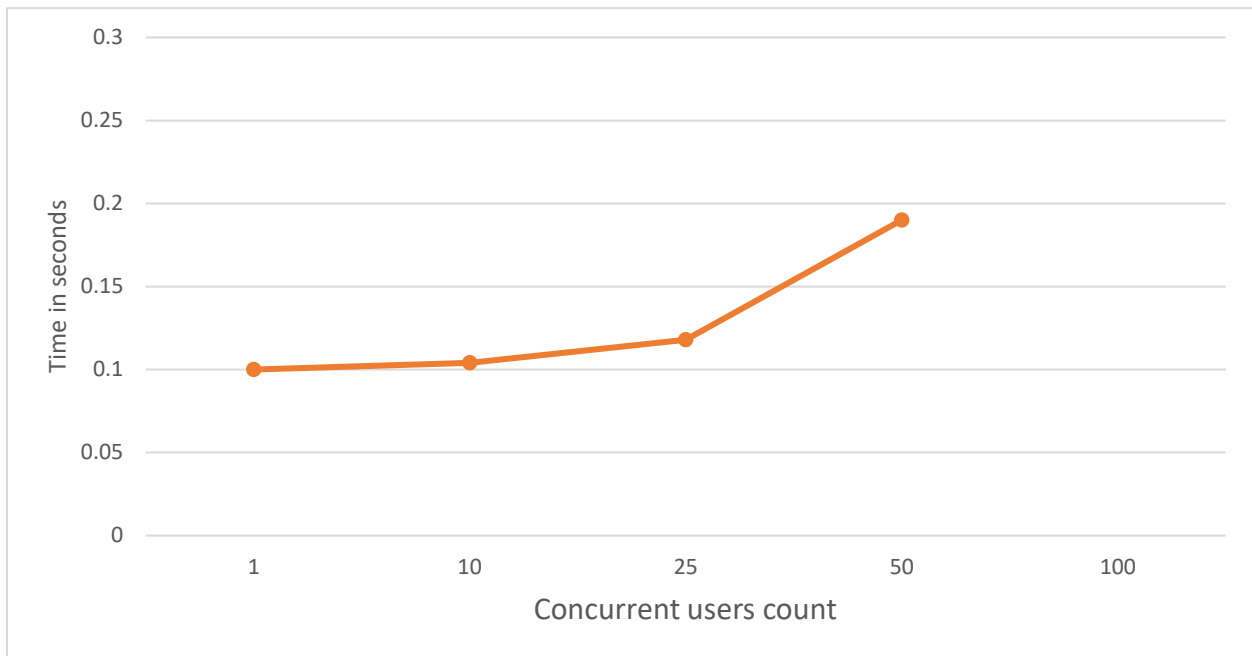
User count	Average time (seconds)
1	0.129
10	0.132
25	0.151
50	0.343

GetCustomAttr: Retrieve workflow data of single process instance



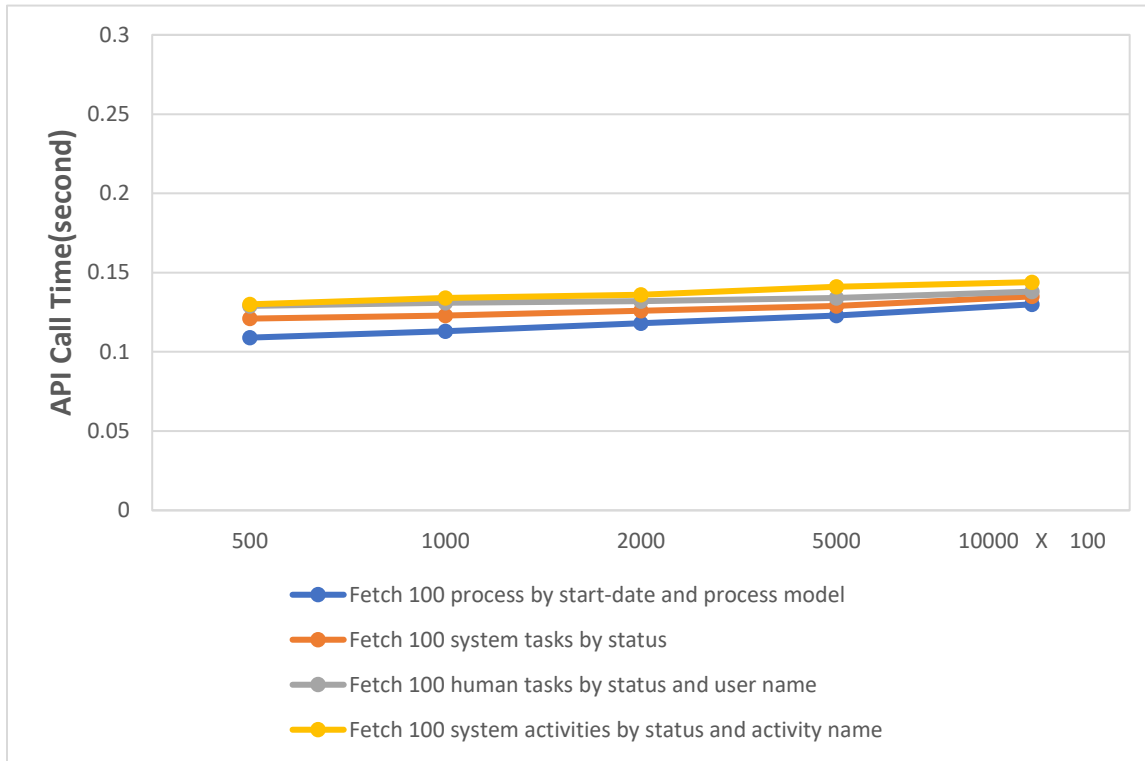
User count	Average time (seconds)
1	0.097
10	0.103
25	0.110
50	0.181

GetWorkListByUserID: Query to get all the manual work items/tasks for a specified user



User count	Average time (seconds)
1	0.100
10	0.104
25	0.118
50	0.190

Growth in Average API time as the number of process instance increases



Total Number of process in Database	50,000	100,000	200,000	500,000	1,000,000
Fetch 100 process by start-date and process model	0.109	0.113	0.118	0.123	0.130
Fetch 100 system activities by status and activity name	0.130	0.134	0.136	0.141	0.144
Fetch 100 system tasks by status	0.121	0.123	0.126	0.129	0.135
Fetch 100 human tasks by status and user name	0.129	0.131	0.132	0.134	0.138

Contributing Factors

The results mentioned here are under simulated conditions as described throughout this document. Performance may vary based of the following factors:

- Hardware specifications for the systems where the test is conducted.
- Network bandwidth.
- Usage pattern — i.e., is the load distributed through the day, and does it occur in a random way, or is it a frequent burst of high activity.
- If more users are accessing system, it is required to add more servers into Network Load balancing.
- It is required to increase the AgilePoint Server thread count as the load increases for better performance.

Impact Analysis on AgilePoint NX Portal

The following impact analysis results are based on activities performed in conjunction with loading the Sign In page, Home page, Work Center page and eForms on various browsers like Internet Explorer, Google Chrome, and Firefox.

Objective

The objective of this test case is to measure the time taken for the Sign In page, Home page, Work Center page with 10, 20, 30 and 40 tasks, eForms with 10, 20, 30 and 40 form controls to load completely in the various browsers like Internet Explorer, Google Chrome, and Firefox.

Test Method

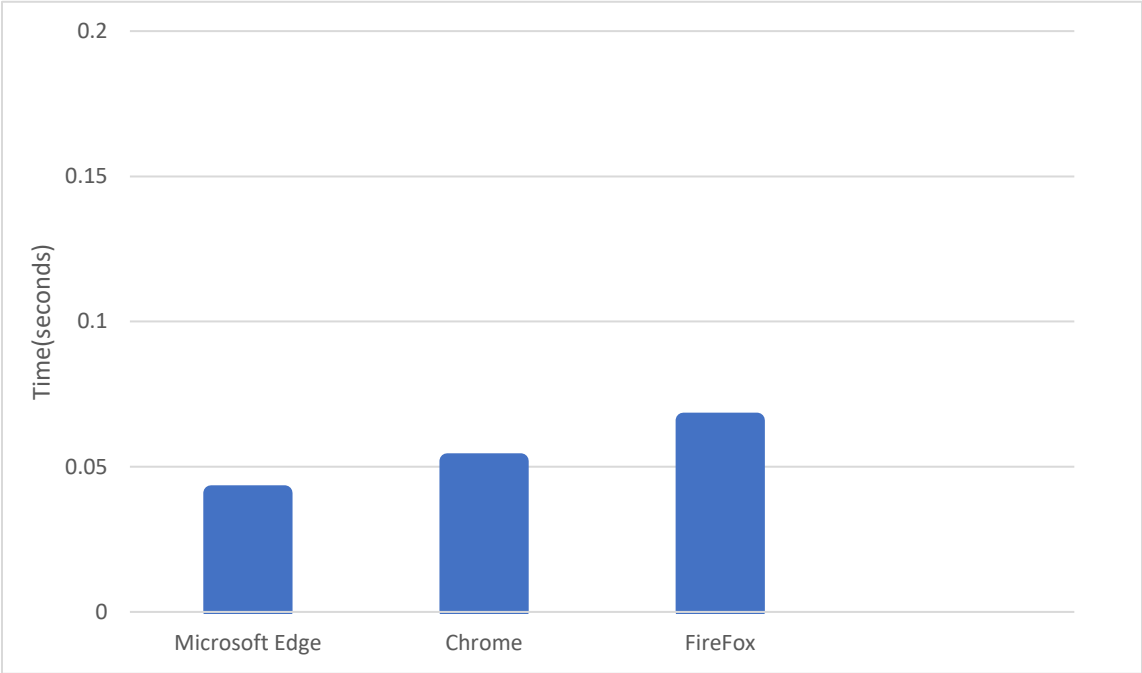
Execution time was captured using the browser developer tool. The developer tool is kept open when each of the page load time is captured.

Test Criterion

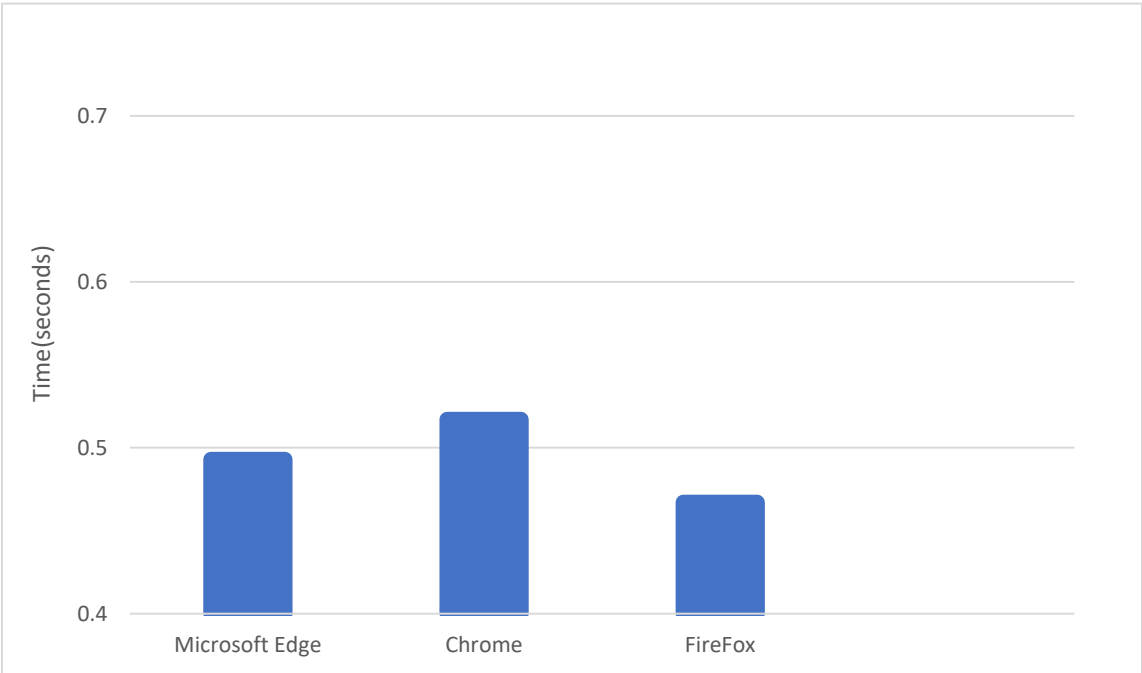
Each page load time was captured 10 times. The average page load time is considered for the test result. Tests use a single user because the page execution happens on the client side. It is not required to check the concurrent user performance because page rendering uses the client browser's processing power, rather than server-side rendering. Concurrent user performance is required for AgilePoint APIs, which is analyzed elsewhere.

Test Results

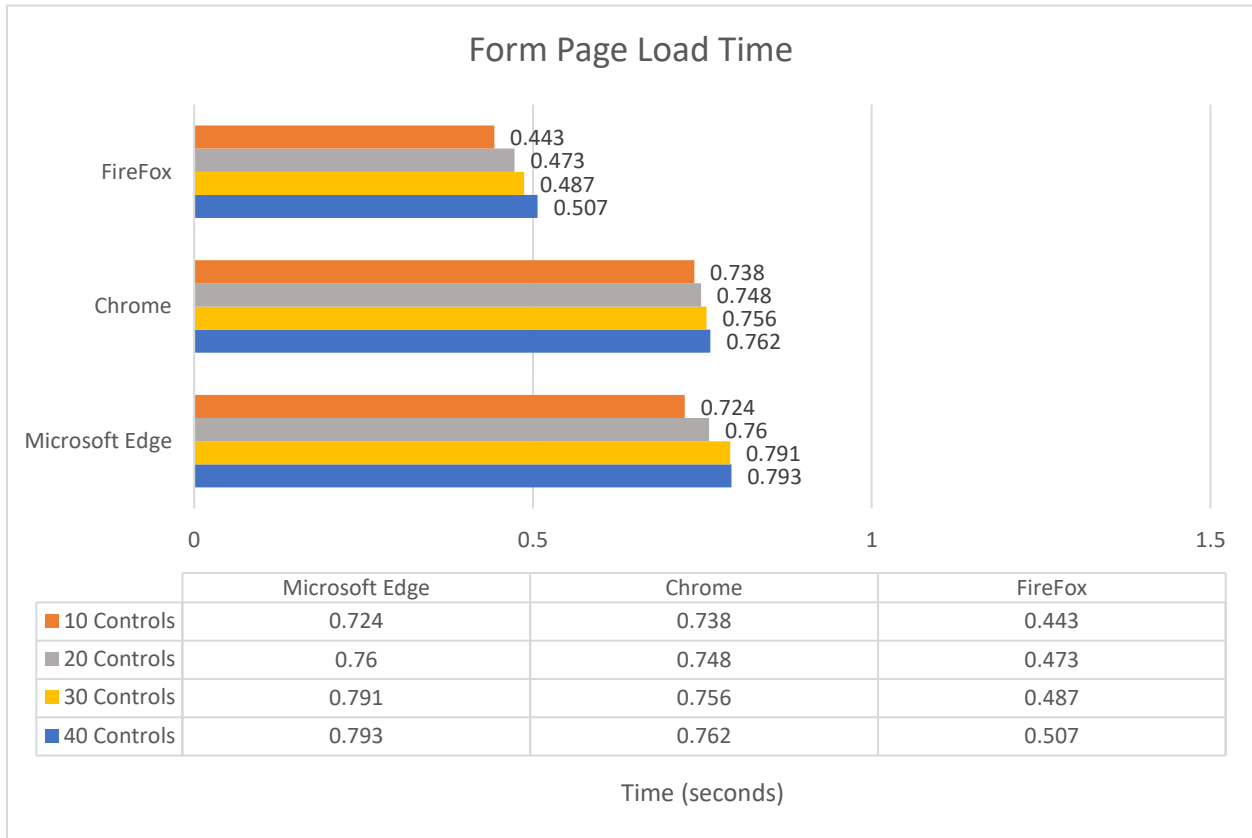
Results for Sign-In Page Load Time



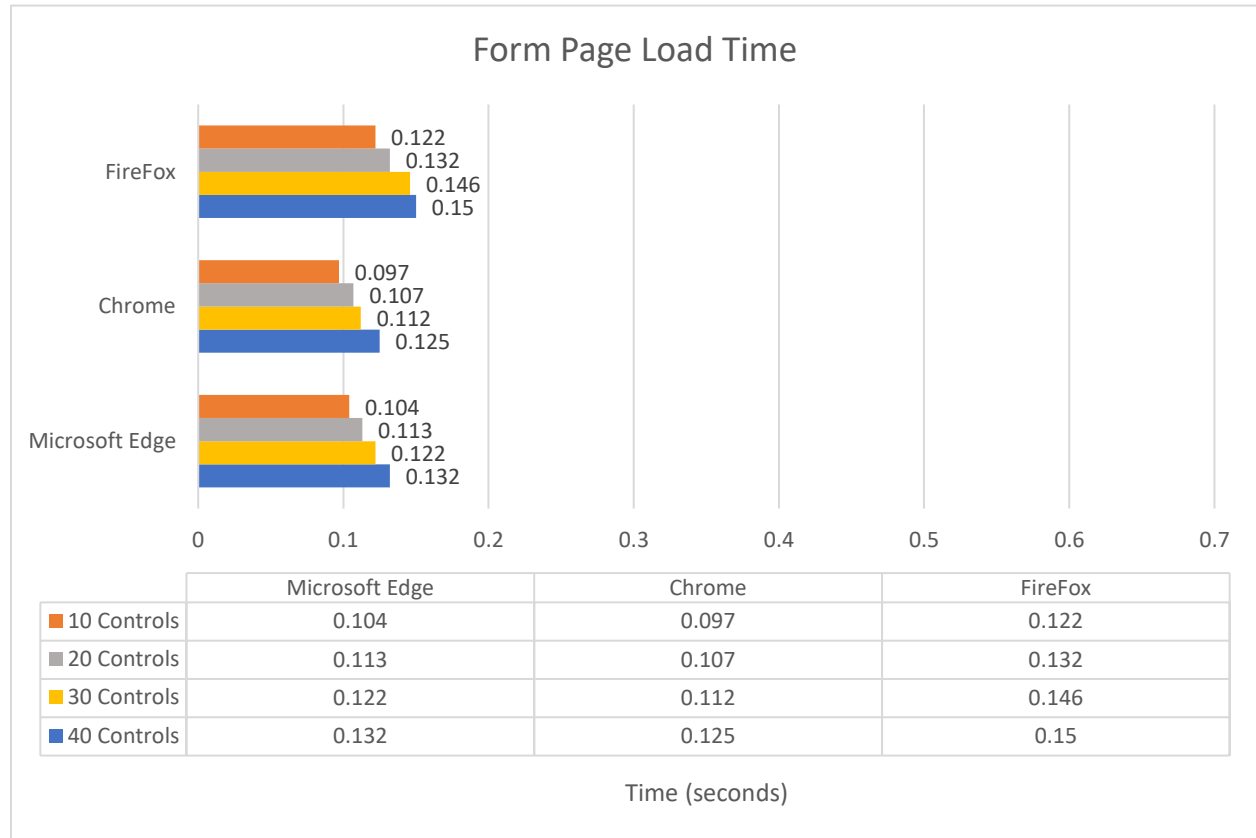
Results for Home Page Load Time



Results for Work Center Load Time



Results for eForms Load Time



Contributing Factors

The results mentioned here are under simulated conditions as described throughout this document. Performance may vary based on the following factors:

- Hardware specifications for the client machine.
- Browsers used.
- Network bandwidth.
- For eForms, the performance may vary based of the following factors:
 - Number of form controls on the forms
 - Data requested from external systems using lookups

- Number of sections in the forms