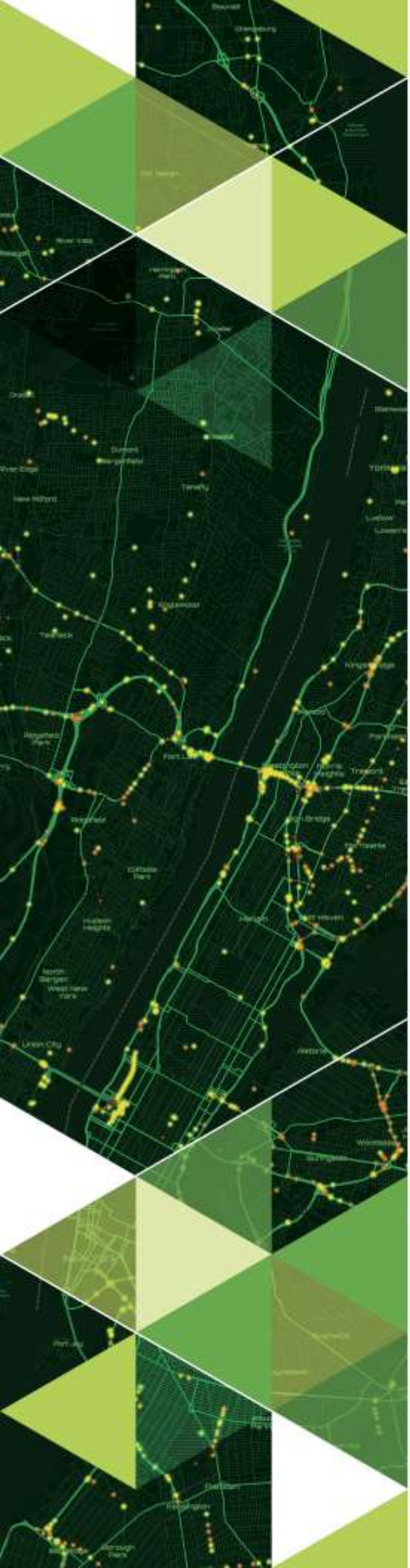

AN ESRI
TECHNICAL PAPER

May 2021

ArcGIS Enterprise deployment guide—Scene layer benchmark testing

380 New York Street
Redlands, California 92373-8100 USA
909 793 2853
info@esri.com
esri.com

esri | THE SCIENCE OF WHERE

Table of contents

ArcGIS Enterprise deployment guide—Scene layer benchmark testing

Executive summary

Scene layers represent geospatial data in 3D. Due to its size and complexity, you need to ensure you have the right resources available to store and access this data in an efficient way. Every use case is individual, and deployment planning needs to be adapted to your specific needs. The best way to successfully deploy scene layers is to establish a performance baseline for your system.

This guide describes hypothetical scenarios that represent cities in 3D using different data. Use the test data provided for the example scenarios to determine how it performs in your environment and compare that to how the scenarios performed in an Esri test environment. Based on the results of your tests, you can estimate the capacity of your environment for the scenario and interpolate what capacity can be expected for your own use cases.

This guide is to be used in combination with the information presented in the ArcGIS Enterprise deployment guide—3D data as scene layers technical paper. The sections in this paper mirror those in the deployment guide to help you connect the test scenarios presented in this paper with the concepts presented in the deployment guide.

The detailed information in this guide is intended for IT staff who are responsible for acquiring, configuring, and maintaining the hardware and software necessary to run scene layers, store scene layer data, and consume scene layers.

Understand the data

A common use of 3D GIS is the visualization of cities—including buildings, street furniture, and vegetation—to allow a broad audience to find information about specific buildings or get a first look at planned changes in the city. In this guide, there are two examples of how buildings in such a 3D city could be represented in scenes. The scenarios are invented to illustrate the workflows of scene layer deployment.

Download the ArcGIS Enterprise Deployment Guide - Scene Layers.zip file. This package includes the ArcGIS Pro project `DeploymentGuideScenarios.aprx` with scenes of New York and Montreal. The scene includes bookmarks that mimic visiting different locations within the scene. It also includes the test plans for each scenario. You will need these files to test your environment.

Thematic scene of New York City

Scenario 1: The City of New York wants to provide a scene to the public that allows navigation to individual areas of New York using a global scene. The scene should give users a general overview of the urban area, including buildings and trees. The scene also needs to contain a thematic basemap and elevation information. Because this is publicly available data, you don't plan to update the data yourself but plan to replace the entire contents frequently when the City updates its open data. You anticipate that people who interact with the data will visit tourist attractions and other places of interest and click individual buildings to get more information about them. At this point, you're not sure how many users would visit the website, but because this is publicly available, your assumption is that there will often be many users simultaneously viewing the scene.

A realistic scene of Montreal

Scenario 2: The City of Montreal has captured digital representations of buildings in the city, including building textures, and would like to make this accessible for visitors planning a trip to Montreal. The data will allow the general public to get an impression of the city. Consumers of the data will navigate the scene layer and visit individual landmarks. Also, in this scenario, no minimum number of users to visit the web scene is defined. Instead, you want to test your environment to find out the number of users who can be served in your software and hardware configuration.

Scene layer types and complexity

Both scenarios use scene layer packages to create the scene layers, but the data types and level of complexity are different.

Scene layer types and complexity in New York thematic scene

The New York example has untextured buildings and thematic trees. The buildings scene layer package (SLPK) was created from a multipatch feature class and contains over 1 million buildings. The trees SLPK was created from a point feature class. The data source was a .csv file that included geographic coordinates of the trees. You can publish these directly to your ArcGIS Enterprise portal.

The complexity of the scene layer representing the buildings and trees of New York is low. The building layer has over a million individual buildings, but the buildings are untextured and are represented using a simple shape defined by few vertices. No

additional symbology or transparency is applied to the scene. Because of the low complexity, high performance can be assumed.

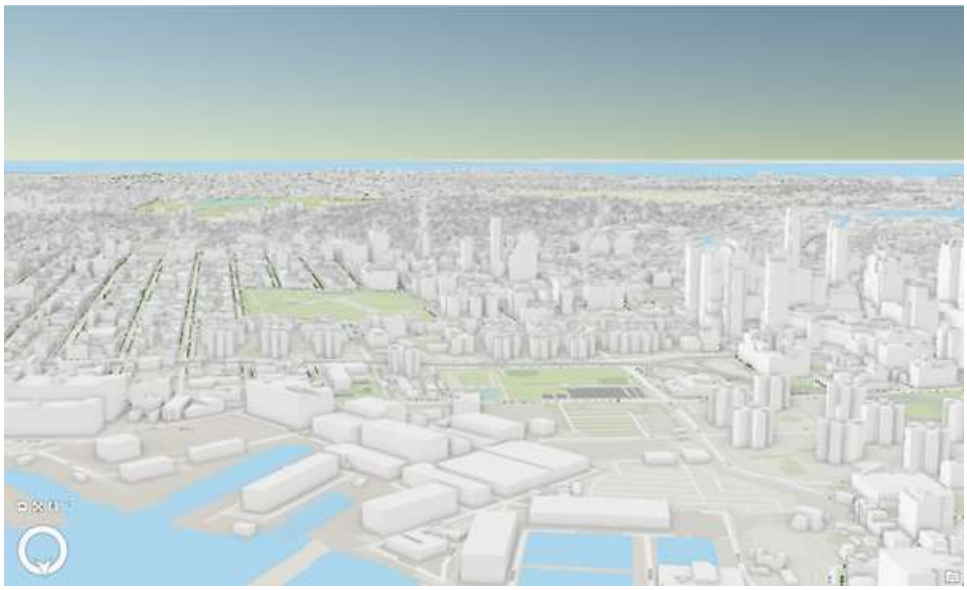

Figure 1: New York scene including buildings, trees, basemap, and elevation layers

Scene layer types and complexity in Montreal realistic scene

This example uses a data layer of textured buildings. Buildings with textures have already been processed, and a 3D object SLPK has been created.

Figure 2: Montreal scene including textured buildings, a basemap, and elevation layers

Complexity in the scene layer representing buildings in Montreal is high. Even though the number of buildings is much smaller (8,911 buildings), all buildings are textured. The textures do not repeat; every building is represented with an individual texture, which increases the number of textures used. The compression types used for the textures are JPG and DDS. Because of the higher complexity, it is likely that more

resources are needed to achieve high performance, as discussed in the technical paper ArcGIS Enterprise deployment guide—3D data as scene layers.

Note: *The more layers there are that participate in a scene, the more resources are required in ArcGIS Enterprise. For your own scenes, evaluate what other content you want to include. For example, in addition to the buildings and trees, you could also include point cloud data or other point scene layers representing street furniture. Scenes often include a basemap and base elevation provided by Esri or your organization. If you use basemaps and elevation provided by Esri, scene performance is also affected by the scene's access to ArcGIS Online in addition to accessing the scene layers running on your Enterprise portal.*

Usage patterns

In both scenarios, the data is considered relatively static. The test scene contains scene layers that have an SLPK as their data source. When publishing the scene from ArcGIS Pro, the local SLPKs are uploaded to an ArcGIS Enterprise portal and published as scene layers. The scene layer is shared with everyone (public). When updates are needed, the layer owner replaces it with another scene layer. Those who access the scene layer will use it to visit areas of interest and navigate the scene.

Note: *There are no plans to maintain individual buildings by editing each feature's geometry or attributes in the scenarios. If an update of the data is needed, the entire service is replaced. See Manage hosted scene layers in the ArcGIS Enterprise help for instructions on replacing scene layers*.

Caching options

In both scenarios, the data is cached locally. Therefore, no associated feature layer is created.

Hardware and software

The ArcGIS Enterprise deployment guide—3D data as scene layers technical paper provides a list of required software and publisher privileges as well as guidelines to help you estimate what hardware you will need.

When the test data and scenarios were run at Esri, no new hardware was purchased. Instead, existing hardware in the performance and scalability lab at Esri was used. Like in the real world, existing hardware must be used and evaluated to determine if it is adequate to meet your needs or if new purchases must be initiated. The hardware used by Esri would be suitable for a large deployment with many different layers and a large user base consuming these services.

For example, a scene such as the New York layer included in the test scenario could be viewed by 1,410 users at the same time on the hardware tested without experiencing a degradation in drawing performance. The details of the hardware and software deployment used for the test runs performed at Esri are summarized in the next two sections.

Software

In the test scenarios at Esri, a web scene with the scene layers was published from ArcGIS Pro 2.7 to ArcGIS Enterprise 10.8.1 (this includes Portal for ArcGIS, an ArcGIS GIS Server site, relational and tile cache data stores, and ArcGIS Web Adaptor).

Hardware

A total of five machines were used when running the test scenario at Esri. For specifications of each machine and the software installed on each, see Table 8 in the Esri test results for New York and Montreal scenarios section. All machines were running Microsoft Windows. The use of other operating systems is not addressed in this test scenario.

Test your environment

By testing your environment, you can establish benchmark values of the maximum throughput your environment can deliver. Two datasets are provided in the .zip file you downloaded. One is the New York dataset, which has no textures; the other is the Montreal dataset, which does include textures. You can use your environment's benchmarks to compare with known benchmarks obtained from the test lab at Esri. Additionally, a calculation using Little's Law will provide a theoretical estimate of the number of users your environment can handle. Ultimately, you will want to test using your own data along with the anticipated usage patterns of your users.

It is important to note the following:

- A benchmark value is particular to the dataset used, and the use of other datasets (such as your own data) can yield benchmark values different from the New York and Montreal example scenes.
- The calculation of the number of users that your system can handle is an approximation based on anticipated usage patterns of your users. Some factors are "best guess" assumptions, such as how and where they pan and zoom, as well as how long a user pauses between mouse clicks.

Testing requirements

The following requirements must be met before you begin testing:

- Your ArcGIS Enterprise test environment must be up and ready.
- Download the ArcGIS Pro project and point the layer data sources to the downloaded SLPKs. This project contains the 10 bookmarks used for load testing.
- Publish the scene in the test project as a web scene layer to your ArcGIS Enterprise portal. Keep the naming of the scene as is. The JMeter project included in the .zip file uses the same web scene layer names as is. You can keep basemap and elevation information in the scene, but the JMeter project excludes the basemap and elevation layer from the test run.
- You must have administrator login access to all machines (required for metric collection).
- Two test machines are required. One is the **test client**, and the second is referred to as the **remote host**. The **remote host** works with the **test client** to apply the load. While it is not an absolute requirement, the **remote host** is

included in this scenario because it was used in the Esri test lab. Both the **test client** and **remote host** must have the same version of JMeter installed, and the machines must be in the same subnetwork. Machine names are color coded through the remainder of this document to help you identify them.

There are six steps to obtaining the performance benchmark for your ArcGIS Enterprise environment:

1. Define the test plan. This has been done for the New York and Montreal scenarios.
2. Configure the **test client** machine and a **remote host**.
3. Configure the Apache JMeter test project for your environment.
4. Verify that the test requests are working successfully using a validation run.
5. Execute the load test.
6. Validate the results.

Each of these items is addressed in the remaining sections of this paper.

Define the test plan

A test plan is used to define the objectives and establish limits on what is tested and how it is tested. Most of this is addressed by using the ArcGIS Pro and Apache JMeter projects you downloaded. You can use these as an introduction to familiarize yourself with the load testing process. Eventually, you will want to create a test plan for your own data and scene layers. Below are some minimum basic elements of a test plan:

- Define the objective. You should have an expected capacity goal, for example, 100 concurrent users per hour.
- Identify the web scene layer to be tested.
- Define the anticipated usage patterns of your users.
    - Use your best judgement to define the amount of time your users will take to complete a task (known as think time). This is used in Little's Law calculations, which are explained in the Esri test results for New York and Montreal scenarios section.
    - Identify any areas of your scene layer that you anticipate will experience the highest visitor traffic. For the New York and Montreal benchmark scenarios, bookmarks were used.

Configure the test client machine and a remote host

Perform the steps below on both the **test client** and the **remote host** machines.

Note: *The steps to install, configure, and run JMeter only refer to the testing scenarios. For more details on using JMeter, visit the Apache JMeter website.*

1. Download Java (Open JDK for Windows/x64) from the JDK Java website. Unzip the file to the location of your choosing on the **test client** and on the **remote host** and install it.

2. After installing Java Open JDK, add the `<JDK installation>\bin` path to the Path of the System Environment variable on each machine. See instructions on the Oracle website.
3. Download Apache JMeter and unzip it to the location of your choosing on each machine. Once unzipped, it is ready to use.
4. Additional JMeter plug-ins are required to complete the test. Download the `plugins-manager.jar` file from https://jmeter-plugins.org/install/Install/. Place the downloaded `.jar` file in the JMeter <JMeter installation folder>\`lib\ext` directory.

To allow the **test client** and **remote host** to work together, several steps must be completed.

1. Open a command prompt on the **test client** machine, change directories (cd) to the `JMeter\bin` folder, and run the following:

   ```
   create-rmi-keystore.bat
   ```

   The script displays several questions about your organization to which you must respond. If you encounter problems, see Remote Testing in the JMeter online documentation.

   You should now have a file named `rmi-keystore.jks` in your `JMeter\bin` directory on the **test client** machine.
2. Copy the `rmi-keystore.jks` file from the **test client** machine and paste it into the `JMeter\bin` folder on the **remote host** machine.
3. While still in the `JMeter\bin` folder on the **remote host** machine, open the `jmeter.properties` file in a text editor.
4. Search for **server.rmi.ssl.disable**. Uncomment this parameter and set it equal to **true**.
5. Save and close the `jmeter.properties` file.
6. In the `JMeter\bin` folder on the **remote host** machine, double-click the `jmeter-server.bat` file. Running this file here is what configures this machine as the **remote host.**
7. On the **test client** machine, follow these steps to alter the `jmeter.properties` file:
   a. Open the `JMeter\bin` folder of the **test client** machine.
   b. Open the `jmeter.properties` file in a text editor.
   c. Search for **server.rmi.ssl.disable**. Uncomment this parameter and set it equal to **true**.
   d. Search for the variable **remote_hosts** and change the value to the IP address of the **remote host** machine.
   e. Append the following two lines to the bottom of the file:

      ```
      jmeterPlugin.perfmon.interval=5000

      forcePerfmonFile=true
      ```

   f. Save and close the `jmeter.properties` file.

Configure the JMeter test project for your environment

To test your environment, you can use the JMeter test projects for New York and Montreal included in the ArcGIS Enterprise Deployment Guide—Scene Layers package you downloaded earlier.

Open the JMeter project

The next steps refer to the New York test plan. You can repeat the same steps for Montreal.

Ten bookmarks are defined for each scenario (New York and Montreal) to mimic users visiting different landmarks within the scene. In the following steps, you will change the values of the variables in each project to make them work with your environment.

1. In the `<JMeter Installation Folder>\bin` on the **test client** machine, double-click the `JMeter.bat` file to open the JMeter console. You may get a pop-up about required plug-ins. Select **Yes** to install any missing plug-ins.

   The JMeter console opens.

2. Go to **File** > **Open** and browse to the **NewYork** JMeter test project.

   Tip: *When you run the test for the Montreal data, open the **Montreal** project.*

JMeter components

You'll see the following components in the New York and Montreal JMeter projects:

Component	Description
HTTP Cookie Manager	This controls how cookies are handled. Keep the settings as defined in the project.
bzm—Concurrency Thread Group	This is the component that controls the load that will be applied to your environment. Generally, it will start at one thread, increasing in number until the system reaches its limit during the test. This is the point of maximum throughput. Apply a few threads beyond this point so that the throughput curve levels out. This indicates that your ArcGIS system has reached its maximum throughput or that some other bottleneck is limiting the system.
Transactions 1 through 10	These are the requests that are submitted to the ArcGIS Enterprise environment.
View Results Tree	Before executing a full load test, you will execute a single thread practice run or a validation run. A validation run allows you to verify that the test is functioning properly and make any corrections. The View Results Tree allows you to view the requests and responses of the validation run. You need to enable the View Results Tree before use. Disable it when not in use or when executing a full load test.
jp@gc—Perfmon Metrics Collector	Allows JMeter to monitor the CPU, Memory, Network, and Disk metrics of the servers you test. All machines to be monitored must have the ServerAgent-2.2.3 or later file downloaded to them. See the Set up JMeter to monitor machines section for installation instructions.

Table 1: Description of JMeter components in the test JMeter projects

Configure the JMeter variables

You must configure the variables to correctly match your environment.

1. In the JMeter project, click the Test Plan named **NewYork**. Modify the following values in the main window to match your environment.

    - **webAdaptorHostName**—Set to the host name of the machine where the hosting server's ArcGIS Web Adaptor is installed.
    - **newYorkWebSceneLayer1**—Set to the name of the 3D object scene layer in the New York web scene.
    - **newYorkWebSceneLayer2**—Set to the name of the point scene layer in the New York web scene.
    - **webAdaptorArcGISServer**—Set to the context name of the ArcGIS Web Adaptor that is associated with the ArcGIS Server site that is acting as the hosting server.
    - **projectFolder**—Set to the path to your JMeter project containing the `newYork.jmx` file.
    - **webSceneItemID**—Set to the item ID of the New York web scene item in your portal. To obtain the web scene item ID, sign in to your Enterprise portal and open the item details for the New York web scene. You can copy the scene layer's item ID from the URL. For example, if the URL for the New York web scene item is `https://myserver.mynetwork.com/portal/home/item.html?id=1234567890abcdefghijk`, 1234567890abcdefghijk is the item ID of the New York web scene.

    The following is an example of how the variables would look in the New York JMeter project:

    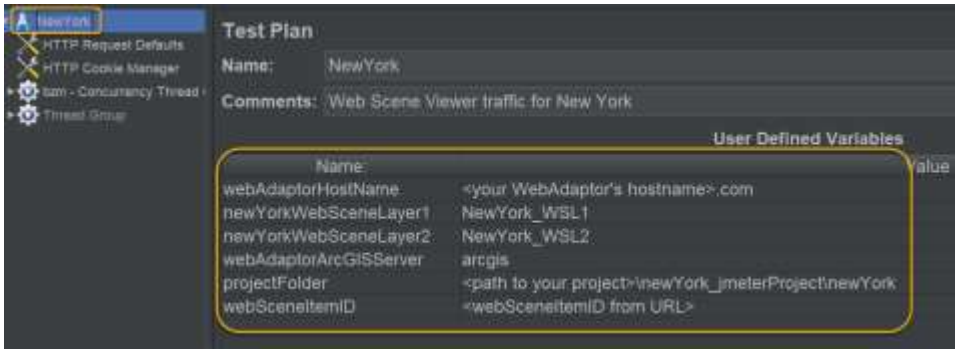

    Figure 3: JMeter variables for the New York project

2. Save your project.

Tip: *When you repeat the previous two steps for the **Montreal** test plan, the variables are as follows:*

- **webAdaptorHostName**—Set to the host name of the machine where the hosting server's ArcGIS Web Adaptor is installed.
- **montrealWebSceneLayer**—Set to the name of the building scene layer in the Montreal web scene.

- **webAdaptorArcGISServer**—Set to the context name of the ArcGIS Web Adaptor that is associated with the ArcGIS Server site that is acting as the hosting server.
- **projectFolder**—Set to the path to your JMeter project containing the `montreal.jmx` file.
- **webSceneItemID**—Set to the item ID of the Montreal web scene item in your portal. To obtain the web scene item ID, sign in to your Enterprise portal and open the item details for the Montreal web scene. You can copy the scene layer's item ID from the URL. For example, if the URL for the Montreal web scene item is `https://myserver.mynetwork.com/portal/home/item.html?id=8675309zyxwvutsrq`, 8675309zyxwvutsrq is the item ID of the Montreal web scene.

The following is an example of how the variables would look in the Montreal JMeter project:

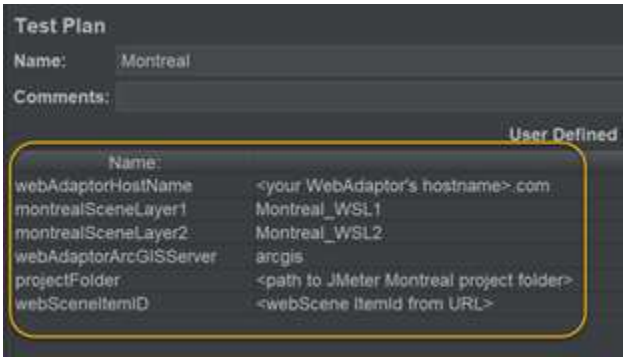

Figure 4: JMeter variables for the Montreal project

Monitor metrics

Monitoring the metrics of the machines you test is critical to understanding the test results. A brief description of each metric is provided in the following table:

Computer metrics	Description
CPU	A bottleneck occurs when this is at or near 100%.
Memory	A bottleneck occurs when the amount of free memory (in MB) approaches 0.
Network bandwidth	A bottleneck occurs when network bandwidth is saturated. (Not addressed in this document.)
Disk I/O	A bottleneck occurs when Disk % idle time is at or near 0. (Not addressed in this document.)

Table 2: Descriptions of computer metrics you can monitor

The metrics of CPU and memory are the expected bottlenecks for a scene layer. The network and disk metrics may also be monitored but are not the expected bottlenecks. For these reasons, and because a comprehensive analysis of all these metrics is beyond the scope of this document, this document focuses primarily on CPU and memory, which are expected to be the limiting components.

Set up JMeter to monitor machines

Monitoring is accomplished with two software components:

- **jpgc-perfmon**—A JMeter plug-in that you add to your JMeter project.
- **Perfmon ServerAgent**—Allows JMeter to obtain metrics on Windows machines. You can download Perfmon ServerAgent from GitHub.

All the following machines in your deployment should be monitored:

- ArcGIS Server
- ArcGIS Data Store (tile cache)
- ArcGIS Web Adaptor
- Portal for ArcGIS
- Test client
- Remote host

The following steps must be accomplished on each machine that is to be monitored:

1. If Java is not already installed, install Java or copy the Java JRE folder from an existing installation to each machine to be monitored.
2. Unzip the `ServerAgent-2.2.3.zip` file.
3. Open the `startAgent.bat` file in a text editor and modify the `java` folder to point to the bin folder of the existing Java JRE installation. For example, set `"<path to jdk>\bin\java" -jar %0\..\CMDRunner.jar --tool PerfMonAgent %*`.
4. Save and close the `startAgent.bat` file.
5. Double-click the newly edited `startAgent.bat` file to start the metric collection agent.

In the JMeter project on the **test client**, use the following steps to configure JMeter to monitor specific machines:

1. Click **jp@gc – Perfmon Metrics Collector** under the **NewYork** project.
2. Click the **Add Row** button under **Servers to Monitor**.
3. Type the name of your ArcGIS Server machine, and select **CPU** as the metric to collect. Leave the port default of 4444.
4. Click the **Add Row** button again and type the name of your ArcGIS Server machine to add a metric for memory.
5. In the **Metric parameter** column, click the options (…) button.
6. When the new window opens, select **free** and select **Megabytes.** Click **Apply** when metrics are set.
7. Repeat steps 2 through 6 for the rest of the machines to be monitored, typing the name of the other component machines in steps 3 and 4.
8. Save the project.

The JMeter validation run: A final pretest check of your project

Use the JMeter GUI to verify that your test project is working properly. Once you have verified that all requests are getting a valid response and that metrics are being collected, you can close the JMeter GUI and run the full load test using the `runme.bat` file in your project folder on the **test client**. Make sure that the Perfmon ServerAgent is running on all machines.

To execute the validation run, complete the following steps:

1. In the project in JMeter, click the **bzm – Concurrency Thread Group** and verify that the three values in the main panel are set to **1**. Doing so will make JMeter execute one user thread for one minute using one step.

   Note: *After completing a successful validation run, you must change these values for execution of the full load test run. Instructions for changing these values are provided in the next section.*

2. Right-click **View Results Tree** and select **Enable**.

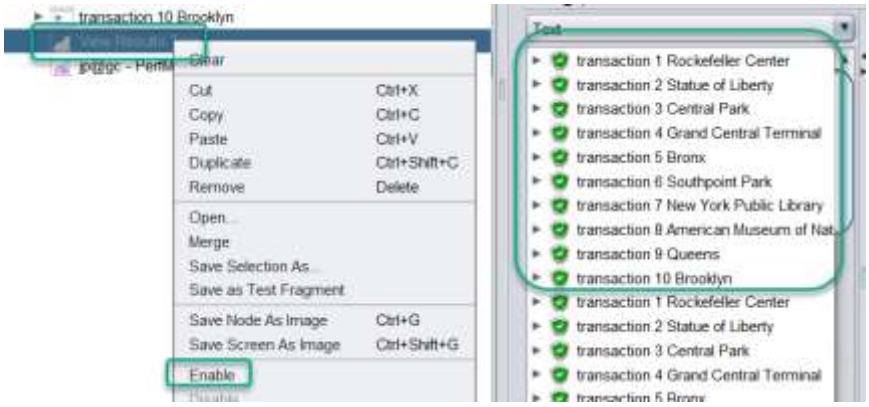

Figure 5: Enable the Results Tree in JMeter.

Note: *Remember to disable the **Results Tree** after you complete this validation run.*

3. Click the start button located in the top menu bar.

   The start button is highlighted in the image below:

Figure 6: Click Start on the menu bar in the JMeter interface.

4. Watch the **Results Tree** panel. If nothing shows, click **View Results Tree** again so it is the active component. All responses should have a green icon. A red icon indicates an error. If you encounter an error, click the error and examine the **Response Body** for clues.

   Note: *If JMeter does not respond after this step, you may need to increase the allocated memory that the JMeter GUI mode (or non-GUI mode) uses; refer to JMeter documentation for instructions.*

5. Click **jp@gc - Perfmon Metrics Collector** and verify that metrics from all the machines are showing. It should resemble the following example (Figure 7):

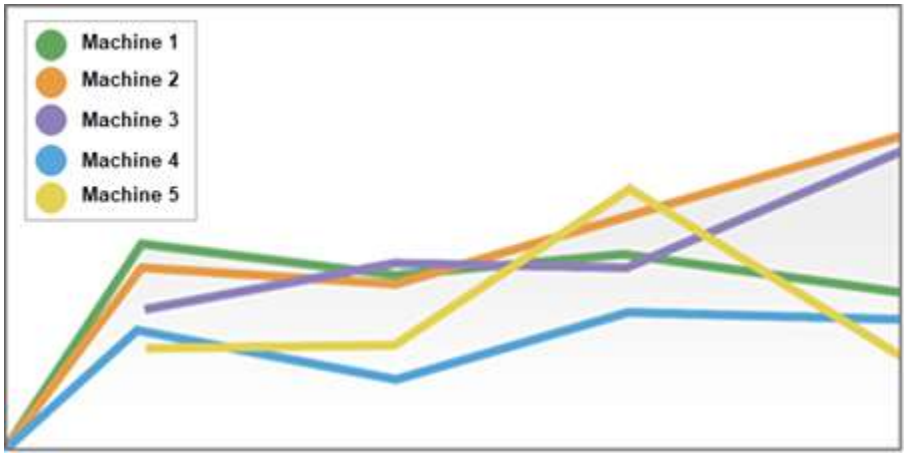

Figure 7: Performance metrics for all machines

6. After you validate that all requests are green and metrics are showing, disable the **View Results Tree** (repeat step 2 above, but select **Disable**) and save the project.

   Note: *For the final run, it is important that you disable the View Results Tree because having it enabled can impact the test results.*

Execute the full load test

Two final items remain before you execute the full load test. They are described in the next two sections.

Set the load in bzm – Concurrency Thread Group

The bzm - Concurrency Thread Group properties determine how much load is applied. The bzm-Concurrency thread group value will be slightly different for every hardware and software configuration.

You must configure the three variable values described in the following table:

Variable	Description
Target Concurrency	The maximum number of concurrent threads that will apply load. This value should be about 1.5 x the total number of CPUs on all the machines in the ArcGIS Server site. For example, if the ArcGIS Server site contains two 8-core machines, the value would be 16 x 1.5% = 24. This should be sufficient in most cases. This is done to ensure that all cores have at least one thread of execution to work on. This value is addressed in more detail in the Validate the result section to determine if the value should be adjusted.
Ramp Up Time	The total time of the test in minutes. A reasonable value is to set it equal to the Target Concurrency value above.
Ramp-Up Steps Count	The number of steps (concurrent threads) the test will execute to reach the Target Concurrency. Set this equal to the Target Concurrency value as well.

Table 3: Descriptions of the bzm – Concurrency Thread Group variables to set

Follow these steps to configure all three variables to the same value:

a. Click **bzm- Concurrency Thread Group** in the JMeter console. Use the value explained in the table above for all three variables: **Target Concurrency**, **Ramp Up Time**, and **Ramp-Up Steps Count**.

   All three should now have the same value, as shown on the next page.

Figure 8: The bzm - Concurrency Thread Group settings in JMeter

b. Save your project.

Modify the values in the runme.bat file

For the full load test run, you must modify the `runme.bat` file. This file tells JMeter where to put test results and what to name them, as well as the IP address of the **remote host** and other information.

Variable	Description
run	A sequence number that you manually change with each successive test execution, allowing you to keep track of each test.
jmeterbin	The path to the JMeter `bin` folder. Modify this to match the location in your environment.
remoteHosts	The IP address of the **remote host** machine.

Table 4: Descriptions of the runme.bat variables to set

Follow these steps to alter the variables in the `runme.bat` file:

a. Open File Explorer and browse to the location of the New York or Montreal project.

b. Edit the `runme.bat` file in a text editor.

c. Set the **run** variable to the name of the run you want to use.

   Note: *If you want to create a separate output for each test run, you must alter this value and save the `runme.bat` file before each test run.*

d. Set the **jmeter** variable to the location of the `JMeter\bin` folder on your **test client** machine.

e. Set the **remoteHosts** variable to the IP address of the **remote host** machine. Replace the xxx.xxx.xxx.xxx text with your IP address.

f. Save and close the `runme.bat` file.

Execute the test

After editing the `runme.bat` file, double-click the file to run the test. A green command window appears showing that the test is running and the test's status. When the test ends, the command window closes. You can view the test results in the reports folder of your project.

View the results

After the test successfully runs, you can view the results. The diagram below illustrates the ideal trend of the test run over time.

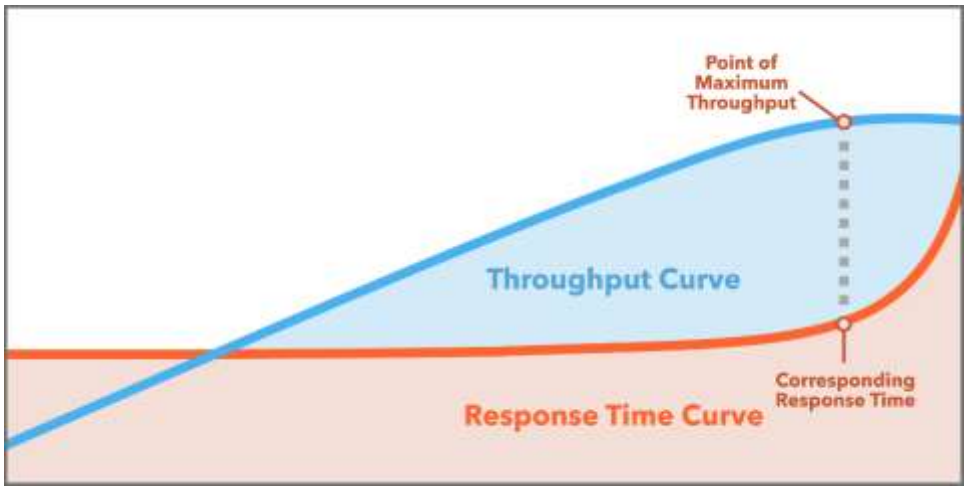

Figure 9: Throughput and response time curve showing the point of maximum throughput

Ideally, the throughput curve will have the form of the blue line above. The curve flattens at some point, indicating that the system has reached its highest level of performance and a further increase in throughput is not possible due to a hardware or software bottleneck. The section of the graph where the curve bends is referred to as the knee of the curve. The value for maximum throughput of your system is taken just after the knee. The orange line in the graph is a representation of the response time of a transaction. In the report, it can be found under **Charts** > **Response Times** > **Time Vs Threads**. There will be 10 response timelines in the graphs, one for each bookmark. As the response timeline passes the point of maximum throughput, it trends upward because requests take longer to be serviced.

When the load test is finished, the results can be found in an index file in the reports folder of your project. Each test run will be saved in a separate folder if you alter the run variable in the `runme.bat` file before each test run. The latest run will be inside the latest folder.

1. Open the test result file and click the `index.html` file.

   A browser window opens containing your test results.

2. Click **Charts** > **Throughput** > **Total Transactions Per Second**.

   You should see a graph like the following. Using this image as a guide, find the point of maximum throughput.

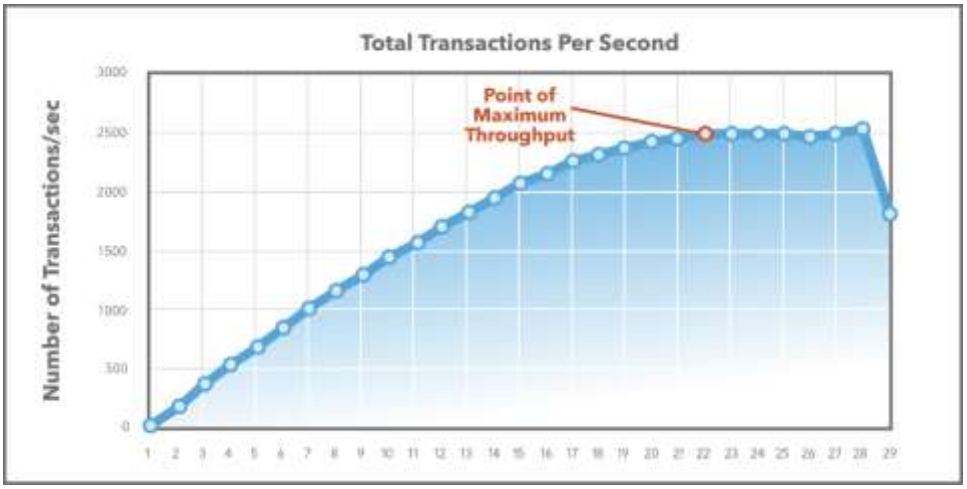

Figure 10: Schematic view on the point of maximum throughput in the test run metrics.

Validate the test results

After the completion of every test, you need to validate the results. The following questions to help you with validation:

Was the knee of the throughput curve captured?

You should be able to identify the knee in your graphs. If the curve looks more like a line that continues to go up, increase the Target Concurrency, Ramp Up Time, and Ramp-Up Steps. Try executing the test again with twice the original values.

Are there any errors?

In the `index.html` file, click the dashboard and scroll to the bottom sections where errors are shown. You should see zero errors. If the percentage of errors is greater than 1 percent, view the type of error and any error messages in the `index.html` file for clues. Other places to look are ArcGIS Server Manager logs.

Did the CPU on the machines show activity?

Inspect the PerfMonMetrics to assure you had a valid run and the test executed the different transactions.

1. In the project tree, right-click **jp@gc PerfMon Metrics Collector** and click the browse button to open the perfmon file corresponding to your test run. This will be a file in the JMeter project folder. The file name will be in the format `perfmon_<yearMonthDay>-<timeStamp>.csv`; for example, `perfmon_20210209_153359.csv`.
2. Click the **Rows** tab and click **(Un)Check All**.
3. Check every box that has **CPU** in the **Row Name**.

4. Click the **Chart** tab.

You should see activity on all CPUs. You should expect higher activity on the ArcGIS Server machine and the tile cache data store machines. How much activity will depend on the number of CPUs on each of these machines. The ArcGIS Server graph should show a knee in its curve.

Other items to inspect to validate results include the following:

- Memory—Check that memory was not exhausted.
- Network bandwidth—Check that the network was not saturated. You will need to know the network bandwidth between all machines.
- Disk I/O—Check that the disk was not being overutilized.

Esri test results for New York and Montreal scenarios

After you validate your test results, compare the values of your maximum throughput with the benchmark numbers obtained by Esri.

	Total throughput in requests/second at maximum throughput
New York	2,488
Montreal	822

Table 5: Throughput numbers obtained in requests per second

Comparing your maximum throughput value with the benchmark will give you a sense of how well your system performs in general.

JMeter gives response times of individual bookmarks in the **Time Vs Threads** graph. Listed below are the benchmark response time values for each bookmark at maximum throughput that were seen when tested in the Esri lab. These can be used to identify which transactions take longer to complete.

New York bookmarks	Bookmark average response time in milliseconds at maximum throughput
1 Rockefeller Center	554.47
2 Statue of Liberty	61.71
3 Central Park	909.47
4 Grand Central Terminal	101.04
5 Bronx	888.9
6 Southpoint Park	467.83
7 New York Public Library	6210.37
8 American Museum of Natural History	213.46
9 Queens	230.56
10 Brooklyn	873.35

Table 6: Average response time per bookmark in the New York project

The average response time is 1051.116 milliseconds, or approximately 1.0 second.

Montreal bookmarks	Bookmark average response time in milliseconds at maximum throughput
1 Mont Royal	3238.24
2 Vieux-Montreal	2535.15
3 Notre Dame Basilica	2023.96
4 Park Jean Drapeau	1069.96
5 Musee Des Beaux Arts	3327.66
6 St. Mary Queen of the World	2892.09
7 McCord Museum	4771.33
8 St Louis and Rue Denis	2439.24
9 Overview 1	1928.12
10 Overview 2	594.84

Table 7: Average response time per bookmark in the Montreal project

The average response time is 2482.059 milliseconds, or approximately 2.48 seconds.

Calculate capacity with Little's Law

Interactive Response Time Law, or Little's Law with think time, is a formula applied to queueing systems to calculate how many users a system can handle. Little's Law states that the average number of items in a system is equal to the average rate at which items enter and exit the system multiplied by the average amount of time an item spends in the system. Think time is the time between the completion of one request and the start of the next request. Examples of applying Little's Law with think time to the New York and Montreal test results are provided in this section.

Given the following formula:

$$N = X * (R + Z)$$

- N = Number of jobs or concurrent users
- X = Throughput per second in the system
- R = Response time, or average time a job spends in the system
- Z = Think time

The value for X is equal to the sum of each value of each curve in the **Transactions Per Second** graph in JMeter at maximum throughput. To find it, open the `index.html` file in your test results, go to **Charts** > **Throughput** > **Transactions Per Second**. Find the point of maximum throughput on your graph and hover your mouse pointer over each value. Sum all 10 values. The sum of these at maximum throughput is the value for X. Take note of the number of minutes at which the maximum throughput occurs. A simple way to find this information is to count the number of points to the left of the maximum throughput point.

Tip: *Some curves will be underneath other curves on the chart, so you will have to remove the one on top by clicking its name in the legend at the bottom of the chart.*

The value for R is also obtained from the load test results. In the `index.html` file, open **Charts** > **Response Times** > **Time Vs Threads**. Count the same number of points from the left to find the corresponding values. Take the average of these. This is the value for R.

The value for Z is the think time and can range between 3 and 9 seconds. In the following example, the Z value is set to 6 seconds.

The following formula uses the values from the New York benchmark from Esri and a 6-second think time:

N = X * (R + Z)

N = 19.84 * (1.05 + 6)

N = 140.295 concurrent users/sec

The following formula uses the values from the Montreal benchmark from Esri and a 6-second think time:

N = X * (R + Z)

N = 8.426 * (2.48 + 6)

N = 71.45 concurrent users/sec

The Esri benchmarks were obtained using the hardware described in the following table. Use the SPECint_rate links for each machine to view machine specifications, such as number of CPUs and RAM used.

	Hardware	Role	SPECint_rate2006
Machine 1	Dell PowerEdge R620	ArcGIS Server	SPECint_rate_base2006 = 666
Machine 2	Dell PowerEdge R620	ArcGIS Web Adaptor	SPECint_rate_base2006 = 666
Machine 3	Dell PowerEdge R620	Portal for ArcGIS and ArcGIS Data Store (relational)*	SPECint_rate_base2006 = 666
Machine 4	Dell PowerEdge R640	ArcGIS Data Store (tile cache 1)	SPEC_int_rate_base2006 = 1320
Machine 5	Dell PowerEdge R640	ArcGIS Data Store (tile cache 2)	SPEC_int_rate_base2006 = 1320

Table 8: Machine specifications used in Esri baseline tests with New York and Montreal data

*A relational data store is a required component of a base ArcGIS Enterprise deployment. However, in this test scenario, nothing was stored in the relational data store, so an additional machine was not used. Because the relational data store is empty and is not being queried, its presence on the Portal for ArcGIS machine does not affect performance.

Note: *Ensure the hardware you use is supported based on your software license.*

Compare the benchmark data with your own deployment

After you have completed the test of your system with the provided data and established the benchmark for your own system, you can repeat the testing of the system with more of your deployed web scenes.

With the knowledge from the scene layer benchmark testing, you can adapt the test plan to use your data and web scenes. This will provide an approximate benchmark of your system based on the simulated user behavior that mimics the use of web scenes in your organization.

About the authors

This paper was composed by staff from multiple teams across Esri:

- Aaron Lopez, Advanced Enterprise Systems
- Ivonne Seler, PE, ArcGIS Pro
- Jill Edstrom-Shoemaker, PM, ArcGIS Enterprise
- Kimberly Peter, PE, ArcGIS Enterprise
- Philip Heede, PM, ArcGIS Enterprise
- Ricardo Perez, Technical Consultant, Advanced Enterprise Systems
- Richard Vargas, PE, ArcGIS Pro
- Satish Malipeddi, PE, ArcGIS Enterprise
- Sean Morrish, PE, ArcGIS Pro
- Zoe Veale, PE, ArcGIS Pro

esri | THE SCIENCE OF WHERE

Esri, the global market leader in geographic information system (GIS) software, offers the most powerful mapping and spatial analytics technology available.

Since 1969, Esri has helped customers unlock the full potential of data to improve operational and business results. Today, Esri software is deployed in more than 350,000 organizations including the world's largest cities, most national governments, 75 percent of Fortune 500 companies, and more than 7,000 colleges and universities. Esri engineers the most advanced solutions for digital transformation, the Internet of Things (IoT), and location analytics to inform the most authoritative maps in the world.

Visit us at esri.com.

Contact Esri

380 New York Street
Redlands, California 92373-8100 USA

1 800 447 9778
T 909 793 2853
F 909 793 5953
info@esri.com
esri.com

Offices worldwide
esri.com/locations

For more information, visit **esri.com/URL**.