**AEM PERFORMANCE STANDARDS & TESTING**

Performance testing is an important part of any AEM deployment. Depending on customer requirements, performance testing may be performed on the publish instances, author instances, or both.

This documentation will outline overall strategies and methodologies of performing performance tests as well as some of the tools that are made available by Adobe to assist in the process. Finally, we will analyze some of the tools that are available in AEM 6 to assist with performance tuning, both from a code analysis and system configuration perspective.

### Simulating Reality

What is most important when performing performance tests is making sure that you mimic a production environment as closely as possible. While this can often be difficult, it is imperative to ensure the accuracy of these tests. When working on designing performance tests, it is important to take the following points into consideration:

* **Production-like content**

Many performance measurements in AEM, such as query response time, can be impacted by the size of the content on the system. It is important to make sure that the test environment has as close of a copy of the production data as possible.

* **Production code**

The AEM version and hotfixes deployed in production should be the same in the test environment. It is also important to test on the version of the code that is deployed to production.

* **Production-like hardware and network configuration**

The tests will be meaningless without an environment that resembles the production one as closely as possible. Ideally, the hardware specifications, network interfaces, load balancers and CDN should be identical to production in the test environment.

* **Production load**

Many performance issues are not seen until the system is under heavy load. Good performance tests should simulate the load that the production systems will be under at their peak.

### 

### Setting Goals

Before starting on performance testing, it is necessary to set non functional requirements to specify load and response times. If you are migrating from an existing system, make sure that response time are similar to your current production values. For load, it is best to take the current peak load and double it. This will ensure that the website can continue to perform well as it grows.

### Tools

There are many commercially available performance testing tools on the market.  When running a load generating tool, it is important to ensure that the computers which are performing the tests have sufficient network bandwidth. Otherwise, once the test machine reaches the limits of its connection, no additional load will be generated on the environment under test.

#### **Testing Tools**

* Adobe’s **Tough Day** tool can be used to generate load on AEM instances and collect performance data.  Adobe’s AEM engineering team actually uses the tool to do load testing of the AEM product itself. The scripts executed in Tough Day are configured via property files and JMX XML files. For more information, see the [Tough Day documentation](https://helpx.adobe.com/experience-manager/6-3/sites/developing/using/tough-day.html).
* AEM provides out of the box tools to quickly see problematic queries, requests and error messages. For more information, see the [Diagnosis Tools](https://helpx.adobe.com/experience-manager/6-3/sites/administering/using/operations-dashboard.html#main-pars_title_8) section of the Operations Dashboard documentation.
* Apache provides a product called **JMeter** that can be used for performance and load testing as well as functional behavior. It is open source software and free to use, but has a smaller feature set than enterprise products and a steeper learning curve. JMeter can be found on Apache’s website at <http://jmeter.apache.org/>
* **Load Runner** is an enterprise grade load testing product produced by HP. A free evaluation version is provided. More information can be found on HP’s website at <http://www8.hp.com/us/en/software-solutions/loadrunner-load-testing/>
* Cloud based load testing tools like [Neustar](https://www.neustar.biz/services/web-performance/load-testing) can also be used.
* When it comes to testing mobile or responsive websites, a separate set of tools need to be used. They work by throttling network bandwidth, simulating slower mobile connections like 3G or EDGE. Among the more widely used tools are:  
  + [**Network Link Conditioner**](http://nshipster.com/network-link-conditioner/) - it provides an easy to use UI and works at a fairly low level on the networking stack. It includes versions for OS X and iOS;
  + [**Charles**](http://www.charlesproxy.com/) - a web debugging proxy application that in addition to several other uses, provides network throttling. Versions are provided for Windows, OS X and Linux.

#### 

#### **Optimization Tools**

**Monitoring**

The [Monitoring Performance](https://helpx.adobe.com/experience-manager/6-3/sites/deploying/using/monitoring-and-maintaining.html#main-pars_title_3) documentation is a good resource for tools and methods that can be used to diagnose issue and pinpoint areas for tuning.

**Developer Mode in Touch UI**

One of the new features in AEM 6’s touch UI is the Developer Mode. Just as authors can switch between edit and preview modes, developers can switch to developer mode in the author UI to see the render time for each of the components on the page and to see stack traces of any errors. For more information on developer mode, see this [CQ Gems presentation](http://dev.day.com/content/ddc/en/gems/aem-6-0-developer-mode.html).

**Using the rlog.jar to read the request logs**

For a more comprehensive analysis of the request logs on an AEM system, **rlog.jar** can be used to search through and sort the **request.log** files that AEM generates. This jar file is included with an AEM installation in the **/crx-quickstart/opt/helpers** folder. For more information on rlog tool and the request log in general, see the [Monitoring and Maintaining](https://helpx.adobe.com/experience-manager/6-3/sites/deploying/using/monitoring-and-maintaining.html) documentation.

**The Explain Query Tool**

The [Explain Query tool](https://helpx.adobe.com/experience-manager/6-3/sites/administering/using/operations-dashboard.html#main-pars_title_1097830066) in ACS AEM Tools can be used to view the indexes that are used when running a query. This can be very useful when optimizing slow running queries.

**PageSpeed Tools**

Google’s PageSpeed tools offer site analysis for adherence to best practices for page performance as well as a plugin that can be installed alongside the dispatcher on an Apache instance for additional optimizations.  For more information, see the [PageSpeed Tools Website](https://developers.google.com/speed/pagespeed/).

## 

## Author Environment

### Performing Tests

In order to conduct performance testing on the author environment it is necessary that you simulate the experience of porduction authors. This means that the author installations must contain all the components, OSGi bundles, UI customization, custom indexes and any other additions you have in place for the production author instances.

There are many automation frameworks available that are designed for performance and load testing. Custom scripts can be recorded in these tools and then played back to simulate a peak number of authors performing similar content creation and activation activities simultaneously. It is recommended you use the Tough Day tool to simulate activities like uploading thousands of assets or activating large numbers of pages.

For the types of environments that have requirements of heavy asset loading or page authoring it is imperative to use tools like Tough Day in order to ensure that the environment will operate efficiently under peak load. [WebDAV](https://helpx.adobe.com/experience-manager/6-3/sites/administering/using/webdav-access.html) is a tool that does not require scripting and can also be used to load large amounts of assets.

#### **MongoDB Specific Steps**

On systems with MongoDB backends, AEM provides several [JMX](https://helpx.adobe.com/experience-manager/6-3/sites/administering/using/jmx-console.html) MBeans that need to be monitored when performing load or performance tests:

* The **Consolidated Cache Statistics** MBean. It can be accessed directly by going to *http://server:port/system/console/jmx/org.apache.jackrabbit.oak%3Aid%3D6%2Cname%3D%22Consolidated+Cache+statistics%22%2Ctype%3D%22ConsolidatedCacheStats%22*

For the cache named **Document-Diff**, the hit rate should be over **.90**. If the hit rate falls below 90%, it is likely that you will need to modify your **DocumentNodeStoreService** configuration. Adobe product support can recommend optimal settings for your environment.

* The **Oak Repository Statistics** Mbean. It can be accessed directly by going to *http://server:port/system/console/jmx/org.apache.jackrabbit.oak%3Aid%3D16%2Cname%3D%22Oak+Repository+Statistics%22%2Ctype%3D%22RepositoryStats%22*

The **ObservationQueueMaxLength** section will show the number of events in Oak’s observation queue over the last hours, minutes, seconds and weeks. Find the largest number of events in the "per hour" section. This number needs to be compared to the **oak.observation.queue-length** setting which can be found in the **SlingRepositoryManager** component in the OSGi console. If the highest number shown for the observation queue exceeds the **queue-length** setting, contact Adobe Support for assistance with raising the setting. The default setting is 1,000, but most deployments usually need to raise it to 20,000 or 50,000.

## Publish Environment

### Performing Tests

The most important part of a deployment that needs to be subjected to load tests is the end user facing publish or dispatcher environment.

Third party automated testing tools can be used to test the performance of the website. These tools will allow you to record steps that users will go through on the site and run many of these sessions at the same time to simulate the load that is typical of a production website.

Most production websites have optimizations in place like dispatcher caching and a content delivery network in place. When testing, you need to make sure that these optimizations are available for the test environment as well. In addition to monitoring response times for the end users, you also need to monitor system metrics on the publish servers and dispatchers to ensure that the system is not constrained by hardware resources.

On a system that does not require a high level of personalization, the dispatcher should cache most requests. As a result, the load on the publish instance should remain relatively flat. If a high level of personalization is required, it is recommended to use technologies such as iFrames or AJAX requests for the personalized content so as to allow as much dispatcher caching as possible.

For basic tests, Apache Bench can be used to measure web server response times and help to create load for measuring things like memory leaks. For more see the example in the [Monitoring documentation](https://helpx.adobe.com/experience-manager/6-3/sites/deploying/using/monitoring-and-maintaining.html#main-pars_title_7-ztphmx-refd).

## Troubleshooting Performance Issues

After running performance tests on the author instance, any issues will need to be investigated, diagnosed and addressed. You can use several tools and techniques when performing analysis and addressing issues:

* You can inspect the [Request Performance log](https://helpx.adobe.com/experience-manager/6-3/sites/administering/using/operations-dashboard.html#main-pars_title_10) in the Operations Dashboard. This tool can be used to identify slow page requests
* Analyze slow running queries with the [Query Performance tool](https://helpx.adobe.com/experience-manager/6-3/sites/administering/using/operations-dashboard.html#main-pars_title_11)
* Watch the error lof for errors or warnings. For more information, see [Logging](https://helpx.adobe.com/experience-manager/6-3/sites/deploying/using/configure-logging.html)
* Monitor system hardware resources such as memory and CPU utilization, disk I/O or network I/O. These resources are often the causes of performance bottlenecks
* Optimize the architecture of the pages and how they are addressed to minimize the usage of URL parameters to allow for as much caching as possible
* Follow the [Performance Optimization](https://helpx.adobe.com/experience-manager/6-3/sites/deploying/using/configuring-performance.html) and [Performance tuning tips](https://helpx.adobe.com/experience-manager/kb/performance-tuning-tips.html) documentation
* If issues are present with editing certain pages or components on author instances, use the TouchUI Developer Mode to inspect the page in question. This will provide a breakdown of each content area on the page as well as its load time
* Minify all JS and CSS on the site. For more information on how to do this, see this [blog post](https://blogs.adobe.com/foxes/enable-js-and-css-minification/).
* Eliminate embedded CSS and JS from the components. They should be included and minified with the client-side libraries to minimize the number of requests required to render the page
* Use browser tools like Chrome’s Network tab to inspect the server requests and see which are taking the longest.

Once problem areas are identified, application code can be inspected for performance optimizations.  Any out of the box AEM features that are not performing properly can be addressed with Adobe Support.

**PERFORMANCE TREE**

The below diagram is meant to provide guidance on the steps that need to be taken in order to troubleshoot performance issues. It is split in 5 sections for easier reading.

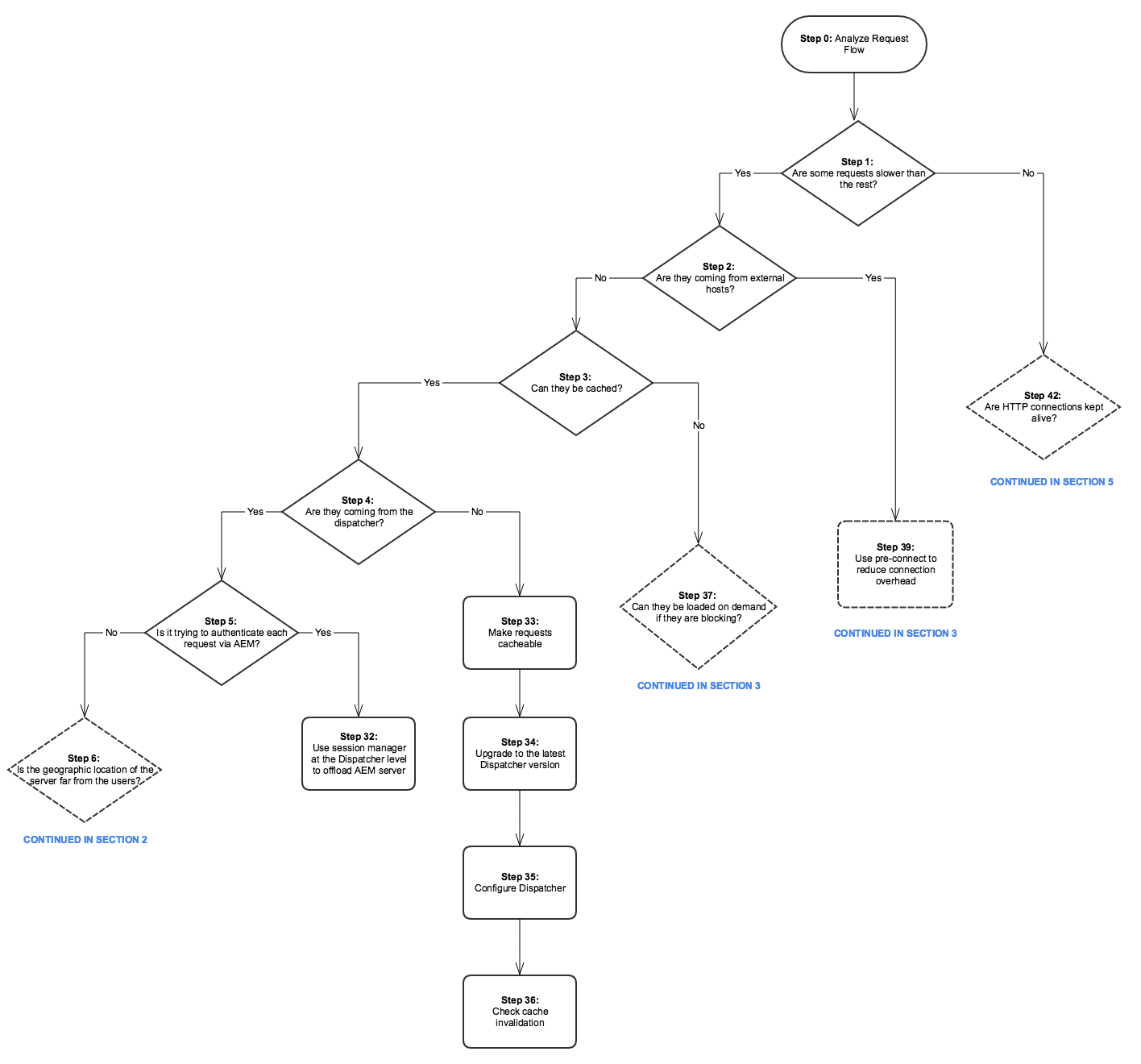
Each step in the diagram is linked to a documentation resource or a recommendation.

**Prerequisites and Assumptions**

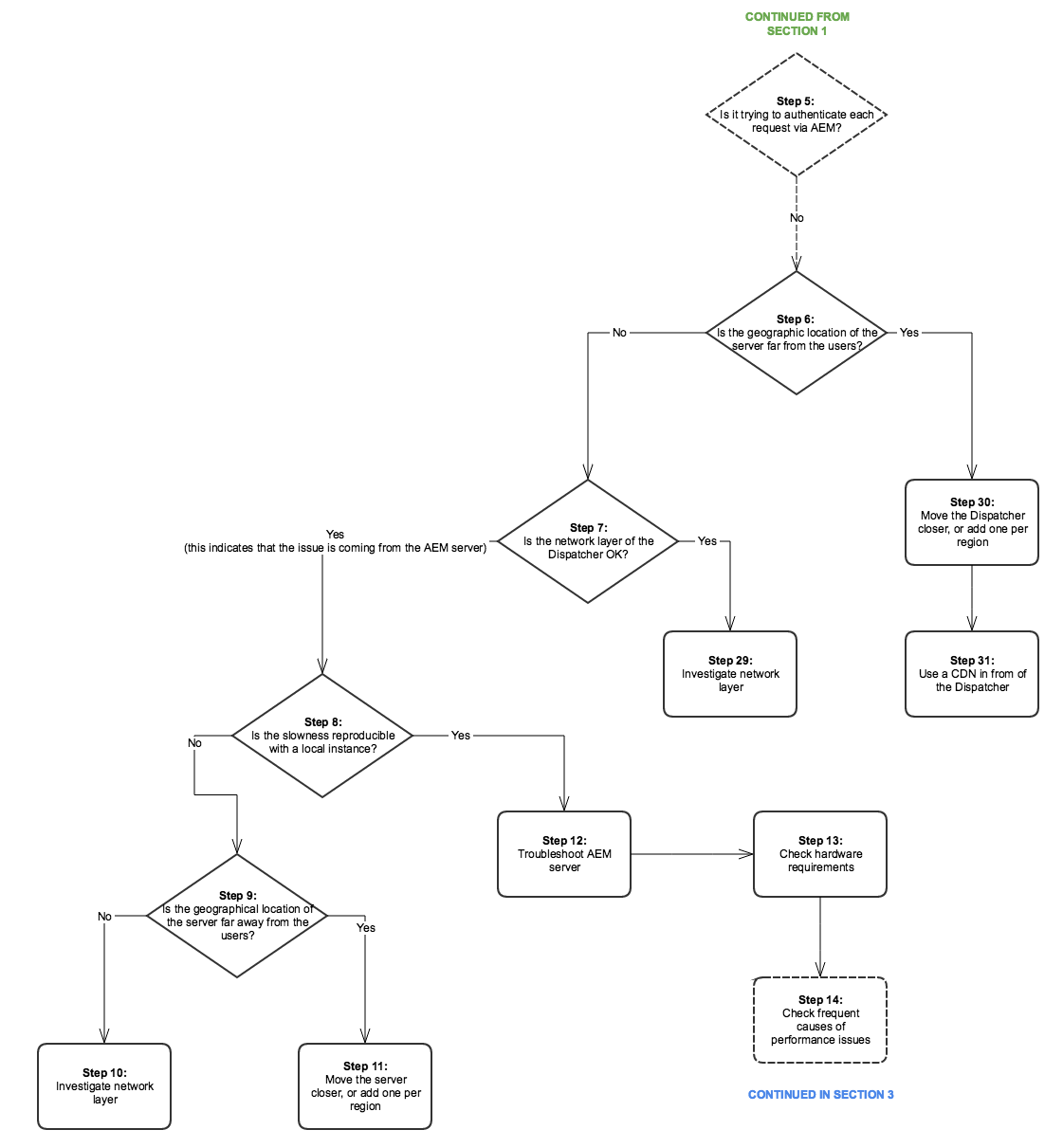
The assumption is that a performance issue is observed on a given page (either an AEM console or a web page) and can be reproduced consistently. Having a way to test or monitor the performance is a pre-requisite before starting the investigation.

The analysis starts at step 0. The goal is to determine which entity (dispatcher, external host or AEM) is responsible for the performance issue then determine which area (server or network) should be investigated.

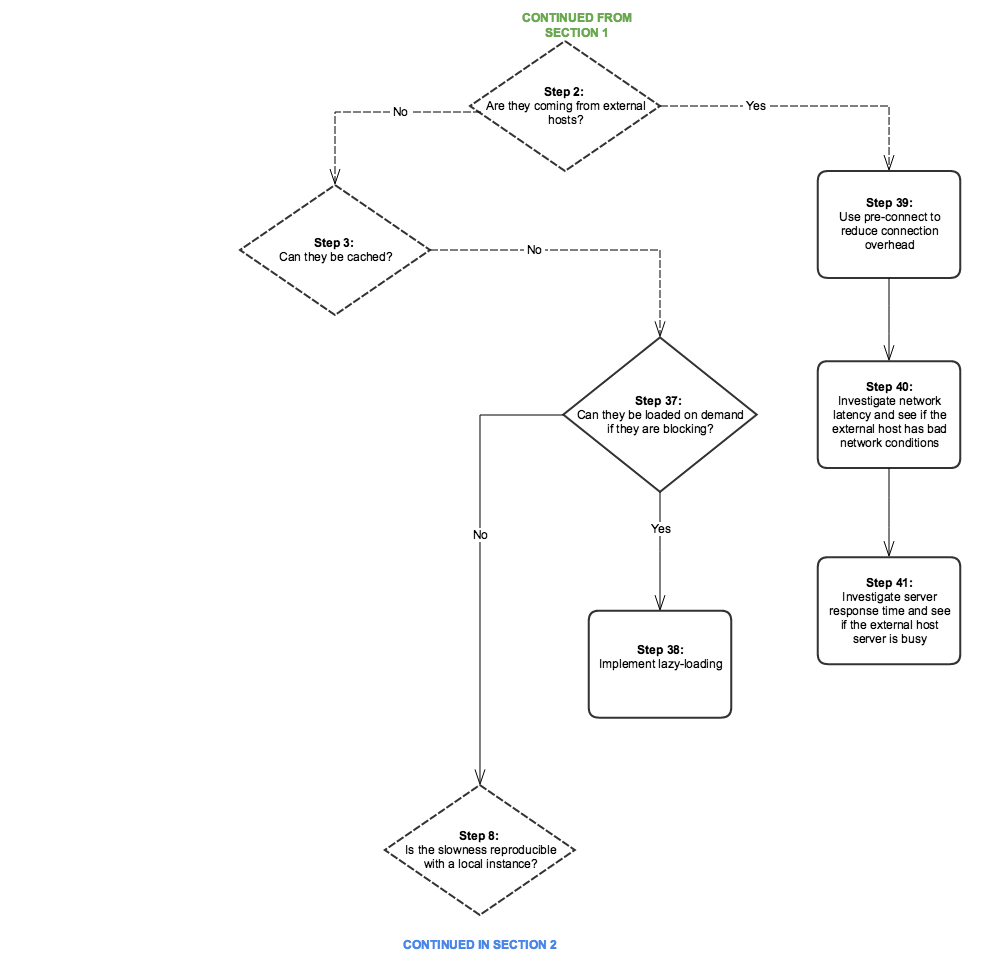
**Section 1**



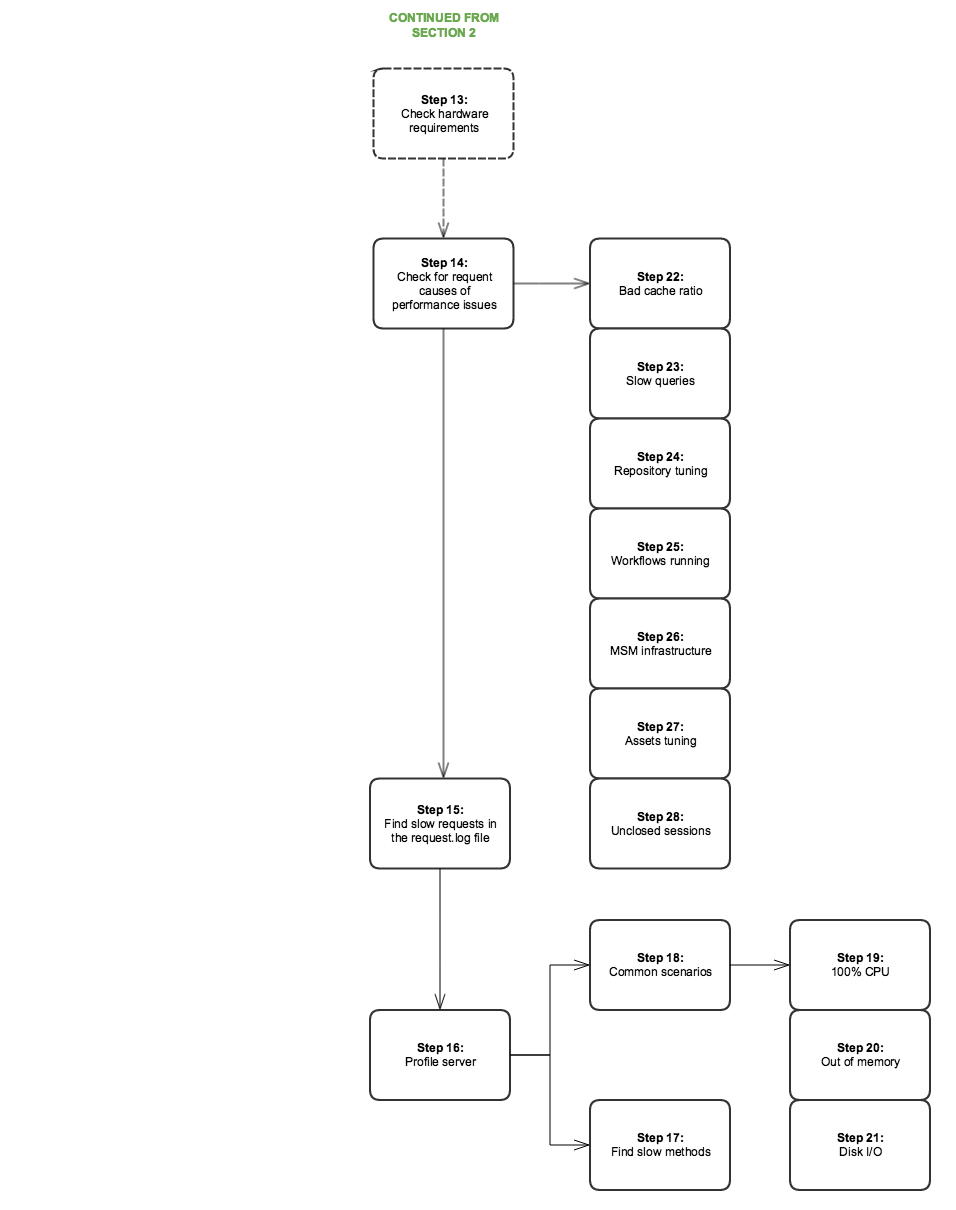
**Section 2**



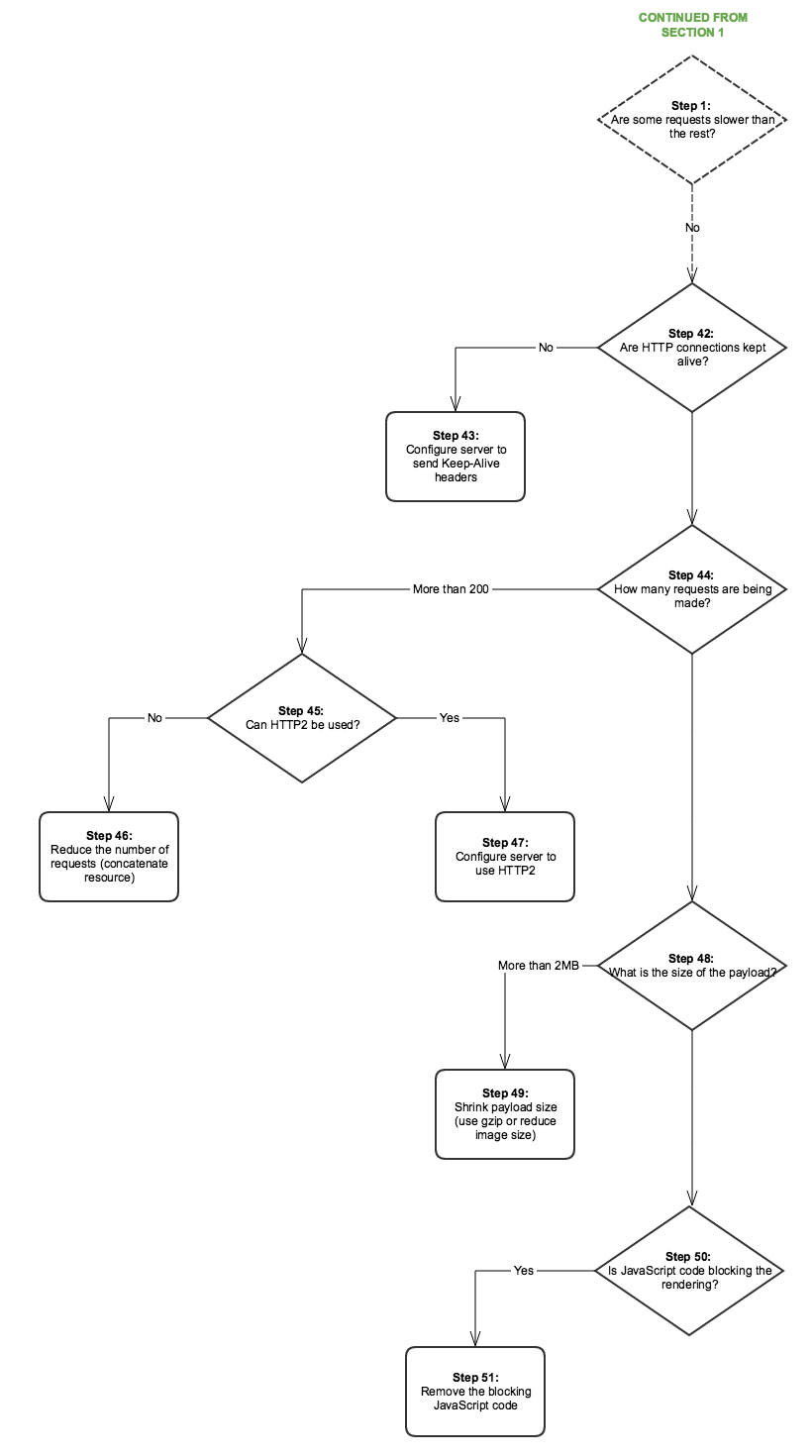
**Section 3**



**Section 4**



**Section 5**



**Reference Links**

|  |  |  |
| --- | --- | --- |
| **Step** | **Title** | **Resources** |
| **Step 0** | Analyze Request Flow | You can use standard HTTP request analysis in the browser to analyze the request flow. For more info on how to do this on Chrome, see:  <https://developers.google.com/web/tools/chrome-devtools/profile/network-performance/resource-loading><https://developers.google.com/web/tools/chrome-devtools/profile/network-performance/understanding-resource-timing> |
| **Step 2** | Are requests coming from external hosts? | You can use standard HTTP request analysis in the browser to analyze the request flow. See the above links on how to do this on Chrome. |
| **Step 3** | Can the requests be cached? | For more information on cacheable requests and general Dispatcher performance optimization advice, see [Dispatcher Performance Optimization](https://helpx.adobe.com/experience-manager/6-3/sites/deploying/using/configuring-performance.html#OptimizingPerformancewhenusingtheDispatcher). |
| **Step 4** | Are requests coming from the Dispatcher? | Check the [Dispatcher debugging documentation](https://helpx.adobe.com/experience-manager/dispatcher/using/dispatcher-configuration.html#Debugging) to see if the requests are cached properly. |
| **Step 5** | Is the Dispatcher trying to authenticate each request via AEM? | Check if the dispatcher sends **HEAD** requests to AEM for authentication before delivering the cached resource. You can do this by looking for **HEAD** requests in the AEM **access.log**. For more information, see [Logging](https://helpx.adobe.com/experience-manager/6-3/sites/deploying/using/configure-logging.html). |
| **Step 6** | Is the geographic location of the Dispatcher far away from the users? | Move the Dispatcher closer to the users. |
| **Step 7** | Is the network layer of the Dispatcher OK? | Investigate the network layer for saturation and latency issues. |
| **Step 8** | Is the slowness reproducible with a local instance? | Use [Tough Day](https://helpx.adobe.com/experience-manager/6-3/sites/developing/using/tough-day.html) to replicate "real world" conditions from the production instances. If this is not realistic for the slace of your development, make sure to test the production instance (or an identical staging one) in a different network context. |
| **Step 9** | Is the geographical location of the server far away from the users? | Move the server closer to the users. |
| **Steps 10 and 29** | Investigate network layer | Investigate the network layer for saturation and latency issues.  For the author tier, it is recommended that the latency does not surpass 100 milliseconds.  For more information on performance optimization tips, see [this page](https://helpx.adobe.com/experience-manager/kb/performance-tuning-tips.html). |
| **Step 11** | Move server closer or add one per region |  |
| **Step 12** | Troubleshoot AEM server | Check the following sub steps in the diagram for more information. |
| **Step 13** | Check hardware requirements | Check the documentation on [Hardware Sizing Guidelines](https://helpx.adobe.com/experience-manager/6-3/managing/using/hardware-sizing-guidelines.html). |
| **Step 14** | Check for frequent causes of performance issues |  |
| **Step 15** | Find slow requests | You can check for slow requests by analysing the **request.log** or by using **rlog.jar**.  For more information about using rlog.jar, see this page.  See [Using rlog.jar to find requests with long duration times](https://helpx.adobe.com/experience-manager/6-3/sites/deploying/using/monitoring-and-maintaining.html#Usingrlogjartofindrequestswithlongdurationtimes). |
| **Step 16** | Profile server | For information about profiling tools you can use with AEM, see [Tools for Monitoring and Analyzing Performance](https://helpx.adobe.com/experience-manager/6-3/sites/deploying/using/monitoring-and-maintaining.html#ToolsforMonitoringandAnalyzingPerformanc). |
| **Step 17** | Find slow methods in profiling |  |
| **Step 18** | Common scenarios of profiling | See [Analyzing Specific Scenarios](https://helpx.adobe.com/experience-manager/6-3/sites/deploying/using/monitoring-and-maintaining.html" \l "AnalyzingSpecificScenarios) in the Performance Optimization section. |
| **Step 19** | 100% CPU | <https://helpx.adobe.com/experience-manager/6-3/sites/deploying/using/monitoring-and-maintaining.html#MonitoringPerformance> |
| **Step 20** | Out of memory | 1. [Out Of Memory](https://helpx.adobe.com/experience-manager/6-3/sites/deploying/using/monitoring-and-maintaining.html#OutofMemory) 2. [My application throws out-of-memory errors](https://helpx.adobe.com/experience-manager/6-3/sites/deploying/using/troubleshooting.html) 3. [Analyze Memory Problems on Helpx.](https://helpx.adobe.com/experience-manager/kb/AnalyzeMemoryProblems.html) |
| **Step 21** | Disk I/O | See the [Disk I/O](https://helpx.adobe.com/experience-manager/6-3/sites/deploying/using/monitoring-and-maintaining.html#DiskI2FO) section in the Monitoring and Maintaining documentation. |
| **Steps 22 and 22.1** | Cache ratio | See [Calculating the Dispatcher Cache Ratio](https://helpx.adobe.com/experience-manager/6-3/sites/deploying/using/configuring-performance.html#CalculatingtheDispatcherCacheRatio). |
| **Step 23** | Slow queries | [Best Practices for Queries and Indexing](https://helpx.adobe.com/experience-manager/6-3/sites/deploying/using/best-practices-for-queries-and-indexing.html) |
| **Step 24** | Repository tuning | * [Performance Tuning Tips](https://helpx.adobe.com/experience-manager/kb/performance-tuning-tips.html) * [Configuring for Performance](https://helpx.adobe.com/experience-manager/6-3/sites/deploying/using/configuring-performance.html#ConfiguringforPerformance) * [Repository Performance Tuning](http://www.slideshare.net/jukka/repository-performance-tuning) |
| **Step 25** | Workflows running | * [Concurrent Workflow Processing](https://helpx.adobe.com/experience-manager/6-3/sites/deploying/using/configuring-performance.html#ConcurrentWorkflowProcessing) * [Configure the Queue for a Specific Workflow](https://helpx.adobe.com/experience-manager/6-3/sites/deploying/using/configuring-performance.html#ConfiguretheQueueforaSpecificWorkflow) * [Regular Purging of Workflow Instances](https://helpx.adobe.com/experience-manager/6-3/sites/administering/using/workflows-administering.html#RegularPurgingofWorkflowInstances) * [Transient Workflows](https://helpx.adobe.com/experience-manager/6-3/sites/developing/using/workflows.html?#TransientWorkflows) |
| **Step 26** | MSM Infrastructure | [Multi Site Manager Best Practices](https://helpx.adobe.com/content/docs/en/aem/6-3/administer/sites/msm/msm-bp.html) |
| **Step 27** | Assets tuning | 1. [Assets Synchronization Service](https://helpx.adobe.com/experience-manager/6-3/sites/deploying/using/configuring-performance.html#CQ5DAMAssetSynchronizationService) 2. [Multiple DAM Instances](https://helpx.adobe.com/experience-manager/6-3/sites/deploying/using/configuring-performance.html#MultipleDAMinstances) 3. Performance tuning tips articles [here](https://helpx.adobe.com/experience-manager/kb/performance-tuning-tips.html) and [here](https://helpx.adobe.com/experience-manager/kb/performance-tuning-tips.html). |
| **Step 28** | Unclosed sessions | [Checking for unclosed JCR sessions](https://helpx.adobe.com/experience-manager/6-3/sites/administering/using/troubleshoot.html#CheckingforunclosedJCRsessions) |
| **Step 30** | Move dispatcher closer (add one per "region"?) |  |
| **Step 31** | Use CDN in front of dispatcher | [Using Dispatcher with a CDN](https://helpx.adobe.com/experience-manager/dispatcher/using/dispatcher.html#UsingDispatcherwithaCDN) |
| **Step 32** | Use session management at the dispatcher level to offload AEM server | [Enabling Secure Sessions](https://helpx.adobe.com/experience-manager/dispatcher/using/dispatcher-configuration.html#EnablingSecureSessionssessionmanagement) |
| **Step 33** | Make requests cacheable | 1. [General Dispatcher Configuration](https://helpx.adobe.com/experience-manager/dispatcher/using/dispatcher.html) 2. [Configuring the Dispatcher Cache](https://helpx.adobe.com/experience-manager/dispatcher/using/dispatcher-configuration.html#ConfiguringtheDispatcherCachecache)   How to improve cache ratio; make requests cache-able (Dispatcher best practices)  Also, take into consideration the below settings in order to optimize your caching configurations   1. Set a no-cache rule for HTTP requesrst that are not GET 2. Configure query strings to not be cacheable 3. Do not cache URLs with missing extensions 4. Cache authentication headers (possible since Dispatcher version 4.1.10) |
| **Step 34** | Upgrade dispatcher version | You can download the latest Dispatcher version at this location:<https://www.adobeaemcloud.com/content/companies/public/adobe/dispatcher/dispatcher.html> |
| **Step 35** | Configure dispatcher | [Configuring the Dispatcher](https://helpx.adobe.com/experience-manager/dispatcher/using/dispatcher-configuration.html) |
| **Step 36** | Check cache invalidation | * [Cache Invalidation for the Author tier;](https://helpx.adobe.com/experience-manager/dispatcher/using/page-invalidate.html#InvalidatingDispatcherCachefromtheAuthoringEnvironment) * [Cache Invalidation for the Publish tier.](https://helpx.adobe.com/experience-manager/dispatcher/using/page-invalidate.html#InvalidatingDispatcherCachefromaPublishingInstance) |
| **Steps 37 and 38** | Lazy-loading | [See the Gem Session on AEM Web Performance.](https://docs.adobe.com/ddc/en/gems/aem-web-performance.html) |
| **Step 39** | Use pre-connect to reduce connection overhead | See the Gem Session indicated above. Also, additional documentation preconnect on W3c:<https://www.w3.org/TR/resource-hints/#dfn-preconnect> |
| **Steps 40 and 41** | External hosts latency and response time | Investigate the latency and response time for the external hosts. |
| **Steps 45 and 47** | Using HTTP/2 | See the Gem Session for steps 37,38 and 39. Also, check out [this](http://help-forums.adobe.com/content/adobeforums/en/experience-manager-forum/adobe-experience-manager.topic.html/forum__kdzc-does_anyoneknowwhe.html) forum post on HTTP/2 support. |
| **Step 49** | Shrink payload size | [Enable Gzip](https://helpx.adobe.com/experience-manager/6-3/sites/deploying/using/osgi-configuration-settings.html#AdobeGraniteHTMLLibraryManager) and [shrink the image size](https://docs.adobe.com/ddc/en/gems/aem-web-performance.html). |
| **Steps 42 and 43** | Keep-Alive | Is the **Keep-Alive** header present in the different requests to re-use connections? Otherwise, it would mean that each requests leads to another connection establishment, which introduces unnecessary overhead. (Standard HTTP request analysis in the browser)  You can check the [Proxy Server tool](https://helpx.adobe.com/experience-manager/6-3/sites/administering/using/proxy-jar.html) to check for Keep-Alive connections. |
| **Step 44** | How many requests are made? | Perform standard HTTP request analysis in the browser. |
| **Step 46** | Reduce number of requests | 1. Concatenate resources (images, CSS sprites, JSON, etc.) 2. Clientlibs embedding:    1. <https://helpx.adobe.com/experience-manager/6-2/sites/developing/using/clientlibs.html#CreatingClientLibraryFolders> - see heading Using embedding to minimize requests |
| **Step 48** | What is the size of the payload? | Standard HTTP request analysis in the browser |
| **Steps 50 and 51** | JS code blocking | <https://docs.adobe.com/ddc/en/gems/aem-web-performance.html> |

**SCALABILITY, PERFORMANCE & DISASTER RECOVERY - Whitepaper**

<https://helpx.adobe.com/content/dam/help/en/experience-manager/6-3/sites/deploying/using/performance/_jcr_content/main-pars/download-section/download-1/aem_scalability_whitepaperfinal-06122015je.pdf>