Logi Ad Hoc Reporting
Load Balancing Configuration

Version 12
July 2016

Table of Contents

Introduction

This document discusses most of the Ad Hoc configuration considerations for load balancing in “sticky session” and “non-sticky session” configurations. It does not cover other aspects of load balancing configuration or Ad Hoc configuration.

About Sticky Sessions

In a load balanced environment, there are essentially two configuration models regarding the processing of requests; sticky sessions (also referred to a “session affinity”) and non-sticky sessions.

In the sticky session configuration, the session is established and a server is “assigned” to process all of the requests for the life of the session. From Microsoft:

“The request forwarder filter is an ISAPI filter that monitors incoming requests and decides whether to allow requests to pass through and be executed locally or forward the request to another server, and if it must forward the request, it does the necessary preparatory work. It also gives out and interprets routing cookies, which allow for session coherency (sticky sessions) where a client is attached (stuck) to a single server for the lifetime of its browser session. “

In the non-sticky configuration, each request is independent of previous requests and may be routed to any of the servers in the server pool for processing.

From an Ad Hoc configuration point of view, little has to be done to accommodate the sticky session. Each session behaves much like a standalone web application. For the non-sticky environment, the primary considerations are the management of session state and centralizing the location of the data cache repository.

Configuration Details

An ASP.NET session is defined as the period of time a unique user interacts with a particular web application.

HTTP is a stateless protocol, in the sense that a Web server is concerned only with the current HTTP request for any given Web page. The server retains no knowledge of previous requests. The stateless nature of HTTP requests presents unique challenges when writing Web applications. ASP.NET applications that require session state to be maintained use session management techniques.

ASP.NET offers three options for the management of session state:

InProc	Session state is stored locally on the web server and is managed in the same worker process as the ASP.NET application.
StateServer	Session state is managed by ASP.NET state service, which runs outside of the ASP.NET worker process. The service can run local to the web server to support web gardens or on a different server to support web farms.
SQLServer	Session state is managed outside of the ASP.NET worker process and is stored in a SQL Server database. Like the StateServer option, this method can support web gardens and web farms.

Note: There may also be configuration considerations other than load balancing affecting the session state management decision.

Session State Management

Typically, by default, the web server is configured to manage session information as *InProc*. In both standalone and load balanced, sticky environments this setting allows a single server to manage the session information for the life of the session.

For non-sticky, load balanced configurations the session state needs to be centrally managed. Since requests can be processed by any of the servers in the pool, the servers need to access the state information for the session from a common location.

Either the *StateServer* or *SQLServer* session state management options may be configured to meet the goal of centralizing the session state information management.

Configuration of the session state options is found under the ASP.NET configuration dialog in IIS management. The details of configuring the session state options are outside the scope of this paper.

Centralizing Data Cache for Non-Sticky Configurations

The data cache repository is, by default, the `rdDataCache` folder in the Ad Hoc instance. In a standalone or sticky environment where all the requests are processed by the same server, the default cache configuration is sufficient.

In a non-sticky environment, centralizing the data cache repository is required.

In the `_Definitions/_Settings.lgx` file of each instance, there is a `<General>` element with a *DataCacheLocation* attribute. By default this is set to:

```
@Function.AppPhysicalPath~\rdDataCache
```

which identifies the path to the cache folder. Replace this attribute value with the path to a central data cache location in each `_Settings.lgx` file on each instance serving the Ad Hoc application.

The `_Settings.lgx` file is an XML file that can be edited with any plain-text editor, such as Notepad. If you also have Logi Info Studio product, you can use the Studio to open your Logi Ad Hoc application and edit the settings file.

General Load Balancing Configuration

The following should be configured as suggested:

Ad Hoc Instances – The Ad Hoc instances must be structurally replicated across all servers in the server pool. This includes any custom files, stylesheets, themes, etc.

Metadata Database – In a load balanced environment, all instances must be directed to a single metadata database. Using the Management Console, for each Ad Hoc instance click *Instance Configuration → Metadata Connection* to set and test the connection string to the metadata database.

Report Repository – A central repository for the report definition files is necessary in a load balanced environment. To set the report repository for each Ad Hoc instance, use the Management Console and click *Instance Configuration → Application Settings → Reporting Options*.

Dashboard Preferences – Centralizing the repository for the dashboard preference files is also necessary in a load balanced environment. To set the dashboard preferences repository for each Ad Hoc instance, use the Management Console and click *Instance Configuration → Application Settings → Reporting Options*.

Scheduling – The Logi Ad Hoc Scheduler service must be used in a load balanced environment. Using the Management Console, for each Ad Hoc instance click *Instance Configuration → Scheduling* and select the Logi Scheduling Service. Provide the *Server, Password*, and *Port Number* information to the same scheduler service.

Archiving – A centralized location for archived reports is necessary in a load balanced environment. Using the Management Console, for each Ad Hoc instance click *Instance Configuration → Archiving* and set the archive folder and URL for the archive repository.

Note: Consult the *Management Console Usage Guide* for additional information regarding the above configuration options.

Licensing – An appropriate license file (for the correct number of CPUs) must be placed in the root folder of each active server in the load balanced environment.

SecureKey Security – If the SecureKey authentication method is being used, requests are managed in Application scope and the "SecureKey Shared Folder" option must be configured in a load balanced environment:

In the `_Settings.lgx` file is a `<Security>` element. If its `AuthenticationSource` attribute is set to “SecureKey”, add a `SecureKeySharedFolder` attribute and set the value to a network path (e.g. `//mySharedServer/SecureKeyFolder`)

This must be added to the `_Settings.lgx` file for all Ad Hoc instances.

From our documentation:

Used only when AuthenticationSource="SecureKey", SecureKeySharedFolder allows SecureKey to work in clustered configuration with web farms and web gardens.

In a single-server configuration, SecureKey keeps SecureKey requests in Application state. With multiple servers, this information must be stored in files in this folder that is shared among the web servers. The account used by (or impersonated by) the web application must have network access rights to read, write and delete files from this folder.

Old files in the SecureKeyFolder are automatically deleted over time, so do not use this folder to store other files.

Contact Us

For more information about other Logi Analytics products or assistance beyond this user manual, please contact Logi Analytics in the following ways:

Corporate Headquarters

Phone: 1-888-LOGIXML (1-888-564-4965)
(703) 752-9700

Fax: (703) 995-4811

Email: info@logianalytics.com

Address: 7900 Westpark Drive, Suite A200
McLean, VA 22102

Web Site: www.logianalytics.com

Sales Department

Phone: 1-888-LOGIXML (1-888-564-4965)
(703) 752-9700

Email: sales@logianalytics.com

Customer Support

Phone: 1-888-LOGIXML (1-888-564-4965)
(703) 752-9700

Link: http://www.logianalytics.com/support/