CHAPTER 12

Configuring Application Acceleration and Optimization

Date: 4/10/09

With application acceleration and optimization features on ACE appliances, you can configure application delivery and application acceleration options that increase productivity and efficiency. The application acceleration features optimize network performance and improve access to critical business information. This capability accelerates the performance of Web applications, including customer relationship management, portals, and online collaboration by up to 10 times.

This section includes:

Note Application acceleration performance on the ACE appliance is 50 to 100 Mbps throughput. With typical page sizes and browser usage patterns, this equates to roughly 1,000 concurrent connections. Subsequent connections bypass the application acceleration engine. This limitation applies only to traffic that is explicitly configured to receive application acceleration processing (for example, FlashForward, Delta Optimization). Traffic that is not configured to receive application acceleration processing is not subject to these limitations. Also, because the ACE HTTP compression is implemented separately in hardware, it is not subject to these limitations. For example, if you have a mix of application-accelerated and non-application-accelerated traffic, the former is limited; the latter is not. If you have 50 Mbps of application-accelerated traffic, the ACE can still deliver up to 1.9 Gbps throughput for the non-application-accelerated traffic.

Optimization Overview

The application acceleration functions of the ACE appliance apply several optimization technologies to accelerate application performance. This functionality enables enterprises to optimize network performance and improve access to critical business information.

The ACE appliance provides the following application acceleration and optimization functionality:

- Delta optimization eliminates redundant traffic on the network by computing and transmitting only the changes that occur in a Web page between successive downloads of the same page or similar pages.
- FlashForward object acceleration technology eliminates network delays associated with embedded Web objects able to be cached. such as images, style sheets, and JavaScript files by placing the responsibility for validating object freshness on the ACE appliance, rather than on the client, making the client more efficient.
- Just-in-time object acceleration enables acceleration of non-cacheable embedded objects, resulting in improved application response time by eliminating the need for clients to download these objects on each request.
- Adaptive dynamic caching accelerates enterprise application performance and improves server system scalability by enabling the ACE appliance itself to fulfill requests for dynamic content, which offloads application servers and databases.

Refer to Configuring Application Acceleration and Optimization, page 12-1 or the *Cisco 4700 Series Application Control Engine Appliance Application Acceleration and Optimization Configuration Guide* for more information about application acceleration and optimization.

Related Topics

- Optimization Traffic Policies and Typical Configuration Flow, page 12-2
- Configuring Traffic Policies for HTTP Optimization, page 12-7
- Configuring Global Application Acceleration and Optimization, page 12-10

Optimization Traffic Policies and Typical Configuration Flow

To define the different optimization and application acceleration functions that you want the ACE appliance to perform, you must configure at least one each of the following:

- HTTP optimization action list—This action list specifies the actions that the ACE is to perform for application acceleration and optimization. You can configure action lists when configuring a virtual server, or as a separate procedure. See:
  - Configuring Application Acceleration and Optimization, page 4-42
  - Configuring an HTTP Optimization Action List, page 12-3
- Layer 7 server load-balancing class map—This class map identifies the Layer 7 server load-balancing match criteria to apply to incoming traffic, such as URL, HTTP cookie, HTTP header, or source IP address. See Configuring Virtual Context Policy Maps, page 11-30
- Layer 7 HTTP optimization policy map—This policy map applies the HTTP optimization action list and optionally an optimization parameter map to Layer 7 HTTP traffic. See Configuring Virtual Context Policy Maps, page 11-30.

- Layer 3 and Layer 4 class map—By using match criteria, this class map identifies the network traffic that can pass through the ACE appliance. The match criteria includes the VIP address for the network traffic. The ACE appliance uses these Layer 3 and Layer 4 traffic classes to perform server load balancing. See Configuring Virtual Context Policy Maps, page 11-30.
- Layer 3 and Layer 4 policy map—This policy map associates server load-balancing actions and HTTP optimization action lists with the VIP. See Setting Policy Map Rules and Actions for Layer 3/Layer 4 Network Traffic, page 11-38 and Configuring Traffic Policies for HTTP Optimization, page 12-7.
- Layer 7 server load-balancing policy map—This policy map specifies the server load-balancing actions that the ACE appliance is to perform. See Configuring Virtual Context Policy Maps, page 11-30.

You can also configure:

- Optimization parameter maps—Optimization parameter maps allow you to configure specific options for action list items. You can configure optimization parameter maps when configuring a virtual server or as a separate procedure.

  When you configure a parameter map with an action list for a class map, the ACE appliance validates the action list and parameter map configurations before deploying them.

  See:

  - Configuring Application Acceleration and Optimization, page 4-42
  - Configuring Optimization Parameter Maps, page 7-10.
- Global application acceleration and optimization options—The acceleration and optimization options allow you to apply specific acceleration and optimization features for logging and debugging on a global level on the ACE appliance. See Configuring Global Application Acceleration and Optimization, page 12-10.

Related Topics

- Configuring Traffic Policies for HTTP Optimization, page 12-7
- Optimization Overview, page 12-2
-

Configuring an HTTP Optimization Action List

An HTTP optimization action list groups a series of individual application acceleration and optimization operations that you want the ACE to perform.

Use this procedure to configure an HTTP optimization action list.

Tip You can also configure action lists when configuring a virtual server. For more information, see Configuring Application Acceleration and Optimization, page 4-42.

Procedure

Step 1 Select the item to configure:

- To configure a virtual context, select **Config > Devices >** ***context*** **> Expert > Optimization Action List**. The Action List table appears.
- To configure a configuration building block, select **Config > Global > All Building Blocks >** *building_block* **> Expert > Optimization Action List**.

Note The options in this procedure appear for ACE 4710-type configuration building blocks only.

Step 2 Click **Add** to add a new optimization action list, or select an existing action list, then click **Edit** to modify it.

Step 3 Configure the optimization action list using the information in Table 12-1.

Table 12-1 Action List Configuration Options

Field	Description
Action List Name	Enter a unique name for the action list. Valid entries are unquoted text strings with a maximum of 64 alphanumeric characters.
Enable Delta	Delta optimization dynamically updates client browser caches directly with content differences, or deltas, resulting in faster page downloads. Check this check box to enable delta optimization for the specified URLs. Clear this check box to disable delta optimization for the specified URLs. **Note** The ACE restricts you from enabling delta optimization if you have previously specified either Cache Dynamic or Dynamic Dynamic Entity Tag.
Enable AppScope	AppScope runs on the Management Console of the optional Cisco AVS 3180A Management Station and measures end-to-end application performance. Check this check box to enable AppScope performance monitoring for use with the ACE appliance. Clear this check box to disable AppScope performance monitoring for use with the ACE appliance.
Flash Forward	The FlashForward feature reduces bandwidth usage and accelerates embedded object downloading by combining local object storage with dynamic renaming of embedded objects, thereby enforcing object freshness within the parent HTML page. Specify how the ACE appliance is to implement FlashForward: • N/A—Indicates that this feature is not enabled. • FlashForward—Indicates that FlashForward is to be enabled for the specified URLs and that embedded objects are to be transformed. • FlashForward Object—Indicates that FlashForward static caching is to be enabled for the objects that the corresponding URLs refer to, such as Cascading Style Sheets (CSS), JPEG, and GIF files.

Table 12-1 Action List Configuration Options

Field	Description
Cache Dynamic	Check this check box to enable Adaptive Dynamic Caching for the specified URLs even if the expiration settings in the response indicate that the content is dynamic. The expiration of cache objects is controlled by the cache expiration settings based on time or server load. Clear this check box to disable this feature. **Note** The ACE restricts you from enabling Cache Dynamic if you have previously specified either Enable Delta or Dynamic Dynamic Entity Tag.
Cache Forward	Check this check box to enables the cache forward feature for the corresponding URLs. Cache forward allows the ACE to serve the object from its cache (static or dynamic) even when the object has expired if the maximum cache TTL time period has not yet expired (set by specifying the Cache Time-To-Live Duration (%): field in an Optimization parameter map). At the same time, the ACE sends an asynchronous request to the origin server to refresh its cache of the object. Clear this check box to disable this feature.
Dynamic Dynamic Entity Tag	This feature enables the acceleration of noncacheable embedded objects, which results in improved application response time. When enabled, this feature eliminates the need for users to download noncacheable objects on each request. Check this check box to indicate that the ACE appliance is to implement just-in-time object acceleration for noncacheable embedded objects. Clear this check box to disable this feature. **Note** The ACE restricts you from enabling Dynamic Dynamic Entity Tag if you have previously specified either Enable Delta or Cache Dynamic.

Step 4 Click:

- **Deploy Now** to deploy this configuration on the ACE appliance. The ACE appliance validates the action list configuration.
- Click **OK** to save your entries. This option appears for configuration building blocks.
- **Cancel** to exit this procedure without saving your entries.
- **Next** to save your entries and to configure another action list.

Related Topics

- Optimization Traffic Policies and Typical Configuration Flow, page 12-2
- Configuring Optimization Parameter Maps, page 12-6
- Configuring Traffic Policies for HTTP Optimization, page 12-7
- Configuring Global Application Acceleration and Optimization, page 12-10

Configuring Optimization Parameter Maps

Use this procedure to configure an Optimization parameter map for use with a Layer 3/Layer 4 policy map.

Optimization parameter maps can be configured for ACE appliances and ACE 4710-type configuration building blocks only.

Tip You can also configure optimization parameter maps when configuring a virtual server. For more information, see Configuring Application Acceleration and Optimization, page 4-42.

Procedure

Step 1 Select the item to configure:

- To configure a virtual context, select **Config > Devices >** ***context*** **> Load Balancing > Parameter Maps > Optimization Parameter Maps**.
- To configure a configuration building block, select **Config > Global > All Building Blocks >** *building_block* **> Load Balancing > Parameter Maps > Optimization Parameter Maps**.

The Optimization Parameter Maps table appears.

Step 2 Click **Add** to add a new parameter map, or select an existing parameter map, then click **Edit** to modify it. The Optimization Parameter Maps configuration screen appears.

Step 3 In the Parameter Name field, enter a unique name for this parameter map. Valid entries are unquoted text strings with no spaces and a maximum of 64 alphanumeric characters.

Step 4 Configure optimization using the information in Table 7-6.

Step 5 Click:

- **Deploy Now** to immediately deploy this configuration. The ACE validates the parameter map configuration and deploys it. This option appears for virtual contexts.
- **OK** to save your entries. This option appears for configuration building blocks.
- **Cancel** to exit this procedure without saving your entries and to return to the Parameter Map table.
- **Next** to accept your entries and to add another parameter map.

Related Topics

- Optimization Traffic Policies and Typical Configuration Flow, page 12-2
- Configuring an HTTP Optimization Action List, page 12-3
- Configuring Traffic Policies for HTTP Optimization, page 12-7
- Configuring Global Application Acceleration and Optimization, page 12-10

Configuring Traffic Policies for HTTP Optimization

Table 12-2 provides a high-level overview of the steps required to configure HTTP optimization on an ACE appliance.

Note Table 12-2 includes only the significant steps in each task. For detailed information on configuring these items, select the links provided, click **Help** in the ANM GUI, or refer to Configuring Traffic Policies, page 11-1.

Assumption

A virtual IP address has been configured for the context in which you configure HTTP optimization.

Table 12-2 Configuring Traffic Policies for HTTP Optimization

	Task	Procedure
Step 1	Create a Layer 7 class map for server load balancing.	1. Select **Config > Devices >** ***context*** **> Expert > Class Maps**. 2. Click **Add** to add a new class map. 3. In the Class Map Type field, select **Layer 7 Server Load Balancing**. 4. In the Match Type field, select the method the ACE appliance is to use to evaluate multiple match statements when multiple match conditions exist in the class map. 5. Click **Deploy Now**. 6. Configure match conditions for this class map. For more information, see: • Configuring Virtual Context Class Maps, page 11-6 • Setting Match Conditions for Layer 7 Server Load Balancing Class Maps, page 11-14
Step 2	Create an HTTP optimization action list to specify the optimization actions that are to be performed.	1. Select **Config > Devices >** ***context*** **> Expert > Action Lists**. 2. Click **Add** to add a new action list. 3. Configure the action list using the information in Table 12-1. 4. Click **Deploy Now**. For more information, see Configuring an HTTP Optimization Action List, page 12-3.

Table 12-2 Configuring Traffic Policies for HTTP Optimization (continued)

	Task	Procedure
Step 3	Create a Layer 7 HTTP optimization policy map and associate it with the server load-balancing class map in Step 1 and the action list configured in Step 2.	1. Select **Config > Devices >** ***context*** **> Expert > Policy Maps**. 2. Click **Add** to add a new policy map. 3. In the Type field, select **Layer 7 HTTP Optimization**. 4. Click **Deploy Now**. 5. In the Rules table, add the server load-balancing class map created in Step 1. 6. In the Action table, add the action list created in Step 2. For more information, see: • Configuring Virtual Context Policy Maps, page 11-30 • Setting Policy Map Rules and Actions for Layer 7 HTTP Optimization, page 11-53
Step 4	Create a Layer 3/Layer 4 class map for server load balancing.	1. Select **Config > Devices >** ***context*** **> Expert > Class Maps**. 2. Click **Add** to add a new class map. 3. In the Class Map Type field, select **Layer 3/4 Network Traffic**. 4. In the Match Type field, select the method the ACE appliance is to use to evaluate multiple match statements when multiple match conditions exist in the class map. 5. Click **Deploy Now**. 6. Configure Virtual Address match conditions for this class map. For more information, see: • Configuring Virtual Context Class Maps, page 11-6 • Setting Match Conditions for Layer 3/Layer 4 Network Traffic Class Maps, page 11-9

Table 12-2 Configuring Traffic Policies for HTTP Optimization (continued)

	Task	Procedure
Step 5	Create a Layer 7 policy map for server load balancing and associate it with the Layer 7 server load-balancing class map from Step 1.	1. Select **Config > Devices >** ***context*** **> Expert > Policy Maps**. 2. Click **Add** to add a new policy map. 3. In the Type field, select **Layer 7 Server Load Balancing**. 4. Click **Deploy Now**. 5. Associate the Layer 7 server load-balancing class map configured in Step 1 with this policy map by adding it to the Rule table. For more information, see: • Configuring Virtual Context Policy Maps, page 11-30 • Setting Policy Map Rules and Actions for Layer 7 Server Load-Balancing Traffic, page 11-56
Step 6	Create a Layer 3/Layer 4 network traffic policy map and associate it with the: • Layer 3/Layer 4 server load-balancing class map configured in Step 4 • Layer 7 server load-balancing policy map configured in Step 5 • HTTP optimization policy map configured in Step 3	1. Select **Config > Devices >** ***context*** **> Expert > Policy Maps**. 2. Click **Add** to add a new policy map. 3. In the Type field, select **Layer 3/4 Network Traffic**. 4. Click **Deploy Now**. 5. In the Rule table, add the Layer 3/Layer 4 server load-balancing class map configured in Step 4. 6. In the Action table, add the: – Layer 7 server load-balancing policy map created in Step 5 – HTTP optimization policy map created in Step 3 For more information, see: • Configuring Virtual Context Policy Maps, page 11-30 • Setting Policy Map Rules and Actions for Layer 3/Layer 4 Network Traffic, page 11-38

Related Topics

- Optimization Traffic Policies and Typical Configuration Flow, page 12-2
- Configuring an HTTP Optimization Action List, page 12-3
- Optimization Overview, page 12-2

Enabling HTTP Optimization Using Virtual Servers

Use this procedure to configure HTTP optimization using virtual servers.

Procedure

Step 1 Create a virtual server by following the instructions in Configuring Virtual Servers, page 4-2.

Step 2 Configure HTTP optimization by following the instructions in Configuring Application Acceleration and Optimization, page 4-42.

Related Topics

- Configuring Traffic Policies for HTTP Optimization, page 12-7
- Optimization Traffic Policies and Typical Configuration Flow, page 12-2

Configuring Global Application Acceleration and Optimization

Note This functionality is available for Admin contexts only and only on ACE appliances.

ANM allows you to configure global application acceleration and optimization options for logging and debugging as performed by the ACE appliance.

Procedure

Step 1 Select **Config > Virtual Contexts >** ***admin_context*** **> System > Application Acceleration And Optimization**. The Application Acceleration And Optimization configuration screen appears.

Step 2 In the Debug Level field, enter the maximum level of system log messages to be sent to the syslog server, using the values in Table 3-4. The severity level that you specify indicates that you want syslog messages at that level and the more severe levels. For example, if you enter 3 for Error, syslog displays Error, Critical, Alert, and Emergency messages.

Step 3 Check the AppScope Log check box to indicate that the ACE appliance is to upload optimization statistical log information to the optional AVS 3180A Management station. Clear the check box to indicate that the ACE appliance is not to upload this information.

Step 4 Click **Deploy Now** to immediately deploy this configuration on the ACE appliance.

Related Topics

- Optimization Overview, page 12-2
- Optimization Traffic Policies and Typical Configuration Flow, page 12-2