TECHNICAL OVERVIEW

CISCO WIDE AREA FILE SERVICES

Cisco® Wide Area File Services (WAFS) offers enterprises and organizations with multiple branch offices the benefits of centralized storage with local file services. Cisco WAFS helps enable companies to consolidate servers and storage and centralize backup and disaster recovery processes, while providing fast, near-LAN file access across the WAN.

The benefits of the Cisco WAFS solution include:

- *Lower total cost of ownership (TCO)*—Consolidate file and print servers, and remove unreliable tape backup at the branch offices.
- *Enhanced data protection*—Master copy of all files generated from the branch resides in the data center, providing better protection, simpler disaster recovery plan implementations, and enhanced management and usage of storage resources.
- *Reduced administration*—IT administrators can centrally manage file services such as usage quota, backups, disaster recovery, restores, access control, and security policies.
- *Global file sharing*—Cisco WAFS allows users to consistently share files across sites, increasing user productivity and facilitating distributed collaboration.

Cisco WAFS uses sophisticated protocol-level caching, compression, and network-optimization techniques to minimize the latency penalty associated with file-server access over the WAN. It helps ensure efficient operation of standard file-system protocols (Common Internet File System (CIFS) with Windows, Network File System with UNIX) over the WAN while preserving the protocol semantics including locking, coherency, security, and data integrity. The Cisco WAFS solution requires no software to be installed on client machines or file servers. Its operation is completely transparent to the end user and is seamlessly integrated into the existing networking and storage infrastructure.

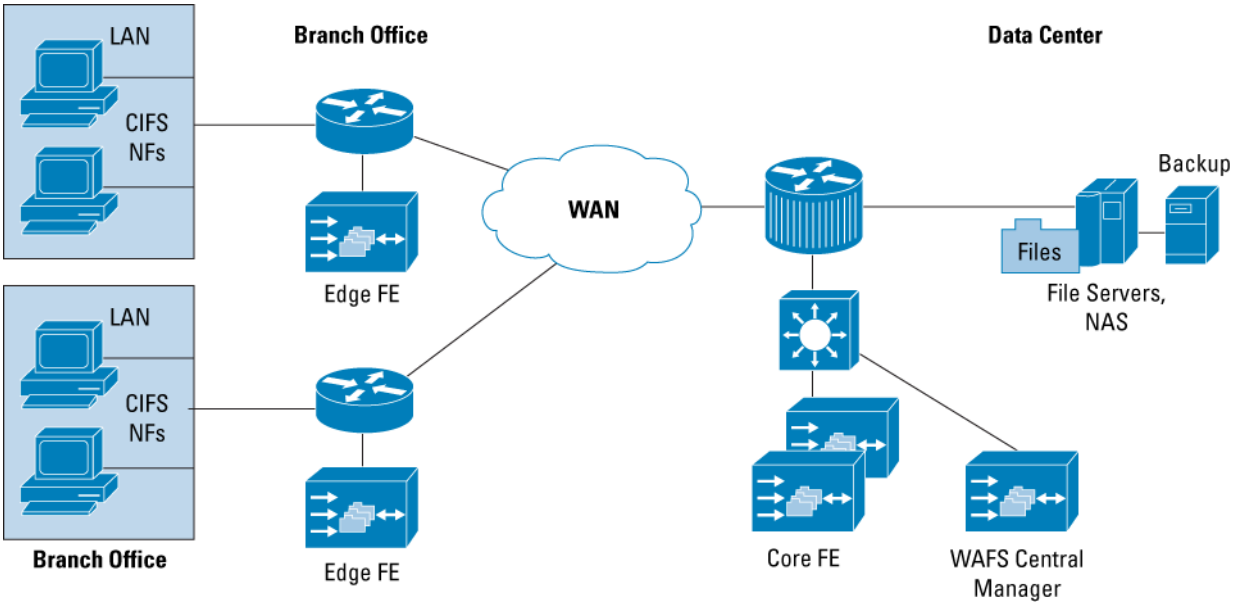
This document provides a high-level technical overview of the Cisco WAFS solution. It looks at the product architecture, the primary components and features, and the hardware platform, the File Engine. The information in this document is intended for overview purposes only. To learn more about specific product capabilities or limitations, please refer to the WAFS User Guide or contact Cisco Systems® directly.

CISCO WAFS DEPLOYMENT

The Cisco WAFS solution is deployed on Cisco File Engine appliances. Figure 1 describes the general deployment configuration. A WAFS deployment typically involves the following elements:

- *Edge File Engine*—A file-caching module deployed at each branch office or remote campus, replacing file and print servers and providing fast, near-LAN read and write access to a cached view of the centralized storage to local clients.
- *Core File Engine*—A colocated, server-side module that resides at the data center and connects directly to one or more file servers or network attached storage (NAS) servers, performing WAN-optimized file requests on behalf of the remote Edge File Engines.
- *WAFS Central Manager*—A management module that provides Web-based facilities for central management, configuration, monitoring, and maintenance operations of all the File Engines.

Figure 1. Cisco WAFS Deployment

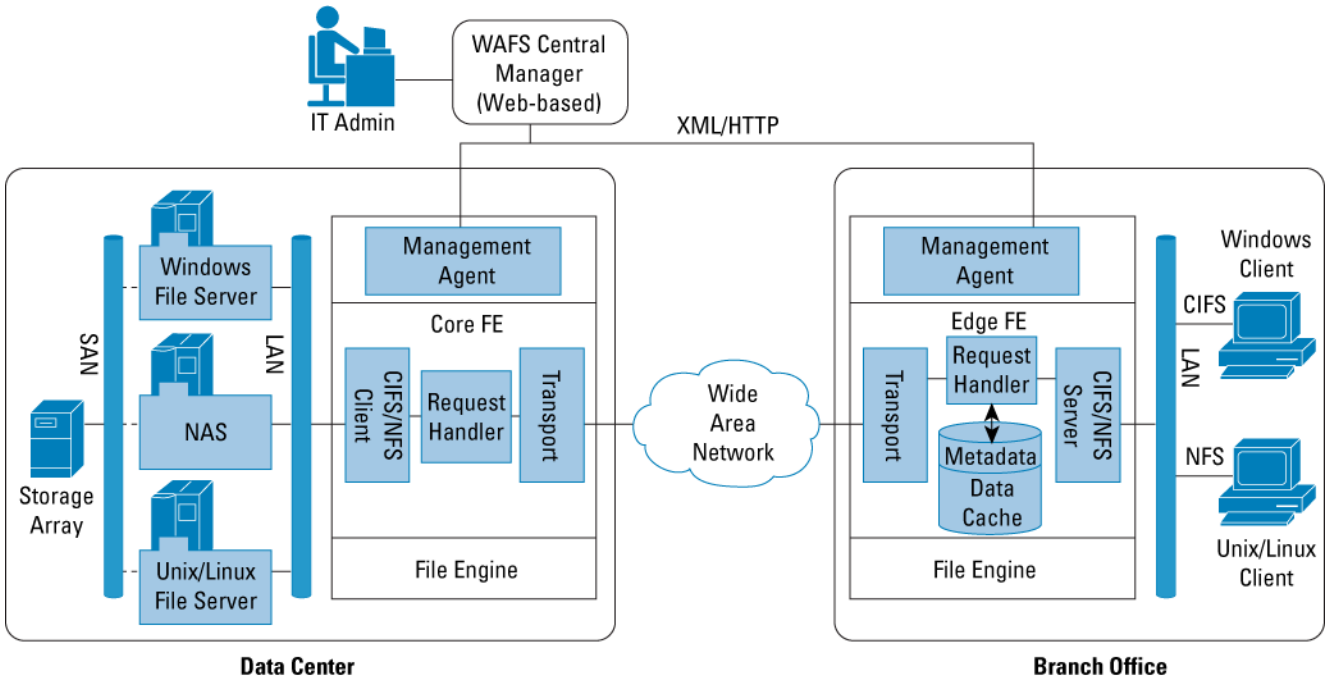


CISCO WAFS ARCHITECTURE

Figure 2 depicts the high-level architecture of Cisco WAFS. For ease of deployment and installation, each File Engine ships preloaded with WAFS software that allows it to run as an Edge File Engine, Core File Engine, or WAFS Central Manager. Administrators may choose to selectively activate one or more of the available modules (Edge File Engine, Core File Engine, or Central Manager). The term *WAFS gateway* often refers to an appliance configured with one or more active modules.

The central management module should be activated on a designated appliance. It is designed to control a logical group of gateways, regardless of their physical location.

Figure 2. Cisco WAFS Architecture



A typical Cisco WAFS deployment consists of a set of Edge File Engines that are distributed in remote sites, and one or more colocated, server-side Core File Engines that provide access to central file servers. From a logical viewpoint, Windows and UNIX users and applications work with the central file server using the standard CIFS and NFS protocols, respectively. They do this in the same manner that they access any other CIFS or NFS file server or NAS.

Physically, all file-server requests are first directed to the local Edge File Engine, which in turn determines whether to manage the request locally using its file system cache, or forward the request to the remote file server. In the latter case, the Edge File Engine encapsulates the CIFS or NFS request and uses the Cisco WAFS transport protocol to deliver the request efficiently over the WAN (TCP/IP) to the Core File Engine. The Core File Engine decodes the WAFS request into a standard CIFS or NFS request, issues the request to the actual file server, and then encapsulates the response back into the WAFS protocol for transmission to the Edge File Engine, and finally back to the CIFS/NFS client.

A central architectural aspect of Cisco WAFS is that it operates at the network file-server protocol layer (CIFS and NFS). That is, WAFS implements its own NFS and CIFS server and client layers (at the Edge File Engine and Core File Engine, respectively), and directly manages CIFS and NFS requests and responses. Specifically, for each CIFS session opened by a user, WAFS maintains a virtual channel to the file server on behalf of that user, which allows WAFS to support the full range of CIFS semantics.

An additional important aspect of the Cisco WAFS architecture is that an Edge File Engine behaves as a pure caching and acceleration system for remote file servers, as opposed to an actual file server. This has various important implications. For example, there is no need to define and manage users and permissions in WAFS, or to manage the capacity of the cache. There is also no need to back up the storage inside an Edge File Engine appliance, because it stores only transient copies of the original files. This greatly simplifies the operation and maintenance of WAFS, requiring minimal IT resources for system administration.

Component Connectivity Summary

The components of the WAFS solution are connected in the following manner:

- *Core File Engine, CIFS and NFS file server*—The Core File Engine component may be connected to multiple colocated file servers. It assumes LAN connectivity to these file servers. Core File Engines can be clustered for load balancing and failover purposes. After being defined, a cluster of Core File Engines is managed as a single "logical" Core File Engine for configuration purposes.
- *Edge File Engine, CIFS, and NFS clients*—Each Edge File Engine component can manage multiple concurrent CIFS and NFS clients. The Edge File Engine assumes LAN connectivity to its clients. Each Edge File Engine component supports both CIFS and NFS clients.
- *Core File Engine, Edge File Engine*—This is assumed to be a WAN connection. Each Core File Engine serves multiple Edge File Engines. An Edge File Engine can connect to either multiple Core File Engines that serve distinct file servers, or to a cluster of Core File Engines that serve a common set of file servers, or both.
- *Central Manager, Edge File Engine, Core File Engine*—A single manager controls all gateways in the enterprise.

Name Space

The Edge File Engine appears as a standard host on the LAN, and clients connect to the Edge File Engine to access remote file-server data. For NFS clients, the Edge File Engine is registered in the local Domain Name Server (DNS) tables, and it exports a mount point for each directory that is exported by each of the original file servers. NFS clients can then connect to the exported directories using the standard Mount protocol. The Edge File Engine is "virtually" mounted to the relevant directories at the original file server through the Core File Engine, which physically mounts the relevant directories from the original file server, again, using the standard Mount protocol.

For CIFS clients, a single Edge File Engine also represents multiple remote file servers. The Edge File Engine can use either WINS, DNS, or broadcast to register the logical file-server name. The logical names are mapped to the Edge File Engine host, which in turn is mapped to the proper file server at the data center, through the proper Core File Engine. Thus, a Windows client can browse the network neighborhood, discover the available (logical) file servers, and connect to a file server through the standard Session Setup CIFS command. As for the actual name of the logical file server, administrators can select any name as an alias to the origin file server, except the exact same name of the origin file server, so as to avoid potential name conflicts.* A common scenario is to name the logical file server after the name the old file server used to have in the branch; this provides for transparent migration into the central file server, where users continue to use the name of the local file server they used to have.

Global Name Space Using DFS and Network Drives

In certain deployments administrators want to provide a global name space, to allow all branches to share a common name for the file server. This scenario is particularly appealing in cases where users need to share documents across sites. The simple solution is to agree on a common network letter drive for the central file server. In each branch then, that network drive is mapped onto the logical file-server name (automatic mapping can be done using a central startup or login script).

An alternative solution is to use Distributed File System (DFS). DFS is a logical name space mechanism provided by Microsoft, which allows using logical tree of folders and files, which are mapped to physical file servers. Cisco WAFS supports DFS. Logical file servers can be registered as DFS replicas for a given logical folder, and using "site information," users are mapped onto their local replica to get file services. A side benefit of DFS, other than global name space, is that it provides for a simple failover—if the Edge File Engine fails, clients automatically use the second replica, which could be either another local standby Edge File Engine, or the original file server.

* In certain environments it is possible to use same names without leading to name conflicts. See the deployment guide for more details.

SYSTEM MAIN FEATURES

Read and Write Data and Meta-Data Caching

File caching is central to Cisco WAFS operation. The file cache module resides in the Edge File Engine, and it addresses most client requests (such as file read) locally without having to go over the WAN to the file server. For other commands (such as file write), the Edge File Engine can reply to the user request locally while issuing the command against the file server asynchronously, without affecting latency in user response. Some notable features of the cache include:

- The cache storage capacity is self-managed. When its storage fills up, it frees up disk space for new files by evicting from the cache files that have not been in recent use. Thus, cache storage is used only for the "working set" of files, which is typically a small fraction of the original repository.
- The metadata is kept in a dedicated file-based database optimized for fast I/O retrieval and is also cached in RAM, allowing fast response times for the frequently accessed metadata, including operations that involve mostly metadata, such as browsing, even when the actual file data is not in the cache.
- *Segment-based caching*—Cisco WAFS does not require the whole file to be cached to serve read requests to clients. In particular, it supports random access and responds to client request as soon as the requested byte range has arrived to the Edge File Engines. Similarly, upon writes, WAFS propagates to the file server only the segments of the file that have been actually updated, as opposed to sending the entire file. This allows WAFS to support applications that use very large files with small (even if frequent) changes.
- *Read-ahead caching*—In cases where a file is missing in the cache, when Cisco WAFS realizes that the application is performing sequential reads on a file, it predicts that the rest of the file will be read by the application and reads ahead the file in the background, thereby reducing the latency on subsequent reads.
- *Asynchronous buffered write*—Cisco WAFS supports efficient and consistent write operations. Efficiency is gained by performing buffered-write, which allows for the operations to proceed from the Edge File Engine to the Core File Engine (and thereafter to the file server) asynchronous from the client request. However, WAFS attempts to minimize the amount of cached updates that are uncommitted in the file server by continuously streaming updates to the Core File Engines. Most importantly, WAFS flushes all buffered data to the file server when a Close command is issued, to help ensure that the application is aware of any write errors that might have occurred at the file server (such as exceeding disk quota).
- *Negative caching*—Cisco WAFS implements negative caching by storing information about missing files in the cache, to avoid unneeded round trips over the WAN in cases it is known that the requested file does not exist. This feature is very useful for applications that repeatedly look for certain files in certain directories (such as configuration files) even though in most cases these files do not exist.

Optimizations of Protocol Signaling Operations

In addition to data and metadata caching, Cisco WAFS applies protocol-level caching of certain signaling messages, without compromising on protocol semantics. Examples include Microsoft Remote Procedure Call (MS-RPC) caching, Open caching, and Lock reduction techniques, which convert a set of locks into a single composite lock request.

Concurrency and Coherency

Locking and Share Modes

Cisco WAFS fully supports CIFS Open share modes by propagating them along with the File Open request to the file server. This allows WAFS to support correct intersite access behavior to shared files, when accessed by applications such as Microsoft Word and Excel. Furthermore, because share modes propagate all the way to the file server, files are also protected against concurrent access that is made by clients that are directly connected to the file server (not through the File Engine).

Similarly, all CIFS byte-range lock requests are passed through to the file server, further ensuring correct global behavior for applications that use file system locks. As for NFS, local locks are supported, that is, lock semantics are preserved for multiuser, intrasite access.

Coherency

The data cache within the Edge File Engine component serves as a temporary storage area for frequently accessed data. When a client requests a file or a block of data, the Edge File Engine component first checks its local cache. If the data requested exists locally and is considered "fresh," it will be served to the client immediately. If the data is missing or invalid (stale or expired), the latest data is retrieved from the remote server. The policies and algorithms involved in keeping the cached data fresh are referred to as *coherency*.

Cisco WAFS offers an "open-close" global coherency model, meaning that when a user opens a file, it is the latest version of the file since the last time this file was closed by any other user, including users from the same site, from different sites, or even users that access the file server directly.

In cases where the administrators know that certain files are only going to be used by users from a single site (no intersite file sharing will occur on these files or directories), Cisco WAFS offers an optional "local coherency" mode. This mode provides full coherency, but only within the site. In return, performance is improved, especially for applications that require extended "signaling" operations, such as Microsoft Access. Local coherency mode can be defined on any desirable granularity, including specific files, file types, directories, and up to the entire file server share.

Access Control and Authentication

As part of its end-to-end architecture, Cisco WAFS delegates access control and authentication decisions to the file server. Thus, while WAFS can cache results of previous resolutions for enhanced performance, it does not need to actually manage and maintain persistently these definitions.

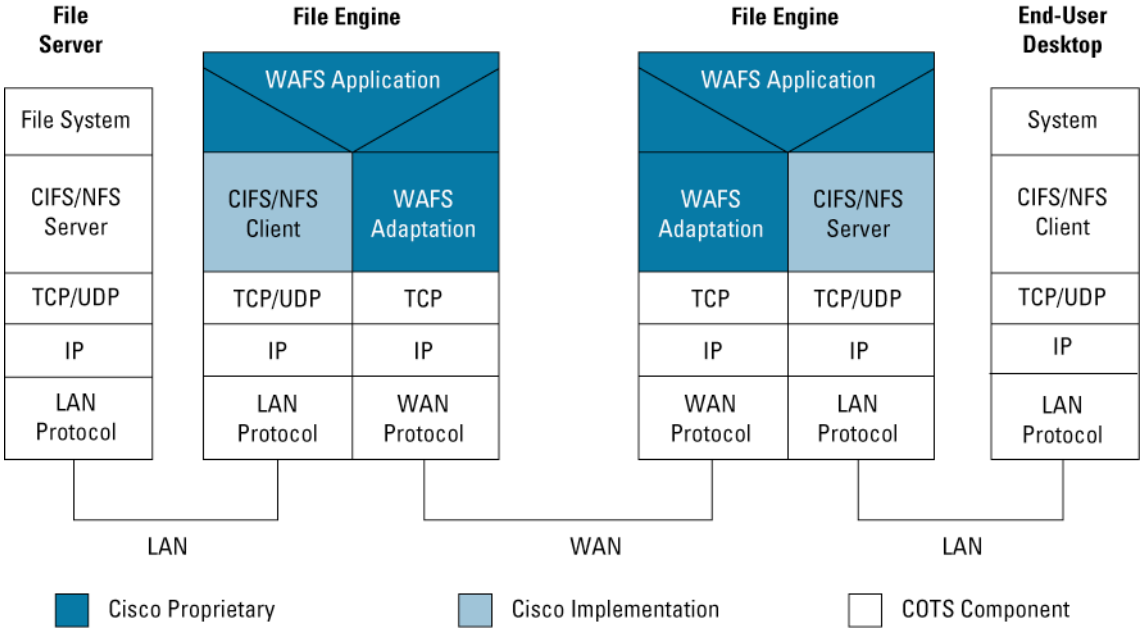
For CIFS, both user authentication (on Session Setup) and authorization (on each file Open request) are passed through and managed directly by the file server. Thus, there is no need to define new users or new permissions on specific files, greatly simplifying the configuration of Cisco WAFS.

Cisco WAFS also supports cross-domain authentication and authorization, in case a user defined in a given domain wants to access a file that resides in a different domain. In Windows, this is transparently managed through Windows' trusted-domain mechanism. In UNIX, WAFS allows optional cross-domain user and group mapping based on UNIX user directory services (such as NIS and YP), or by explicitly mapping all users of a given domain onto a specific UID-GID mapping in the domain of the file server.

WAN Transport Layer

Both NFS and CIFS are LAN-based protocols, not intended to operate over WAN links. Furthermore, most applications that access the file system assume LAN-based access as well, exacerbating the problem of file-system access over the WAN. To address this limitation, Cisco WAFS uses its own proprietary adaptation protocol layer between the Edge File Engine and Core File Engine over the WAN, while retaining the standard NFS or CIFS at the client and server ends. Figure 3 depicts the layered network architecture of WAFS.

Figure 3. WAN Adaption Layer



The following summarizes the protocol:

- The protocol is layered over TCP/IP, and requires only a single port per Core File Engine, making it a firewall-friendly protocol.
- Cisco WAFS uses a configurable number of concurrent TCP connections for each Edge File Engine-Core File Engine link. Requests and responses may be delivered across any open connection. In particular, multiple requests (and responses) for data delivery may be split across multiple connections to increase the effective usage of the network in case of high-latency or high-loss WAN connection, where TCP performance degrades. In addition, WAFS offers a bandwidth-control mechanism that prevents from WAFS from saturating the link when other applications need to share it.
- Protocol messages (both requests and responses) are compressed. Before compression, the message is encoded, allowing efficient delivery of both textual and binary data. The protocol layer, regardless of the message content, applies the compression automatically.
- As mentioned previously, Cisco WAFS pipelines the write operations so as to minimize the impact of latency on the throughput of the WAN. Similarly, WAFS pipelines Read requests by using larger buffers and TCP window sizes than the application.

Network Failure

The Cisco WAFS transport layer manages temporary network failures by reestablishing connections and retransmitting unresponded requests on the disconnected socket.

The transport layer notifies the Cisco WAFS application layer upon prolonged disconnects. After a “permanent disconnection” is detected, the system switches to disconnected mode and denies access to the remote file server, until reconnection occurs. Many client applications (such as Microsoft Word) can continue to work despite such disconnections, however, and automatically recover when reconnection occurs.

Management

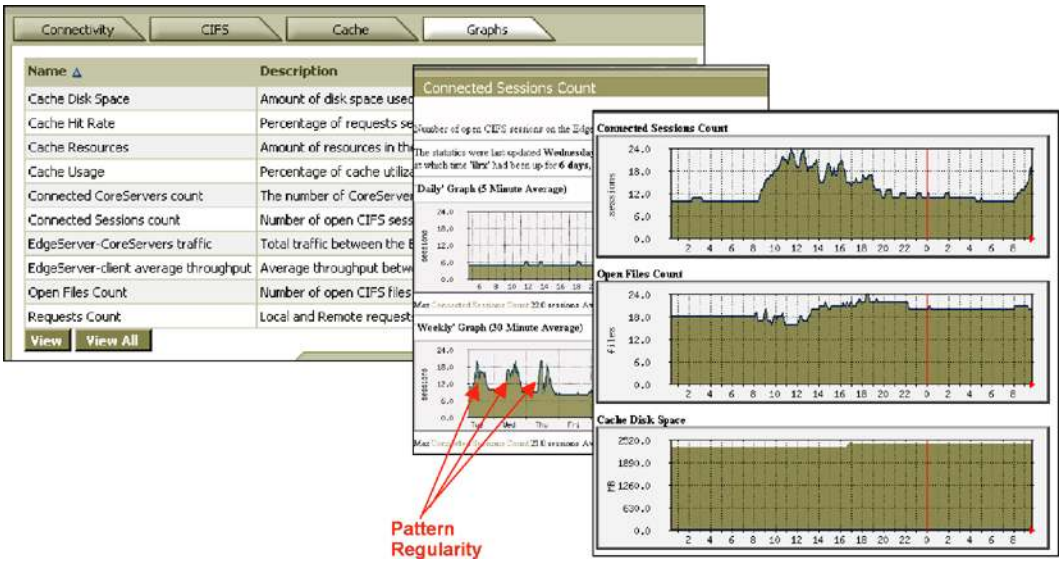
The Cisco WAFS Central Manager is responsible for central management and configuration of remote gateways, as well as monitoring their health and generating usage statistics reports based on logs collected from the gateways. Using a simple Web-based interface, the administrator edits and views configuration and policy information and receives usage reports from them. WAFS central management consists of two management applications: a *Gateway Manager* that runs on each gateway (both Edge File Engine and Core File Engine), and a *Central Manager*, which runs on a

single designated gateway. The WAFS Gateway Manager manages setup and local monitoring of the WAFS gateway. It contains a setup wizard for first-time setup, including connecting to the central manager, a logging agent, and a Simple Network Management Protocol (SNMP) agent that monitors an extensive set of system parameters and traps, which can be graphed. Figure 4 shows example graphs with Edge File Engine statistics.

The Cisco WAFS Central Manager (with a sample screen shown in Figure 4) plays two main roles:

1. Administrating the Cisco WAFS overlay network—the connections and relationships between the various gateways in a system. This includes linking an Edge File Engine to one or more Core File Engines, defining the parameters of their connection, and viewing the network of gateways.
2. Defining global policies that apply to a set of gateways. Specifically, administrators can define policies for setting the Coherency levels on specific files, and pre-position jobs.

Figure 4. Sample Edge File Engine Statistics



Pre-Position

Cisco WAFS offers customers the ability to pre-position or "push" files from the data center onto the Edge File Engines, to improve the cache hit. Using WAFS Central Manager, administrators can specify when to distribute files, what files to distribute, and to what destinations, with high flexibility in specifying these parameters. WAFS employs the transport layer for efficient transfer, transferring only missing blocks. In addition, WAFS can be constrained to push files only within a time window (for example, to avoid spillover of a file-distribution job into peak hours), as well as constrained by the amount of files to transfer (to avoid quick cache recycling).

Print Services

In addition to file services, Cisco WAFS also provides print services to branch users, eliminating the need for a separate print server in the branch. WAFS print services are comparable to standard Windows print services. They work with any printer (using raw mode—the client is in charge of rendering the data). They come with a library of print drivers that are automatically downloaded by clients as needed, and full print-queue management including job-status monitoring.

DESIGN FOR SCALABILITY AND RELIABILITY

Cisco WAFS has been architected for optimal scalability. Specifically, less than 20 percent of all client messages reach the Core File Engine, greatly offloading both the Core File Engine and the file server. Furthermore, the Core File Engine performs very simple tasks, mainly decoding the WAN transport layer. No caching is involved, and hence no state is preserved there. Furthermore, the stateless architecture of the Core File Engine makes it straightforward to cluster and load balance between multiple Core File Engines, hence providing near linear scalability by adding Core File Engine devices. At the edge, increased number of users can be facilitated through DFS, or using Web Cache Communication Protocol WCCP.

The stateless nature of the Core File Engine greatly improves the reliability of the system as well, because the failure of the Core File Engine does not result in loss of state. Hence, clustering for failover is simple. Edge reliability is achieved by closely monitored system watchdogs and a transactional metadata database that can recover cache state upon failure. Finally, high availability is provided through DFS or WCCP.

HARDWARE PLATFORM

Cisco Systems offers the Cisco WAFS solutions on prepackaged File Engine appliances available in multiple configurations. Please contact Cisco for the currently shipping configurations. World-class Cisco SMARTnet® service options are available for each File Engine appliance.

ADDITIONAL RESOURCES

For more information about the Cisco WAFS solution, please visit http://www.cisco.com or contact your local account representative.

Corporate Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 526-4100

European Headquarters
Cisco Systems International BV
Haarlerbergpark
Haarlerbergweg 13-19
1101 CH Amsterdam
The Netherlands
www-europe.cisco.com
Tel: 31 0 20 357 1000
Fax: 31 0 20 357 1100

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
www.cisco.com
Tel: 408 526-7660
Fax: 408 527-0883

Asia Pacific Headquarters
Cisco Systems, Inc.
168 Robinson Road
#28-01 Capital Tower
Singapore 068912
www.cisco.com
Tel: +65 6317 7777
Fax: +65 6317 7799

Cisco Systems has more than 200 offices in the following countries and regions. Addresses, phone numbers, and fax numbers are listed on **the Cisco Web site at www.cisco.com/go/offices.**

Argentina • Australia • Austria • Belgium • Brazil • Bulgaria • Canada • Chile • China PRC • Colombia • Costa Rica • Croatia • Cyprus Czech Republic • Denmark • Dubai, UAE • Finland • France • Germany • Greece • Hong Kong SAR • Hungary • India • Indonesia • Ireland • Israel • Italy • Japan • Korea • Luxembourg • Malaysia • Mexico • The Netherlands • New Zealand • Norway • Peru • Philippines • Poland • Portugal • Puerto Rico • Romania • Russia • Saudi Arabia • Scotland • Singapore • Slovakia • Slovenia • South Africa • Spain • Sweden • Switzerland • Taiwan • Thailand • Turkey • Ukraine • United Kingdom • United States • Venezuela • Vietnam • Zimbabwe

204103.15_ETMG_DB_01.05

Printed in the USA