IBM Software Group

z/OS® HTTP Server Performance Tuning and Troubleshooting

ON DEMAND BUSINESS

Agenda

- Exploring StandAlone and Scalable Server
- Timers
- Persistent and non-Persistent connections
- Running executables
- Caching
- Monitoring your Server
- Tracing and TroubleShooting

Scalable HTTP Server

- Benefits:
  - WLM workload balancing
  - Isolate certain types of workload using multiple ApplEnvs
  - Scales with workload. WLM will start additional QServers as per the defined policy.

Scalable continued…

- Disadvantages:
  - 2 or more Address Spaces, increasing paging or other resource issues.
  - Longer code path, 7.6% longer code path
  - Some extra work to install.

“To scale or not to scale, that is the question.”

- Use the StandAlone Server unless isolating workload or Dynamic Scaling is critical.
- Can implement WLM Transaction classification to manage performance using the Enclave directive.

Scaling a StandAlone Server

- Set shareport option on the TCPIP PORT statement.
  - This will allow more than one Standalone WebServer to read requests from the same port.
  - TCP/IP selects the listener with the least number of connections (both active and in the backlog) at the time the incoming client connection request is received.
  - **http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/BOOKS/f1a1b420/CONTENTS?SHELF=&DT=20020604120755#1.2.42**
- Use Sysplex Distributor to spray requests across multiple StandAlone HTTP Servers.
  - http://www-03.ibm.com/servers/eserver/zseries/library/techpapers/pdf/gm130165.pdf
- Asyncsockets

What are Asyncsockets?

- What are Asyncsockets? It is a socket that will allow a Client connection to be maintained without tying up a WorkerThread.
- Added in APAR PQ86769. White Paper at end of Presentation.
- Think of this as sockets being put in pools. When a Client connection is established it will be placed in a pool.
- Each WorkerThread will select a Client request(socket) from this pool and run the request.
- On completion of that request, if the connection should persist, the WorkerThread will place the socket back in the pool and select another piece of work, or socket to process.
- Number of available sockets limited by MAXSOCKET() setting in the AF_INET domain setting.

Asyncsockets

- **StandAlone**
  - Setup
    - AsyncSockets Yes
    - NO ApplEnv directives
    - Add “-SN any_string” to ICSPARM= in JCL
      - 1 to 32 character name for the application environment
      - The name cannot begin with the letters SYS.
- **Scalable**
  - Asyncsockets is the default when using a Scalable Server (with QServers).

Asyncsockets continued…

- White Paper “Benefit of Asynch I/O Support Provided in APAR PQ86769”
- http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP100719

Understanding Timers

- **InputTimeout:** Specify time allowed for the client to send a request
  - Beware of Denial of Service attacks. If InputTimeout is too high, it is easy to tie up all WorkerThreads waiting for client requests.
  - Recommended Setting: 3 seconds is a good value here, but may have to be tuned.
  - Referred to as Timer (1) in tracing
  - Current default is 30 seconds
  - Action upon timeout:
    - Socket is closed
    - WorkerThread returned to pool

Understanding Timers continued…

- **OutputTimeout:** Specify maximum time for sending output to the client
  - After a request has been completely read, the request then runs under this timer for the duration of the request.
    - The only exception is in the case of a CGI, where the OutputTimer will be turned off and the duration of the request will run under the ScriptTimer.
  - Recommended Setting: if serving files, 10 seconds is more than enough.

    If running GWAPI's you will have to tune based on testing.
  - Referred to as Timer (2) in tracing
  - Current default is 2 minutes
  - Action upon timeout:
    - Socket closed
    - Request is allowed to continue. Although a response will never be returned to the client.
    - WorkerThread will be returned to pool ONLY when request completes

Understanding Timers continued….

- **ScriptTimeout:** Specify time allowed for a CGI program to complete
  - Timer that runs during a CGI request.
  - Recommended Setting: don't have one, you will have to tune based on testing
  - Referred to as Timer (3) in tracing
  - Current default is 2 minutes
  - Action upon timeout:
    - Socket Closed
    - CGI is cancelled with a sigkill

Understanding Timers continued…

- **PersistTimeout:** Specify time to wait for the client to send next request
  - NOTE: a persistent connection ties up a WorkerThread, unless AsyncSockets is enabled.
  - Recommended Setting:
    - varies based on environment. Number of WorkerThreads, type of workload (SSL, NON-SSL), running AsyncSockets, etc.
    - EVERYONE should be running AsyncSockets…
    - Persistent connections are particularly important in SSL environments.
    - Real recommendations are hard to do here. Here are factors to consider:

  1. Set AsyncSockets.
  2. If SSL, what is your SSLV3Timeout and SSLV2Timeout settings?
  3. With AsyncSockets you should consider “Think Time” especially for SSL requests. Bottom line: Figure into the Timeout some average think time for a client’s next visit to the website. 5 minutes should certainly be near the ceiling.
  4. Understand the number of sockets that could be consumed in 5 minutes. Verify the setting of MAXSOCKET() in the TCP/IP AF_INET domain setting.

  - Current default is 5 seconds
  - Action upon timeout:
    - Socket is closed

Persistent vs non-Persistent Connections

- Persistent Connections: Client connections will remain open after the WebServer returns the response.
  - If the following is true, the WebServer will leave the Client Connection open:
    - Client requests a KEEP-ALIVE connection
    - WebServer is enabled for Persistent Connections
      - PersistTimeout
      - MaxPersistRequest
  - Persistent Connection benefits:
    - save the overhead of tearing down and creating a new socket for each request.
    - Speeds up client access time.
    - Will maintain an SSL session. If an SSL client session is moved from one socket to the next, an abbreviated handshake must be done. In an SSL environment this is a big saving.
  - Disadvantages:
    - Will tie up a WebServer WorkerThread for the duration of the persistence. This is a danger in a Denial of Service attack.
- You should setup Persistent connections to include no “think time”. Intended to speed up links in HTML pages, not wait for clients to read and hit enter.

CGI / FCGI / GWAPI

- Common Gateway Interface(CGI)
  - Runs in a separate address space spawned by the HTTP Server
  - Provides isolation of work from the HTTP Server
  - More expensive
    - Longer code path
    - Address Space create and cleanup for each run.
  - New Address Space may not run at same priority as HTTP Server
    - Check WLM policies

CGI / FCGI / GWAPI continued…

- Fast CGI
  - A CGI address space is started at HTTP initialization and remains until termination of HTTP Server
  - Prevents expense in creating and cleanup of Address Space.
  - Does not perform quite as well as a GWAPI, but offers isolation of Application code.

CGI / FCGI / GWAPI continued…

- Go WebServer Application Programming Interface (GWAPI)
  - Runs inside of HTTP Server
  - Shortest code path
  - Almost no setup
  - HTTP Server at risk of Application errors.
    - Some Application errors can cause termination of HTTP Server
  - Ties up WorkerThread for extended time. Same as CGI. Only difference is that CPU will run in the HTTP Server in the case of GWAPI.
  - Will increase storage footprint of HTTP Server.

Caching

- Improvements from caching depend on application design and client behavior.
- Things that will affect cache benefits.
  - Does the client have a local cache, browser cache for instance?
  - Are your clients repetitive users? If so, client cache will benefit you.
  - Do you often serve new users? This will drive up bytes served. Server side cache will benefit here.
  - Do responses have expiration dates? Expires: header, etc.
- HTTP Server provides a local cache function. Selected files are loaded into memory at initialization time. CacheLocalFile
- HTTP Server provides a TCP/IP FRCA cache mechanism

Caching continued… FRCA (1 of 2)

- Fast Response Cache Accelerator
- The WebServer provides exit code to TCPIP. This exit code is driven on each cache query. The exit code allows things like tracking incoming requests, for instance.
- This Cache will not prevent bytes served. Content will still be served, just at the TCPIP layer.
- Benefits:
  - Prevents wasted cycles in running each request though the HTTP Server
  - Prevents I/O required to read from the HFS.
- Pages cached at the TCP/IP layer. Pages cached in TCP/IP Data Spaces.
- See *PIU* for pages NOT eligible for caching. One such restriction is that only files which are not protected by a Protect directive can be cached.
  - TCPIP Fast Response Cache Accelerator Access Control allows you to protect cached objects using USERID and RACF(or comparable Security product). This was added in z/OS V1.4.
  - BEWARE: If you do not use FRCA Access Control, there is absolutely no authentication done on objects cached in FRCA. Even if your are using USERID %%CLIENT%% , there will never be any authentication done. Any client that can find your server, can access these files.

Caching continued… FRCA (2 of 2)

- FRCA is emptied when the HTTP Server is ended.
- Use MVS Display command to query FRCA statistics
  - D TCPIP,TCPCS,NETSTAT,CACHINFO
  - Can setup to run WLM Transaction classes to manage performance and reporting by WLM. FRCAWLMParms
  - FRCAAccessLog logs cache activity.

Monitoring your WebServer

- Use the MVS modify command to monitor WebServer Performance
  - F serverName,APPL=-D STATS
  - See PIU for details.

Note: That this just gave you a snap shoot at that time.

Monitoring your WebServer continued…

```
F <name_of_HTTP_Server>,APPL=-D STATS

IMW3502I Stats: Threads running: 125, Threads idle: 119,
   Requests: 3, Bytes rcvd: 21, Bytes sent: 7848,
   Active Inbound Connections: 6, Active Outbound Connections: 0.
   Connections since last SMF: 36,
   DNS Max: 0.000000, DNS Min: 0.000000, DNS Avg: 0.000000,
   Service Plugins Max: 0.000000, Service Plugins Min: 0.000000,
   Service Plugins Avg: 0.000000,
   CGI Max: 545.828640, CGI Min: 521.727259, CGI Avg: 533.315421,
   SSL Handshake Max: 0.000000, SSL Handshake Min: 0.000000,
   SSL Handshake Avg: 0.000000,
   Proxy Response Max: 0.000000, Proxy Response Min: 0.000000,
   Proxy Response Avg: 0.000000
   Non-SSL Waiting Threads: 58, SSL Waiting Threads: 61,
   Async I/O Waiting Threads: 0, Msg Queue Waiting Threads: 0,

Those fields "-d stat" fields Bytes rcvd:  and  Bytes sent: are defined as "unsigned int" which
   means the highest value carried before wrap is 4294967295 decimal or FFFFFFFF hexidecimal.
   One more unit will in effect set the field to zero.
```

TroubleShooting

HTTP Protocol

- What comes across the wire? Just bits and bytes.
- The HTTP Protocol tells us how to parse/handle those bytes.
  - All the headers flow first.
  - Each header is separated by a “carriage control/line feed”
  - When we reach 2 “carriage control/line feed”s the headers are done.
  - If a content-length header was sent, we assume the next data is the “message”/form data/content, whatever you want to call it. We will assume data stops after we have reached the content-length header value.
- We often see problems where the content-length doesn’t match actual data sent. This should not occur using a browser. Applets or other programming clients can cause this.
  - If data sent is less than content-length, hang conditions can occur.
  - If data sent is more than content-length, application errors occur.

TroubleShooting continued…

- Traces:
  - Can be turned on by adding parm to ICSPARM= in JCL. Required for initialization tracing.

Or

- MVS Modify command to turn on and off dynamically
  - **F serverNane,appl=-vv** to enable
  - **F serverName,appl=-nodebug** to disable
  - Verbose –v contains only the final request summary record
  - Very Verbose –vv most useful
  - Much to Verbose –mtv rarely required. Only enable at request of support.
- need to understand some basics of HTTP protocol
- Identify the beginning and end of a request.
- In a multi-tasking environment like HTTP server you have to isolate a thread

TroubleShooting continued…

ISOLATING A THREAD

- All traces start with a ThreadID that identifies a unique thread.
- Some traces will span multiple lines, but only the first line contains the ThreadID.
- Isolate the traces based on ThreadID.
  - Can use various editors. The trickiest part is picking up the entire trace in a multi-line trace.
  - We have a java tool written by Chad McDowell in our Level2 Group that has some nice features. We will see this later in the pitch.
    - You can download this tool from the WebSphere Application Server for z/OS Support Page

      http://www-06.ibm.com/software/webservers/appserv/zos_os390/support/
    - Search on the title or Document#
      - Trace reading tools for WebSphere and IBM Http Server on z/OS
      - Reference # **1231133**

Verbose tracing

- **Verbose Trace**
- -v Trace
- contains Timestamp information if Breadcrumb directive is NOT set to Off.
- Sample:

  E969A00 27/Jan/2006:14:29:34.221363]: V 9.37.238.141 8160 GET 19 200
  1138390149.724069 1138390174.210292 0000000000.000000 0000000000.000000
  1138390174.211095 1138390174.212735 0000000000.000000 0000000000.000000
  1138390174.215938 1138390174.220685 1138390174.221360
  /dpb/gwapi/readSTDIN
- First-Level Tracing Format

  The first-level (-v) trace is designed to provide basic tracing information on all requests without significantly impacting Webserver performance. The fields are designed to be automatically parsed for their information. The format of the trace is as follows:

Verbose tracing continued…

Chars	Field
-39	Thread Identifier and Timestamp :
40	Space
41	‘V' (indicates that this is a first-level tracing line)
42	Space
43-57	Client IP Address
58	Space
59-63	Port
64	Space
65-68	Request Method
69	Space
70-79	Content Length
80	Space
81-83	Response Code
84	Space
85-281	Event Timestamps
282	Space
283-on	Request URI

Verbose tracing continued…

The event timestamps are given in elapsed seconds since Jan 1, 1970. They are listed in entry/exit pairs and the elapsed time spent in each module is determined by calculating the difference between the timestamps. The 11 fields are as follows:

Acceptwait entry/exit
SSL Handshake entry/exit
Request Parsing entry/exit
CGI Exit entry/exit
Service Exit entry/exit
Request Completion Timestamp

Very Verbose tracing

Used to Isolate Thread function.

- -vv trace

BEGINNING:

- **Request headers:**
  - These headers give the Server information about the client and its capabilities.
  - Including headers that describe the USER to the Server, i.e.Cookie, Authorization, etc.
  - -vv trace will cut request header trace with “Client sez..”

Very Verbose tracing continued…

Sample:

[2665EE80 21/Nov/2007:11:42:59.856681]: Init Fn..... for module "HTDaemon" being called.

[2665EE80 21/Nov/2007:11:42:59.856701]: Init Fn..... not defined for module "HTTimer".

[2665EE80 21/Nov/2007:11:42:59.856712]: Init Fn..... for module "HTMemPool" being called.

[2665EE80 21/Nov/2007:11:42:59.856723]: Init Fn..... not defined for module "HTList".

[2665EE80 21/Nov/2007:11:42:59.856734]: Init Fn..... not defined for module "HTPresentation".

[2665EE80 21/Nov/2007:11:42:59.856746]: HTTimer... setting timer off->set (1) on socket 9.

[2665EE80 21/Nov/2007:11:42:59.856758]: HTTimer... set, old=0, cur=0, new=1

[2665EE80 21/Nov/2007:11:42:59.856780]: Client sez.. GET /posttest/index.html HTTP/1.0

[2665EE80 21/Nov/2007:11:42:59.856809]: Protocol version.... 1.0

[2665EE80 21/Nov/2007:11:42:59.856825]: Client sez.. User-Agent: ApacheBench/2.0.40-dev

[2665EE80 21/Nov/2007:11:42:59.856837]: User-Agent.. ApacheBench/2.0.40-dev

[2665EE80 21/Nov/2007:11:42:59.856849]: Client sez.. Host: 9.57.4.225:8000

[2665EE80 21/Nov/2007:11:42:59.856861]: Host..... 9.57.4.225

[2665EE80 21/Nov/2007:11:42:59.856872]: Host Port... 8000

[2665EE80 21/Nov/2007:11:42:59.856884]: Client sez.. Accept: */*

Very Verbose tracing continued…

END:

Response headers:

Describe the Content being returned

–vv trace will cut response headers following the “Header section for client” label.

Sample:

[2665EE80 21/Nov/2007:11:42:59.873618]: ..... Header section for client
HTTP/1.1 200 Document follows
Accept-Ranges: bytes
Content-Type: text/html
Content-Length: 154
Last-Modified: Tue, 10 May 2005 13:52:52 GMT
Content-Language: en
Server: WebSphere Application Server/5.1
Date: Wed, 21 Nov 2007 16:42:59 GMT
..... End of headers
[2665EE80 21/Nov/2007:11:42:59.873680]: Writer..... 249-byte block arriving.
[2665EE80 21/Nov/2007:11:42:59.873692]: Writer..... converting 0-byte header.
[2665EE80 21/Nov/2007:11:42:59.873706]: Writer..... 154-byte block arriving.
[2665EE80 21/Nov/2007:11:42:59.873717]: HTWriter.... buffering 154 bytes.
[2665EE80 21/Nov/2007:11:42:59.873729]: GWAPI: HTTPD_write()... successful

Additional Comments on Troubleshooting.

- For CGI issue, try to place debug statements in the application
  - As in print line statement

- For plug-in issue, enable plug-in tracing
  - Edit the plugin-cfg.xml file.

Change

<Log LogLevel**="Error"** Name="/WebSphere/V5R0M0/DeploymentManager/logs/http_plugin.log"/>

To

<Log LogLevel=**"Trace"** Name="/WebSphere/V5R0M0/DeploymentManager/logs/http_plugin.log"/>

Dependin g on your “RefreshInterval="60“ “, the tracing will be enable when the timer pop.
The default is 60 seconds

Documentation

HTTP Server Planning, Installing and Using
- http://www.elink.ibmlink.ibm.com/public/applications/publications/cgibin/pbi.cgi
- Search for **Publication SC34-4826-07**

HTTP Server DCF items
- Search from WebSphere Application Server for z/OS Support page at http://www-306.ibm.com/software/webservers/appserv/zos_os390/support/

HTTP Protocol

http://www.w3.org/

Select HTTP in left pane

Select RFC 2616

z/OS Books

http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/LIBRARY

Select Browse Shelves

Search for books using appropriate string

MVS

“Communications Server” is TCPIP

“UNIX System Services” is USS

“HTTP Server”

Benefit of Asynch I/O Support Provided with PQ86769

http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP100719

Trademarks, Copyrights, and Disclaimers

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

IBM	CICS		IMS		MQSeries	Tivoli	
IBM(logo)		Cloudscape		Informix	OS/390		WebSphere
e(logo)business		DB2		iSeries	OS/400		xSeries
AIX	DB2 Universal Database	Lotus		pSeries	zSeries		

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are registered trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a registered trademark of Linus Torvalds.

Other company, product and service names may be trademarks or service marks of others.

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements and/or changes in the product(s) and/or program(s) described herein at any time without notice. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectual property rights, may be used instead.

Information is provided "AS IS" without warranty of any kind. THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted, if at all, according to the terms and conditions of the agreements (e.g., IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.