

Web servers, web development environments, and GT.M

Chris Munt

M/Gateway Developments Ltd



Agenda

- Web Servers
 - Apache, Microsoft IIS, Sun JSWS
 - Development trends
 - Performance, Scalability and Security
- Extending Web Servers
 - Support for application development environments
- A role for GT.M ?

Evolution

- *“Since 1999, the web has grown from a document retrieval system into an application delivery system.”*
 - *Douglas Crockford at AjaxWorld 2008*
- Implications for:
 - Web Server extensibility
 - Resilience
 - Performance and scalability
 - Architecture
 - Security

Web Servers: Market Share

- 2005:
 - Apache 71%
 - IIS 20%
- 2008:
 - Apache 50% (Trend: falling slightly)
 - IIS 35% (Tend: increasing slightly)
- *Source: Netcraft*

Extending Web Servers: CGI

- Common Gateway Interface
 - Standard Interface
- Extensions implemented as stand-alone scripts/executables
- Application requests processed in separate server process
- Non-optimal but secure
 - Overhead of starting new process to serve each request; Overhead of inter-process communication
 - Application crash does not impact hosting web server
- Architecturally the same for all web servers

Extending Web Servers: Proprietary APIs

- Extensions implemented as dynamically loadable modules
 - UNIX Shared Objects/Libraries (SOs)
 - Windows DLLs
 - VMS Shareable images
- Common supported contexts
 - Filter (e.g. GZIP)
 - Extension (e.g. PHP, JSP)
- Application requests serviced in process space of web server host
- Optimal but there are risks
 - No overhead in process management and inter-process communication
 - Application crash can impact hosting web server

Apache

- v1.0 Consolidated in mid-90s
 - Robert McCool
 - Successor to NCSA HTTPd (National Center for Supercomputing Application)
- Modular architecture
- Multi-threaded for Windows; Multi-process for UNIX
- Extensibility
 - Common Gateway Interface (CGI)
 - Apache API

Apache: API

- Modules implemented as Dynamic Shared Objects (DSOs)
 - DLLS for Windows, Shared Objects/Libraries for UNIX, Shareable Images for OpenVMS
- Modules can process any phase of request/response cycle
 - Request handling (e.g. mod_csp, mod_weblink)
 - Authentication (e.g. mod_auth)
 - Security (e.g. mod_ssl)
 - Filters (e.g. mod_deflate)
 - Miscellaneous (e.g. mod_rewrite)

Apache: v2.0 - 2002

- Substantial rewrite of core with improved modularization
- Multi-threaded for Windows and OpenVMS
- Support for UNIX threading
 - Multi-process/Multi-threaded hybrid server
- New API
 - Incompatible with v1.3 API

Apache: v2.2 - 2005

- Improved Modules – particularly for authorization
- Same API but binary incompatible with earlier versions
 - Modules must be recompiled

Web Servers based on Apache

- HP Secure Web Server (HPSWS)
 - For OpenVMS
 - v1.3-1 based on Apache v1.3.26
 - v2.1-1 based on Apache v2.0.52
 - Tomcat (Java/JSP), Perl, PHP
 - CSP
- Many others

Apache: Resources

- www.apache.org
 - Access to various forums for support

Internet Information Services (IIS)

- v1.0 Introduced as free add-on for NT v3.51 (mid 1990s)
 - Single multi-threaded process
- Extensibility
 - Common Gateway Interface (CGI)
 - Internet Server Application Programming Interface (ISAPI)
 - Supported for all versions of IIS
 - The only web server API (excluding CGI) that's completely backwards (and forwards) compatible!

IIS v5.0 & Windows 2000

- Concept of isolation levels (or application protection levels).
 - Low
 - All in same process
 - Medium
 - ISAPI extensions run in a separate process
 - High
 - ISAPI extensions run in a separate process per ***application***
 - An ***application*** is broadly defined in terms of its path

IIS: Security alert: code red worm

- Exploiting a buffer overrun in the Windows 2000 Indexing Service DLL
 - GET /default.ida?NNNNNNNNNN
 - Inject code after provoking overrun
 - Microsoft reference: MS01-033
- Highlighted the serious consequences of using 'buggy' extensions

IIS v6.0 & Windows 2003

- Concept of Worker Process Isolation and Application pools
 - Application pool
 - Applications associated with one or more worker processes
 - Web Gardens
 - Multiple worker processes supporting an application
 - Not to be confused with web farms where multiple web server installations manage the work load
 - Process recycling
 - Process idle-time timeout

IIS v7.0 & Windows 2008

- Major upgrade
 - Previewed in Vista
 - Modular architecture
 - Administrators choose the modules required
 - Improved security
 - Application pools
 - New configuration schema. XML based.
- New API
 - ISAPI still supported (as a supplied module)
 - Third-party modules can be added

IIS Resources

- www.microsoft.com
- www.iis.net
 - Particularly interesting because IIS developers participate

Sun JSWS

- Netscape Enterprise (mid 1990s)
 - FastTrack – lightweight offering
- iPlanet (late 1990s)
 - America Online, Sun
- Sun ONE (early 2000s)
- Sun Java System Web Server (mid 2000s)
 - Resources: www.sun.com

Sun JSWS

- Multi-threaded for Windows
- Multi-process/Multi-threaded hybrid for UNIX
- Extensibility
 - Common Gateway Interface (CGI)
 - Netscape Server Application Programming Interface (NSAPI)
 - Modules implemented as DLLS for Windows, Shared Objects/Libraries for UNIX
 - Binary incompatibility between Netscape Enterprise v2 and v3
- Architecturally ahead of its time

Trends: All web servers

- Multi-process/Multi-threaded servers
- Increased flexibility/extensibility through modularization
- Better security

Impact of Web Standards

- Applications using AJAX techniques generate much more HTTP traffic than conventional web applications.
 - Poor performance unless connection between client and server is kept open between requests.
- HTTP v1.0: Asymmetric protocol by default
 - Client opens connection to server then sends request
 - Server sends response then closes connection to server

HTTP Keep-Alive

- Connection kept open between requests
 - HTTP v1.0: Default off
 - Connection: Keep-Alive (switch on)
 - HTTP v1.1: Default on
 - Connection: Close (switch off)
- Must send response size notification to client
 - Content-Length
 - Chunked transfer (HTTP v1.1)
 - Transfer-Encoding: chunked

HTTP Connections: The current status

- Standard specifies the maximum number of simultaneous connections to a given server
 - HTTP v1.0: usually 4
 - HTTP v1.1: always 2 (Section 8.1.4 RFC2616)
- Objective: to improve response times and avoid congestion.
- Can change setting in browser configuration
 - Inappropriate for web based applications
 - Proxys will implement standard

HTTP Connections: The future

- Now recognised that high-bandwidth connections are now commonplace
 - Key development since HTTP v1.1 which was drafted in January 1997.
 - Client-side bandwidth no longer gating factor in connection speed.
- IE v8 will almost certainly increase the number of connections to 6
 - Direct response to needs of AJAX applications

Security: High grade

- Two aspects
 - Client authentication
 - Protecting content
- SSL/TLS
 - Transport Layer Security (successor to SSL)
 - Usually authenticates server to a non-authenticated client
 - Certificate issued to server.
 - Client identifies him/herself using a username/password over the secure channel (all communications encrypted).
 - Supports mutual authentication
 - Certificate issued to both client and server
- Kerberos
 - Possible (apparently)

Security: Low grade authentication

- HTTP Basic authentication
 - Supported by all browsers
 - Client credentials passed to server in the clear
 - Really only useful for secured (internal) networks
- HTTP digest authentication
 - Protection for password
 - Some compatibility issues with browsers
 - Still not as strong as authentication over SSL/TLS or Kerberos.

Security: “through Obscurity”

- Most attacks on web servers are against known vulnerabilities
- Hide identity of web server
 - Use header masking facilities if they are available
 - Don't rely on standard error pages

Web Servers: Extending to support Applications

- Java/JSP
- ASP.NET
- PHP
- Python
- Ruby
- Perl

Each has a bewildering array of higher level development environments or 'frameworks'

A role for GT.M?

- As a browser?
 - No.
- As a web server?
 - No.
- As a backend database for existing development environments (PHP, JSP, Python etc ...)
 - Yes.
- As a development environment in its own right working directly to the web server?
 - Yes.

M/Gateway Service Integration Gateway (MGWSI)

- M Adapters/Interfaces for:
 - PHP (m_php)
 - JSP (m_jsp)
 - Python (m_python)
 - Ruby (m_ruby)
 - ASP.NET (m_aspx)
 - (Perl) (m_perl)
- A HTTP adapter for Apache (m_apache)
- A generic HTTP adapter based on CGI (m_cgi)
 - Should work with any web server

A role for GT.M

To be continued in the next talk ...

*“Options for web enabling GT.M via Apache
using MGWSI”*