Agenda

• Introduction to Apache 2.0
  • Apache 2.0
  • Worker MPM

• Configuration tips and tricks
  • Apache Tuning
  • Content Analysis
  • Kernel Tuning

• Performance measurement & Analysis
  • Performance Measurement
  • Performance Analysis

• Documentation / Links

• Q & A
Introduction to Apache 2.0
Demystifying Apache Performance
Performance Measurement & Analysis
Documentation / Links
Q & A
Background

- Awareness: Apache 1.3 vs. Apache 2.0
- How many are using Apache 1.3 vs. Apache 2.0
- Concerns about Apache 2.0
  - Stability ?
  - Performance ?
  - Why migrate ?
Introduction to Apache 2.0

- In the beginning
- Apache 2.0 comes along
- Worker MPM
In the beginning..

- Apache project started in 1995
  - Open Source!
  - Available on multiple platforms
  - A lightweight web server, designed for correctness
  - Instant hit among web administrators
  - Good enough for most users
  - ~70% of the websites run Apache

Potential Drawbacks

- Classic case of something that started small and grew big
- Apache 1.x: ‘A Patchy’ Web Server?
- Same processing model for all platforms
  - Win32, Linux & Unix are inherently different
Apache 2.0 comes along..

- Designed with **performance** as one of the key factors
- **Uses Multi-processing Models (MPM)**
  - Worker
  - Prefork
  - Winnt
  - New/experimental ‘event’ has been proposed
  - …(your own)
- **Uses Layered architecture**
  - New Filtering mechanism
  - Can serve different protocols (POP3, HTTP, FTP, SSL)
  - Apache Portable Runtime (APR)
Worker MPM

Most UNIX uses ‘worker’ model

- Hybrid multi-threaded multi-process model
- Fixed # of threads per process
- Threads Vs Processes
  - Threads are light-weight than forking new processes
  - Thread context is smaller than process context
  - Fewer context switches in kernel (user-space thread scheduling when possible)
Worker MPM (contd.)

- **Worker MPM is scalable**
  - Works great on multi-processor systems
  - Can easily handle multiple 1000’s of connections/sec
  - Low Memory foot-print
    - Memory is no longer a limitation
    - Worker MPM consumes less memory than Apache 1.3 or Pre-fork MPM

- **Why worker MPM may not be the best**
  - 3rd party add-on modules are not thread-safe
  - Takes time to port the in-house developed modules to Apache 2.0 style and make them thread-safe
  - Users need not be limited to just the worker MPM
    - Re-build apache with a different MPM

- **Default configuration rarely works well right out of the box**
  - Minor configuration changes results in huge performance benefits
Demystifying Apache Performance:

• **Apache Tuning**
  – Compile-time optimizations
  – Choose your Apache modules
  – Using directives

• **Content Analysis**
  – Use Cache/Content Accelerators
  – Load Balancing
  – Authentication/Authorization Modules

• **System Tuning**
  – Choose your hardware
  – Kernel tuning
  – Network tuning
Apache Tuning: Compile-time Optimizations

• Compiler Options
  – Optimization: CC -O3
  – 32-bit Vs 64-bit

• Apache ‘configure’ Options
  – Choose your MPM
    • Worker is ideally-suited for most UNIX flavors
  – Choose the default locking mechanism
    • pthread mutex locks > sem locks
    • Use atomic locks if available
  – Bring-in extra modules only if required
Apache Tuning: Choose your Apache modules

- **New modules in Apache 2.0:**
  - Dynamic Content: `mod_cgid`
    - Daemon process to handle cgi requests, light weight and faster than `mod_cgi`
  - Caching: `mod_disk_cache`, `mod_mem_cache`
    - Great for serving images, Static HTML content
  - Security: `mod_auth_ldap`, `mod_auth_digest`, `mod_ssl`
    - Includes Session Caching by default
  - Proxy: `mod_proxy`, `mod_proxy_http`
  - Compression: `mod_deflate`
Apache Tuning: Using directives

- Lots of details on the Apache website
  - [http://httpd.apache.org/docs-2.0/misc/perf-tuning.html](http://httpd.apache.org/docs-2.0/misc/perf-tuning.html)

<table>
<thead>
<tr>
<th></th>
<th>Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Global</strong></td>
<td></td>
</tr>
<tr>
<td>Keepalive</td>
<td>On</td>
</tr>
<tr>
<td>HostNameLookup</td>
<td>Off</td>
</tr>
<tr>
<td>ExtendedStatus</td>
<td>Off</td>
</tr>
<tr>
<td><strong>URL Handling</strong></td>
<td></td>
</tr>
<tr>
<td>FollowSymLinks</td>
<td>Enable globally</td>
</tr>
<tr>
<td>SymLinksIfOwnerMatch</td>
<td>Enable on a per-directory basis</td>
</tr>
<tr>
<td>AllowOverride</td>
<td>None</td>
</tr>
<tr>
<td>DirectoryIndex</td>
<td>List index files instead of using wild-cards like ‘index’</td>
</tr>
<tr>
<td><strong>Content</strong></td>
<td></td>
</tr>
<tr>
<td>EnableMMap</td>
<td>On (if mmap is faster than read)</td>
</tr>
<tr>
<td>EnableSendfile</td>
<td>On (if sendfile is available)</td>
</tr>
<tr>
<td><strong>MPM Specific</strong></td>
<td></td>
</tr>
<tr>
<td>ServerLimit</td>
<td>25</td>
</tr>
<tr>
<td>ThreadLimit</td>
<td>40</td>
</tr>
<tr>
<td>ListenQueue</td>
<td>1000</td>
</tr>
<tr>
<td>AcceptMutex</td>
<td>pthread</td>
</tr>
</tbody>
</table>
Apache Tuning: Using directives (contd.)

- Threads vs. Processes – what is the correct combination
  - Process context-switches are more expensive than a thread context-switches
  - # of threads depends upon the number of CPU’s available
    - Use Processor binding when possible
  - **Default:** ThreadLimit is 50 and ServerLimit is 16
  - **Study:** A maximum of 40 threads per process gives optimal performance on a HP-UX 2-way box
Demystifying Apache Performance

- **Apache Tuning**
  - Compile-time optimizations
  - Choose your Apache modules
  - Using directives

- **Content Analysis**
  - Use Cache/Content Accelerators
  - Load Balancing
  - Authentication/Authorization Modules

- **System Tuning**
  - Choose your hardware
  - Kernel tuning
  - Network tuning
Content Analysis: Using Cache/Content Accelerators

• **Use Cache Accelerators**
  - Static content: mod_disk_cache
  - SSI content: mod_mem_cache
  - Alternative: Network Server Accelerator for HP-UX 11i

• **Use Content Accelerators**
  - Great for both Static and Dynamic content
  - Apache Module: mod_deflate
  - Alternative: 3rd Party plug-in: mod_gzip
    [http://webcompression.org/gzip-compress.html](http://webcompression.org/gzip-compress.html)
Content Analysis: Load Balancing

- **Load Balancing**
  - Apache modules: mod_rewrite, mod_proxy

- **URL rewriting**
  - Often used as a hammer to squash an ant
  - Can do simple load-balancing, and lots of other cool stuff

- **Proxy**
  - Use Caching
  - Light weight, and very effective

- **Misc.**
  - Deny access to well mannered spiders or web crawlers by creating a robots.txt file
Content Analysis: Authentication/Authorization Modules

- Security Modules are inherently slower
- Take advantage of the new modules
  - mod_ssl, mod_auth_ldap, mod_auth_digest
  - Use Session caching when possible
  - Use Auth checking as a point for Load Balancing
    - Ex. Japanese company uses Apache as a SSL Proxy server to verify Client certificate, and then re-direct the requests using mod_proxy to backend servers listening HTTP.
- Specify configuration in httpd.conf rather than .htaccess
Demystifying Apache Performance

• Apache Tuning
  – Compile-time optimizations
  – Choose your Apache modules
  – Using directives

• Content Analysis
  – Use Cache/Content Accelerators
  – Load Balancing
  – Authentication/Authorization Modules

• System Tuning
  – Choose your hardware
  – Kernel Tuning
  – Network Tuning
System Tuning: Choose your hardware

- Smaller/In-expensive systems
  - Easier to replace/scale
- SSL traffic
  - Accelerator Cards Vs Itanium
    - Itanium has special instructions for handling SSL Crypto
- Avoid using the server for multiple/un-related applications
- Browse [www.spec.org](http://www.spec.org) for disclosures related to your platform
- Use hardware firewall vs. general-purpose firewall
- How much memory do you need?
  - Cache Accelerators need more memory
  - PHP, Perl applications can be memory intensive
System Tuning: Kernel Tuning

- Default Kernel tunable values are rarely optimal for Web Server performance
- Get more bang for your buck
  - Increase CPU usage
    - Increase Data Segment size
    - Increase # of kernel threads
    - Increase Shared Memory Size
    - Increase # of Locks that can be created
    - Increase # of Open Files
  - Decrease disk activity
    - Increase Virtual Page Size
    - Increase Dynamic Buffer Cache Size
    - Enable async File System writes
    - Increase SCSI max depth
    - Choose your file system correctly
      - Use Large File System block size
System Tuning: Network Tuning

- Use the latest/greatest performance patches (esp. ARPA, STREAMS & LAN)

100 Mbps cards Vs Gigabit cards?
- Gigabit cards are expensive, but fewer CPU interrupts
- Gigabit cards can also do checksum off-loading

Network Tunable Parameters
- Increase Max Outstanding Connection Requests
- TCP Transmit flow control
- TCP Stack caching for socket structures
- Use TCP Segmentation Offload (Large Send)
  - Sends large bursts of TCP data to network card

LAN Cards
- Send/Receive buffers
- Send/Receive coalesce ticks
System Tuning: Recommendations

- Sample values for HP-UX 11i v2 (2-way)
  - Data Segment size: maxdsiz, 0x40000000
  - Number of kernel threads: nkthread, 4096
  - Virtual Page Size: vps_pagesize, 64
  - Dynamic Buffer Cache Size: dbc_min_pct, 75 (% of RAM)
  - Shared Memory Size: shmmmax, 0x40000000
  - Max # of Locks: semmni, 4096
  - Number of Open Files: maxfiles, 60000
  - Async File System writes: fs_async, 1
  - SCSI max depth: scsi_max_qdepth, 64
  - Swap space/chunk size: swchunk, 4096
  - TCP Stack Caching: tcp_conn_strategy, 4096
  - TCP Transmit flow control: tcp_xmit_hiwater_def, 100000
  - TCP Outstanding connections: tcp_conn_request_max, 4096
Introduction to Apache 2.0
Demystifying Apache Performance
Performance Measurement & Analysis
Documentation / Links
Q & A
Performance Measurement & Analysis

- Performance Measurement
  - What to measure?
  - Tools Used
    - SPECweb99/SPECweb99_SSL
    - Web-Bench
    - Apache-Bench
    - flood
    - httperf

- Performance Analysis
  - What to look for?
  - Tools Used (and where to get it)
    - GlancePlus
    - gprof
    - truss/tusc
Performance Measurement: What to Measure?

- Tuning is not like Voodoo magic
  - Performance has to be measured to determine if it meets the goals
- Throughput
  - CPU/Memory cost per transaction
- Response Time
  - Most users get frustrated after 6 – 8 seconds
- The sample data used
  - Test using static-only content can be a waste
Performance Measurement: Tools Used

- **SPECweb99/SPECweb99_SSL**
  - Most widely recognized benchmark
  - Published by the SPEC [http://www.spec.org](http://www.spec.org)
  - Very sensitive to minor changes in Web Server, TCP stack and File System
  - Requires a dedicated setup

- **Web Bench**
  - Smaller version of SPECweb99
  - Meets most requirements, and easy to setup
  - Lacks SSL support
Performance Measurement Tools Used (contd.)

- **Apache Bench (ab)**
  - Lightweight
  - Download: Shipped along with Apache
  - Difficult to stress Web Server unless multiple instances are used simultaneously
  - Built-in support for SSL

- **Flood**
  - profile-driven HTTP load tester
  - Built-in support for SSL

- **httperf**
  - Robust and can give a wide-range of tests, extensible
  - Download: [http://freshmeat.net/projects/httperf](http://freshmeat.net/projects/httperf)
  - Requires faster/powerful clients
  - Lacks SSL support
Performance Measurement & Analysis

• Performance Measurement
  – What to measure?
  – Tools Used
    • SPECweb99/SPECweb99_SSL
    • Web-Bench
    • Apache-Bench
    • httperf

• Performance Analysis
  – What to look for?
  – Tools Used (and where to get it)
    • HP-UX Workload Manager
    • GlancePlus
    • gprof
    • truss/tusc
Performance Analysis: What to look for?

- CPU Consumption
- Memory Usage
- Network Bandwidth
  - Connection drops
  - TCP packet failures
- System Calls
  - Caching should decrease the number of file-system read’s
  - ‘EnableSendFile’ should result in fewer write’s
Performance Analysis: Tools Used

- **HP-UX Workload Manager**
- **Top**
  - Lightweight, very basic information is available
- **GlancePlus**
  - Used for monitor and optimize system performance
  - Availability: HP-UX Process Resource Manager
- **gprof**
  - Used to identify what functions the kernel spends the most time
  - Availability: on most HP-UX systems
  - Compile the source with “-G” option (available with cc, aCC)
- **tusc/truss**
  - Used to trace the system calls of a process
Introduction to Apache 2.0
Demystifying Apache Performance
Performance Measurement & Analysis
Documentation / Links
Q & A
Useful documentation / Links

- **DOWNLOAD:** http://httpd.apache.org
- **MAILING LIST:** dev@httpd.apache.org
  users@httpd.apache.org
- **DOCUMENTATION:** http://httpd.apache.org/docs-2.0
- http://www.spec.org
- http://www.securityspace.com
- http://www.netcraft.com
Introduction to Apache 2.0
Demystifying Apache Performance
Performance Measurement & Analysis
Documentation / Links
Q & A
Apache 2 Future directions

- Apache 2.2 release being planned for November 2004
  - Authentication / Authorization modules rewrite
  - Enhanced SSL support
    - SSL Connection Upgrade / Re-negotiation
    - Distributed SSL Session Caching (www.distcache.org)
  - Mod_proxy now supports Java Connectors (mod_jk)
    - Enhanced Caching support
    - Better performance with Load Balancing