Apache Overview HOWTO

Table of Contents

Table of Contents

Table of Contents

Apache Overview HOWTO

Daniel Lopez Ridruejo, `ridruejo@apache.org`

v0.7, 2002−02−28

This document gives you an overview of the Apache webserver and related projects. It provides pointers for further information and implementation details.

---

1. Introduction

This document gives you an overview of the Apache web server and related projects. Apache is the most popular server on the Internet. New Apache users, especially those coming from a Windows background, are often unaware of the possibilities of Apache, its useful addons and, more in general, how everything works together. This document aims to show a general picture of such possibilities with a brief description of each one and pointers for further information. The information has been gathered from many sources, including projects' web pages, conference talks, mailing lists, Apache websites and my own hands−on experience. Full credit is given to these authors. Without them and their work, this document would not have been possible or necessary.

Disclaimer: I work for Covalent. We provide products and support services for the Apache webserver, and I mention some of them here, as I do for our competitors and similar open source projects.

If you find typos, errors, or if you have suggestions for improvement or comments, please let me know so I can correct the document.

2. Apache

Apache is the leading internet web server, with over 60% market share, according to the Netcraft survey. Several key factors have contributed to Apache's success:

- The Apache license. It is an open source, BSD−like license that allows for both commercial and non−commercial uses of Apache.
- Talented community of developers with a variety of backgrounds and an open development process based on technical merits.
- Modular architecture. Apache users can easily add functionality or tailor Apache to their specific enviroment.
- Portable: Apache runs on nearly all flavors of Unix (and Linux), Windows, BeOs, mainframes...
- Robustness and security

Many commercial vendors have adopted Apache−based solutions for their products, including Oracle, Red Hat and IBM. In addition, Covalent provides add−on modules and 24x7 support for Apache.

The following websites use Apache or derivatives. Chances are that if Apache is good enough for them, it is also good enough for you :)

- Amazon.com
- Yahoo!
- W3 Consortium
- Financial Times
- Network solutions
- MP3.com
- Stanford

>From the Apache website:

The Apache Project is a collaborative software development effort aimed at creating a robust, commercial−grade, featureful, and freely−available source code implementation of an HTTP (Web) server.

The Apache project has grown beyond building just a web server into other critical server side technologies like Java or XML. The Apache Software Foundation, described in the next section, serves as an umbrella for these projects.

Related talk

- W09: Introduction to the Apache Web Server
- F16: Licensing issues in commercial OSS products

3. Apache Software Foundation

The Apache Software Foundation exists to provide organizational, legal, and financial support for the Apache open−source software projects. Formerly known as the Apache Group, the Foundation has been incorporated as a membership−based, not−for−profit corporation in order to ensure that the Apache projects continue to exist beyond the participation of individual volunteers, to enable contributions of intellectual property and funds on a sound basis, and to provide a vehicle for limiting legal exposure while participating in open−source software projects.

Or, as Roy T. Fielding, the chairman of the ASF describes it: *The mission of the Apache Software Foundation is to facilitate and support collaborative software development projects that use the Apache methods of collaboration over the Internet to create, maintain, and extend the infrastructure of the Web and enforce the standards that define it.*

You can learn more about the foundation here.

4. Developing web applications with Apache

There are several ways of providing content with Apache.

Related talk

- W07: Web Application Technologies − surveying the landscape

4.1 Static Content

Apache can serve static content, like HTML files, images, etc. If this is all you need, Apache is probably right for you. A low end Pentium running Linux and Apache can easily saturate a 10Mbps line serving static content. If that is your primary use of Apache, make sure you also check the performance section.

4.2 Dynamic content

For many websites, the information changes constantly and pages need to be updated periodically or generated on the fly. This is what server side programming is all about: programming languages, tools and frameworks that help developers query and modify information from different sources (databases, directory services, customer records, other websites) and deliver the content to the user.

4.3 CGI scripts

CGI stands for Common Gateway Interface. CGI scripts are external programs that are called when a user requests a certain page. The CGI receives information from the web server (form variable values, type of browser, IP address of the client, etc) and uses that information to output a web page to the client.

Pros: Since it is an external program, it can be coded in any language. The same script will also be portable among different web servers. The CGI protocol is simple, and the return result consists of writing the response to the standard output. It is a mature technology, and there are plenty of online and book references and examples.

Cons: Spawning and initializing a process takes time. Since a CGI script is external to the server and an instance has to be launched/destroyed for every request there is a performance hit. If the process has to load external libraries or perform a connection to an external database the delay can be important. Same thing if the number of hits per second is high. CGIs are stateless and session management has to be achieved by external means.

Since CGI usually involves heavy text manipulation, scripting languages are the natural choice. Part of Perl's popularity stems from its being the CGI programming language of choice. This is due to its extensive support for string handling and text processing. There are plenty of freely available CGI scripts and libraries. A good starting point is: the Open Directory CGI section

4.4 Site generators

If your site is high volume, you may run into performance problems when generating content dynamically. Offline content generators are an alternative. These solutions separate content from presentation. The HTML generator reads both content and presentation data and outputs the static files that build the website. The generator can be run periodically or triggered by content changes.

Future versions of Cocoon plan on having a batch mode to accomplish this. Another option is the Web site meta language.

4.5 Out of process servers

The web server can pass dynamic requests to another program. This program sits idle until a request arrives. The request is processed and returned to the webserver which in turn returns it to the client. This eliminates the overhead associated with CGI scripts. Examples of this approach are Fast CGI, Java servlets, etc.

4.6 Fast CGI

This standard was developed to address some shortcomings of the CGI protocol. The main improvement is that a single spawned process can process more than one request. There is an Apache module that implements the Fast CGI protocol and libraries for Tcl, Perl etc. More information at http://www.fastcgi.com

Related talk

- F18: FastCGI −− The forgotten treasure

4.7 Java servlets

To provide Java servlets, a Java virtual machine (in a process separate from the web server) processes requests. The JVM can reside in the same computer or in a different one. This is how many application servers work. Usually standard libraries are included for server side processing. JServ and Tomcat are Apache projects that provide this functionality. Related Java application server projects can be found here

Related talk

- W16: Recommendations for Java−Based Web Application Architectures

4.8 Embeded interpreters

An alternative to out−of−process webservers is to embed the interpreter in the server itself. There are roughly two categories in this kind of modules: Modules that answer or modify requests directly and modules that process commands embedded in HTML pages before serving it to the client. The most representative approaches are mod_perl and PHP

---

5. Performance and bandwidth management

Raw performance is only one of the factors to consider in a web server (flexibility and stability come usually first).

Having said that, there are solutions to improve performance on heavy loaded webservers serving static content. If you are in the hosting business Apache also provides ways in which you can measure and control bandwidth usage. Throttling in this context usually means slowing down the delivery of content based on the file requested, a specific client IP address, etc. This is done to prevent abuse.

- **mod_mmap**: Included in current Apache releases, it maps to memory a statically configured list of files that are frequently requested but infrequently changed.
- **Mod_bandwidth**: *Enables the setting of server−wide or per connection bandwidth limits, based on the specific directory, size of files and remote IP/domain.*
- **Bandwidth share module**: provides bandwidth throttling and balancing by client IP address. It is actively maintained.
- **Mod_throttle**:Throttle bandwidth per virtual host or user.
- **Mod_throttle_access**: useful if you are slashdotted. Allows throttling based on resources (file, directory, etc.)

---

6. Virtual hosting

Apache provides extensive virtual hosting support; there are additional modules that provide specific functionality:

- mod_dynvhost
- mod_pweb
- mod_v2h

In addition, Apache 2.0 allows children serving different domains to have different user ids, improving security.

---

7. Load balancing

Apache has several modules that allow distribution of requests among servers, for redundancy, increased availability, etc.

- **Reverse proxying + mod_rewrite**: There is nothing in Apache that you can not do with mod_rewrite ... :) This technique consists of having an Apache front−end server acting as a proxy for the backend servers. You can find more information here
- **Mod_backhand**: *Allows seamless redirection of HTTP requests from one web server to another. This redirection can be used to target machines with under−utilized resources, thus providing fine−grained, per−request load balancing of web requests*. More information at http://www.backhand.org/.

Related talk

- TH06: mod_backhand: Internals explained

8. Secure transactions

There are several solutions that provide secure transactions for Apache servers. This enables Apache servers to be used for e−commerce or other scenarios where sensitive information (like your credit card number) is exchanged.

- Mod_ssl and Apache−SSL are open source implementations. They are European based, unencumbered by RSA patents.
- Red Hat offers a secure server derived from Apache. Red Hat acquired C2Net, makers of StrongHold, another Secure server derived from Apache.
- Covalent sells secure versions of Apache as well as the CovalentSSL module that plugs into existing Apache installations.

Credit card transactions

Apache specific solutions exist for credit card transactions:

- Covalent credator, multiple clearinghouse support, failover operation, PHP, Perl, Java support.

9. SNMP

SNMP stands for Simple Network Management Protocol. It allows monitoring and management of network servers, equipment, etc. SNMP modules for Apache help manage large deployments of web servers, measure the quality of service offered and integration of Apache in existing management frameworks.

- Open source Mod SNMP for Apache 1.3.
- Covalent SNMP provides a commercial SNMP module, support for the latest SNMPv3 standard, integration with HP−Openview, Tivoli, etc.

10. Authentication modules

In many situations (subscription services, sensitive information, private areas), user authentication is required. Apache includes basic authentication support. Additional authentication modules exist that connect Apache to existing security frameworks or databases, including: NT Domain controller, Oracle, mySQL, PostgresSQL, etc.

The LDAP modules are specially interesting, as they allow integration with company and enterprise wide existing directory services.

You can find these modules at http://modules.apache.org.

11. GUIs for Apache

Apache is configured thru text configuration files. This has advantages and disadvantages. Management can be done from any computer that has internet access via ssh. Editing a configuration file by hand implies a learning curve. There are open source graphical tools that make this task easier:

- Comanche: It is crossplatform and runs on Unix/Linux, Windows and Mac. Check the website for screenshots and in−depth information. Disclaimer: I am the main author of Comanche, so remember, there are no bugs, only undocumented features :)
- gui.apache.org: GUI interfaces for Apache project. Programs are in various degrees of development.
- Webmin: A nice web based interface.

12. Writing Apache modules

Apache, like many other successful open source projects, has a modular architecture. This means that to add or modify functionality you do not need to know the whole code base. Source code access for Apache means that you can custom build the server with only the modules that you need and include your own.

Extending Apache can be done in C or in a variety of other languages using appropriate modules. These modules expose Apache's internal functionality to different programming languages like Perl or Tcl.

Writing modules in C: Apache is written in C and so are the modules distributed with Apache. The best way to get started writing Apache modules is to read Doug MacEachern and Lincoln Stein's book Writing Apache modules with Perl and C. It is a well−written, easy to read book by two Apache and Perl gurus. The above link will lead you to the book website, which has some of its chapters online. If you don'e have the money to buy the book and cannot borrow it from a friend, there are other ways. You can read some of the online tutorials on writing Apache modules: Ken Coar, an Apache Group member, has a nice tutorial and slides online. An overview of the Apache architecture can be found here. The Apache website has some API notes that can help you get started. You are also encouraged to browse the source code of the modules included with Apache. Apache includes a simple one (mod_example.c) for that purpose.

Writing Apache modules in other languages: There is a variety of Apache modules that enable third party languages to access the internal Apache API. The most popular is mod_perl.

If you have any questions about the development of an Apache module you should join the Apache modules mailing list at http://modules.apache.org. Remember to do your homework first, research past messages and check all the documentation previously described. Chances are somebody had the same problem that you are experiencing and he got an useful response.

If you are interested in the development of core Apache itself, you should checkout the Apache development site.

---

13. Apache books

A comprehensive list of Apache books can be found here.

A couple of books that I personally recommend are:

- Writing Apache Modules with Perl and C if you are interested in Apache internals.
- Apache server for dummies if you want to get started with Apache. Do not get fooled by the name. This is a comprehensive book packed with useful information.

---

14. WebDAV

>From the WebDAV website: *WebDAV stands for "Web−based Distributed Authoring and Versioning". It is a set of extensions to the HTTP protocol which allows users to collaboratively edit and manage files on remote web servers.*

It is the open standards equivalent of the MS FrontPage protocol, but it takes the idea several steps further. It enables other protocols to be built on top of it (See the Subversion website for an example.

---

15. Java projects

For historical reasons, Java projects can be found both under the java.apache.org and jakarta.apache.org umbrellas. The final goal is that over time all Java projects will move under the Jakarta umbrella.

The goal of the Jakarta Project is to provide commercial−quality server solutions based on the Java Platform that are developed in an open and cooperative fashion.

The Java on Apache community is a very dynamic and active one, as shown by the quantity and quality of its subprojects, which are described now.

15.1 Ant

You can think of Ant as the Java equivalent of make. It is a big success with Java related projects. Developers can write Java instead of shell commands. This means increased portability and extensibility. Instead of Makefiles Ant has XML files. You can learn more about ANT here.

Related talk

- F19: Using Ant to build Java code

15.2 ORO and Regexp

ORO is a complete package that provides regular experession support for Java. It includes Perl5 regular expression support, glob expressions, etc. All under the Apache license. You can learn more about ORO here. You can find another lightweight regular expression package, Regexp.

15.3 Slide

Slide is a high−level content management framework. Conceptually, it provides a hierarchical organization of binary content which can be stored into arbitrary, heterogenous, distributed data stores. In addition, Slide integrates security, locking and versioning services.

If you are familiar with WedDAV, Slide uses it extensively. In simple words, what Slides provides is an unified, simple way to access resources and information. These resources can be stored in a database, the filesystem, etc. and accessed either thru a WebDAV interface or Slide's own API.

You can learn more at the Slide home page.

15.4 Struts

Struts is an Apache project that tries to bring the Model−View−Controller (MVC) design paradigm to web development. It builds on Servlet and JavaServer Pages technologies. The model part is made up of Java server objects, which represent the internal state of the application. Enterprise Java Beans are commonly used here. The view part is constructed via JavaServer Pages (JSP), which is a combination of static HTML/XML and Java. JSPs also allow the developer to define new tags. The controller part consists of servlets, which take requests (GET/POST) from the client, perform actions on the model and update the view by providing the appropriate JSP. You can learn more at the Struts project pages.

15.5 Taglibs

The JavaServer pages technology allows developers to provide functionality by adding custom tags. The Taglibs project intends to be a common repository for these extensions. It includes tags for common utilities (i.e. date), SQL database access, etc.

You can learn about TagLibs here. More documentation is included in the package.

15.6 Tomcat

Tomcat is the flagship product of the Jakarta project. It is the official reference implementation for the Java Servlet 2.2 and JavaServer Pages 1.1 technologies.

You can learn more in the Tomcat homepage. The Tomcat project was started with a code donation from Sun Microsystems.

15.7 Velocity

Velocity is a Java based template engine. It can be used as a stand−alone utility for generating source code, HTML, reports, or it can be combined with other systems to provide template services. Velocity has a Model View Controller paradigm that enforces separation of Java code and the HTML template.

You can learn more about Velocity here. Velocity is part of other projects like Turbine

15.8 Watchdog

The Watchdog project provides the validation tests for the Servlet and JavaServer Pages specifications. You can find more information here

15.9 JServ

Apache JServ is in a maintenance only mode at this point. This means that there will be no new official releases and that only well tested patches are being committed. No new features are being added. If you are looking for the latest implementation of a Java Servlet Engine and/or Java Server Pages (JSP) then you should consider using the Jakarta Tomcat product available from the Jakarta Project.

15.10 JSSI

JSSI is an implementation of server side included in the Java language. Server side includes are tags includes in files that get processed before the page is served to the client (for example to include the current date) You can find more information here.

15.11 Apache JMeter

The Apache JMeter is a 100% pure Java desktop application designed to load test functional behavior and measure performance. It was originally designed for testing Web Applications but has since expanded to other test functions.

It can be used to test static and dynamic resources and get inmediate visual feedback.

You can see some screenshots and learn more here.

15.12 Server Pages Foundation Classes

SPFC is a set of libraries to help solve common problems in server side application development. They focus on two of them:

- **Mixing HTML and Java**: Provides a library of classes that takes care of the HTML generation and that can be integrated with the rest of the Java code.
- **HTTP is a stateless protocol**: SPFC provides session support, so applications can keep track of users as they navigate the website. The application developer does not need to worry about the specific details of page generation. He can think in more general traditional application terms. You can learn more about SPFC here

15.13 Element Construction Set

Element Construction Set (ECS) is a JAVA API for generating elements for various markup languages. It directly supports HTML 4.0 and XML, but can easily be extended to create tags for any markup language.

It allows the generation of mark up tags using Java function calls, leading to a much cleaner solution than mixing HTML and Java code. You can learn more at the ECS project page.

15.14 Avalon

If you are familiar with Perl or BSD systems, Avalon is roughly the equivalent of CPAN or the Ports collection for Java Apache technologies. It does not only provide guidelines for a common repository of code, it goes one step further: *is an effort to create, design, develop and maintain a common framework for server applications written using the Java language.* It provides the means so server side Java projects can be easily integrated and build on each other.

15.15 JAMES (Java Apache Mail Enterprise Server)

Complementary to the other Apache server side technologies, JAMES provides *a 100% pure Java server designed to be a complete and portable enterprise mail engine solution based on currently available open protocols (SMTP, POP3, IMAP, HTTP)*

More information can be found here.

15.16 PicoServer

A lightweight HTTP/1.0 server in pure Java. The project seems to be stalled and no code is available. The website and CVS are no longer available.

15.17 Jetspeed

Jetspeed is a web based portal written in Java. It has a modular API that allows aggregation of different data sources (XML, SMTP, iCalendar)

Related talk:

- TH11: Writing an enterprise information portal with JetSpeed

15.18 Turbine

Turbine is a servlet based framework that allows experienced Java developers to quickly build secure web applications. Turbine brings together a platform for running Java code *and* reusable components, everything under the Apache license. Some of its features include:

- Integration with template systems
- MVC style development
- Access Control Lists

- Localization support
- etc.

If you are interested, you can visit the Turbine web site.

15.19 Jyve

The Jyve project is built on top of the Turbine framework. It is an application that provides a web based FAQ system

15.20 Alexandria

Alexandria is an integrated documentation management system. It brings together technologies common to many open source projects like CVS and JavaDoc. The goal is to integrate source code and documentation to encourage code documentation and sharing. More information here

Related talk

- W06: An introduction to Alexandria

15.21 Log4j

This package provides a logging framework that Java applications can use. It can be enabled at runtime without modifying the binary and has been designed with performance in mind. It can be found here

---

16. XML projects

Directly from the Apache XML project website, its goals are:

- *To provide commercial−quality standards−based XML solutions that are developed in an open and cooperative fashion.*
- *To provide feedback to standards bodies (such as IETF and W3C) from an implementation perspective.*
- *To be a focus for XML−related activities within Apache projects*

The project homepage is located at http://xml.apache.org. It is an umbrella for a variety of subprojects.

16.1 Introduction to XML

This is a quick introduction to XML. To know more about XML, a good starting point is http://www.xml.com. XML is a markup language (think HTML) for describing structured content using tags and attributes. Once content is separated from presentation, you can choose how to display (cellphone, html, text) or exchange it. The XML standard only describes how the tags and attributes can be arranged, not its names of what they mean. Apache provides the tools described in the following sections.

16.2 Xerces

The Xerces project provides XML parsers for a variety of languages, including Java, C++ and Perl. The Perl bindings are based on the C++ sources. There are Tcl bindings for Xerces in the 2.0 version of TclXML, by Steve Ball. This 2.0 version is available thru the SourceForge project page. An XML parser is a tool used for programatic access to XML documents. This is a description of the standards supported by Xerces:

- DOM: DOM stands for Document Object Model. XML documents are hierarchical by nature (nested tags). XML documents can be accessed thru a tree like interface. The process is as follow:
    - Parse document
    - Build tree
    - add/delete/modify nodes
    - Serialize tree
- SAX:Simple API for XML. This is a stream based API. This means that we will receive callbacks as elements are encountered. These callbacks can be used to construct a DOM tree for example.
- XML Namespaces
- XML Schema: The XML standard provides the syntax for writing documents. XML Schema provides the tools for defining the *contents* of the XML document (semantics). It allows to define that a certain element in the document must be an integer between 10 and 20, etc.

The Xerces XML project initial code base was donated by IBM. You can find more information in the Xerces Java, Xerces C++ and Xerces Perl homepages.

16.3 Xalan

Xalan is an XSLT processor available for Java and C++. XSL is a style sheet language for XML. The T is for Transformation. XML is good at storing structured data (information). We sometimes need to display this data to the user or apply some other transformation. Xalan takes the original XML document, reads transformation configuration (stylesheet) and outputs HTML, plain text or another XML document. You can learn more about Xalan at the Xalan Java and Xalan C++ project homepages.

16.4 FOP

From the website: *FOP is a Java application that reads a formatting object tree and then turns it into a PDF document*. So FOP takes an XML document and outputs PDF, in a similar way that Xalan does with HTML or text. You can learn more about FOP here.

16.5 Cocoon

Cocoon leverages other Apache XML technologies like Xerces, Xalan and FOP to provide a comprehensive publishing framework. Cocoon is based around XML and XSL and targeted to sites of medium − high complexity. It separates content, logic and presentation as described in the website:

- **XML creation**: *the XML file is created by the content owners. They do not require specific knowledge on how the XML content is further processed rather than the particular chosen DTD/namespace. This layer is always performed by humans directly through normal text editors or XML−aware tools/editors.*
- **XML process generators**: *the logic is separated from the content file.*
- **XSL rendering**: *The created document is then rendered by applying an XSL stylesheet to it and formatting it to the specified resource type (HTML, PDF, XML, WML, XHTML)*

You can learn more about Cocoon at the project homepage

16.6 Xang

The goal of the Xang project is to *make it easy for developers to build commercial quality XML aware applications for the Web*. The application logic is defined in a hierarchical XML file which can be scripted via JavaScript. This file defines how to access the data (which can be other XML files, Java plug−ins, etc.). The Xang engine takes care of mapping HTTP requests to the appropriate handlers. You can learn more about Xang at the project homepage.

16.7 SOAP

Apache SOAP ("Simple Object Access Protocol") is an implementation of the SOAP submission to W3C. It is based on, and supersedes, the IBM SOAP4J implementation.

From the draft W3C specification: SOAP is a lightweight protocol for exchange of information in a decentralized, distributed environment. It is an XML based protocol that consists of three parts:

- *An envelope that defines a framework for describing what is in a message and how to process it,*
- *a set of encoding rules for expressing instances of application−defined datatypes*, and
- *a convention for representing remote procedure calls and responses*.

Think of SOAP as an XML based remote procedure call or CORBA system. It is based on HTTP and XML. On the one hand this means it is verbose and slow compared to other systems. On the other hand it eases interoperatibility, debugging and development of clients and servers for a variety of languages (C, Java, , Perl, Python, Tcl, etc.) since most modern languages have HTTP and XML modules. You can learn more at the Apache SOAP homepage

Related talk

- W02: Rub−a−dub−dub−dubya: SOAP and the Web

16.8 Batik

Batik is a Java based toolkit for applications that want to use images in the Scalable Vector Graphics (SVG) format for various purposes, such as viewing, generation or manipulation.

It is XML centric and compliant with the W3C specification. It is a bit atypical from other Apache projects, in that it provides a graphical component. Batik provides hooks to extend the framework thru custom tags and it allows conversion from SVG to other formats like JPEG or PNG.

Batik homepage

Related talk

- W14: Introduction to the Batik project.

16.9 Crimson

Crimson is an alternative, Java−based, XML parser with support for XML 1.0 thru a variety of interfaces. It is the parser currently shipping in Sun products, and an intermediate step until the version 2 of Xerces is released.

Crimson homepage

Related talk

- TH08: Java API for XML processing (JAXP) version 1.1

16.10 Other XML projects

There are other projects based on Apache and XML that do not live under the Apache XML umbrella

- mod_xslt is a C based module for delivering XML/XSL based content. It has a GPL license.
- AxKit is an XML based Application Server for mod_perl and Apache. It allows separation of content and presentation.

Related talk

- TH04: AxKit − An XML Application server for Apache

---

17. Perl

Perl and Apache make a powerful and popular combination. There are several projects that use these two technologies.

17.1 Embperl

Embperl allows embedding of Perl in HTML pages. These pages are processed in the server before they are delivered to the client. It is similar to PHP. You can learn more here.

17.2 Mason

The Mason project embeds Perl in HTML with a reusable component model approach. It allows caching, templating, etc.

17.3 Mod_Perl

mod_perl is one of the most veteran and successful Apache projects. It embeds a Perl interpreter in Apache and allows access to the web server internals from Perl. This allows for entire modules to be written in Perl or a mixture of Perl and C code. In the 1.3 Apache versions, one interpreter has to be embedded in each child, since the server is multiprocess based. In heavy traffic dynamic sites, the increased size could make a difference. Apache 2.0 is multithreaded, as recent versions of Perl are. The next generation of mod_perl takes advantage of this and allows for sharing of code, data and session state among interpreters. This results in a faster, leaner solution.

Make sure you also check axkit

18. PHP

From the PHP website: *PHP is a server−side, cross−platform, HTML embedded scripting language.* PHP is a scripting language like Perl, Python or Tcl. It is the most popular module for Apache and this is due to a variety of reasons:

- Learning curve is quite low
- Great documentation
- Extensive database support
- Modularity

PHP has a modular design. There are modules that provide support for:

- Database connetivity for Oracle, ODBC, mySQL, mSQL, PostgreSQL, MS−SQL server... and many more, check the PHP website.
- XML support
- File transfer: FTP
- HTTP
- Directory support: LDAP
- Mail support: IMAP, POP3, NNTP
- PDF document generation
- CORBA

and many more. You only need to compile/use the modules you need.

PHP can be used with Apache, as an external CGI or with other webservers. It is crossplatform and it runs on most flavors of Unix and Windows.

If you come from a Windows background, you probably have used Internet Information Server with Active Server Pages and MS−SQL Server. A common replacement in the Unix world for this trio is Apache with PHP and mySQL. Since PHP works:

- with Apache and with Microsoft IIS
- with mySQL and with MS−SQL server
- on Unix and on Windows

you have a nice migration path from a Microsoft−centric solution to more secure, stable, high performance Unix based solutions (like FreeBSD, Solaris, Linux or OpenBSD)

19. Python

Python is an scripting language similar to Perl or Tcl. Several modules embed Python in the Apache web server:

- Mod Python
- Mod Snake: runs both in Apache 1.3.x and the upcoming 2.0

Both modules would be useful if you plan on writing Apache modules in Python or run existing Python CGIs faster. Mod Snake allows you to embed Python in HTML , much like PHP does.

Related talk:

- F08: mod_snake: Boosting productivity with Python

---

20. Tcl

The Tcl Apache project integrates Tcl with the Apache webserver. Tcl is a lightweight, extensible scripting language. You can learn more about Tcl here. There are several modules currently under the Apache Tcl umbrella:

- Mod_dtcl allows for embedding Tcl on HTML pages like PHP does.
- Neowebscript takes a similar approach
- Mod_tcl takes an approach similar to mod_perl and runs both in 1.3.x and 2.x versions of Apache.

Other Tcl Apache projects can be found at WebSH.

---

21. Modules for other languages

This document has described modules for popular server side languages such as Perl, Python, PHP. You can find additional language modules (JavaScript, Haskell, etc.) at the Apache modules directory.

---

22. Apache 2.0

The current version of Apache (the 1.3 series) is process based. That means that the server forks itself a number of times to answer simultaneous requests. The children are isolated from each other. This is reliable: if a module misbehaves, the parent process kills that child and it only affects the request being served, not the server as a whole. Threads are similar to lightweight processes. Threads can share common data. If a thread misbehaves it can corrupt other threads and the server as a whole can go down. On the other hand, the thread model allows for faster, leaner webservers. Apache 2.0 brings the best of both worlds, allowing the user to define number of processes and number of threads per process. Apache 2.0 introduces APR, the Apache Portable Runtime, which increases Apache's portability even more. Finally, layered I/O brings a new level of modularity to Apache development.

---

23. Migrating from Netscape (iPlanet) web servers

The bulk of the work may reside in converting custom modules from NSAPI to the Apache API. Nearly all the other server side technologies (Java, Perl, CGIs) should be portable with little or no change. Netscape is tightly integrated with LDAP servers. You may be also interested in LDAP modules in http://modules.apache.org.

---

24. Migrating from Microsoft IIS

Common reasons why people migrate from IIS to Apache (and not the other way around) include stability, performance and security. This is partly because most people running Apache do it on an Unix variant (like Solaris, FreeBSD or Linux). Fortunately, Apache is multiplatform and runs on both Unix and Windows, offering a sensible migration path.

Common Windows based web development environments like Coldfusion or Active Server Pages have Unix ports or compatible environments (some are commmercial, some are freely available):

- Coldfusion for Linux
- Perl ASP module
- Halcyon ASP

Apache for Windows supports also the ISAPI interface.

If you want to go for a complete open source solution and you come from a Windows background (IIS + ASP + MS−SQL server) the roughly equivalent (and highly popular) combination is Apache + PHP + MySQL or PostgresSQL. You can learn more about PHP here

Support for Windows is greatly improved in the new 2.0 Apache version, still in beta stage at the time of this writing.

---

25. Links

Additional Apache related resources

25.1 Websites

- Apache
- Apache modules directory
- Apache today
- Slashdot Apache section

25.2 Java application servers

These are open source application servers that build on or are known to play well with Apache.

- Resin: Servlets, JSP, XSL
- Enhydra: Java/XML application server.
- JBoss: Enterprise Java Beans container, J2EE

---

26. Contacting the author

You can contact me at ridruejo@apache.org. I welcome suggestions and corrections, but please, please, do not send me messages asking me to troubleshoot your Apache installation. I just do not have the bandwidth and your mail will be most likely ignored. If you need support:

- Check the error logs, read the docs, especially the FAQ.
- Try comp.infosystems.www.servers.unix at http://groups.google.com. Search for a similar problem.
- If you are still stuck, provide as much information as you can, including relevant error_log entries and steps you have taken so far, and post to that newsgroup. This will increase the chances someone will answer your question.

If you want commercial support, consider contacting Covalent, which provides expert support for Apache (at a fee, of course). If you are using Apache on Linux, your Linux vendor may have support plans that include Apache too.

26.1 Translations

If you want to contribute a translation of this document you should use the SGML source. Check http://www.linuxdoc.org for info. Please drop me a note so I can make sure you get the most recent version.

---

27. Open Content Open Publication License

27.3 SCOPE OF LICENSE

The following license terms apply to all Open Publication works, unless otherwise explicitly stated in the document.

Mere aggregation of Open Publication works or a portion of an Open Publication work with other works or programs on the same media shall not cause this license to apply to those other works. The aggregate work shall contain a notice specifying the inclusion of the Open Publication material and appropriate copyright notice.

SEVERABILITY. If any part of this license is found to be unenforceable in any jurisdiction, the remaining portions of the license remain in force.

NO WARRANTY. Open Publication works are licensed and provided "as is" without warranty of any kind, express or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose or a warranty of non−infringement.

27.4 REQUIREMENTS ON MODIFIED WORKS

All modified versions of documents covered by this license, including translations, anthologies, compilations and partial documents, must meet the following requirements:

- 1. The modified version must be labeled as such.
- 2. The person making the modifications must be identified and the modifications dated.
- 3. Acknowledgement of the original author and publisher if applicable must be retained according to normal academic citation practices.
- 4. The location of the original unmodified document must be identified.
- 5. The original author's (or authors') name(s) may not be used to assert or imply endorsement of the resulting document without the original author's (or authors') permission.

27.5 GOOD−PRACTICE RECOMMENDATIONS

In addition to the requirements of this license, it is requested from and strongly recommended of redistributors that:

- 1. If you are distributing Open Publication works on hardcopy or CD−ROM, you provide email notification to the authors of your intent to redistribute at least thirty days before your manuscript or media freeze, to give the authors time to provide updated documents. This notification should describe modifications, if any, made to the document.
- 2. All substantive modifications (including deletions) be either clearly marked up in the document or else described in an attachment to the document.
- 3. Finally, while it is not mandatory under this license, it is considered good form to offer a free copy of any hardcopy and CD−ROM expression of an Open Publication−licensed work to its author(s).

27.6 LICENSE OPTIONS

The author(s) and/or publisher of an Open Publication−licensed document may elect certain options by appending language to the reference to or copy of the license. These options are considered part of the license instance and must be included with the license (or its incorporation by reference) in derived works.