Assembling Web Development Environments with Perforce

Stephen Vance
Perforce User Conference 2001
Once Upon A Time ...

- Experiences from
  - Consulting for Web companies
  - Involvement with Internet Start-up
  - Problem-solving from personal interest
- Internet start-up is main storyline.
- Of course, names changed.
- And now, on with our story.
The Characters

- Me: “I can’t let chaos ruin us”
- The CFO: “We’re on a tight venture budget”
- The CTO: “I can develop that in two hacks”
- The Marketing Guy: “I can’t live without my Mac”
- The Sub-contractor: “We do what we do”
The Case for SCM in Web Development

- Many disciplines converging in one medium
  - Programmers
  - HTML designers
  - Advertising executives
  - Marketers
- Many skill levels from novice to expert
- Large-scale sites are still driven by custom-coded solutions and large numbers of contributors
In The Beginning ...

- There was the void.
  - a.k.a. a start-up with funding and no developers
- And then he said, “Let them have options” and product development commenced.
- Against better judgment we started hacking out of a common directory tree
And as one might expect, some code was irretrievably flattened at 3:00 a.m. by two people working on the same file.

After some “discussions” with the CTO and the CFO, Perforce was purchased.
The Web Site Conundrum

- Each developer has his own workspace
- But there is still only one Web server
- Although source changes are safe against multiple authors, there is only one testing environment
- Until …
Virtual Hosts

- IP- or name-based
- IP-based virtual hosting has shortcomings
- Name-based virtual hosting addresses these shortcomings
Name-based Virtual Hosts

- Server distinguishes Web sites based on name in Host header
- DNS wildcards avoid need for DNS reconfiguration
- Apache configuration handles the rest

```
GET / HTTP/1.1
Host: www.vance.com
```

```
```

```
NameVirtualHost *
<VirtualHost *>
  DocumentRoot /usr/local/apache/htdocs
  ServerName www.vance.com
</VirtualHost>
<VirtualHost *>
  DocumentRoot /usr/local/dev/site1/htdocs
  ServerName site1.dev.vance.com
</VirtualHost>
```
Visualizing Virtual Hosts

devhost.vance.com. A 86400 192.168.1.1

DNS

/sit1.dev.vance.com?

Apache
(devhost.vance.com)

Displays page

http://sit1.dev.vance.com/test.html

Gets file

Returns page

GET /test.html HTTP/1.1
Host: sit1.dev.vance.com

/site1.dev.vance.com/htdocs/test.html

/sit1.dev.vance.com?

http://site1.dev.vance.com/test.html
Enter The Marketing Guy

- Technically savvy, but non-developer
- Rejected disk sharing technologies
- No Mac solution for non-developers
- Used Windows, but
  - Hated it
  - Required logging in to sync virtual host
- Until …
P4Web to the Rescue

- P4Web is a Perforce-specialized Web server
- Provides GUI on any platform with a browser
- Runs on almost any platform supported by Perforce
- Allows management and viewing of Web content in Perforce
P4Web Configurations
The Next Challenge

- The Marketing Guy is happy.
- Then he announces that he has hired a sub-contractor to do the Web design.
- Quite reasonably, the sub-contractor doesn’t feel he needs to learn Perforce to fulfill the contract.
Easy FTP Integration

- Web design tools support ftp
- Put sub-contractor work on its own branch
- Create a client with “allwrite” option
- Create virtual host with document root of <client root>/htdocs
- Tell sub-contractor to set ftp location to <client root>/htdocs
- Populate /cgi-bin if necessary
- Use Tech Note #2 techniques when sub-contractor notifies of updates
Improved Integration Using P4FTP

- P4FTP is Perforce FTP server where session is a submission
- Isolate work to a branch
- Excellent solution, but two drawbacks
  - Content only transiently populates disk, so not fit to work with virtual host for viewing
  - Each ftp user consumes a Perforce license and only uses a single client workspace
Where We Are

- Virtual hosting allows multiple Web servers on minimal hardware
- P4Web allows browser-based use of Perforce client functionality with viewing of static content
- FTP and P4FTP allow smooth integration with Web design tools
- However, we still have to sync a Unix client to fully view content
Viewing Content with WebKeeper

- **What?**
  - Apache module allowing direct viewing of content from Perforce depot
  - Open source project in the Perforce public depot
  - Recently enhanced

- **Who?**
  - Originally authored through Perforce
  - I was recently designated curator

- **Where?**
  - public.perforce.com:1666://public/perforce/webkeeper/…
How WebKeeper Works

Client Machine  Web Server  Perforce Server

Browser  Apache WebKeeper  P4D

Simplest

Browser  Apache WebKeeper  P4D

Most Capable

Browser  Apache WebKeeper  P4D

Client

Depot

Depot
WebKeeper Installation

- Instructions in README.WEBKEEP in Perforce public depot
- Builds either statically or as loadable module
- Builds either using full Apache source tree or using `apxs` with Red Hat apache-devel.
- Built and tested on FreeBSD and Linux.
- Other platforms on request
- Plan to add binary loadable modules for FreeBSD and Red Hat Linux to public depot.
Basic WebKeeper Configuration

```html
<VirtualHost *>
    ServerName site1.dev.vance.com
    <IfModule mod_webkeep.c>
        WebKeepPort perforce.vance.com:1666
        WebKeepUser steve
        WebKeepPasswd mypassword
        WebKeepDirectoryIndex index.html
        WebKeepAlias / //depot/dev/site1/htdocs/
    </IfModule>
</VirtualHost>
```

- WebKeep directives replace DocumentRoot
- WebKeepDirectoryIndex should correspond to Apache DirectoryIndex
- WebKeepAlias maps Web root to Perforce depot
- IfModule makes configuration portable
Why Do We Need More?

- In two words, “Dynamic Content”
  - Text-based
    - Server-side includes (SSI)
    - PHP
    - Perl
    - ASP
    - JSP
    - Server-processed XML and XSLT
  - Binary
    - C/C++ CGI
    - EJB
    - Servlets
<IfModule mod_webkeep.c>
  WebKeepPort perforce:1666
  WebKeepUser perforce_user
  WebKeepDirectoryIndex index.html
  WebKeepAlias / //my_client/htdocs/
  WebKeepClient my_client
  WebKeepSync On
  WebKeepRefresh //my_client/include/...
</IfModule>

- WebKeepClient gives a client spec
- WebKeepSync tells WebKeeper to sync the files to the client for Apache
- WebKeepRefresh (added after paper) always refreshes include directory
Adding CGI Support

<VirtualHost *>
  ServerAdmin webmaster@vance.com
  ServerName site1.dev.vance.com
  DocumentRoot /usr/local/dev/steve/htdocs
  ScriptAlias /cgi-bin /usr/local/dev/steve/cgi-bin
  <IfModule mod_webkeep.c>
    WebKeepPort perforce.vance.com:1666
    WebKeepUser steve
    WebKeepDirectoryIndex index.html
    WebKeepAlias /cgi-bin/ //depot/dev/steve/cgi-bin/
    WebKeepAlias / //depot/dev/steve/htdocs/
    WebKeepClient my_client
    WebKeepSync On
  </IfModule>
</VirtualHost>

- DocumentRoot reintroduced for file access
- ScriptAlias tells Apache which content to treat as CGI scripts
- New WebKeepAlias tells WebKeeper how to populate the CGI directory
- WebKeeper handles aliases in the order presented in the configuration, so the CGI alias must come first
Compiled Web Elements

- Web elements that
  - Require compilation before being sent to browser
  - Require compilation before executing on server

- Examples
  - Java applets
  - C/C++ CGIs
  - Java Servlets
  - Enterprise JavaBeans
Compiled Element Issues

- Platform dependence
  - C/C++ and Java with JNI are platform-dependent
  - Java without JNI and scripts are independent
  - Impacts location of development workspaces

- Deployment location
  - Combines with platform to constrain choices
Supporting Compiled Elements

- **C/C++ CGIs**
  - Compile on server or like platform
  - Target or copy to /cgi-bin on virtual host

- **Java applets**
  - Compile on any platform (JNI usually non-issue)
  - Target or copy to virtual host

- **EJBs and Servlets**
  - Compile on any platform
  - JRE compatibility
  - Target or copy to server’s `<app-path>/WEB-INF/classes`
    - `java –d <app-path>/WEB-INF/classes Some.class`
    - Manually update deployment descriptors if necessary
    - Setup up application server to dynamically reload classes
Web Site Deployment

- Perforce can be used to deploy Web sites
- WebKeeper not advised for production sites
  - Additional DNS lookups and network requests
  - Potential security hole through firewall
  - Large Perforce operations result in Web site unavailability
- Syncing is advised method
Deployment Techniques

- Check in binaries
- Maintain two document trees
  - Ensures atomicity of deployment
  - Reduces timing demands for deployment
  - Method
    - One is live
    - Deploy to the non-live
    - Switch
  - Analogous to graphics double-buffering
Web Site Double-Buffering

Web Server

Apache

Copy 1

Copy 2

httpd.conf

Perforce Server

p4d

Depot

p4 sync
Advanced Deployment

- Coordinating deployment to and switchover in server farm
- Deploying to multiple hosting centers
- Sub-dividing content space for partial deployment
- See P4UC00 presentation by David Markley and Scott Money of Lycos, Inc. for ideas on large-scale deployment
Branching for Web Content

- Different from branching for traditional software products
  - Shorter cycles
  - Smaller functional change sets
  - More evolutionary
  - More continuous

- Typically two- or three-tiered approach
  - Development mainline
  - QA codeline
  - Possible release codeline
WCM Branching

- Development Mainline
  - Central for ongoing enhancements and fixes
  - Project branches originate here

- QA Codeline
  - Prevents “code freeze” delays
  - Allows selective integration
  - Allows direct fixes of QA defects
  - Possible simple release codeline

- Release Codeline
  - Prevents QA stall waiting for release
  - Encapsulates final release configuration
  - Allows selective integration of independent components
  - Allows final release transformations
  - Deployment master
For More Info

- Additional info in paper on
  - Installing WebKeeper
  - Whether and when to archive binaries
  - Handling shared resources like databases
  - Managing Apache configurations
  - Product and literature references
Future Directions

- Expand application server integration
  - J2EE servers
  - Zope
- Expand large-scale deployment strategies
- Enhance WebKeeper
  - Started basic authentication before conference based on Chris Seiwald’s work
- Mass virtual hosting support
- Add binary modules to depot
- Support Apache 2.0
- Miscellaneous smaller items
Conclusion

- Perforce + Open-Source Software + Best Practices = Effective Web Development Management
- As always, consider the needs when implementing solutions
- Hopefully, this material gives you the information to consider wisely