The Apache HTTP Server Project

Lessons Learned from Collaborative Software Development

Roy T. Fielding
University of California, Irvine
http://www.ics.uci.edu/~fielding/

Overview

- History of the Apache Project
- Evolution of the development process
- Global collaboration techniques
- WWW architectural style
- Apache architecture
- Lessons for Software Engineers

The Apache Project

- A common goal
  - To provide an open source, secure, efficient and extensible server that provides HTTP services in sync with non-proprietary World Wide Web standards
- Apache Group
  - Self-selected volunteers that guide the project and perform most of the development work
  - US, UK, Canada, Germany, Italy (EC)
- Current status
  - #1 server (56% of the public Internet sites)
  - ~20 Apache Group members, including IBM

Once upon a time ... mid 1994

- Rob McCool and the NCSA httpd 1.3
  - public domain source code
  - beta testers
- Mosaic (Netscape) Communications grabs RobM
  - NCSA httpd development stagnates
- Rewrite of HTTP specification begins
- Patches proliferate
  - webmasters exchange patches via www-talk@info.cern.ch

Once upon a time ... Feb. 1995

- Private e-mail discussion starts, proposing to
  - compile individual patches into a single source base
  - provide feedback to new NCSA team
  - ensure that the results remain open source and HTTP a non-proprietary, implemented standard
- Brian Behlendorf offers workspace on Hyperreal
- We decide how to decide (the voting process)
- Apache is chosen for the group name
- Discussion moves to new-httpd@apache.org

Founders

Brian Behlendorf	HotWired, California
Roy Fielding	UC Irvine, California
Rob Hartill	LANL, New Mexico
David Robinson	Cambridge, UK
Cliff Skolnick	Sun Microsystems, California
Randy Terbush	Zyzzyva, Nebraska
Robert Thau	MIT, Massachusetts
Andrew Wilson	Elsevier, Oxford, UK

Development Constraints

- Globally distributed
  - multiple time zones, varying work schedules
  - synchronous communication is expensive, conflicting
- Voluntary organizational environment
  - no Apache CEO, manager, or even secretary
  - organizational roles are shared, rotated
- Heterogeneous development platforms
  - any required tools must be ubiquitous
- Communication is limited to e-mail

Development Process Evolution

- Fostering Contributions
  - developer focus and avoiding starvation
  - code, code review, documentation, support
- Recognizing Ego
  - trust and good intentions
  - beware of maniacal focus
- Limits of volunteerism
  - eight knives and an apple (dining developer problem)
  - eight knives and a pumpkin
  - eight pumpkins and no knives

Patch - Vote - Build 1995

- Initial development issues
  - choosing among features and alternative fixes
  - avoiding server bloat
  - setting project direction
- Small quorum consensus
  - votes: +1 = yes, 0 = *shrug*, -1 = no/veto
  - three +1 and no veto required for patch approval
  - emphasizes code review
- One person would collect and build new release from old sources plus approved patches

Conflict begets Guidelines

- Equality versus Meritocracy
  - stepping on toes and starving volunteers
  - equal opinions among unequal developers
- Voters - Vote Coordinator - Release Builder
  - recognized that roles are separable, allowing rotation
- Apache Project Guidelines
  - established rights of main contributors
  - provided visible means of attaining membership
  - explained the process to new volunteers
  - revealed more opportunities to contribute

Replication 1996

- Improving the development experience
  - progress hindered by separate vote and build
  - patch conflicts lead to delay, bickering
- Concurrent Versioning System (CVS)
  - distributed the build task, avoiding costly merges
  - free-for-all during period between big releases
  - review-and-commit during beta testing
- Secure Shell (ssh)
  - eases remote actions
  - improves site security (just in time)

Dislocation 1996-97

- No structure, no focus
  - shifts in primary developers
  - HTTP/1.1 specification "finished"
  - code review weakens, disappears
- GNATS problem tracking system
  - allow users to help document and track problems
- STATUS agenda
  - focused development on 1.2 release
  - document votes on current patches, issues
  - highlight showstoppers, problems needing patches

Commit-then-Review 1998

- Improving the development experience (again)
  - fragmentation of primary developer time
  - disjunct between reviews and working time
  - imbalance of contributions
- Lazy consensus when consensus is likely
  - commit changes first and review based on logs
- Automate some administrative actions
  - status in CVS, posted every other day
  - open PR summary posted once a week
- Jury is still out ...

26-Oct-98 13

Collaboration Techniques

- Collaborative development requires
  - at least one common goal
    - *but not all goals need to be common*
  - a means for communication
    - *both public and private*
  - a shared information space
    - *access to past communication (organizational memory)*
    - *access to past and current products*
  - coordination
    - *to make all of the above possible*

26-Oct-98 14

Mailing Lists @apache.org

- apache-announce
  - used only for important announcements to users
- new-httpd
  - primary developer discussion area
- apache-cvs
  - notifications of changes to shared repositories
- apache-bugdb
  - notifications of problem report creation/update
- others for related projects
  - http://dev.apache.org/mailing-lists.html

26-Oct-98 15

Shared Information Space

- www.apache.org
  - information for users, official public releases
- **dev.apache.org**
  - project guidelines and information for developers
  - tips for development and building a release
  - mailing list and tool information
- bugs.apache.org
  - problem report database
- modules.apache.org
  - third-party module registry

26-Oct-98 16

Coordination Tools

- ssh: Secure Shell remote login facility
  - authentication for remote access
  - http://www.cs.hut.fi/ssh/
- CVS: Concurrent Versioning System
  - manages replication, versioning, change notification
  - http://www.cyclic.com/cyclic-pages/CVS-sheet.html
- GNATS: Problem Reporting and Tracking System
  - entry, search, and notification [heavily modified]
  - http://www.alumni.caltech.edu/~dank/gnats.html
- Agenda: manually updated STATUS file

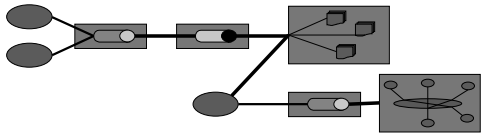
26-Oct-98 17

WWW Architectural Style

- Representational State Transfer
  - component roles
    - *client, server, user agent, origin server, proxy, cache*
  - connector semantics
    - *resource*
    - *representation of a resource*
    - *communication to obtain/modify representations*
  - application state and behavior
    - *web "page" as an instance of application state*
    - *engines to move from one state to the next*
      - browser, spider, any media type handler

26-Oct-98 18

Representational State Transfer



- optimized for transfer of typed data streams
- caching of representations allows application interaction to proceed without using network
- all components can be pipe-and-filter

HTTP Request/Response

```
GET /Test/hello.html HTTP/1.1
Host: kiwi.ics.uci.edu:8080
User-Agent: GET/7 libwww-perl/5.40

HTTP/1.1 200 OK
Date: Fri, 07 Jan 1997 15:40:09 GMT
Server: Apache/1.2b6
Content-type: text/html
Transfer-Encoding: chunked
Etag: "a797cd-465af"
Cache-control: max-age=3600
Vary: Accept-Language

<HTML><HEAD>
 …
```


Apache Architecture

- Central core
  - server initialization and configuration primitives
  - connection setup and listen/accept
  - request protocol parsing and input/output buffers
  - pool-based memory allocation and utilities
  - HTTP phase-oriented module API hooks
- Modules
  - request rewriting or redirection
  - authentication and content handlers
  - miscellaneous features

Apache 2.0 Design

- Primary goals
  - layered abstractions for multithreading, shared memory, portability, and protocol streams
  - HTTP protocol extensions, WebDAV
  - new configuration language and run-time interface
  - more flexible, detailed module hooks and API
  - front-end caching and proxy/gateway awareness
- Waiting on …
  - issues with NSPR and Netscape Public License
  - fewer distractions from 1.3.x maintenance

Lessons for Software Engineers

- Disconnected Operation
  - network delays/failures interfere with focused work
  - the best tools for Internet collaboration are those that effectively minimize use of the Internet
- User-driven Development
  - generic benefits of open source
    - *more eyes to find problems and examine security*
    - *protection against obsolescence and discontinued products*
  - emphasizes features known to be useful
  - requires modularity and more extensible designs

Questions?

- Places to see:
  - Front Door — www.apache.org
  - Developer Notes — dev.apache.org
  - PR Database — bugs.apache.org
  - Apache Week — www.apacheweek.com
  - ApacheCon'98 — www.apachecon.com
- www.ics.uci.edu/~fielding/talks/apache98/