New Features in Apache HTTP Server 2.2

Justin R. Erenkrantz
University of California, Irvine
http://www.erenkrantz.com/oscon/
justin@erenkrantz.com

Why should I pay attention?

- *Apache HTTP Server Committer since 2001*
- *Involved in several of these new 2.2 features*
- *Been Release Manager (RM) several times*
- *Also involved with APR and Subversion*
- *Director and Treasurer, ASF*
- *'Spare' time: PhD student at UC Irvine*

Apache HTTP Server History

- *1.3.0 released in June, 1998*
- *2.0a1 released in March, 2000*
- *2.0.35 (first 2.0 GA) released April, 2002*
- *2.1 branch created in November, 2002*
- *2.1.1 released in November, 2004*
- *2.2.0 released on ?*

Overview of 2.0 Features

- *2.0 was a major architectural rewrite*
- *Provides a solid platform for further work*
- *Introduction of APR abstracts OS details*
- *Threadable MPMs = lower memory footprint*
- *Filters bridge a long-time gap present in 1.x*
- *IPv6 support, mod_ssl & mod_dav bundled*

Versioning

- *Stable: 2.0, 2.2, 2.4, etc.*
- *Developer (unstable): 2.1, 2.3, 2.5, etc.*
- *Branched 2.1 in November, 2002*
- *Branched 2.3 at ApacheCon EU last month*
- *Several alphas and betas of 2.1 available*
- *Haven't released as GA yet - Real Soon Now!*

Our Promise To You

- *Learned hard lesson from early 2.0 GAs*
- *Users kept getting frustrated at moving target*
- *Stable: Review-than-Commit (R-T-C)*
- *No changes enter ‘stable’ without review*
- *No binary-incompatible changes in ‘stable’*
- *Development: Commit-than-Review (C-T-R)*

Goals of 2.2 release

- *Incremental improvements*
- *Keep core architecture the same*
- *Almost all 2.0 modules are source-compatible*
- *Add features...*
  - *that our users want*
  - *that our developers need*

New Features in 2.2

- *Authentication providers introduced*
- *Production-quality caching*
- *AJP support and load balancing proxy*
- *More expressive filtering directives*
- *Event MPM*
- *...and more...*

Authentication

- *Stood up here at OSCON 2003 touting this refactored authentication engine*
- *Still not part of a GA release*
- *Digest authentication is now first-class*
- *No more authoritative directives*
- *Easy authentication config aliasing*

What is a Provider?

- *Originally introduced as part of mod_dav*
- *Slowly being incorporated into the rest of tree*
- *Registered on startup; called at run-time*
- *Therefore, for authentication:*
  - *Basic and Digest Auth each* ***use*** *providers*
  - *File systems, DBs, LDAP* ***are*** *providers*

Hooks versus Providers

- *Hooks are traditional Apache interfaces*
    - *Each request phase exposed as a hook*
- *Hooks: Haphazard; RUN_REALLY_LAST*
- *Provider: registers vtable of functions*
- *Lookup a provider by name; call functions*
- *Versioned provider interfaces allow growth*

Provider API Example 1/2

Common code:

```
#define AUTHN_PROVIDER_GROUP "authn"

typedef struct {
    /* Given a username and password, expected to return AUTH_GRANTED
     * if we can validate this user/password combination. */
    authn_status (*check_password)(request_rec *r, const char *user,
                                   const char *password);
    /* Given a user and realm, expected to return AUTH_USER_FOUND if we
     * can find a md5 hash of 'user:realm:password' */
    authn_status (*get_realm_hash)(request_rec *r, const char *user,
                                   const char *realm, char **rethash);
} authn_provider;
```

Provider API Example 2/2

Provider:

```
static const authn_provider authn_file_provider ={
        &check_password,
        &get_realm_hash,
};
ap_register_provider(p, AUTHN_PROVIDER_GROUP, "file", "0",
&authn_file_provider);
```

Authentication Engine:

```
/* lookup and cache the actual provider now */
newp->provider = ap_lookup_provider(AUTHN_PROVIDER_GROUP,
                                    newp->provider_name, "0");
auth_result = newp->provider->check_password(r, sent_user, sent_pw);
```

Digest Authentication

- *Previously, digest auth only with files*
  - *Complicated to implement ‘digest’*
- *Implement via provider (get_realm_hash)*
- *mod_authn_dbm now handles digest auth*
- *Auth providers can easily implement digest*
- *Be aware of broken MSIE digest auth!*

2.0 Authentication Example

```
<Location /use-basic>
   AuthType Basic
   AuthName “Private Area”
   AuthUserFile /example/.htpasswd
   Require valid-user
</Location>
<Location /use-digest>
   AuthType Digest
   AuthName “Private Area”
   AuthUserFile /example/.htpasswd
   Require valid-user
</Location>
```

- *Still works with 2.2!*

2.2 Authentication Example 1/3

```
<Location /use-basic>
   AuthType Basic
   AuthName “Private Area”
   AuthBasicProvider file
   AuthUserFile /example/.htpasswd
   Require valid-user
</Location>
<Location /use-digest>
   AuthType Digest
   AuthName “Private Area”
   AuthDigestProvider file
   AuthUserFile /example/.htpasswd
   Require valid-user
</Location>
```

2.2 Authentication Example 2/3

```
<Location /use-file-and-ldap>
    AuthType Basic
    AuthName “Private Area”
    AuthBasicProvider file ldap
    AuthUserFile /example/.htpasswd
    AuthLDAPURL ldap://ldap.example.com/o=Example
    Require valid-user
</Location>
```

mod_authn_alias

- *Allows complicated authentication setups*
  - *Not previously possible to do*
  - *Had to introduce 'new' modules*
- *Can refer to same provider more than once*
- *Dynamically register aliases with config*
- *Use alias in the Auth*Provider directives*

2.2 Authentication Example 3/3

```
<AuthnProviderAlias ldap ldap-alias>
   AuthLDAPBindDN cn=youruser,o=ctx
   AuthLDAPBindPassword yourpassword
   AuthLDAPURL ldap://ldap.host/o=ctx
</AuthnProviderAlias>
<AuthnProviderAlias ldap ldap-other>
   AuthLDAPBindDN cn=yourotheruser,o=ctx
   AuthLDAPBindPassword yourotherpassword
   AuthLDAPURL ldap://other.ldap.host/o=ctx
</AuthnProviderAlias>
<Location /use-aliased-ldap>
   AuthBasicProvider ldap-alias ldap-other
   Require valid-user
</Location>
```

mod_cache

- *Rewrite evolved out of discussions at last year’s OSCON with HP’s Madhu Mathihalli*
- *“Why was Zeus beating Apache 2.0?”*
- *Correctness and flexibility over speed*
- *Madhu identified a number of causes*
- *First and foremost, was lack of caching*

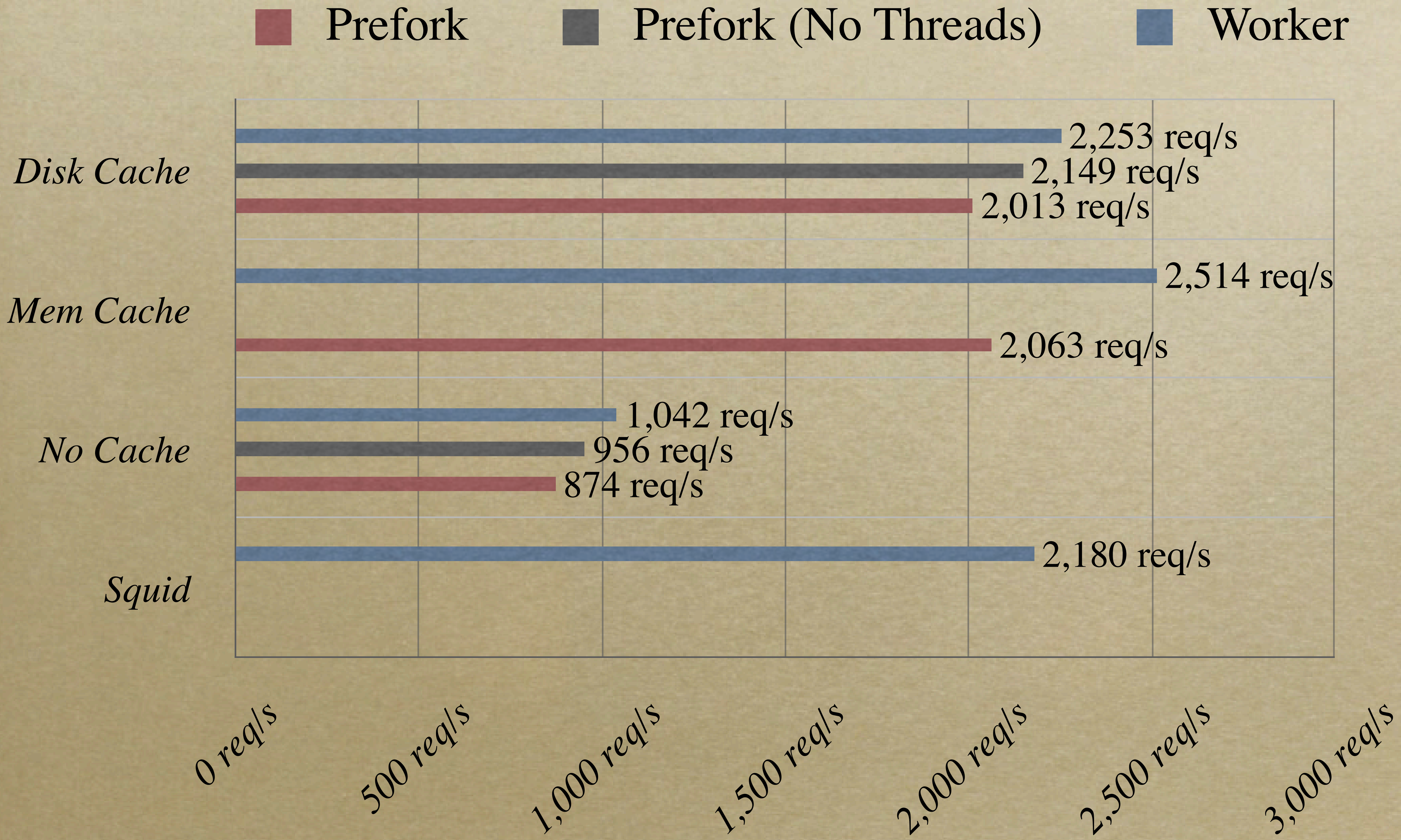
Experimental mod_cache

- *Yet, 2.0 did indeed have a cache engine*
- *Code was twisty passages all alike; Ugh!*
- *Horribly broken - disk caching ineffective*
    - *Read files byte-by-byte; no zero-copy*
- *Not RFC-compliant; labeled ‘experimental’*
- *Started a rewrite in August 2004*

Memory cache or disk cache?

- *Prior efforts focused on memory cache*
- *Disk cache was a second thought*
- *Shared memory cache implies locking*
  - *Lack of persistence: restart loses cache*
- *Disk cache allows zero-copy transfers*
- *mod_cache is no longer ‘experimental’*

Cache Performance Results



mod_cache Example

```
CacheEnable disk / [CacheEnable mem /]
CacheRoot /var/cache/apache
CacheDirLevels 5
CacheDirLength 3

CacheIgnoreCacheControl off
CacheIgnoreHeaders None
CacheIgnoreNoLastMod On
CacheStorePrivate On

CacheDefaultExpire 600
CacheMaxExpire 3600

ExpiresDefault “now plus 2 hours”
```

htcacheclean

- *mod_disk_cache does not limit cache size*
- *Use htcacheclean to limit cache size*

```
htcacheclean -d15 -p/var/cache/apache -l100M
```

- *Runs every 15 min. and limits size to 100MB*
- *-p option should match CacheRoot directive*
- *-n (nice) option makes it sleep; be careful!*

mod_proxy rewrite

- *mod_proxy has an unusual history*
  - *First included in 1.1 (circa 1996)*
  - *Punted when 2.0 started as it was broken*
  - *Came back just in time for 2.0 GA*
  - *Rewritten in 2.1/2.2 (Mladen Turk)*
- *Is the third time the charm? We hope.*

Multi-protocol support

- *HTTP/0.9, HTTP/1.0, HTTP/1.1*
- *SSL traffic*
    - ***Very** useful for reverse proxies*
- *AJP13 - Tomcat's mod_jk protocol*
- *FTP (only supports GET)*
- *CONNECT (SSL Proxying)*

Load balancing

- *Distribute load to several servers*
- *Request counting or weighted traffic average*
- *Backend connection pooling / reuse*
- *Sticky sessions based on cookie values*
- *Configurable while system is online*

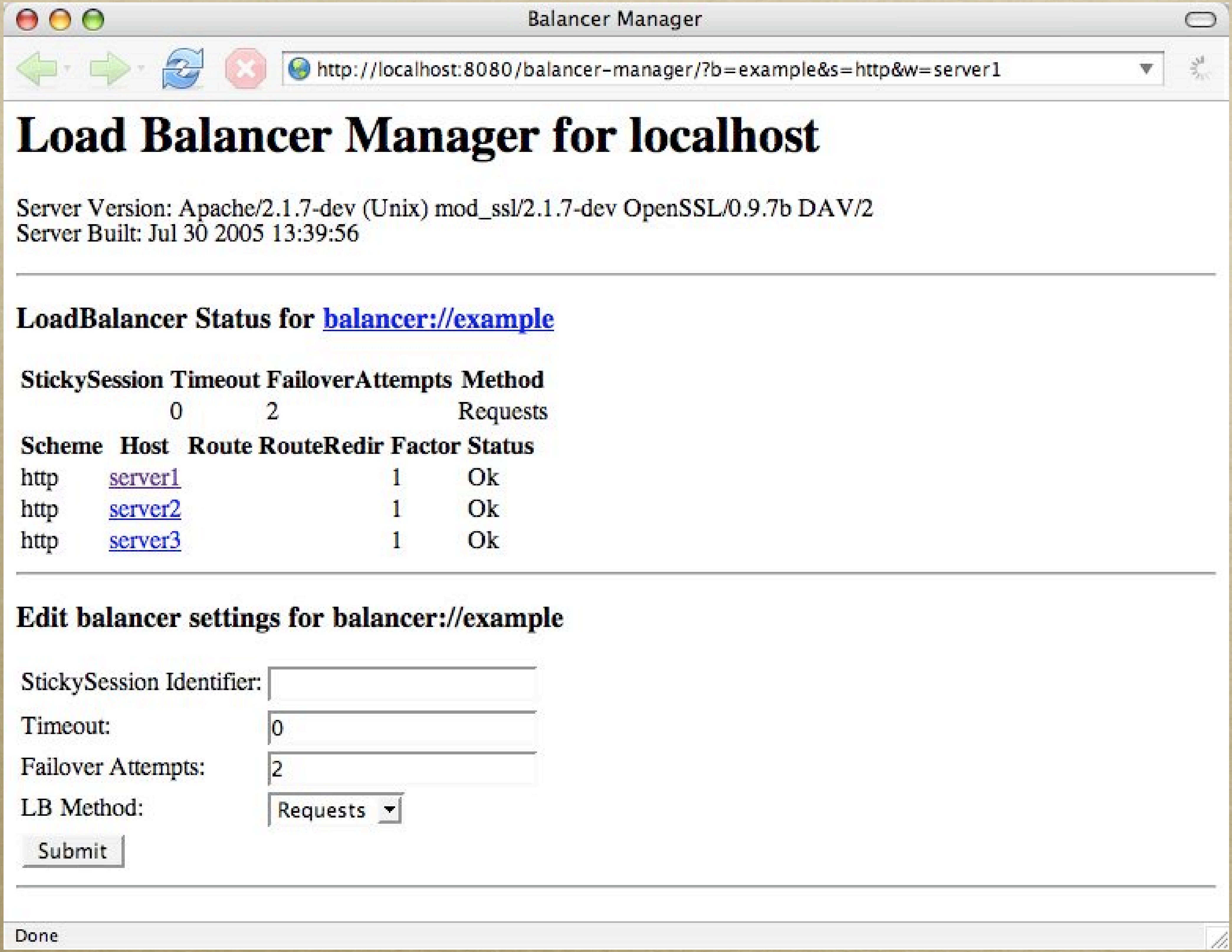
mod_proxy Example 1/2

Using Balancers:

```
ProxyPass / balancer://example/
<Proxy balancer://example/>
    BalancerMember http://server1/
    BalancerMember http://server2/
    BalancerMember http://server3/
</Proxy>
```

Balancer Manager Interface:

```
ProxyPass /balancer-manager !
<Location /balancer-manager>
    # <insert authentication here>
    SetHandler balancer-manager
</Location>
```



Balancer Manager
http://localhost:8080/balancer-manager/?b=example&s=http&w=server1

Load Balancer Manager for localhost

Server Version: Apache/2.1.7-dev (Unix) mod_ssl/2.1.7-dev OpenSSL/0.9.7b DAV/2
Server Built: Jul 30 2005 13:39:56

LoadBalancer Status for balancer://example

StickySession	Timeout	FailoverAttempts	Method
	0	2	Requests

Scheme	Host	Route	RouteRedir	Factor	Status
http	server1			1	Ok
http	server2			1	Ok
http	server3			1	Ok

Edit balancer settings for balancer://example

StickySession Identifier:	
Timeout:	0
Failover Attempts:	2
LB Method:	Requests

mod_proxy Example 2/2

Connection reuse:

```
ProxyPass /example http://backend.example.com min=0 max=20 smax=5
    ttl=120 retry=300
ProxyPassReverse /example http://backend.example.com/
```

Option	Description	Default
min	*Minimum number of connections to keep open*	0
max	*Maximum connections to keep open to server*	*1 or n**
smax	*(Soft maximum) Try to keep this many connections open*	***max***
ttl	*Time to live for each connection above smax*	*none*
retry	*If conn. fails, wait this long before reopening conn.*	*60 sec*

* = *If threaded MPM, use ThreadsPerChild; otherwise 1*

mod_filter

- *Dynamic configuration of output filters*
- *More powerful than AddOutputFilterByType*
  - *Suitable for proxies (where ByType fails)*

```
FilterDeclare SSI
FilterProvider SSI INCLUDES \
          resp=Content-Type $text/html
FilterChain SSI
```

- *Also allows for chaining*

mod_filter Example

```
FilterProvider unpack jpeg_unpack Content-Type $image/jpeg
FilterProvider unpack gif_unpack Content-Type $image/gif
FilterProvider unpack png_unpack Content-Type $image/png

FilterProvider downsample downsample_filter \
                Content-Type $image
FilterProtocol downsample "change=yes"

FilterProvider repack jpeg_pack Content-Type $image/jpeg
FilterProvider repack gif_pack Content-Type $image/gif
FilterProvider repack png_pack Content-Type $image/png

<Location /image-filter>
   FilterChain unpack downsample repack
</Location>
```

Event MPM (Experimental)

- *Dispatches ‘idle’ requests to ‘event’ thread*
  - *Resolves DoS with hogging connections*
- *Experimental: doesn’t yet work with SSL*
- *Requires:*
  - *Linux 2.6 (EPoll), Solaris 10 (Event Ports)*
  - **BSD / Mac OS X 10.4+ (KQueue)*

mod_dbd

- *apr_dbd introduced in APR-util 1.2.0*
    - *Abstracts out database layers*
    - *Postgres, SQLite, MySQL (separate)*
- *mod_dbd provides conn. pooling and sharing*
    - *Best on threaded platforms / MPMs*
- *See http://www.apachetutor.org/dev/reslist*

Other Features in 2.2

- *mod_ssl: TLS Upgrade inside HTTP*
- *mod_info: Hooks, file / line numbers*
- *Large file support enabled by default*
- *libpcre updated to 5.0*
- *Dump loaded modules, httpd -M*
- *httxt2dbm: DBM Files for RewriteMap*

HTTP Server’s Summer of Code

- *mod_smtpd: SMTP protocol handler*
    - *c.f. Apache::Qpsmtpd via mod_perl*
- *Perchild MPM: Per-user vhosts*
    - *Many have tried; many have failed.*
- *mod_cache prefetching*
- *mod_mbox interface changes*

Future Directions

- *mod_ftp donation from Covalent*
- *Getting rock solid caching and proxy*
- *Asynchronous server support*
- *Rewrite configuration internals*
- *Abstract out file system backend*

Thanks!

Questions?

Slides: http://www.erenkrantz.com/oscon/

Thanks to Paul Querna for his slides.