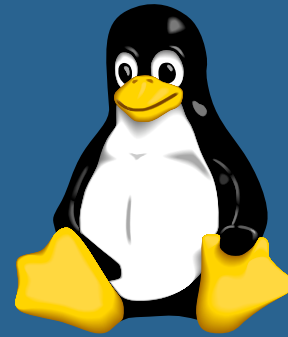
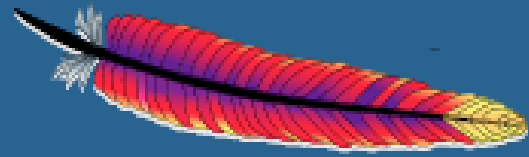


# apc@facebook

Brian M. Shire

PHP Works 2007, Atlanta  
September 14th, 11:15 am - 12:15 pm



Customized source for Facebook's requirements.

Global collaboration is beneficial.

“Own the stack”.

Original Facebook releases:

**Thrift, PHP Shell** and the **Firefox Toolbar**.

<http://developers.facebook.com/opensource.php>

<http://pecl.php.net/packages/APC/>

“APC is a free, open, and robust framework for caching and optimizing PHP intermediate code.”

APC Developers:

George Schlossnagle	Ilia Alshanetsky
Daniel Cowgill	Marcus Börger
Rasmus Lerdorf	Sara Golemon
Gopal Vijayaraghavan	Brian Shire
Edin Kadribasic	

Provides a significant PHP interpreter performance increase.

PHP variable cache can also be used for application specific benefits.

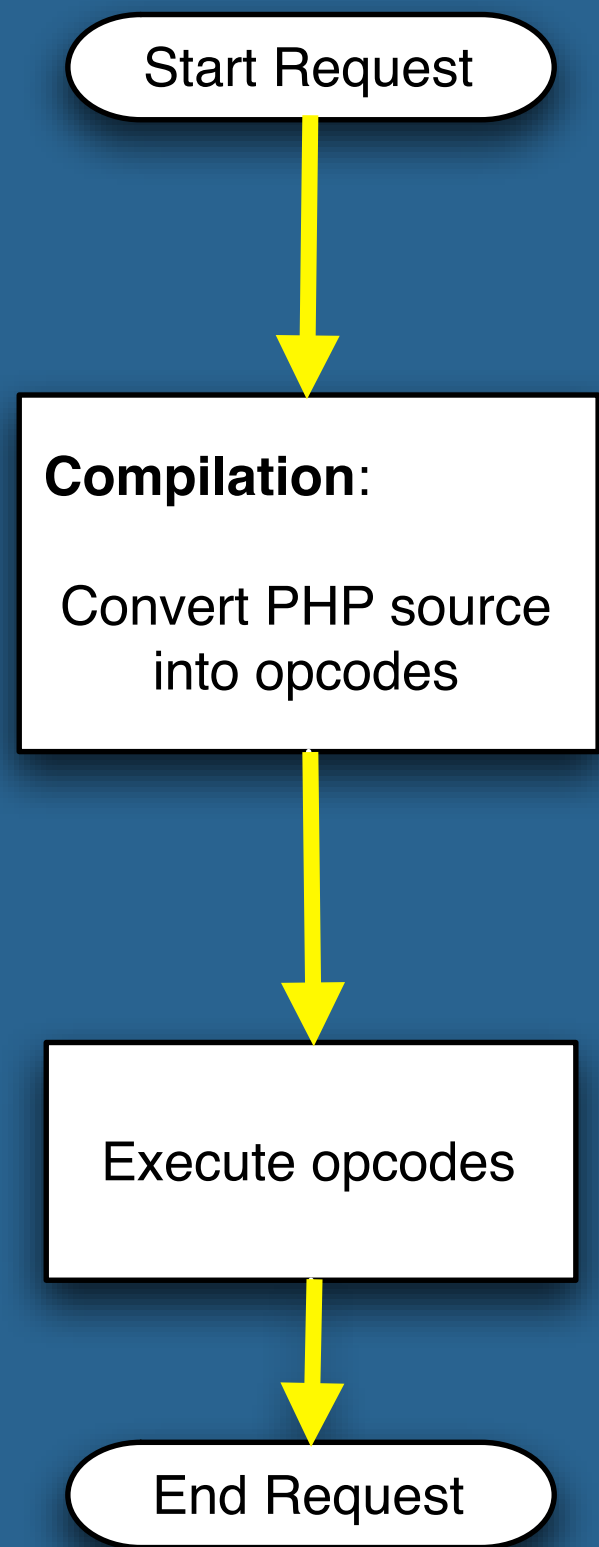
Open source allows collaboration, bug fixes and optimizations.

Requests per second provide a blunt measurement of performance gains.

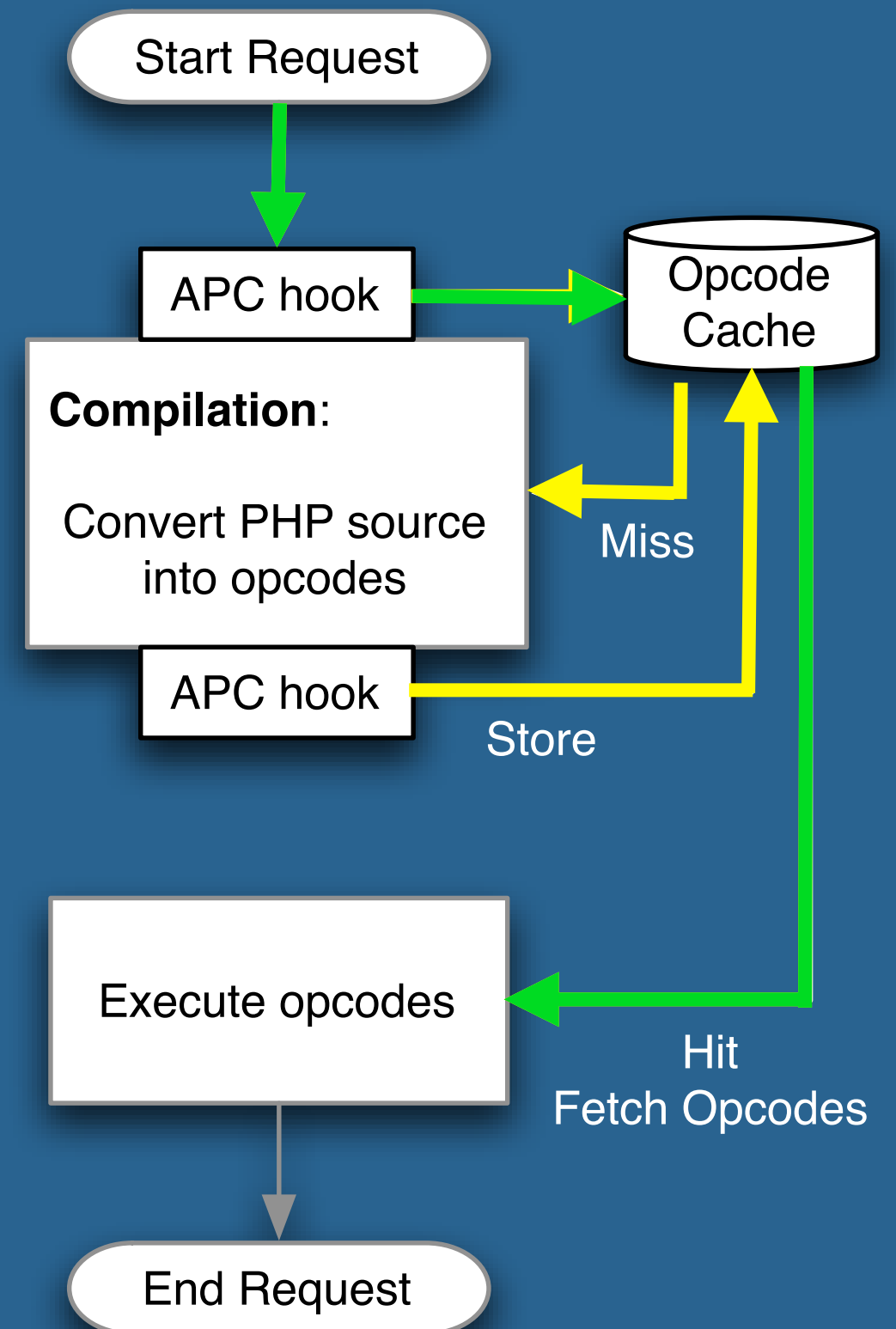
Configuration	Requests Per Second		
PHP	2.47 rps.		—————
APC User	5.24 rps.	2.12 x	—————
APC User & File	22.01 rps.	8.91 x	—————
Facebook	29.56 rps.	11.96 x	—————

Measurements made using Apache Bench (ab) with 1000 total requests, concurrency of 40. Executed on a dual Dual Core AMD Opteron 2.2Ghz, 8GB RAM.

PHP



APC



## PHP Source

```
1 <?php
2 $output = 'Hello World!';
3 echo $output;
4 ?>
```

## Opcodes

```
1 ASSIGN !0 'Hello+World%21'
2 ECHO !0
3 RETURN 1
4 ZEND_HANDLE_EXCEPTION
```

## PHP Source

```
1 <?php
2 $output = 'Hello World';
3 if($_GET['exclaim']) {
4     $output .= '!';
5 } else {
6     $output .= '.';
7 }
8 echo $output;
9 ?>
```

## Opcodes

```
1 ASSIGN          !0, 'HELLO+WORLD'
2 FETCH_R        GLOBAL $1, '_GET'
3 FETCH_DIM_R    $2, $1, 'exclaim'
4 JMPZ           $2, ->6
5 ASSIGN_CONCAT  !0, '%21'
6 JMP            ->7
7 ASSIGN_CONCAT  !0, '.'
8 ECHO           $0
9 RETURN        1
10 ZEND_HANDLE_EXCEPTION
```

APC Support	enabled
Version	3.0.15-dev
MMAP Support	Enabled
MMAP File Mask	/tmp/apc.7JwUre
Locking type	File Locks
Revision	\$Revision: 3.150 \$
Build Date	Aug 28 2007 17:08:57

Directive	Local Value	Master Value
apc.cache_by_default	On	On
apc.enable_cli	On	On
apc.enabled	On	On
apc.file_update_protection	0	0
apc.filters	<i>no value</i>	<i>no value</i>
apc.gc_ttl	3600	3600
apc.include_once_override	Off	Off
apc.localcache	Off	Off
apc.localcache.size	512	512
apc.max_file_size	10M	10M
apc.mmap_file_mask	/tmp/apc.7JwUre	/tmp/apc.7JwUre
apc.num_files_hint	200	200
apc.report_autofilter	On	On
apc.rfc1867	On	On
apc.rfc1867_freq	0	0
apc.rfc1867_name	APC_UPLOAD_PROGRESS	APC_UPLOAD_PROGRESS
apc.rfc1867_prefix	upload_	upload_
apc.shm_segments	1	1
apc.shm_size	500	500
apc.slam_defense	0	0
apc.stat	Off	Off
apc.stat_ctime	Off	Off
apc.ttl	7500	7500
apc.user_entries_hint	162000	162000
apc.user_ttl	7500	7500
apc.write_lock	On	On

phpinfo() displays configuration information

Verify configuration settings have taken place

Debug configuration problems



<b>apc.enable</b>	<b>0</b>	Enable APC
-------------------	----------	------------

<b>apc.enabled_cli</b>	<b>0</b>	Enable APC when running via the command line
------------------------	----------	--

<b>apc.mmap_file_mask</b>	<b>NULL</b>	Name of the file mask where as specified by mmap
---------------------------	-------------	--

<b>apc.shm_segments</b>	<b>1</b>
<b>apc.shm_size</b>	<b>30</b>

@ Facebook:

```
apc.shm_segments=1  
apc.shm_size=648
```

One shared memory segment required

Size needs to hold all file and user entries

Monitor usage using apc.php

Extra space is required for deleted entries

Behaves poorly when memory is at capacity

<b>apc.num_files_hint</b>	<b>1000</b>
<b>apc.user_entries_hint</b>	<b>4096</b>

@ Facebook:

```
apc.num_files_hint=100  
apc.user_entries_hint=640000
```

Hints optimize hash lookup tables

Maximum number of files or user entries

## Locking type

### File Locks

*Default*

Uses file locking operations

Stable, not effecient

### IPC Semaphore Locks

Faster alternative to file locks

### Linux Futex Locks

*EXPERIMENTAL*

Architecture specific, Linux Kernel 2.6.x or later

Significant performance gain

### pthread mutex Locks

*EXPERIMENTAL*

*Currently Used @ Facebook*

Better alternative to Linux Futex Locks

Same performance gain

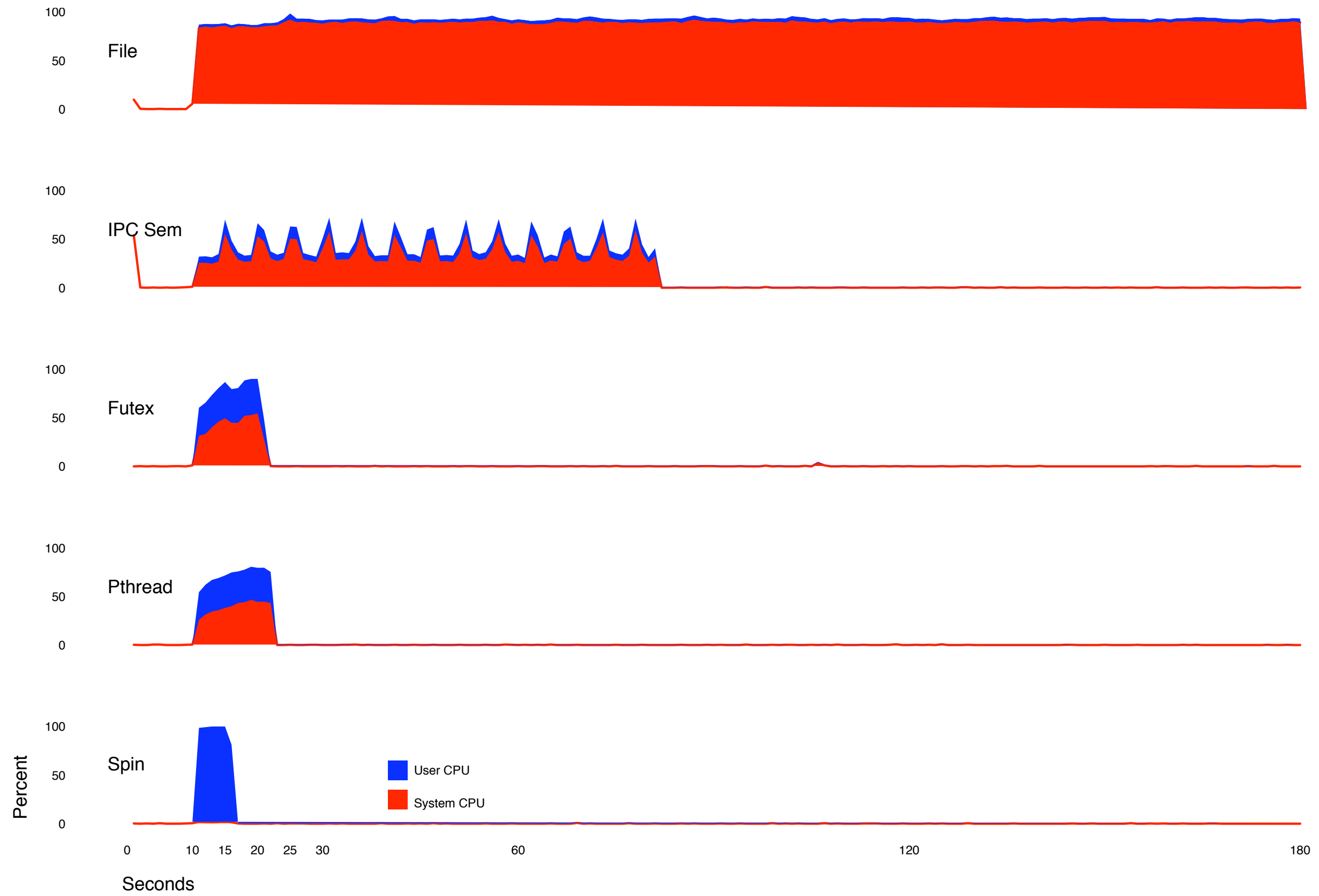
More stable, more architecture support

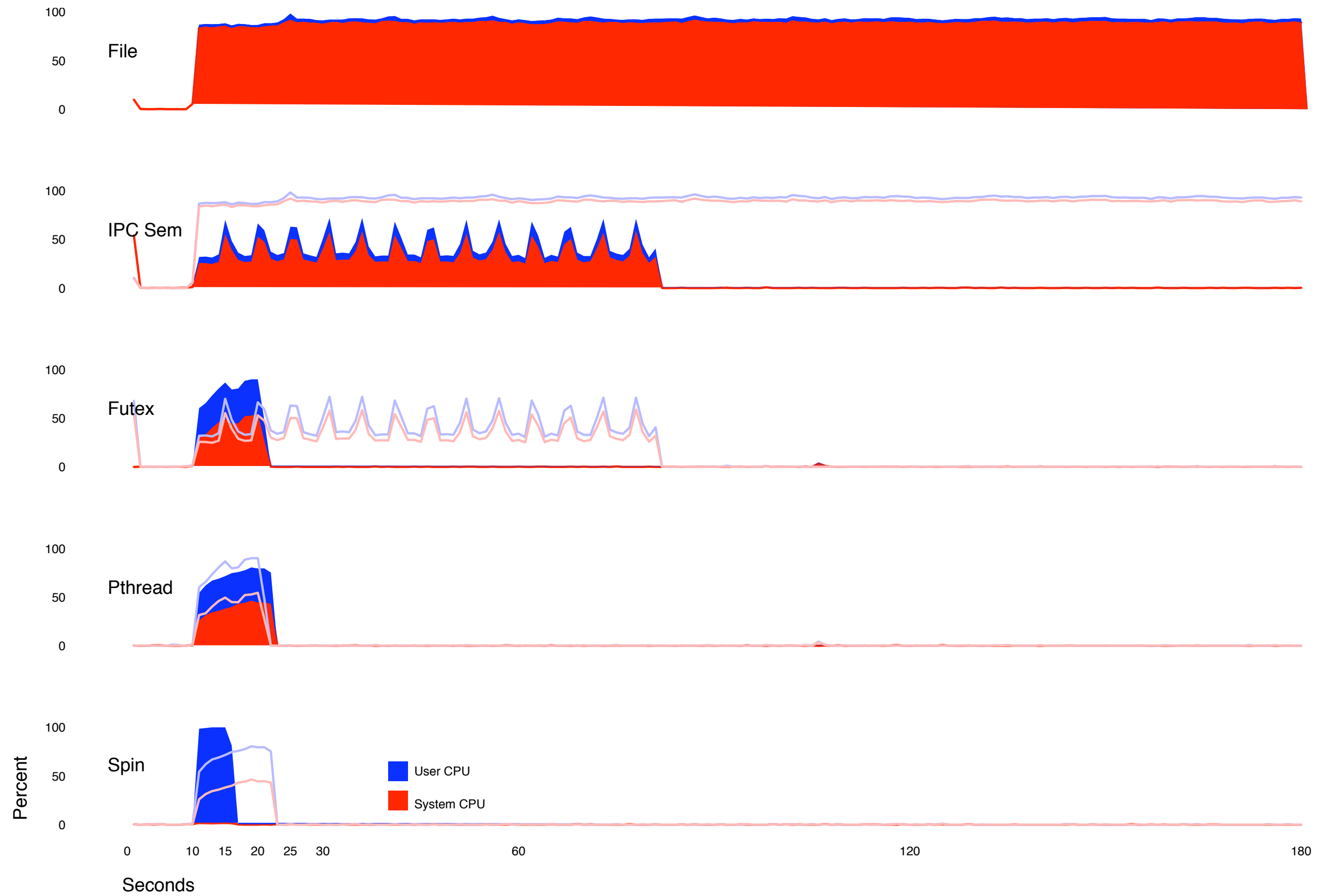
### spin Locks

*EXPERIMENTAL*

Ported from the PostgreSQL project

Runs in user space





**apc.stat** **TRUE**

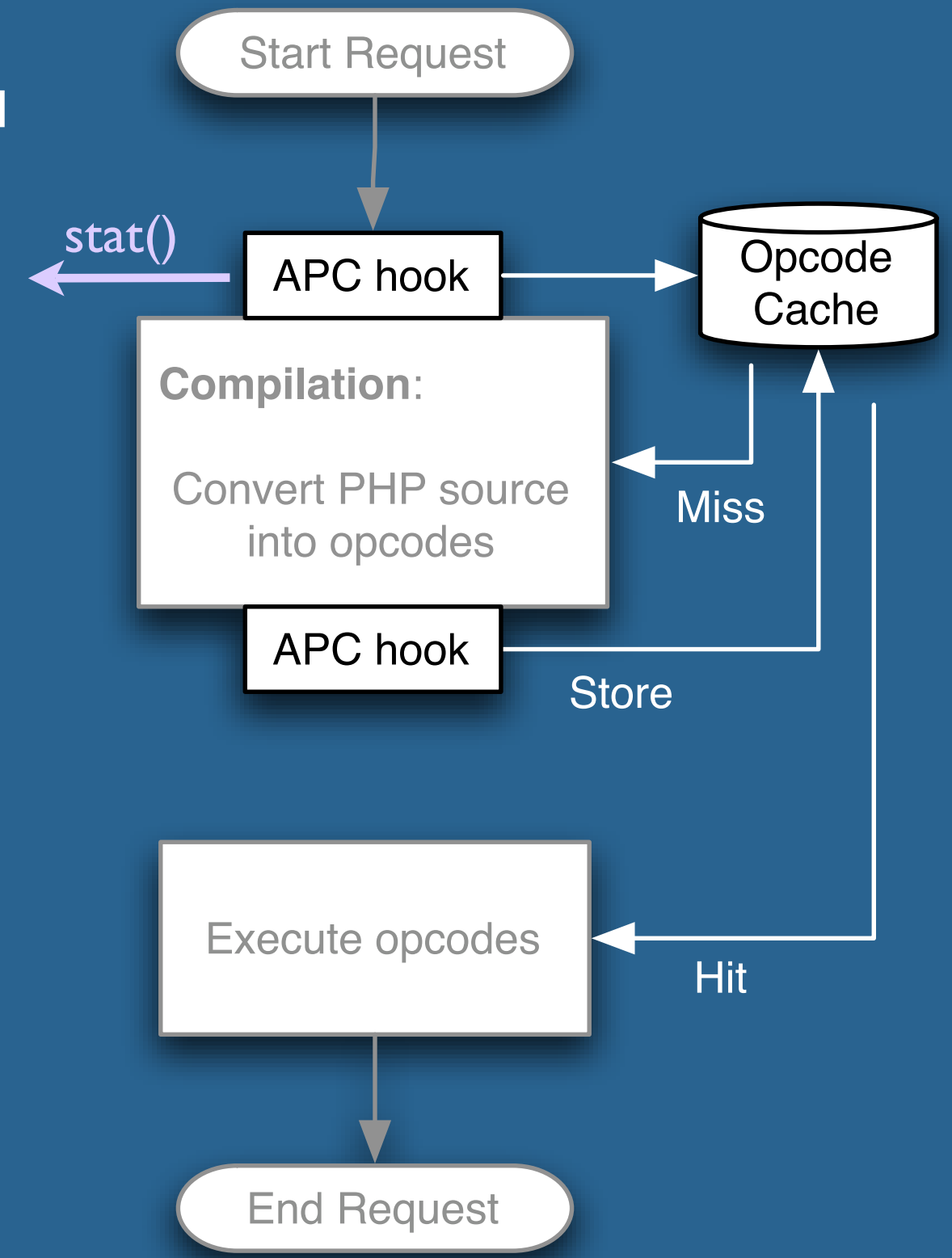
@ Facebook:  
`apc.stat=FALSE`

APC stats files to determine if they've been updated  
Disabling updates will increase performance  
Requires restart or `apc_cache_clear()` to update

**apc.stat\_ctime** **FALSE**

@ Facebook:  
`apc.stat_ctime=FALSE`

CVS, SVN and rsync backdate modified times  
`stat_ctime` checks the creation time for updates



<b>apc.slam_defense</b>	<b>0</b>
-------------------------	----------

Spreads load on startup by only caching given percent of requests

<b>apc.write_lock</b>	<b>1</b>
-----------------------	----------

A non-blocking write lock provides a better solution to setting slam\_defense

<b>apc.file_update_protection</b>	<b>2</b>
-----------------------------------	----------

Read updated files after given delay to prevent loading incomplete files

<b>apc.ttl</b>	<b>0</b>
----------------	----------

“Time to Live”

<b>apc.user_ttl</b>	<b>0</b>
---------------------	----------

Maximum time a cache entry can remain in cache.

<b>apc.gc_ttl</b>	<b>3600</b>
-------------------	-------------

Inline garbage collector removes deleted entries from cache as soon as possible. Entries still in use by a request will be removed after the gc\_ttl has expired.

<b>apc.max_file_size</b>	<b>IM</b>	Limits the maximum file size that will be cached.
<b>apc.filters</b>	<b>NULL</b>	A regular expression that excludes files from being cached.
<b>apc.cache_by_default</b>	<b>I</b>	Setting to zero causes files to only cache if they match apc.filters.
<b>apc.report_autofilter</b>	<b>0</b>	Logs files excluded due to early/late binding issues



<b>apc.include_once_override</b>	<b>0</b>
----------------------------------	----------

EXPERIMENTAL

Optimizes include\_once() calls

<b>apc.rfc1867</b>	<b>0</b>
--------------------	----------

EXPERIMENTAL

Upload progress support

<b>apc.localcache</b>	<b>0</b>
-----------------------	----------

<b>apc.localcache_size</b>	<b>512</b>
----------------------------	------------

EXPERIMENTAL

A process localized cache

**APC**  
Opcode Cache

Refresh Data | **View Host Stats** | System Cache Entries | User Cache Entries | Version Check

**General Cache Information**

APC Version	3.0.15-dev
PHP Version	5.2.2-dev
APC Host	shirebook.local
Server Software	Apache/1.3.37 (Darwin) PHP/5.2.2-dev
Shared Memory	1 Segment(s) with 800.0 MBytes (mmap memory, file locking)
Start Time	2007/04/30 23:32:27
Uptime	3 minutes
File Upload Support	1

**File Cache Information**

Cached Files	2 (397.6 KBytes)
Hits	15
Misses	2
Request Rate (hits, misses)	0.08 cache requests/second
Hit Rate	0.07 cache requests/second
Miss Rate	0.01 cache requests/second
Insert Rate	0.01 cache requests/second
Cache full count	0

**User Cache Information**

Cached Variables	0 ( 0.0 Bytes)
Hits	0
Misses	0
Request Rate (hits, misses)	0.00 cache requests/second
Hit Rate	0.00 cache requests/second
Miss Rate	0.00 cache requests/second
Insert Rate	0.00 cache requests/second
Cache full count	0

**Runtime Settings**

apc.cache_by_default	1
apc.enable_cli	1
apc.enabled	1
apc.file_update_protection	0
apc.filters	
apc.gc_ttl	3600
apc.include_once_override	0
apc.localcache	0
apc.localcache.size	512
apc.max_file_size	1M
apc.mmap_file_mask	/tmp/apc.IXpNut
apc.num_files_hint	200
apc.report_autofilter	0
apc.rfc1867	0
apc.shm_segments	1
apc.shm_size	800
apc.slam_defense	0
apc.stat	1
apc.stat_ctime	0
apc.ttl	7500
apc.user_entries_hint	162000
apc.user_ttl	7500
apc.write_lock	1

**Host Status Diagrams**

Memory Usage

Hits & Misses

**Detailed Memory Usage and Fragmentation**

Fragmentation: 0%

apc.php is located in the APC source directory

User and file cache browser

Graphs of hit rates and memory usage

<http://us.php.net/manual/en/ref.apc.php>

*array* **apc\_cache\_info**(*string* cache, *boolean* limited)

Cache information

*array* **apc\_sma\_info**(*boolean* limited)

Shared memory segment information.

*boolean* **apc\_store**(*string* key, *mixed* value, *int* ttl)

Store key/value pair in cache.

*boolean* **apc\_add**(*string* key, *mixed* value, *int* ttl)

Add key/value pair if key isn't in cache.

*mixed* **apc\_fetch**(*string* key)

Fetch the value associated with key.

*boolean* **apc\_delete**(*string* key)

Delete the value associated with key.

*boolean* **apc\_clear\_cache**(*string* cache)

Clear all entries from the cache.

*boolean* **apc\_compile\_file**(*string* file)

Compile given file and store in cache, bypass all filters.

*boolean* **apc\_define\_constants**(*string* key,  
                                  *array* constants  
                                  *bool* case\_sensitive)

Define an array of key/value constants.

*boolean* **apc\_load\_constants**(*string* key,  
                                  *bool* case\_sensitive)

Assign each key/value constants in cache using define()

User cache stores PHP variables across multiples requests on a per server basis.

Primary commands for utilizing the user cache:

```
boolean apc_store(string key, mixed value)
```

```
mixed apc_fetch(string key)
```

Optimize...

Application configuration

Statistics such as site usage, request types, error conditions, timing

N<sup>th</sup> tier cache in addition to memcache or other caching service

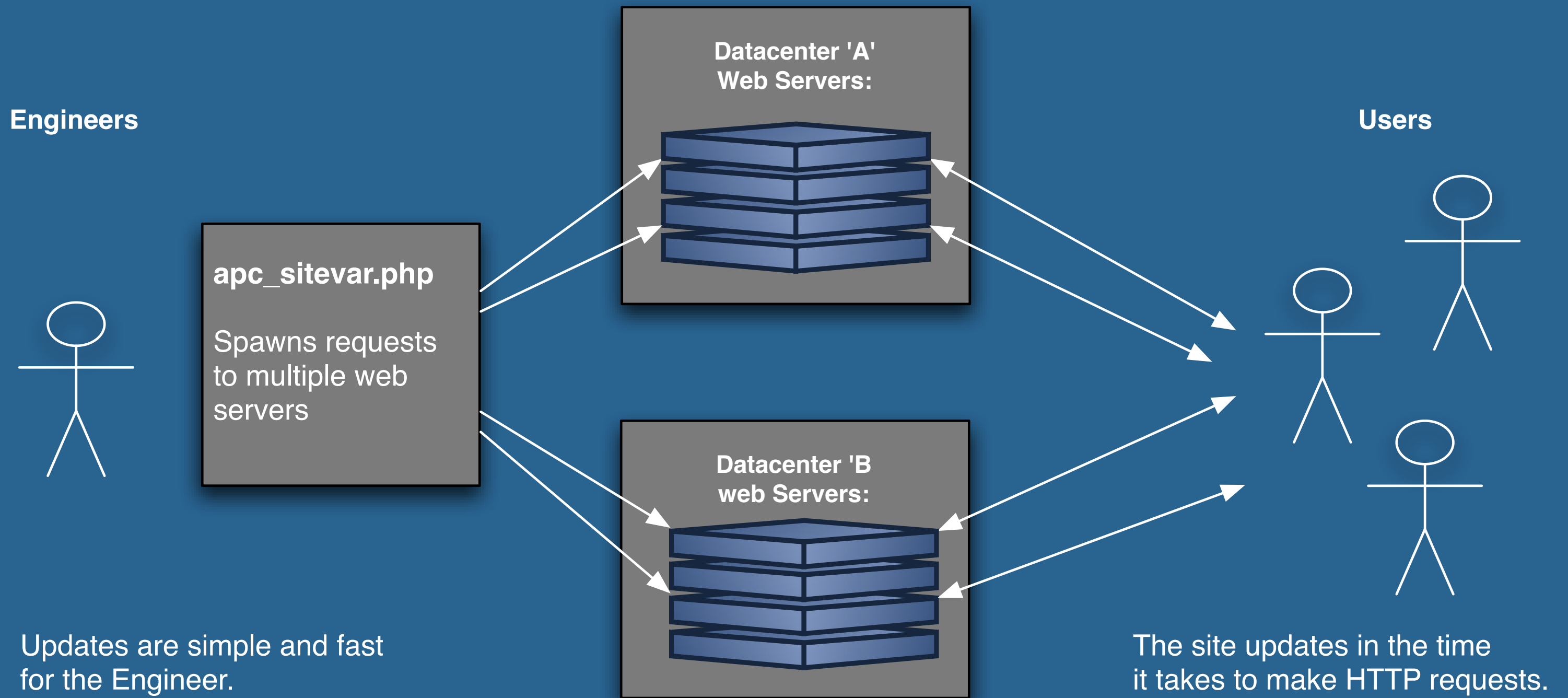
Database backed values that only change rarely like product listings

HTML or other output

Site behavior like new features, A/B tests, or rate controls

Facebook controls site configuration and features via the user cache.

### Controlling Site Behavior with "Sitevars"



Facebook primes it's cache before each code push or server restart.

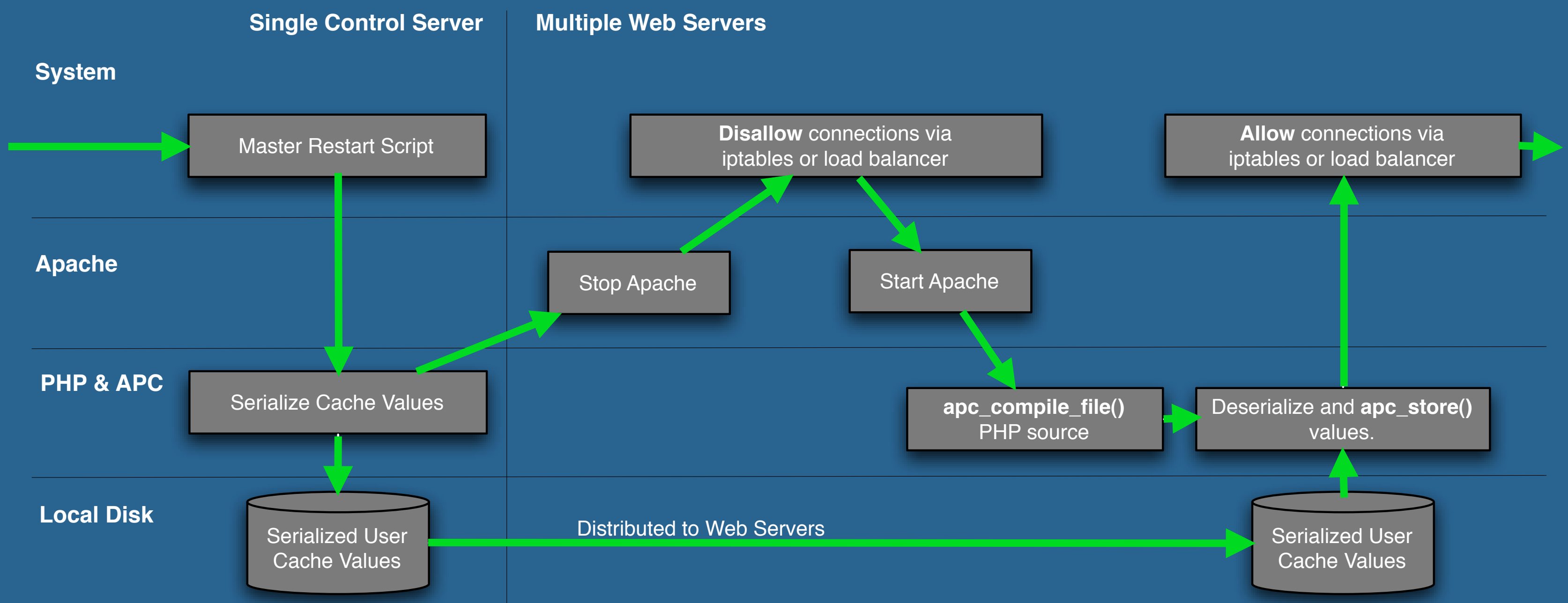
Ready to serve requests without delay due to compilation or updating cache values.

Handles immediate flood of requests with limited warm-up time.

Starts with same cache state limiting differences between servers.

```
boolean apc_store(string key, mixed value)
```

```
boolean apc_compile_file(string file)
```



Installation via “pecl install apc”  
or source <http://pecl.php.net/packages/APC/>

Start with a basic apc.ini file:

```
apc.enabled=1  
apc.shm_size=100M
```

Install apc.php under the DocumentRoot, configure the USER and PASS variables  
Monitor apc.php for usage and adjust configuration

Try tuning some of the discussed settings to meet application needs

Add some APC user variables

Try different locking types

Try the apc.stat=0 setting

Measure changes in CPU usage and maximum requests per second



# Q & A

