

SUPERCHARGING PHP pages with mod_lsapi in CloudLinux OS



suPHP / CGI

- slow
- doesn't support .htaccess

mod_php

- insecure
- requires MPM prefork

mod_fcgid

- hard to keep stable / requires tuning
- sub-optimal caching
- no support for .htaccess
- hard to do multi-versions / customers

FPM

- sub-optimal security due to the way sockets used
- no support for .htaccess
- hard to do multi-versions
- separate pool per customer / sub-optimal memory usage



- Blazing fast
- Outperforms any other Apache PHP handler
- Supports mod_php options format

```
<Directory /usr/local/www/phpMyAdmin>  
php_admin_value open_basedir none  
</Directory>
```

- Full opcode caching
- No tuning required and easy to install
- Supports .htaccess PHP directives

```
php_value include_path      "./usr/local/lib/php"  
php_flag display_errors    Off  
php_value upload_max_filesize 2M
```

- Supports CageFS and PHP selector
 - opens possibility of multiple PHP versions per client
- Secure, based on suexec mechanism
- Potential for even better performance



Why mod_lsapi?





FULLY SUPPORTS:

- APC
- xCache
- Zend OPcache

OPCODE CACHE

Opcode cache uses shared memory, that shares code across requests for end user. `mod_lsapi` configured not to reset cache on Apache restarts. Something impossible to do with `mod_php`

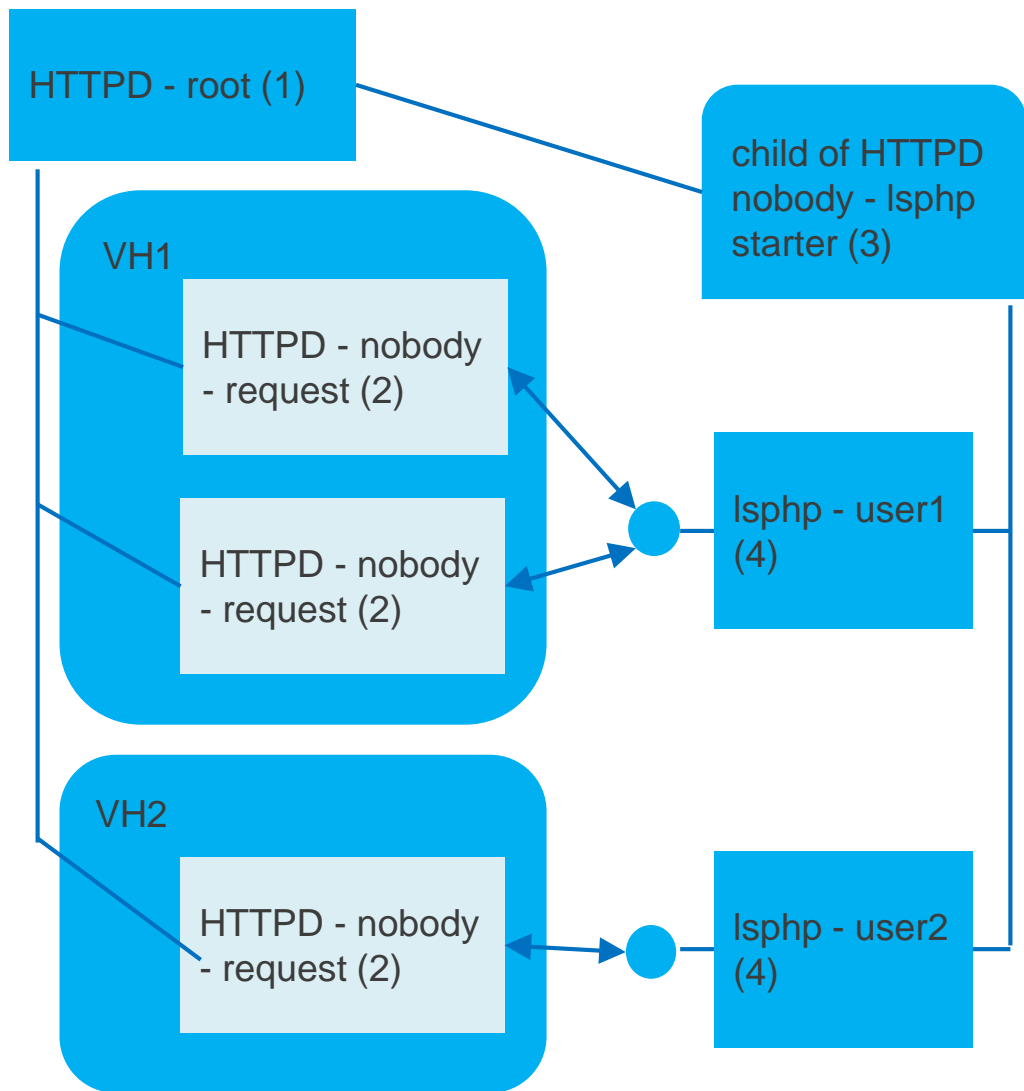
Isphp - PHP + LSAPI

What is LSAPI?

LiteSpeed Server Application Programming Interface (LSAPI) is designed specifically for seamless, optimized communication between LiteSpeed Web Server and third party web applications. Now this protocol is available for Apache 2.2/2.4.



Using LSAPI, we have seen greater performance than Apache with mod_php, easy installation than php-fpm, easy integration with control panel. LSAPI means faster, more stable dynamic web pages.



HOW IT WORKS

- ✓ apache passes handling for PHP request to mod_Isapi
- ✓ mod_Isapi uses libIsapi to transfers request Isphp daemon
- ✓ Isphp processes request and returns data to mod_Isapi.
- ✓ If no Isphp processes available when new request comes, new Isphp process created.

1. apache module handle requests (2) - mod_Isapi.so;
2. starter daemon - starts Isphp on demand (3) - from mod_Isapi.so, child of HTTPD - block (1);
3. Isphp - executes php file and sends results back to (2) (4) - native Isphp or Isphp from alt-php)



1. Install common packages:

```
yum install liblsapi liblsapi-devel
```

2. Install mod_lsapi sources and hooks:

```
yum install mod_lsapi
```

3. Prepare environment for work

```
/usr/bin/switch_mod_lsapi --setup
```

Next steps for cPanel with EasyApache3 only

4. Detect native php version and build correct lsphp binary file:

```
/usr/bin/switch_mod_lsapi --build-native-lsphp
```

5. Enable mod_lsapi:

```
/usr/bin/switch_mod_lsapi --enable-global
```

6. Restart apache:

```
service httpd restart
```

INSTALLATION



INSTALLATION

EasyApache 4



1. Install common packages:

```
yum install liblsapi liblsapi-devel
```

2. Install mod_lsapi package:

```
wget http://repo.cloudlinux.com/cloudlinux/sources/cloudlinux_ea3_to_ea4  
sh cloudlinux_ea3_to_ea4 --convert --mod_lsapi
```

3. Prepare environment for work (creating directory with correct rights for socket and make native lsphp as php 5.6 version):

```
/usr/bin/switch_mod_lsapi --setup
```

4. Enable mod_lsapi:

```
/usr/bin/switch_mod_lsapi --enable-global
```

5. Restart apache:

```
service httpd restart
```




DirectAdmin custombuild has full support of mod_lsapi. Isphp native builds during custombuild rebuilding process.

1. ***cd /usr/local/directadmin/custombuild***
2. ***./build update***
3. ***./build set php1_release lsphp***
4. ***./build php n***
5. ***./build apache***

DirectAdmin INSTALLATION



How to enable mod_Isapi on ISPManager:

1. `yum install liblsapi liblsapi-devel --enablerepo=cloudlinux-updates-testing`
2. `yum install mod_Isapi --enablerepo=cloudlinux-updates-testing`
3. `/usr/bin/switch_mod_Isapi --setup`
4. uncomment string `LoadModule lsapi_module modules/mod_Isapi.so` from file `/etc/httpd/conf.d/mod_Isapi.conf`
5. disable php support for needed domain (this action comment out `AddHandler` or `AddType` for `VirtualHost`) or for all domains.

6. remove from `/etc/httpd/conf/httpd.conf` strings:

```
<Directory /var/www/*/data/>  
php_admin_flag engine off  
</Directory>
```

or

6. add to needed (where `mod_Isapi` should be enabled) `VirtualHost` such strings:

```
<Directory /var/www/[username]/data/www/[domain]>  
Options -ExecCGI -Includes  
php_admin_flag engine on  
</Directory>
```

7. uncomment string `AddType application/x-httpd-lsphp .php5 .php4 .php .php3 .php2 .phtml` in file `/etc/httpd/conf.d/mod_Isapi.conf`
8. `service httpd restart`



	No CP	cPanel	DirectAdmin	Plesk	InterWorx	ISPManager
install	+	+	+ (no need in manual calling)	+	+	+
uninstall	+	+	-	+	+	+
enable-domain	-	+	-	-	-	-
disable-domain	-	+	-	-	-	-
enable-global	-	+	+/- (custombuild)	-	-	-
disable-global	-	+	-	-	-	-
build-native-lsphp	-	+	+/- (custombuild)	-	-	-

What commands are available for different control panels

There are two ways to make native lsphp:

1. QUICK - supports all type of panels - native lsphp makes from alt-php56 (switch_mod_lsapi --setup)

```
cp /opt/alt/php56/usr/bin/lsphp /usr/local/bin/
```

2. SLOW - detects version of native php and build needed sources according to installed php (cPanel only):

```
switch_mod_lsapi --build-native-lsphp
```

! DirectAdmin has own native lsphp builder

```
/usr/local/directadmin/custombuild/build set php1_release lsphp  
/usr/local/directadmin/custombuild/build php n
```

**Native php - php installed and used before alt-php packages was installed. lsphp binary usually not present on server without LiteSpeed and this means that it should be created (build from php sources with such options as usual php binary file but with LSAPI protocol built-in)*



mod_lsapi is a drop in replacement for suPHP. No configuration changes need to be done.

RECOMMENDED PROCEDURE:

1. Install mod_lsapi and call

```
/usr/bin/switch_mod_lsapi --setup
```

2. Switch individual domain to check how it goes:

```
/usr/bin/switch_mod_lsapi --enable-domain example.com
```

3. Switch whole server globally so all domains start working on mod_lsapi:

```
/usr/bin/switch_mod_lsapi --enable-global
```



All mod_lsapi configuration options are placed in single config file. The file will not be changed or updated by any package reinstalls, easyapache builds on cPanel or custombuild on DirectAdmin.

cPanel with EasyApache 3

`/usr/local/apache/conf/conf.d/lsapi.conf`

cPanel with EasyApache 4

`/etc/apache2/conf.d/mod_lsapi.conf`

DirectAdmin

`/etc/httpd/conf/extra/mod_lsapi.conf`

RPM based

`/etc/httpd/conf.d/mod_lsapi.conf`



- mod_lsapi errors will be located in regular apache **error_log** and own **sulspHP_log**.
- Errors from clients like PHP notices, code errors are written to **error_log** file.
- Errors from PHP starter are written to **sulspHP_log** file, they are related running lsphp as service and are mostly helpful in getting error 500 in browser.

WHERE TO FIND ERRORS





mod_lsapi allow to use different handlers for different php versions. For example file with extension .php53 can be handled by php5.3 and file with extension .php handled by php5.5 without PHP Selector.

This is done setting handlers and php binaries for them in **/etc/container/php.handler** file

EXAMPLE:

```
# cat /etc/container/php.handler  
application/x-lsphp53 /opt/alt/php53/usr/bin/lsp  
application/x-lsphp55 /opt/alt/php55/usr/bin/lsp
```

Default handler for lsp is - application/x-httpd-lsp . Setting following options in domain .htaccess file :

```
AddType application/x-httpd-lsp .php .php5 .php4.php3 .php2 .phtml  
AddType application/x-lsphp53 .php53
```

Means that file index.php53 will be processed by php5.3, but index.php processed by native lsp

LoadModule lsapi_module modules/mod_lsapi.so

```
<IfModule lsapi_module>
```

```
<FilesMatch "\.(php|php4|php5|php53|php54|php55|php56|php6|phtml)$">
```

```
AddType application/x-httpd-lsphp .php .php4 .php5 .php53 .php54 .php55 .php56 .phtml
```

```
AddHandler application/x-httpd-lsphp .php .php4 .php5 .php53 .php54 .php55 .php56 .phtml
```

```
</FilesMatch>
```

```
lsapi_debug Off
```

```
lsapi_set_env TEMP "/tmp"
```

```
lsapi_set_env TMP "/tmp"
```

```
lsapi_set_env TMPDIR "/tmp"
```

```
</IfModule>
```



Most of php file extensions are covered by AddType option. If you want to use additional handlers for directory for existing extension, you should remove conflicting extension from main lsapi.conf file



Isapi_backend_children

- # Maximum number of simultaneously running child backend processes.
- # How many requests in parallel it can process
- # Optional, default value is 120

Isapi_backend_pgrp_max_idle

- # How long a control backend process will wait for a new request before it exits, in sec. Default value is 30
- # Decrease to minimize memory usage. Increase to make faster response from sites
- # 0 stands for infinite.

Isapi_backend_max_process_time

- # The maximum processing time allowed when processing a request, in sec.
- # Optional, default value is 300.



Isapi_backend_max_idle

- # How long a backend child process will wait for a new request before it exits, in sec
- # Optional, default value is 300. 0 stands for infinite.
- # Decrease to minimize memory usage

Isapi_backend_max_reqs

- # How many requests each child process will handle before it exits automatically.
- # Optional, default value is 10000.



Q: Is it compatible with EasyApache3/4?

Q: Is it compatible with PHP directives in .htaccess?

Q: I have httpd.conf with **SuExecUserGroup** options. Do I need to add mod_Isapi related options?

Q: How do I install mod_Isapi on Plesk panel?

Q: Can mod_Isapi work without CageFS?

Q: How many servers are running mod_Isapi now?



Connection pool

mod_lsapi tries to use pool of persistent connections to lsphp processes. Persistent connections to backends significantly increase module performance. mod_lsapi can keep up to 50 persistent connections for every socket name. In the case when the number of connections used at the same time exceeds the maximum, the module will process the next request in conventional, non-persistent mode.

Random socket name

Mod_lsapi will be use for communication with lsphp socket with random name

lsphp cold start improvement

Will be added notification mechanism for detection if lsphp was started

mod_lsapi+criu

Will be added criu support to speed up lsphp startup time

https://criu.org/Main_Page

FUTURE IMPROVEMENTS



UNINSTALL mod_lsapi



cPanel Servers

```
$ /usr/bin/switch_mod_lsapi --uninstall
```

DirectAdmin servers

```
$ cd /usr/local/directadmin/custombuild  
$ ./build update  
$ ./build set php1_release [any other php type]  
$ ./build php n  
$ ./build apache
```

RPM:

```
$ yum erase mod_lsapi  
$ rm [path to mod_lsapi.conf]  
# restore standard php handler  
$ service httpd restart
```

TEST SERVER CONFIGURATION:



Intel(R) Core(TM) i5-2300 CPU @ 2.80GHz

8Gb memory

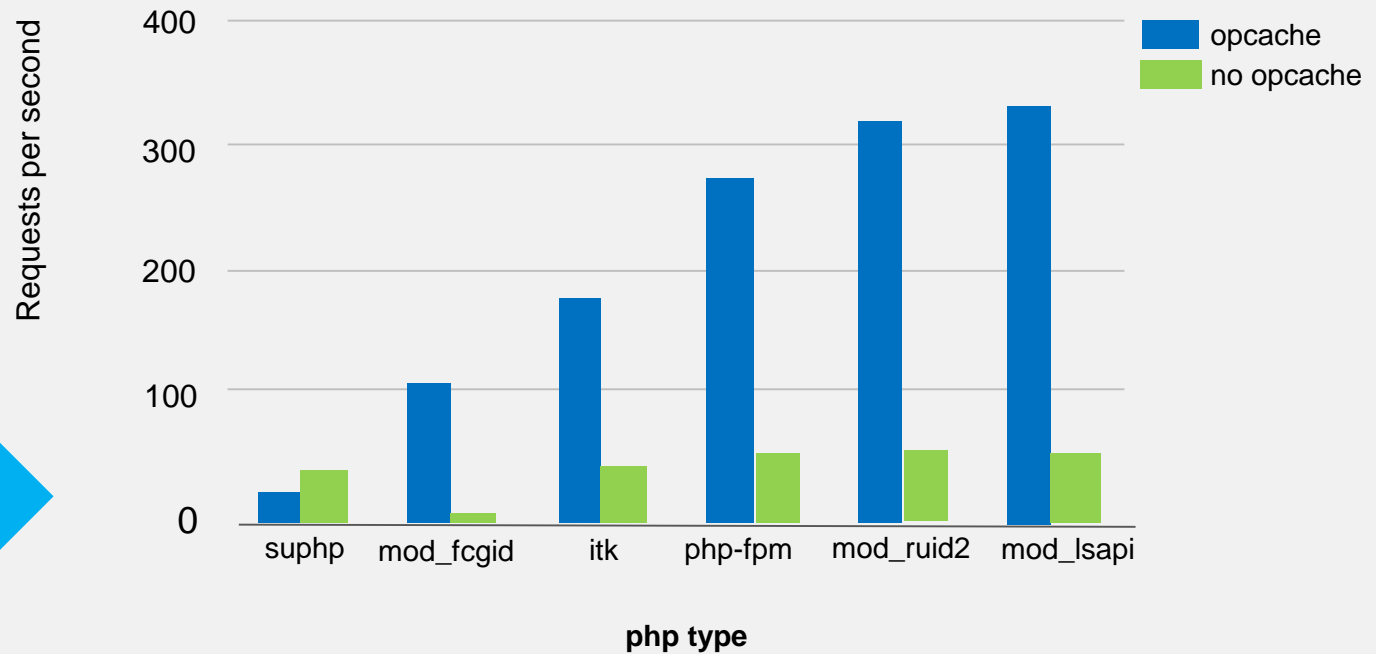
CloudLinux Server release 6.6 (Leonid Kizim)

php 5.3+no cagefs+no LVE(unlimited)

apache 2.2

- ✓ Test consists of hitting sites powered by php installed different ways (suphp, itk+mod_php, mod_ruid2 + mod_php etc).
- ✓ Tested site: wordpress (for checking opcache effect), simple script for checking environment effects, more complex php script for checking environment and opcache effects.
- ✓ Environment effects - fork, connect, poll and other algorithms in php, which can affect response time.

Wordpress test



wordpress test	opcache (req per second)	no opcache (req per second)
suphp	23.86	41.36
mod_fcgid	112.21	5.42
itk	180.49	45.36
php-fpm	277.42	52.71
mod_ruid2	322.14	53.77
mod_lsapi	325.56	54.12

Questions?



[Twitter.com/CloudLinuxOS](https://twitter.com/CloudLinuxOS)

[Visit CloudLinux.com](https://www.CloudLinux.com)