PHP Extension调研

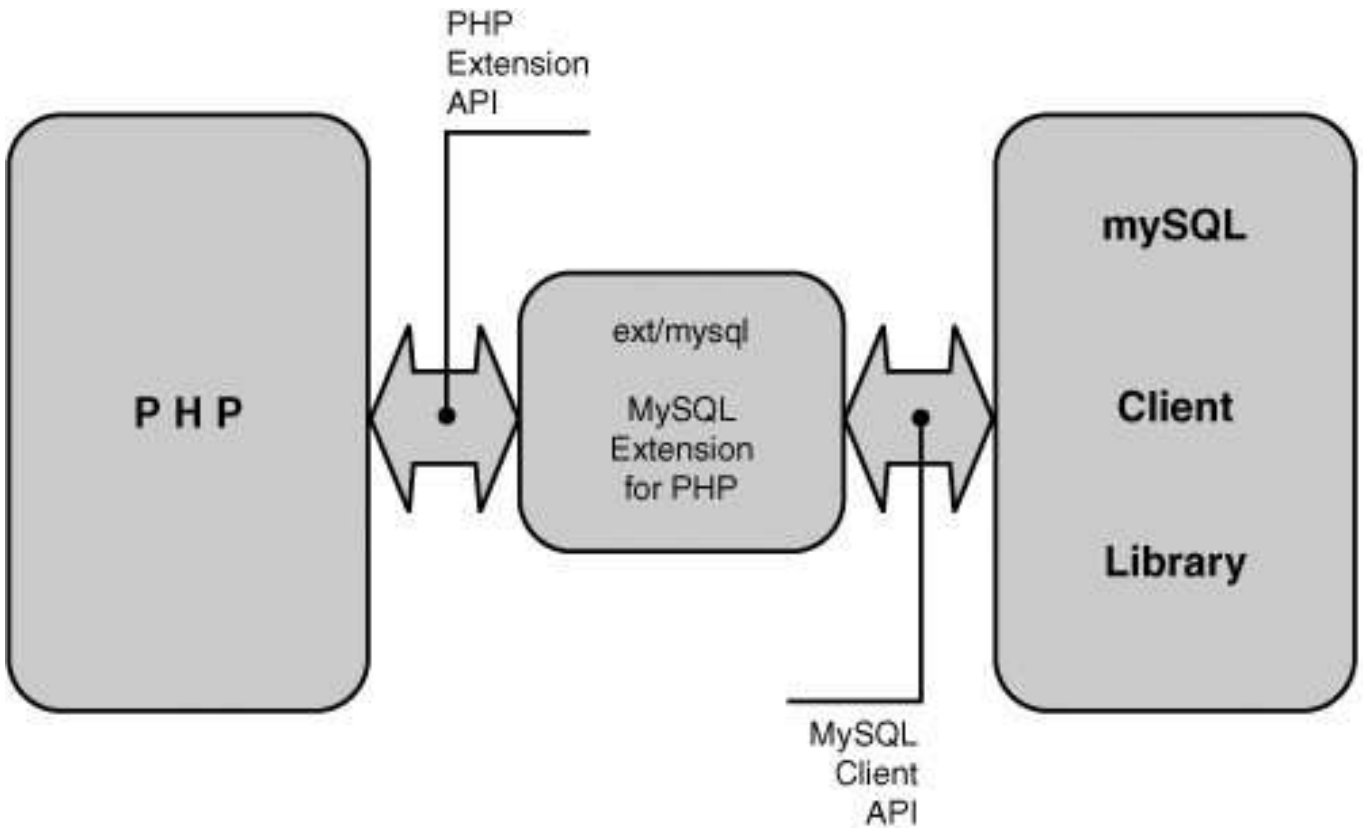
Ecom

jingmi@baidu.com

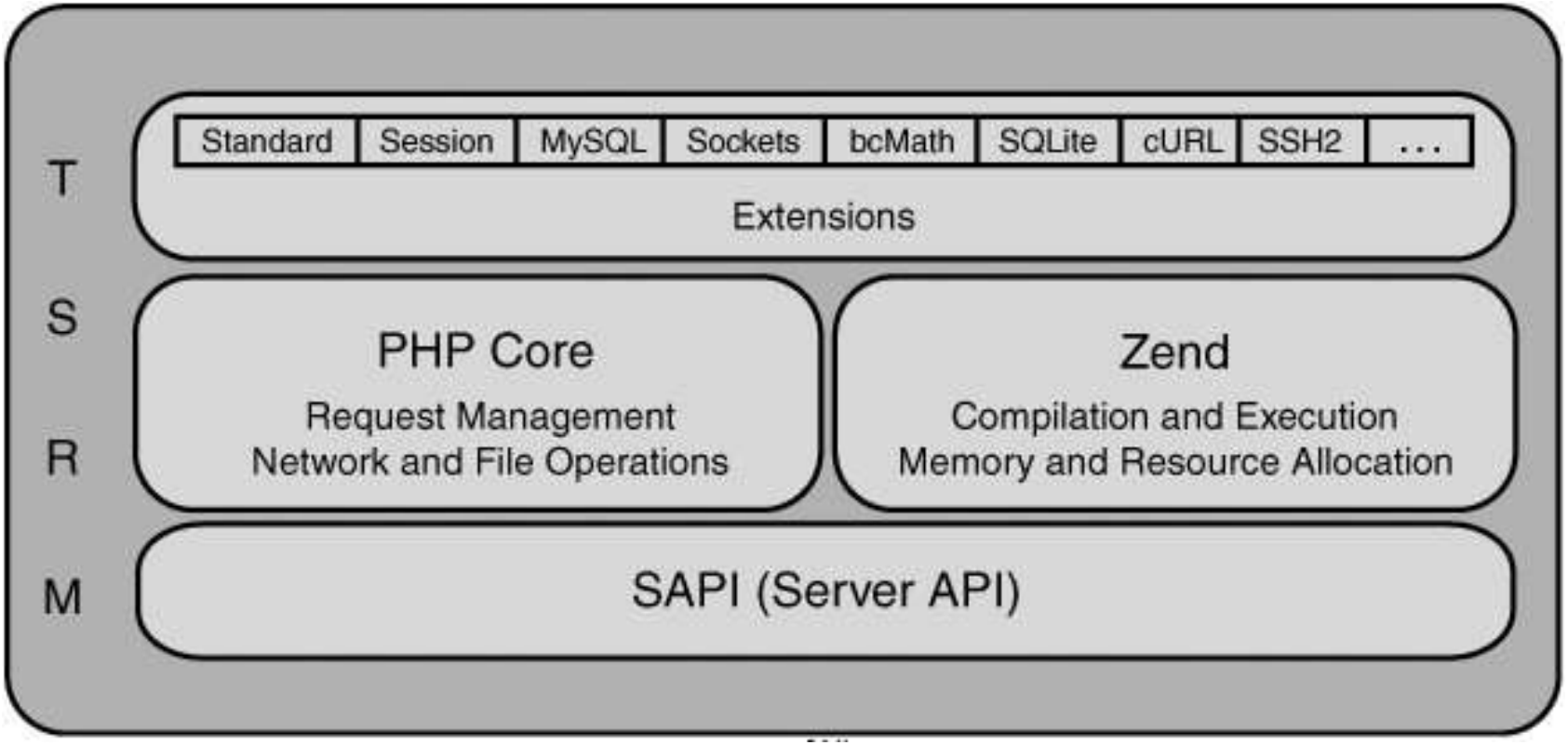
What is Extension？

- An extension is a discrete bundle of code that can be plugged into the PHP interpreter in order to provide additional functionality to userspace scripts.

Em， Extensions Is The Glue !

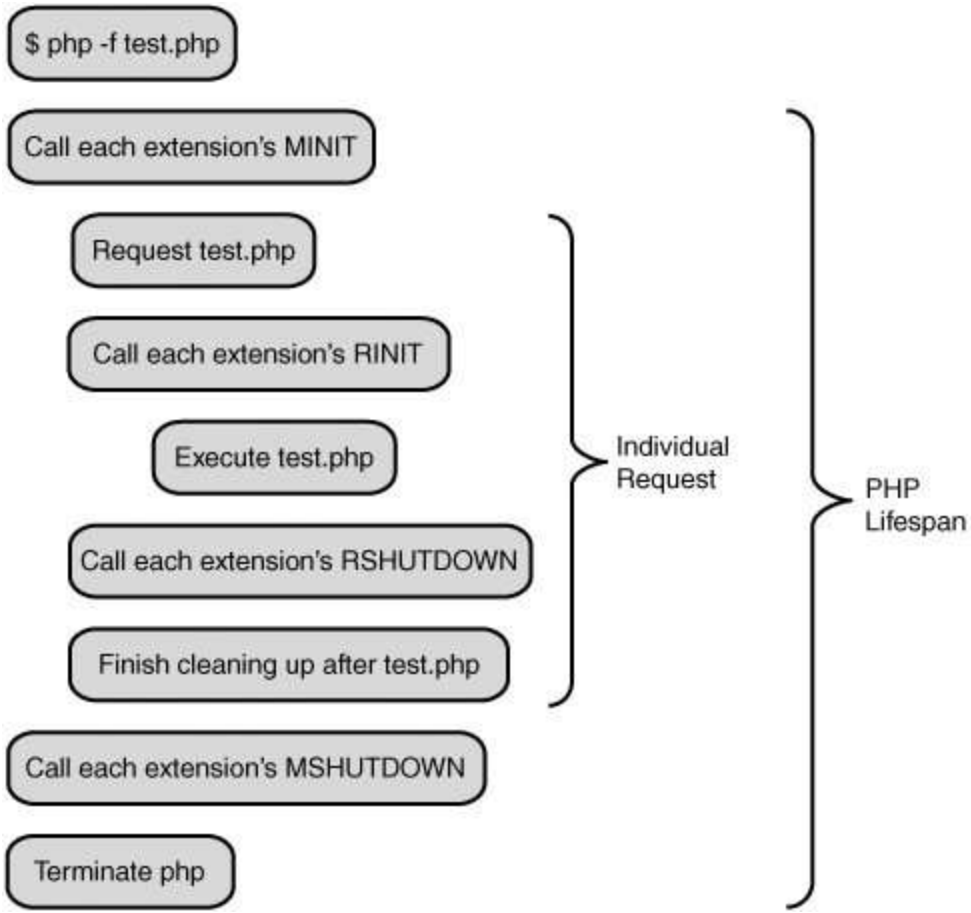


构架（Architecture）



TSRM (Thread Safe Resource Management)
SAPI (Server Application Programming Interface)

生命周期（Life Cycle）



Source:

```
<?php
echo “Hello, world\n”;
?>
```

Output:

```
[module] init
[module] rinit
hello, world
[module] rshutdown
[module] mshutdown
```

How to build extension? (1)

- 1. cd php-4.4.1/ext
- 2.建立文件夹

  ./ext_skel --extname=bd_base64
- 3.修改配置文件

  cd bd_base64

  vim ./config.m4

How to build extension? (2)

dnl If your extension references something external, use with:

```
PHP_ARG_WITH(bd_base64, for bd_base64 support,
Make sure that the comment is aligned:
[  --with-bd_base64           Include bd_base64 support])
```

dnl Otherwise use enable:

dnl PHP_ARG_ENABLE(bd_base64, whether to enable bd_base64 support,

dnl Make sure that the comment is aligned:

dnl [--enable-bd_base64   Enable bd_base64 support])

The simplest “Hello, world”

- Insert “PHP_FE(bd_base64_echo, NULL)” to bd_base64.c
- Insert “PHP_FUNCTION(bd_base64_echo);” to php_bd_base64.h
- Coding

```
PHP_FUNCTION(bd_base64_echo)
{
    printf("[module] hello, world\n");
}
```

Baidu 百度

Make and setup

```
phpize
./configure --with-bd_base64
make
```

- bd_base64.so will be generate under ./modules

```
cp modules/bd_base64.so   PHP_MODULE_DIR
echo ‘extension= bd_base64.so’ >> php.ini
```

另类调用

- **dl**
- dl -- Loads a PHP extension at runtime
  Loads the PHP extension given by the parameter library.

```
if (!extension_loaded('sqlite')){
    $prefix = (PHP_SHLIB_SUFFIX === 'dll') ? 'php_' : '';
    dl($prefix . 'sqlite.' . PHP_SHLIB_SUFFIX);
}
```

Let’s run it! ^_^

- Touch test php file

```
<?php
bd_base64_echo();
?>
```

- Exit the editor and execute ‘php test.php’

We need input!

```
{
int len;
   if (zend_parse_parameters(ZEND_NUM_ARGS() TSRMLS_CC, "l", &len) ==
FAILURE)
      return;
}
```

```
{
   char *filename;
   if (zend_parse_parameters(ZEND_NUM_ARGS()
TSRMLS_CC, "s", &filename, &readlen) == FAILURE)
      return;
}
```

Baidu 百度

How to return value?

```
{
   int i = 999;
   RETURN_LONG(i);
}
```

```
{
   RETURN_STRING("Hello World", 1);
}
```

```
{
   char *str;
   str = estrdup("Hello World");
   RETURN_STRING(str, 0);
}
```

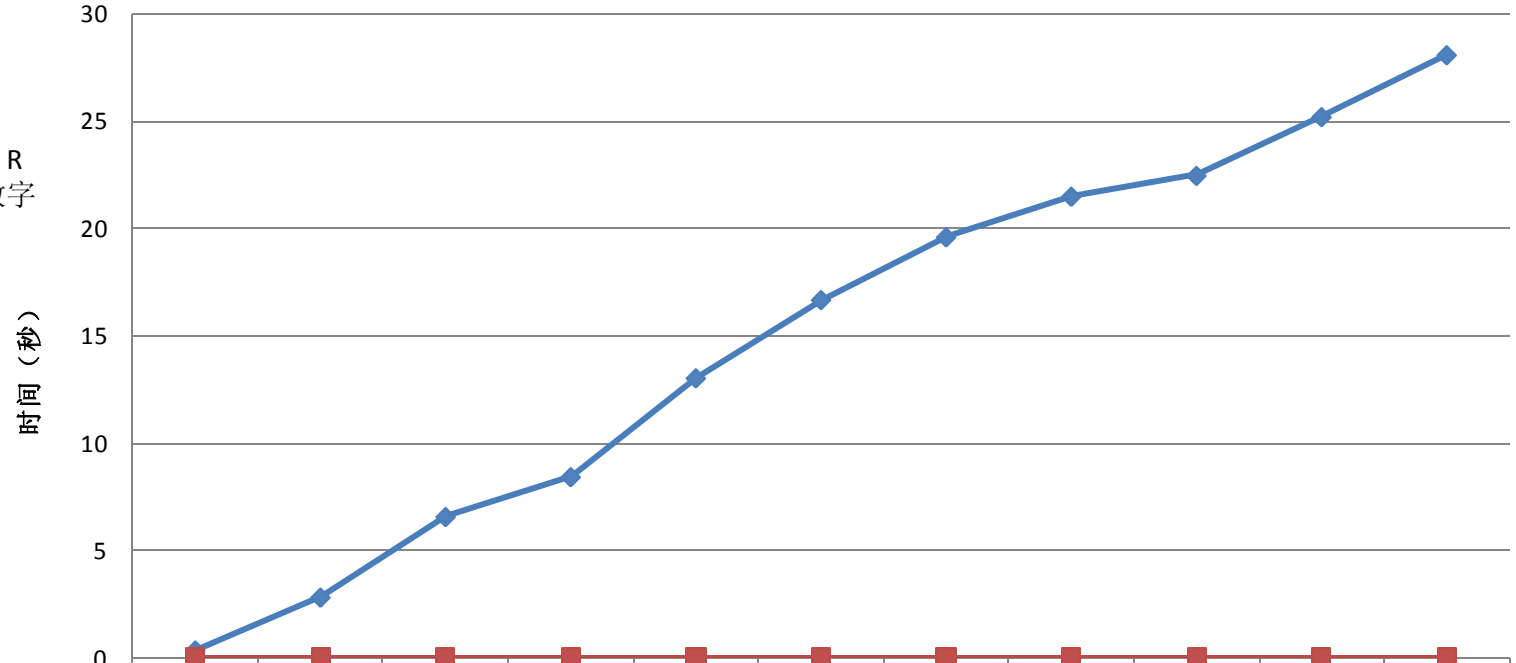
Danger! Be care!

- Thread safety（线程安全）
- Memory access control（内存访问控制）
- Memory leak（内存泄漏）
- Interface: transfer the data/parameter, return value（与zend的引擎及其他模块的接口）

Performance: PHP code VS C (1)

PHP代码 VS PHP Extension时间开销（单纯运算速度对比）

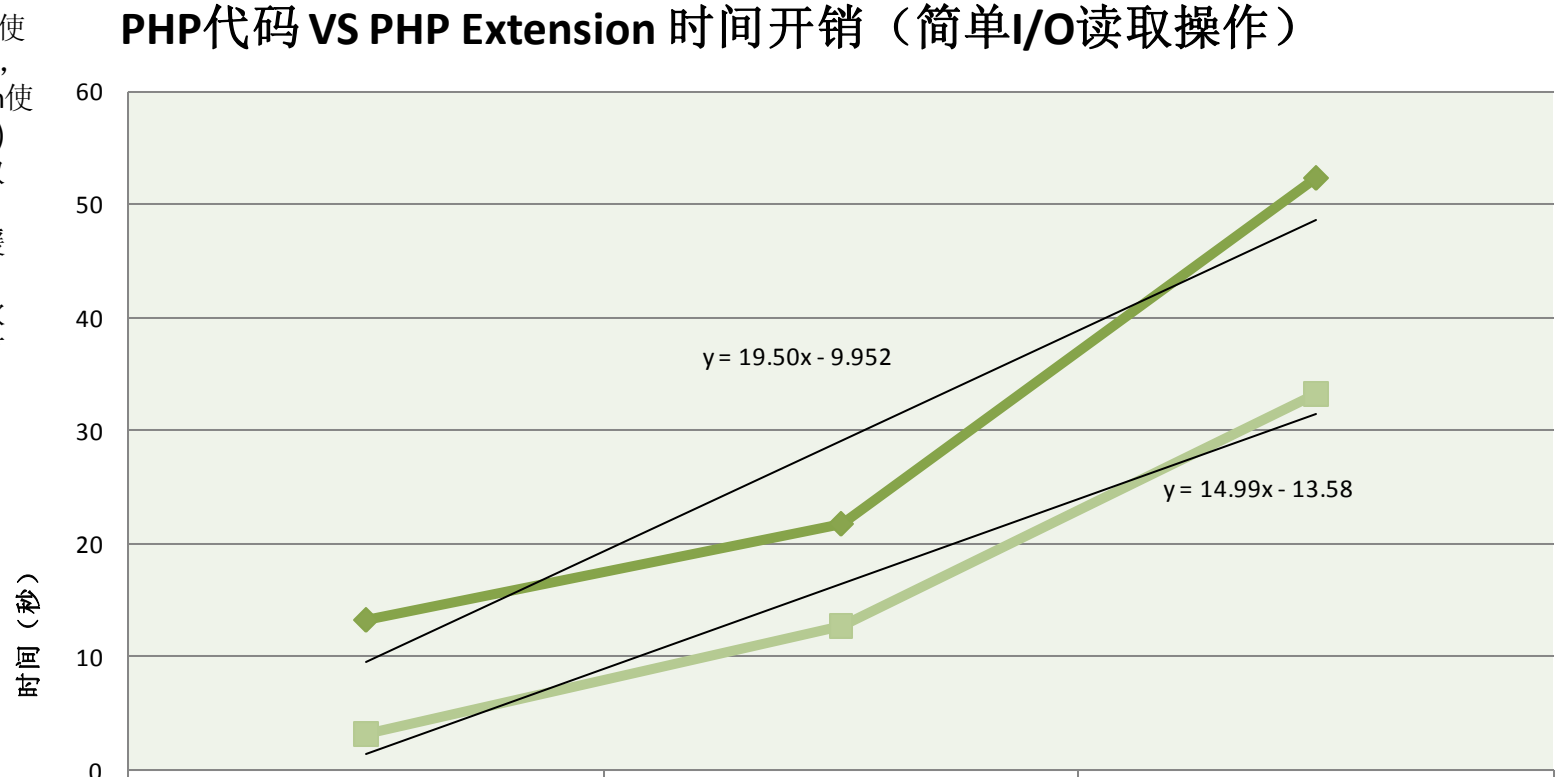
运算公式：
Area = PI * R * R
R = 0至某个数字



	10万次面积运算	100万次面积运算	200万次运算	300万次运算	400万次运算	500万次运算	600万次运算	700万次运算	800万次运算	900万次运算	1000万次运算
PHP代码计算速度	0.3441393	2.8107173	6.560929	8.4148706	13.018678	16.662212	19.5837	21.476891	22.443821	25.179521	28.062906
PHP Extension	0.0001253	0.001009	0.0017013	0.001557	0.003786	0.004157	0.0056796	0.00584	0.0051033	0.0057613	0.005066

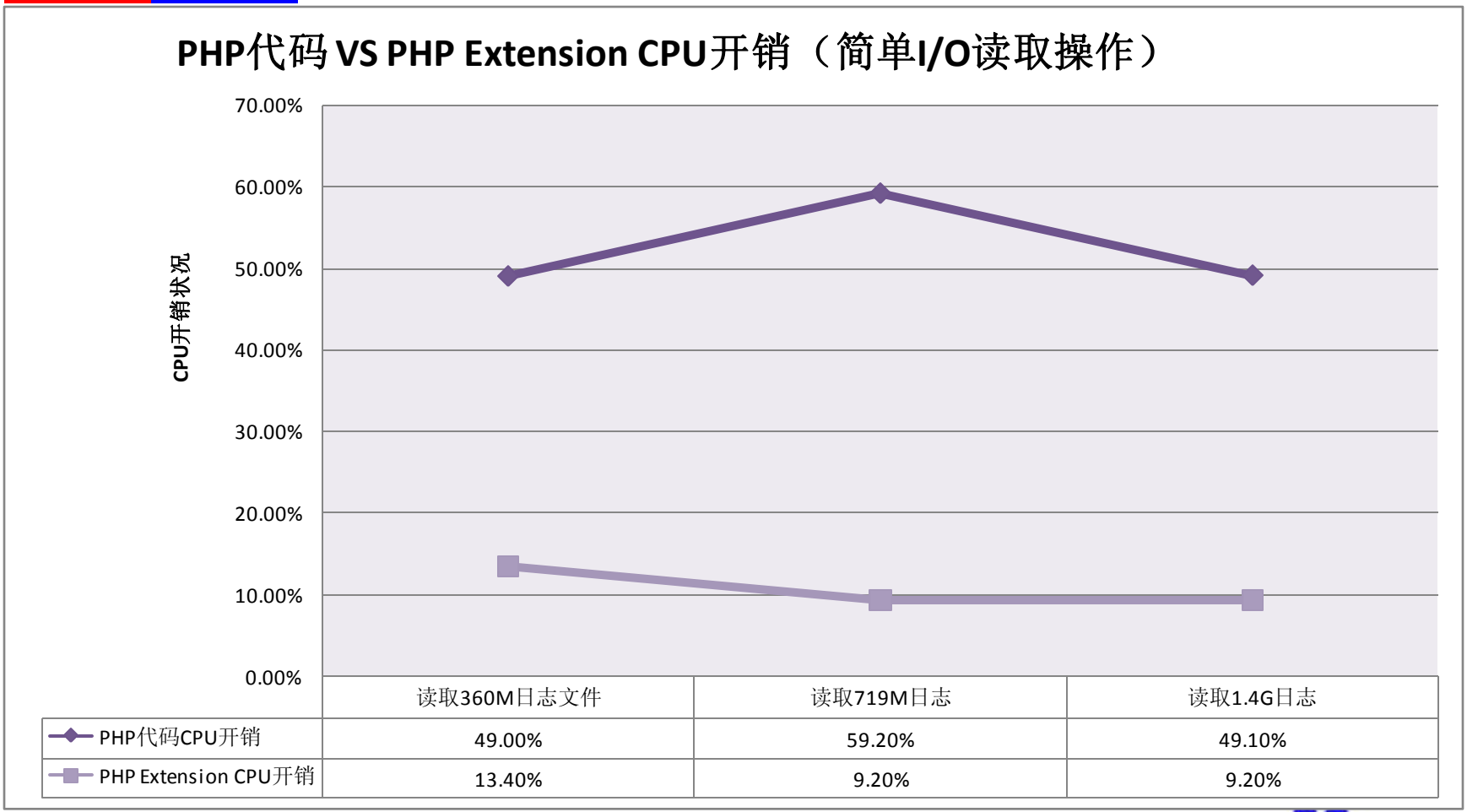
Performance (2)

PHP代码使用fgets()，extension使用read(2)每次读取1k数据。为消除缓存影响，取第一次读取为有效数据



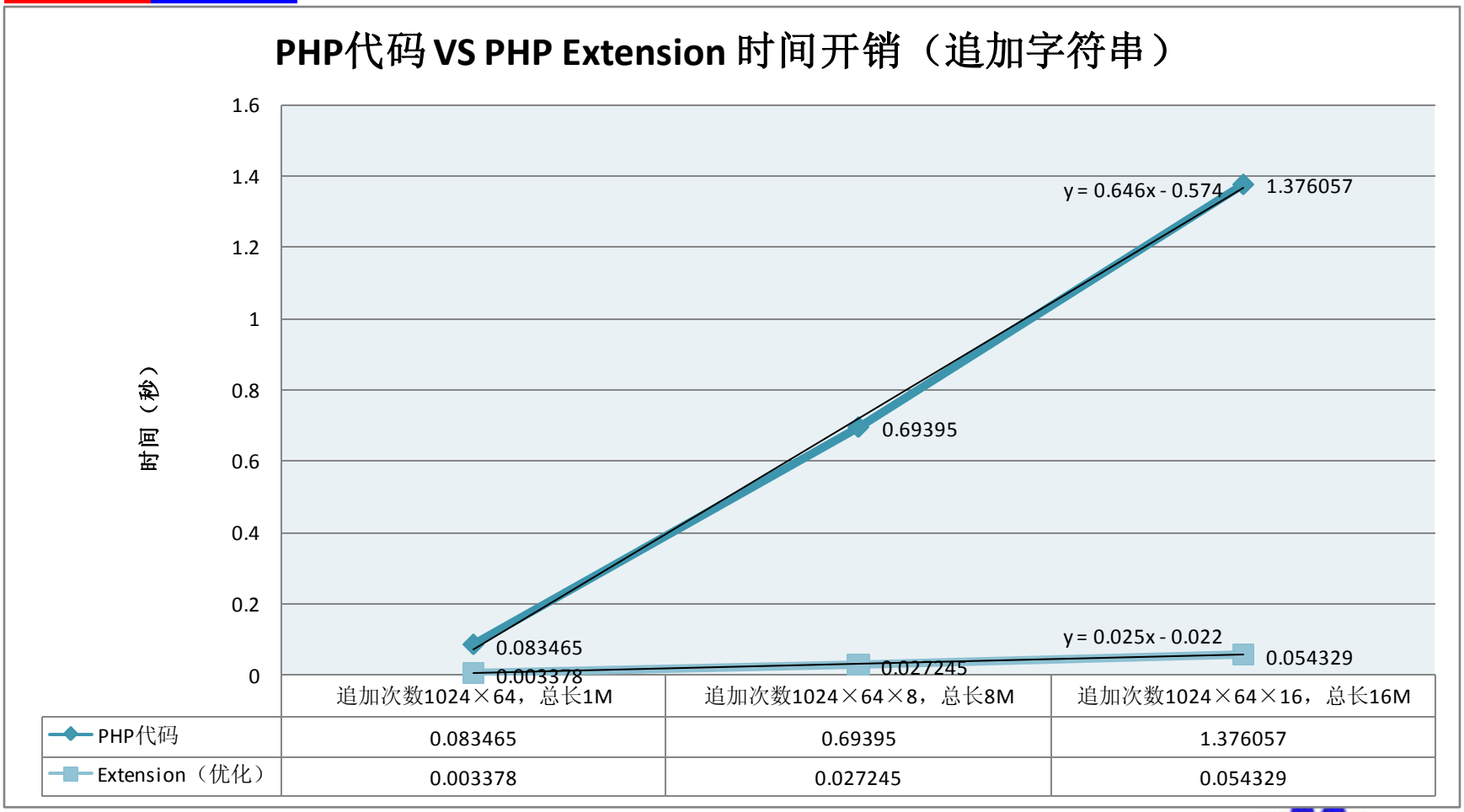
	读取360M日志文件	读取719M日志	读取1.4G日志
PHP代码	13.230902	21.715145	52.248417
PHP Extension	3.224038	12.78798	33.21332

Performance (3)



	读取360M日志文件	读取719M日志	读取1.4G日志
PHP代码CPU开销	49.00%	59.20%	49.10%
PHP Extension CPU开销	13.40%	9.20%	9.20%

Performance (4)



	追加次数1024×64，总长1M	追加次数1024×64×8，总长8M	追加次数1024×64×16，总长16M
PHP代码	0.083465	0.69395	1.376057
Extension（优化）	0.003378	0.027245	0.054329

Develop VS Run (1)

- 单纯的运算（CPU密集）

	开发效率	运行效率
PHP:	100	1
Extension	80	10000

Develop VS Run (2)

- 单纯的I/O操作（不使用mmap(2)）

	开发效率	运行效率
PHP	100	60
Extension	>50	100

Develop VS Run (3)

- 复杂逻辑

	开发效率	运行效率
PHP	100	>20
Extension	<10	100

Choice, it’s your turn now!

- 开发效率
- 实现难度
- 调试
- 支撑环境（各类支撑库）
- 运行效率