

我们都知道 memcache 是解决大并发量的很好一个工具，但是有一个问题值得注意，当突然有很多用户请求我们的某个页面，请求的参数相同，但对应的值却在 memcache 中没有，此时这些请求都将会去查询数据库操作，瞬间对数据库造成了很大的压力。

下面我做了一个小例子来模拟这种情况。

原页面代码：

```
<?php
$memcache = new Memcache();
$memcache -> connect("127.0.0.1",11211) or die("connect failed");

$key = $_GET['key'];
if($result = $memcache->get($key)) {
    $s = "get result form memcache: ".$key."/". $result. "\n";
    file_put_contents("/var/www/html/log1.txt",$s,FILE_APPEND);
    echo $s;
}else {
    // 模拟查询数据库

    sleep(5); // 休眠 5 秒

    $result = mt_rand(1,100);
    $memcache -> set($key,$result,0,3600);
    $s = "get result not from memcache: ".$key."/". $result. "\n";
    file_put_contents("/var/www/html/log2.txt",$s,FILE_APPEND);
    echo $s;
}
?>
```

此代码的意思是，我通过某个参数 key 来访问这个页面，如果此 key 在 memcache 中存在，则在 log1.txt 中写入一条数据，如果不存在则在 log2.txt 中写入一条数据，同时将此 key 得到的结果放入 memcache 中。

这里值得注意的是我在连接 memcache 时候加了一句 “ or die("connect failed"); ” , 即当连接报错时退出, 因为有可能当 memcache 的连接数超出了它本身限制的次数后, 导致连接不上, 从而会执行 else 中的语句 (而非 memcache 中无值)。

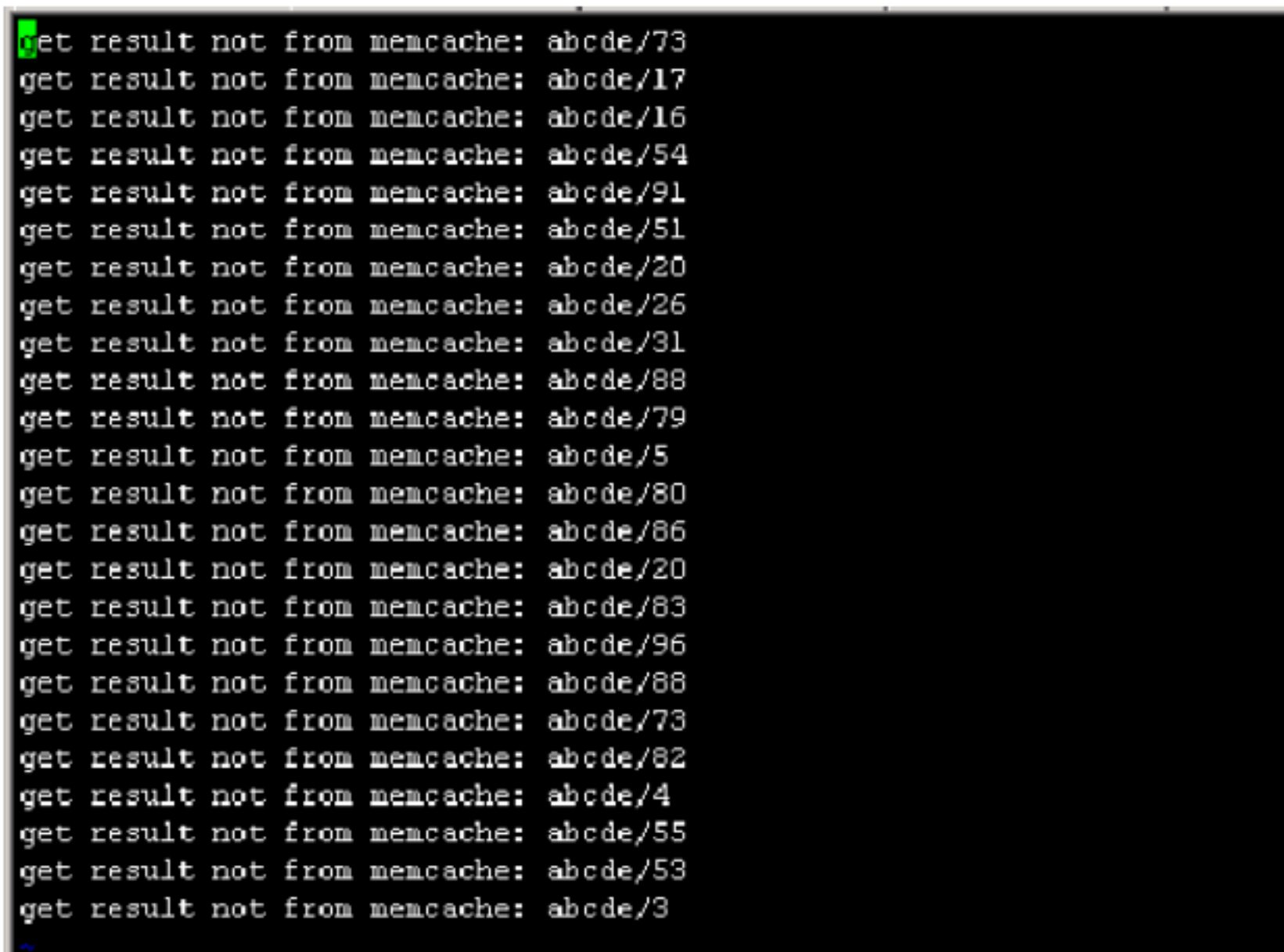
测试并发

1) 使用 ab 命令测试

```
# ab -c 200 -n 30000 http://192.168.1.78/complicateTest.php?key=abcde
```

以上命令表示 200 个并发请求 30000 次

可以看到在 log2 文件中有许多数据:



```
get result not from memcache: abcde/73
get result not from memcache: abcde/17
get result not from memcache: abcde/16
get result not from memcache: abcde/54
get result not from memcache: abcde/91
get result not from memcache: abcde/51
get result not from memcache: abcde/20
get result not from memcache: abcde/26
get result not from memcache: abcde/31
get result not from memcache: abcde/88
get result not from memcache: abcde/79
get result not from memcache: abcde/5
get result not from memcache: abcde/80
get result not from memcache: abcde/86
get result not from memcache: abcde/20
get result not from memcache: abcde/83
get result not from memcache: abcde/96
get result not from memcache: abcde/88
get result not from memcache: abcde/73
get result not from memcache: abcde/82
get result not from memcache: abcde/4
get result not from memcache: abcde/55
get result not from memcache: abcde/53
get result not from memcache: abcde/3
```

可以使用对数据库查询加锁来处理, 即在查询数据的时候, 先在缓存中存一个值 K, 其他客户端需要查询数据库的时候要先判断 K 是否存在, 若存在则休眠, 如下:

```
<?php
$memcache = new Memcache();

$memcache -> connect("127.0.0.1",11211) or die("connect failed");
```

```
$key = $_GET['key'];  
getResult($memcache,$key);  
  
function getResult($memcache,$key) {  
    if($result = $memcache->get($key)) {  
        $s = "get result form memcache: ".$key."/". $result."\n";  
        file_put_contents("/var/www/html/log1.txt",$s,FILE_APPEND);  
        echo $s;  
    }else {  
        $tmp_key = "_temp_". $key;  
        if($memcache -> add($tmp_key, "temp key",0,60)) {  
            sleep(5);  
            $result = mt_rand(1,100);  
            $memcache -> set($key,$result,0,3600);  
            $s = "get result not from memcache: ".$key."/". $result."\n";  
            file_put_contents("/var/www/html/log2.txt",$s,FILE_APPEND);  
            echo $s;  
            $memcache -> delete($tmp_key);  
        }else {  
            sleep(3);  
            file_put_contents("/var/www/html/log3.txt","sleep...\n",FILE_APPEND);  
            getResult($memcache, $key);  
        }  
    }  
}  
?  
?>
```

此时我们再可以观察 log2.txt 中的日志，发现只有一条的，即只有一条是从数据库中查询的数据。

但如果 sleep 的线程如果太多的话，也可能造成进程阻塞，

另一种方法是，将数据设置在 memcache 的时间为永久性的，在数据中设置一个过期时间，每次获取数据时判断是否过期，若过期则加锁重新查询数据库。加锁失败则返回旧数据，但是这样做的前提是要确保 memcache 有这个数据，若没有则还是会返回空。

```
<?php
$memcache = new Memcache();
$memcache -> connect("127.0.0.1",11211) or die("connect failed");

$key = $_GET['key'];
print_r(getResult($memcache,$key));

function getResult($memcache,$key) {
    $result = array();
    if($result = $memcache->get($key)) {
        if($result["expired"] > strtotime(date("Y-m-d H:i:s"))) {
            //memcache 中有数据，且未过期，返回原数据
            return $result;
        }else {
            return lockData($result,$memcache,$key);
        }
    }else {
        //memcache 中无数据
        return lockData($result,$memcache,$key);
    }
}
```

```

function lockData($result,$memcache,$key) {
    $tmp_key = "_temp_".$key;
    if($memcache -> add($tmp_key, "temp key",0,60)) {
        sleep(5);
        $result["data"] = mt_rand(1,100);
        $result["expired"] = strtotime(date("Y-m-d H:i:s")." +1 hour");
        $memcache -> set($key,$result,0,0);
        $memcache -> delete($tmp_key);
    }else {
        // 若 memcache 中没有数据则会返回空，若有值则是返回旧数据
    }
    return $result;
}

?>

```

使用 gearmand 来处理并发，gearmand 可以确保相同的请求只执行一次，gearmand 的介绍可以参考[这里](#)

1) gearmand 安装

```
#yum install libdrizzle libdrizzle-devel
```

```
#rpm -ivh http://dl.iuscommunity.org/pub/ius/stable/Redhat/6/x86\_64/epel-release-6-5.noarch.rpm
```

```
#yum install gearmand
```

2) 安装 php 的 gearmand 扩展

首先要先确定有 phpize，如果没有可以使用：

```
#yum install php-devel
```

```
#wget http://pecl.php.net/get/gearman-1.1.0.tgz
```

```
#tar zxf gearman-1.1.0.tgz
```

```
#yum install libgearman-devel.x86_64
```

```
#yum install re2c
```

```
#cd gearman-1.1.0
```

```
#/usr/bin/phpize
```

```
#!/configure
```

```
#make && make install
```

修改 php.ini 文件，加入

```
extension=gearman.so
```

3) 启动 gearman

```
#!/etc/init.d/gearman start
```

```
#service httpd restart
```

代码示例：

服务端 gearmanWorker.php：

```
<?php
```

```
    $worker= new GearmanWorker();
```

```
    $worker->addServer("127.0.0.1", 4730);
```

```
    $worker->addFunction("querydb","querydb");
```

```
    while ($worker->work());
```

```
function querydb($job) {
```

```
    $key = $job -> workload();
```

```
    sleep(5);
```

```
    $result = mt_rand(1,100);
```

```
    $s = "get result not form memcache: ".$key."/". $result."\n";
```

```
    file_put_contents("/var/www/html/log2.txt",$s,FILE_APPEND);
```

```
    return $result;
```

```
}
```

```
?>
```

客户端 complicateTest.php :

```
<?php
```

```
$memcache = new Memcache();
```

```
$memcache -> connect("127.0.0.1",11211) or die("connect failed");
```

```
$client= new GearmanClient();
```

```
$client->addServer("127.0.0.1", 4730);
```

```
$key = $_GET['key'];
```

```
echo getResult($memcache,$key,$client);
```

```
function getResult($memcache,$key,$client) {
```

```
    if($result = $memcache->get($key)) {
```

```
        // 从缓存中取
```

```
        $s = "get result form memcache: ".$key."/". $result. "\n";
```

```
        file_put_contents("/var/www/html/log1.txt",$s,FILE_APPEND);
```

```
    }else {
```

```
        //memcache 中无数据
```

```
        $result = $client -> doNormal("querydb",$key,$key);
```

```
        $memcache -> set($key, $result,0,3600);
```

```
    }
```

```
    return $result;
```

```
}
```

?>

首先启动服务端

```
#nohup /usr/bin/php gearmanWorker.php > log.txt 2>&1 &
```

使用 ab 命令测试：

```
ab -c 200 -n 10000 http://192.168.1.78/complicateTest3.php?key=sc842da
```

可以发现 log2.txt 中也只有一条记录，即只有一次从数据库中取，其他的都是从缓存中获取