

——高性能PHP应用开发之

第6讲 Memcached

目录

CONTENTS

1 / 安装Memcached

2 / 使用Memcached

3 / ThinkPHP中使用Memcached

目录

CONTENTS

1 / 安装Memcached

2 / 使用Memcached

3 / ThinkPHP中使用Memcached

Memcached简介

- Memcached是高性能的分布式内存缓存服务器；通过缓存数据库查询结果，减少数据库访问次数，以提高动态Web应用的速度、提高可扩展性。
- ✓ 工作原理：通过在内存里维护一个统一的巨大的hash表，用来存储各种格式的数据，包括图像、视频、文件以及数据库检索的结果等；将数据存储到内存中，然后从内存中读取，从而大大提高读取速度。
- **Memcached、memcache和memcached：**
 - ✓ Memcached：服务器端，即安装的Memcached服务（类似于MySQL服务）
 - ✓ memcache：使用PECL扩展库形式操作Memcached（类似于 php_mysqli ）
 - ✓ memcached：使用libmemcached形式操作Memcached（类似于 PDO）

安装Memcached

- 下载地址: <http://www.runoob.com/memcached/window-install-memcached.html>
- ✓ Linux环境: <http://memcached.org/downloads>
- ✓ 建议安装 1.4.4 版本, 会自动把 memcached 作为Windows的一个服务处理
- 安装Memcached: 以1.4.4版本为例
 - ✓ 管理员身份使用CMD, 执行 目录\mememcached.exe -d install (或 uninstall)
 - ✓ 管理员身份使用CMD, 执行 目录\mcached.exe -d start (或 stop)
 - ✓ 安装完成后, 在 Windows 服务管理窗口中查看是否安装成功 (可能需要重启系统)
 - ✓ 默认安装的端口号是 11211

目录

CONTENTS

1 / 安装Memcached

2 / 使用Memcached

3 / Laravel中使用Memcached

memcache 或 memcached?

- 在PHP中，可以使用memcache 或 memcached 连接Memcached
 - ✓ memcache：只具有基本功能，通过PECL扩展的形式进行连接
 - ✓ memcached：具有更多的功能，通过 libmemcached 方式进行连接
 - ✓ 注意：memcached 可以认为是**更好**的操作Memcached的方式，**但是**，memcached目前只能在Linux上使用，**没有官方Windows版本扩展**。
- 安装 php_memcache.dll
 - ✓ 下载地址：<http://pecl.php.net/package/memcache/3.0.8/windows>
 - 使用phpinfo查看PHP的版本、编译架构、线程安全等信息
 - ✓ 在php.ini配置文件中，加载 php_memcache.dll
 - ✓ 重新启动服务器，通过phpinfo查看

Memcached可视化管理工具

- memadmin: 图形化形式管理Memcached数据
 - ✓ 下载地址: <https://github.com/junstor/memadmin>
 - ✓ 安装和使用方式: 直接解压缩到Web服务器根目录, 通过浏览器访问即可 (类似于 phpMyAdmin) ; 初始用户名和密码均为 admin

Memcached可视化管理工具

- 连接信息
 - 连接参数
- 服务器信息
 - 统计信息
 - 设置信息
 - 区块统计
 - 数据项统计
 - 对象数量统计
- 性能监控
 - 统计监控
 - 数据监控
 - 命中监控
- 数据存取
 - 读取数据
 - 写入数据
 - 计数命令
 - 全部失效
- 扩展功能
 - 数据遍历

数据遍历

» 默认连接 127.0.0.1 : 11211 记录总数: 1 由于memcached源码对cachedump命令的限制, 最多遍历2M的k

选择区块: SLAB : 3 区块内共有记录: 1 遍历前 1 条记录 字符集: UTF-8

遍历结果: 数组/对象 序列化后显示, JSON字符串反序列化后以数组形式显示

KEY : *key*

字符集: UTF-8

0:8:"stdClass":2:{s:8:"str_attr";s:4:"test";s:8:"int_attr";i:123;}

Flags: 1 类型: object 大小: 66 byte 失效时间: 2017-03-20 08:40:33 [反序列化](#) [刷新](#) [删除](#)

获取总数 : 1

PHP中连接Memcached

■ PHP连接Memcached基本方法

- ✓ 创建 Memcache连接对象: `$memcache = new Memcache();`
- ✓ 建立连接: `$memcache->connect(连接地址, 端口号);`
- ✓ 写入数据: `$memcache->add()`、`$memcache->replace()`、`$memcache->set()`
- ✓ 读取数据: `$memcache->get()`
- ✓ 增加/降低数据值: `$memcache->increment()`、`$memcache->decrement()`
- ✓ 删除/清空元素: `$memcache->delete()`、`$memcache->flush()`
- ✓ 关闭连接: `$memcache->close();`

Memcached实现数据库缓存

■ 使用Memcached

- ① 建立Memcached连接
- ② 首先从Memcached缓存中读取数据，若读取成功，直接执行后续操作
- ③ 若②失败，则从数据库中读取数据，并把数据结果写入到Memcached中，执行后续操作
- ④ 关闭Memcached连接

```
// 1. 建立Memcached连接
$memcache = new Memcache();
$memcache->connect('127.0.0.1', 11211);

// 2. 从缓存中或数据库读取数据
$key = md5($sql);
$data = $memcache->get($key);
if(!$data){
    // 执行数据库查询
    $data = $db->query($sql);
    // 把数据查询结果存储到Memcached中
    $memcache->add($key, $data);
}

// 3. 执行后续操作

// 4. 关闭Memcached连接
$memcache->close();
```

Session数据写入Memcached

- “硬配置”：在php.ini中直接修改（需要重启服务器）
 - ✓ `session.save_handler = memcache`
 - ✓ `session.save_path = "tcp://127.0.0.1:11211"`
- “软配置”：`ini_set()`方法，只对当前脚本有效
 - ✓ `ini_set("session.save_handler", "memcache");`
 - ✓ `ini_set("session.save_path", "tcp://127.0.0.1:11211");`
- `session_set_save_handler()`方法：只针对当前脚本有效
 - ✓ 操作方式更灵活，不仅仅局限于 Memcache

Session数据写入Memcached

■ session_set_save_handler : 设置用户自定义会话存储机制

✓ 函数原型: bool session_set_save_handler (SessionHandlerInterface \$sessionhandler)

✓ SessionHandlerInterface接口类:

```
SessionHandlerInterface {  
  
    /* 方法 */  
    abstract public bool close ( void )  
    abstract public bool destroy ( string $session_id )  
    abstract public bool gc ( int $maxlifetime )  
    abstract public bool open ( string $save_path , string $name )  
    abstract public string read ( string $session_id )  
    abstract public bool write ( string $session_id , string $session_data )  
}
```

Session数据写入Memcached

■ session_set_save_handler : 设置用户自定义会话存储机制

```
class MemSession implements SessionHandlerInterface
{
    protected $_memcache;

    public function close()
    {
        return $this->_memcache->close();
    }

    public function destroy($session_id)
    {
        return $this->_memcache->flush();
    }

    public function open($save_path , $name)
    {
        $this->_memcache = new Memcache();
        return $this->_memcache->connect('localhost', 11211);
    }

    public function read($session_id)
    {
        return $this->_memcache->get($session_id);
    }

    public function write($session_id , $session_data)
    {
        return $this->_memcache->set($session_id, $session_data);
    }
}
```

目录

CONTENTS

1 / 安装Memcached

2 / 使用Memcached

3 / **ThinkPHP中使用Memcached**

ThinkPHP中使用Memcached

■ TP5中原生支持Memcached缓存服务，使用方法如下：

✓ 在 cache.php 配置文件中指明要使用的缓存类型

```
return [  
    // 驱动方式  
    'type' => 'memcache',  
    // 缓存保存目录  
    'path' => '',  
    // 缓存前缀  
    'prefix' => '',  
    // 缓存有效期 0表示永久缓存  
    'expire' => 0,  
];
```


ThinkPHP中使用Memcached

- 使用缓存方法完成基本操作
 - ✓ `Cache::set()` 设置缓存数据
 - ✓ `Cache::get()` 获取缓存数据
 - ✓ `Cache::inc()`、`Cache::dec()` 自增/自减缓存数据
 - ✓ `Cache::rm()` 删除缓存数据
 - ✓ `Cache::clear()` 清空缓存数据
 - ✓ 使用助手函数 `cache()` 操作缓存数据

感谢聆听！

THANK YOU FOR YOUR ATTENTION