# memcache

兄弟连IT教育

转载请注明出处

# 第一章

## memcache定义

memcahe是一项开源的缓存服务器,以键值对的方式存储数据,数据可读性非常好,每秒可以处理上万次数据

Name=>xy

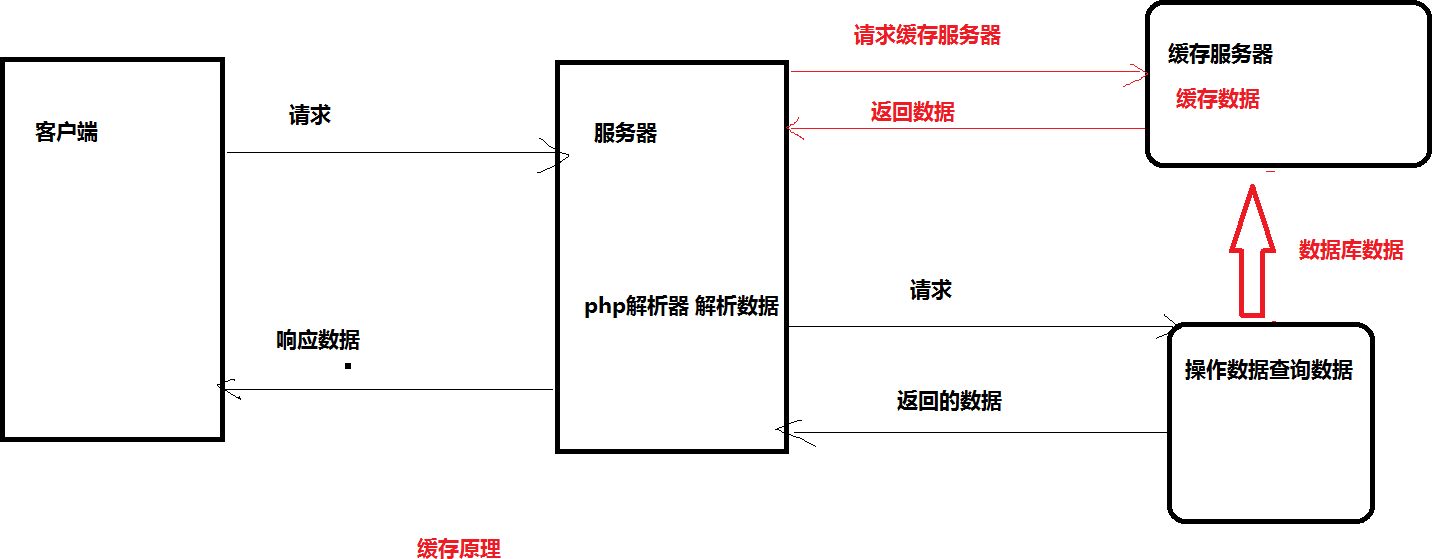
Name=>love

## Memcache好处

* + 1. 减少高并发,减轻数据库压力 ,高并发:每秒每秒成千上万次对数据库访问
    2. 提高海量数据的访问效率

Memcache是以内存的方式存储数据,内存读取效率远比磁盘读取效率高

* + 1. 原理图



## memcache 存储方式\*\*

key-value键值对

## windows下memcache安装及扩展开启

* + 1. windows 安装
       1. 开启电脑telnet客户端

控制面板-》程序-》打开或者关闭windows-》telnet客户端勾选中

* + - 1. memcached.exe应用程序放在wamp里面bin目录里-》启动cmd(必须以管理员身份运行) 去安装

memcache服务

* + - 1. memcache操作相关命令

Cd 切到memcache.exe所在的目录

* + - * 1. memcached.exe -d install 安 装 memcache

查看自己是否安装成功 打开服务 services.msc

有的人可能报错误 msvcp71.dll 丢失 别的文件丢失

360 人工服务 丢失文件打上 修复

* + - * 1. memcached.exe –d start -m 64 -u root -l 127.0.0.1 -p 11211 带参数启动

-p 监听的端口 11211 端口号

-l 连接的IP地址, 默认是本机

-d start 启动memcached服务

-u 以的身份运行 (仅在以root运行的时候有效)

-m 最大内存使用，单位MB。默认64MB

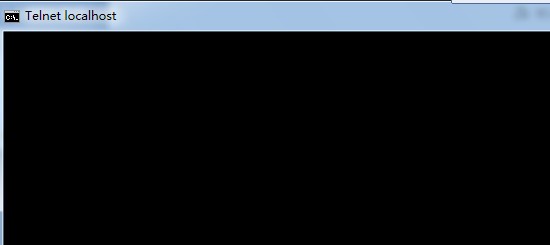
* + - * 1. memcached.exe -d start 简写开启memcache (推荐)

在看一眼服务确定已经成功打开 ~

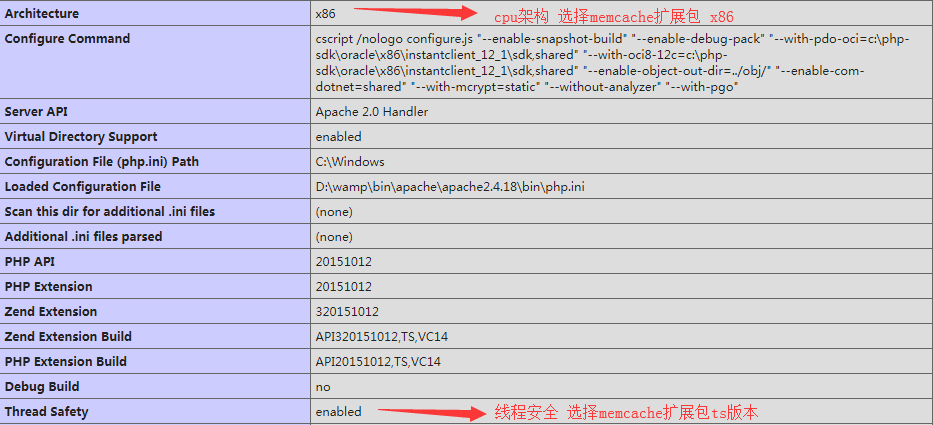
* + - * 1. memcached.exe -d stop关闭memcache服务

想要卸载 ,必须先停止服务才能卸载

* + - * 1. memcached.exe -d uninstall 卸载memcache
      1. 连接户端
         1. memcache客memcache服务必须开启
         2. 使用telnet localhost 11211



* + 1. windows下开启memcache扩展
       1. memcache服务必须开启
       2. php下开启与memcache相关的扩展功能
          1. 将php\_memcache.dll 扩展文件 放在wamp->bin->php->ext 目录下扩展文件选择



* + - * 1. 在配置文件里php.ini添加配置选项并开启

extension=php\_memcache.dll

* + - * 1. 重启Apache
        2. 地址栏输入:<http://localhost/?phpinfo=1>查看,出现memcache表示开启扩展成功

## linux下memcache安装及扩展开启

* + 1. linux下memcache安装

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

1.将下载好的安装包上传到linux服务器下2.首先安装依赖包libevent

yum –y install libevent\* #centos6.8 错误?

换光盘安装 CentOS-6.8-i386-bin-DVD2.iso 先把CentOS-6.8-i386-bin-DVD1.iso进行卸载umount /mnt/cdrom

再重新进行挂载

mount /dev/sr0 /mnt/cdrom

3.安装libevent-devel\*,不然./configure过不去yum -y install libevent-devel\*

tar zxvf memcached-1.4.36.tar.gz 解压memcached

cd memcached-1.4.36

进入 memcached目录

./configure --prefix=/usr/local/memcache 配 置

make && make install

编译与安装

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

useradd memcache

添加memcache用户

4.因为系统不能用root运行memcache软件

/usr/local/memcache/bin/memcached –umemcache & 后台运行

netstat –tunpl|grep :11211

查看端口

telnet 192.168.185.252 11211

注意:

win+r打开命令行窗口 连接测试

如果输入telnet命令的时候 出现telnet不是内部命令

windows系统->控制面板->程序和功能->打开和关闭Windows功能->勾选 telnet客户端

stats

memcache命令：查看当前状态

1. 写入自启动：

vi /etc/rc.local

/usr/local/memcache/bin/memcached –umemcache &

1. 如何杀掉后台进程pkill memcached
2. 连接memcache客户端

telnet 192.168.22.34 11211

* + 1. linux下memcache 扩展开启

1. 将pecl-memcache-php7.zip文件传到服务器上 /root
2. 解压文件后进入此目录

unzip pecl-memcache-php7.zip cd pecl-memcache-php7

1. 由于目录中没有configure配置环境文件

执行 /usr/local/php/bin/phpize 来准备一个编译环境

1. 配置： 加配置

./configure --with-php-config=/usr/local/php/bin/php-config

1. make && make install 编译和安装
2. 安装成功后就会产生一个memcache.so文件

/usr/local/php/lib/php/extensions/no-debug-non-zts-20151012/

1. 修改php配置文件

vim /usr/local/php/etc/php.ini

添加以下信息

722 行 extension\_dir='/usr/local/php/lib/php/extensions/no-debug-non-zts-20151012/';

extension="memcache.so";

1. 重启apache服务即可(查看phpinfo())

1.首先下载php7.0支持的扩展包 pecl-memcache-php7.zip

地址:https://github.com/websupport-sk/pecl-memcache/archive/php7.zip

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

## memcache命令

* + 1. set

set的命令在memcache中的使用频率极高。set命令不但可以简单添加，如果set的key已经存在，该命令可以 更新该key所对应的原来的数据，也就是实现更新的作用。

格式:Set 键名 1 过期时间 值长度, 1代表每个键名的标识例如:set name 1 10 4

set user 1 0 6 回车

内容

* + 1. get

获取缓存数据格式:get 键名

* + 1. delete

删除数据

格式:delete 键名

* + 1. add

只有数据不存在时进行添加,

格式:add 键名 1 过期时间(单位为秒) 值长度，1代表的是标识号例如:add name 1 20 5，永不过期,过期时间设置为0

* + 1. replace set

只有数据存在时进行替换

例如:replace 已经存在的键名 1 过期时间 新值的长度

* + 1. prepend

在现有的缓存数据前添加缓存数据

例如:prepend 缓存数据键名 1 过期时间 前置值长度

* + 1. append

在现有的缓存数据后添加缓存数据

例如:append 缓存数据键名 1 缓存时间 追加值长度

* + 1. incr 与 decr

自增或自减 自 -> 自己

例如:Incr 键名 加的值

Decr 键名 减的值

* + 1. ﬂush\_all

清除所有缓存数据

* + 1. quit 退出

手册查找memcache

Connect函数

## memcache缓存实例一\*\*

1. <?php
2. //缓存实例
3. //实例化Memcache
4. $memcache=new Memcache();
5. //连接memcache

6 $memcache->connect("192.168.22.34",11211);

1. //键名
2. $key="data";
3. //获取键名为data的缓存数据
4. $res=$memcache->get($key);
5. //判断
6. if(empty($res)){
7. //获取数据库的数据
8. $pdo=new PDO("mysql:host=localhost;dbname=php42","root","123456");
9. $pdo->exec("set names utf8");
10. //执行sql
11. $list=$pdo->query("select \* from stu");
12. //获取结果集
13. $res1=$list->fetchAll(PDO::FETCH\_ASSOC);
14. echo "这是获取数据库的数据";
15. //把数据库的数据存储在memcache里
16. $memcache->add($key,$res1);
17. }else{
18. //获取缓存数据
19. $data1=$memcache->get($key);
20. echo "<pre>";
21. var\_dump($data1); 28 }

29 ?>

**第二章**

* 1. **memcache存储session实战开发\*\***

看用户登录实例



* + 1. 目的

提高session的访问效率

* + 1. 使用到函数

session\_set\_save\_handler 用户自定义session存储函数

* + 1. 演示代码

1

2

3

4

5

6

7

8

<?php

//把用户登录的session信息存储在memcache里class Memsession{

//成员属性 memcache实例对象

public static $handler; public static $pre="LAMP";

//初始化方法

public static function init($mem){

9

10 }

self::$handler=$mem;

* + - 1. public static function start($mem){
      2. self::init($mem);
      3. //session 自定义存储 session\_set\_save\_handler open 打开session close 关闭session read 读取session write 写入session destroy 删除session gc 无效session文件回收
      4. session\_set\_save\_handler(
      5. //把open close read write destroy gc 函数引入到类里 CLASS 当前类
      6. array( CLASS ,"open"),
      7. array( CLASS ,"close"),
      8. array( CLASS ,"read"),
      9. array( CLASS ,"write"),
      10. array( CLASS ,"destroy"),
      11. array( CLASS ,"gc") 22

23 );

24

25 }

26

1. //open 打开session 路径 $sess\_path session存储路径 $sess\_name session名字
2. public static function open($sess\_path,$sess\_name){
3. return true; 30 }

31

1. //close 关闭session
2. public static function close(){
3. return true; 35 }

36

1. //read 读取session $id 读取session的序列号
2. public static function read($id){
3. echo self::getkey($id)."<br>";
4. $data=self::$handler->get(self::getkey($id));
5. return $data; 42 }

43

1. //write 写入session 产生session数据 自动获取 $data 要存储的session数据 $id session序列号 随 机 字 符 LAMPsffwerwrwersdfrft =>key LAMPsffwerwrwersdfrfterwer=>key
2. //MEMCACHE\_COMPRESSED 唯一标记 压缩数据
3. public static function write($id,$data){
4. self::$handler->add(self::getkey($id),$data,MEMCACHE\_COMPRESSED,30000);
5. return true; 49 }

50

1. //destroy 删除session $id 删除session序列号
2. public static function destroy($id){
3. self::$handler->delete(self::getkey($id));
4. return true; 55 }

56

1. //gc 回收无效的session $maxlifetime session最大生存时间 1440(24分钟)
2. public static function gc($maxlifetime){
3. return true;

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

}

//处理键名 加私有前缀

public static function getkey($id){ return self::$pre.$id;

}

}

//实例化Memcache

$mem=new Memcache();

//连接memcache

$mem->connect("192.168.22.34",11211);

Memsession::start($mem);

?>