

enjoyeclipse

Asp.net mvc 网站之速度优化 -- Memcache

前一章说了一下使用页面缓存的问题，这次说一下数据缓存,我们使用的是Memcache作为数据缓存。


下面弱弱地引用一下百度百科关于Memcache的定义：

Memcache是一个高性能的分布式的内存对象缓存系统，通过在内存里维护一个统一的巨大的hash表，它能够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。Memcache是danga的一个项目，最早是LiveJournal 服务的，最初为了加速 LiveJournal 访问速度而开发的，后来被很多大型的网站采用。

1.Memcache服务器端的安装与启动


1. 下载memcache,我们使用的版本是1.2.5。
2. 使用命令行方式安装memcached-1.2.5.exe:

memcached-1.2.5.exe -d install
3. 启动memcached: *memcached-1.2.5.exe -d start*
4. 确认Windows 服务中memcached服务已经安装成功：


5. 需要注意的是memcached服务是使用的11211端口，如果开启了防火墙，需要将11211添加到例外中。

2.网站中配置BeITMemcached

1. 下载BeITMemcached_source_2010_08_04.zip, 并编译出BeITMemcached的DLL文件添加到项目的解决方案中:


2. 配置Web.Config

<!-- 第三方的组件配置-->

<configSections>

```

<section
                                name="beitmemcached"
type="System.Configuration.NameValueSectionHandler"
/>
```

导航

[博客园](#)
[首页](#)
[联系](#)
[订阅](#)
[管理](#)

≤	2011年6月						≥
日	一	二	三	四	五	六	
29	30	31	1	2	3	4	
5	6	7	8	9	10	11	
12	13	14	15	16	17	18	
19	20	21	22	23	24	25	
26	27	28	29	30	1	2	
3	4	5	6	7	8	9	

公告

昵称: [enjoyeclipse](#)

园龄: [2年](#)

粉丝: [2](#)

关注: [0](#)

统计

随笔 - 7 文章 - 6 评论 - 20

搜索

常用链接

[我的随笔](#)

[我的空间](#)

[我的短信](#)

[我的评论](#)

[更多链接](#)

我的标签

[ASP.NET MVC](#)(2)

[速度优化](#)(1)

[缓存](#)(1)

[memcached](#)(1)

[数据缓存](#)(1)

随笔档案

[2011年6月](#) (2)

[2009年5月](#) (5)

```
</configSections>

<beitmemcached>

<add key="MemcachedConfig" value="localhost" />

</beitmemcached>
```

我们目前的访问量还不是很大, 因此将Memcached与网站部署在一台机器上, 使用的是localhost. 如果网站和Memcache部署在不同机器, 这里的value 就应该是memcached所在机器的IP, 请再次确认11211端口可以访问。

3. Memcache在业务代码中的设计

前面把配置都弄完了, 现在需要说一下Memcache在业务代码中的设计了。我们使用的是Controller->Service->Model的模式, 切入点就在Service层。

首先构造一个接口ICachePolicy:

```
1 public interface ICachePolicy    {
2
3 void Add<T> (string key, T value);
4
5 void Add<T> (string key, T value, DateTime dt);
6
7 T Get<T> (string key);
8
9 void Add (string key, object value);
10
11 void Add (string key, object value, DateTime dt);
12
13 object Get (string key);
14
15 void Delete (string key);
16
17 }
```

最新评论

[1. Re: Asp.net mvc 网站之速度优化 -- Memcache](#)

不知道这个memcache与asp.net mvc 有什么必然的联系, 以使你要使用这样一个标题

--jomoleo

[2. Re: Asp.net mvc 网站之速度优化 -- Memcache](#)

原来如此, 非常感谢楼主的耐心回复!

--meicai

[3. Re: Asp.net mvc 网站之速度优化 -- Memcache](#)

鄙人写了一个缓存组件, 主要是集成Enterprise Library 5, 支持设计器配置, 集成MemCache, 下一步准备集成WindowsServerAppFabric. Memcache还不错.

--DYStudio.Net

[4. Re: Asp.net mvc 网站之速度优化 -- Memcache](#)

这个得顶, 想学 memcache 很久了, 可是都不得章法, 入门不了

--indexRoad

[5. Re: Asp.net mvc 网站之速度优化 -- Memcache](#)

支持下

--john23.net

阅读排行榜

[1. Asp.net mvc 网站之速度优化 -- Memcache\(1193\)](#)

[2. Asp.net mvc 网站之速度优化 -- 页面缓存\(677\)](#)

[3. 学习SIP的一些方法\(249\)](#)

[4. SIP入门\(190\)](#)

[5. Vista Home下解决ASP.NET连接服务器不正常的问题\(120\)](#)

评论排行榜

[1. Asp.net mvc 网站之速度优化 -- Memcache\(15\)](#)

[2. Asp.net mvc 网站之速度优化 -- 页面缓存\(4\)](#)

[3. SIP入门\(1\)](#)

[4. Vista Home下解决ASP.NET连接服务器不正常的问题\(0\)](#)

[5. What is SIP?\(0\)](#)

Google广告

接着实现MemcachedCachePolicy实现ICachePolicy:

```
1 public class MemcachedCachePolicy : ICachePolicy
```

```
2     {
3         private static readonly ILog Logger = LogManager.GetCurrentClassLogger();
4
5         private readonly MemcachedClient _cache;
6
7         public MemcachedCachePolicy()
8         {
9             _cache = MemcachedClient.GetInstance("MemcachedConfig");
10
11             _cache.MaxPoolSize = 10000;
12         }
13
14         public void Add<T>(string key, T value)
15         {
16             if (_cache.Set(key, value))
17             {
18                 Logger.Debug("Set _cache for key succeeded, key[" + key + "]);
19             }
20             else
21             {
22                 Logger.Debug("Set _cache for key failed");
23             }
24         }
25
26         public void Add<T>(string key, T value, DateTime dt)
27         {
28             _cache.Set(key, value, dt);
29         }
30     }
```



Powered by:

[博客园](#)

Copyright © enjoyeclipse

```

30
31     public T Get<T>(string key)
32     {
33         try
34         {
35             return (T)_cache.Get(key
36 );
37         }
38         catch (Exception e)
39         {
40             Logger.Debug("Get _cache
41 for key failed, key[" + key + "]", e);
42             _cache.Delete(key);
43             return default(T);
44         }
45     }
46
47     public void Add(string key, object value)
48     {
49         _cache.Set(key, value);
50     }
51
52     public void Add(string key, object value, DateTime dt)
53     {
54         _cache.Set(key, value, dt);
55     }
56
57     public object Get(string key)
58     {
59         return _cache.Get(key);
60     }
61
62     public void Delete(string key)
63     {
64         _cache.Delete(key);
65     }
66 }

```

最后构造一个BaseCacheService，让BussinessService都继承BaseCacheService，比如什么ProductService，ShopService等等。对外可见的是BaseCacheService，

而MemcachePolicy对于其他代码来说就是透明的了。

```

1 public abstract class BaseCacheService
2     {
3         protected readonly ICachePolicy
_cachePolicy;
4
5         protected BaseCacheService()
6         {
7
8         }
9
10        protected BaseCacheService(ICachePolicy cachePolicy)
11        {
12            _cachePolicy = cachePolicy;
13        }
14
15        protected T GetCache<T>(string key) where T: class
16        {
17            return _cachePolicy.Get<T>(key);
18        }
19
20        protected void Add<T>(string key, T items)
21        {
22            _cachePolicy.Add(key, items);
23        }
24
25        protected T GetOrAdd<T>(string key, T loadedItems) where T : class
26        {
27            var items = _cachePolicy.Get<T>(key);
28
29            if (items == null)
30            {
31                _cachePolicy.Add(key, loadedItems);
32            }
33            return loadedItems;
34        }

```

```

35         return items;
36     }
37
38     protected T GetOrAdd<T>(string key, Func<T> howToGet) where T : class
39     {
40         var items = _cachePolicy.Get
<T>(key);
41
42
43         if (items == null)
44         {
45             var loadedItems = howToG
et();
46             _cachePolicy.Add(key, lo
adedItems);
47             return loadedItems;
48         }
49
50         var type = items.GetType();
51         if (type == typeof(int) && i
tems.Equals(0))
52         {
53             var loadedItems = howToG
et();
54             _cachePolicy.Add(key, lo
adedItems);
55             return loadedItems;
56         }
57
58         return items;
59     }
60
61     protected T GetOrAdd<T>(string key, Func<T> howToGet, DateTime dt) where T
: class
62     {
63         var items = _cachePolicy.Get
<T>(key);
64         if (items == null)
65         {
66             var loadedItems = howToG
et();
67             _cachePolicy.Add(key, lo
adedItems, dt);

```

```
68             return loadedItems;
69         }
70
71         var type = items.GetType();
72         if (type == typeof(int) && items.Equals(0))
73         {
74             var loadedItems = howToGet();
75             _cachePolicy.Add(key, loadedItems, dt);
76             return loadedItems;
77         }
78         return items;
79     }
80 }
```


posted on 2011-06-16 15:59 [enjoyeclipse](#) 阅读(1196) [评论\(15\)](#) [编辑](#) [收藏](#)

评论

#1楼 2011-06-16 16:49 軒轅劍 

支持一下下

具体的Service怎么使用，还有怎么刷新缓存
博主能在说细点吗 谢谢 [回复](#) [引用](#) [查看](#)

#2楼 2011-06-16 16:55 我是一头快乐的小毛驴-(Zealot Yin) 

呵呵，好文不能让他埋没了，交个朋友..哈哈 [回复](#) [引用](#) [查看](#)

#3楼 2011-06-16 17:28 meicai 

顶一下

没看懂memcache的定义

弱弱的问下楼主，啥情况下或者说哪种类型的页面适合用这个。 [回复](#) [引用](#) [查看](#)

#4楼 2011-06-16 17:44 funying 

--! 我也木有看懂 [回复](#) [引用](#) [查看](#)

#5楼[楼主] 2011-06-16 17:47 enjoyeclipse 

@meicai

我们的项目memcache是缓存数据库的, 页面缓存我们用的是OutputCache或者PartOutputCache, 见我前一篇文章.

memcache其实本质上就是一个服务器端的应用程序, 只要跑起来就OK了; 我们一般用的都是客户端的库, 比如读数据库需要磁盘I/O, 很耗时间, 但使用memcache后, 就可以把数据库的数据缓存到memcache中, 下次就更快了。 [回复](#) [引用](#) [查看](#)

#6楼[楼主] 2011-06-16 17:57 [enjoyeclipse](#) 

@軒轅劍

memcache的更新有三种方式:

- 1.重启memcache所在的机器
- 2.调用ICachePolicy.Delete(string key) 方法, 清空这个key的缓存。
- 3.ICachePolicy.Add(string key, object value, DateTime dt)中的dt是过期时间, 过了之后自然就清空这个key的缓存了。

[回复](#) [引用](#) [查看](#)

#7楼 2011-06-16 18:21 軒轅劍 

```
<beitmemcached>
```

```
<add key="MemcachedConfig"
value="localhost" />
```

```
</beitmemcached>
```

我们目前的访问量还不是很大, 因此将Memcached与网站部署在一台机器上, 使用的是localhost. 如果网站和Memcache部署在不同机器, 这里的value 就应该是memcached所在机器的IP, 请再次确认11211端口可以访问。

怎么做分布式的, 多台缓存服务器的情况下怎么配置呢?

抱歉 memcache菜鸟 [回复](#) [引用](#) [查看](#)

#8楼 2011-06-16 18:30 [pjdpeng01](#) 

```
"private readonly MemcachedClient _cache;"
定义成 readonly??? 回复 引用 查看
```

#9楼[楼主] 2011-06-16 18:42 [enjoyeclipse](#) 

@pjdpeng01

我们使用的是依赖注入, 所以这里用readonly是没有问题的, 只是没有把代码都贴出来。 [回复](#) [引用](#) [查看](#)

#10楼[楼主] 2011-06-16 18:44

[enjoyeclipse](#) 

@軒轅劍

目前我们的业务量还没达到分布式阶段，但愿以后能用到，同求分布式方案。 [回复](#) [引用](#) [查看](#)

#11楼 2011-06-16 18:50 john23.net 

支持下 [回复](#) [引用](#) [查看](#)

#12楼 2011-06-16 20:25 indexRoad 

这个得顶，想学 memcache 很久了，可是都不得章法，入门不了 [回复](#) [引用](#) [查看](#)

#13楼 2011-06-16 20:31 DYStudio.Net 

鄙人写了一个缓存组件,主要是集成Enterprise Library 5,支持设计器配置,集成MemCache,下一步准备集成WindowsServerAppFabric.Memcache还不错. [回复](#) [引用](#) [查看](#)

#14楼 2011-06-17 08:36 meicai 

原来如此，非常感谢楼主的耐心回复！ [回复](#) [引用](#) [查看](#)

#15楼 2011-06-17 11:31 jomoleo 

不知道这个memcache与asp.net mvc 有什么必然的联系，以使你要使用这样一个标题 [回复](#) [引用](#) [查看](#)

 注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[返回博客园首页](#)。

最新IT新闻:

- [iCloud 将没有网页端界面?](#)
- [天梭和当当网战火升级：天梭称不提供任何质保](#)
- [腾讯发布移动安全解决方案 成立移动安全实验室](#)
- [光线传媒上市文件乌龙频现：注册资本相差600万](#)
- [迷你发行版Tiny Core Linux 3.7发布](#)
- » [更多新闻...](#)





知识库最新文章:

[微内核 \(Micro kernel\)](#)

[像人脑一样思考 揭秘Kinect工作原理](#)

[浅谈Facebook的服务器架构](#)

[JAVA与.NET的相互调用——TCP/IP相互调用基本架构](#)

[你的编程语言能这样做吗?](#)

» [更多知识库文章...](#)

网站导航: [博客园首页](#) [IT新闻](#) [我的园子](#) [闪存](#) [程序](#)
[员招聘](#) [博问](#)



[China-pub 计算机图书网上专卖店! 6.5万品种2-8折!](#)

[China-Pub 计算机绝版图书按需印刷服务](#)

简洁版式: [Asp.net mvc 网站之速度优化 --](#)

[Memcache](#)