ThinkPHP Framework 1.5
Cache Design

ThinkPHP 1.5
多元化缓存机制

编写：ThinkPHP 文档组

最后更新：2008-12-24

目录

1 概述

简述了 ThinkPHP 的缓存方式和使用。

2缓存方式

ThinkPHP 在数据缓存方面包括 SQL 查询缓存、数据对象缓存、Action 缓存、视图缓存、静态页面缓存以及浏览器缓存等多种机制，采用了包括文件方式、共享内存方式和数据库方式在内的多种方式进行缓存 通过插件方式还可以增加以后需要的缓存类 让应用开发可以选择更加适合自己的缓存方式，从而有效地提高应用执行效率。

3缓存使用

ThinkPHP 把各种缓存方式都抽象成统一的缓存类来调用，而且 ThinkPHP 把所有的缓存机制统一成一个 S 方法来进行操作，所以在使用不同的缓存方式的时候并不需要关注具体的缓存细节。

那么如何操作缓存呢？很简单，使用内置的 S 方法，例如：

```
// 使用 data 标识缓存$Data 数据
S('data',$Data);
// 缓存$Data 数据 3600 秒
S('data',$Data,3600);
// 获取缓存数据
$Data = S('data');
// 删除缓存数据
S('data',NULL);
```

系统默认的缓存方式是采用 File 方式缓存，我们可以在项目配置文件里面定义其他的缓存方式，例如

修改默认的缓存方式为 Xcache（当然，你的环境需要支持 Xcache）

```
'DATA_CACHE_TYPE'=>'Xcache'
```

通过上面的定义，相同的代码就会使用 Xcache 方式来缓存了，而事实上，代码并没有任何改变。

```
// 使用 data 标识缓存$Data 数据  有效期为默认的设置
S('data',$Data);
// 缓存$Data 数据 3600 秒
S('data',$Data,3600);
// 获取缓存数据
$Data = S('data');
```

当然，我们还可以在 S 方法里面显式的指定缓存方式，例如

```
S('data',$Data,3600,'File');
// 或者动态切换缓存方式
C('DATA_CACHE_TYPE','Xcache');
S('data',$Data,3600);
$data = S('data');
// 操作完成后切换会默认的缓存方式
C('DATA_CACHE_TYPE','File');
```

对于 File 方式缓存下的缓存目录下面因为缓存数据过多而导致存在大量的文件问题，ThinkPHP 也给出了解决方案，可以启用哈希子目录缓存的方式，只需要设置

```
'DATA_CACHE_SUBDIR'=>true
```

就可以根据缓存标识的哈希自动创建子目录来缓存。

4 浏览器缓存

可以在 ThinkPHP 里面启用浏览器缓存功能，设置 LIMIT_RESFLESH_ON 为 True，并且设置缓存有效时

间 LIMIT_REFLESH_TIMES （单位为秒），该功能可以防止页面频繁刷新，系统会根据用户访问页面的$_SERVER['PHP_SELF']的值来进行缓存标识。

5 相关配置

相关的缓存配置参数还包括：

```
'DATA_CACHE_TIME'=>-1,              // 数据缓存有效期
'DATA_CACHE_COMPRESS'=>false,       // 数据缓存是否压缩缓存
'DATA_CACHE_CHECK'         =>false,  // 数据缓存是否校验缓存
'DATA_CACHE_TYPE'         =>'File',  // 数据缓存类型 支
持 File Db Apc Memcache Shmop Sqlite Xcache Apachenote Eaccelerator
'DATA_CACHE_SUBDIR'=>false, // 使用子目录缓存（自动根据缓存标识的哈希创建子目录）
```

6 注意事项

下面的一些问题需要引起注意，便于在开发过程中更快的解决问题：