

libshmcache是基于共享内存的本地缓存库。libshmcache写有锁，读无锁，性能非常高。libshmcache直接访问本地共享内存，速度比远程接口如redis快100倍以上。libshmcache包含C库和PHP扩展（支持PHP 5和PHP 7）。在众多开源缓存软件的今天，libshmcache作为后起之秀，凭借诸多特性，有望成为本地缓存开源软件的一颗明星。

### 1. 高性能特性

- 1) 写采用pthread互斥锁，读无锁；
- 2) 使用hash table进行快速set、get和delete；
- 3) 使用对象池（FIFO队列）对hash table键值对进行分配；
- 4) value buffer分配使用striping（条带化）分配机制，在striping内按顺序分配buffer。当释放value buffer时，只需要减少striping已使用空间（used size）。释放value buffer后，若striping已使用空间变为0，回收整个striping；
- 5) 淘汰算法使用简单高效的FIFO而不是LRU。

### 2. 稳定性特性

- 1) 死锁检测和自动解锁机制。死锁发生场景：正在修改数据的进程挂掉，会导致死锁；
- 2) 初始化时检查一些关键字段的一致性，当内存相关的参数发生变化时，系统会自动提示共享内存需要清理和重新初始化；
- 3) 当回收了有效的（未过期）键值对时，休眠一定时长以避免其他进程读到脏数据。

### 3. 其他特性

- 1) 除了支持有亲缘关系的进程（如父进程和子进程），还支持无亲缘关系的进程（如PHP FPM进程和PHP CLI进程，多个PHP CLI进程等等）；
  - 2) 按需增量分配value buffer，有效减少初始内存大小，避免内存浪费（通过配置参数segment\_size来设置）；
  - 3) 提供丰富的统计信息：get、set和delete次数统计、内存回收统计、锁使用统计等；
  - 4) 支持原子增加（减少）；
  - 5) PHP扩展支持多种序列化方式：igbinary、msgpack、PHP序列化，以及无需序列化。因为键值对中存储了所用的序列化方式，这几种序列化方式可以并存；
- \* 友情提示：key长度不能超过64字节。

### libshmcache使用场景

如果需要缓存的数据量不是太大，比如不超过100w个key，对缓存读写性能要求又比较高的情况下，可以考虑使用libshmcache。总结为一句话：追求极致性能，以空间换时间。

libshmcache采用的开源协议为BSD，托管在github，地址：<https://github.com/happyfish100/libshmcache>