共享内存cache架构设想

需求：

总的来说：设计一种使用共享内存作为cache的系统，cache由key-> value 组成

细的来说：此cache，有如下特点：

**强容错机制**

1. 当某挂接进程abort掉后，只破坏当时操作的key对应value；
2. 如果某进程操作指针破坏了共享内存某些地方，影响应该减到最小；

**各value之间严格隔离；**

**Lru淘汰简单，且与具体业务，具体数据无关；**

**内存分配简单，gc简单；**

**适合大单位复杂数据value；**

**key 采用定长//简单；**

**Value采用自描述结构组织；**

设计：

**MemHeader**

Header header//4096字节

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CreateTime | Revision | ChunkSize | ChunkNum | VirginChunkIndex | FreeChunkIndex |
| int32 | Int32 | Int32 | Int32 | Int32 | Int32 |
| UsedChunkNum | MaxKeyNum | KeyNum | MaxKeyLen | KeyType | KeyHbNum |
| int32 | int32 | Int32 | Int16 | Int8 |  |
| GetCnt | GetHitCnt | AddCnt | WahCnt | FailedCnt | MaxKeySearchN |
| int32 | int32 | Int32 | Int32 | Int32 | Int32 |
| AllKeySearchN | MaxChunkE |  |  |  |  |

注：avrage search = AllKeySearchN / GetHitCnt

KeyHashtable

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |

内存布局

|  |  |  |
| --- | --- | --- |
| LruHead | HashBuckts | KeyNode |

Struct keyNode {

T \_key;

int \_ck\_index;

};

MemHeader 放在一块共享内存里面；

MemData 放另外一块共享内存；

这样方便key或者data数目扩展；

**MemData**

一个value包括：value，struct；一个 mem mgr

Mem分配原则：一次向mem data申请分配一个chunk，当chunk用完再向memdata申请；各chunk用链表串联；

|  |  |  |
| --- | --- | --- |
| Verify | Random(8) | Version(4) |
| FreeIndex | ChunkNum | UsedMem |
| Struct | ChunkVector |  |