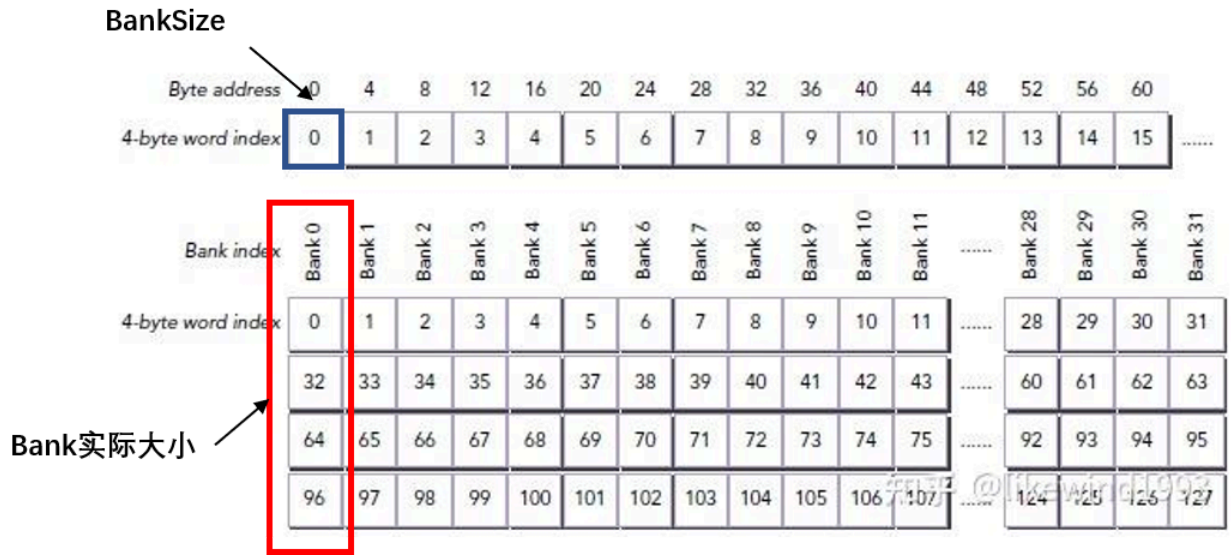
04 消除BankConflict

Bank概念

英伟达GPU上的`shared_memory`，会被映射到大小相等的32个Bank上（即`BankNum`为32）。`shared_memory`可以被一个warp中的所有线程（32个线程）进行访问，Bank的数据读取带宽为32 bit / cycle。



`BankSize`的大小为4 byte，而Bank实际大小是`BankSize`**×层数**，也就是这里的16个byte。`BankIndex`是Bank索引，范围是[0，BankNum)，关于BankIndex的计算，见如下的公式：

$$\mathrm{BankIndex} = \mathrm{addr}/\mathrm{BankSize} \bmod \mathrm{BankNum}$$

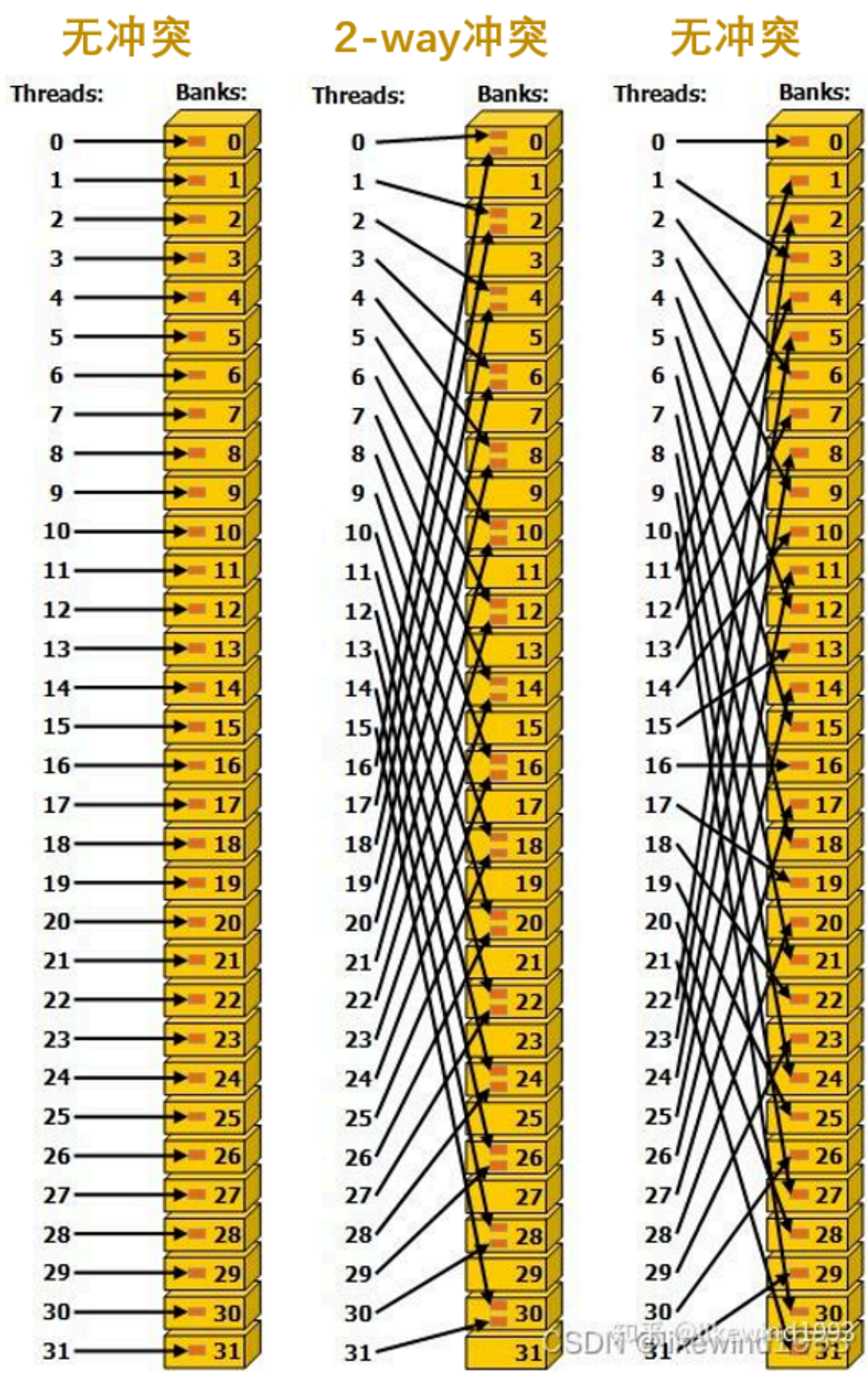
Bank冲突原理

> However, if multiple threads' requested addresses map to the same memory bank, the accesses are serialized. The hardware splits a conflicting memory request into as many separate conflict-free requests as necessary, decreasing the effective bandwidth by a factor equal to the number of colliding memory requests. An exception is the case where all threads in a warp address the same shared memory address, resulting in a broadcast. Devices of compute capability 2.0 and higher have the additional ability to multicast shared memory accesses, meaning that multiple accesses to the same location by any number of threads within a warp are served simultaneously.
>
> - -- 《Using Shared Memory in CUDA C/C++》

有如下一些重要信息：

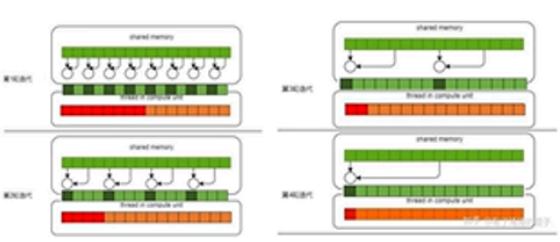
- 当多个线程读写**同一个Bank**中的数据时，会由硬件把内存读写请求，拆分成 **conflict-free requests**，进行顺序读写；
- 特别地，当一个warp中的多个线程读写**同一个地址**时，会触发**broadcast**机制，此时不会退化成顺序读写。

因此，Bank Conflict是指：**在访问shared memory时，因多个线程读写同一个Bank中的不同数据地址时，导致shared memory 并发读写 退化 成顺序读写的现象。**示意图如下所示：


无冲突
2-way冲突
无冲突
Threads:
Banks:
0 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 31
0 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 31

Bank冲突解决方式

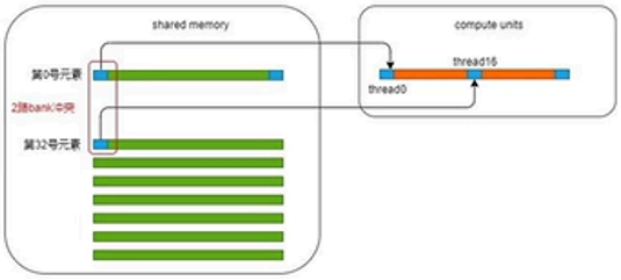
在示例程序中，存在2路/4路Bank 冲突，示意图如下：

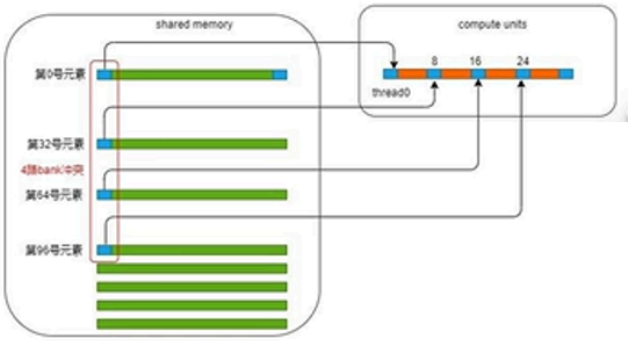


只看一个block中的0号warp
- 第一次迭代：

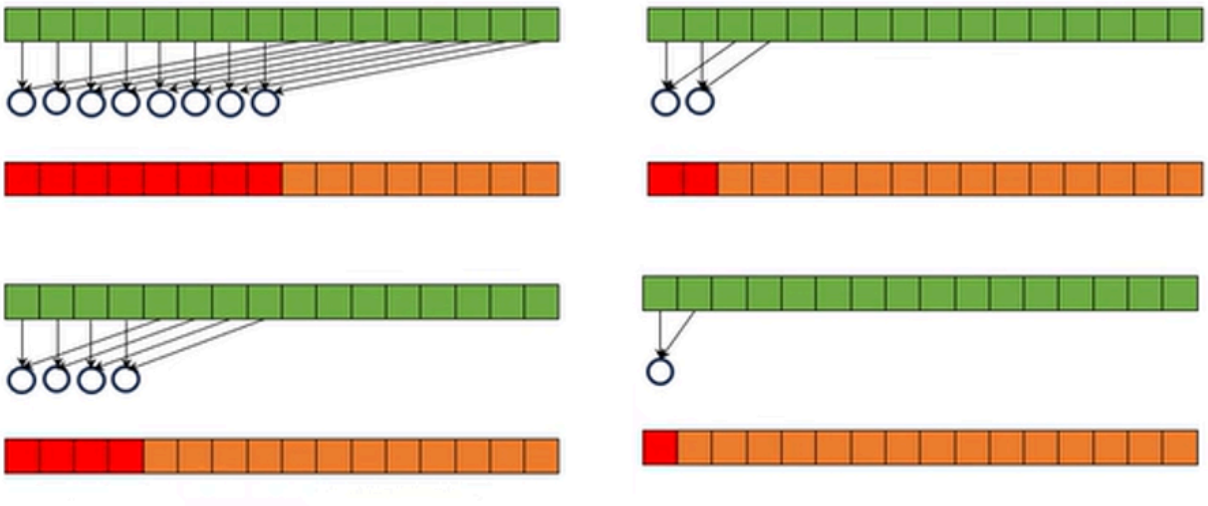
thread0访问：0号，1号
thread16访问：32号，33号
- 第二次迭代：

thread0访问：0号，2号
thread8访问：32号，34号
thread16访问：64号，66号
thread24访问：96号，98号





解决的一种思路是：**当前线程与一半的block线程进行相加**，示意图如下所示：



只看一个block中的0号warp
· 第一次迭代：
thread0访问：0号，128号
thread8访问：8号，136号
thread16访问：16号，144号
thread24访问：24号，152号

· 第二次迭代：
thread0访问：0号，64号
thread8访问：8号，72号
thread16访问：16号，80号
thread24访问：24号，88号

工程链接

https://github.com/JingliangGao/CudaRoad/tree/main/reduce_case/4_reduce_no_bank_conflict

参考

1. 《【CUDA编程概念】一、什么是bank conflict？》