**Caribou: Intelligent Distributed Storage**

主机处理器➕固态硬盘内嵌入式处理器

应用领域：数据库

设计动机：数据迁移开销

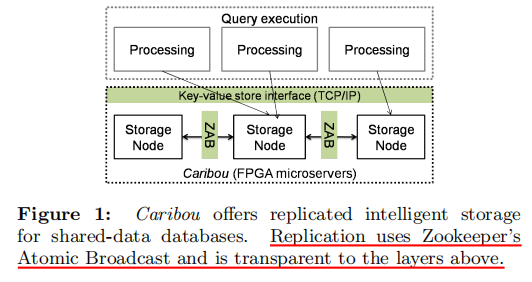
对于带宽、延迟和能耗来说，数据移动的开销很大，尤其是在许多数据库操作的计算复杂度很低的情况下。（计算复杂度低则更容易将操作移近存储数据）

Caribou通过一个简单的键值存储接口通过网络提供对DRAM/NVRAM存储的访问，每个存储节点提供高带宽的近数据处理并通过复制容错。

Caribou使用FPGA搭建其原型。

Caribou具有以下特点：

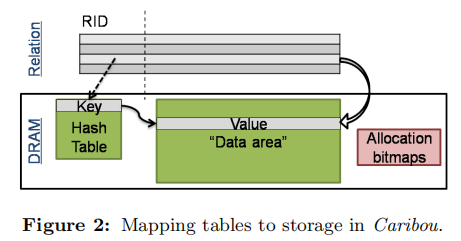
1. 低延迟、高吞吐量
2. 查找、扫描机制→减少访问存储中数据的请求，减少数据移动
3. 近数据处理
4. 通过复制进行容错
5. 建立在传统网络上的简单接口



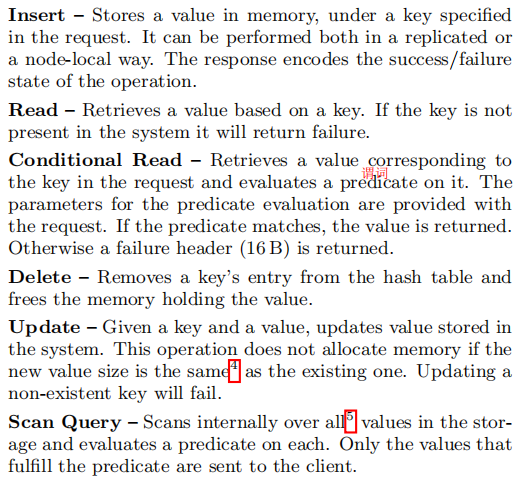
特点5

特点4

其中Storage Node的键值存储结构如下：

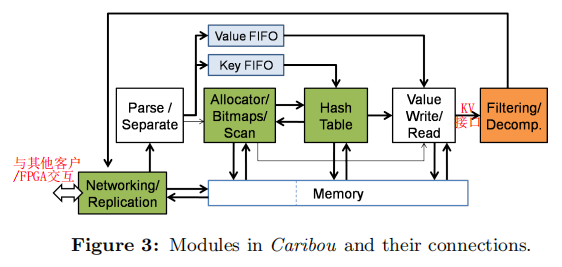


Processing能提供如下操作：



特点3

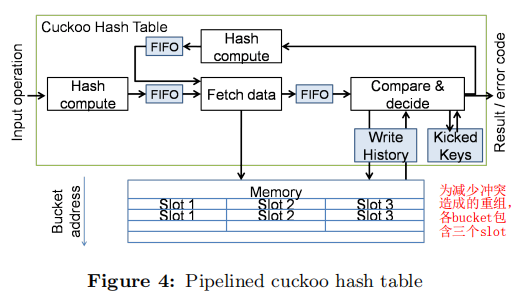
单个Caribou模块及操作流程：



以哈希表为核心，键与值分开管理，橙色模块为近数据处理模块，可并行。

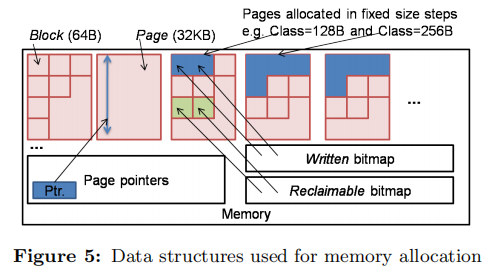
特点1

哈希表采用cuckoo哈希表，其与slab-based内存分配机制，能使延迟低，带宽高。（改善对冲突和不同长度值的处理）



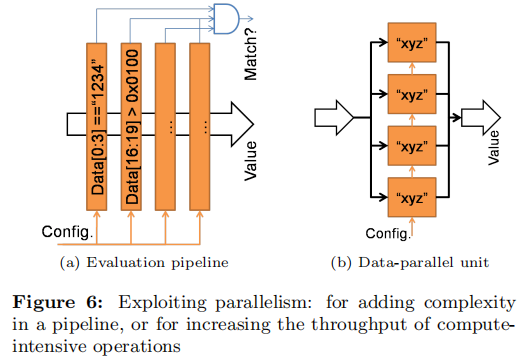
在cuckoo哈希表中，各项都有两个可能的位置（由两个不同的哈希函数确定）。当搜索一个键时，它必须位于它的两个位置之一，否则它不在表中。如果在插入时两个位置都被占用，则在其两个位置中随机选择一个键“踢出”。然后在后台重新插入此键（使用上图顶部的连接完成）。

数据结构如下：（以值的尺寸定义类、位图用于扫描）



特点2

各类处理操作的卸载均应用了并行性：



Summary：

Caribou, a hardware-managed distributed storage solution with near-data computation, was proposed in 2017 to reduce data movement. Its first prototype was created by FPGAs.

Caribou provides access to DRAM/NVRAM storage over traditional network through a simple key-value store interface and push operations down to the storage layer with high-bandwidth. With integrating replication into a complex processing pipeline, fault tolerance is provided for each storage node. By implementing a cuckoo hash table with a slab-based memory allocator to improve the handling of collisions and various value sizes, Caribou gains low latency and high throughput. Thanks to the dataflow parallelism of hardware, Caribou delivers competitive performance while also providing rich function offloading and both point-lookup and scan-based access methods, which minimize data transferred over the network.