# 12.哈希库

DPDK提供了一个用于创建哈希表的哈希库，哈希表可以用于快速查找。哈希表是针对一组条目进行搜索而优化的数据结构，每个条目由唯一Key标识。为了提高性能，DPDK哈希要求所有的Key值具有与哈希创建时指定的相同字节数。

## 12.1.哈希API概述

哈希的主要配置参数包括：

* 哈希条目总数（哈希容量）。
* Key的字节数。

哈希还允许配置一些低级实现相关的参数：

* 将Key转换为哈希桶索引值的哈希函数

哈希库导出的主要方法包括：

* 使用Key值添加条目：Key值作为输入参数。如果新条目成功添加到指定Key的哈希中，或者已经有指定Key的条目，则返回该条目的位置。如果操作不成功，例如由于在哈希中缺少空闲条目，则返回负值;
* 使用Key删除条目：Key值作为输入参数。如果在哈希中找到具有指定Key的条目，则会从哈希中删除该条目，并返回该条目在哈希中找到的位置。如果哈希中没有指定Key的条目存在，则返回一个负值；
* 使用Key查找条目：Key值作为输入参数。如果在哈希（查找命中）中找到具有指定Key的条目，则返回条目的位置，否则返回（查询未命中）一个负值。

除了这些方法，API还为用户提供了三个选项：

* 使用Key和precomputed hash来查找/添加/删除条目：Key及其precomputed hash都作为输入。这允许用户更快地执行操作，因为已经预先计算了散列。
* 使用Key和数据来查找/添加条目：提供Key-value作为输入。这允许用户不仅存储Key，还可以存储8byte的整形或是一个指向外部数据的指针（数据超过8byte）。
* 上述两个选项的组合：用户可以提供Key、precomputed hash或是data。

此外，API包含一种方法，允许用户在突发中查找条目，实现比查找单个条目更高的性能，因为该函数在与第一个条目操作时预取下一个条目，显著降低了必要的内存访问。请注意，此方法使用8个条目（4个阶段2条目）的流水线，因此强烈建议每个突发使用至少8个条目。

与每个Key相关联的实际数据可以由用户使用单独的表格进行管理，该表格根据哈希条目数量和每个条目的位置来反映哈希表，如以下部分中描述的流分类用例所示，当然，也可以直接存储在哈希表本身。

L2 / L3转发示例应用程序中的哈希表根据由五元组查找标识的数据包流定义将数据包转发到哪个端口。然而，该表还可以用于更复杂的特征，并提供可以在分组和流上执行的许多其他功能和动作。

## 12.2.多进程支持

哈希库可以在多进程环境中使用，只需查找线程安全即可。只能在单进程模式下使用的唯一函数是rte\_hash\_set\_cmp\_func()，它设置一个自定义的比较功能，分配给一个函数指针（因此在多进程模式下不支持）。

## 12.3.实现细节

哈希表有两个主表：

* 第一个表是一组条目，进一步分为桶，每个桶中具有相同数量的连续数组条目。每个条目包含计算的给定Key的主要和次要散列（如下所述）和第二个表的索引。
* 第二个表是存储在哈希表中的所有Key的数组及其与每个Key相关联的数据。

哈希库使用Cuckoo hash（布谷鸟散列）方法来解决冲突。对于任何输入Key，有两个可能的桶（主要和次要/替代位置），其中该Key可以存储在散列中，因此只有当查询Key时才需要检查桶中的条目。与通过线性扫描阵列中的所有条目的基本方法相反，通过将散列条目的总数减少到两个哈希桶中的条目数来减少要扫描的条目数以提升查找速度。哈希使用散列函数（可配置）将输入Key转换为4字节Key签名。桶索引值是将哈希Key签名对哈希桶数取模数的值。

一旦识别出桶，哈希添加，删除和查找操作的范围就减少到这些存储区中的条目（很可能条目在主存储桶中）。

为了加快桶内的搜索逻辑，每个散列条目将4字节Key签名与每个哈希条目的完整Key一起存储。对于大的Key，将输入Key与来自存储桶的Key进行比较比将输入Key的4字节签名与来自存储桶的Key签名进行比较要花费更多的时间。因此，首先完成签名比较，仅在签名匹配时才完成Key比较。完全Key比较仍然是必要的，因为来自相同存储桶的两个输入Key仍然可能具有相同的4字节签名，尽管对于该组输入密钥提供良好的均匀分布的散列函数，该事件相对较少。

查找实例：  
首先，主桶被识别，条目可能存储在那里。如果签名存储在那里，我们将其Key与提供的Key进行比较，并返回其存储位置和/或与该密钥相关联的数据（如果有匹配）。如果签名不在主桶中，则查找辅助桶，在那里执行相同的过程。如果没有匹配，Key对应条目被认为不在表中。

添加实例：  
像查找操作一样，Key标识主和二级桶。如果主桶中有一个空槽，则主签名和辅助签名存储在该槽中，Key和数据（如果有的话）被添加到第二个表中，并且第二个表中的位置的索引被存储在第一张表上。如果主桶中没有空槽，则该桶中的一个条目将被推送到其替代位置，并将要添加的Key插入第一个条目的位置上。要知道驱逐条目（第一个条目）的替代桶的哪个位置，则查找器辅助签名，并从上面的模数中计算备用桶索引。如果替代桶中有空间，则将被驱入的条目存储在其中。如果没有，则重复相同的过程（其中一个条目被推送），直到找到非完整的数据桶。请注意，尽管所有的条目移动都在第一张表中，第二张表没有被触动，这也将在性能上受到很大影响。

在非常不太可能的事件中，该表进入循环，其中相同的条目被无限期地驱逐，则认为Key不能被存储。使用随机Key，该方法允许用户获取约90％的表利用率，而不必放弃任何存储的条目（LRU）或分配更多内存（扩展桶）。

## 12.4.哈希表中的条目分发

如上所述，如果有一个新的条目要被添加到哪个主桶，而当前已经有数据在里面时，则将数据推送到他们的替代位置，Cuckoo哈希实现了将元素推出他们的存储区。

因此，当用户向哈希表添加更多条目时，桶中散列值的分布将发生变化，其中大部分位于主要位置，并且其次要位置会随之增加，随后表将增加。

这些信息是非常有用的，因为随着更多条目逐出其次要位置，性能可能会降低。

下表显示了表利用率增加时的示例条目分布。

Table 12.1 Entry distribution measured with an example table with 1024 random entries using jhash algorithm

| **%Table used** | **%In Primary location** | **%In Secondary location** |
| --- | --- | --- |
| 25 | 100 | 0 |
| 50 | 96.1 | 3.9 |
| 75 | 88.2 | 11.8 |
| 80 | 86.3 | 13.7 |
| 85 | 83.1 | 16.9 |
| 90 | 77.3 | 22.7 |
| 95.8 | 64.5 | 35.5 |

Table 12.2 Entry distribution measured with an example table with 1 million random entries using jhash algorithm

| **%Table used** | **%In Primary location** | **%In Secondary location** |
| --- | --- | --- |
| 50 | 96 | 4 |
| 75 | 86.9 | 13.1 |
| 80 | 83.9 | 16.1 |
| 85 | 80.1 | 19.9 |
| 90 | 74.8 | 25.2 |
| 94.5 | 67.4 | 32.6 |

注意：上表上的最后值是具有随机密钥和使用Jenkins散列函数的平均最大表利用率。

## 12.5.用例：流分类

流分类用于将每个输入数据包映射到它所属的连接/流。这种操作是必需的，因为每个输入分组的处理通常在其连接的上下文中进行，因此相同的操作集合被应用于来自相同流的所有分组。

使用流分类的应用通常具有要管理的流表，每个单独的流具有与该表相关联的条目。流表条目的大小是特定于应用程序的，典型值为4,16,32或64字节。

使用流分类的每个应用通常具有被定义为从输入报文中读取一个或多个字段来构成Key，用于标识流。一个例子是使用由IP和传输层数据包头的以下字段组成的DiffServ 5元组：源IP地址，目标IP地址，协议，源端口，目标端口。

DPDK哈希提供了一种通用的方法来实现应用程序指定的流分类机制。 给定一个用数组实现的流表，应用程序应该创建与流表相同数量的条目的哈希对象，并将哈希密钥大小设置为所选流Key中的字节数。

应用侧的流程表操作如下：

* 添加流：将流Key添加到哈希。如果返回的位置有效，则使用它来访问流表中用于添加新流或更新与现有流相关联的信息的流条目。否则，流添加失败，例如由于缺少用于存储新流的空闲条目。
* 删除流：从哈希中删除流Key。如果返回的位置有效，则使用它来访问流表中的流条目以使与流相关联的信息无效。
* 查找流：在哈希中查找流Key。如果返回的位置有效（流查找命中），则使用返回的位置来访问流表中的流条目。否则（流查找未命中）表示当前数据包没有注册流。

## 12.6.参考

Donald E. Knuth, The Art of Computer Programming, Volume 3: Sorting and Searching (2nd Edition), 1998, Addison-Wesley Professional

原文链接：[http://www.jianshu.com/p/4db3afc630d6](https://www.jianshu.com/p/4db3afc630d6" \t "_blank) 。