Storm 分享

邓志华

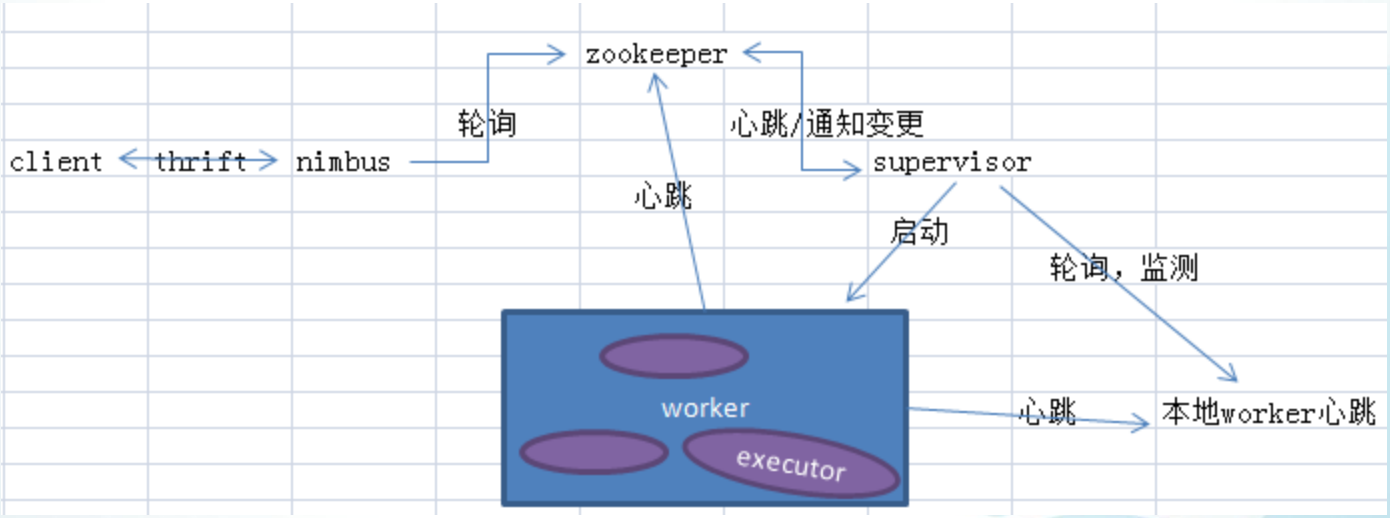
2015年6月

酷讯旅游 KUXUN.CN

主要内容

- Storm 是如何维护集群的?
- Storm 是如何分配任务的?
- Storm 是如何分发消息的?
- Task 是如何收到消息?
- Ack机制
- Storm调优手段

整体结构

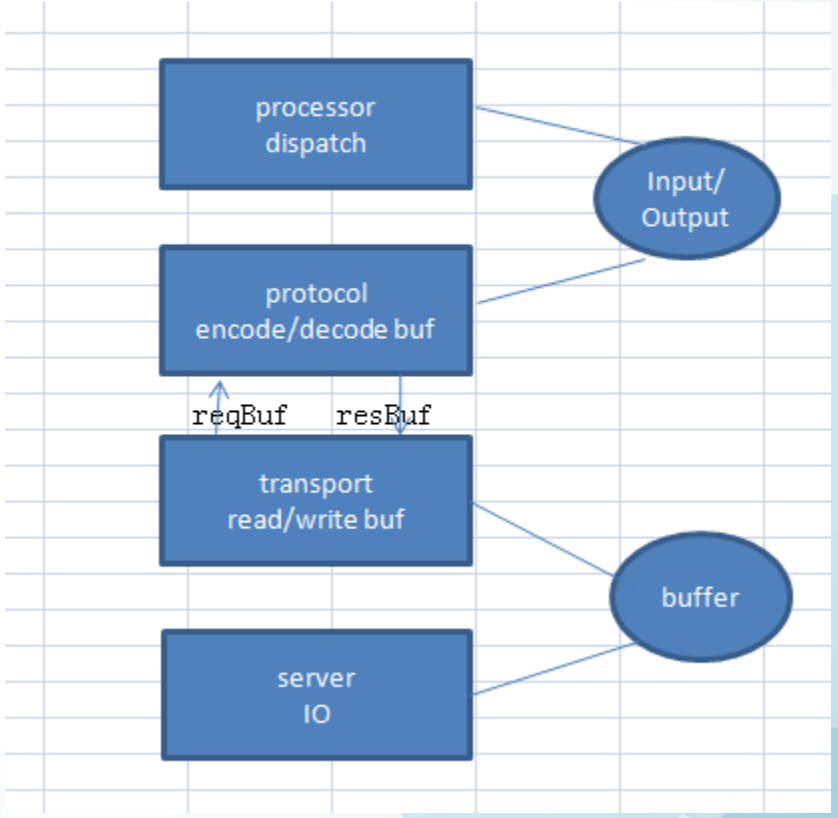

zookeeper
轮询
心跳/通知变更
client
thrift
nimbus
supervisor
心跳
启动
轮询，监测
worker
executor
心跳
本地worker心跳

Nimbus-维护集群，分配任务

thrift 服务
zookeeper
java.util. PriorityQueue
抽屉算法

Nimbus- thrift 服务

C/S架构
NIO
实现MVC中VC



酷讯旅游 KUXUN.CN

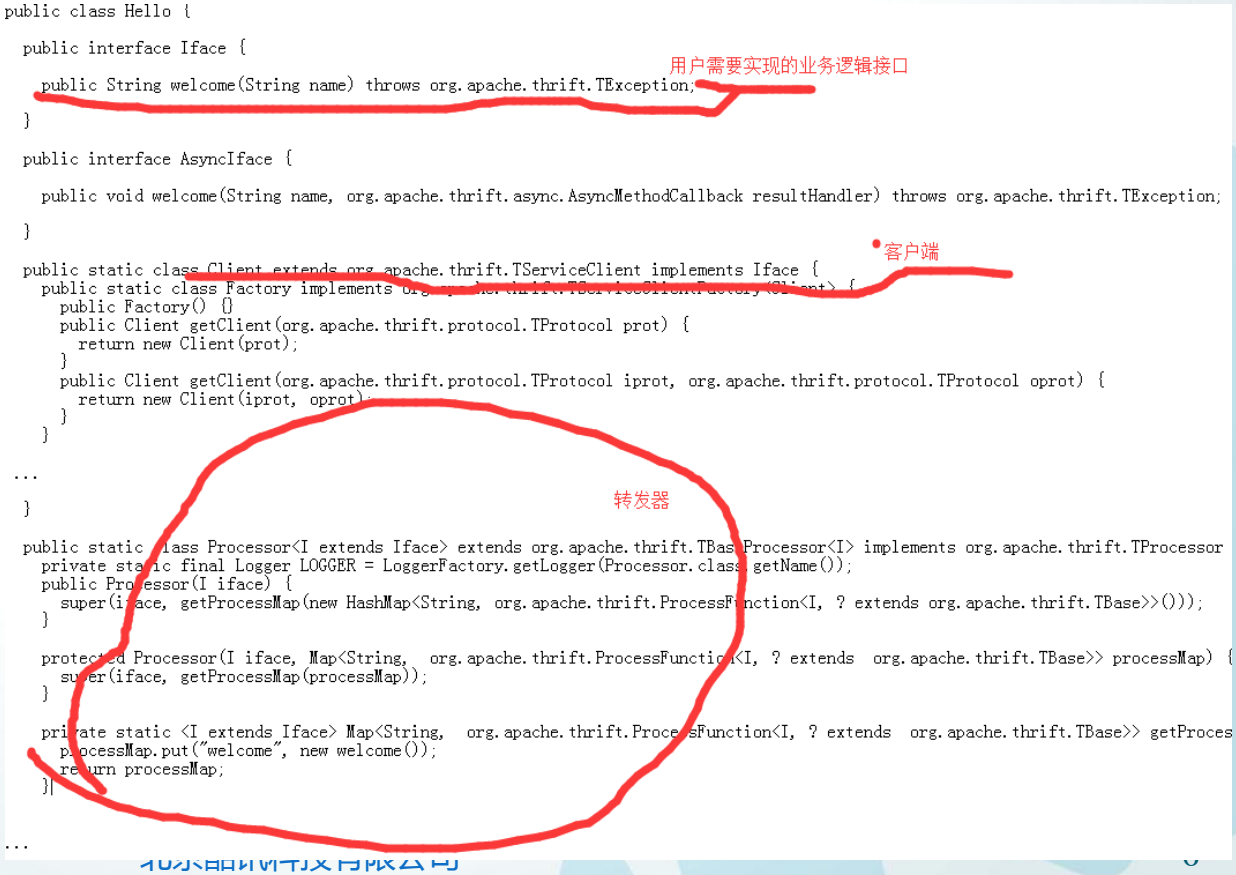
namespace java com.kuxun.generated

service Hello{

string welcome (1: string ame)

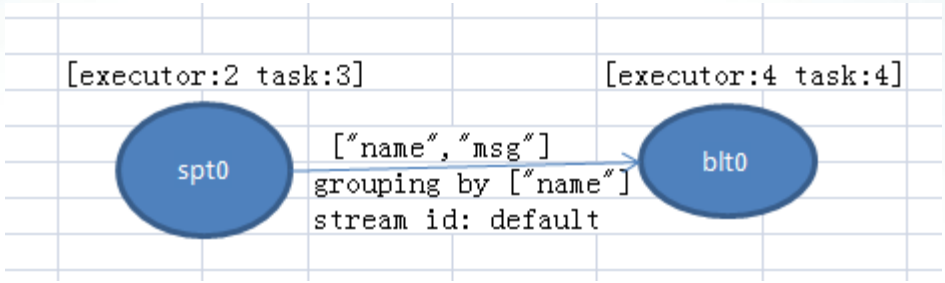
} ->

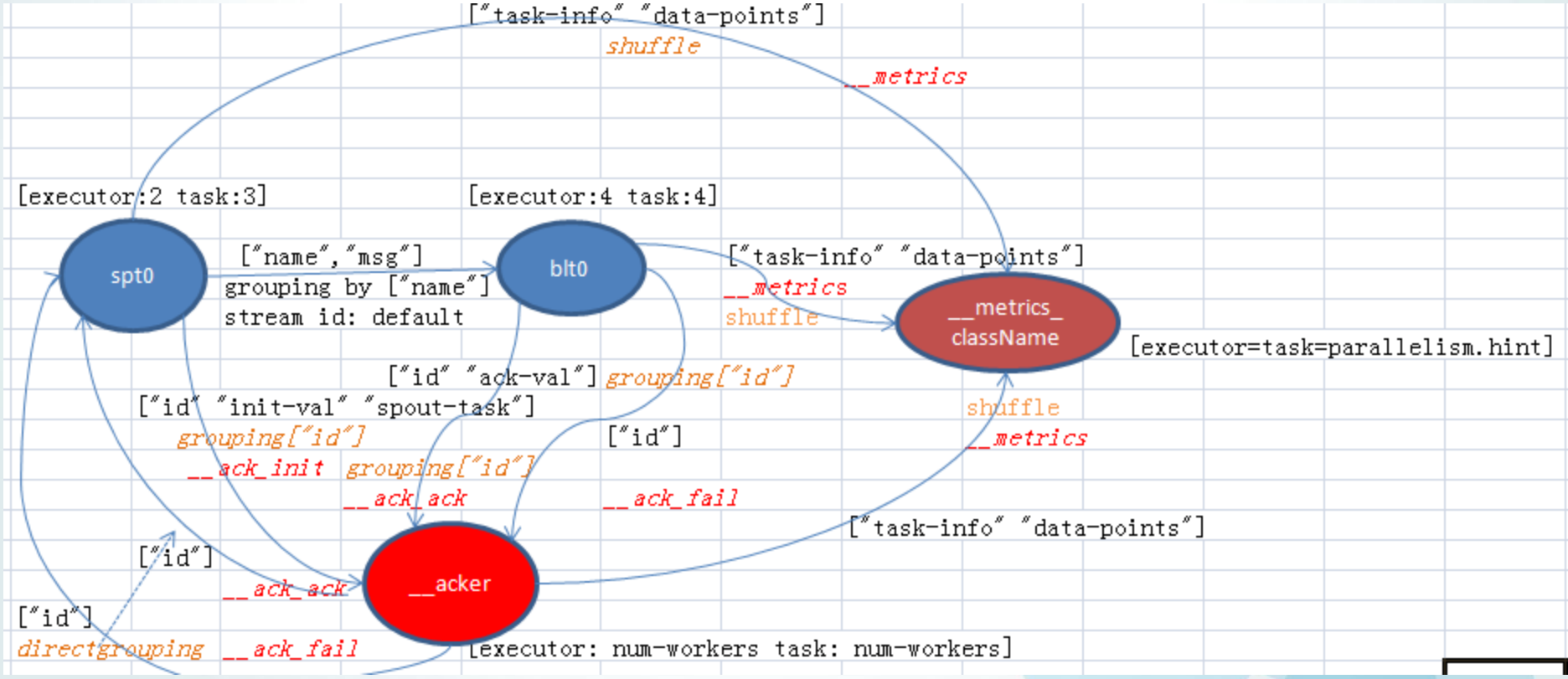
thrift --gen java

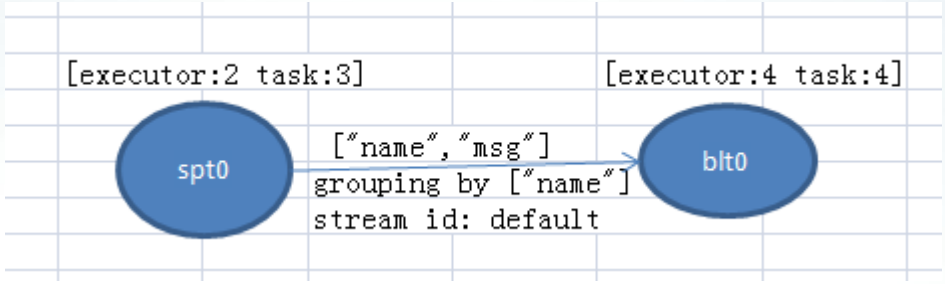


Nimbus-Zookeeper

-/
 --/supervisors/$supervisor-id
 :used-ports->(6703 6704 6705 6706 ...)
--/assignments/$storm-id
 [193 193]->["73f3da46-85b3-4d7f-86a9-f94cf03872a7" 6713]
--/storms/$storm-id
--/workerbeats/$storm-id/$worker-id


[executor:2 task:3]
[executor:4 task:4]
spt0
["name","msg"]
grouping by ["name"]
stream id: default
blt0


["task-info" "data-points"]
shuffle
__metrics
[executor:2 task:3]
[executor:4 task:4]
spt0
["name","msg"]
grouping by ["name"]
stream id: default
blt0
["task-info" "data-points"]
__metrics
shuffle
__metrics_
className
[executor=task=parallelism.hint]
["id" "ack-val"]
grouping["id"]
["id" "init-val" "spout-task"]
grouping["id"]
__ack_init
grouping["id"]
__ack_ack
["id"]
__ack_fail
shuffle
__metrics
["task-info" "data-points"]
["id"]
__ack_ack
__acker
["id"]
directgrouping
__ack_fail
[executor: num-workers task: num-workers]



{1 blt0 2 blt0 3 blt0 4 blt0 5 spt0 6 spt0 7 spt0}

blt0: {1 4} -> (1 1 1 1) ->(1) (2) (3) (4)
-> [1 1] [2 2] [3 3] [4 4]

spt0 : {1 1 2 1} ->(1 2) ->(2 1) ->(5 6) (7)
->[5 6] [7 7]

Utils.integerDivided 方法实现

blt0: -> [1 1] [2 2] [3 3] [4 4]

spt0：->[5 6] [7 7]

设有三台supervisor机器(i, j,k), [i 6703] [i 6704]
[j 6705] [j 6711] [k 6700] 资源可用。

排序：

[i 6703] [j 6705] [k 6700] [i 6704] [j 6711]

take 2(假设topology使用2个worker):

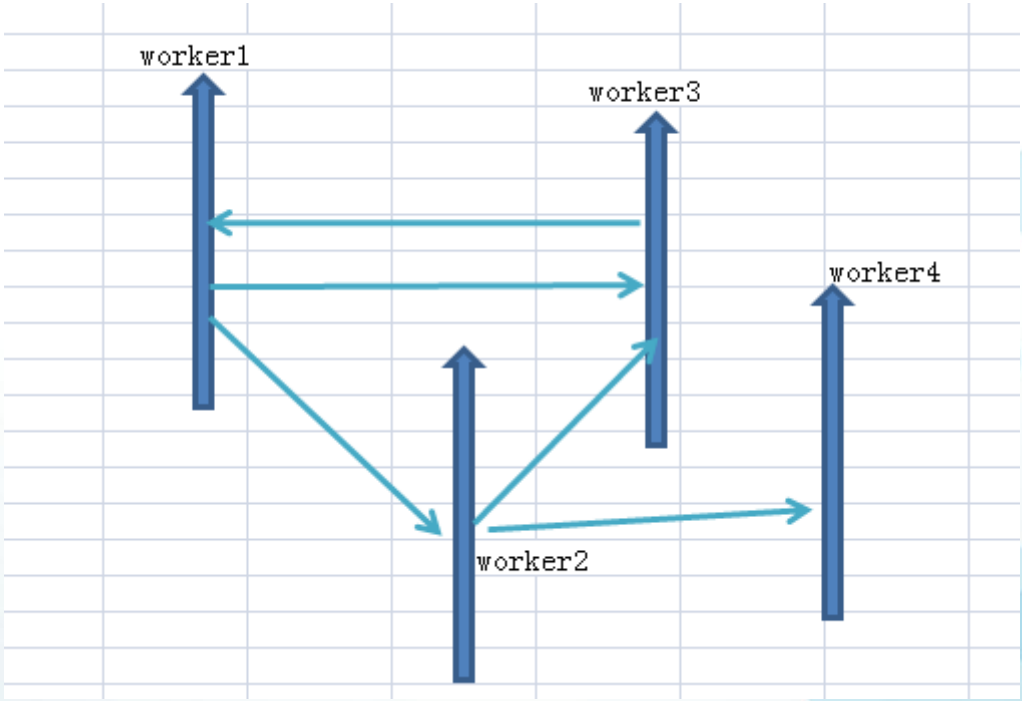
[i 6703] [j 6705]

按照executor的数量重复输出上一步结果形成集合l。

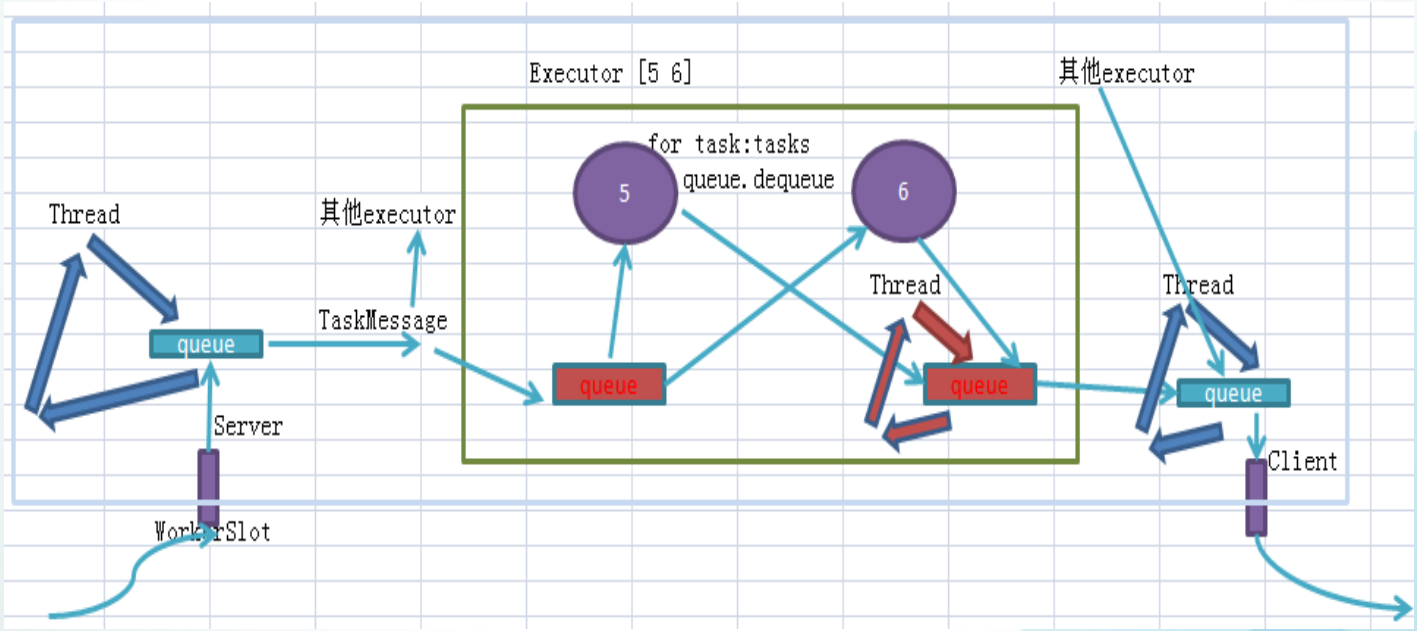
映射 f(x in executors) = g(x in l)

Worker互联

建立TCP长连接



Worker内部




```
private int _task;
private byte[] _message;
```

1. fieldGrouping

```
(-> (.select out-fields group-fields values)
        tuple/list-hash-code
        (mod num-tasks)
        task-getter))))
(defn list-hash-code
  [^List alist]
  (.hashCode alist))
```

2. shuffleGrouping

Collections/shuffle(List targetTasks)

洗牌 取出List第一个

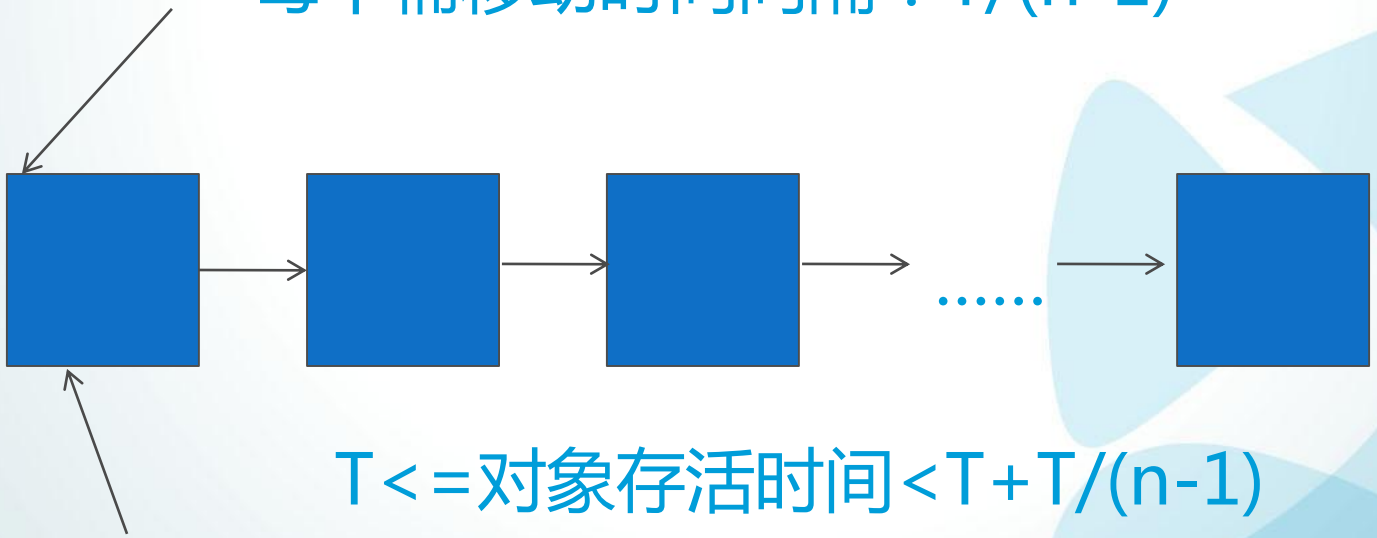
shuffle

```
    int size = list.size();
    if (size < SHUFFLE_THRESHOLD || list instanceof
RandomAccess) {
        for (int i=size; i>1; i--)
            swap(list, i-1, rnd.nextInt(i));
    } else {
        Object arr[] = list.toArray();
        // Shuffle array
        for (int i=size; i>1; i--)
            swap(arr, i-1, rnd.nextInt(i));
```

TimeCacheMap/ RotatingMap

1. 删除尾部节点

每个桶移动时间间隔：T/(n-1)



T<=对象存活时间<T+T/(n-1)