# Java 并发容器和框架

本章主要讲了ConcurrentHashMap、非阻塞线程安全队列ConcurrentLinkedQueue、阻塞线程安全队列BlockingQueue和Fork/Join框架的设计与实现。

## 1 ConcurrentHashMap (结合绿皮书看)

**1.1 前言**：

HashMap不是线程安全的，HashMap在并发执行put操作时会引起死循环，是因为多线程会导致HashMap的Entry链表形成环形数据结构，一旦形成环形数据结构，Entry的next节点永远不为空，就会产生死循环获取Entry。

HashTable基于Synchronized保证线程安全，效率低。

**1.2设计原理**：

ConcurrentHashMap在jdk1.7中是基于锁分段技术实现的，具体为Segment数组结构+HashEntry数组结构。分别解释Segment和HashEntry如下：

Segment：继承于ReentrantLock，扮演锁的结构。一个Segment类似于一个HashMap，包含一个HashEntry数组。这样在操作某个HashEntry数组的数据时，就必须获取对应Segment锁。

HashEntry：一种链表结构，即每个HashEntry数组元素对应一个HashEntry链表。

整个ConcurrentHashMap结构如下：

略

**1.3源码解读**：

**初始化参数**：

initialCapacity(默认值 16)：ConcurrentHashMap的初始化容量，就是所有Segment的HashEntry数组的总数量。注意，不是存储元素的总数量。

loadFactor(默认值 0.75)：每个segment的负载因子，即segment中非空hashEntry元素占hashEntry总数的loadFactor时，就要进行扩容。

concurrencyLevel(默认值 16)：并发等级，用于计算segment数组的大小ssize。concurrencyLevel的最大值是65535，对应二进制是16位。

**其他参数**：

ssize：Segment数组大小，由concurrencyLevel计算得出，同时要保证值是2的N次方，即当concurrencyLevel等于14、15或16时，ssize都会等于16，即Segment数组大小为16。(默认值 16，由concurrencyLevel默认值16导出)，最大值65536，由concurrencyLevel最大值65535导出。

sshift：1执行左移操作(即乘2)到ssize的次数，(默认值 4，2^4 = 16)，最大值16,2^16=65536达到ssize最大值

segmentShift：用于定位参与散列运算的位数，等于32-sshift。(默认值 28，32-4=28)，最小值16，32-sshift最大值=16

segmentMask：用于散列运算的掩码，等于ssize减1。(默认值 15，16-1=15) 最大值 65535,ssize最大值65536-1=65535

**补充**：初始化操作会保证每个Segment的HashEntry[]的大小也为2的指数。

**定位Segment**: