Java Concurrency

Allen

Agenda:

1:basic concepts of thread

2:basic synchronisation methods

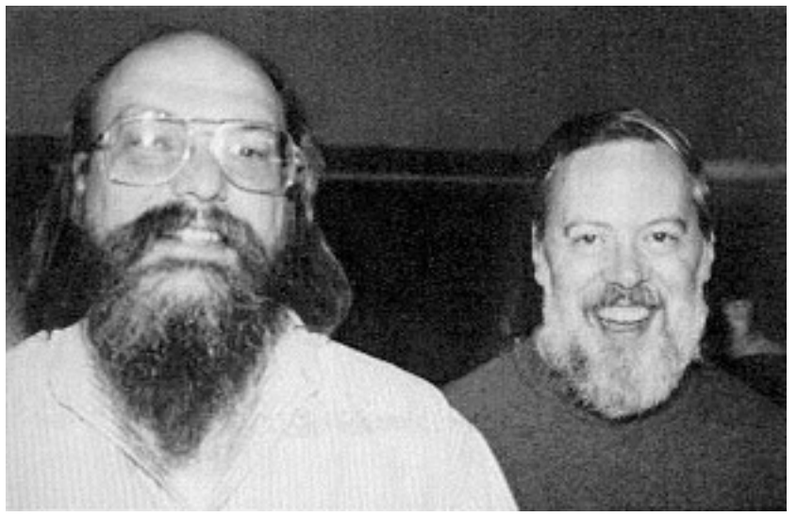
3:concurrency collections

4:thread management

5:concurrency test

6:some classical problems

To me, process is a concept and thread is an implementation.
I would like to see the implementation get closer to the concept



Ken Thompson

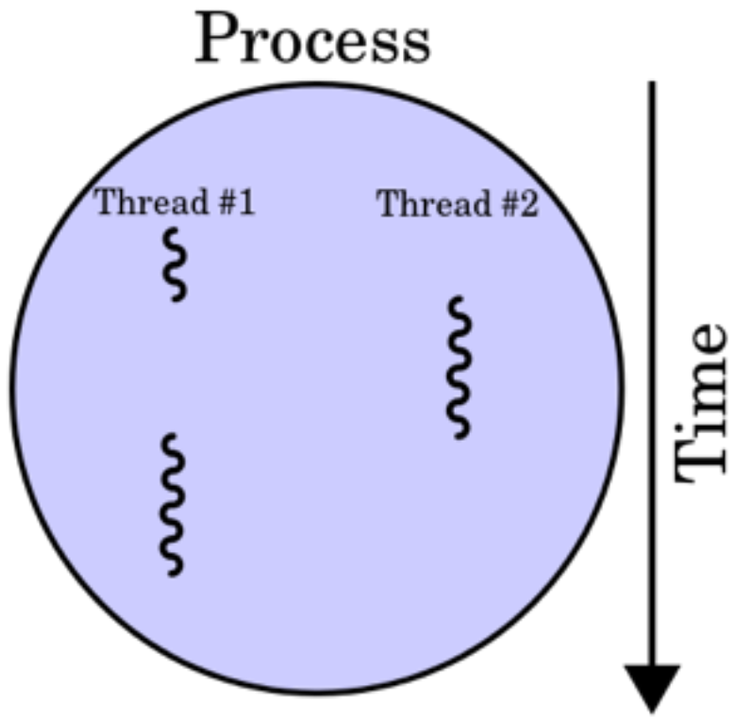
Unix system
The B programming language

basic concepts of thread

(1)the smallest sequence of programmed instructions

(2)sharing code,data and much lighter context switch

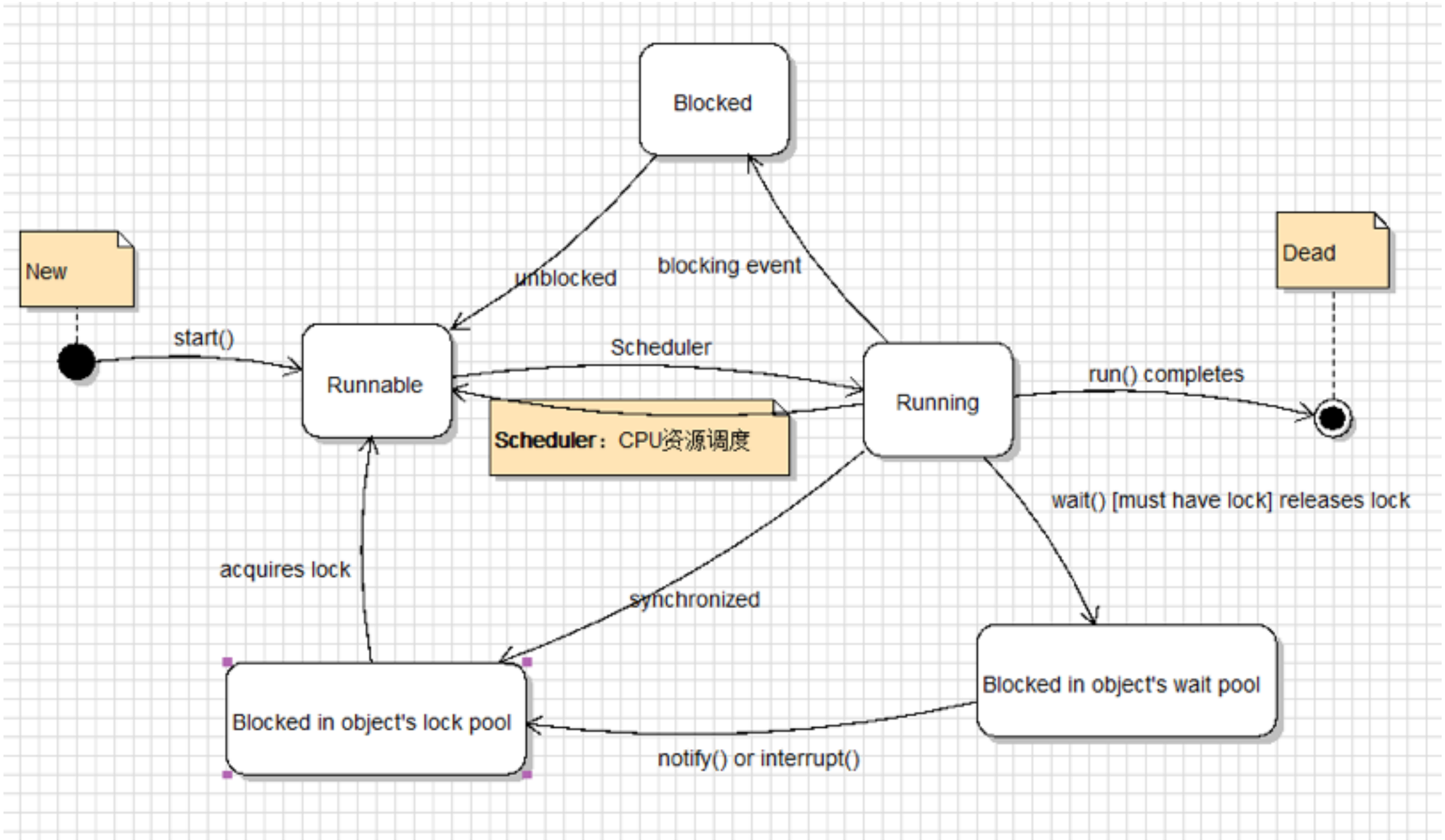
(3)advantages and disadvantages of multithreading



multithread comparision

advantages	disadvantages
Responsiveness *Faster execution* *Lower resource consumption (Apache Http Server)* *Better system utilization* *Simplified sharing and communication* *Parallelisation*	*synchronisation* *Thread crashes a process*

thread status



basic synchronisation methods

(1):synchronized && volatile

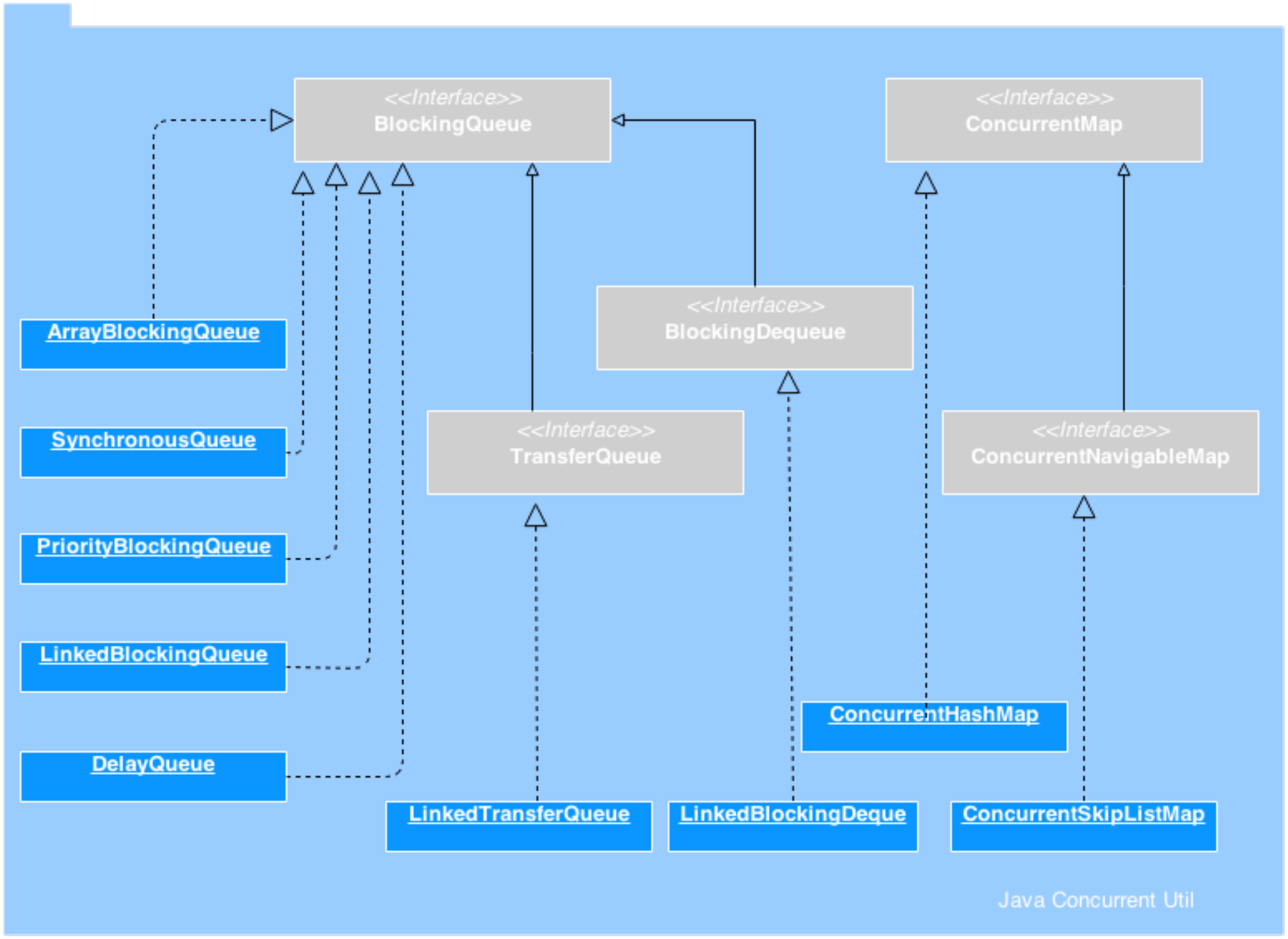
(2):Lock

ReentrantLock
ReentrantReadWriteLock

basic synchronisation helpful classes

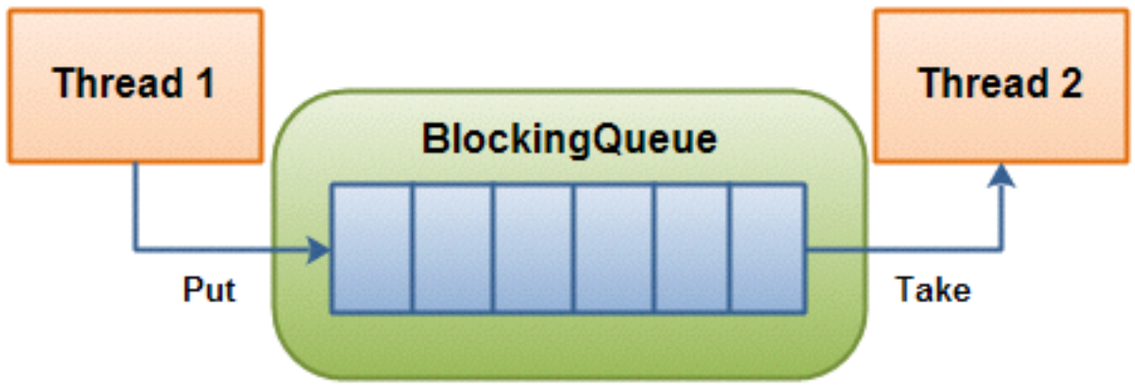
(1):Semaphore
(2):CountDownLatch
(3):CyclicBarrier
(4):Exchanger
(5):Phaser(JDK 1.7)

concurrency collections



BlockingQueue

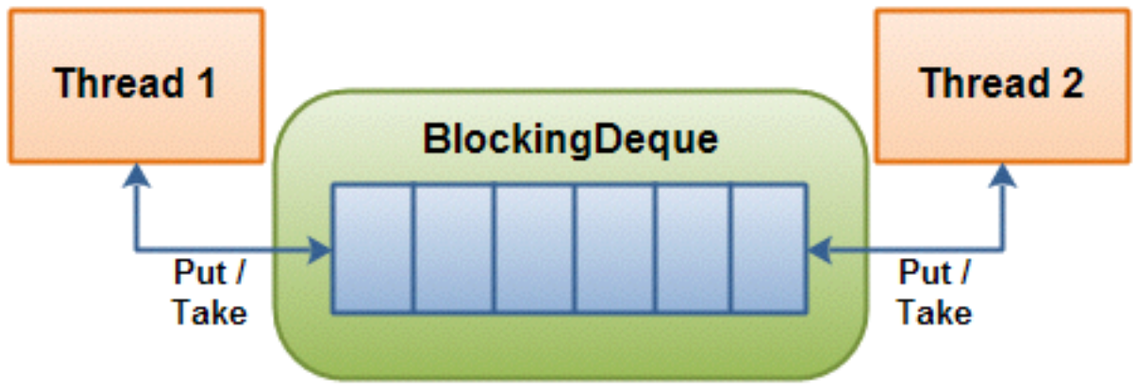
A queue that can be blocked when full or empty



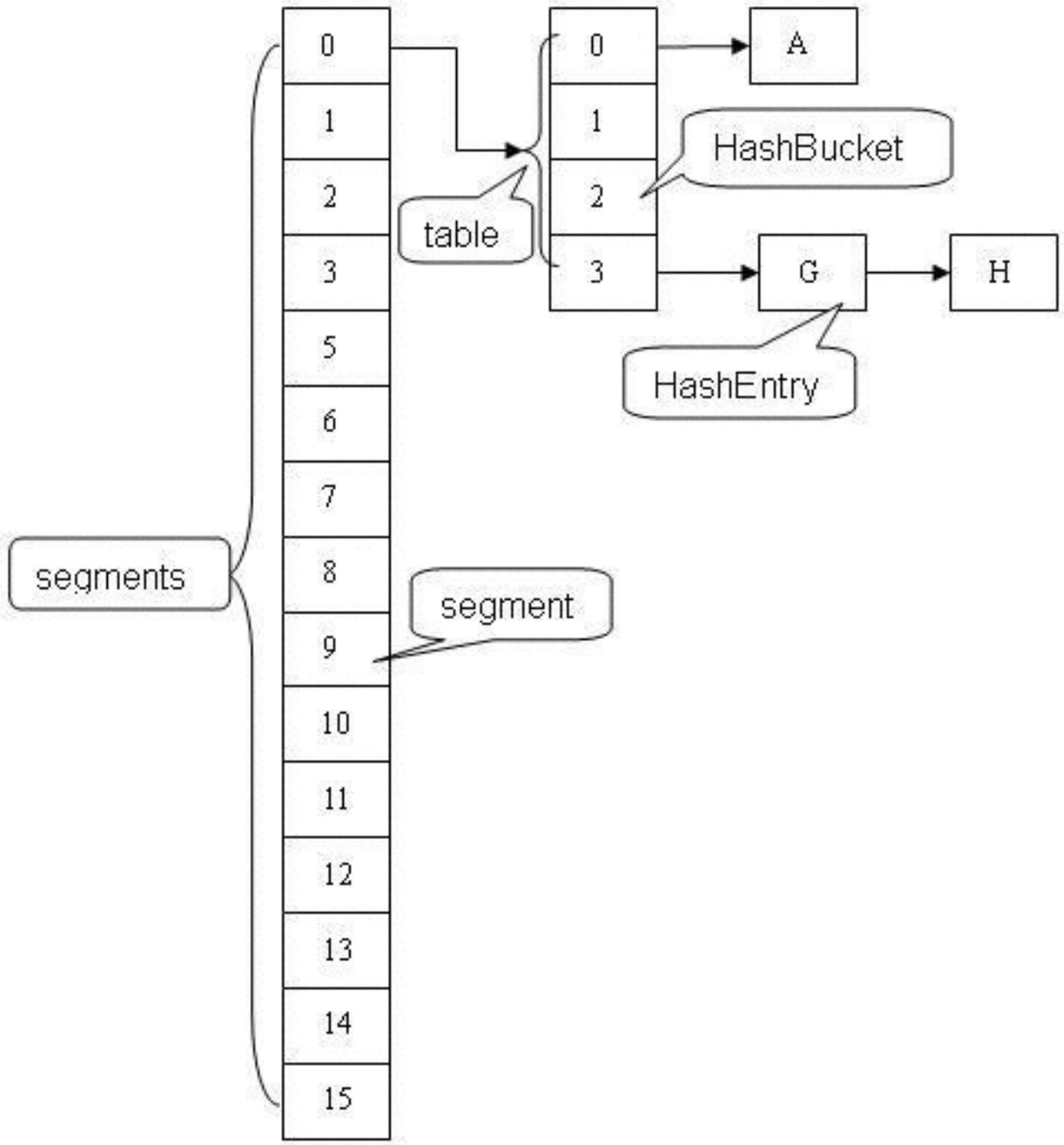
Queue Name	Usage
`ArrayBlockingQueue`	fixed bounded buffer&&elements FIFO
`LinkedBlockingQueue`	fixed bounded buffer&&elements FIFO
`PriorityBlockingQueue`	unbounded buffer&&with priority
`SynchronousQueue`	holding no data&&just channel
`DelayQueue`	used for Cache or close unused connections
`LinkedTransferQueue(JDK1.7)`	blockingqueue+waiting for consumer

BlockingDeque(JDK1.6)

Deque Name	usage
`LinkedBlockingDeque`	threads can put and take from both ends of the deque



ConcurrentHashMap



Segments
HashEntry
HashBucket

Atomic Variable

A small toolkit of classes that support lock-free thread-safe programming on single variables

thread management

benefits of Executor Framework:
(1)no need to write the code about the thread creation, ending and result get(Callable interface);
(2)no need to create the Thread Object;
(3)have better management of the computer resources;

some most used thread pools

ThreadPool	usage
newFixedThreadPool	This executor is suitable for the web AppServer that deny the extra request to protect current user experience.
newSingleThreadExecutor	this executor is used only for one thread to start and can’t be reconfigurable
newCachedThreadPool	This executor is suitable for applications that launch many short-lived tasks.
newScheduledThreadPool	a fixed size thread pool that supports delayed and timed task execution.

concurrency test

1:test for correctness with JUnit

2:test for performance

Classic problems && used in RDS

(1)Producer&&Consumer
(2)Reader&&Writer
(3)Dining Philosophers Problem(Deadlock&&Solutions)

Reference:

thank you