

Striped

[Striped](#) , released in Guava 13.0, is a tool for assigning locks to objects in a configurable, flexible manner.

When you want to associate a lock with an object, the key guarantee you need is that if `key1.equals(key2)` , then the lock associated with `key1` is the same as the lock associated with `key2` .

The crudest way to do this is to associate every key with the same lock, which results in the coarsest synchronization possible. On the other hand, you can associate every distinct key with a different lock, but this requires linear memory consumption and concurrency management for the system of locks itself, as new keys are discovered.

`Striped` allows the programmer to select a number of locks, which are distributed between keys based on their hash code. This allows the programmer to dynamically select a tradeoff between concurrency and memory consumption, while retaining the key invariant that if `key1.equals(key2)` , then `striped.get(key1) == striped.get(key2)` .

`Striped` contains overloads allowing it to be used for [Lock](#) , [ReadWriteLock](#) , or [Semaphore](#) .

Additionally, `Striped` offers variants of each of these with lazily initialized, weakly referenced locks. Essentially, when no thread still references a lock, it can't be waited on or waiting to be released, so it can be safely garbage-collected; this can reduce memory consumption if only a few locks will be in use at any given time.