

```
1  public class HBOLock implements Lock {
2      private static final int LOCAL_MIN_DELAY = ...;
3      private static final int LOCAL_MAX_DELAY = ...;
4      private static final int REMOTE_MIN_DELAY = ...;
5      private static final int REMOTE_MAX_DELAY = ...;
6      private static final int FREE = -1;
7      AtomicInteger state;
8      public HBOLock() {
9          state = new AtomicInteger(FREE);
10     }
11     public void lock() {
12         int myCluster = ThreadID.getCluster();
13         Backoff localBackoff =
14             new Backoff(LOCAL_MIN_DELAY, LOCAL_MAX_DELAY);
15         Backoff remoteBackoff =
16             new Backoff(REMOTE_MIN_DELAY, REMOTE_MAX_DELAY);
17         while (true) {
18             if (state.compareAndSet(FREE, myCluster)) {
19                 return;
20             }
21             int lockState = state.get();
22             if (lockState == myCluster) {
23                 localBackoff.backoff();
24             } else {
25                 remoteBackoff.backoff();
26             }
27         }
28     }
29     public void unlock() {
30         state.set(FREE);
31     }
32 }
```

FIGURE 7.17 The HBOLock class: a hierarchical back-off lock.