```java
public class MatrixAddTask extends RecursiveAction {
  static final int N = ...;
  static final int THRESHOLD = ...;
  Matrix lhs, rhs, sum;
  public MatrixAddTask(Matrix lhs, Matrix rhs, Matrix sum) {
    this.lhs = lhs;
    this.rhs = rhs;
    this.sum = sum;
  }
  public void compute() {
    int n = lhs.getDim();
    if (n <= THRESHOLD) {
      Matrix.add(lhs, rhs, sum);
    } else {
      List<MatrixAddTask> tasks = new ArrayList<>(4);
      for (int i = 0; i < 2; i++) {
        for (int j = 0; j < 2; j++) {
          tasks.add(
                  new MatrixAddTask(
                          lhs.split(i, j),
                          rhs.split(i, j),
                          sum.split(i, j)
                  )
          );
        }
      }
      tasks.stream().forEach((task) -> {
        task.fork();
      });
      tasks.stream().forEach((task) -> {
        task.join();
      });
  }
}
```

FIGURE 16.4 The MatrixAddTask class: fork-join parallel matrix addition.