UNIVERSITÀ
DEGLI STUDI
FIRENZE

Parallel Edit Distance

Firenze 26/04/2020

Edit Distance

The purpose of ”Edit Distance” is to calculate the difference between two strings.

1. ”bar” - ”bir” (replacement of ’a’ with ’i’)
2. ”bir” - ”biro” (insertion of ’o’)

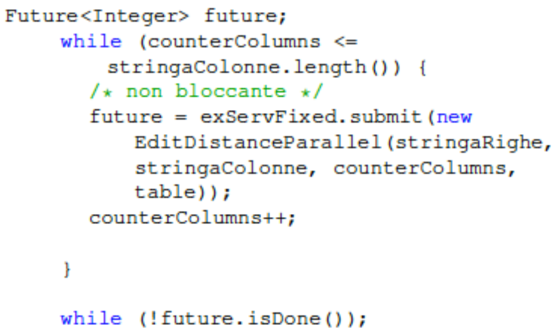
The algorithm in question is used in many types of applications, such as:

- The automatic correction of spelling errors
- The analysis of ”Natural language processing”
- The analysis of nucleotide DNA sequences

Used Technologies

-ThreadPool.
-Executor
-Future

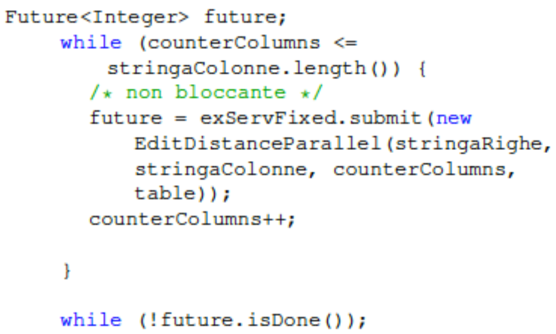
In particular the ThreadPool can be used in different way:
-exServFixed=Executors.newFixedThreadPool(4)
-exServFixed=Executors.newScheduledThreadPool(4)



```
Future<Integer> future;
    while (counterColumns <=
        stringaColonne.length()) {
      /* non bloccante */
      future = exServFixed.submit(new
          EditDistanceParallel(stringaRighe,
          stringaColonne, counterColumns,
          table));
      counterColumns++;

    }

    while (!future.isDone());
```

First implementation

-The algorithm starts from the while loop int the code.

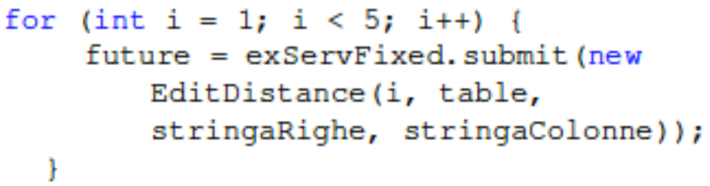


```
Future<Integer> future;
    while (counterColumns <=
        stringaColonne.length()) {
      /* non bloccante */
      future = exServFixed.submit(new
          EditDistanceParallel(stringaRighe,
          stringaColonne, counterColumns,
          table));
      counterColumns++;

    }

    while (!future.isDone());
```

-Each instance manage **one** column or a Diagonal

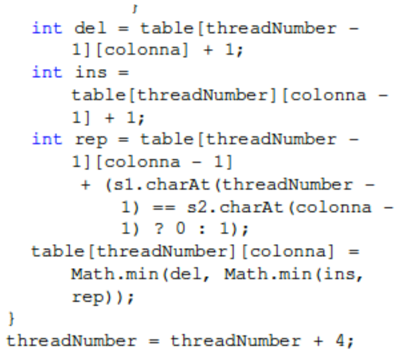
Second implementation

-The algorithm starts from the while code



```
for (int i = 1; i < 5; i++) {
    future = exServFixed.submit(new
        EditDistance(i, table,
        stringaRighe, stringaColonne));
  }
```

-Each instance manage more column or a Diagonal



```
                ,
    int del = table[threadNumber -
        1][colonna] + 1;
    int ins =
        table[threadNumber][colonna -
        1] + 1;
    int rep = table[threadNumber -
        1][colonna - 1]
         + (s1.charAt(threadNumber -
             1) == s2.charAt(colonna -
             1) ? 0 : 1);
    table[threadNumber][colonna] =
        Math.min(del, Math.min(ins,
        rep));
  }
  threadNumber = threadNumber + 4;
```

-At the end of the computation each threads add 4 to the current diagonal column which need to be computed.