Design fork join pool

<https://dzone.com/articles/a-look-at-forkjoinpools>

**MY Design**

There will be a Pool class with the following:  
Members:

* A queue of tasks
* Number of processor cores

Constructor: takes the number of processor cores

* Pool(numCores)

Methods:  
execute(task) : TaskResult={

join(fork(Task))

}

Pool(numProcessor):Pool

fork(Task):List[Task] //devide into subtasks

join(List[Task]): Task //combines result of multiple threads into

Each processor core has it’s own

. ForkJoin is based on parallel computing, where a problem gets divided into sub-problems until the sub-problems are simple enough to solve simultaneously in separate threads, after which the results are aggregated.

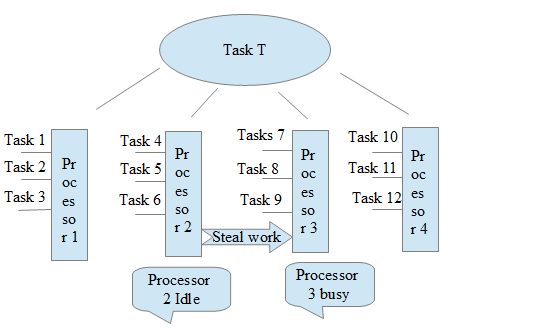
The algorithm that's been used in ForkJoin implementations is the 'Work-Stealing Algorithm' to ensure that no CPU is idle.

Let's take a look at work-stealing algorithm.

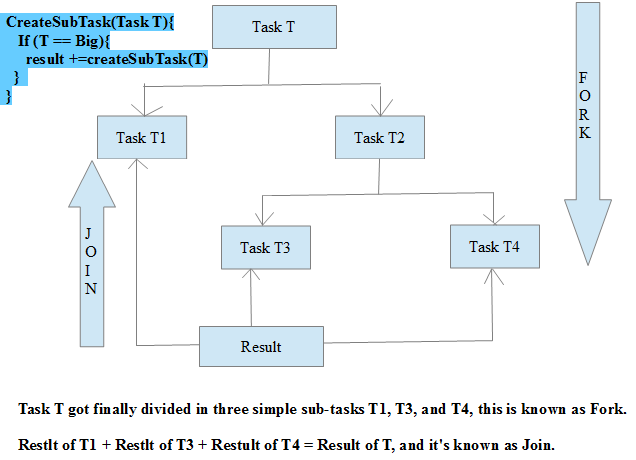
Suppose there are four processors on a system and Task T has been divided into 12 subtasks, shown below, from T1 to T12.

Each processor has three subtasks. But suppose Processor 3 is too busy with work and Processor 2 is idle—it either finished all the tasks or it's in a waiting state. Then, Processor 2 will ask to Processor 3 if it needs help and takes a percentage of the tasks. So, Processor 2 is 'stealing the work' from Processor 3.

The below snapshot depicts this process:



Now let's look at the snapshot below to understand the ForkJoin framework:



So above snapshot shows the recursive approach to divide the task until subtask is a simple task.

We create an instance of ForkJoinPool as below -

 ForkJoinPool pool = new ForkJoinPool(parallelism);

Here, parallelism is the target parallelism level (number of processors), which can be seen below:

 Runtime.getRuntime().availableProcessors(); // returns 4 on my system

To support parallelism for collections using parallel streams, a common ForkJoinPool is used internally.

We can get a common pool using the ForkJoin static method below:

 ForkJoinPool commonPool = ForkJoinPool.commonPool();