

TD 1 - Java concurrency: parallel algorithms

To set up for the practical exercises open your favorite IDE (e.g., Eclipse, Netbeans) and create a Java project. Create two packages called `polytech.tours.di.parallel.td1.exo1` and `polytech.tours.di.parallel.td1.exo2`. You're now good to go.

1 Estimating the value of π

The value of π can be calculated in a number of ways. Consider the approach below:

- Inscribe a circle in a 1x1 square
- Randomly generate points in the square
- Determine the number of points in the square that are also in the circle
- Let p be the number of points in the circle divided by the number of points in the square, then $\pi \approx 4 \times p$

Figure 1 illustrates the approach. Note that the quality of the approximation increases with the number of generated points.

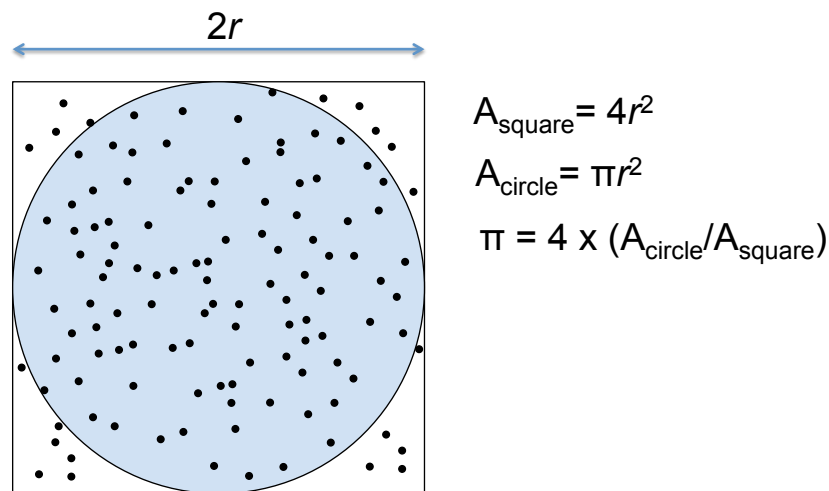


Figure 1: Approximating π

It is possible to solve this problem using what is known as an embarrassingly parallel solution; that is a solution which is computationally intensive and has minimal communication and minimal I/O. The objective of this exercise is to come up with such solution.

1. Propose a parallel algorithm for this problem
 - What decomposition strategy better fits the problem?
 - What is the task dependency graph of your algorithm?
 - What is the task interaction graph of your algorithm?
2. Propose a Java implementation for your algorithm