

PiEstimator

Pi Estimator

The value of PI can be calculated in a number of ways. In this example, we are estimating PI using following way:

- Each task executes locally its portion of the loop a number of times.

```
{{{  
iterations = 10000  
circle_count = 0
```

```
do j = 1, iterations  
generate 2 random numbers between 0 and 1  
xcoordinate = random1  
ycoordinate = random2  
if (xcoordinate, ycoordinate) inside circle  
then circle_count = circle_count + 1  
end do
```

```
PI = 4.0*circle_count/iterations  
}}}
```

- One task acts as master and collects the results through the BSP communication interface.

```
PI = pi_sum / n_processes
```

1) Each process computes the value of Pi locally, and 2) sends it to master task using send() function. Then, 3) the master task can receive the messages using sync() function. Finally, we can calculate the average value of sum of PI values from each peers as below:

```

@Override
public void bsp(
    BSPPeer<NullWritable, NullWritable, Text, DoubleWritable> peer)
    throws IOException, SyncException, InterruptedException {

    int in = 0, out = 0;
    for (int i = 0; i < iterations; i++) {
        double x = 2.0 * Math.random() - 1.0, y = 2.0 * Math.random() - 1.0;
        if ((Math.sqrt(x * x + y * y) < 1.0)) {
            in++;
        } else {
            out++;
        }
    }

    double data = 4.0 * (double) in / (double) iterations;
    DoubleMessage estimate = new DoubleMessage(peer.getPeerName(), data);

    peer.send(masterTask, estimate);
    peer.sync();
}

@Override
public void setup(
    BSPPeer<NullWritable, NullWritable, Text, DoubleWritable> peer)
    throws IOException {
    // Choose one as a master
    this.masterTask = peer.getPeerName(peer.getNumPeers() / 2);
}

public void cleanup(
    BSPPeer<NullWritable, NullWritable, Text, DoubleWritable> peer)
    throws IOException {
    if (peer.getPeerName().equals(masterTask)) {
        double pi = 0.0;
        int numPeers = peer.getNumCurrentMessages();
        DoubleMessage received;
        while ((received = (DoubleMessage) peer.getCurrentMessage()) != null) {
            pi += received.getData();
        }

        pi = pi / numPeers;
        peer
            .write(new Text("Estimated value of PI is"), new DoubleWritable(pi));
    }
}
}

```