

Software architecture documentation

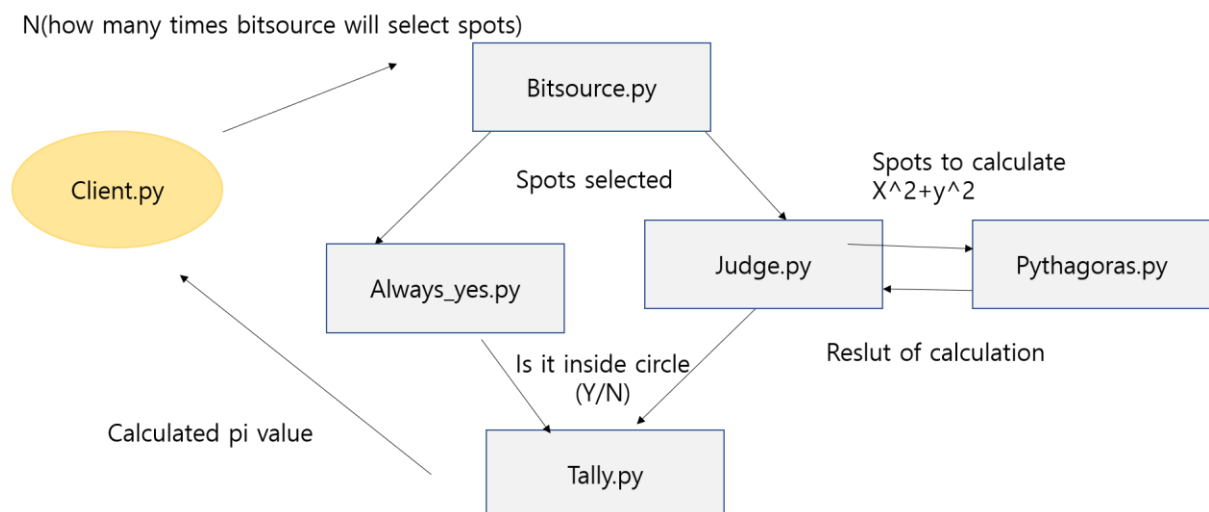
20192698 Sim WonJun , CSE , Soongsil University

1. Overview

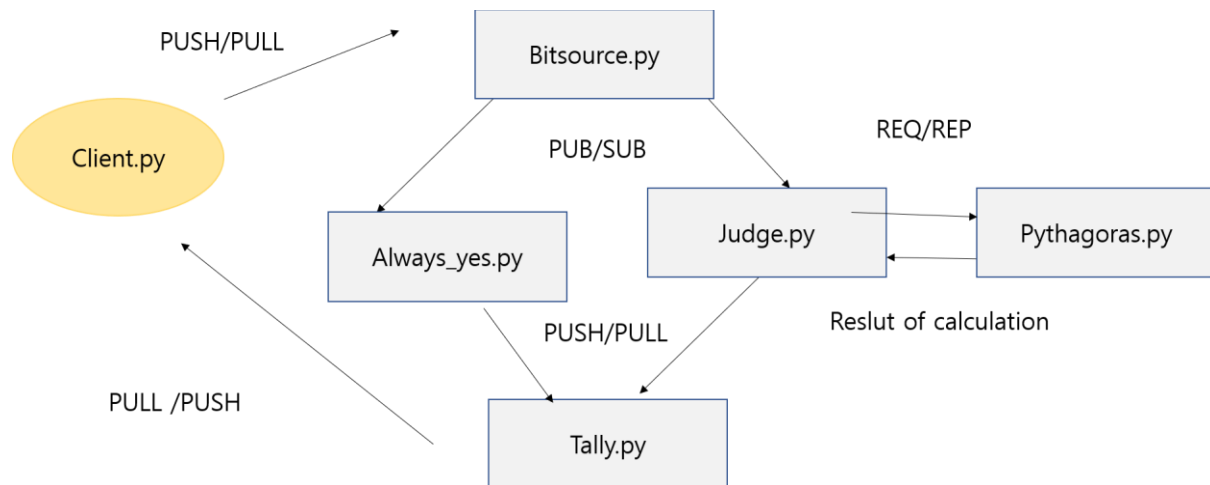
This software has its purpose to calculate $\pi(\pi)$ by using the space of square and circle inside that square. If we calculate the space of square, we use the method of n^2 ($n = \text{width}$). When calculating that of circle, we use $r^2 \cdot \pi$. If n stands to $2r$, which means the whole circle is in one square, square's space is $4r^2$. Therefore we can calculate the value of π by dividing $4 \times (\text{space of circle})$ by space of square ($4r^2$). This program calculates it by randomly selecting spots of square, and calculate the value of $x^2 + y^2$. By comparing this value with the unit value (2^{64}), we can see if that randomly-selected spot is inside or outside of circle, and if it's inside, calculate the value of π . If we calculate very many times this value, we can get closer and closer to the value of π .

This program gets input from user about how many times will we select the spots randomly. That works in the client.py. On other computers, bitsource.py selects spots randomly, judge.py judges if it is inside the circle using Pythagoras.py running on the other computer. Always_yes.py returns the spot that is always inside the circle. And finally tally sends to client the value of π calculated in those processes.

Image of how this program works.



2. What kind of protocol this program uses at each processes



All the communications between processes(computers) are using zero message queue.

Client pushes value of N to bitsource.py and bitsource.py publishes spots it randomly selected. Always_yes.py and Judge.py subscribes those spots and Judge.py requests calculation to Pythagoras.py. And finally they pushes Y/N about whether or not those spots are inside the circle. Tail.py pulls those data and pushes json data to Client the data of decision and p,q for calculation of pi value at client.py.

3. Process View

When client.py starts at the computer, it takes value of N from your in put. And sleeps 32seconds before all the other processes are ready. Than bitsource.py gets that value(pull) and loop until it selects N-spots. It publishes those data of spots and than Judge/Always_yes subscribe those data. If judge.py subscribes data, it sends to Pythagoras.py to calculate x^2+y^2 value. And than judge if it is inside the circle. Finally, Tally gets data from Always_yes and Judge.py whether it is inside or outside of circle, and then if the spot is inside the circle, it sends to Client.py the list of decision, p, y value to calculate pi value inside Client.py. Client.py prints it on the screen.

4. Notion

Before starting this program, you shall install ZMQ with "pip install pyzmq"

Before these programs start, they wait for 30seconds until all programs are opened in computers. Also it waits for 15seconds until all programs are shut down.

You shall write your ip address and port number to use in the given source code.