Semih Yılmaz 2020400171

1. Design Choices

Our project has two source code files. They are "master.py" and "slave.py". "master.py" has the source code of the main control room and "slave.py" has the source code of machines. Main control room utilizes machine class while reading the given input file. Machine class has fields to keep necessary information about the parent and children machines, initial operation and initial product if it is a leaf node. After input handling, the master process spawns slave processes.

There will be one more slave processes than the number of machines and the extra one which has rank 0 is used to broadcast threshold, number of cycles and wear factors to machines. The information which is special for each machine is passed one by one with different tags via blocking communication. The information is the ids of the children, initial operation, initial product and id of the parent.

The machine without any parent is selected as the terminal. Leaf nodes start operations and send their products to their parents by blocking communication. Tag=1 is used for product transfers among machines. The terminal machine waits their children for products. When it receives all of them, it adds them and send the final product to the main control room via inter-communicator. Tag=1 is used for final product transfers to the main control room.

The machines inform main control room for required maintenances. The maintenance logs are send via inter-communicator using non-blocking communication and tag=2. Machines keep track of the number of maintenance they needed. hen they finish all production cycles, they send the number to the master process by blocking communication with tag =3. Then the main control room takes maintenance logs as many as the number.

2. Advantages of Parallel Programming

Thanks to parallel programming, we can utilize multiple cores of the processor. The tasks performed by different machines run on different cores simultaneously. If we did not use parallel programming, the program would be executed on just one core. Parallel programming enables us to increase performance by parallelism. Also, it provides scalability.

3. Example Input / Output

Example input:

```
9
14
2 5 6 1 8
32
2 1 chop
3 1 trim
4 2 split
5 2 reverse
6 4 split
7 4 reverse
8 6 enhance
9 6 trim
XAX
SF
RD
AN
TANA
```

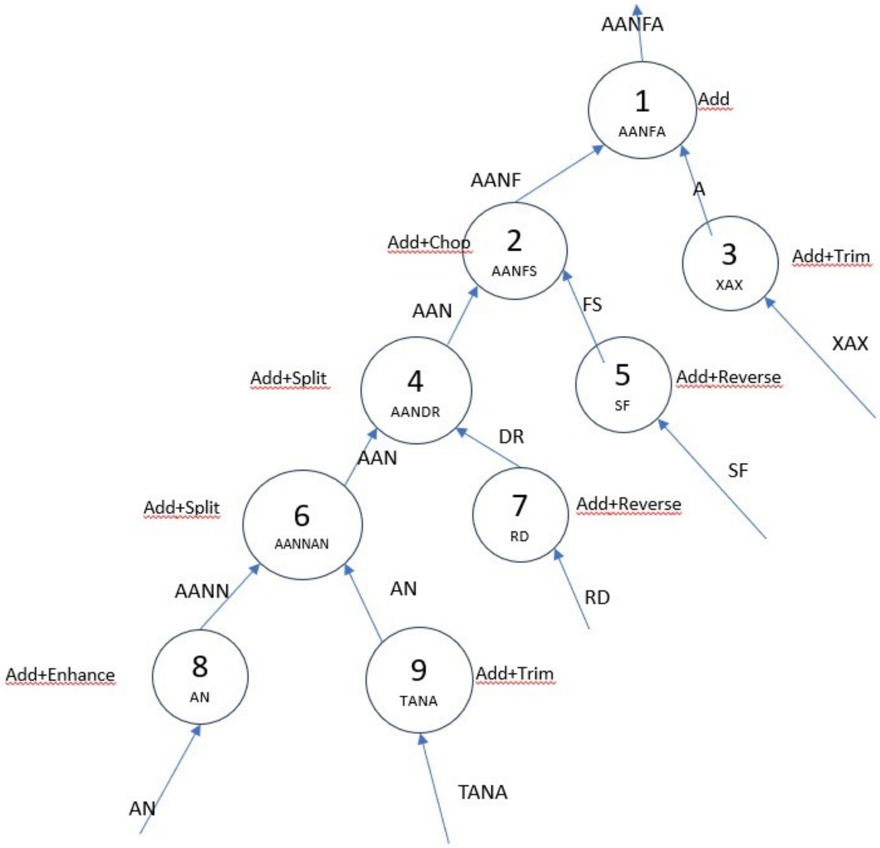

Figure 1: The first production cycle for the example input.

Output of the example input:

```
AANFA
AAANARSFFXAX
AAAANNA
AANSXAX
AAADFSSA
AAAANATXAX
AANFA
AAANARSFFXAX
AAAANNA
AANSXAX
AAADFSSA
AAAANATXAX
AANFA
AAANARSFFXAX
2-8-6
2-8-12
3-25-12
4-2-6
4-2-12
5-20-11
6-2-6
6-2-12
7-20-11
8-6-6
8-6-12
9-25-12
```

4. Challenges of Digital Twin Implementations

First of all, implementing digital twin technology would mean a serious investment for large scale manufacturers in the short run. Therefore, there has to be risk takers to implement it.

Secondly, security is the main issue for digital twins since physically stealing is not as easy as stealing data from a database. Moreover, bugs and errors may cost a fortune with digital twin method so it would be better to use it with high caution. Also for processes that change in size and operations, maintenance may be costly depending on the usage. Additionally, regulations from international organizations and governments may be difficult to satisfy in terms of security.

On the broader sense, if digital twin technology doesn't get recognized and widely used by high tech companies, using them as a smaller one would get hurt because they would not be able to work as easily with high tech firm projects.