

Summary of the analysis of Extendedbinomialtree class

GROUP2: André CHHOA, Tarik HAJJAM, TRIKI Anis, Zhun XIANG

We reviewed the tree framework and tested the extendedbinomialtree for equity options. We have noticed some anomalies which may help us to improve the efficiency of the extendedbinomialtree class.

Firstly, we priced a put with the following methods:

- ExtendedJarrowRudd_2
- ExtendedCoxRossRubinstein_2
- ExtendedAdditiveEQPBinoialTree_2
- ExtendedTrigeorgis_2
- ExtendedTian_2
- ExtendedLeisenReimer_2
- ExtendedJoshi4_2

We have found that the function driftStep in the class ExtendedBinomialTree_2 were called an enormous amount of time :

```
//time dependent drift per step
Real driftStep(Time driftTime) const {
    return this->treeProcess_>drift(driftTime, x0_) * dt_;
}
```

Here is a resume for each method:

| Method | DriftStep Called Times |
|------------------------|------------------------|
| Jarrow_Rudd | 320K |
| LeisenReimer | 640K |
| Joshi | 640K |
| Tian | 320K |
| CoxRossRubinstein | 4 |
| AdditiveEQPBinoialTree | 1280K |

As showed in the previous table, except for the CoxRossRubinstein method, the function driftStep was called millions times when timeteps = 801. Additionally, the driftStep is called repeatedly for the same drifttime in the same method.

So, to sum up, creating a cache zone for the function driftStep is a potential solution for us to improve the performance. We may have some possible solutions:

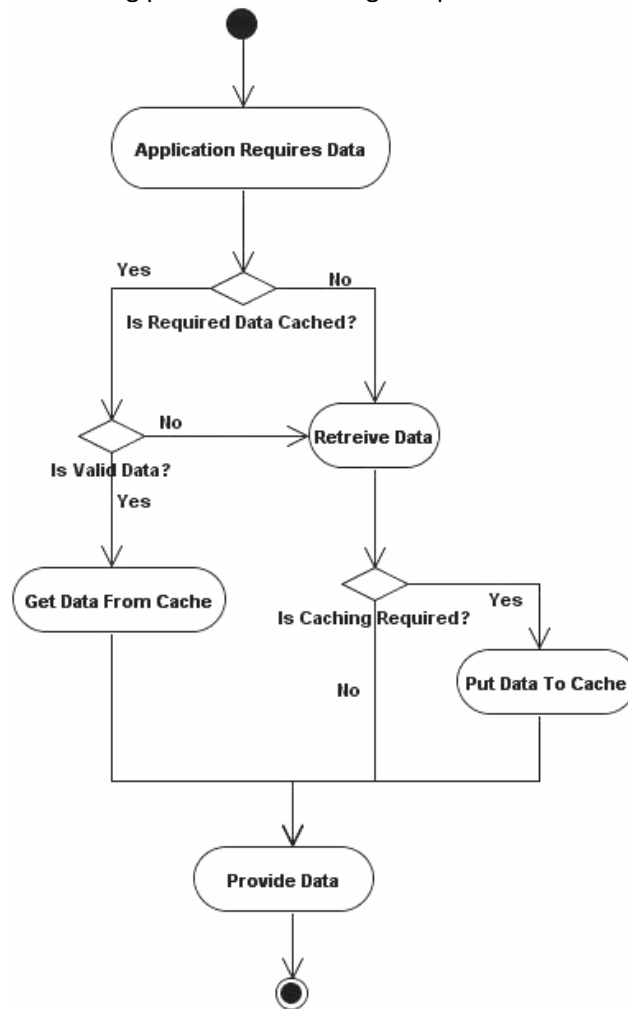
→ The first solution is to create a dictionary for the driftStep and just initialize it in the constructor.

Advantage: easy to implement

Drawback: once the dictionary is initialized, we cannot change values in the dictionary anymore.

→ A second solution is to use a design pattern. The caching design pattern is a very interesting one.

You can find below the caching process when using this pattern:



If we take this solution, we will have more flexibility and the DriftStep function will be called only when necessary.

We think that it is the best solution for us and it is the one we intend to implement.

We would like to have your feedback on that solution before starting the implementation.

Bibliography: Romania O. Caching patterns and implementation[J]. Leonardo Journal of Sciences, 2006 (8): 61-76.