Clustering Coefficient

군집계수 분석

조정근
wjdrms1388@naver.com
010.5006.1388

CONTENTS

- 프로젝트 목표

  대용량 그래프의 군집계수 (Clustering Coefficient) 분석을 위한 효율적인 분산 알고리즘을 설계, 구현 및 실험한다.

  프로젝트는 다음과 같은 세부 과업으로 구성된다

  - Task 1) 그래프의 중복 edge 및 self-loop 제거 (Hadoop)
  - Task 2) 각 node의 degree 구하기 (Hadoop)
  - Task 3) 각 node u 마다, u를 포함하는 삼각형의 수 구하기 (Spark)
  - Task 4) 각 node 마다, 군집계수 구하기 (Spark)

Task 1. Simple Graph (Hadoop)

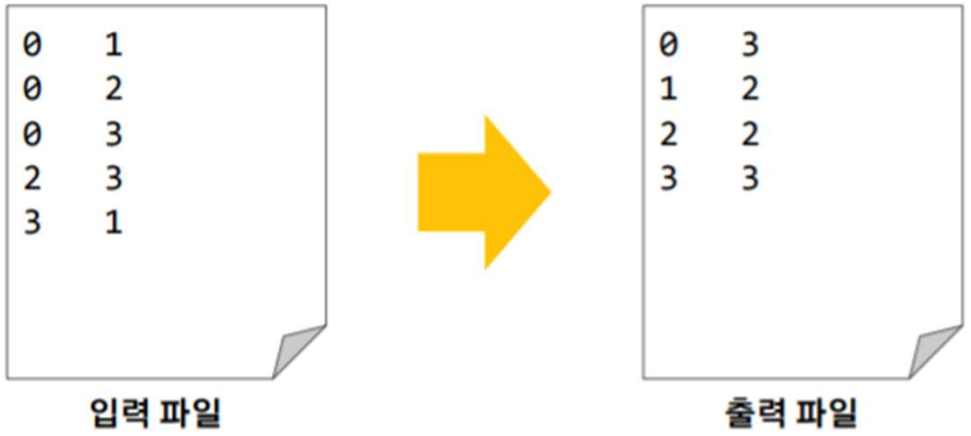
그래프가 edge list file로 주어졌을 때, 중복 edge와 self-loop을 제거하여 simple graph를 생성하는 Hadoop 프로그램 작성

- Edge list file은 한 줄에 하나씩 text로 저장됨
- 각 node는 음이 아닌 정수
- 각 줄의 edge는 두 node가 탭(＼t) 또는 공백으로 구분되어 표현됨
  - 예) 23 41 : node 23과 node 41을 연결하는 edge
- edge(u, v) 와 edge(v, u) 는 동일한 것으로 간주함
- Self-loop은 양 끝 노드가 동일한 edge
  - 예) (u, u)
- 입력 파일 경로를 args[0]으로, 출력 파일 경로를 args[1]로 부터 받음

Task 2. Degree Computation (Hadoop)

Task 1의 결과 그래프가 주어졌을 때,
각 node의 degree(이웃 node 수)를 구하는 프로그램 작성

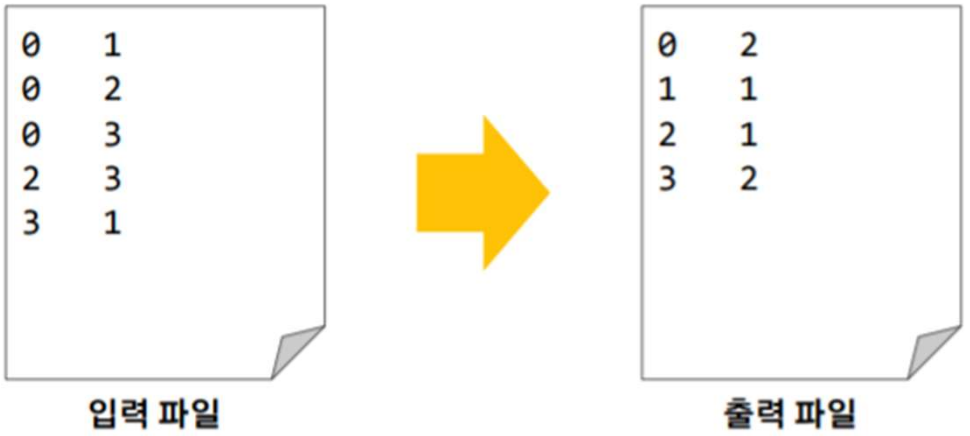
- 입력 파일 경로를 args[0]으로, 출력 파일 경로를 args[1]로 부터 받음
- 출력 파일의 각 줄에는 node u와 degree d(u)를 탭으로 구분하여 text로 저장



Task 3. Triangle Counting (Spark)

Task 1의 결과 그래프가 주어졌을 때,
각 node u 마다 u를 포함하는 삼각형의 수 t(u)를 계산하여 출력하는 프로그램 작성

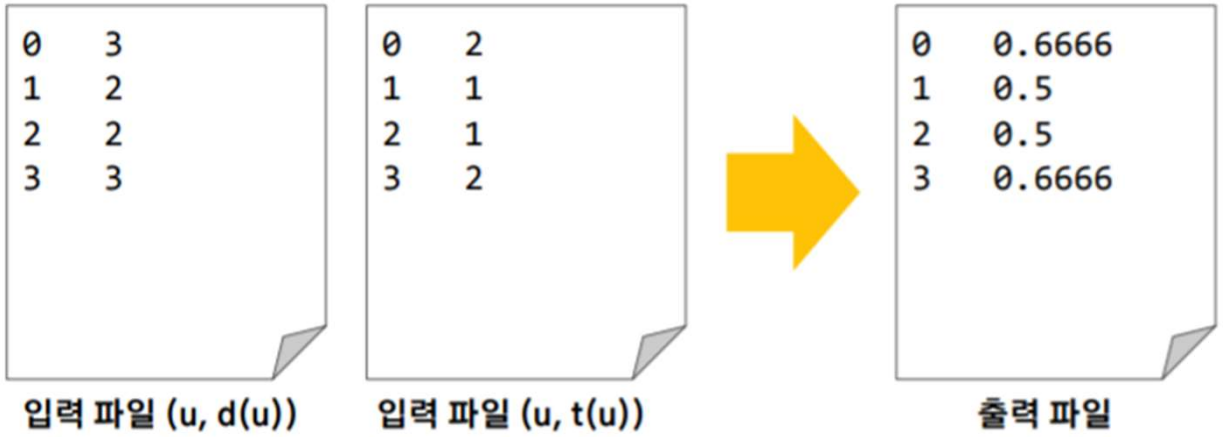
- 입력 파일 경로를 args[0]으로, 출력 파일 경로를 args[1]로 부터 받음
- 출력 파일의 각 줄에는 node u와 삼각형의 수 t(u)를 탭으로 구분하여 text로 저장



Task 4. Clustering Coefficient (Spark)

Task 2 의 결과와 Task 3 의 결과가 주어졌을 때,
각 node u 의 군집계수 cc(u)를 계산하여 출력하는 프로그램 작성

- 입력 파일 경로를 args[0], args[1]로, 출력 파일 경로를 args[2]로 부터 받음
- 출력 파일의 각 줄에는 node u 와 군집계수 cc(u)를 탭으로 구분하여 text로 저장



세부사항

First Second Third

데이터 전처리

데이터 처리를 위해서 데이터를 설명하는 주석을 제거한다.

데이터는 약 1GB의 무방향 그래프(Undirected Graph)를 사용하였다.

```
# Directed graph (each
# Directed LiveJournal
# Nodes: 4847571 Edges:
# FromNodeId    ToNodeI
0   1
0   2
0   3
0   4
0   5
0   6
...
```

```
0   1
0   2
0   3
0   4
0   5
0   6
0   7
0   8
0   9
0   10
...
```

세부사항

First Second Third

Task 1

IntPairWritable Class를 구현하여 edge 중복 제거에 사용했다.

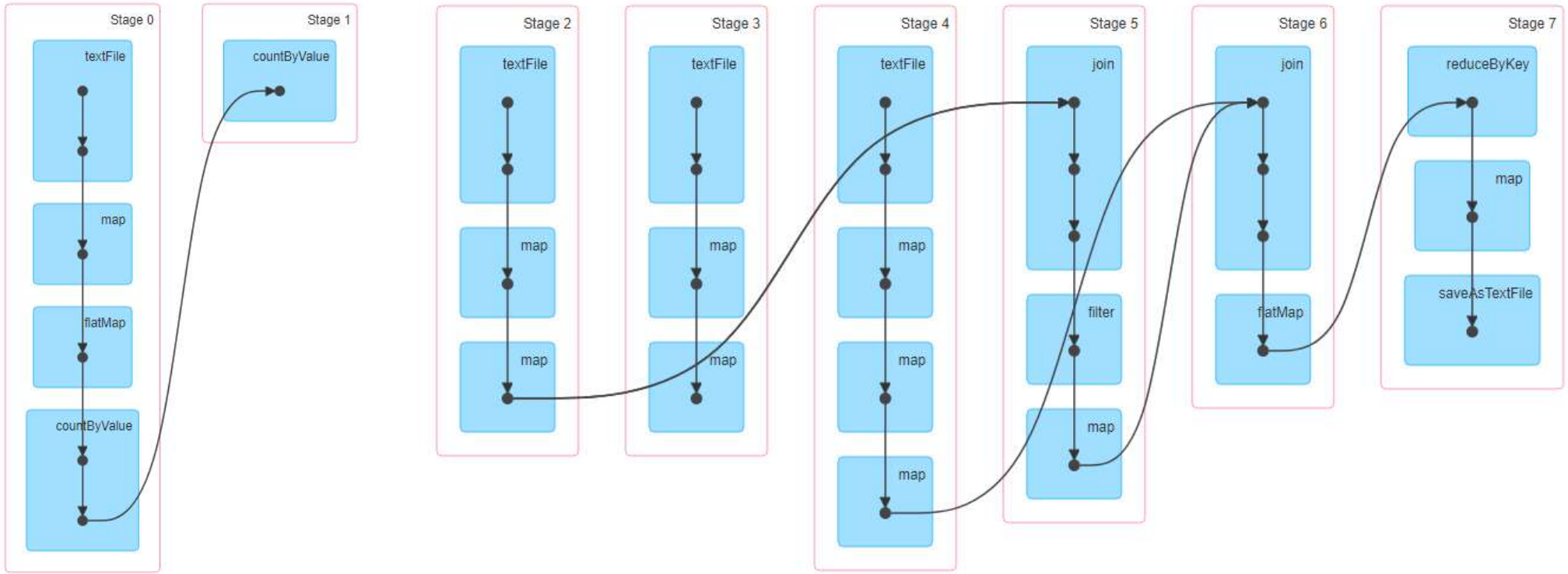
```java
public class IntPairWritable implements WritableComparable<IntPairWritable> {
    int u, v;

    public IntPairWritable() {
        this.u = 0;
        this.v = 0;
    }

    public IntPairWritable(int u, int v) {
        this.u = u;
        this.v = v;
    }

    public void set(int u, int v) {
        this.u = u ;
        this.v = v;
    }

    public void write(DataOutput out) throws IOException {
        out.writeInt(u);
        out.writeInt(v);
    }

    public void readFields(DataInput in) throws IOException {
        u = in.readInt();
        v = in.readInt();
    }

    public int compareTo(IntPairWritable o) {
        if(this.u != o.u) return Integer.compare(this.u, o.u);
        return Integer.compare(this.v, o.v);
    }

    @Override
    public String toString() {
        return u + "\t" + v;
    }

}
```

세부사항

First Second Third

Task 3 - DAG

Spark를 통해서 데이터가 처리 될 때의 워크플로우를 보여준다.



세부사항

First Second Third

Task 4 - DAG

Spark를 통해서 데이터가 처리 될 때의 워크플로우를 보여준다.