Large Graph Mining: Power Tools and a Practitioner’s guide

Task 8: hadoop and Tera/Peta byte graphs

Faloutsos, Miller, Tsourakakis

CMU

Outline

- Introduction – Motivation
- Task 1: Node importance
- Task 2: Community detection
- Task 3: Recommendations
- Task 4: Connection sub-graphs
- Task 5: Mining graphs over time
- Task 6: Virus/influence propagation
- Task 7: Spectral graph theory
- **Task 8: Tera/peta graph mining: hadoop**
- Observations – patterns of real graphs
- Conclusions

Scalability

- How about if graph/tensor does not fit in core?
- How about handling huge graphs?

Scalability

- How about if graph/tensor does not fit in core?
- [‘MET’: Kolda, Sun, ICMD’08, best paper award]
- How about handling huge graphs?

Scalability

- Google: > 450,000 processors in clusters of ~2000 processors each [Barroso, Dean, Hölzle, *“Web Search for a Planet: The Google Cluster Architecture”* IEEE Micro 2003]
- Yahoo: 5Pb of data [Fayyad, KDD’07]
- Problem: machine failures, on a daily basis
- How to parallelize data mining tasks, then?

Scalability

- Google: > 450,000 processors in clusters of ~2000 processors each [Barroso, Dean, Hölzle, *“Web Search for a Planet: The Google Cluster Architecture”* IEEE Micro 2003]
- Yahoo: 5Pb of data [Fayyad, KDD’07]
- Problem: machine failures, on a daily basis
- How to parallelize data mining tasks, then?
- A: map/reduce – hadoop (open-source clone) http://hadoop.apache.org/

2’ intro to hadoop

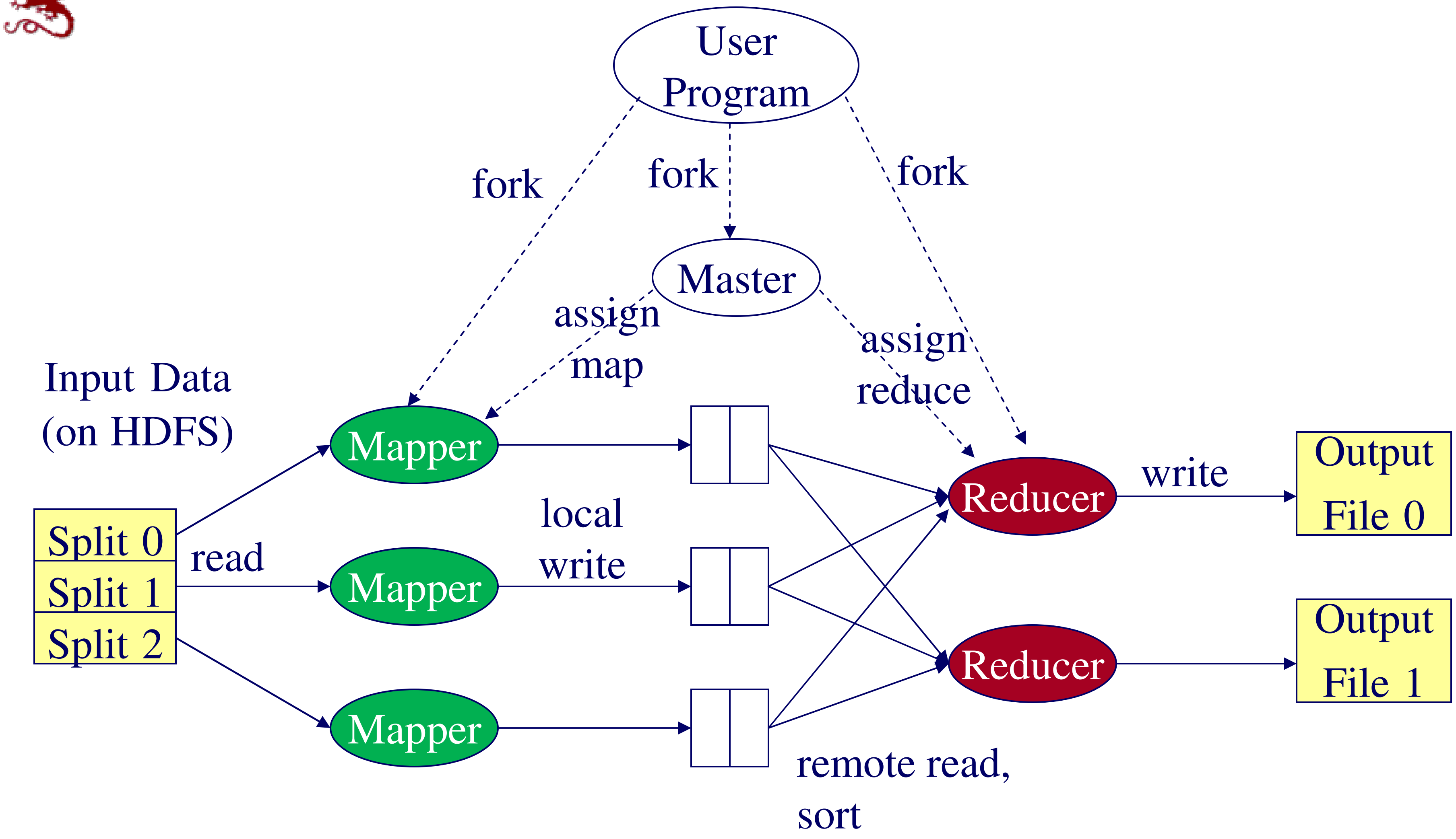
- master-slave architecture; n-way replication (default n=3)
- ‘group by’ of SQL (in parallel, fault-tolerant way)
- e.g, find histogram of word frequency
  - compute local histograms
  - then merge into global histogram

```
select course-id, count(*)
from ENROLLMENT
group by course-id
```

2’ intro to hadoop

- master-slave architecture; n-way replication (default n=3)
- ‘group by’ of SQL (in parallel, fault-tolerant way)
- e.g, find histogram of word frequency
  - compute local histograms
  - then merge into global histogram

```
select course-id, count(*)          reduce
from ENROLLMENT
group by course-id                  map
```



By default: 3-way replication;
Late/dead machines: ignored, **transparently** (!)

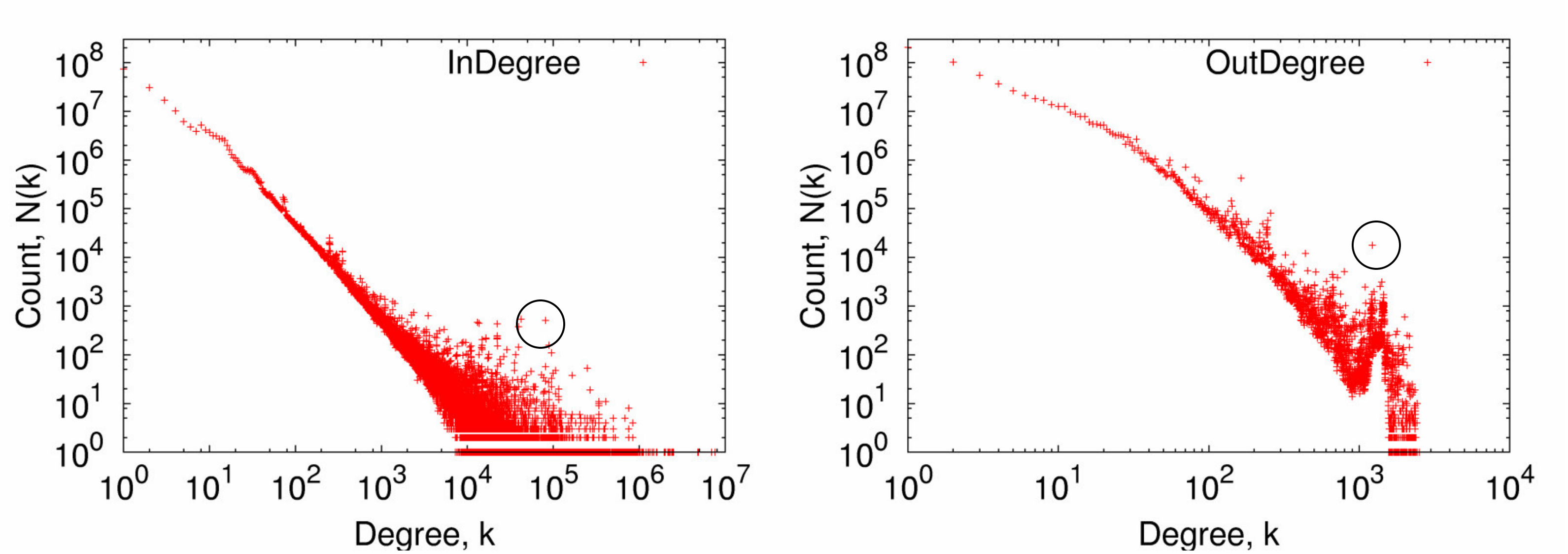
D.I.S.C.

- ‘Data Intensive Scientific Computing’ [R. Bryant, CMU]
    - ‘big data’
    - www.cs.cmu.edu/~bryant/pubdir/cmu-cs-07-128.pdf

Analysis of a large graph

~200Gb (Yahoo crawl) - Degree Distribution:

- in 12 minutes with 50 machines
- Many link spams at out-degree 1200



Conclusions

- Hadoop: promising architecture for Tera/Peta scale graph mining

Resources:

- http://hadoop.apache.org/core/
- http://hadoop.apache.org/pig/

  Higher-level language for data processing

References

- Jeffrey Dean and Sanjay Ghemawat, *MapReduce: Simplified Data Processing on Large Clusters,* OSDI'04
- Christopher Olston, Benjamin Reed, Utkarsh Srivastava, Ravi Kumar, Andrew Tomkins: *Pig latin: a not-so-foreign language for data processing*. SIGMOD 2008: 1099-1110