CS492 Special Topics in Computer Science
Distributed Algorithms and Systems

Lecture 3
MapReduce

- Simplified data processing on large clusters
  - A programming model and an associated implementation for processing and generating large data sets
- Inspired by “map” and “reduce” primitives present in Lisp and many other functional languages
- Simple and powerful interface
  - that enables automatic paralization and distribution of large-scale computations
Map and Reduce

- **Map**
  - map \(<k_1, v_1> \rightarrow \text{list (k_2, v_2)}\)
  - example
    - \(k_1 = \text{document, } v_1 = \text{contents}\)
    - \(k_2 = \text{words, } v_2 = 1\)

- **Reduce**
  - reduce \((k_2, \text{list(v_2)}) \rightarrow \text{list (v_2)}\)
    - \(k_2 = \text{words, } v_2 = 1\)
    - \(\text{list (v_2) = total count of v_2 (or k_2)}\)
MapReduce Execution Overview

Figure 1: Execution overview
Examples

- Distributed grep
- Count of URL access frequency
- Reverse web-link graph
- Term-vector per host
**Pig**

- platform for analyzing large data sets that consists of a high-level language for expressing data analysis programs, coupled with infrastructure for evaluating these programs. The salient property of Pig programs is that their structure is amenable to substantial parallelization, which in turns enables them to handle very large data sets

**Dryad**

- Distributed data-parallel programs from sequential building blocks (EuroSys 2007)