Lesson 3

Local Aggregation

Capture the fort

Wholeness of the Lesson

The communication cost is the single most factor that determines the efficiency of a MapReduce job. The best way to achieve better efficiency is by carefully planning and organizing the data so that only the least amount of data is shuffled across the network.

“Capturing the fort” is the very first course of action.

LOCAL AGGREGATION

An improvement on the basic wordcount algorithm is shown next. Only mapper is modified in this example.

An associative array is introduced inside the mapper to tally up term counts within a single input-split. Instead of emitting a key-value pair for each term in the input-split, this version emits a key-value pair for each unique term in the input-split.

Given the fact that some words appear frequently within an input-split, this can yield substantial savings in the number of intermediate key-value pairs emitted.

Example

Assume that an input-split contains the word “science” 10 times.

The wordcount algorithm will emit the pair (“science”, 1) ten times.

The “modified” algorithm shown next will emit only one pair: (“science”, 10).

LOCAL AGGREGATION

```
class Mapper
    method initialize()
        H =  new AssociativeArray()
    method map(docid a; doc d)
        for all term t in record r do
            H{t} =  H{t} + 1 .
    method close()
        for all term t in H do
            Emit(t, H{t})
```