**15-440 Map Reduce**

**Tyler Healy – Justin Greet**

**System Administration**

**Starting the Map Reduce System**

Our Map Reduce package contains many Java files that work together to allow problems following the Map-Reduce paradigm to be run in parallel across multiple systems. However, in order for a machine to be a participant in the computation of one of these problems, it must have the package installed on a machine. The package can be installed by copying the files of the package into a directory on the participating system.

Once the package is installed, a system can be prepared for participation by doing the following:

**15-440 Map Reduce**

**Tyler Healy – Justin Greet**

**Reference Manual**

**Mapper and Reducer**

A Map-Reduce program must implement these two interfaces: Mapper and Reducer. These form the core of the Map-Reduce job to be completed.

**Mapper**

The mapper takes a set of input key/value pairs and produces a set of intermediate key/value pairs.

The number of mappers to be used for a given job is defined in the configuration file. The input is split amongst the mappers, and each mapper takes a partition of the input and calls the map function on each input record.

The map reduce takes the following inputs:

* Input key
* Input value
* OutputCollecter

The outputs of the map operation are collected in the provided OutputCollecter, which is explained later.

If applicable, the intermediate outputs are passed through a combiner. More information about a combiner is given later.

The intermediate outputs are written to a file. The Output Path, as set in the configuration file, determines where the results are written. The locations of these files are then passed to the master worker, which concatenates the results together to be reduced.

**Reducer**

The reducer takes the intermediate key/value pairs as produced by the mappers, and combines them into the final result of the Map-Reduce job. This is done by grouping all values with equal keys together, and running the reduce method on the key and its associated set of values.

The number of reducers to be used for a given job is defined in the configuration file. The intermediate key/value pairs are split amongst the reducers by hashing the keys into regions. The reducer then collects the data to be reduced, collects all values with equal keys together, and calls the reduce method on each key and its associated set of values.

The reduce method takes the following inputs:

* Intermediate key
* Set of Intermediate values
* Output Collecter

The outputs of the map operation are collected in the provided OutputCollecter, which is explained later.

The intermediate outputs are written to a file. The Output Path, as set in the configuration file, determines where the results are written.