Big Data and Analytics - Weatherlink
Personal Report
Matt Winchester
Milestone: Literature Review

I was very interested in learning about some core big data concepts, so I looked for papers about some of the topics mentioned in class.
There were several papers I found, and there were three I decided to read: a Map Reduce paper, a HDFS paper, and a paper on Kafka

Paper Name	Topic	Link
MapReduce: Simplified Data Processing on Large Clusters	Map Reduce	https://static.googleusercontent.com/media/research.google.com/en//archive/mapreduce-osdi04.pdf
The Hadoop Distributed Filesystem	HDFS	https://hadoop.apache.org/docs/r1.2.1/hdfs_design.pdf
Kafka: A Distributed Messaging System for Log Processing	Kafka messaging (streaming communication)	https://github.com/jeffrey-xiao/papers/blob/master/systems/kafka-a-distributed-messaging-system-for-log-processing.pdf

The Kafka paper seemed to be very specific to logging (and very specific to LinkedIn), so I did not write about that in the literature review, but the other two were very good and relevant.

I also organized our team by getting a google docs folder going so we could all put in the papers we had found and wanted to write about.

The papers I read were some of the very early and fundamental groundwork for Big Data. The Map Reduce paper was a great read, I was happy that the language in both papers was clear and understandable. I have been intimidated by papers before since oftentimes the language used is very specific and hard to understand, but this was not the case for either of the papers. It's possible my CS background Is influencing that, but the papers laid out their ideas very clearly and understandably.

Before reading the paper,I thought that map reduce was a graph theory algorithm, using graph theory to ‘map out’ a task and then ‘reduce’ it. When the paper began it compared the ‘map’ portion to the python method ‘map’, which just applies a function across an iterable list of elements, and it clicked! This was a way to process large amounts of data by doing the same

thing to each element, in a distributed fashion. The ‘reduce’ portion was a little harder to wrap my head around, but eventually that made sense too - it was just pooling the results together so you have one final answer (or set of answers) from the processing of the data. The examples presented in the paper were very helpful.

The Hadoop HDFS paper was also very insightful. I had wondered how a distributed file system could work, how could an operating system know how to access the data? And how could data collisions be avoided? This paper was also a great read, and HDFS really just made sense to me. Not only was distributing the nodes a way to insure data integrity, allow for ‘large’ amounts of data to exist in the same file system, but I also saw how this drastically increases read and write speeds to the data, since multiple data portions can be read simultaneously from various nodes. Plus, keeping that data in memory across many many computers could make running ‘real time’ applications that needed huge amounts of data to work possible. Being able to network together the RAM pool of a huge server finally made sense to me. (This isn’t necessarily specific to HDFS, but the principle of having a distributed file system across the memory of many computers is the same).

These two papers together answered some big questions in my head. How do you harness a huge amount of memory for a specific task? How do you coordinate processing and handling errors? How do you account for hardware failures? The Map Reduce algorithm and HDFS account for these issues. I know there have been big improvements since these papers were published, but I suspect the fundamental ideas of Big Data still revolve around this core idea of pooling compute and memory resources by coordinating them gracefully.