**Final project - Real time stream processing using Kafka and Storm**

**Deeptha kannan Rangarajan**

Hadoop is a powerful framework to do big-data analysis. It takes care of the complex work of utilizing hardware resources efficiently in a cluster, breaking huge files into chunks, running our map-reduce programs, monitoring, etc

But there is a problem with Hadoop which we all might have thought of at one point of time or another. **IT REQUIRES A FILE.** But most of the times, we have streaming data coming in to be processed. What if we have fast-big-data? Does hadoop support real time data processing, answer is NO.

Next big questions is, do we have a framework equivalent of hadoop that supports big-fast-real time data? Fortunately, yes. In fact, it is combination of two beautiful open sourced projects, Kafka and Storm.

**Kafka**

Real Time data is not new to this world. Big companies with large streams of data use Queues with Workers to support stream processing. There are several bottlenecks here, what if the node goes down, scalability, maintenance, **throughput**, etc.

Kafka, originally developed by Linkedin, was build to be durable, scalable, provide high throughput with great efficiency when working with fast streams of data.

Now kafka solves the problem of consuming the streams, storing it(if you want to process again) and sending it out to be processed by stream processing frameworks.

**Storm**

Storm provides a framework where you write simple programs very similar to the map-reduce function you would write for Hadoop.

Storm, originally developed at BackType has similar components compared to Hadoop when it comes to job allocation and tracking but in a much more elegant way.

Storm takes input from Spouts and processes them in Bolts. Bolts are created as required(or configured) by the framework thus abstraction away the complexities.

One would get peak efficiency when Storm and Kafka work with each other. Please watch my presentations to know more about these two amazing technologies and to understand how to make them work with each other to get the best out of real-time-stream-processing.