**HU Extension School E-185 Big Data Analytics**

**Handed out: 04/09/2013 Due by 10 PM on Thursday, 05/09/2013**

**Final Project, Assignment 11 - Project Description**

**Real time stream processing using Kafka and Storm**

This project is about the usage of two very popular streaming data processing frameworks available today, their interoperation and how they can be used to solve some interesting computational problems. These frameworks are Kafka and Storm.

**Kafka** is a project that was originated at LinkedIn that is publish-subscribe messaging system. It is able to handle all activity stream data and processing on a consumer-scale web site, which is a big deal if we consider LinkedIn's size and scope. It is comparable to Apache Flume and Facebook's Scribe in its capabilities and high speed throughput, but its API behaves more like a messaging system.

Some of the most interesting features of Kafka are its persistence in handling the messages and it's capability to parallel load the data to Hadoop at the same time it is streamed to a real-time framework (such as Storm).

**Storm** is a real-time system that was developed by BackType, that was acquired by Twitter on 2011. It's main abstraction is that you are able to create a "topology" of nodes (that are called Spouts for data-generating sources, and Bolts for worker nodes) that process the data as it passes through them and passes the result over to any other nodes that are connected to them. It has a lot of automation to create and rebalance nodes in a cluster using Zookeeper and provides a very useful guarantee of delivery and process of each message between nodes.

This project will demonstrate the process of installing and integrating these two components by the use of a Spout that specifically handles Kafka messages and runs it through a word count Topology and saves the intermediate outputs to a group of local text files. The Kafka messages in the demo are generated by a sample program that reads a local text file and sends its though the Kafka messaging framework.

The example, although simple, demonstrates all the important features of the API and also covers the installation process of the tools, which is very involved due to the fact that the frameworks are respectively based on Scala and Clojure, and requires the correct usage of build tools for both dialects to JVM compatible bytecode.