**COP 5611 - Advanced Operating Systems**

**Weekly Project Report - 3**

**Team: Diablo**

**Coding Implementation Design:** We decided to first read the input log files and store it in an array. Then we would construct a list of sessions from the stored array which would be stored into a resultant array. Sessions would be created on the basis of IP address of the websites visited and a time window value of 10 minutes. Further, we would sort the array and then would start implementing the Apriori algorithm on it to decipher patterns.

**Team member: Gaurav Sinha (gs13h)**

**1)** I read through the following paper that is associated with our project.

Self-\* Storage: Brick-based storage with automated administration - Gregory R. Ganger, John D. Strunk, Andrew J. Klosterman (*Carnegie Mellon University*)

This paper discusses a new project idea with respect to the implementation and designing of self-storage systems. The project encapsulates ideas of self-organizing, self-configuring, self-tuning, self-healing, self-managing systems. The paper strives to reduce the administrative load encountered by data center administrators by integrating ideas from control systems and AI. It also strives to improve the reliability, availability, and performance of the new systems.

**2)** I worked on the following part related to the implementation:

I worked on the coding implementation that is related to the creation of sessions from the file that was parsed into an array.

**Team member: Harish Chetty**

**1)** I read through the following paper that is associated with our project.

Exploiting Lustre File Joining for Effective Collective IO - Weikuan Yu, Jeffrey Vetter (Oak Ridge National Laboratory), R. Shane Canon (Oak Ridge National Laboratory), Song Jiang (Wayne State University)

The paper proposes two methods to combat striping overhead and to benefit collective IO. These methods are hierarchical striping and split writing. When implemented together, these methodologies possess the ability to prevent overhead that are brought along with great stripe width, however, it would still be capable of combining available bandwidth from countless storage devices.

**2)** I worked on the following part related to the implementation:

I worked on the coding implementation that is related to the parsing of the log into an array and then sorting the resultant array using merge sort.