Abhijeet Ranadive
Long Beach, California

abhijeet.ranadive@student.csulb.edu   **(562)-787-8850**

Education

Candidate for Master of Science, Computer Science   **December   2016**
California State University, Long Beach

Bachelor of Engineering in Information Technology   **May   2014**
Pune Institute of Computer Science   **GPA   3.5**
University of Pune, India

Computer Proficiency

- **Programming Languages:** C, C++, Java, JavaScript and SQL
- **Software Framework:** Hadoop
- **Databases:** RDBMS, NoSQL, Hive.
- **IDE:** Eclipse, NetBeans 8.0, Microsoft Visual Studio
- **Operating Systems:** Microsoft Windows, Linux, Mac OS X
- **GitHub Repository –** Email Security

Work Experience

PICT – College Stock Database Management System   **August 2012 - December 2012**

- System kept track of various hardware components such as printers, scanners, monitors etc. and assigned them to various departments on need basis.
- System also helped college to keep track of defective stock and create a maintenance ticket for those.
- Technology used: Oracle 10g and VB.

Projects

E-Mail Security for Gmail (Master's Project in Information Security)   **Dec 2015**

The main reasons behind implementing this project is to reap benefits from Gmail that protects our data from external attackers, and also protect our confidential data from Gmail itself. The body of the email is encrypted, as a result email service providers (such as Gmail) will not be able to read our conversations and thus complete privacy of user emails will be maintained. To put a perspective, client side encryption and decryption is performed and if one opens Gmail from web-browser all the emails will be seem gibberish as the body of the email is encrypted indicating that even if Gmail decides to read our emails, no meaningful information can be recovered. Only our Java application that is residing on the client-side will be able to decrypt email for the user. It goes without mentioning that even if the email service provider is compromised, your data will still be secure. For implementing the solution we have used RSA public-private key pairs along with cryptographic algorithms such as AES-symmetric encryption, HMAC, RSA-public key encryption (OAEP) and PBKDF2.
GitHub Repository – Email Security.

Hybrid Approach for Fault Tolerant Scheduling in Grid (Master's Project in Fault Tolerance)   **Dec 2015**

Grid is type of distributed architecture that aims to provide a super computer like processing power for solving complex problems. With the exponential demand for distributed data processing, fault tolerance schemes to ensure fault-free execution have gain tremendous importance. Keeping that in mind, we implemented a job scheduling scheme that works with multiple processors and states how jobs should be rescheduled in case of a fault. The main

idea of the scheduling as suggested by our reference paper is to reap benefits from both static as well as dynamic scheduling and implement a new hybrid algorithm. Inspired by the paper, our hybrid algorithm implements the check-pointing mechanism from the dynamic scheduling scheme and duplication mechanism from static scheduling. For the implementation part of the project we have used Java programming language along with GridSim 5.2 Library which provides useful tools for studying the efficiency of scheduling algorithms in a heterogeneous GRID environment.

<u>Breadth First Search Algorithm using Hadoop (Master's Project in Distributed Computing)</u>   **Dec 2015**

Hadoop was developed by Apache Corporation for the purpose of big-data analysis. Hadoop architecture consists of three main layers MapReduce, HDFS (Hadoop Distributed File System) and YARN (Yet another Resource Negotiator). Using the platform provided by Hadoop, Breadth First Search using Dijkstra's Algorithm was implemented. The purpose of the algorithm is to find the shortest number of hops (or nodes) that are required to reach from a single source to all the nodes in the graph. The input file was a text file was taken from Stanford dataset repository that contained all the information of the nodes and their neighbors. The algorithm was implemented in Java with the help of Hadoop-core libraries.

<u>Performance Improvement of Data Processing for Hadoop (B.E. Project)</u>   **May 2014**

Our project cycle started from an in-depth study of the MapReduce Framework on its open source implementation, Hadoop. There are basically four factors that affect the performance of MapReduce (data-processing layer of Hadoop) namely, I/O mode, Parsing, Scheduling and Indexing. We have implemented a modification in the I/O mode, which is replacing Streaming I/O by Direct I/O in the MapReduce Layer. We conducted performance measurement of Original Package versus Modified Package of Hadoop using a Word Count program. Our comparison of Original versus Modified Package show performance improvement about 20-25%.

Honors and Membership

- Member of Hardware & Software Committee of a National level technical event 'Impetus & Concepts' at PICT, Pune -2013.
- Conducted a Lecture on the emergence of Big-Data and Hadoop, with over 200 participants at PICT, Pune.
- Organizing member of 'Impetus & Concepts' at PICT, Pune – (2013).