Last Updated on 23rd February 2018

Aashutosh Taikar

aashutoshtaikar.github.io
ataikar@outlook.com | 541.908.5980

EDUCATION

OREGON STATE UNIVERSITY

MEng in Electrical and Computer Engineering
Expected Dec 2018 | Corvallis, OR
GPA: 3.17
Conc. in Embedded Systems

MUMBAI UNIVERSITY

BE in Electronics and Telecommunication
May 2015 | Mumbai, India
Cum. GPA: 3.68 / 4.0

LINKS

Github:// **aashutoshtaikar**
LinkedIn:// **aashutoshtaikar**

COURSEWORK

High Performance Computer Architecture
Parallel Programming
Computer Architecture
Interconnection Networks
Distributed Systems
Advanced Computer Networking

SKILLS

PROGRAMMING

Shell • Matlab
Python • LaTeX
C • C++ • Assembly

Familiar:
Java • GoLang• Hadoop •

Other Softwares:
•Diptrace (Circuit Designing Tool)
• Proteus (Circuit Simulator)
•LTspice

EXPERIENCE

CONTINUUM MANAGED SOLUTIONS | Network Support Engineer

Oct 2015 – June 2016 | Mumbai, India

- Maintaining windows server network via RDP
- Analyzing the issues by checking server event logs
- Remote monitoring and troubleshooting servers and its services

PROJECTS

AUTOCORRELATION USING OPENMP, SIMD ON CPU AND OPENCL ON GPU | C++

Spring17

This project reads a random signal file in an array, duplicates the array, runs the Autocorrelation function on multiple threads using OpenMP to increase the performance. The same task is implemented using the SIMD and OpenCL where the performance obtained is maximum of all by using OpenCL, as the data level parallelism is exploited using GPU elements.

CACHE SIMULATION | BookSim - A cycle-accurate interconnection network simulator

Spring17 | Interconnection Networks

Simulated the performance of the cache under the following conditions: least-recently-used (LRU) replacement policy,32 to 512 sets, 1-way to 8-way associativity, 16-byte cache lines. least-recently-used (LRU) replacement policy, 16-byte cache lines (block size), Sets/associativity combinations: 128/1, 128/2, 128/4, 2048/1, 2048/2, 2048/4, (Only L1 caches assumed present).

IMPLEMENTATION OF LINK MANAGEMENT PROTOCOL CONTROL CHANNEL FINITE STATE MACHINE | C

Winter17 | Advanced Computer Networking

In this project, we implemented a part of LMP, which controls the Finite State Machine of the Control Channel. We obtained the information from the RFC4204 document, which entirely describes the LMP protocol. We implemented a part of the LMP, which is the Control Channel Finite State Machine using socket programming by studying various states and events of the FSM.

RAFT-LEADER ELECTION IMPLEMENTATION | GoLang

| Winter17 | Distributed Systems

Raft is a consensus algorithm. The main part of the Raft is the leader election. The leader election is implemented using the logic described in the Raft algorithm. Three states described in the Raft were setup in this algorithm, which changed as per the Election Timeout in the program. The servers agreed upon the leader election and detected a leader getting failed to re-elect a leader for the next term.

HADOOP MAP-REDUCE TASK IMPLEMENTATION | Java

| Winter17 | Distributed Systems

- Created a program to dictionary sort the words by taking input from a text file.
- Created a program to compute the number of times items in pair were purchased from a Supermarket.

FARMBOT-AGRICULTURAL AUTOMATION ROBOT | Diptrace | Proteus | C

| March15 | Final Year Project