HARSHA RASTOGI

+1 412.499.1634 | harshar@cmu.edu | rastogiharsha.com

SUMMARY

Full stack developer who is keen about building data engineering solutions and back-end web development. Quick learner with a strong background in distributed systems, storage and applied machine learning

EDUCATION

Master of Science
2014 - 2015
Carnegie Mellon University, Pittsburgh, USA
Electrical and Computer Engineering, 3.9/4.0

Bachelor of Engineering
2009 - 2013
Birla Institute of Technology and Science (BITS), Pilani, India
Electrical and Electronics, 9.55/10

SKILLS

Courses
Distributed Systems, Storage Systems**, Machine Learning, Computer Systems, Mobile and Pervasive Computing*, Search Engine*, Computer Architecture, Digital Electronics and Computer Organization

Languages
C, Java, Python, CSS3, HTML5, SQL, x86, ARM

Tools
Django, Kaggle, Eclipse, GDB, QEMU, MATLAB, Ubuntu 14.04, IAR Workbench, Cadence Palladium

EXPERIENCE

SanDisk, Milpitas, USA
Software Engineer Intern
May'15 - Aug'15
Enhanced code coverage by developing a fault injection framework which supports run time control and system fault simulation *(C, Python)*

Broadcom, Bangalore, India
Hardware Design Engineer
Aug'13 - Jul'14
Enhanced mobile baseband SoC performance by identifying logical and functional flaws in the design and power management code *(C, Cadence Palladium)*

Intel Labs, Bangalore, India
Research Intern
Feb '13 - Jun'13
Improved personal safety by designing a low power wearable tracking device as part of Intel's Internet of Things (IoT) *(IAR workbench)*

Carnegie Mellon University
Pittsburgh, USA
Summer Research Intern
May'12 - Jul'12
Under Professor Onur Multu (SAFARI group), worked on designing efficient memory scheduling algorithms at the memory controller in a multi core system. Increased system throughput by 1%, fairness by 5.9% and performed exploratory work on *Blacklisting Memory Schedule*r presented at 32nd IEEE, ICCD, 2014

PROJECTS

Distributed Systems
(*C, Java)*
Scalable Web Service (Load Balancer) - Designed a simulated multi-tier web hosting service (online store) to maximize total revenue by ensuring short client response times, while minimizing running cost (number of VMs running). The service scales out dynamically and has a caching tier to reduce storage latency

Distributed Proxy Caching Servers (CDN) - Designed a multiple proxy system which performs whole file caching in a client-server RPC system. Proxy servers implement LRU eviction policy to reduce latency

Remote Procedure Call - Build an RPC system which supports transparent client file operations (open, read write, lseek, lstat etc.) at the server

Storage Systems
(*C, FUSE)*
Hybrid File System - Designed a hybrid file system based on FUSE which uses local SSD and amazon S3 servers for storage. It supports caching, segment level de-duplication and snapshots

File System Checker (fsck) - Designed a fsck utility to identify, parse, read, and manipulate on-disk image of ext2 file system

Machine Learning
(*Kaggle, MATLAB)*
Designed a novel image classifier to perform ten class classification task on small color images in CIFAR-10 data set. Configured a majority-voting algorithm (Accuracy 62%) that uses results of the best supervised classifiers to generate combined prediction

Systems Programming
(*C, ARM, QEMU)*
Memory Allocator - Designed a dynamic memory allocator and optimized its performance by implementing a segregated list with coalescing, four byte pointers and footer elimination

Multi-threaded Proxy - Designed a multi threaded HTTP proxy server that supports caching of web objects and concurrent client requests

Linux Shell - Designed an interactive command line interpreter in C supporting job control, signal handling and I/O redirection

Kernel Programming - Designed a single-task mini-kernel on QEMU that supports read/write/exit system calls and user space applications

* On-going course **Current TA