ANEESH KHER

1707-5 Crest Road, Raleigh, NC 27606, United States • Email: aakher@ncsu.edu • http://aneeshkher.github.io

EDUCATION

Raleigh, NC **North Carolina State University** **Fall 2014 – May 2016**
Master of Science in Computer Science, May 2016 (Expected). GPA: 3.44/4.00
Coursework: Design and Analysis of Algorithms, Operating Systems, Computer Networks, Internet Protocols, Database Management Systems, DevOps, Advanced Data Structures, Software engineering, Object Oriented Design and Development, Software Engineering

Pune, India **University of Pune** **Fall 2008 – May 2012**
Bachelor of Engineering in Electronics and Telecommunication Engineering, May 2012.
Coursework: Data Structures, Computer Architecture, Microcontrollers and Applications, Mobile Communication.

TECHNICAL SKILLS

- **Programming:** C, Perl, Java. Familiar with: Python, JavaScript, HTML, CSS, MySQL, Shell Scripting, TCL/Expect.
- **Operating Systems:** Linux, Windows, VMware ESX Hypervisor.
- **Software Tools and Networking:**
  - Node.js, Express.js, Cisco Nexus 1000v, Cisco VSG, Cisco VEM, Socket Programming, Cisco PNSC.
  - Configuration and Infrastructure: Docker, Puppet, Ansible, Jenkins, AWS, Vagrant, DigitalOcean, VMware vSphere
  - Git, Vim, Eclipse IDE, GDB, GNU Screen, ClearQuest, JIRA, ClearCase, Redis, Mocha, ATS AutoEasy.

WORK EXPERIENCE

Technical Intern **Broadcom Corporation** **Summer, 2015 - Present**

- Developed a file dependency graph generation tool which maps all the calling files to the called scripts in a directory.
- Developed a Perl module to interact with the internal JIRA server using REST APIs to create, retrieve, and edit issues.
- Wrote stored procedures in SQL and fixed CGI scripts. This improved device metric information retrieval.
- Wrote scripts to parse log files and generate CSV files from the parsed data, giving better memory metric information.

Software Engineer **GS Lab Pvt. Ltd.** **July, 2012 – June, 2014**

- Developed an automation tool in Perl which improved network firewall test coverage by more than 400%.
- Improved test coverage by covering IP Address, port, protocol, custom VM attributes, zone based attributes, and by adding test scenarios (multiple conditions in one firewall rule, multiple rules under one policy).
- Fixed and maintained over 20K lines of existing virtual infrastructure automation code on a continuous basis.
- Contributed to manual testing of some switch features – VXLAN, ACL, QoS, FTP, TFTP, and RSH.

ACADEMIC PROJECTS

- **RAM File System** (C Programming)
  Basic file system in memory using the FUSE library. Implemented user level functions for each of the basic system calls like open, close, read, write, etc. Tested using Postmark Benchmark.
- **DevOps Software Delivery Pipeline** (Various technologies. Team of 3)
  End to end software delivery pipeline demonstrating *Build*, *Test and Analysis*, and *Deployment*. Configured *Jenkins* and *git hooks* for build. Used *Mocha* and *Istanbul* for unit testing and analysis. Used *Express.js* to build a sample web application for deployment. Developed system health check tools and used *Redis* feature flags with *http-proxy* for canary release.
- **Simple FTP using Go-Back-N ARQ** (Java. Team of two)
  Developed a simple file transfer application. Implemented *reliable* file transfer with UDP packets using the Go-Back-N ARQ protocol. Used threads for retransmission timer and for resending lost packets.
- **University Housing Application** (Java, SQL. Team of four)
  Database application which allows students to register for housing and parking options. Write stored procedures for dynamic generation of invoices for each registered user and for roommate matching functionality.
- **Automatic VM provisioning** (JavaScript, Node.js)
  Developed a tool to automatically provision a VM on AWS and Digital Ocean. Configured the machines using Ansible.
- **Advanced Data Structures** (C Programming)
  - Disk based merge sort using basic merge, multistep merge, and replacement selection merge.
  - Created a B-Tree supporting search and insert. Made it persistent by storing it on disk after program termination.
- **Test Generation and Coverage** (JavaScript, Node.js)
  Developed a tool to analyze the source code, generate test cases and measure code coverage. Improved statement coverage from **71% to 100%** and branch coverage from **40% 100%**.