

VISHAL SAHU

Mobile: +1 631 542 3903
Email: vishalsahunitt@gmail.com

<http://www.fsl.cs.sunysb.edu/~vishal/>
<https://github.com/vishsahu>

SUMMARY

- I am member of File systems & Storage Lab (FSL), working under guidance of Prof. Erez Zadok. My primary areas of interest are Operating systems, Virtualization, Storage, Security and Networking.

EDUCATION

- Master of Science** in Computer Science **Aug 2015 - Dec 2016 (expected)**
Stony Brook University, NY **GPA: 3.50/4.0**
- Bachelor of Technology** in Electrical Engineering **Jul 2007 - May 2011**
National Institute of Technology, Tiruchirappalli, India **CGPA: 8.76/10.0, First Class with Distinction**

RELEVANT COURSES

Operating Systems, Analysis of Algorithms, Computer Networks, Computer Architecture, Asynchronous Systems.

TEHCNICAL SKILLS

- Languages:** C++, C, Python, Bash, Assembly
- Operating System:** Linux (kernel & user space programming), Windows
- Tools & platforms:** gdb, valgrind, git, QEMU, MATLAB
- Technologies/protocols:** Virtualization, Cloud Storage, TCP/IP, 9P

PROFESSIONAL EXPERIENCE

- MTS Intern** **Jun 2016 - Aug 2016**
VMware - Palo Alto, CA
VDFS development group, Storage and Availability BU
- Optimized extent insertion into in-memory B-tree to reduce meta-data foot-print of Virtual Distributed File System (VDFS).
 - Implemented read interface of VDFS to connect to remote AWS S3 storage backend using **AWS C++ SDKs**.
 - Implemented proc-like **posix interface** to list all the backend storage objects in organized directory structure for easy access, eliminating the need of additional protocol.
 - Added feature of formatting a VSAN volume using user defined storage policy.

- Lead Engineer** **Jun 2013 - Jul 2015**
Samsung R&D Institute - Bangalore, India
Multimedia IP design, System Software group
- Developed scalar for Pinch-to-Zoom feature capable of scaling images in range of 0.25x - 4x in **C++ on Android**.
 - Implemented SPIHT, wavelet coefficients based image compression algorithm in **C++**.
 - Optimized run time of multimedia module by 60% using **openGL** vectorization on Qualcomm Adreno GPU.

ACADEMIC PROJECTS

Aug 2015 - May 2016

- Asynchronous utility module for Linux** [\[link\]](#)
 - Implemented asynchronous job queuing mechanism to solve producer-consumer problem in **Linux Kernel**.
 - Implemented appropriate locking mechanisms to avoid races and dead-locks because of concurrency.
 - Defined fair scheduling policy to prevent starvation of low priority jobs.
- Anti-malware stackable file system (amfs)** [\[link\]](#)
 - Implemented stackable file system to detect malware patterns in files and quarantine them efficiently.
 - Developed mechanism to update pattern database with minimal re-scanning overhead by implementing update-on-write version keeping mechanism.
 - User can define forbidden words database during mount time and can update it later using `ioctl`.
- Design of JOS operating System** [\[link\]](#)
 - Designed and implemented 64-bit JOS operating system from scratch. Wrote boot loader, memory management, process management and file system in **C**.
 - Implemented fork using copy-on-write mechanism to reduce number of pages allocated per process.
- Implementation of 32-bit SPARC processor** [\[link\]](#)
 - Designed 32-bit SPARC in-order Scalar processor with 5-Stage pipeline.
 - Implemented write-allocate write-back Direct-mapped instruction and data caches.
 - Implemented data forwarding from memory and write-back stages to resolve RAW dependencies to improve throughput of pipeline.
- Performance study of Autoscaler** [\[link\]](#)
 - Done benchmarking of autoscaling of web servers by creating client – server test architecture on Google Cloud.
 - Implemented packet sniffing logic in **C++** to measure request rate, response time and queue length metrics.
 - Ran regressions for various web server workloads including traces from Google and Amazon.