utkarshjaiswal

full stack developer

contact

uj@sn0v.com
http://sn0v.com
(310) 467 5948

languages

Java, Python, JavaScript, HTML5, CSS3, bash, C++, C

frameworks

TitanDB, Storm, Redis, Pyramid, Kafka, Gremlin, AngularJS, NodeJS, Docker, Vagrant

education

2013 - 2014 **Master of Science** in Computer Science — UC Los Angeles
GPA - 3.82/4
Focus - Cloud computing, internet architecture, computer graphics

2009 - 2013 **Bachelor of Technology** in Computer Science — NIT Trichy, India
GPA - 8.43/10

experience

Mar '15 - **Network Data Platform, Cisco Systems Inc** — Full stack dev
Worked on end-to-end system for **network visualization**, threat detection and enterprise policy application
Developed a **multi-threaded, multi-bolt Storm topology** for reading inputs from various network sensors and persisting them in Titan
Developed a Redis based cache for Titan that **improved performance by 15%**
Interim UI lead for wireframing and development of the application UI
Developed **REST APIs** using Python's Pyramid web framework and the Gremlin graph query language
Developed Kafka consumers for several network sensors such as APIC-EM

Jun-Sep '14 **Media Plane Services Set, Cisco Systems Inc** — Summer Intern
Developed Java applications to demo the Media Plane Services Set API
Developed **RESTful web application** for customer demos

select projects

Sep-Dec '14 **Application of OCR to guitar tablature** — UCLA
Used Optical Character Recognition to convert images of guitar tabs to editable ASCII tabs
Auto-recognized tab tuning and presented alternate fingerings for chords used in the tab
Used a Python wrapper around Google's **Tesseract OCR engine**

Apr-Jun '14 **Analysis of discrete collision detection algorithms** — UCLA
Used **axis aligned bounding boxes** to render and compare the performance of the octree, KD-tree and the sweep and prune algorithms for up to **1024 objects each**
Used **pyglet**, a windowing and multimedia library in Python that provides OpenGL bindings

May-Jul '12 **SVD-RD using CUDA (GPGPU)** — IIT Madras, India
SVD-RD approximates the **Singular Value Decomposition** of the new matrix using a series of operations including matrix copying and multiplication, and SVD computation of a much smaller matrix.
Utilized the CULATools library based on CUDA (NVidia's parallel computing architecture for GPGPU computing) for the algorithm's implementation and **achieved 18x speedup**