Shankha Banerjee

Email: shankha@vt.edu Phone No: +91-9739900473

SUMMARY

- *Acknowledged in Java Language Specification (SE 8) for contributions to Eclipse tool chain https://docs.oracle.com/javase/specs/jls/se8/jls8.pdf : Preface Section*
- Masters in Computer Science and Applications
- 7 years' work experience in software design and development
- Hands on experience with large and complex code bases
- Committer access to open source Java Development Tool code base

EDUCATION

- **M.S.,** Computer Science - Virginia Tech, Blacksburg, VA *2012*
  Advisor: Dr. Srinidhi Varadarajan
- **B.E.,** Computer Science and Engineering PESIT, Bangalore India 2006

TECHNICAL SKILLS

Programming: C, Java, x86-64 Assembly, and Python
Parallel Programming: Message Passing Interface (MPI), OpenMP, Pthreads, Multi-threading, Thread Local Storage and CUDA
Framework/Tools: Low Level Virtual Machine (LLVM), Java Development Tools (JDT) Core, GDB, PDB (Python) cProfile (Python), TCP/IP, Valgrind, PurifyPlus, GNU gprof, Visual Studio
System Administration: Linux kernel, and GCC, LLVM tool chain

PROFESSIONAL EXPERIENCE

- **Senior Systems Software Engineer, IBM** *Bangalore* *Mar 2013 – Present*
  - Designed and developed a tool to detect binding comparison in the JDT code base. Binding comparisons were required to be modified due to introduction of Type Annotation in Java 8.
  - Implemented section in JLS 8 dealing with choice of most specific method in the scenario of varargs and method overloading.
  - Developed and supported features specified for Type Annotation (JSR 308) and Repeating Annotation (JSR 337).
  - Implemented features in Null analysis to help detect potential null pointer exceptions at compile time.
  - Worked on JDT core sub components: AST, search, error recovery, intermediate code, exception handling, type hierarchy and disassembler.
- **System Software Engineer, Qualcomm Incorporated** *San Diego* *Mar 2012 – Feb 2013*
  - Developed native hardware math calls for OpenCL tool chain new graphics processing chip.
  - Developed test suite in Python to check math algorithms for correctness on X86.
  - Debugged and isolated issues on experimental hardware and helped design performance metrics for the tool chain for mobile GPU.
  - Developed a tool to profile the tools chains on new graphics architectures for size on disk.
- **Engineering Intern, Qualcomm Incorporated** *San Diego* *May 2011 – Aug 2011*
  Developed testing framework based on Python for comparing Qualcomm's OpenCL 1.0 tools with OpenCL products from different vendors.
- **System Software Engineer, IBM** *Bangalore* *Jul 2006 – Jun 2009*
  - Developed and supported instrumentation engine and runtime library for detecting memory access errors, code coverage issues, and performance bottlenecks of UNIX applications.
  - Mentored two students as part of IBM extreme blue internship program.

RESEARCH EXPERIENCE

- **Graduate Research Assistant, Computer Systems Research Laboratory, Virginia Tech** *Blacksburg* *Aug 2009 – Feb 2012*
  Research in areas of parallel and distributed computing.
  Project: *Weaves:* Design and developed an instrumentation tool based on LLVM for single threaded applications. Weaves architecture independent design modifies single threaded code to run on multi-core machines to take

advantage of parallelism. Weaves provides semantics to specify data sharing and mutual exclusion between threads. Weaves instruments the code under observation to modify global variables into thread local variables and subsequently modifies their accesses.

- **Graduate Research Assistant, Advanced Research Computing (ARC), Virginia Tech** *Blacksburg* *Aug 2010 – Dec 2012*

  Programming consultant for high-end computing. Worked in parallelizing, porting, and fine tuning code for supercomputers. Also involved in administering and managing 100+ node clusters.

THESIS/PUBLICATION

Thesis: A framework to analyze file system performance of MPI application.

Developed and designed MPI I/O replay (MPIOR), an I/O performance modeling and prediction tool used to trace and replay a parallel application to determine application performance under a new I/O sub system. The trace collector deduces synchronization inter-dependencies between nodes and I/O demands placed by each node on the storage subsystem. The re-player mimics the behavior of the application across a variety of storage systems by mapping multiple processes to multiple threads running on a single node.

PROJECT

TCP/IP Stack

Developed and designed TCP/IP Stack taking care of variable RTT, bandwidth and drops due to congestion. Go-Back-N and selective repeat protocols were implemented as part of the project.

PROFESSIONAL SERVICE/OPEN SOURCE CONTRIBUTION

- Active involvement with **Python** community
- Committee member and organizer of Eclipse Day India in Bangalore 2014