



Sandeep Dasgupta

Email: sdasgup3@illinois.edu

Email: sandigame.123@gmail.com

Webpage: <http://webhost.engr.illinois.edu/~sdasgup3>

Github: <https://github.com/sdasgup3>

Phone: (+1) 2174172344

INTERESTED IN

- Compiler (LLVM, GCC) (Frontend/Backend/Optimizations)
- Static/Dynamic Program Analysis
- Design of Interpreter & Dynamic typed languages
- Data Analysis using Statistical tools
- Binary Code Decompilation
- Language Formal Semantics
- Parallel & Power Aware Computation
- Symbolic Execution

ACADEMICS

PhD Computer Science – GPA 3.95/4.00

August 2013 - April 2019 (Expected)

CS @ University of Illinois Urbana Champaign

- Working with Prof. [Vikram S. Adve](#) on [ALLVM Research Project](#)
- Current Research
 - Formal semantics of X86-64 ISA. [Public Repository](#)
 - Binary decompilation to LLVM: Extracting high level semantic information from binaries. [Poster Presentation](#) @LLVM Developers' Meeting, 2016.

M.Tech Computer Science & Engineering – CPI 9.00/10.00

August 2009 - June 2011

Indian Institute Of Technology Kanpur, Kanpur, India.

- Secured **rank 1** in M. Tech 2009 Batch, IIT Kanpur.

B.E. Computer Science & Engineering – *First Class with Honours*, 85.625/100.00 *August 2002 - June 2006*

Indian Institutes of Engineering Science and Technology, Shibpur, West Bengal, India.

- Awarded **University Gold Medal** for securing 1st Rank in BE, Computer Science & Engineering, 2002 batch.
- Awarded **Best Student Award**, sponsored by [Tata Consultancy Services](#), for outstanding performance in BE, Computer Science & Engineering, 2002 batch.

PUBLICATIONS

Conference Submissions

- Sandeep Dasgupta, Daejun Park, Theodoros Kasampalis, Vikram S. Adve & Grigore Rosu. “**A Complete Formal Semantics of x86-64 User-Level Instruction Set Architecture**”, in *Programming Language Design and Implementation*, PLDI 2019. [\[PDF\]](#) [\[PLDI'19\]](#)
- Sandeep Dasgupta & Amey Karkare. “**Precise shape analysis using field sensitivity**”, in *Proceedings of the 27th Annual ACM Symposium on Applied Computing*, SAC 2012. [\[PDF\]](#) [\[DOI\]](#) [\[BIB\]](#) [\[SAC'12\]](#)
- Barnali Basak, Sandeep Dasgupta & Amey Karkare. “**Heap Dependence Analysis for Sequential Programs**”, *International Conference on Parallel Computing*, ParCo 2011. [\[PDF\]](#) [\[DOI\]](#) [\[BIB\]](#) [\[ParCo'11\]](#)

Poster Submissions

- Sandeep Dasgupta, Joshua Cranmer, Edward Schwartz and Vikram S. Adve. “**Binary Decompile to LLVM IR**”, LLVM Developers’ Meeting, 2016.
[\[PDF\]](#) [\[LLVM Dev Meet’16\]](#)
- Sandeep Dasgupta, Barnali Basak & Amey Karkare. “**Dependence Analysis for Parallelization of Sequential Programs**”, the 8th ASIAN Symposium on Programming Languages & Systems, APLAS 2010.
[\[PDF\]](#) [\[APLAS’10\]](#)

Journal Submissions

- Sandeep Dasgupta, Amey Karkare & P. Vinay K. Reddy. “**Precise shape analysis using field sensitivity.**”, in *Innovations in Systems and Software Engineering*, ISSE 2013.
[\[PDF\]](#) [\[DOI\]](#) [\[BIB\]](#) [\[ISSE’13\]](#)

PROJECTS

- **Profiler Using Clang based AST Instrumentation** [\[Github\]](#)
Abstract: Clang based AST reformatting tool used for injecting instrumentation code in C/C++ programs. The goal is collecting profiles (run-times and execution frequencies) on C/C++ programs.
- **LLVM based Dwarf Type Reader** [\[Github\]](#)
Abstract: To read type information from *debug info* section of executables using LLVM based APIs.
- **Pointer Analysis Debugger** [\[Presentation\]](#) [\[Github\]](#)
Abstract: To detect bugs in LLVM’s pointer analysis using a mix of static program analysis and dynamic symbolic execution.
- **Partial Redundancy Elimination(PRE)** [\[Report\]](#) [\[Github\]](#)
Abstract: Implemented a PRE optimization pass in LLVM and measured results on a variety of applications and did a detailed study on the performance numbers.
- **Extended GCC’s Global Dataflow Analyzer** [\[Report\]](#) [\[Github\]](#)
Abstract: Extended the Generic Data Flow Analyzer GDFA (of gcc) to the data flow frameworks where data flow information can be represented using bit vectors but the frameworks are not bit vector frameworks because they are non-separable e.g., faint variable analysis, possible undefined variable analysis, strongly live variable analysis.
- **Designing an Interpreter of a dynamic language for Graph algorithms** [\[Report\]](#)
Abstract: Designed a dynamically typed language and an interpreter for it and achieved a slowdown of 2X w.r.t the execution time of statically compiled C language. This is obtained by providing built-in compiled functions for simple graph computation which in turn help to build complex ones.
- **Mitigating Impact of Heterogeneity Across Power-constrained Nodes on Parallel Applications through Load Balancing** [\[Report\]](#) [\[Github\]](#)
Abstract: Different processors across the nodes have different execution times for the same work-loads. This performance imbalance is seen only when the CPU power is capped to low values. This performance imbalance causes increased execution times of the parallel applications. We did a detailed study and proposed a power aware load balancer (using Charm++) which minimized the performance imbalance at the lower power caps by tackling this heterogeneity.
- **Designing superscalar processor** [\[Github\]](#)
Abstract: Developed a superscalar processor (with an issue rate of 2) using verilog hdl, and an assembler for that processor using flex and bison.
- **Graph Coloring Using State Space Search** [\[Report\]](#) [\[Github\]](#)
Abstract: Leveraged the state space search model for implementing graph coloring in parallel in Charm++. Some of the challenges for efficient exploration of space by chares include intelligent pruning of the state space, load balancing, grain-size control and low-overhead communication between chares. Evaluated multiple options for each of these, and come up with design decisions which would work for a large category of real-life graph applications.

PROFESSIONAL EXPERIENCE

Google Inc., Software Engineering Engineer

September 2018 - December 2018

- Worked in “Google kernel performance” team to augment the perf tools with “*Topdown Analysis*” capability.
 - Intel developed a cycle decomposition methodology called *Topdown*. It uses about 60 hardware events to breakdown micro-architectural performance bottlenecks encountered by running software on Intel processors. It is very useful in characterizing workloads and focusing on where to apply optimizations. Topdown is described as a tree of metrics, each one requiring many events, and published as a complex spreadsheet. The goal of the project is to automate the conversion of the formulas in the spreadsheet into C code that can be integrated into the perf tool. Today, this conversion is done by hand. As the spreadsheet is constantly evolving to support new hardware and tweak of the formulas, we need an automated way of doing the conversion. Such an automated converter is realized in this project.
 - We developed tool `topdown_parser` which takes as input, the topdown `xlsx`, and generates C code to collect and report all levels of topdown tree across all supported processors. The tool is general enough to emit code (encoding the topdown formulas) for other projects like, *Public perf tool*, in which case we emit json files encoding the topdown metric expression.
- Data Profiling to extract type information from binaries. Given a set of hot load/store instructions (generated by sampling the application using tools like perf), the problem is to find the type of the data address associated with those instructions. The found type information (which can be referred to as hot types) are useful in optimizing memory allocation strategies or cache utilization. We developed the following prototypes.
 - Based on Allocation Based Profiling: We generate samples of allocation information (like allocation base address, size, etc) and use that to find which allocation is feeding a particular load/store data address.
 - Based on Dwarf debug Info: Dwarf debug information stores, as a best effort, the variables in a given scope along with its associated type. The idea is to find which variable, in load/store instruction’s scope, is getting consumed by a given load/store data address. For that we needed to parse the location list information of dwarf.

Google Inc., Software Engineering Engineer

May 2017 - August 2017

- Worked in “Google continuous profiling and efficiency” team to augment the profiling tools with analysis capable of guiding the service owners to pinpoint unexpected performance changes (called “havocs” in this work) in CPU time. Challenges are:
 - To determine “havocs” amid noisy data.
 - To reduce false positives and pinpoint actionable “havocs” to the service providers.
- Investigated the effectiveness of existing strategies in literature and open source.
- Developed a statistical Go tool which determines potential “havocs” with reduced false positive rate. Came up with an idea of correlating the statistically detected trend changes in the data with build CL changes which not only gives confidence on trend change detection but also helps in reducing the false positives.
- Carried out the research part of the project independently aligning the milestones and the work progress with the project supervisor. Collaborated with the team on the details of integrating with the data querying pipeline.

Apple Inc., Compiler Engineer, PhD Intern

April 2016 - August 2016

- Worked as a member of Swift Performance Team on an exploratory project which requires sound knowledge in compiler backend. Achieved functional completeness with real work loads.

Intel Technology India Pvt. Ltd., Component Design Engineer

August 2011 - June 2013

- Worked as Design Automation Engineer for Formal Equivalence Verification (FEV) of hardware designs for Broadwell (BDX) and Skylake (SKL) projects. Involves building flows and methodologies and interacting closely with global FEV teams to provide FEV solutions.
- Owner of tools and infrastructures for driving FEV central runs for BDX. Involves working closely with teams of various design styles.

Interra Systems India Pvt. Ltd., Senior Member Of Technical Staff

August 2006 - July 2009

- Developer of Interra's premiere front-end analyzer products - Cheetah (SystemVerilog) and MVV(Mixed Verilog Vhdl) and provided support for several new constructs of System Verilog IEEE-1800-2005.
- Worked with [Atrenta \(I\) Pvt. Ltd.](#) (service project) for the development of System Verilog features in Spyglass DFT.
- Awarded **Interra Humming Bird Award** in recognition of & appreciation for providing excellent support to Atrenta (I) Pvt. Ltd. in the project "IEEE compliance for Spyglass".

TEACHING EXPERIENCE

University Of Illinois Urbana Champaign,

August 2013 - December 2013

- **Teaching Assistant** for [CS 427: Software Engineering I](#): A Graduate level course.
 - Designing the questions for homework and term examinations and grading those.
 - Conducting discussion sessions with the students to answer questions related to lectures/homework.

Indian Institute of Technology, Kanpur,

August 2009 - 2011

- **Tutor** for [ESc 101: Fundamentals of Computing](#): An undergraduate course.
 - Responsible for weekly lecture class on C - programming language, supervision of programming laboratory and grading assignments and term examinations.
- **Teaching Assistant** for [CS 335: Principles of Compiler Design](#): An undergraduate course.
 - Responsible for mentoring a student group on a course project of developing a simple compiler (using a subset of C-language constructs) demonstrating most of the phases of compiler design starting from Lexical & Syntax analysis upto Intermediate code generation.
 - Grading assignments and term examinations.
- **Teaching Assistant** for [CS355: Programming Tools and Techniques](#): An undergraduate course.
 - Grading assignments related to Software management tools such as make; Programming tools such as Python, Perl; Document preparation systems such as tex; Tools for building programs like Lex and Yacc.

HOBBIES

[Kuk Sool Won](#)

- At the moment I am actively learning "Kuk Sool Won", a Korean martial arts system, which involves systematic study of all traditional fighting arts. This incorporates physical exercise, as well as mental development and traditional weapons training.

Dancing

- I am also fond of dancing and choreography, preferably on [Bollywood](#) Beats. Have a look
 - [Group@UIUC Durga Puja Festival 2016](#)
 - [Group@UIUC Tagore Festival 2016](#)
 - [Solo@UIUC New Year Festival 2015](#)
 - [Solo@IIT Kanpur, Hall Festival 2011](#)

Table Tennis

- I have spent considerable amount of time playing Table Tennis with excellent players, so ping pong comes naturally to me.