Computer Science Department Colloquium
CS Faculty Rapid-Fire Presentations of Current Research | Group A

Sanjay Rajopadhye, Louis-Noel Pouchet, Hamid Chitsaz, Yashwant Malaiya, Sudipto Ghosh, Shrideep Pallickara, and Sangmi Pallickara,
Department of Computer Science, Colorado State University

Presenters

• Sanjay Rajopadhye
• Louis-Noel Pouchet
• Hamid Chitsaz
• Yashwant Malaiya
• Sudipto Ghosh
• Sangmi Pallickara
• Shrideep Pallickara
Upcoming Talks by CS Faculty

Dr. Francisco Ortega
September 24th

Dr. Craig Partridge
November 26th

Dr. Vinayak Prabhu
TBA
Sanjay Rajopadhye
Colorado State University

Research Interests

Systems Research

- Automatic Parallelization & Algorithms
- Functional/Equational Programming Languages
- Embedded Systems/Architecture
- High Performance Computing

Also Applications
What is MÉLANGE

- Models
  - Performance models
  - Programming models
  - MDE
- Equations
  - Equational programming
- Languages
- Algorithms
- NGE: Next Generation Executables
  - GPUs accelerators, FPGAs
  - Multi- and many-core processors
- Weekly meeting Wednesday 10–11 CSB 315

Foundation: polyhedral model

- Mathematical framework for programs
  - Abstraction of loop programs
  - Compute-intensive applications
- Quantitative measures of cost
  - Leads to mathematical optimization
- Closure properties under program transformations
- Workshop on polyhedral compilation techniques
  [http://impact.gforge.inria.fr/impact2016](http://impact.gforge.inria.fr/impact2016)
- Spring Schools (2013, 2016)
  [http://labexcompilation.ens-lyon.fr/polyhedral-school](http://labexcompilation.ens-lyon.fr/polyhedral-school), [https://mathsinfohpc.sciencesconf.org](https://mathsinfohpc.sciencesconf.org)
- Keynote at WOLFHPC workshop at Supercomputing 2018
RNA-RNA Interaction (RRI) is an important scientific challenge
- Potential cure for cancer
- Computational models of RRI (piRNA, BPMax, BPMaxW) are important but very expensive in
  - Time: $\Theta(N^3M^3)$ for lengths M and N
  - Space: $\Theta(N^2M^2)$
- We want 100-million-fold speedup

Need for Speed
- piRNA is slow and a memory hog
  - For two sequences of length 100 each, piRNA takes 3.5 hours on a 64-core machine with 512GB RAM
  - Cannot handle sequences longer than 200 ($N^2M > 40k$)
    - Machine goes unresponsive must be rebooted
- For whole genome analysis:
  - 30,000 genes (~2k length)
  - 500 “interesting” small RNAs (length ~100)
    - i.e., 15 million calls to piRNA
  - Each call to piRNA would take 3.5*8000 hrs
  - On an 8 Terabyte machine
  - 15 million calls would take 50 million years
  - We will do it in six months on 100 department machines
How to get there

- Easy parallelization (use $10^6$ “large enough” machines on the cloud) is too expensive
- Make piRNA run efficiently on small RAM machines (e.g., 16GB)
- Speed it up on 100 machines in the department
  - Need $10^8$-fold speedup on each machine
    - 1000-fold by using locality/parallelism/vectorization
    - 1000-fold by filtering on only 0.1% interesting pairs of sequences
    - Still needs 1000-fold speedup of filtering program

WE can get there

Showed 100-fold speedup of “miniapp” called (OSP)$^2$
- 100-times simpler than piRNA
- On small (fits in RAM) problem sizes
- We have the expertise to do this on multi-cores & also GPUs
Polyhedral Challenges

- **piRNA** is beyond current tools (few tens of lines to kLoC)
- **Multilevel tiling**: virtual memory, DRAM, caches, (and 2-levels of parallelism: cores & vector units)
- **Legality of tiling**: all six dimensions must be tiled: is that legal?
  - GKT: middle serialization (known since 1979, still not fully automatic)
- **Raise** the level of abstraction:
  - Sloppy Equations: Hamid should write ~100 eqns, not kLoC in C++
- Simultaneously schedule & tile **reductions**
  - On (OSP) PLuTO slows down the program (mostly)
What is Optimizing Compilation?

Main idea: ask a computer to find an equivalent program which executes faster than your own program

- Must preserve the program semantics, must exploit parallel/distributed architectures, etc.
- Multiple disciplines are leveraged: algorithmic, programming, architecture, mathematics, machine learning, experimental computer science, etc.

+ A compelling example: programming distributed systems (PIPSES)

Input: 20 lines, nearly identical to textbook, compiler generates 2000+ lines of code!

Opportunities in Computing

Many applications are driven by massive computation needs
- Modern physics, computer-aided design (CAD), deep learning, etc.
- Healthcare tomorrow, especially with genomics-related work?

Compilers are one of the cornerstone of modern computing
- The pace of new architecture development, and the complexity and specificity of the code needed makes programming-by-computers necessary
- High impact by enabling new applications
  - Examples: in-situ lung tumor detection, deep learning

So, why study compilers?
- A great way to learn many different topics: programming, algorithmic, hardware architecture, mathematics, etc.
- A special opportunity to interact with pluri-disciplinary teams, and learn a domain
- Compiler researchers/engineers are in high demand in industry

Some Research Opportunities [1/2]

Many topics possible, come talk with me! Office: CS346
Projects/research can be tailored to your skills / expectations, several funding opportunities

1. **Polyhedral compilation**: Design program equivalence classes, build new optimization algorithms
   - Multi-platform optimization (CPU/GPU/FPGA/SoC, collaborations with Intel and UCLA)

2. **Machine learning (ML)**: Use ML inside compilers to find better implementations of algorithms
   - Optimize ML applications (Tensors for deep learning, collaboration with Facebook AI)

3. **Hardware/software co-design**: Hardware/software partitioning: choosing whether to use the CPU, GPU, or programmable IP
   - Algorithms for efficient FPGA synthesis (using Vivado HLS, collaboration with UCLA)

4. **High-performance computing for Exascale science**: Optimization framework for in-node computing, deploy PolyOpt on DoE applications
   - Analyze and optimize RAJA applications (collaboration with LLNL)
Some Research Opportunities [2/2]

Many topics possible, come talk with me! Office: CS346
Projects/research can be tailored to your skills / expectations, several funding opportunities

5. **GPU acceleration of graph analytics**
   - Performance and energy optimization of sparse linear algebra computations
   - ML models to select best implementation for a particular graph (collab. with OSU and UCSD)

6. **Performance modeling**
   - Build analytical models to predict the performance of a program, without running it
   - Use ML to build more accurate performance models (collaboration with INRIA, France)

7. **Macro-dataflow and task-based programming**
   - Design new compilation algorithms for shared-memory and distributed-memory clusters
   - Runtime resource allocation and task placement (collaboration with Rice University)

8. - - - **Put your idea here - - -**
   - If you have a research idea, come discuss it with me!
Yashwant K. Malaiya

- Professor Computer Science
  - Teach Fault Tolerant Computing CS 530 on-campus/on-line

- Research areas
  - Fault modeling
  - Reliability/Risk
  - Testing and testable design
  - Quantitative security risk evaluation
  - Human/economic/social factors

- Field contributions
  - 200 publications, advised 65 graduate students, served on 250 graduate committees.

Research Approach

- Multiple perspectives, new and unusual approaches

- Contributions: concepts introduced include
  - Alhazmi-Malaiya Logistic Vulnerability Discovery Model
    - Used actual data for OSs, browsers, web-servers etc
    - Modeling long-term trends, cyclicity, underlying causes
  - Test coverage – defects relationship model
    - Relationship among test effectiveness, defects and reliability
  - Antirandom testing
    - Apply tests as different as possible ("cannot be done")
  - Detectability Profile
    - No such thing as an average fault
Some of the Research Interests

- Vulnerability discovery in evolving software
  - Modeling evolution and inheritance of defects
  - Zero day vulnerabilities
- Predictability of exploitation & Risk evaluation
  - Breach probability: regularly occurring or rare events?
- Money flow in vulnerability markets
  - Modeling participants and prices/rewards
- Assessing cost of a breach:
  - corrective/preventive actions, price paid by organizations and society (insurance)

Vulnerability Markets

![Diagram of vulnerability markets](image-url)
Information/Contact

- **Contact:**
  - 356 CSB,
  - email Malaiya@cs.colostate.edu

- **Recent research results:**
  - Website: [Yashwant Malaiya](http://example.com) > Publications,
  - Google Scholar

- **Recent students**
  - A. M. Algarni, A. Younis, O.H. Alhazmi, H. Joh, S.-W. Woo...
My Teaching and Research

SUDIPTO GHOSH
PROFESSOR OF COMPUTER SCIENCE
Sudipto.Ghosh@ColoState.EDU
http://www.cs.colostate.edu/~ghosh

Recent Courses for Grad Students

Spring 2018  CS414 – Object Oriented Design
Fall 2017    CS514 – Software Product and Process Evaluation
Spring 2017  CS580A5 – Software Testing and Analysis
Various     CS793 – Software Engineering Group
Recent Activities

Associate Editor:
- IEEE Transactions on Reliability
- Journal of Software Testing, Verification, and Reliability
- Software Quality Journal

Editorial Board:
- Information and Software Technology

Conference Program Co-Chair
IEEE International Symposium on Software Reliability Engineering, 2018

Core Research in Software Engineering

Modeling software
- Precise specification of structural and behavioral properties
- Composition of models
- International repository of software models (ReMoDD)
- Funding source: NSF

Testing and verification
- Regression testing
- Model-based test generation
- Mutation testing
- Fault localization
- Past funding: IBM, Qwest, NSF
Interdisciplinary Research

Chemical and Biological Engineering
- Population Physiologically based Pharmacokinetic modeling
- Funding: FDA

Infectious Diseases
- Mobile data collection for Dengue Decision Support System
- Funding: NIH

Anschutz Medical Center
- Testing ETL transforms
- Funding: CU Denver

Current Students

Mohammed Al-Refai (PhD Candidate)
Hajar Homayoni (PhD Student)
Erica Shin (Graduate Student)
Vidya Gaddy (Undergraduate Student)
B I G  D A T A  &  C L O U D  C O M P U T I N G

S A N G M I  L E E  P A L L I C K A R A
S H R I D E E P  P A L L I C K A R A
C o m p u t e r  S c i e n c e  D e p a r t m e n t
C o l o r a d o  S t a t e  U n i v e r s i t y

Research Overview

- Voluminous data management
  - Number of files and data packets can be very high
- Real time: analytics, stream processing, and query evaluations
- DOMAINS: Epidemiology, geosciences, environmental science, healthcare, and IoT
- CURRENT FUNDING SOURCES:
  - National Science Foundation [2]
  - Department of Homeland Security
  - Advanced Research Projects Agency-E (ARPA-E)

August 26, 2018
### GALILEO: Managing multidimensional time series data

- High throughput storage and retrieval of observations
  - Support for large number (~$10^{11}$) of small files
  - Petascale datasets
- Query support: Range queries, analytic queries, approximate queries, and probabilistic queries


---

### Tracking Methane Gas Leaks using Google Streetview Cars

- Joint effort with Environmental Defense Fund, Google, and Biology
- Process spatiotemporal mobile sensing data
  - Hosted by Google Streetview cars
- Collaboration with Google EarthEngine

NEPTUNE: High throughput stream processing for Internet of Things

- Online scheduling of streams in the presence of resource uncertainty
- Refinements for high-throughput
  - Interference alleviation
  - Application buffering
  - Backpressure for flow control
  - Entropy-based dynamic message compactions


SYMPHONY: Exploring consequences of disease outbreaks and vaccination strategies

- Analytics of voluminous epidemiological data
  - Statistical, machine learning, and ensemble methods to build analytical models
- Economic consequences
- Planning exercises
  - Real-time analytics and visualizations
- Scale: Manage over a trillion files

From the Center to the Edges, Like Ripples in a Pond

<table>
<thead>
<tr>
<th>□ Cloud Computing</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>□ Edge Computing</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>□ Distributed Generation of Sketches</th>
</tr>
</thead>
</table>