SHASHI GOWDA

Office: 32-780, CSAIL, Massachusetts Institute of Technology
gowda@mit.edu — +1 (617) 899 4295

With my software engineer hat on, I love building well-rounded software that is simple, beautiful and fast. With my CS researcher hat on, I love interpreters, compilers, program synthesis, symbolic computation and machine learning, and the coming together of these fields.

EDUCATION

2018-current ***PhD Candidate* in Computational Science and Engineering**
Massachusetts Institute of Technology, Cambridge, MA

My thesis project, Symbolics.jl, is the basis for the next generation of programming abstraction in scientific simulation. Combining symbolic programming and partial evaluation, it allows scientists to make a ”compiler” for every problem. Right now, 54 other projects with 217 second-degree dependents use Symbolics.jl (source)
Advisor: Prof. Alan Edelman | The Julia Lab

2010-2014 ***B. Tech.*, Information Technology**
National Institute of Technology, Surathkal, India

EXPERIENCE

2016-2018 ***Principal Software Engineer*, Julia Computing Inc.**

Built ***JuliaDB***–A distributed-memory analytical database competitive with Spark, pandas (+dask) (more details in our PyData NYC 2017 presentation), and kdb. I built about 80% of the project working at every level, including indexed relational tables (*IndexedTables.jl*), text parser compiler (*TextParse.jl*), compression, distributed blob storage (*MemPool.jl*), parallel scheduling (*Dagger.jl*), distributed table operations and statistics (*JuliaDB.jl*). I also helped clients deploy JuliaDB in machine learning applications and achieve orders of magnitudes of speedups over systems it replaced.

2014-2016 ***Research Software Engineer*, (Remote) CSAIL, MIT**

Developed a research distributed-memory array implementation and scheduler (*Dagger.jl*). Also developed DLSs for pedagogical visualizations (*Interact.jl*) and functional-reactive dashboards (*Escher.jl*) which were used in the famous MIT 18.06 and other courses. Leter distilled the core of these into a framework called WebIO, it allows creation of libraries of UI widgets widgets that work in Jupyter, VS Code, and over a standalone web app, without requiring modification.

See shashi.biz for a detailed software portfolio.

PUBLICATIONS

2021 High-performance symbolic-numerics via multiple dispatch

S. Gowda, Y. Ma, A. Cheli, M. Gwóźdź, V.B. Shah, A. Edelman, C. Rackauckas.
ACM Communications in Computer Algebra Vol. 55 (dl.acm.org)

2021 ModelingToolkit: A Composable Graph Transformation System For Equation-Based Modeling

Y. Ma, S. Gowda, R. Anantharaman, C. Laughman, V. Shah, C. Rackauckas.
preprint arXiv:2103.05244

2019 Sparsity Programming: Automated Sparsity-Aware Optimizations in Differentiable Programming

S. Gowda, Y. Ma, V. Churavy, A. Edelman, C. Rackauckas.
NeurIPS Program Transformations for Machine Learning Workshop. (pdf)

TEACHING EXPERIENCE

Spring 2019 & Fall 2020 Introduction to computational thinking (18.S191)

Fall 2018 High-performance computing (18.337)

NOTABLE TALKS

2022 Strange Loop, St Louis. ”Symbolic-numeric programming in Julia”

2022 SciMLCon, Boston, MA ”Symbolic arrays: past, present and future”

2018 JuliaCon, London, UK ”How JuliaDB works”

2019 NeurIPS, Workshop on Programming languages for ML, Vancouver BC ”Sparsity aware optimizations in differentiable programming”

2017 PyData NYC, NYC ”JuliaDB: A data system for Julia (with Jeff Bezanson and Josh Day)”

2016 Microsoft, Bangalore ”Keynote: A Functional Algebra of UIs”

2015 JuliaCon, Boston, MA ”Escher.jl-a new way to make and deploy UIs”

September 23, 2023