

Awards and recognitions

2020 ACM Distinguished Speaker

I have been selected as ACM Distinguished Speaker by the Association of Computing Machinery (https://speakers.acm.org/speakers/kumar_12579). Through the ACM Distinguished Speaker Program, ACM provides direct access to top technology leaders and innovators from nearly every sector of the computing industry for speaking engagements at colleges and universities, corporations, conferences, and ACM local Chapter events.

<https://speakers.acm.org/about>. Currently, there are only 204 ACM Distinguished speakers who have been recognized for their contributions across the world. World-renowned innovators like Dr. Brian A. Barsky, a professor from UC Berkeley, and Dr. Salman A Baset, the CTO of IBM Blockchain Solutions are members. The full list of speakers can be found at <https://speakers.acm.org/>

Qualstar Hall of Fame [By Qualcomm]

Jan 2015: For the outstanding work in improving MPPS significantly on LLVM builds of the Jolokia modem. This work was conducted under immense time pressure and your efforts directly resulted in ensuring that LLVM was successful on low-tier modem products.

Sep 2014: For excellent efforts on analyzing and implementing various compiler optimizations to significantly improve Coremark benchmark.

Program Chair of LLVM CGO Workshop 2017-2021:

LLVM Performance Workshop brings engineers and academicians to present their innovations and findings on compilation and interpretation techniques. It is co-located with prestigious CGO, CC, HPCA, and PPOPP conferences. In the previous conferences, employees of companies like Samsung, Facebook, Qualcomm, Google, Intel have contributed. We also have had contributions from renowned universities like ETH Zurich, Saarland University, MIT, Argonne National Laboratory, Carnegie Mellon University, and Argonne National Laboratory.

Link of previous LLVM CGO workshops:

- <http://llvm.org/devmtg/2017-02-04/>
- <http://llvm.org/devmtg/2018-02-24/>
- <http://llvm.org/devmtg/2019-02-16/>
- <http://llvm.org/devmtg/2020-02-23/>
- <http://llvm.org/devmtg/2021-02-28/>

Mentions of my work by reputed entities

On GVNHoist

- Lenovo enables GVNHoist while reporting SPEC 2017 numbers:
<https://www.spec.org/cpu2017/results/res2019q3/cpu2017-20190722-16310.html>

- AMD enables GVNHoist while reporting SPEC 2017 numbers:
<https://www.spec.org/cpu2017/flags/aocc130-flags-revA2-HPE.html>
- DELL enables GVNHoist while reporting SPEC 2017 numbers:
<http://www.crc.nd.edu/~rich/ROME/Dell/R6525/7742/cpu2017-20190930-18850.pdf>
- HP Enterprise enables GVNHoist while reporting SPEC 2017 numbers:
<https://www.spec.org/cpu2017/results/res2019q1/cpu2017-20190204-10941.flags.html>
- Gigabyte technology enables GVNHoist while reporting SPEC 2017 numbers:
<http://www.spec.org/cpu2017/results/res2019q3/cpu2017-20190722-16241.flags.html>
- Super micro enables GVNHoist while reporting SPEC 2017 numbers:
<https://www.spec.org/cpu2017/results/res2019q3/cpu2017-20190806-16717.html>
- <https://azenydub.cf/3bc4e8ac77ec/>
- <https://news.ycombinator.com/item?id=14072166>
- GVNHoist enabled in LLVM
(https://www.phoronix.com/scan.php?page=news_item&px=LLVM-Clang-4.0-Features)
- Embedded Linux presentation referring to GVNHoist
(<https://elinux.org/images/8/80/Status-of-embedded-Linux-2018-03-JJ64.pdf> page-42)
- Intel's senior software engineer referring to GVNHoist
(https://easypref.net/blog/2017/10/27/Vectorization_warmup)
- The paper "OpenCL Vector Swizzling Optimization under Global Value Numbering" refers to GVNHoist (
https://cpc2018.scs.ted.ie/wp-content/uploads/CPC_2018_paper_3.pdf page-6)
- GVNHoist contributes to peak performance in GPUs as mentioned in "Impact of Compiler Phase Ordering When Targeting GPUs" at the European Conference on Parallel Parallel Processing 2017 (<https://arxiv.org/pdf/1810.10496.pdf>)

On hot cold splitting optimization:

- A video recording to show why hot cold splitting optimization is used in all the iPhone apps: https://www.youtube.com/watch?v=YlKpO_q2HRw
- Huge performance improvements in some of the Apple's core libraries:
<https://github.com/apple/swift/pull/21016> For example: 48% In NSStringConversion
- The dlang community referring to hot cold splitting optimization in LLVM
(<https://forum.dlang.org/post/yhouqgqaxvqljsccpqxx@forum.dlang.org>)
- Intel's senior software engineer referring to the hot cold splitting optimization
<https://easypref.net/blog/2019/03/27/Machine-code-layout-optimizatoins>

On string::find

- The performance of libcxx's implementation of string::find has been improved dramatically. (<http://llvmweekly.org/issue/157> r290761)
- Reference to string::find on libstdc++
(https://en.wikipedia.org/wiki/String-searching_algorithm)

On Demacrofier:

- Visual studio replacing macros:
<https://devblogs.microsoft.com/cppblog/convert-macros-to-constexpr/>
- Repara project's Cevalop IDE: <http://repara-project.eu/>
https://iris.unito.it/retrieve/handle/2318/1523738/52857/15_RePara_ISPA.pdf
<https://www.cevelop.com/>

PlatformIO referred to my work on linkedin:

PlatformIO: <https://www.linkedin.com/feed/update/urn:li:activity:6714130215875973120/>

Research Papers with citations

[28 citations] A. Kumar, A. Sutton, and B. Stroustrup, "Rejuvenating C++ Programs through Demacrofication" in Software Maintenance, 2012. ICSM 2012. IEEE Conference on. IEEE, 2012

[1 citation] A. Kumar, A. Sutton, and B. Stroustrup, "The Demacrofier", in Software Maintenance, 2012. ICSM 2012. IEEE Conference on. IEEE, 2012

[5 citations] A. Kumar, Sebastian Pop, "SCoP Detection: A Fast Algorithm for Industrial Compilers", in IMPACT 2016 6th International Workshop on Polyhedral Compilation Techniques

Google Scholar: <https://scholar.google.com/citations?user=Tm57yZwAAAAJ&hl=en>

Co-authors of research papers:

Bjarne Stroustrup: The designer and original implementer of C++ programming language.
<http://www.stroustrup.com/>
<https://scholar.google.com/citations?user=Rr9Y8acAAAAJ&hl=en>

Sebastian Pop: Author of several major compiler optimizations, pioneer of polyhedral compiler frameworks in industrial compilers GCC and LLVM.
<https://scholar.google.com/citations?user=OpmnkcAAAAJ&hl=en>

Andrew Sutton: Professor at the University of Akron, Ohio. <http://asutton.github.io/index.html>
<https://scholar.google.com/citations?user=7ZrkgVMAAAAJ&hl=en>

Marshall Clow: has been programming professionally for 35 years. He is the author of Boost.Algorithm, and has been a contributor to Boost for more than 15 years. He is the chairman of the Library working group of the C++ standard committee. He is the lead developer for libc++, the C++ standard library for LLVM.

Patent

Patent title System and method for merging divide and multiply-subtract operations

Patent pending: 15/853628

<https://patents.google.com/patent/US20190102197A1/en?q=15%2f853628>

Co-authors of patent:

Bonnie Sexton: Author of 9 patents; Distinguished inventor at Samsung Electronics.

Patent profile:

<https://patents.google.com/?inventor=Bonnie+Sexton&assignee=Samsung+Electronics+Co.%2c+Ltd.&language=ENGLISH&dups=language>

LinkedIn profile: <https://www.linkedin.com/in/bonnie-sexton-590b2611/>

Invited course by RISC-V international:

edX.org course in collaboration with **RISC-V international** and **Linux Foundation** has invited me to create a course on “RISC-V Toolchain and compiler optimization techniques”. edX is a leading American massive open online course provider created by Harvard and MIT. It hosts online university-level courses in a wide range of disciplines to a worldwide student body. It has more than 20 million learners.

Email: <https://drive.google.com/drive/folders/1yvocvSRYdxO16fPl7W2c3yxgg1LQDyEz>



Megan Lehn

to me ▾

Thu, Nov 5, 5:21 PM



Hi Aditya,

Your course has been approved to move forward. Please let me know if you have any questions. Once you are ready to move forward please let me know and I will connect you with the LF Training team to get you started.

Thanks!

Megan



Technical Talks

Open Source Summit + Embedded Linux Conference North America (Approximate audience: 3500-4000)

- 2020 Code Size Compiler Optimizations and Techniques for Embedded Systems (https://static.sched.com/hosted_files/ossna2020/f9/OSSEmbeddedLinuxConference.pdf)

RISC-V Global Forum (Approximate audience: 700)

- 2020 Code Size Compiler Optimizations and Techniques for Embedded Systems (https://static.sched.com/hosted_files/riscvglobalforum2020/1d/OSSEmbeddedLinuxConference.pdf)

LLVM Developers' Meeting Presentations (Approximate audience: 300-500)

- 2020 Code Size Compiler Optimizations and Techniques
https://whova.com/embedded/session/llvm_202010/1162331/
(<https://llvm.org/devmtg/2020-09/slides/Kumar-LLVMDevCodeSizePdf.pdf>)
- 2020 A fast algorithm for global code motion of congruent instructions
https://whova.com/embedded/session/llvm_202010/1193951/
(<https://llvm.org/devmtg/2020-09/slides/Kumar-LLVMGlobalSched.pdf>)
- 2019 Hot Cold Splitting Optimization Pass In LLVM
<http://llvm.org/devmtg/2019-10/talk-abstracts.html#tech8>
(<https://llvm.org/devmtg/2019-10/slides/Kumar-HotColdSplitting.pdf>)
- 2019 -Wall Found Programming Errors and Engineering Effort to Enable Across a Large Codebase <http://llvm.org/devmtg/2019-10/talk-abstracts.html#lit4>
(<http://llvm.org/devmtg/2019-10/slides/Kumar-WallFoundCompilationErrors.pdf>)
- 2018 Porting Function merging pass to thinlto
<http://llvm.org/devmtg/2018-10/talk-abstracts.html#talk2>
(<https://llvm.org/devmtg/2018-10/slides/Kumar-FunctionMergingPortThinLTO.pdf>)
- 2017 Introsort based sorting function for libc++
<http://llvm.org/devmtg/2017-10/#lightning3>
- 2016 GVN-Hoist: Hoisting Computations from Branches
<http://llvm.org/devmtg/2016-11/#talk11>
(<http://llvm.org/devmtg/2016-11/Slides/Kumar-Pop-GVNHoist.pdf>)
- 2016 Performance improvements in libcxx
<https://llvmdevelopersmeetingbay2016.sched.com/event/8Yzk/performance-improvements-in-libcxx>
- 2014 Implementation of global instruction scheduling in LLVM infrastructure
<http://llvm.org/devmtg/2014-10/#talk17>
(<http://llvm.org/devmtg/2014-10/Slides/Larin-GlobalInstructionScheduling.pdf>)

LLVM Performance workshop at CGO (Approximate audience: 20-50)

- 2020 Cheap function entry instrumentation to collect runtime metrics
<http://llvm.org/devmtg/2020-02-23/#ak>
- 2017 Performance analysis of libcxx at LLVM CGO
<http://llvm.org/devmtg/2017-02-04/#kumar>
(<http://llvm.org/devmtg/2017-02-04/Performance-analysis-of-libcxx.pdf>)
- 2017 Efficient clustering of case statements for indirect branch predictors:
<http://llvm.org/devmtg/2017-02-04/#menezes> (goo.gl/qxBSxS)

GCC Cauldron (Approximate audience: 150-200)

- 2015 Loop optimizer and vectorization BOF
(http://gcc.gnu.org/wiki/cauldron2015?action=AttachFile&do=view&target=Aditya+Kumar,+Sebastian+Pop_+Loop+optimizer+and+vectorization+BOF.pdf)

CPP Now (Approximate audience: 100-150)

- 2017 <http://sched.co/A8J7>
(<https://github.com/hiraditya/std-benchmark/blob/master/docs/slides/slide-cppnow.pdf>)

OOPSLA SPLASH-I Invited Talk (Approximate audience: 40-50)

- 2020 <https://2020.splashcon.org/profile/adityakumar> (PLMW Mentoring)
- 2017 <https://2017.splashcon.org/track/splash-2017-SPLASH-I#program>

Design Automation Conference (Approximate audience: 30-40)

- 2017 <http://www2.dac.com/events/eventdetails.aspx?id=223-134>
(<https://github.com/hiraditya/std-benchmark/blob/master/docs/slides/slide-DAC-2017.pdf>)

References of my work

Improvements to the C++ Standard Library:

- Improve iostream: <https://reviews.llvm.org/D30268>
- Improve string::find in libstdc++ and libc++
(<https://gcc.gnu.org/ml/libstdc++/2016-12/msg00097.html>,
<https://reviews.llvm.org/D27068>)

Compiler optimizations:

- Global Instruction Scheduler in SSA IR level <https://reviews.llvm.org/D32140>
- Global Instruction Scheduler for VLIW architecture.
(<https://llvm.org/devmtg/2014-10/Slides/Larin-GlobalInstructionScheduling.pdf>)
- GVN Hoist in SSA IR level
<https://llvm.org/devmtg/2016-11/Slides/Kumar-Pop-GVNHoist.pdf>
- Hot-Cold Splitting in SSA IR level (<https://reviews.llvm.org/D50658>)
- A fast algorithm for Single Entry Single Exit region detection
<http://impact.gforge.inria.fr/impact2016/papers/impact2016-kumar.pdf>
- A cheap function entry instrumentation technique to help detect dead code in large codebase <https://reviews.llvm.org/D74362>
- Merging similar functions across the whole program in ThinLTO
<https://reviews.llvm.org/D52896>
<https://llvm.org/devmtg/2018-10/slides/Kumar-FunctionMergingPortThinLTO.pdf>

- Efficient clustering of case statements for indirect branch prediction:
<https://llvm.org/devmtg/2017-02-04/Efficient-clustering-of-case-statements-for-indirect-branch-prediction.pdf> <https://arxiv.org/pdf/1910.02351.pdf>