BERKELEY LAB
Bringing Science Solutions to the World

U.S. DEPARTMENT OF ENERGY
Office of Science

Producing Software for Science with Class

Damian Rouson

Computer Languages and Systems Software (CLaSS) Group (http://go.lbl.gov/class)

SIAM CSE 2023, 1 March 2023

Software Developed with CLaSS

- LLVM Flang, Caffeine, Matcha, Inference-Engine
- GASNet-EX, UPC, UPC Runtime, UPC++
- MetaHipMer, SIMCoV, symPACK
- Berkeley Quantum Synthesis Toolkit (BQSKit)

Thoughts on Sustainability

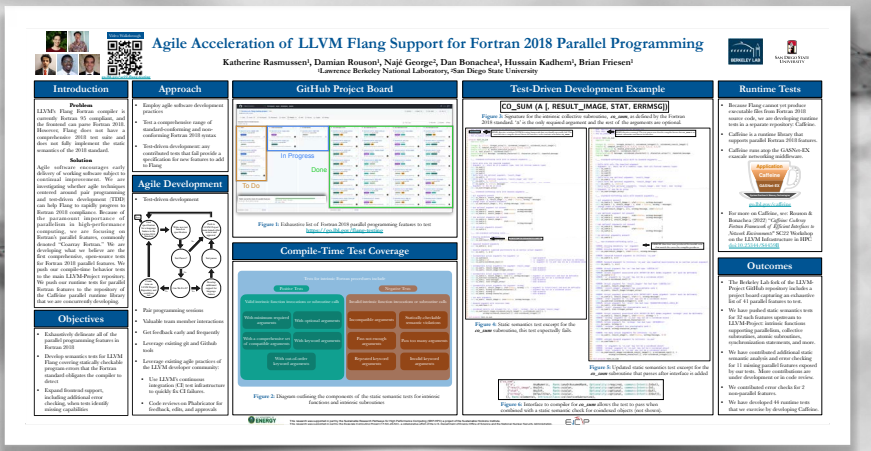
- Socially sustainable development
- Technologically sustainable development
- At the intersection of the social and the technological

Conclusions

LLVM Flang

The Fortran front-end in the LLVM Compiler Infrastructure Project.

- Using agile techniques employed across many CLaSS projects, we aim to accelerate Flang's support for Fortran's parallel features.
- Our agile practices include test-driven development (TDD), continuous integration, pair programming, and git workflows.
- TDD
  - Compile-time semantics tests for parallel Fortran 2018 features drive our contributions to the LLVM Flang frontend*.
  - Parallel runtime tests drive the development of Caffeine.


Agile Acceleration of LLVM Flang Support for Fortran 2018 Parallel Programming

Katherine Rasmussen[1], Damian Rouson[1], Najé George[2], Dan Bonachea[1], Hussain Kadhem[2], Brian Friesen[1]

[1]Lawrence Berkeley National Laboratory, [2]San Diego State University

Introduction

Approach

GitHub Project Board

Test-Driven Development Example

Runtime Tests

Agile Development

Compile-Time Test Coverage

Objectives

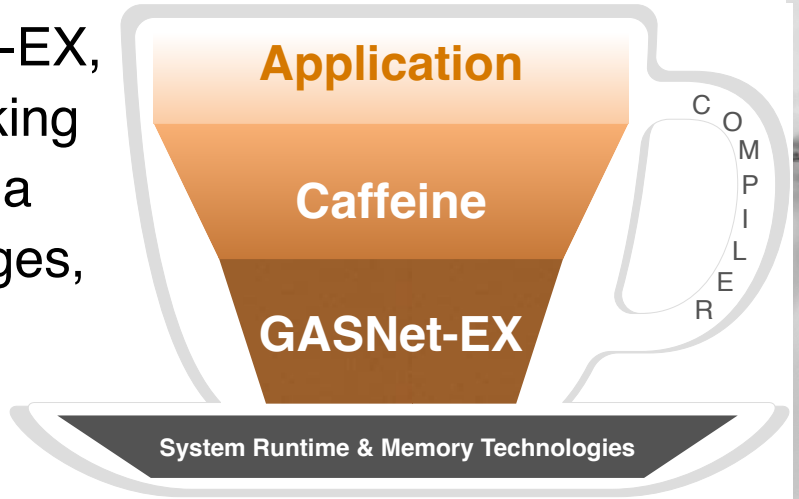
Outcomes

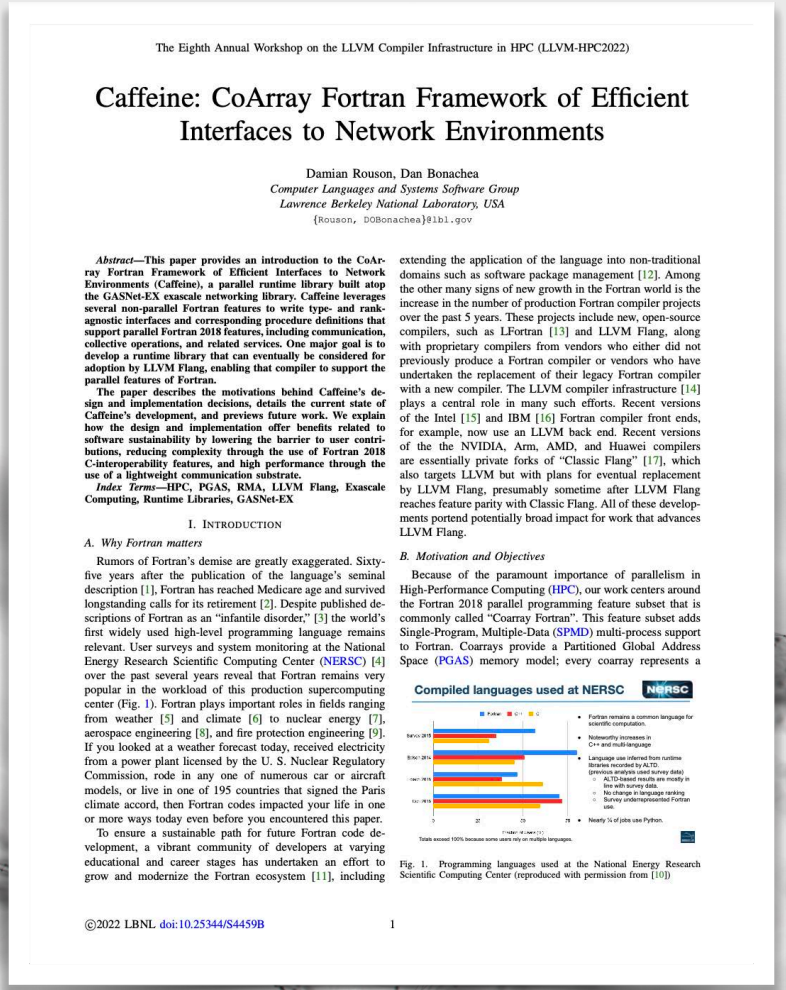
SC22 research poster: doi: 10.25344/S4CP4S:

Caffeine

Co-Array Fortran Framework of Efficient Interfaces to Network Environments (Caffeine)

- Caffeine supports the parallel features of Fortran 2018 for compilers.
- Caffeine leverages GASNet-EX, a high-performance networking middleware that undergirds a broad ecosystem of languages, libraries, frameworks, and applications.





The Eighth Annual Workshop on the LLVM Compiler Infrastructure in HPC (LLVM-HPC2022)

Caffeine: CoArray Fortran Framework of Efficient Interfaces to Network Environments

Damian Rouson, Dan Bonachea
Computer Languages and Systems Software Group
Lawrence Berkeley National Laboratory, USA
{Rouson, DOBonachea}@lbl.gov

***Abstract*—This paper provides an introduction to the CoArray Fortran Framework of Efficient Interfaces to Network Environments (Caffeine), a parallel runtime library built atop the GASNet-EX exascale networking library. Caffeine leverages several non-parallel Fortran features to write type- and rank-agnostic interfaces and corresponding procedure definitions that support parallel Fortran 2018 features, including communication, collective operations, and related services. One major goal is to develop a runtime library that can eventually be considered for adoption by LLVM Flang, enabling that compiler to support the parallel features of Fortran.**

The paper describes the motivations behind Caffeine's design and implementation decisions, details the current state of Caffeine's development, and previews future work. We explain how the design and implementation offer benefits related to software sustainability by lowering the barrier to user contributions, reducing complexity through the use of Fortran 2018 C-interoperability features, and high performance through the use of a lightweight communication substrate.

***Index Terms*—HPC, PGAS, RMA, LLVM Flang, Exascale Computing, Runtime Libraries, GASNet-EX**

I. Introduction

A. Why Fortran matters

Rumors of Fortran's demise are greatly exaggerated. Sixty-five years after the publication of the language's seminal description [1], Fortran has reached Medicare age and survived longstanding calls for its retirement [2]. Despite published descriptions of Fortran as an "infantile disorder," [3] the world's first widely used high-level programming language remains relevant. User surveys and system monitoring at the National Energy Research Scientific Computing Center (NERSC) [4] over the past several years reveal that Fortran remains very popular in the workload of this production supercomputing center (Fig. 1). Fortran plays important roles in fields ranging from weather [5] and climate [6] to nuclear energy [7], aerospace engineering [8], and fire protection engineering [9]. If you looked at a weather forecast today, received electricity from a power plant licensed by the U. S. Nuclear Regulatory Commission, rode in any one of numerous car or aircraft models, or live in one of 195 countries that signed the Paris climate accord, then Fortran codes impacted your life in one or more ways today even before you encountered this paper.

To ensure a sustainable path for future Fortran code development, a vibrant community of developers at varying educational and career stages has undertaken an effort to grow and modernize the Fortran ecosystem [11], including extending the application of the language into non-traditional domains such as software package management [12]. Among the other many signs of new growth in the Fortran world is the increase in the number of production Fortran compiler projects over the past 5 years. These projects include new, open-source compilers, such as LFortran [13] and LLVM Flang, along with proprietary compilers from vendors who either did not previously produce a Fortran compiler or vendors who have undertaken the replacement of their legacy Fortran compiler with a new compiler. The LLVM compiler infrastructure [14] plays a central role in many such efforts. Recent versions of the Intel [15] and IBM [16] Fortran compiler front ends, for example, now use an LLVM back end. Recent versions of the the NVIDIA, Arm, AMD, and Huawei compilers are essentially private forks of "Classic Flang" [17], which also targets LLVM but with plans for eventual replacement by LLVM Flang, presumably sometime after LLVM Flang reaches feature parity with Classic Flang. All of these developments portend potentially broad impact for work that advances LLVM Flang.

B. Motivation and Objectives

Because of the paramount importance of parallelism in High-Performance Computing (HPC), our work centers around the Fortran 2018 parallel programming feature subset that is commonly called "Coarray Fortran". This feature subset adds Single-Program, Multiple-Data (SPMD) multi-process support to Fortran. Coarrays provide a Partitioned Global Address Space (PGAS) memory model; every coarray represents a

Fig. 1. Programming languages used at the National Energy Research Scientific Computing Center (reproduced with permission from [10])

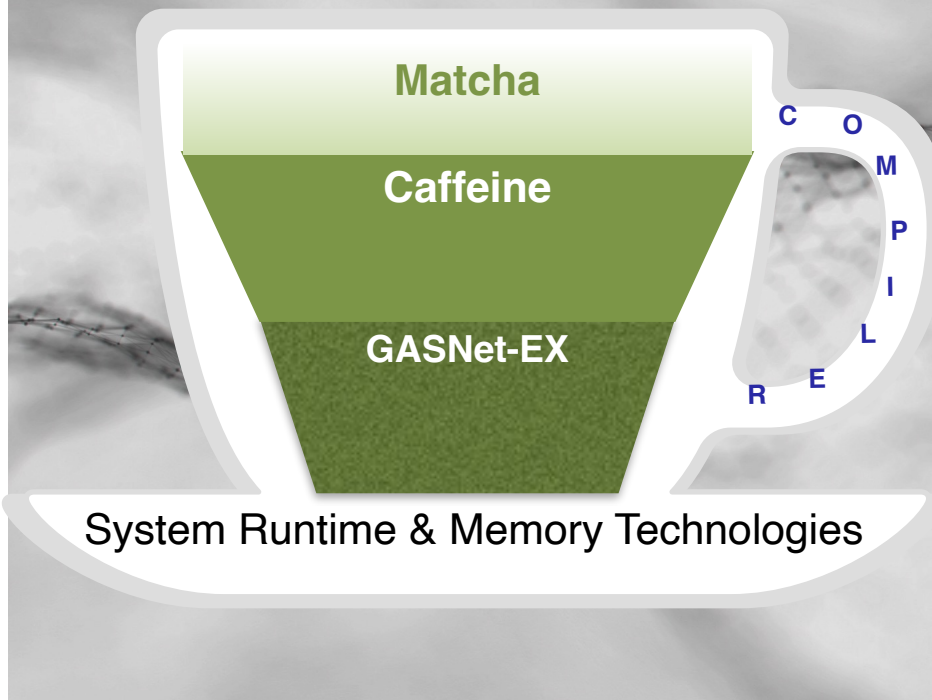
©2022 LBNL doi:10.25344/S4459B

LLVM for HPC Workshop paper: doi:10.25344/S4459B

Matcha

Motility analysis of T-cell histories in activation (Matcha)

- A parallel virtual T-cell model that captures the speed and turning angle distribution of T-cell motions in tissue.
- Matcha is the first target application for Caffeine.

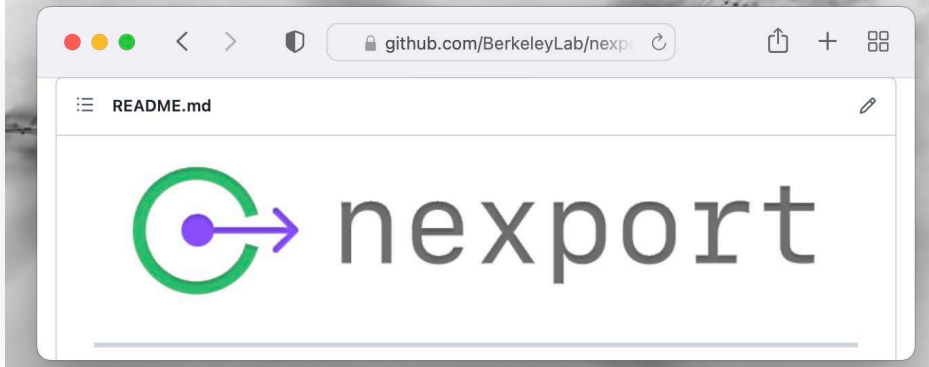


Inference Engine & nexport

Inference-Engine is a library for researching the efficient runtime inference in high-performance computing (HPC) applications using deep neural networks exported from Python by the companion package nexport.

- The implementation language, Fortran 2018, makes it suitable for integration into high-performance computing (HPC) applications. First target: the Intermediate Complexity Atmospheric Research (ICAR) model.
- A `pure`, `elemental` inference procedure facilitates optimized, including GPU-accelerated, large-batch inference via array statements or `do concurrent` loops.

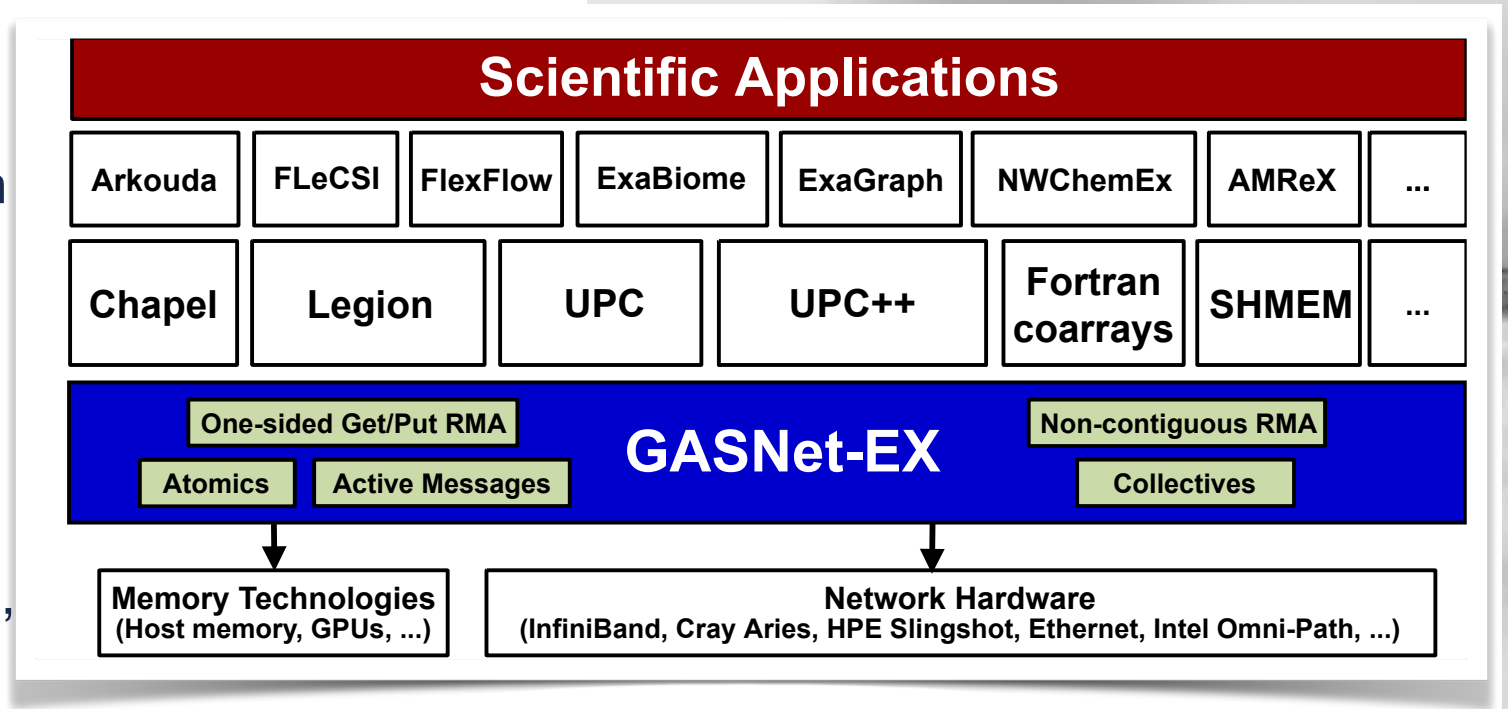
https://go.lbl.gov/inference-engine



https://go.lbl.gov/nexport

UPC++ & GASNet-EX

- UPC++ is a C++ template library supporting Partitioned Global Address Space (PGAS) parallel programming and interoperability with other common HPC frameworks, including MPI, OpenMP, C++/POSIX threads, CUDA, ROCm/HIP. https://go.lbl.gov/upcxx
- UPC++ leverages the GASNet-EX networking middleware to deliver low-overhead, fine-grained communication, including Remote Memory Access (RMA) and Remote Procedure Call (RPC). https://gasnet.lbl.gov/



MetaHipMer, SIMCoV, symPACK

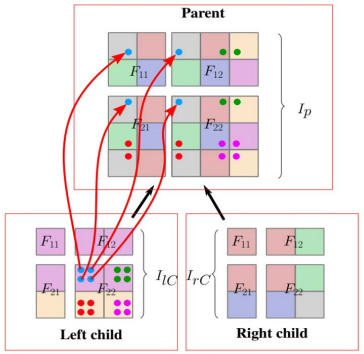
MetaHipMer

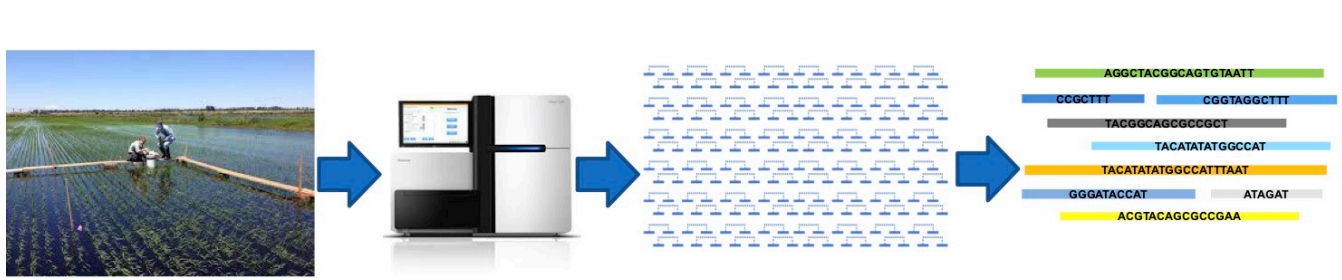
Application case studies

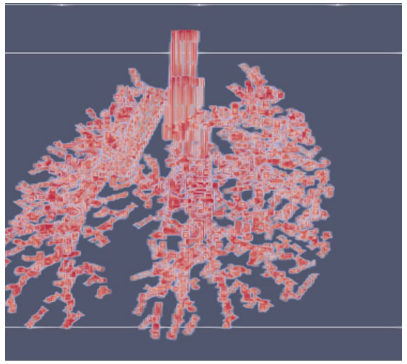
UPC++ has been used successfully in many applications to improve programmer productivity and runtime performance

We discuss several applications written in UPC++:

- symPack, a sparse symmetric matrix solver
- SIMCoV, agent-based simulation of lungs with COVID
- MetaHipMer, a genome assembler







SC21 Tutorial: https://go.lbl.gov/sc21

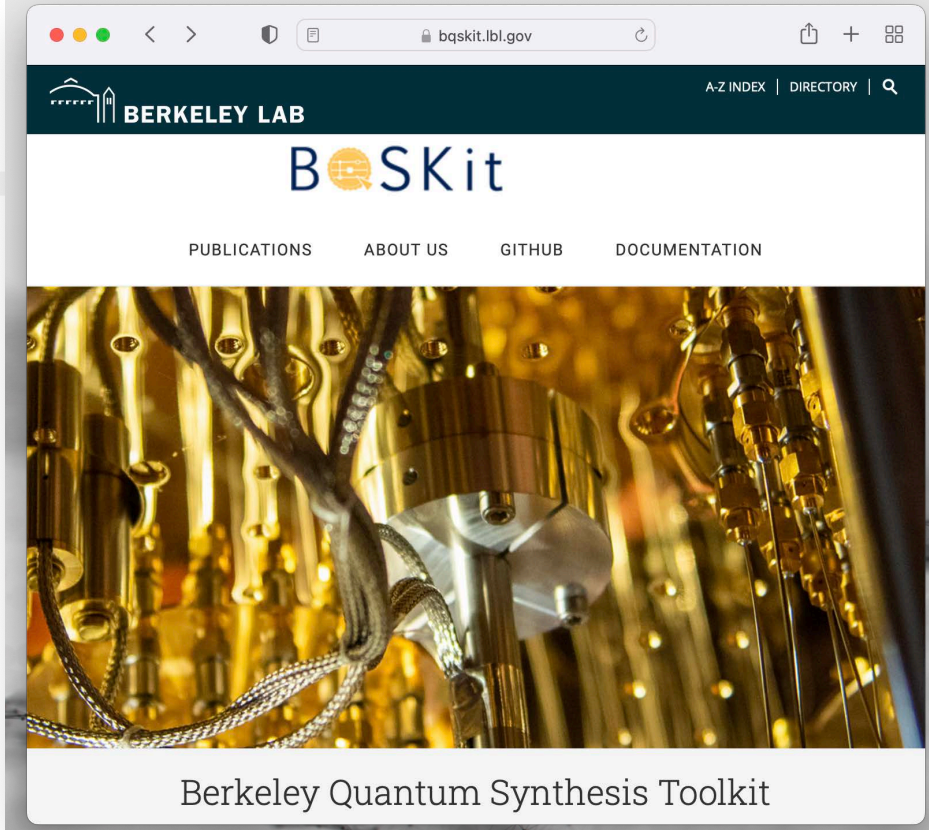
BQSKit

Toolbox

- QSearch: Optimal depth synthesis up to four qubits
- LEAP: Best quality of solution synthesis up to eight qubits
- QFAST: Scales good solution quality synthesis up to eight qubits
- QGO: Optimizing compiler combining partitioning and synthesis
- QUEST: Scalable circuit approximations
- QFACTOR: Fastest quantum optimizer

Berkeley Quantum Synthesis Toolkit

- An optimizing quantum compiler framework.
- Quantum synthesis converts a quantum program's mathematical description, given as a unitary matrix, to an executable quantum circuit.



Software Developed with CLaSS

- LLVM Flang, Caffeine, Matcha, Inference-Engine
- GASNet-EX, UPC, UPC Runtime, UPC++
- MetaHipMer, SIMCoV, symPACK
- Berkeley Quantum Synthesis Toolkit (BQSKit)

Thoughts on Sustainability

- Socially sustainable development
- Technologically sustainable development
- At the intersection of the social and the technological

Conclusions

Socially Sustainable Development

In a diversifying workforce, any sufficiently large project must diversify to be sustainable.

Sustainable Research Pathways

2015 - 2022

BERKELEY LAB


David Brown


Mary Ann Leung


Silvia Crivelli

HPCwire Workforce Diversity and Inclusion Award 2021

Sustainable Research Pathways (SRP)

- Build relationships centered on research collaborations
- Recruit
  - Faculty working with underrepresented students
  - Students from underrepresented backgrounds
- Provide opportunities for staff scientists
  - Research collaborations
  - Learn/contribute to diversity and inclusion efforts
- Supplement existing D&I Laboratory programs

Leung ASCAC presentation, July 2021

2022-

ECP EXASCALE COMPUTING PROJECT







Mary Ann Leung, Lois Curfman-Mcinnes, Dan Martin, Ashley Barker, Julia White, Erik Draeger

Sustainable Research Pathways for HPC (SRP-HPC)

- Expands SRP into a multi-lab program throughout the ECP community.
- Integrates participants into the broader ECP community through research presentations at the ECP Annual Meeting starting in 2023.
- Normalizes inclusion partly through Guided Affinity Groups and engaging DEI exercises open to all at the ECP Annual Meeting, other planned activities
- Blends the benefits of SRP with the Broader Engagement (BE) program led by Sustainable Horizons Institute (SHI) at SIAM CSE.

LLVM Flang & Caffeine Team

Broadening participation includes engaging a diverse ensemble of educational and professional backgrounds.

Hugh Kadhem
B.S. Comp. Sci., Math, Stat.
M.S. Pure Mathematics
Ph.D. Candidate, Math

Katherine Rasmussen
B.S. History/M.S. Linguistics

Damian Rouson
B.S./M.S./Ph.D., Mechanical Engineering

Brad Richardson
B.S./M.S., Nuclear Engineering

Engaging Users in Development Broadens Participation

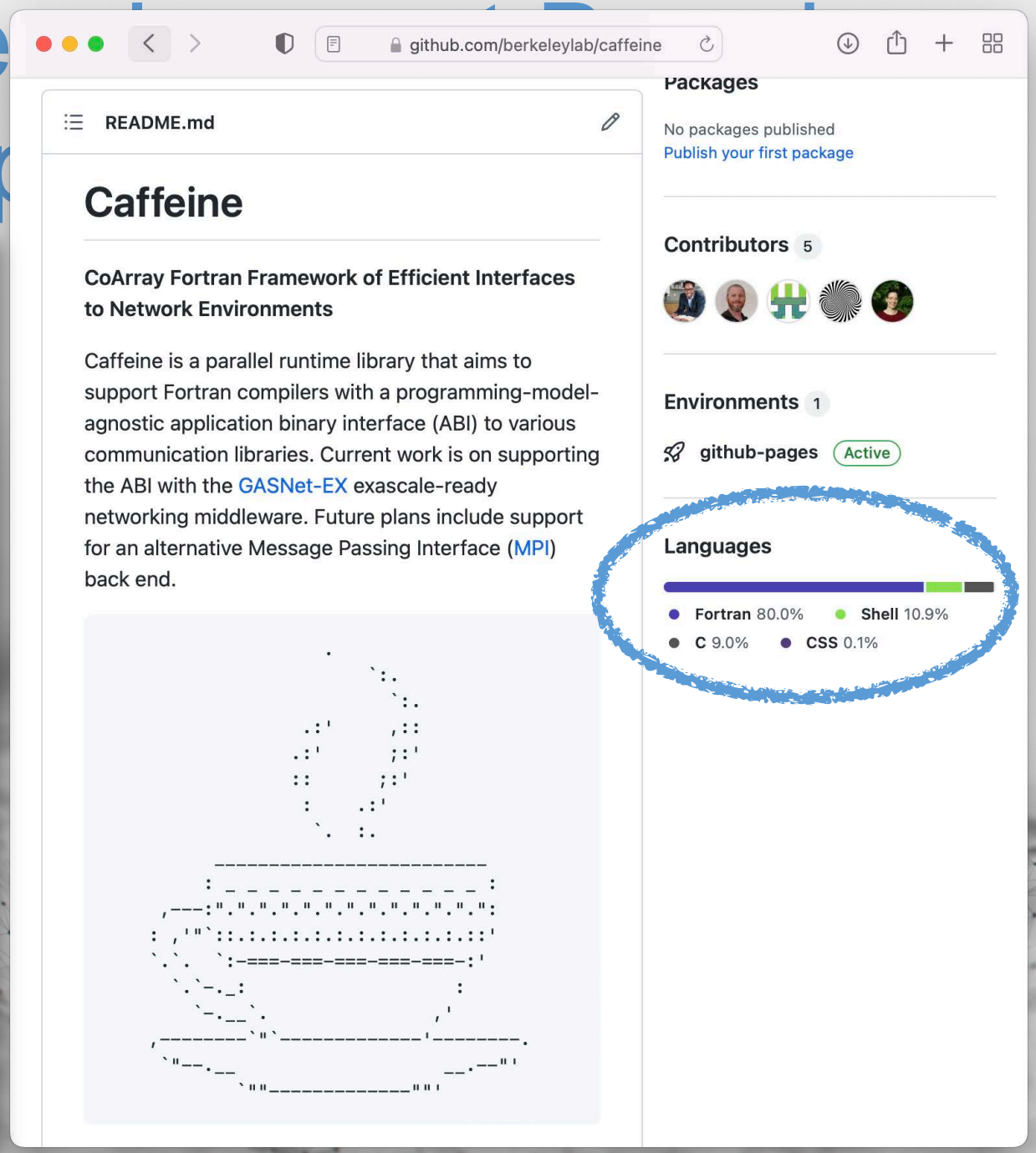
The Caffeine Proposition:

- A subset of the Fortran 2018 *non-parallel* features suffice for writing a runtime library, mostly in Fortran, to support the Fortran 2018 *parallel* features.
- Writing a parallel runtime library in the language of the users improves sustainability by lowering a barrier to community maintenance.
- In Caffeine, writing in Fortran also improves sustainability by reducing complexity and maintenance costs

Engaging Users in De… Particip…

The Caffeine Proposition:

- A subset of the Fortran 2018 *non-parallel* features suffice for writing a runtime library, mostly in Fortran, to support the Fortran 2018 *parallel* features.
- Writing a parallel runtime library in the language of the users improves sustainability by lowering a barrier to community maintenance.
- In Caffeine, writing in Fortran also improves sustainability by reducing complexity and maintenance costs



Develop Collaboratively and Train

bqskit.lbl.gov/

BERKELEY LA

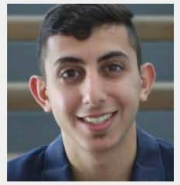
A-Z INDEX | DIRECTORY |

BQSKit

Berkeley Quantum Sy

The BQSKit Team

Costin Iancu
Senior Staff Scientist

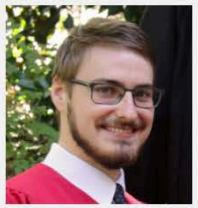


Ed Younis
Computer Systems Engineer

Wim Lavrijsen
Computer Systems Engineer

bqskit.lbl.gov/about-us/

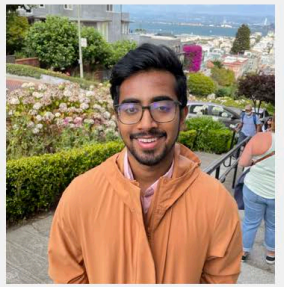
Marc Davis
Graduate Student



Mathias Weiden
Graduate Student

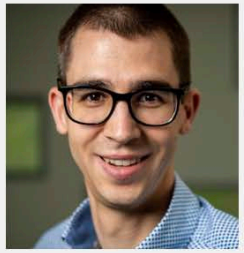
Lindsay Bassman
Postdoctoral Fellow

bqskit.lbl.gov/about-us/



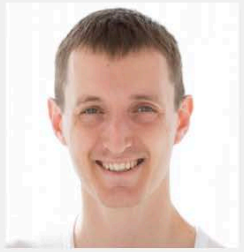
Justin Kalloor
Graduate Student

bqskit.lbl.gov/about-us/



Roel Van Beeumen
Research Scientist

Roel is a computational



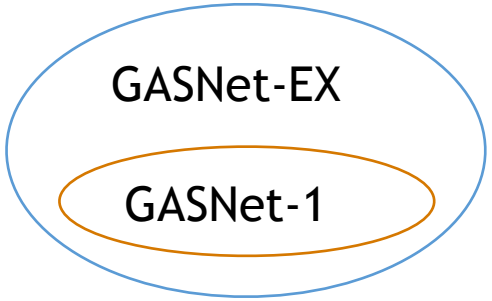
Alon Kukliansky
Graduate Student

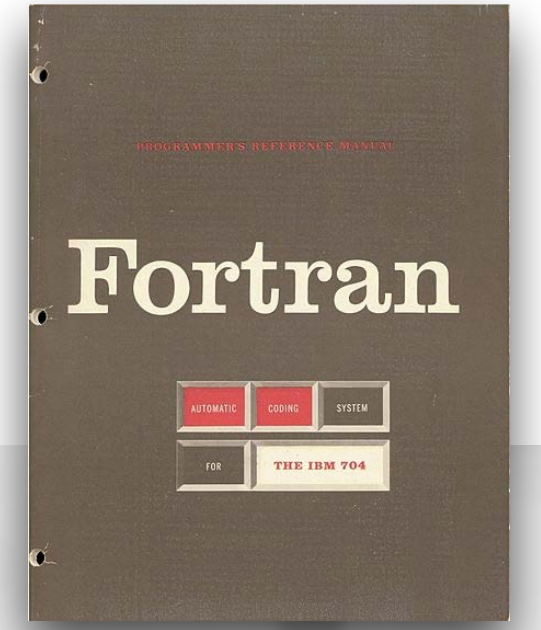
Ji Liu
Postdoctoral Appointee

Technologically Sustainable Development

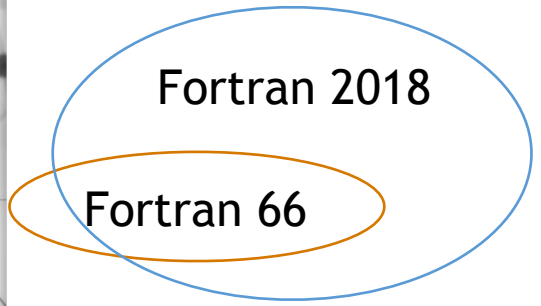
Backwards compatibility ensures continuity of user experience, protects users' investments in code and thus improves sustainability.

2002 –





1957 –



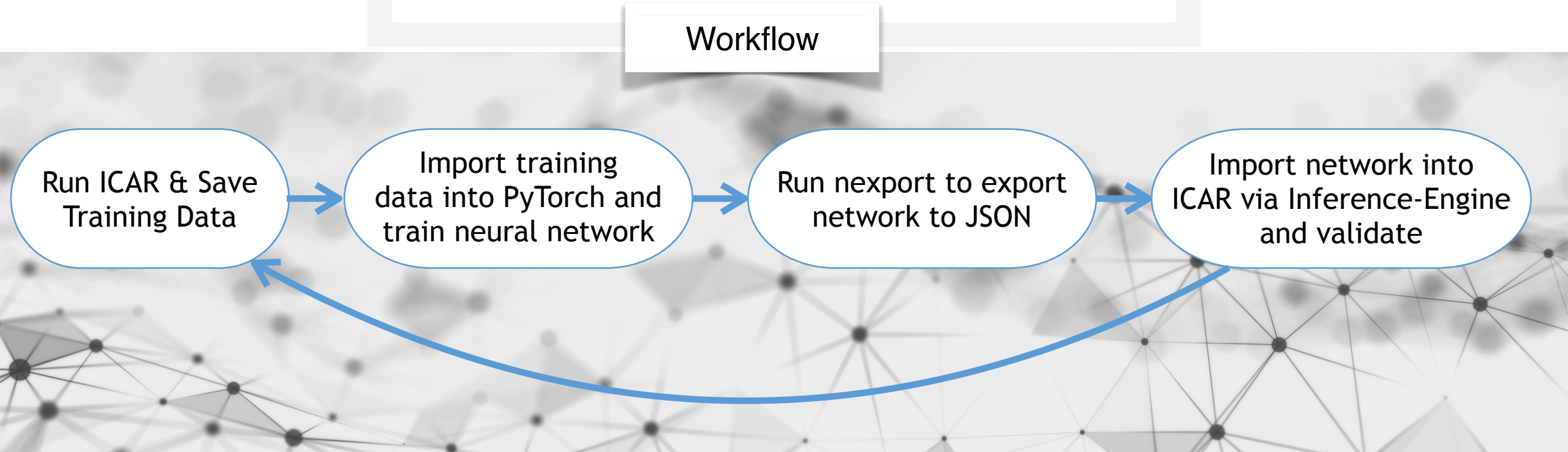
Technologically Sustainable Development

Reducing maintenance costs improves sustainability.

- Application source code never directly references GASNet
- Applications benefit from new GASNet feature releases without revising their applications.

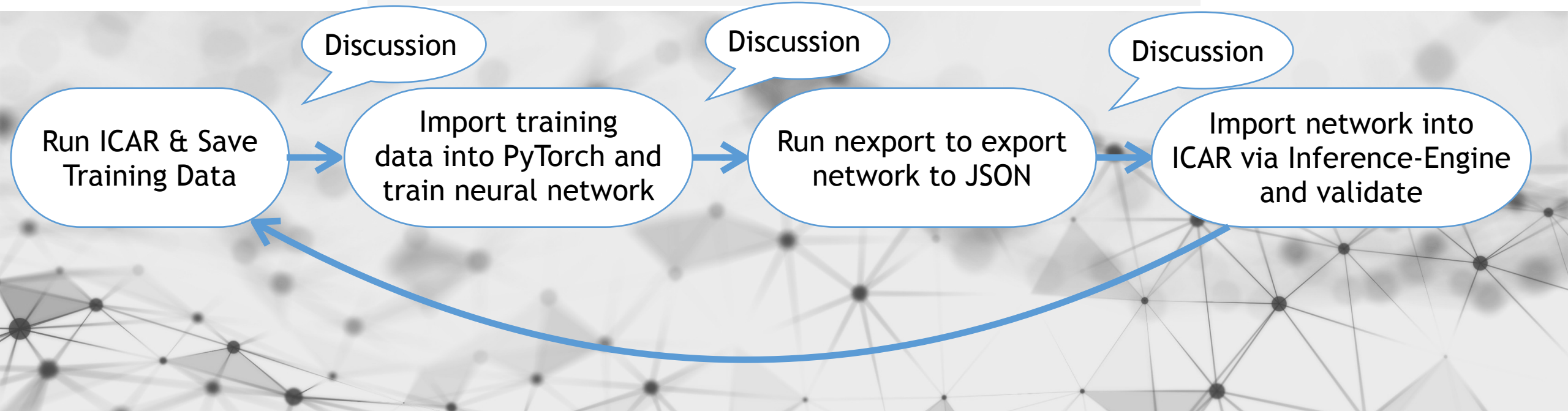
- Parallel programming languages require specialized compilers.
- Using UPC++ requires only a standard C++ compiler.
- This compiler-free approach greatly reduces the of the code base that the developers of UPC++ must maintain.

At the Intersection of Social & Technical

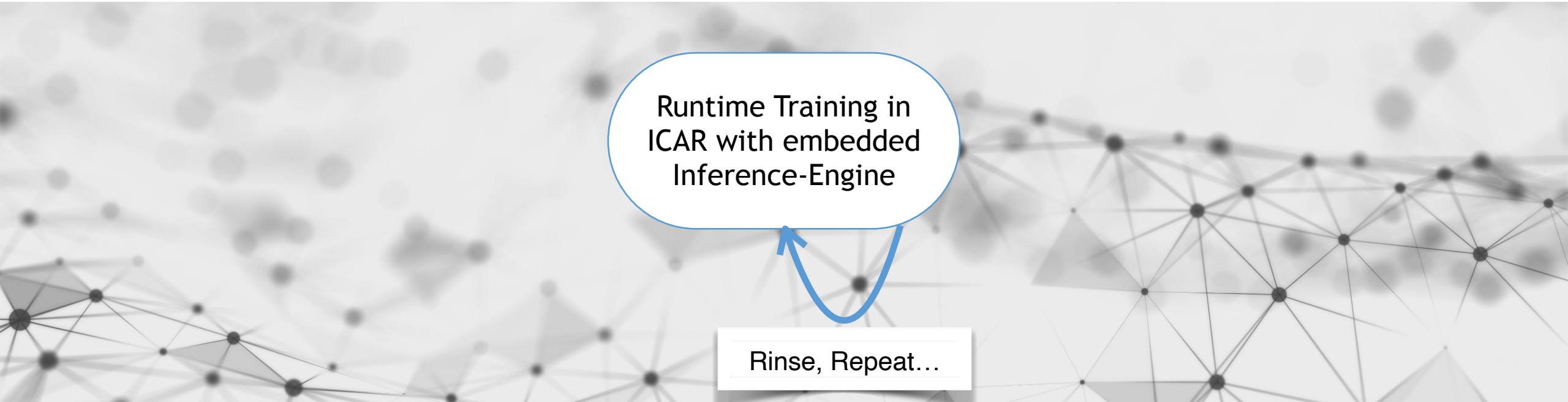


The Telephone Game

Solution Time = Development Time + Runtime



Sustainable Workflow



Software Developed with CLaSS

- LLVM Flang, Caffeine, Matcha, Inference-Engine
- GASNet-EX, UPC, UPC Runtime, UPC++
- MetaHipMer, SIMCoV, symPACK
- Berkeley Quantum Synthesis Toolkit (BQSKit)

Thoughts on Sustainability

- Socially sustainable development
- Technologically sustainable development
- At the intersection of the social and the technological

Conclusions

Conclusions

- The CLaSS Group co-develops open-source software:
  1. Parallel programming compilers, runtime libraries, networking middleware,
  2. An optimizing quantum synthesis toolkit,
  3. HPC deep learning tools, and
  4. Some targeted applications that use the above.
- Our socially sustainable development practices include
  1. Broadening participation through workforce development programs, involving contributors with varied educational backgrounds, and lowering barriers to community maintenance
  2. Developing openly and collaboratively
  3. Training new entrants to the field.
- Our technologically sustainable practices include
  1. High levels of backwards compatibility.
  2. Minimizing maintenance costs for users and contributors.
- Exciting avenues for exploration lie at the intersection between the social and the technological.