DOE CSGF Application - Math/CS Track

Name: Louis Jenkins

Student Data

First name	*Middle name*	*Last name*
Louis	P	Jenkins

Previous names (ex: maiden name)

Email

louis.jenkins@pnnl.gov

Citizenship

Country of Birth: United States

U.S. Citizen: Yes

If no,

- *Country of Citizenship:*
- *Permanent Resident Alien:* ----

If yes,

- *PRA Number:*
- *Port of Entry:*

Name: Louis Jenkins

Current Mailing Address

Street address 1

60 Crittenden Blvd

Street address 2

Apt 110

City	State	Zip code
Rochester	**NY**	**14620**

Home phone	Cell phone	*Work phone*
	610-931-1207	

Address effective through (m/d/y):

After this date, all correspondence will be sent to the permanent address listed below unless otherwise requested.

Notify the Krell Institute if your address changes after the application has been submitted.

Permanent Address

Street address

60 Crittenden Blvd, Apt 110

City	*State*	*Zip code*
Rochester	**NY**	**14620**

Phone

610-931-1207

References

List at least three persons familiar with your academic preparation and your technical abilities.
Please have these individuals mail the reference forms directly to Krell Institute.

	Title	*First name*	*Last name*	*Institution*	*Email*	*Status*
1.		**Cliff**	**Joslyn**	**Pacific Northwest National Laboratory**	**Cliff.Joslyn@pnnl.gov**	**Submitted**
2.		**Andrew**	**Lumsdaine**	**Pacific Northwest National Laboratory**	**Andrew.Lumsdaine@pnnl.gov**	**Submitted**
3.		**Brad**	**Chamberlain**	**Cray Inc.**	**bradc@cray.com**	**Submitted**
4.		**Michael**	**Scott**	**University of Rochester**	**scott@cs.rochester.edu**	**Submitted**

Name: Louis Jenkins

Academic Status

Current Academic Status: **First-year Doctoral Student**

Have you completed any academic credit towards your computational science/engineering doctoral degree? **Yes**

If yes, how many terms have you completed? (exclude summer) **1 Semester**

Official transcripts from every listed institution are a required component of the application including your Fall 2018 transcript, if applicable.

Doctoral Institution (Institution where you plan on completing your computational science and engineering doctorate or first choice doctoral university):

Institution	***Start Date***	***Expected End Date***	***Department***	***Academic Discipline***	***GPA***	***Degree***
University of Rochester	09/2018	04/2023	Computer Science	Computer Science	3.85	PhD

Department Chair at Doctoral Institution:

First Name	***Last Name***	***Email***
Sandhya	Dwarkadas	sandhya@cs.rochester.edu

Other Doctoral Institution Choices (Answer only if not currently at doctoral institution)

			Department Chair Information	
Institution	***Department***	***Academic Discipline***	***Name***	***Email***

Higher Educational History (All university/colleges attended and degrees obtained with the exception of the doctoral degree listed above):

Institution	***Start Date***	***End Date Expected or Actual***	***Department***	***Academic Discipline***	***Degree***	***GPA***
Bloomsburg University	04/2012	12/2017	Mathematics and Computer Science	Computer Science	Bachelors	3.06
					None	
					None	
					None	
					None	

Name: Louis Jenkins

Graduate Advisor

The graduate advisor is the person from the preferred institution **who views and approves the Program of Study.**

First Name
Michael

Last Name
Scott

Institution
University of Rochester

Title (Dr., Ms., Professor, ...)
Dr

Email
michael.l.scott@rochester.edu

Address 1
3401 Wegmans Hall

Address 2

City
Rochester

State
NY

Zip Code
14627-0226

Telephone
(585) 275-7745, 5478

Fax
273-4556

Name: Louis Jenkins

Program of Study

Listed are the courses in science and engineering, applied mathematics, and computer science that you agreed to take on your proposed Program of Study.

University: University of Rochester

Course number	Course Title	Credit hours	Term and Year	Grade	Academic Level
Science/Engineering					
ECE 400	COMPUTER ORGANIZATION	4.0S	Spring 2021		G
ECE 401	ADVANCED COMPUTER ARCHITECTURE	4.0S	Fall 2020		G
Mathematics and Statistics					
DSC 401	TOOLS FOR DATA SCIENCE	4.0S	Spring 2020		G
DSC 462	COMPUTATIONAL INTRODUCTION TO STATISTICS	4.0S	Fall 2019		G
Computer Science					
CSC 455	Software Analysis and Improvement	4.0S	Spring 2019		G
CSC 458	PARALLEL & DISTRIBUTED SYSTEMS	4.0S	Spring 2019		G
CSC 595	PHD RESEARCH IN CSC	4.0S	Fall 2018		G

I have read this program of study and affirm that, in my opinion, it satisfies the fellowship program requirements. This POS has been approved by my advisor, **Michael Scott**, and I understand that, if offered a fellowship, my advisor and I are required to sign this page and send it to the Krell Institute.

Student's signature ___________________________ Date _______________

Graduate Advisor: **Michael Scott**
Graduate Advisor's Institute: **University of Rochester**
Graduate Advisor signature ___________________________ Date _______________

Krell Institute (Office use only) _________________________________

Krell Institute, Attn: DOE CSGF Coordinator
1609 Golden Aspen Drive, Suite 101, Ames, IA 50010
Phone: 515-956-3696, Fax: 515-956-3699, csgf@krellinst.org

Name: Louis Jenkins

Course Description

ECE 400: COMPUTER ORGANIZATION

Instruction set principles; processor design, pipelining, data and control hazards; datapath and computer arithmetic; memory systems; I/O and peripheral devices; internetworking. Students learn the challenges, opportunities, and tradeoffs involved in modern microprocessor design. Assignments and labs involve processor and memory subsystem design using hardware description languages (HDL).

ECE 401: ADVANCED COMPUTER ARCHITECTURE

Instruction set architectures. Advanced pipelining techniques Instruction level parallelism. Memory hierarchy design. Multiprocessing. Storage systems. Interconnection network.

DSC 401: TOOLS FOR DATA SCIENCE

This course provides a hands-on introduction to widely-used tools for data science. Topics include Linux; languages and packages for statistical analysis and visualization; cluster and parallel computing using Hadoop and Spark; libraries for machine learning; no-sql data stores; and cloud services.

DSC 462: COMPUTATIONAL INTRODUCTION TO STATISTICS

This course will cover foundational concepts in probability and statistical inference, with an emphasis on topics of interest to computer scientists. Following an introduction to elementary probability theory, topics will include applications of combinatorics; Markov chains; principles of statistical classification (Bayes' rule, sensitivity and specificity, ROC curves) and random number generation. The theory of statistical estimation and hypothesis testing will be introduced, and applied to one and two sample inference for population means, proportions, variances and correlations. Nonparametric procedures will be discussed. Topics also include statistical modeling (ANOVA, simple and multiple regression), and computational methods.

CSC 455: Software Analysis and Improvement

Programming is the automation of information processing. Program analysis and transformation is the automation of programming itself---how much a program can understand and improve other programs. Because of the diversity and complexity of computer hardware, programmers increasingly depend on automation in compilers and other tools to deliver efficient and reliable software. This course combines fundamental principles and (hands-on) practical applications. Specific topics include data flow and dependence theories; static and dynamic program transformation including parallelization; memory and cache management; type checking and program verification; and performance analysis and modeling. The knowledge and practice will help students to become experts in software performance and correctness

CSC 458: PARALLEL & DISTRIBUTED SYSTEMS

Principles of parallel and distributed systems, and the associated implementation and performance issues. Topics covered will include programming interfaces to parallel and distributed computing, interprocess communication, synchronization, and consistency models, fault tolerance and reliability, distributed process management, distributed file systems, multiprocessor architectures, parallel program optimization, and parallelizing compilers. Students taking this course at the 400 level will be required to complete additional readings and/or assignments. (Teaching Assistant)

CSC 595: PHD RESEARCH IN CSC

Advisor: Michael Scott; Topic: Creation of an analysis/testing tool for persistent (non-volatile) memory systems with volatile CPU caches. Idea is to simulate power-failure and help stress-test durably linearizable algorithms to assist in finding subtle bugs and correctness issues. Designed to simulate 'non-coherent' snapshots of memory that will represent the state of the application following a crash (where volatile registers gets wiped along with write-back cache, while the data in non-volatile memory persists); Durable Linearizable programs must be consistent following a crash, and so this provides an empirical approach to verification via random sampling while also upholding happens-before relationships on stores, flushes, and memory fences. Very experimental.

Name: Louis Jenkins

Other Planned Courses

Listed are the other courses you plan to take that you believe are particularly pertinent to your proposed or current research in the areas of Mathematics, Science and Engineering, and Computer Science.

Course number	Course Title	Credit hours	Term and Year	Grade	Academic Level
Computer Science					
CSC 442	Artificial Intelligence	4.0S	Fall 2018		G
CSC 446	Machine Learning	4.0S	Spring 2019		G
CSC 480	COMPUTER MODELS & LIMITATIONS	4.0S	Spring 2019		G
CSC 481	INTRO TO CRYPTOGRAPHY	4.0S	Fall 2019		G
CSC 573	Synthesis of Systems	4.0S	Fall 2018		G

Name: Louis Jenkins

Course Description

CSC 442: Artificial Intelligence

This is the graduate level Artificial Intelligence course. Introduces fundamental principles and techniques from Artificial Intelligence, including heuristic search, automated reasoning, handling uncertainty, and machine learning, to prepare students for advanced AI courses.

CSC 446: Machine Learning

Mathematical foundations of classification, regression, and decision making. Perceptron algorithm, logistic regression, and support vector machines. Numerical parameter optimization, including gradient descent and quasi-Newton methods. Expectation Maximization. Hidden Markov models and reinforcement learning. Principal Components Analysis. Learning theory including VC-dimension and PAC learning guarantees.

CSC 480: COMPUTER MODELS & LIMITATIONS

This course studies fundamental computer models and their computational limitations. Finite-state machines and pumping lemmas, the context-free languages, Turing machines, decidable and Turing-recognizable languages, undecidability.

CSC 481: INTRO TO CRYPTOGRAPHY

The modern study of cryptography investigates techniques for facilitating interactions between distrustful entities. In this course we introduce some of the fundamental concepts of this study. Emphasis will be placed on the foundations of cryptography and in particular on precise definitions and proof techniques.

CSC 573: Synthesis of Systems

An implementation-oriented survey of software systems that specify, model and synthesize software/hardware systems. The complexity of modern computer architectures places a heavy manual burden on the programmer to write correct, efficient and fast programs. This course investigates programming language, compiler and automated reasoning techniques that transfer this burden to synthesis systems which aim to reduce the burden on programmers by generating code from high-level specifications.

Name: Louis Jenkins

Completed Courses

Please list up to six courses you have completed that are particularly pertinent to your proposed or current research in the areas of Mathematics, Science and Engineering, and Computer Science. Please do not list entry level science/engineering or mathematics courses like Calculus I.

Course number	Course Title	Credit hours	Term and Year	Grade	Academic Level
COMPSCI 350	Organization of Programming Language	4.0S	Fall 2016	A-	U
COMPSCI 355	Algorithms & Data Structures	4.0S	Fall 2016	A-	U
COMPSCI 386	Operating Systems	4.0S	Spring 2017	A	U
COMPSCI 456	Theory of Computation	4.0S	Spring 2017	B+	U
CSC442	Artificial Intelligence	4.0S	Fall 2018	A	G
CSC573	Synthesis of Systems	4.0S	Fall 2018	A	B

Name: Louis Jenkins

Course Description

COMPSCI 350: Organization of Programming Language

This course is designed to enable the student to appreciate the role of programming languages in the programming process. It introduces the formal study of programming languages and the rationale for such study. Students will learn the variety of programming language constructs that have been implemented and their advantages and disadvantages. Students will also be able to critically evaluate particular programming language constructs. Three hours of lecture per week.

COMPSCI 355: Algorithms & Data Structures

This course gives a detailed look at different algorithms and data structures. The students will also examine the performance of various algorithms and data structures. Particular attention is paid to algorithms for searching and sorting, graph algorithms, and techniques for implementing and manipulating various data structures using object-oriented programming techniques. NP-completeness will also be introduced. Three hours of lecture per week.

COMPSCI 386: Operating Systems

A study of the foundations of modern operating systems along with the concurrent programming problems associated with these systems. Students will write and test their own solutions to some of these problems. Problems encountered in the development of all modern systems will be covered along with possible solutions. Three hours of lecture per week.

COMPSCI 456: Theory of Computation

Presents an introduction to automata, formal languages computability and computational complexity. Topics include finite automata, pushdown automata, context-free grammars, Turing machines, and algorithmically unsolvable and computationally intractable problems. The course is intended for students interested in computer science theory and is cross-listed in Mathematics and Computer Science. Three hours lecture per week

CSC442: Artificial Intelligence

This is the graduate level Artificial Intelligence course. Introduces fundamental principles and techniques from Artificial Intelligence, including heuristic search, automated reasoning, handling uncertainty, and machine learning, to prepare students for advanced AI courses.

CSC573: Synthesis of Systems

An implementation-oriented survey of software systems that specify, model and synthesize software/hardware systems. The complexity of modern computer architectures places a heavy manual burden on the programmer to write correct, efficient and fast programs. This course investigates programming language, compiler and automated reasoning techniques that transfer this burden to synthesis systems which aim to reduce the burden on programmers by generating code from high-level specifications.

Name: Louis Jenkins

Research Statements

This information is vital to the overall evaluation of your application.

Field of Interest

a. *In terms a general audience would understand, describe an important, outstanding problem in computer science, applied mathematics or statistics that you would like to pursue in your research. (1/3)*
b. *Discuss the potential impact of your research on your field. How would it advance high performance computing in general? (1/3)*
c. *How would it advance a science or engineering application area or areas? (1/3)*

Fault tolerance is a fundamental issue in High Performance Computing, and while the challenges are exacerbated when computations are distributed across multiple compute nodes, software-based solutions are achievable. The goal of my research is to design an automated approach for providing fault tolerance for user applications, perhaps via application of program synthesis, an application of machine learning that synthesizes programs from high level specifications, or regular compile-time transformation. In distributed computing, the Partitioned Global Address Space (PGAS) memory model allows for accesses of memory that is 'remote', or allocated on another compute node, via one-sided Remote Direct Memory Access (RDMA) operations known as PUT and GET. Chapel, Cray's exascale and PGAS language, created as part of DARPA's High Productivity Computing Systems program, is highly receptive to change and can serve as an ideal test bed for research and experimentation. As a bonus, any solutions designed for Chapel may be generalizable enough to be implemented in other PGAS languages, or to expose new paradigms for software-based fault tolerance in distributed computing that may work even under MPI.

By implementing some kind of transparent or at least highly-abstracted mechanism for fault tolerance in Chapel, we may lower the bar for creating truly fault-tolerant systems. Chapel already makes it easier to write high-performance and distributed-memory programs, making fundamental tasks implicit, (e.g., abstracting PUT and GET operations via compile-time transformation of loads and stores to remote memory). It seems fitting that implicit fault tolerance be paired with implicit high-performance. This would make it easier for new programmers to enter the field of HPC and, for both old and new programmers alike, to write HPC applications.

Besides for just "lowering the bar", implicit fault tolerance can lower the complexity of software systems, and more importantly can make them more portable; systems with non-volatile memory can make use of automatic transformation into durable linearizable programs, while those without could fall-back to checkpointing or other types of fault tolerance techniques. The sky is the limit!

High Performance Computing

a. *Discuss the role of high performance computing in your research. (1/2)*
b. *How will you demonstrate the success of your research? (1/2)*

My aim is to make it easier to build fault-tolerant HPC software. Scientific progress would no longer be limited by the excess requirement of pre-requisite knowledge of fault-tolerance, nor will it be impeded by the non-trivial amount of time needed to design such systems from the ground-up. As well, the main benefactors of such results are programs consisting of long-running computations, which are common in High-Performance and Scientific Computing. Finally, fault tolerance is an inherent problem in High-Performance Computing, and hence is the at the core of my research.

Chapel is an innovative language of astounding amounts of potential, aiming to provide the simplicity and ease-of-use of Python, the portability of Java, the longevity of C, performance that is comparable and on-par to other languages that focus on parallelism such as Go and Julia, and support for distributed computing to boot. However, despite its potential, Chapel is a very ambitious project, and so it faces problems that no other language has faced before; challenges that requires resources that it does not have. When conversing with Brad Chamberlain, technical lead of the Chapel programming language, about what is keeping the language from solving these challenges, his response boiled down to "time and resources". Success in research can help solve these problems, and work towards the language can help vastly improve the language as well as its appeal to the general public. I firmly believe that Chapel will one day become a dominant language in the realm of HPC and that PGAS will replace MPI-based technologies, but even in the case that I am wrong, the solutions themselves may prove to be of use outside of Chapel and/or PGAS.

Program of Study

Describe how the courses listed in your planned program of study would help prepare you to address the challenges you have described in questions 1 and 2. Discuss your rationale for choosing these courses. How will the science or engineering application courses you have selected impact your research?

Research Assistantship (CSC595): Began creating and designing a tool that models the problem of having non-volatile RAM (NVRAM) under volatile CPU Cache and Registers. During its design, I have learned about a new correctness condition for fault tolerance under a full-system crash model, durable linearizability. It is predicted that in the next few years, many systems will have NVRAM + Volatile Cache, meaning this could be seen as a headstart for studying and researching fault tolerance.

Parallel and Distributed Systems (CSC448): Being a TA for such a course, taught by my advisor the Michael Scott, will help me both increase my breadth and depth of knowledge necessary to teach such concepts, and to enhance my communication skills when portraying such concepts.

Software Analysis and Improvement (CSC455): Learning the essentials of compilation techniques and optimization will invariably help in finding automated, compile-time solutions for fault tolerance.

Tools for Data Science (DSC401): Data Science is a very large part of High-Performance Computing, and so becoming more familiar with such tools will be of great benefit to me.

Computational Introduction to Statistics (DSC462): Following from Artificial Intelligence and Machine Learning courses, going more in depth with statistics means I will be able to better analyze data and interpret results. As well, as I have seen in my Synthesis of Systems course, machine learning can be applied to program synthesis, that is synthesizing programs given some specification. If the specification describes fault tolerance, it could be very helpful.

Computer Organization (ECE400): I will benefit from knowing more about the hardware and its limitations when designing solutions, even from the perspective of the software-level.

Advanced Computer Architecture (ECE401): Greater understanding of hardware, greater perspective.

Name: Louis Jenkins

Programming Languages and Models

List (four at most) the programming languages and programming models with which you have experience. Provide a sentence that describes how you use them.

1. Programming Language/Model: **Chapel**
Contracted specialist for DoE's PNNL, Contributor to programming language, and have conducted research and published work on Chapel. Planned as primary language of research.

2. Programming Language/Model: **Partitioned Global Address Space (PGAS)**
In relation to Chapel, the PGAS language.

3. Programming Language/Model: **C/C++**
Proficient, general-purpose and fallback language. Chapel's runtime written in C (Compiler in C++), GASNet and uGNI PGAS communication layer written in C; useful for everything.

4. Programming Language/Model: **LLVM (Intermediate Representation)**
LLVM is the backend for Chapel and many other languages, as well for the clang compiler. Needed to perform compiler optimization; program synthesis can also be performed at this level.

What are the programming languages that you intend to use in your research?

Chapel, C, C++, LLVM (Intermediate Representation)

Name: Louis Jenkins

List of Publications

Papers

Louis Jenkins, Tingzhe Zhou, and Michael Spear. "Redesigning Go's Built-In Map to Support Concurrent Operations." 26th International Conference on Parallel Architectures and Compilation Techniques (PACT), IEEE, 2017.

Louis Jenkins. "RCUArray: An RCU-Like Parallel-Safe Distributed Resizable Array." 2018 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW). IEEE, 2018.

Louis Jenkins, Marcin Zalewski, and Michael Ferguson. "Chapel Aggregation Library (CAL)." To Appear @ "Supercomputing 2018: Parallel Applications Workshop - Alternatives to MPI"

Louis Jenkins, et al. "Chapel HyperGraph Library (CHGL)." 2018 IEEE High Performance Extreme Computing Conference (HPEC). IEEE, 2018.

Cliff Joslyn, [...], Louis Jenkins, et al. "High Performance Hypergraph Analytics of Domain Name System Relationships" To Appear @ "Hawaii International Conference on System Sciences: Symposium on Cybersecurity Big Data Analytics"

Talks

Posters

Louis Jenkins, et al. "Chapel HyperGraph Library (CHGL)", presented at Richland, WA, URL: http://louisjenkinscs.github.io/posters/CHGL.pdf

Name: Louis Jenkins

Laboratory and Research Experience/Other Employment

Begin with current or most recent employment. Please include employer, dates employment started and ended, position, and nature of work.

Research Assistant @ University of Rochester, Fall 2018 --
PhD Intern @ Pacific Northwest National Laboratory, Fall 2018 --
Tech Intern IV @ Pacific Northwest National Laboratory, May 2018 - Aug 2018
Student Coder @ Google Summer of Code (Chapel, Cray Inc.), April 2017 - Aug 2017
Undergraduate Research Assistant (NSF sponsored REU) @ Lehigh University, May 2016 - Aug 2016

Academic Awards and Honors - *Include undergraduate and graduate honors (if applicable).*

Peer's Choice for Outstanding Project @ Lehigh University (2016)
Honorable Mention @ Computing Research Association's Outstanding Undergraduate Researcher's Competition, 2017
Dean's List @ Bloomsburg University, Spring 2014, Fall 2015, Spring 2017

Extracurricular Activities - *Include technical societies and service organizations.*

Rail's Girls Summer of Code Coach for Chapel Organization, July - September 2018

Name: Louis Jenkins

Additional Comments

Practicum planned to continue working with Chief Scientist Andrew Lumsdaine of Northwest Institute of Advanced Computing (NIAC) and continued collaboration with Chief Scientist Cliff Joslyn of Knowledge Sciences on HPDA project. Currently working as PhD Intern for 8 hours per week for PNNL (https://www.pnnl.gov/science/staff/staff_info.asp?staff_num=9451) on the "Chapel HyperGraph Library (CHGL)" (https://www.osti.gov/doecode/biblio/18401)

Website: LouisJenkinsCS.github.io
GitHub: GitHub.com/LouisJenkinsCS

Name: Louis Jenkins

DOE CSGF and Other Fellowships

The information that you provide will allow us to target our advertising more effectively. This information is confidential and is not used in review of the fellowship application.

1. Have you applied to other fellowship programs?

- ☐ DOE NNSA SSGF
- ☑ NSF
- ☐ DOD
- ☑ University-sponsored   Names of fellowships: **University of Rochester's Provost Fellowship**
- ☐ Other   Names of fellowships:

2. How did you find out about the program?

- ☐ DOE CSGF poster
- ☐ DEIXIS, DOE CSGF annual publication
- ☐ Attended DOE CSGF talk
- ☐ Advertisement   Name the source:
- Word of mouth from
  - ☑ faculty
  - ☐ student
  - ☑ administrator
- ☑ Laboratory staff
- ☐ Institutional announcement
- ☐ Conference or meeting   Name:
- ☐ World Wide Web   List URL:
- ☐ Other   Explain:

Applicant Demographics

Applicant data is important in assessing the effectiveness of our efforts to solicit applications from a diverse population. Providing the information on this form is voluntary; omission of information will not affect any decision about your application. We appreciate your cooperation.

Race:	**African American**
Gender:	**Male**
Physical/mental disability:	**No**