# Department of Computer Science

## Texas Southern University

## MONTHLY RESEARCH SEMINAR

# Bounds Check Elimination for High Level Arrays

by

**Mackale Joyner, PhD**
Texas Instruments

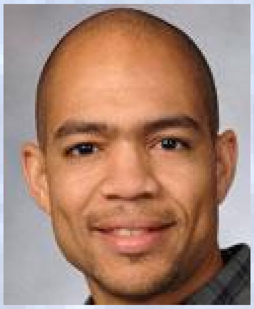
**Wednesday October 23,**
**3:00 PM- 4:00 PM, Nabrit Science 106**

## Biography

Mackale Joyner is a Software Engineer in the Compiler group at Texas Instruments. He completed his PhD at Rice University in 2008. He worked under Ken Kennedy and Vivek Sarkar in the Compiler group from 2002-2008. His research focused on developing compiler strategies for high-productivity languages in domain-specific environments. He believes that both crafting and optimizing high-level languages for domain-specific environments enables the broad community to effectively utilize these languages to develop efficient applications.

## Abstract

For decades, the design and implementation of arrays in programming languages has reflected a natural tension between productivity and performance. Recently introduced HPCS languages (Chapel, Fortress and X10) advocate the use of high-level arrays for improved productivity. For example, high-level arrays in the X10 language support rank-independent specification of multidimensional loop and array computations using regions and points. Three aspects of X10 high-level arrays are important for productivity but pose significant performance challenges: high-level accesses are performed through point objects rather than integer indices, variables containing references to arrays are rank-independent, and all subscripts in a high-level array access must be checked for bounds violations. The first two challenges have been addressed in past work. In this talk, we address the third challenge of optimizing the overhead of array bounds checks by developing a novel region-based interprocedural array bounds analysis to automatically identify redundant checks. Elimination of redundant checks reduces the runtime overhead of bounds checks, and also enables further optimization by removing constraints that arise from precise exception semantics. We have implemented an array bounds check elimination algorithm that inserts special annotations that are recognized by a modified JVM.
We also introduce array views, a high-level construct that improves productivity by allowing the programmer to access the underlying array through multiple views. We describe a technique for optimizing away the overhead of many common cases of array views in X10. Our experiments show that eliminating bounds checks using the results of the analysis described in this paper improves the performance of our benchmarks by up to 22% over JIT compilation.



**Mackale Joyner, PhD**