

X10 Innovation Awards

X10 is a new programming language being developed at IBM Research in collaboration with academic partners. The X10 effort is part of the IBM PERCS project (Productive Easy-to-use Reliable Computer Systems) in the DARPA program on High Productivity Computer Systems.

In late 2009, IBM Research announced the 2010 X10 Innovation Award Program. On behalf of IBM Research and the X10 Programming Language Team, we are pleased to present the 18 winners of the 2010 X10 Innovation Award below. More recently, we have also awarded 12 new Innovation Awards focused on Asia.

[X10 Day, Friday, April 16, 2010 in Hawthorne, NY.](#)

[X10 Days, December 4-5, 2010 at Tsinghua University, Beijing, China.](#)

X10 Application Development

- [Dynamic Graph Data Structures in X10](#), David Bader, Georgia Tech
- [X10 Media Composer \(XMC\)](#), Admela Jukan, University of Braunschweig, Germany
- [Leveraging X10 for Open-Source, Platform-Level, Cloud Computing](#), Chandra Krintz, UC Santa Barbara
- [#Systematic Testing in and for X10](#), Darko Marinov, UIUC
- [Fine-grained Concurrent Compilation with X10](#), Nathaniel Nystrom, UT Arlington
- [Using X10-enabled MapReduce Framework for Intercomparison of Satellite Aerosol Data Analysis](#), Yelena Yesha, U. Maryland
- [MapReduce framework on heterogeneous cluster](#), Yu Wang, Tsinghua University, China
- [Dynamic programming algorithms in X10](#), Guangwen Yang, Tsinghua University, China
- [Financial Risk Computing using X10](#), Hui Liu, Shandong University, China
- [Stream programming on X10](#), Junqing Yu, Huazhong University of Science and Technology, China
- [Exploiting concurrency in large-scale botnet detection](#), Shishir Nagaraja, Indraprastha Institute of Technology, India

X10 Tool Development

- [#Prototyping a Novel Performance Toolset for X10 Applications](#), Alan George, University of Florida
- [#Optimization and Verification of X10 Programs](#), Jens Palsberg, UCLA
- [Lightweight deterministic replay of X10 programs](#), Charles Zhang, The Hong Kong University of Science and Technology, China
- [Checking serializability consistency for concurrent X10 programs](#), Jianjun Zhao, Shanghai Jiao Tong University, China

Framework/Libraries

- [Transactions for Reliable Distributed X10 Execution](#), Anthony Hosking, Purdue University
- [#Automatic Adaptation of Java Frameworks for X10 to Improve Programmer Productivity](#), Eli Tilevich, Virginia Tech
- [High performance clustering algorithms libraries in X10](#), Xiaoyun Chen, Lanzhou University, China

Mapping High-Level Languages to X10

- [Embracing the Parallelism of Forward Linear Logic Programming](#), Frank Pfenning, CMU
- [Scripting on X10 - Ruby Script to X10 Program Translator](#), Koichi Sasada, University of Tokyo, Japan

Language Research

- [Concurrency Types for X10: Race-Freedom, Atomicity, and Determinism](#), Cormac Flanagan, UCSC
- [Improving performance of the X10 runtime for multi-core](#), Haibo Chen, Fudan University, China

Curricula Development

- [A First Programming Course using X10](#), Robert (Corky) Cartwright/Vivek Sarkar, Rice University
 - [Teaching Real-World Parallel and Distributed Programming](#), Steven Reiss, Brown University
 - [#Migrating Undergraduate Parallel Computing Courses to X10](#), Claudia Fohry, University of Kassel, Germany
 - [#X10 Tutorial and Workshop Development](#), Steven Gordon/Dave Hudak, Ohio Supercomputing Center
 - Teaching Parallel Programming as a Pattern Language on the Example of X10, Christoph von Praun, Georg-Simon-Ohm University, Germany
 - [Astronomical research and parallel computing course](#), Ce Yu, Tianjin University, China
 - [Parallel computing and programming in X10 \(graduate\)](#), Shaohua Liu, Beijing University of Posts and Telecommunications, China
 - [Courseware for an advanced undergraduates CS course](#), Xiaoge Wang, Tsinghua University, China
-

Project Descriptions

- **Dynamic Graph Data Structures in X10**
David Bader, Georgia Tech (X10 team contact: Vijay Saraswat)
- **X10 Media Composer (XMC)**
Admela Jukan, University of Braunschweig, Germany (X10 team contact: Vijay Saraswat)
- **Leveraging X10 for Open-Source, Platform-Level, Cloud Computing**
Chandra Krintz, UC Santa Barbara (X10 team contact: Steve Fink)
We will integrate X10 into AppScale – an open-source emulation of the PaaS (Platform-as-a-Service) cloud APIs of Google App Engine. This will include integrating X10 (a) as a front-end development language (an alternative to Python and Java) that facilitates easy development of parallel and distributed applications as user-facing computations that work well in the cloud; (b) as a Tasks API language which will facilitate type-safe, parallel programming of web applications and services for efficient and scalable background computing; (c) as a language users employ for writing their own mappers and reducers, as well as the language for implementation of MapReduce in the AppScale Backend; and (d) as the implementation language for the parallel and concurrent activities within AppScale itself (for MapReduce, request handling, data processing, and other services within the AppScale control center). We also will extend the AppScale debugging and testing system with support for X10; and we will extend scheduling support in AppScale to handle execution of X10 tasks in parallel in support of the front end web service (in addition to scheduling other tasks, MapReduce jobs, internal components, database replicas, and front-end instances). As part of these pursuits, we will investigate application-specific compiler and runtime optimizations for performance, efficiency, and scaling, scheduling and load balancing techniques across both single and multiple machines.
- **MapReduce framework on heterogeneous cluster**
Yu Wang, Tsinghua University, China
In this project we propose a new MapReduce framework implemented in X10, which can be a general solution to accelerate MapReduce algorithms and provide both efficient hardware architecture and an interface to the OS. This framework is constructed by multi-node heterogeneous cluster, where each node is a reconfigurable heterogeneous multi-core system, including multi-core CPU, FPGA and GPU. The “mapper” and “reducer” can be implemented by these computing systems, and they are configurable. The proposed MapReduce framework can act as a common platform for research in the parallel computing area.
- **Dynamic programming algorithms in X10**
Guangwen Yang, Tsinghua University, China
In this project, we aim to design efficient dynamic programming algorithms (DP) in X10 on modern multi-core processors and distributed systems. At the same time, we will compare these algorithms in X10 to the same algorithms in other languages, in order to quantify the performance differences between X10 and other languages. In detail, we will first use X10 and use existing languages such as C++, Pthread and MPI to design efficient implementations of parallel DP algorithms. All implementations are aimed to efficiently utilize

all levels of parallelism on modern multi-core processors and distributed systems, including instruction pipelines, SIMD capability, multiple cores, memory-level parallelism, multiple nodes, etc. Second, we will compare the implementations in X10 to the implementations in other languages, in order to quantify the differences between X10 and other languages in terms of performance. Third, we will try to propose and design some programming tools for bioinformatics in X10, such as efficient task scheduling between multiple cores and multiple nodes.

- **Financial Risk Computing using X10**

Hui Liu, Shandong University, China

We aim to use the X10 language to develop parallel programs in the financial computing area, based on the mathematical models and the numerical computing algorithms for financial derivatives pricing and financial risk measurement. The mathematical models include both typical, frequently-used ones and those developed by ourselves (such as the BSDE-binomial tree model and BSDE-theta scheme model). We plan to construct a problem solving environment and develop scientific computing applications for financial risk measurement in X10. Moreover, in the experimental training for the third- or four- year students, we will teach parallel programming in X10.

- **Stream programming on X10**

Junqing Yu, Huazhong University of Science and Technology, China

Streaming programming has been productively applied in the signal processing, graphics and multimedia domains. X10 provides a flexible programming model for stream programs to exploit multi-grained parallelism and locality. In this project, we propose a compiler technology for planning and executing the stream programs (SPL) on X10. We first plan to build a compiler framework to translate stream programs to X10. Then we plan to design a schedule method to exploit the multi-grained parallelism and locality for stream programs based on the X10 execution model.

- **Exploiting concurrency in large-scale botnet detection**

- **Systematic Testing in and for X10**

Darko Marinov, UIUC (X10 team contact: Steve Fink)

New parallel programming languages, such as X10, make it easier to develop more reliable parallel code. However, despite the advances in parallel programming languages, developing parallel code remains challenging, with common bugs including atomicity violations, data races, and deadlocks. Software testing remains the most widely used approach for finding such bugs. This project proposes to parallelize systematic testing *in* and *for* X10. First, we plan to develop *in* X10 a parallel state-space exploration framework, which we will initially use for Java programs. Second, we plan to instantiate this framework *for* X10 programs, i.e., for systematic testing of programs written in X10. The benefits of the project are twofold. First, it will provide a state-space exploration framework in X10, which is interesting code in itself, e.g., as a performance benchmark. Second, it will provide a testing tool for X10, which can help in developing more reliable X10 programs and in teaching about X10.

- **Fine-grained Concurrent Compilation with X10**

Nathaniel Nystrom, UT Arlington (X10 team contact: Igor Peshansky)

As a demonstration of the utility of the X10 concurrency model, this research project will implement an extensible, fine-grained parallel compiler for Java using the X10 concurrency model. The compiler will leverage the Polyglot compiler framework and the X10 runtime library, producing a concurrent version of Polyglot, thus parallelizing an existing sequential application with complex data structures and complex dependencies between data structures.

- **Using X10-enabled MapReduce Framework for Intercomparison of Satellite Aerosol Data Analysis**

Yelena Yesha, U. Maryland (X10 team contact: David Grove)

We propose to deploy the X10 parallel programming paradigm to implement the functionality of the MapReduce framework for executing an intensive scientific data processing application running on a hybrid cloud computing cluster environment consisting of multiple computer architectures: IBM JS (Power6 or PowerPC), QS (Cell B.E.), and HS (Intel Nehalem) blades. We also will plan to run on a large remote homogeneous IBM BlueGene P system. The aerosol satellite data is unstructured and non-uniform, which is an excellent use case for utilizing the X10 async feature to dynamically balance the computing load. In addition, X10 programming will be taught in the advanced parallel programming course.

- **Prototyping a Novel Performance Toolset for X10 Applications**

Alan George, University of Florida (X10 team contact: Evelyn Duesterwald)

Given the importance of performance tools for productive HPC development, this project aims to explore the challenges involved in enabling experimental performance analysis of X10 applications. Our exploration will leverage the existing Parallel Performance Wizard (PPW) system, one of the foremost performance tools that supports Partitioned Global-Address-Space (PGAS) programming models, and the Global Address Space Performance (GASP) interface, which specifies the interaction between a performance tool and a PGAS programming model implementation. By extending PPW and GASP to support X10's unique features, we aim to provide a prototype performance tool supporting X10 application analysis. Such a tool would be of substantial benefit to X10 application developers, providing them with a variety of capabilities to capture and understand program performance in terms of X10 constructs. Given X10's provisions for high-level parallelism, with abstractions that may often hide the potential performance impact of some language constructs and thus make manual performance monitoring difficult, it is particularly important for application developers to have access to tools that automatically collect and provide views into X10 application performance data. Successful completion of this project would comprise substantial progress toward bring much-needed performance tool capabilities to the world of X10 application development.

- **Optimization and Verification of X10 Programs**

Jens Palsberg, UCLA (X10 team contact: Olivier Tardieu)

We want to provide optimization and verification techniques for parallel programs that are on par with today's standards for sequential programs. We believe that the key enabler is to raise the level of abstraction of parallel programming.

We study parallel programs in the context of X10, a parallel language designed at IBM. Two of X10's key constructs for parallelism are `async` and `finish`. The `async` statement is a lightweight notation for spawning threads, while a `finish` statement (`finish s`) waits for termination of all `async` statement bodies started while executing the statement `s`. Additionally, X10 supports multidimensional distributed arrays.

We have rewritten a state-of-the-art plasma simulation program from Fortran 95 to X10. We have designed a core calculus with `async` and `finish` and shown that the semantics enables manageable proofs of key properties. We have designed a context-sensitive may-happen-in-parallel analysis that gives precise static information about the parallel behavior of a program. We have built a compiler for a subset of X10 1.5 and shown that for our plasma simulation program, our compiler produces code that is about 5,000 times faster than the code produced by IBM's X10 1.5 compiler.

- **Lightweight deterministic replay of X10 programs**

Charles Zhang, The Hong Kong University of Science and Technology, China

The proposed project enhances the debuggability of the X10 programs by designing, implementing, and evaluating a lightweight deterministic record and replay technique that transparently works with X10 programs through the compiler-based program analysis and instrumentation. We seek both the theoretical analysis and the pragmatic treatments that faithfully produce problematic executions of X10 programs by effectively regulating all of the random inputs to the program. In addition, we plan to achieve practicality by designing a technique that incurs low perceivable runtime footprint and requires the minimum programmer intervention.

- **Checking serializability consistency for concurrent X10 programs**

Jianjun Zhao, Shanghai Jiao Tong University, China

Although many efforts have been made to improve the memory model and synchronization, atomicity violations that originate from buggy declarations of atomic blocks may still occur in X10 programs. The objective of this project is to develop a static approach to effectively and efficiently detect atomicity violations for X10 programs. In order to classify both “must” and “may” atomicity violations, we improve unserializable patterns of shared-memory accesses through considering the dependencies between the successive shared-memory accesses. We plan to devise a static analysis to produce sound detection results and eliminate false positives as much as possible. We will also utilize reduction techniques to guarantee the scalability of our approach. The benefits of this project are as follows: First, it provides debugging support for X10 programs through the detection of atomicity violations. Second, since our approach does not rely on the declarations of atomic blocks, it can be applied to infer atomic blocks or eliminate unnecessary atomic blocks. Third, inter-activity data-flow analysis is extremely difficult due to the explosion of state space of activity interleavings; this project partly concerns inter-activity data-flow analysis, and therefore has the potential to provide experience or theoretical foundation in this field. Finally, our approach can be integrated with testing to provide more effective and efficient debugging support for X10 programs.

- **Transactions for Reliable Distributed X10 Execution**

Antony Hosking, Purdue University (X10 team contact: Vijay Saraswat)

- **Automatic Adaptation of Java Frameworks for X10 to Improve Programmer Productivity**

Eli Tilevich, Virginia Tech (X10 team contact: Igor Peshansky)

The stated design goal for X10 is to improve development productivity for parallel applications. When it comes to constructing advanced cloud applications, X10 must provide facilities to systematically express essential non-functional concerns, including persistence, transactions, distribution, and security. In a cloud-based application, non-functional concerns may constitute intrinsic functionality, but implementing them in X10 can be non-trivial. To streamline the implementation of non-functional concerns, the Java development ecosystem features standardized, customizable, and reusable abstractions called frameworks. Java frameworks are a result of a concerted, cooperative, and multi-year effort of multiple stakeholders in the Java technology, which has been tested and proven effective by billions of lines of production code. This project will explore how Java frameworks can be automatically adapted for use by X10 programmers. The proposed approach will help avoid duplicating the effort expended on creating Java frameworks to provide equivalent facilities for X10. By enabling X10 programmers to leverage the collective expertise of the developers and users of Java frameworks, this project aims at improving programmer productivity.

- **High performance clustering algorithms libraries in X10**

Xiaoyun Chen, Lanzhou University, China

This project aims to research and improve data clustering algorithms, which will be developed as application program class libraries using X10. Class libraries developed in X10 are to be divided into two types, one is for classical clustering algorithms, and the other is for clustering algorithms improved by ourselves. Each class library can be discriminated and refined according to the data object to be processed, such as spatial or time-series data, stream data and graph data, etc. To verify the practicability, scalability, robustness and performance of equivalent algorithms from different class libraries, data from different fields, such as GIS, medicine, bioinformatics etc, need to be considered consistently.

- **Embracing the Parallelism of Forward Linear Logic Programming**

Frank Pfenning, CMU (X10 team contact: Vijay Saraswat)

- **Scripting on X10 - Ruby Script to X10 Program Translator**

Koichi Sasada, University of Tokyo, Japan (X10 team contacts: Tamiya Onodera, David Grove)

This research project will explore technologies required for translating Ruby, a scripting language well known for its high productivity, to X10, a parallel programming language. Especially, we will study the following two

topics: (1) a method to translate a Ruby script to an X10 program, and (2) extending X10 with Ruby's high-level features. The goal of this research is to develop a highly consumable parallel language for the Multicore era.

- **Concurrency Types for X10: Race-Freedom, Atomicity, and Determinism**

Cormac Flanagan, UCSC (X10 team contact: Vijay Saraswat)

- **Improving performance of the X10 runtime for multi-core**

Haibo Chen, Fudan University, China

This project aims at characterizing the performance and scalability of the current Java runtime library for X10. Specifically, we will focus on the working scheduling and data distribution for X10 activities, to exploit the parallelism and locality on NUMA-based multicore architecture. We will also investigate the use of software transaction memory to increase the parallelism of X10 programs.

- **A First Programming Course using X10**

Robert (Corky) Cartwright/Vivek Sarkar, Rice University (X10 team contact: Michael Hind)

Dr. Cartwright and Dr. Sarkar of Rice University will create curricular material for a "first programming course" using X10, based on their experience co-teaching COMP 211 (Principles of Program Design, <https://wiki.rice.edu/confluence/display/cswiki/211>) at Rice in Spring 2010. COMP 211 was the first course on software design and programming methodology offered to freshmen in their second semester. These concepts were taught using a combination of Scheme and Java. The idea behind this proposal is to create a modified version of the COMP 211 lecture material by replacing Scheme and Java by X10, and introducing implicit and explicit parallelism wherever there is a natural fit. COMP 211 is taught with the assumption that an IDE is available to the students (specifically, DrScheme for Scheme and DrJava for Java), so it will also be natural to incorporate the use of X10DT in this material.

- **Teaching Real-World Parallel and Distributed Programming**

Steven Reiss, Brown University (X10 team contact: Bard Bloom)

Dr. Reiss will teach the course Programming Parallel and Distributed Systems course using X10 as a unifying language. The course covers a range of topics, including lightweight threads on multi-core machines, large-scale distribution on clouds, and high-performance parallel computing on supercomputers. X10 will serve as a common language for the course, as it is designed to work well in all these domains. Students will get experience programming multi-core, distributed systems, and supercomputer platforms and will look at real applications in each of these domains.

- **Migrating Undergraduate Parallel Computing Courses to X10**

Claudia Fohry, University of Kassel, Germany (X10 team contact: Evelyn Duesterwald)

The computer science curriculum at Kassel University includes courses "Parallel Computing 1 to 3", which are elective courses of 3 ECTS each. Parallel Computing 1 and 2 is typically attended by second- or third-year students, Parallel Computing 3 is for master students. Traditionally, the courses teach programming in OpenMP, MPI, and a third system (e.g. Cuda, Java Threads), as well as other parallel computing topics such as basic terms, architectures, algorithms, and performance optimization. Each course includes programming assignments to be solved in teams of two students, which are the basis for grading. Project goal is to redesign the courses such that they use X10 instead of OpenMP and MPI. After this change, more time should be available to spend on principles of parallel programming, performance factors, parallel algorithms, and advanced example programs. English versions of slides and programming assignments will be published.

- **X10 Tutorial and Workshop Development**

Steven Gordon/Dave Hudak, Ohio Supercomputing Center (X10 team contact: Bard Bloom)

Dr. Gordon and Dr. Hudak of the Ohio Supercomputer Center (OSC) will develop an online tutorial and

conduct a two-day workshop entitled "Introduction to X10". The tutorial is targeted at novice-to-intermediate parallel programmers who are unfamiliar with the new concepts contained in X10 and who may, in fact, be hindered by "thinking MPI" or "thinking OpenMP" when approaching X10. The tutorial is designed to progressively increase the complexity of the parallel concepts. The first module prepares students to write efficient scientific code in an object-oriented environment (like Java's) using X10's points, regions and arrays. The second module provides an introduction to concurrency by describing activities and selected language constructs (notably, async, atomic and finish). The third module introduces the data layout problem by describing multiple places through array distributions. Finally, the students will put activities and places together (using when and at) for structured multithreading applications in the fourth module. The students will use Eclipse PTP and X10DT and all workshop exercises will be conducted using OSC's Glenn cluster. The tutorial and workshop will be publicized through the OSC Ralph Regula School of Computational Science <http://www.rscs.org/> as well as HPC University <http://www.hpcuniv.org/> website. The tutorial materials will also be submitted to the HPC University digital library for wider dissemination.

- **Teaching Parallel Programming as a Pattern Language on the Example of X10**

Christoph von Praun, Georg-Simon-Ohm University, Germany (X10 team contact: Vijay Saraswat)

- **Astronomical research and parallel computing course**

Ce Yu, Tianjin University, China

This project consists of two topics: 1) Research on migrating astronomical computing algorithms to X10 on a super computer, and 2) Parallel Computing course development to introduce the X10 programming language. The research on astronomical computing algorithms is based on our long term cooperation with astronomers. We choose two algorithms from our ongoing work which have been parallelized using pthread and MPI. Course development for X10 will be included in our "Parallel Computing" course. We will design an individual unit for X10, including language concepts and related exercises. Two course versions will be released for graduate and undergraduate students respectively.

- **Parallel computing and programming in X10 (graduate)**

Shaohua Liu, Beijing University of Posts and Telecommunications, China

Having already taught graduate student course "Methods and applications of parallel computing" for some years, Dr. Shaohua Liu will incorporate X10 into his course in next semester. This course consists of curricular development activities in the area of parallel programming based on the X10 programming language. In this course, graduate students will learn new programming skills that will be useful for programming multi-core or cluster computers. The course will be about methods and applications of parallel computing and parallel programming languages. Dr. Shaohua Liu will give three fourths of the lectures, and the participants will give the remainder. This course could help the students start their new research work in this emerging area, and then cause a "snowball effect" in their future research team, work group and networking communities.

- **Courseware for an advanced undergraduates CS course**

Xiaoge Wang, Tsinghua University, China

This project is to develop courseware for a short course on parallel programming with X10. The courseware could be used for a stand alone short course on parallel programming with X10, or be integrated with other materials, such as MPI and OpenMP, to comprise the courseware for a typical one-semester parallel programming course. It would contain lecture notes, program examples, hands-on-exercises, problem sets with sample solutions and a set of term-project topics including the reference report of the term-project.