REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
06/30/2020	Final Technical	07/01/2016 to 06/30/2019

4. TITLE AND SUBTITLE

PopcornXT: System Software for Seamless Thread Migration on Commodity Heterogeneous Multiprocessors

5a. CONTRACT NUMBER

5b. GRANT NUMBER

N000141612711

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Ravindran, Binoy

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Virginia Polytechnic Institute and State University
Office of Sponsored Programs
North End Center, 300 Turner Street NW, Suite 4200
Blacksburg, VA 24061-0001

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Office of Naval Research
ONR REG Office Atlanta
100 Alabama Street, SW Suite 4R15
Atlanta, GA 30303

10. SPONSOR/MONITOR'S ACRONYM(S)

ONR

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

DISTRIBUTION A. Approved for public release: distribution unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The project has developed the second generation of the Popcorn Linux infrastructure -- operating system (OS), compiler, and run-time system -- that enables seamless execution migration on heterogeneous instruction-set-architecture (ISA) multiprocessors, enabling their programming as shared memory SMP multicores. The project has developed a suite of innovative OS, compiler, and run-time techniques including a rack-scale memory coherency protocol, a memory page prefetching mechanism, a high-performance inter-kernel messaging layer that exploits high-latency/bandwidth network infrastructure, a compiler infrastructure that generates heterogeneous-ISA binaries, a run-time system that transforms dynamic program state to overcome application binary interface incompatibilities, and run-time workload distribution policies for effective utilization of heterogeneous-ISA multiprocessors. Popcorn Linux infrastructure is available as open-source software at http://popcornlinux.org/.

15. SUBJECT TERMS

Heterogeneous instruction-set-architectures, multiprocessors, operating system, compiler, run-time system

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Binoy Ravindran
U	U	U	UU	1	19b. TELEPHONE NUMBER *(Include area code)* 540-231-3777

Contract Number: N000141612711

Title: PopcornXT: System Software for Seamless Thread Migration on Commodity Heterogeneous Multiprocessors

Major Goals:

The Popcorn project attempts to maximize the benefits of ISA heterogeneity in a rack-scale setting. There have been discussions in the literature that demonstrate the benefits of heterogeneous ISA systems in terms of performance, power, energy, and so forth. However, to take advantage of heterogeneous ISA systems, developers should understand the characteristics of the system and their applications, and manually rewrite the existing implementation written for cache-coherent shared memory model according to their understanding. Such an approach demands a huge amount of effort, offsets the benefits of heterogeneous systems, and sometimes is infeasible.

Popcorn Linux allows any application written for the usual shared-memory architecture to run on heterogeneous ISA systems seamlessly without any additional effort from the developer. This is achieved by innovations at the system software level, including an operating system (OS) that runs on multiple-ISA processors and a compiler and run-time framework that builds multiple-ISA binaries. In this sense, the goals of the project are twofold; one is to build an operating system that provides applications with various functionalities to exploit the heterogeneity in the system. The other goal is to build a compiler and run-time framework that supports the operating system and developers. This allows developers to leverage today's applications with tomorrow's hardware while minimizing any necessary porting and refactoring of existing application code.

The Popcorn operating system is a replicated-kernel OS based on the Linux kernel. To put it another way, it is a distributed OS for tightly-coupled processors. In a heterogeneous-ISA platform, a different kernel instance is loaded onto each different-ISA processor. These kernel instances communicate with each other by message-passing and provide applications with a single system image over the heterogeneous-ISA platform. Applications written for the shared memory paradigm can run transparently (i.e., without any modification) on the single system image. We are targeting the multi-threaded/multi-program environment where a number of applications comprised of multiple threads run concurrently and simultaneously. To maximize the advantages of the heterogeneous-ISA platform, Popcorn OS should be able to migrate the execution at the thread granularity. This also implies that the Popcorn OS should provide a unified view of memory for the threads running on multiple nodes concurrently. To this end, the Popcorn OS requires a memory consistency protocol between the kernel instances. This is a sort of distributed shared memory (DSM) system that guarantees a memory consistency property such as Total Store Order, similar to traditional SMP (symmetric multi-chip processor) systems. Implementing such a DSM system at the kernel-level is expected to be challenging due to the highly concurrent and performance-critical nature of the memory subsystem. Nevertheless, based on the functionality, an application can be distributed per thread over multiple nodes based on different ISAs, thereby not only maximizing the benefits of heterogeneous-ISA systems, but also promising great flexibilities that can be exploited in many ways. Some use cases for the Popcorn OS include security extensions, performance optimization, flexible job scheduling, computation offloading, live update, and so forth.

The Popcorn compiler/run-time framework is a collection of tools (based on GCC and LLVM) and run-time libraries that masks the gap between the traditional SMP environment and the heterogeneous-ISA environment. The tools operate in multiple steps to create a unified single executable that can be migrated at run-time between different-ISA processors. Firstly, the tools create multiple versions of the source code (one per ISA) by automatically transforming the source code. If necessary, they may add the hooks to deal with the architectural differences between heterogeneous processors. Such hooks include the stack transformation from one ISA to another ISA, register mapping, performance monitoring to capture the ISA affinity, and so forth. Next, the tools compile each version of the source code for its respective ISA with the maximum compiler optimization. During the code compilation, the compiler framework aligns the addresses for symbols so that all symbols in the compiled binary are at the same virtual address irrespective of the ISA. This alignment is performed by merging the requirements of each ISA's application binary interface (ABI). During the applications' run-time, a scheduler module monitors the performance of applications and makes scheduling decisions. The decision is handed to the Popcorn operating system, and the OS performs the thread migration according to the decision. The project will show that migration is achievable using a number of runtime application-state transformation techniques.

The project is aiming to be deployed on a rack-scale setting, which is comprised of tens of nodes based on various ISAs. Thus, the project considers various network technologies (e.g., PCIe, Ethernet, and InfiniBand) and node topologies (loosely coupled configuration versus tightly coupled configuration) to understand the optimal system configuration for the nodes in the heterogeneous rack and the concomitant tradeoffs.

Accomplishments Under Goals:

In the past, we have implemented Popcorn Linux OS to allow threads in a process to be distributed over multiple nodes. Since threads share the same virtual address space, all threads must have the same view of memory even though they are distributed across remote nodes. During this reporting period, we implemented a page-level memory consistency protocol in the virtual memory system in Popcorn OS, which allows distributed threads to access shared memory with conventional load/store instructions with sequential memory consistency – a well understood programming abstraction with high programmability.

To provide up-to-date data to distributed threads, Popcorn OS tracks the location of up-to-date pages by maintaining ownership of pages. Overall, it is similar to an invalidation-based cache coherence protocol in modern processors and multi-reader single-writer memory models in the context of distributed shared memory (DSM). The ownership of each page is maintained at the origin node where the process is created. A thread can only access the pages at the node where the thread is running on has ownership. To grant the ownership and to receive the page data, nodes consult with the origin node, and the origin node controls the ownership of pages. A read to a page replicates the page to the node, granting the ownership of the page to the node. A write to a page claims the ownership of the page. This may accompany ownership revocation from other nodes.

One of the major accomplishments during this reporting period has been to improve the scalability of the memory consistency protocol to fully exploit the heterogeneous-ISA rack.
In the page-level memory consistency protocol, there is an inherent false page sharing issue by which applications may not scale unmodified. When two threads running on different nodes access different data objects in the same page for write, both of the threads need an exclusive access to the page. This makes the page bounce between the nodes, significantly interfering with the application's execution progress.

In this reporting period, we built a software page access tracking tool to effectively remove false page sharing. Firstly, Popcorn OS generates page fault traces while running target applications. The trace is post-processed with an user-space analysis tool, which pinpoints the source code location which caused the most page faults, page fault frequency over time, per-thread memory access patterns, etc. This allows us to identify sources of cross-node-traffic and to apply optimizations to improve scalability. Our evaluation using nine applications with real workloads confirmed the effectiveness of the optimization: optimization takes a few lines of source code modifications, yielding scalability on an 8-node rack.

Another major accomplishment of this reporting period is our optimization of Popcorn OS’s inter-node communication layer to effectively utilize RDMA in InfiniBand. The communication layer is performance-critical in distributed systems, and it is indeed so for Popcorn OS. To this end, we designed and implemented an inter-node communication system over InfiniBand so that Popcorn OS can leverage the high-bandwidth and low-latency of modern interconnect technologies. However, I/O buffers for send and receive over InfiniBand must be explicitly mapped to a DMA-capable address space range so that an InfiniBand host controller adaptor can perform DMA from/to the buffers in memory. To perform remote DMA (RDMA), the buffer must also be associated with an RDMA memory region with a remote key, with which a remote node can perform RDMA from/to the buffer. This DMA-mapping/unmapping and RDMA memory region association/disassociation is so costly that it offsets the benefit of RDMA.

Based on these observations, we devised a hybrid approach using both RDMA and memory copying. Each connection has an RDMA sink, which is set up during connection initialization. The RDMA sink is comprised of chunks of physically contiguous pages, and the chunks are associated with an RDMA memory region during the connection setup. To perform RDMA, a buffer from the RDMA sink is allocated and asks its peer to perform RDMA to the location. When the peer notifies an RDMA completion, the data in the RDMA sink is copied to its final destination (i.e., a page in the application virtual memory) and released. Even though this approach involves one memory copy, it is faster than performing RDMA association for each page in the memory consistency protocol.

The last major accomplishment of this reporting period is the development of a page prefetching mechanism to minimize the overhead of remote memory access. Many data center applications (for example, page rank, belief propagation, k-means clustering, so forth) are comprised of multiple phases, and the phases are repeated until the results converge. This repeated work results in regular memory access patterns from the applications, and we utilize the regular memory access pattern to prefetch pages that will be accessed in the

near future.

The Popcorn compiler analyzes loop variables and their ranges and generates prefetch requests to Popcorn OS based on the analysis. Then, Popcorn OS generates prefetch requests to other nodes. The requests are done in the background and carefully handled so that the foreground execution of applications is minimally disturbed. Our preliminary evaluation reveals that page prefetching can remove approximately 1/3rd of the page faults and improve application performance by 20%.

Training Opportunities:

The project directly supported (partially or fully) postdoctoral scholars, research faculty members, PhD students, MS students, and undergraduate students.
The postdocs and research faculty members were mentored on the following:
- understanding how to formulate research problems
- understanding how to develop initial seed ideas to full-fledged solutions, considering various tradeoffs
- understanding how to evaluate solutions through analysis
- understanding how to evaluate solutions through experimental studies
- understanding how to plan the research, develop schedules and milestones, and identify potential deliverables.

The PhD, MS, and BS students were mentored on a number of activities, mostly focused on developing their system building and research skills. Most of these training activities took the form of one-on-one work with the PI or the postdocs/research faculty members. The main topics of training include system programming, Linux kernel, Popcorn Linux internals (specifically the messaging layer), and distributed scheduling design. Such training substantially improved students understanding of operating system topics – this material is not covered in normal educational curricula.

In order to foster students' professional development, students working on the project attended at least one conference during this performance period and were invited to present the research results via poster or demo sessions. Attending conferences and presenting project results significantly impacts the professional development of the students in a positive manner. The interaction with peers and senior researchers allows for professional growth and valuable networking: interacting with peers stimulates new ideas and leads the student to critique and improve their own work, while senior researchers offer exemplary talks and revolutionary ideas that can influence students' interests and future research directions. Presenting project results also positively impacts students' professional development. Students must possess the ability to present, discuss, and elaborate upon the results of their research in a generalized, but concise terms with peers of different backgrounds. This in turn leads to cross-pollination of ideas between fields, and it can enhance students' productivity and interactions with large diverse teams in their later professional careers.

Further professional development for the personnel working on the project occurred in the form of working on emerging heterogeneous-ISA hardware. To the best of our knowledge, the PI's group is the first to build a heterogeneous-ISA CPU/GPU platform; there is no heterogeneous-ISA platform with such tightly-coupled processors and similar characteristics available today. Moreover, the system software developed for this platform is unique, from the operating system to the compiler framework. We believe that exposure to such technology cannot be achieved anywhere else and represents an important professional development opportunity.

Results Dissemination

The results from the project have been thoroughly disseminated to the relevant researcher and practitioner communities in the following ways:

1) in non peer-reviewed technical reports, which are made publicly available at the project website (http://www.popcornlinux.org/);
2) in peer-reviewed, relevant conference and journal publications of ACM, USENIX, and IEEE;
3) directly described at the project website (http://www.popcornlinux.org/). The documentation describes in detail the experimental settings, how the experiments were conducted, and how the data was measured (to enable replication of the experiments by other researchers and practitioners); and
4) presentations at non-conference events.
The Popcorn Linux infrastructure is open-source and freely available at: http://www.popcornlinux.org/.

During this reporting period, the project results were published and presented at the following international conferences and journals:

Conference publications:
- “OS Support for Thread Migration and Distribution in the Fully Heterogeneous Datacenter,” Pierre Olivier, Sang-Hoon Kim, Binoy Ravindran, The 16th Workshop on Hot Topics in Operating Systems (HotOS XVI), May 7-10 2017, Whistler, British Columbia, Canada
- “libMPRack: An OpenMP Runtime for Effortless Cluster Programming,” Rob Lyerly, Sang-Hoon Kim, and Binoy Ravindran, The 2018 ACM Asia-Pacific Workshop on Systems (APSys), August 27-28, 2018, Jeju Island, South Korea (Under Review)

Journal publications:
- “AIRA: A Framework for Flexible Compute Kernel Execution in Heterogeneous Platforms," Rob Lyerly, Alastair Murray, Antonio Barbalace, and Binoy Ravindran, IEEE Transactions on Parallel and Distributed Systems, Volume 29, Number 2, pages 269 - 282, February 2018

Poster publications:
- "Secure Popcorn: Using Machine Boundaries to Harden Applications," Rob Lyerly, Sergey Bratus, and Binoy Ravindran, The 2017 ARM Research Summit 2017, Cambridge, UK

During this reporting period, the project results were presented at the following non-conference events:
-Binoy Ravindran and Rob Lyerly, “Popcorn Linux: Systems Software for Heterogeneous Hardware,” DoD Cyber CoI Executive Council Meeting and Naval S&T Deep-Dive, SPAWAR Systems Center Pacific, Point Loma, San Diego, CA, USA, February 22, 2018
-Binoy Ravindran, Rob Lyerly, and Sang-Hoon Kim, “The Popcorn Linux Project,” US Naval Surface Warfare Center, Dahlgren, VA, USA, September 8, 2017
-Sang-Hoon Kim, "Popcorn Linux: System Software for Heterogeneous Hardware," 6th Workshop on Runtime and Operating Systems for the Many-core Era (ROME 2018), co-located with IPDPS 2018, May 21 - 25, 2018, Vancouver, British Columbia, Canada
-Pierre Olivier and Sang-Hoon Kim, "Popcorn Linux: Compiler & OS Support for Execution Migration in Heterogeneous ISA Environments," 2017 Linux Plumbers Conference, September 11, 2017, Los Angeles, CA

Plans Next Reporting Period

Our plans include several research directions. First, we are aiming to further improve the scalability of DSM in Popcorn OS. We have discovered a few new scalability challenges in DSM while improving the memory consistency protocol in Popcorn OS. One of the major challenges is to balance network traffic in the rack. In the current implementation, remote nodes request the origin the page data and ownership, and only the origin gives the data to remotes. When the origin does not own a requested page, the origin first brings the page from a remote that owns the page, and then forwards the page to the requesting node. This involves two page data movement over the network (one for inbound and one for outbound from the origin). When a process is distributed over many nodes, this type of faults can stress the network link at the origin significantly, bottlenecking the page fault handling. We plan to improve the current implementation by having the remote nodes exchange pages by themselves without going through the origin. When the origin does not own a requested page, the origin forwards the request to the remote node that owns the page. The remote node sends the page data to the requesting remote node, and finally the requesting node notifies the origin of the receiving. This way, Popcorn OS can reduce the traffic for page data and balance network link usage among

the origin and remotes.

Our second goal is to deal with node failures. As more machines are involved in Popcorn at the rack-scale, it becomes more likely that one of the nodes in the system might fail. However, to the best of our knowledge, there is no practical, widely-deployed fault tolerance mechanism in DSM. This is mainly due to the performance-critical characteristics of DSM; it is not obvious to keep a consistent memory state that can survive across machine failures while providing high performance in DSM.

Our general solution direction to tackle such failures is to perform checkpointing incrementally so that applications can transparently recover their last status and resume execution upon hardware faults. Checkpointing implies preserving application data in memory, values in registers, and the status of kernel objects such as file descriptors. As registers and kernel objects can be easily collected and recovered due to their small volume, we will focus on keeping the application data in the memory safe.

Our initial design leverages the page access control capability in DSM. In order to provide memory consistency, Popcorn's DSM already has a number of hooks in the memory subsystem from where the DSM system can identify page's status and control page accesses according to the current status. We will extend these mechanisms and capabilities to realize the checkpointing feature across multiple nodes.

Another direction that we propose to explore is to improve the compiler toolchain for optimizing applications. The current compiler toolchain only helps developers to identify the source location of cross-node page interference. After the identification, however, developers must figure out the main cause of the interference and manually optimize the implementation by hand. Our preliminary analysis on applications indicates that there are a number of obvious cases that incurs a severe false page sharing but can be easily removed. For example, we can avoid false page sharing by separating out read-only variables and coalescing them together. Also, we can remove many false page sharing by coalescing per-thread variables in a separate, page-aligned address range. From these observations, we aim to optimize the compiler toolchain by automatically allocating program objects from a per-node memory pool and/or adapting the allocation frequency and size of program objects.

Honors and Awards

The project's ASPLOS 2017 paper "Breaking the Boundaries in Heterogeneous-ISA Datacenters," was selected as a Highlight Paper at the 10th ACM International Systems & Storage Conference (SYSTOR 2017), May 22-24, 2017, Haifa, Israel.

The abstract submission of Mr. Rob Lyerly, a PhD student working on the Popcorn Linux project, titled "Seamless POSIX/OpenMP Thread Migration Across ISA Boundaries," won the 2017 ACM Student Research Competition (SRC), co-located with the 2017 Programming Language Design and Implementation (PLDI 2017) conference.

Protocol Activity Status

Technology Transfer

In addition to the technology transfer efforts described for the previous performance year, which are continuing, additional transfer efforts for this performance year include:

- Collaboration with IBM Haifa, Israel. A technology transfer and collaborative research relationship has been established between VT and IBM Haifa, Israel. The IBM Haifa group has been experimenting with Popcorn Linux as a potential infrastructure solution for IBM's datacenter systems. The PI met with IBM Haifa researchers in May 2017 to discuss this partnership. To facilitate this partnership, a formal relationship between VT and IBM Haifa on collaborative research and intellectual property ownership has been established.
- Transfer to US Naval Surface Warfare Center Dahlgren Division (NSWCDD). Several groups at NSWCDD expressed significant interest for downloading and experimenting with Popcorn Linux after the seminar that the PI's group gave at NSWCDD: Binoy Ravindran, Rob Lyerly, and Sang-Hoon Kim, "The Popcorn Linux Project," US Naval Surface Warfare Center, Dahlgren, VA, USA, September 8, 2017. Two of those groups

are currently actively experimenting with Popcorn.

Distribution Statement:

Approved for public release; distribution is unlimited.

Participants

First Name: Binoy **Last Name:** Ravindran

Project Role: PD/PI

National Academy Member: N **Months Worked:** 1

Countries of Collaboration

First Name: Pierre **Last Name:** Olivier

Project Role: Postdoctoral (scholar, fellow or other postdoctoral position)

National Academy Member: N **Months Worked:** 3

Countries of Collaboration

First Name: Sang-Hoon **Last Name:** Kim

Project Role: Postdoctoral (scholar, fellow or other postdoctoral position)

National Academy Member: N **Months Worked:** 7

Countries of Collaboration

First Name: Mohamed **Last Name:** Karaoui

Project Role: Postdoctoral (scholar, fellow or other postdoctoral position)

National Academy Member: N **Months Worked:** 7

Countries of Collaboration

Office of Naval Research (ONR)

Research Performance Progress Report (RPPR)

Contract Number: N000141612711

This document has been developed to provide Principal Investigators (PIs), co-PIs, and research organizations a template to collect information before entering the required information in the online reports. PIs should NOT complete and upload this template document to https://extranet.aro.army.mil in order to meet your reporting requirement. You are required to enter text in the text boxes available online.

ONR's RPPR Instructions provides more detailed instructions and contextual assistance.

***Note**: ONR interim project reports are not cumulative and should always be prepared for the specific project reporting period only.*

Distribution Statement

Select between:

- **DISTRIBUTION A. Approved for public release: distribution unlimited**.
- DISTRIBUTION B. Distribution authorized to U.S. Government Agencies (reason) (date of determination). Other request for this document shall be referred to (controlling DoD office).
- DISTRIBUTION C. Distribution authorized to U.S. Government Agencies and their contractors. (reason) (date of determination). Other request for this document shall be referred to (controlling DoD office).
- DISTRIBUTION D. Distribution authorized to Department of Defense and U.S. DoD contractors only (reason) (date of determination). Other request for this document shall be referred to (controlling DoD office).
- DISTRIBUTION E. Distribution authorized to DoD components only (reason) (date of determination). Other request for this document shall be referred to (controlling DoD office).
- DISTRIBUTION F. Further dissemination only as directed by (controlling office) (date of determination) or higher DoD authority.

Accomplishments

What were the major goals and objectives of the project?

The Popcorn project attempts to maximize the benefits of ISA heterogeneity in a rack-scale setting. There have been discussions in the literature that demonstrate the benefits of heterogeneous ISA systems in terms of performance, power, energy, and so forth. However, to take advantage of heterogeneous ISA systems, developers should understand the characteristics of the system and their applications, and manually rewrite the existing implementation written for cache-coherent shared memory model according to their understanding. Such an approach demands a huge amount of effort, offsets the benefits of heterogeneous systems, and sometimes is infeasible.

Popcorn Linux allows any application written for the usual shared-memory architecture to run on heterogeneous ISA systems seamlessly without any additional effort from the developer. This is achieved by innovations at the system software level, including an operating system (OS) that runs on multiple-ISA processors and a compiler and run-time framework that builds multiple-ISA binaries. In this sense, the goals of the project are twofold; one is to build an operating system that provides applications with various functionalities to exploit the heterogeneity in the system. The other goal is to build a compiler and run-time framework that supports the operating system and developers. This allows developers to leverage today's applications with tomorrow's hardware while minimizing any necessary porting and refactoring of existing application code.

The Popcorn operating system is a replicated-kernel OS based on the Linux kernel. To put it another way, it is a distributed OS for tightly-coupled processors. In a heterogeneous-ISA

platform, a different kernel instance is loaded onto each different-ISA processor. These kernel instances communicate with each other by message-passing and provide applications with a single system image over the heterogeneous-ISA platform. Applications written for the shared memory paradigm can run transparently (i.e., without any modification) on the single system image. We are targeting the multi-threaded/multi-program environment where a number of applications comprised of multiple threads run concurrently and simultaneously. To maximize the advantages of the heterogeneous-ISA platform, Popcorn OS should be able to migrate the execution at the thread granularity. This also implies that the Popcorn OS should provide a unified view of memory for the threads running on multiple nodes concurrently. To this end, the Popcorn OS requires a memory consistency protocol between the kernel instances. This is a sort of distributed shared memory (DSM) system that guarantees a memory consistency property such as Total Store Order, similar to traditional SMP (symmetric multi-chip processor) systems. Implementing such a DSM system at the kernel-level is expected to be challenging due to the highly concurrent and performance-critical nature of the memory subsystem. Nevertheless, based on the functionality, an application can be distributed per thread over multiple nodes based on different ISAs, thereby not only maximizing the benefits of heterogeneous-ISA systems, but also promising great flexibilities that can be exploited in many ways. Some use cases for the Popcorn OS include security extensions, performance optimization, flexible job scheduling, computation offloading, live update, and so forth.

The Popcorn compiler/run-time framework is a collection of tools (based on GCC and LLVM) and run-time libraries that masks the gap between the traditional SMP environment and the heterogeneous-ISA environment. The tools operate in multiple steps to create a unified single executable that can be migrated at run-time between different-ISA processors. Firstly, the tools create multiple versions of the source code (one per ISA) by automatically transforming the source code. If necessary, they may add the hooks to deal with the architectural differences between heterogeneous processors. Such hooks include the stack transformation from one ISA to another ISA, register mapping, performance monitoring to capture the ISA affinity, and so forth. Next, the tools compile each version of the source code for its respective ISA with the maximum compiler optimization. During the code compilation, the compiler framework aligns the addresses for symbols so that all symbols in the compiled binary are at the same virtual address irrespective of the ISA. This alignment is performed by merging the requirements of each ISA's application binary interface (ABI). During the applications' run-time, a scheduler module monitors the performance of applications and makes scheduling decisions. The decision is handed to the Popcorn operating system, and the OS performs the thread migration according to the decision. The project will show that migration is achievable using a number of runtime application-state transformation techniques.

The project is aiming to be deployed on a rack-scale setting, which is comprised of tens of nodes based on various ISAs. Thus, the project considers various network technologies (e.g., PCIe, Ethernet, and InfiniBand) and node topologies (loosely coupled configuration versus tightly coupled configuration) to understand the optimal system configuration for the nodes in the heterogeneous rack and the concomitant tradeoffs.

What was accomplished towards achieving these goals?

In the past, we have implemented Popcorn Linux OS to allow threads in a process to be distributed over multiple nodes. Since threads share the same virtual address space, all threads must have the same view of memory even though they are distributed across remote nodes. During this reporting period, we implemented a page-level memory consistency protocol in the virtual memory system in Popcorn OS, which allows distributed threads to access shared memory with conventional load/store instructions with sequential memory consistency – a well understood programming abstraction with high programmability.

To provide up-to-date data to distributed threads, Popcorn OS tracks the location of up-to-date pages by maintaining ownership of pages. Overall, it is similar to an invalidation-based cache coherence protocol in modern processors and multi-reader single-writer memory models in the context of distributed shared memory (DSM). The ownership of each page is maintained at the origin node where the process is created. A thread can only access the pages at the node where the thread is running on has ownership. To grant the ownership and to receive the page data, nodes consult with the origin node, and the origin node controls the ownership of pages. A read to a page replicates the page to the node, granting the ownership of the page to the node. A write to a page claims the ownership of the page. This may accompany ownership revocation from other nodes.

One of the major accomplishments during this reporting period has been to improve the scalability of the memory consistency protocol to fully exploit the heterogeneous-ISA rack. In the page-level memory consistency protocol, there is an inherent false page sharing issue by which applications may not scale unmodified. When two threads running on different nodes access different data objects in the same page for write, both of the threads need an exclusive access to the page. This makes the page bounce between the nodes, significantly interfering with the application's execution progress.

In this reporting period, we built a software page access tracking tool to effectively remove false page sharing. Firstly, Popcorn OS generates page fault traces while running target applications. The trace is post-processed with an user-space analysis tool, which pinpoints the source code location which caused the most page faults, page fault frequency over time, per-thread memory access patterns, etc. This allows us to identify sources of cross-node-traffic and to apply optimizations to improve scalability. Our evaluation using nine applications with real workloads confirmed the effectiveness of the optimization: optimization takes a few lines of source code modifications, yielding scalability on an 8-node rack.

Another major accomplishment of this reporting period is our optimization of Popcorn OS's inter-node communication layer to effectively utilize RDMA in InfiniBand. The communication layer is performance-critical in distributed systems, and it is indeed so for Popcorn OS. To this end, we designed and implemented an inter-node communication system over InfiniBand so that Popcorn OS can leverage the high-bandwidth and low-latency of modern interconnect technologies. However, I/O buffers for send and receive over InfiniBand must be explicitly

mapped to a DMA-capable address space range so that an InfiniBand host controller adaptor can perform DMA from/to the buffers in memory. To perform remote DMA (RDMA), the buffer must also be associated with an RDMA memory region with a remote key, with which a remote node can perform RDMA from/to the buffer. This DMA-mapping/unmapping and RDMA memory region association/disassociation is so costly that it offsets the benefit of RDMA.

Based on these observations, we devised a hybrid approach using both RDMA and memory copying. Each connection has an RDMA sink, which is set up during connection initialization. The RDMA sink is comprised of chunks of physically contiguous pages, and the chunks are associated with an RDMA memory region during the connection setup. To perform RDMA, a buffer from the RDMA sink is allocated and asks its peer to perform RDMA to the location. When the peer notifies an RDMA completion, the data in the RDMA sink is copied to its final destination (i.e., a page in the application virtual memory) and released. Even though this approach involves one memory copy, it is faster than performing RDMA association for each page in the memory consistency protocol.

The last major accomplishment of this reporting period is the development of a page prefetching mechanism to minimize the overhead of remote memory access. Many data center applications (for example, page rank, belief propagation, k-means clustering, so forth) are comprised of multiple phases, and the phases are repeated until the results converge. This repeated work results in regular memory access patterns from the applications, and we utilize the regular memory access pattern to prefetch pages that will be accessed in the near future.

The Popcorn compiler analyzes loop variables and their ranges and generates prefetch requests to Popcorn OS based on the analysis. Then, Popcorn OS generates prefetch requests to other nodes. The requests are done in the background and carefully handled so that the foreground execution of applications is minimally disturbed. Our preliminary evaluation reveals that page prefetching can remove approximately 1/3rd of the page faults and improve application performance by 20%.

What opportunities for training and professional development did the project provide?

The project directly supported (partially or fully) postdoctoral scholars, research faculty members, PhD students, MS students, and undergraduate students.

The postdocs and research faculty members were mentored on the following:
- understanding how to formulate research problems
- understanding how to develop initial seed ideas to full-fledged solutions, considering various tradeoffs
- understanding how to evaluate solutions through analysis
- understanding how to evaluate solutions through experimental studies
- understanding how to plan the research, develop schedules and milestones, and identify potential deliverables.

The PhD, MS, and BS students were mentored on a number of activities, mostly focused on developing their system building and research skills. Most of these training activities took the form of one-on-one work with the PI or the postdocs/research faculty members. The main topics of training include system programming, Linux kernel, Popcorn Linux internals (specifically the messaging layer), and distributed scheduling design. Such training substantially improved students understanding of operating system topics – this material is not covered in normal educational curricula.

In order to foster students' professional development, students working on the project attended at least one conference during this performance period and were invited to present the research results via poster or demo sessions. Attending conferences and presenting project results significantly impacts the professional development of the students in a positive manner. The interaction with peers and senior researchers allows for professional growth and valuable networking: interacting with peers stimulates new ideas and leads the student to critique and improve their own work, while senior researchers offer exemplary talks and revolutionary ideas that can influence students' interests and future research directions. Presenting project results also positively impacts students' professional development. Students must possess the ability to present, discuss, and elaborate upon the results of their research in a generalized, but concise terms with peers of different backgrounds. This in turn leads to cross-pollination of ideas between fields, and it can enhance students' productivity and interactions with large diverse teams in their later professional careers.

Further professional development for the personnel working on the project occurred in the form of working on emerging heterogeneous-ISA hardware. To the best of our knowledge, the PI's group is the first to build a heterogeneous-ISA CPU/GPU platform; there is no heterogeneous-ISA platform with such tightly-coupled processors and similar characteristics available today. Moreover, the system software developed for this platform is unique, from the operating system to the compiler framework. We believe that exposure to such technology cannot be achieved anywhere else and represents an important professional development opportunity.

How were the results disseminated to communities of interest?

The results from the project have been thoroughly disseminated to the relevant researcher and practitioner communities in the following ways:

1) in non peer-reviewed technical reports, which are made publicly available at the project website (http://www.popcornlinux.org/);
2) in peer-reviewed, relevant conference and journal publications of ACM, USENIX, and IEEE;
3) directly described at the project website (http://www.popcornlinux.org/). The documentation describes in detail the experimental settings, how the experiments were conducted, and how the data was measured (to enable replication of the experiments by other researchers and practitioners); and
4) presentations at non-conference events.

The Popcorn Linux infrastructure is open-source and freely available at: http://www.popcornlinux.org/.

During this reporting period, the project results were published and presented at the following international conferences and journals:

Conference publications:
- “OS Support for Thread Migration and Distribution in the Fully Heterogeneous Datacenter,” Pierre Olivier, Sang-Hoon Kim, Binoy Ravindran, *The 16th Workshop on Hot Topics in Operating Systems (HotOS XVI)*, May 7-10 2017, Whistler, British Columbia, Canada
- “libMPRack: An OpenMP Runtime for Effortless Cluster Programming,” Rob Lyerly, Sang-Hoon Kim, and Binoy Ravindran, *The 2018 ACM Asia-Pacific Workshop on Systems (APSys)*, August 27-28, 2018, Jeju Island, South Korea (Under Review)
- “DEX: Blurring Machine Boundaries in Linux,” Sang-Hoon Kim, Rob Lyerly, Ho-Ren Chuang, Changwoo Min, and Binoy Ravindran, *13th USENIX Symposium on Operating Systems Design and Implementation (OSDI '18)*, October 8–10, 2018, Carlsbad, CA, USA (Under Review)

Journal publications:
- “AIRA: A Framework for Flexible Compute Kernel Execution in Heterogeneous Platforms," Rob Lyerly, Alastair Murray, Antonio Barbalace, and Binoy Ravindran, *IEEE Transactions on Parallel and Distributed Systems*, Volume 29, Number 2, pages 269 - 282, February 2018

Poster publications:
- "Secure Popcorn: Using Machine Boundaries to Harden Applications," Rob Lyerly, Sergey Bratus, and Binoy Ravindran, The 2017 ARM Research Summit 2017, Cambridge, UK

During this reporting period, the project results were presented at the following non-conference events:

- Binoy Ravindran and Rob Lyerly, “Popcorn Linux: Systems Software for Heterogeneous Hardware,” DoD Cyber CoI Executive Council Meeting and Naval S&T Deep-Dive, SPAWAR Systems Center Pacific, Point Loma, San Diego, CA, USA, February 22, 2018
- Binoy Ravindran, Rob Lyerly, and Sang-Hoon Kim, “The Popcorn Linux Project,” US Naval Surface Warfare Center, Dahlgren, VA, USA, September 8, 2017
- Sang-Hoon Kim, "Popcorn Linux: System Software for Heterogeneous Hardware," 6th Workshop on Runtime and Operating Systems for the Many-core Era (ROME 2018), co-located with IPDPS 2018, May 21 - 25, 2018, Vancouver, British Columbia, Canada
- Pierre Olivier and Sang-Hoon Kim, "Popcorn Linux: Compiler & OS Support for Execution Migration in Heterogeneous ISA Environments," 2017 Linux Plumbers Conference, September 11, 2017, Los Angeles, CA

What do you plan to do during the next reporting period to accomplish the goals and objectives?

Our plans include several research directions. First, we are aiming to further improve the scalability of DSM in Popcorn OS. We have discovered a few new scalability challenges in DSM while improving the memory consistency protocol in Popcorn OS. One of the major challenges is to balance network traffic in the rack. In the current implementation, remote nodes request the origin the page data and ownership, and only the origin gives the data to remotes. When the origin does not own a requested page, the origin first brings the page from a remote that owns the page, and then forwards the page to the requesting node. This involves two page data movement over the network (one for inbound and one for outbound from the origin). When a process is distributed over many nodes, this type of faults can stress the network link at the origin significantly, bottlenecking the page fault handling. We plan to improve the current implementation by having the remote nodes exchange pages by themselves without going through the origin. When the origin does not own a requested page, the origin forwards the request to the remote node that owns the page. The remote node sends the page data to the requesting remote node, and finally the requesting node notifies the origin of the receiving. This way, Popcorn OS can reduce the traffic for page data and balance network link usage among the origin and remotes.

Our second goal is to deal with node failures. As more machines are involved in Popcorn at the rack-scale, it becomes more likely that one of the nodes in the system might fail. However, to the best of our knowledge, there is no practical, widely-deployed fault tolerance mechanism in DSM. This is mainly due to the performance-critical characteristics of DSM; it is not obvious to keep a consistent memory state that can survive across machine failures while providing high performance in DSM.

Our general solution direction to tackle such failures is to perform checkpointing incrementally so that applications can transparently recover their last status and resume execution upon hardware faults. Checkpointing implies preserving application data in memory, values in registers, and the status of kernel objects such as file descriptors. As registers and kernel objects can be easily collected and recovered due to their small volume, we will focus on keeping the application data in the memory safe.

Our initial design leverages the page access control capability in DSM. In order to provide memory consistency, Popcorn's DSM already has a number of hooks in the memory subsystem from where the DSM system can identify page's status and control page accesses according to the current status. We will extend these mechanisms and capabilities to realize the checkpointing feature across multiple nodes.

Another direction that we propose to explore is to improve the compiler toolchain for optimizing applications. The current compiler toolchain only helps developers to identify the source location of cross-node page interference. After the identification, however, developers must figure out the main cause of the interference and manually optimize the implementation

by hand. Our preliminary analysis on applications indicates that there are a number of obvious cases that incurs a severe false page sharing but can be easily removed. For example, we can avoid false page sharing by separating out read-only variables and coalescing them together. Also, we can remove many false page sharing by coalescing per-thread variables in a separate, page-aligned address range. From these observations, we aim to optimize the compiler toolchain by automatically allocating program objects from a per-node memory pool and/or adapting the allocation frequency and size of program objects.

Honors: What honors or awards were received under this project in this reporting period?

The project's ASPLOS 2017 paper "Breaking the Boundaries in Heterogeneous-ISA Datacenters," was selected as a Highlight Paper at the 10th ACM International Systems & Storage Conference (SYSTOR 2017), May 22-24, 2017, Haifa, Israel.

The abstract submission of Mr. Rob Lyerly, a PhD student working on the Popcorn Linux project, titled "Seamless POSIX/OpenMP Thread Migration Across ISA Boundaries," won the 2017 ACM Student Research Competition (SRC), co-located with the 2017 Programming Language Design and Implementation (PLDI 2017) conference.

Technology Transfer

In addition to the technology transfer efforts described for the previous performance year, which are continuing, additional transfer efforts for this performance year include:

- Collaboration with IBM Haifa, Israel. A technology transfer and collaborative research relationship has been established between VT and IBM Haifa, Israel. The IBM Haifa group has been experimenting with Popcorn Linux as a potential infrastructure solution for IBM's datacenter systems. The PI met with IBM Haifa researchers in May 2017 to discuss this partnership. To facilitate this partnership, a formal relationship between VT and IBM Haifa on collaborative research and intellectual property ownership has been established.
- Transfer to US Naval Surface Warfare Center Dahlgren Division (NSWCDD). Several groups at NSWCDD expressed significant interest for downloading and experimenting with Popcorn Linux after the seminar that the PI's group gave at NSWCDD: Binoy Ravindran, Rob Lyerly, and Sang-Hoon Kim, "The Popcorn Linux Project," US Naval Surface Warfare Center, Dahlgren, VA, USA, September 8, 2017. Two of those groups are currently actively experimenting with Popcorn.

Participants

Have on hand the following information for each participant to enter into the report:

1. Faculty (PI)
2.
3. Binoy
4. Ravindran
5.
6.
7. 1 month
8. N
9. US based

1. Postdoctoral Scholar
2.
3. Mohamed
4. Karaoui
5.
6.
7. 7 months
8. N
9. US based

1. Postdoctoral Scholar
2.
3. Sang-Hoon
4. Kim
5.
6.
7. 7 months
8. N
9. US based

1. Postdoctoral Scholar
2.
3. Pierre
4. Olivier
5.
6.
7. 3 months
8. N
9. US based

Students

Number of undergraduate STEM participants: 5

Number of graduate STEM participants: 7

Number of participants that received a STEM degree: 0

Products

Below is the information detailed for each product submission:

Conference publications:
- “OS Support for Thread Migration and Distribution in the Fully Heterogeneous Datacenter,” Pierre Olivier, Sang-Hoon Kim, Binoy Ravindran, *The 16th Workshop on Hot Topics in Operating Systems (HotOS XVI)*, May 7-10 2017, Whistler, British Columbia, Canada
- “libMPRack: An OpenMP Runtime for Effortless Cluster Programming,” Rob Lyerly, Sang-Hoon Kim, and Binoy Ravindran, *The 2018 ACM Asia-Pacific Workshop on Systems (APSys)*, August 27-28, 2018, Jeju Island, South Korea (Under Review)
“DEX: Blurring Machine Boundaries in Linux,” Sang-Hoon Kim, Rob Lyerly, Ho-Ren Chuang, Changwoo Min, and Binoy Ravindran, *13th USENIX Symposium on Operating Systems Design and Implementation (OSDI '18)*, October 8–10, 2018, Carlsbad, CA, USA (Under Review)

Journal publications:
- “AIRA: A Framework for Flexible Compute Kernel Execution in Heterogeneous Platforms," Rob Lyerly, Alastair Murray, Antonio Barbalace, and Binoy Ravindran, *IEEE Transactions on Parallel and Distributed Systems*, Volume 29, Number 2, pages 269 - 282, February 2018

Poster publications:
- "Secure Popcorn: Using Machine Boundaries to Harden Applications," Rob Lyerly, Sergey Bratus, and Binoy Ravindran, The 2017 ARM Research Summit 2017, Cambridge, UK

Presentations:
- Binoy Ravindran and Rob Lyerly, “Popcorn Linux: Systems Software for Heterogeneous Hardware,” DoD Cyber CoI Executive Council Meeting and Naval S&T Deep-Dive, SPAWAR Systems Center Pacific, Point Loma, San Diego, CA, USA, February 22, 2018
- Binoy Ravindran, Rob Lyerly, and Sang-Hoon Kim, “The Popcorn Linux Project,” US Naval Surface Warfare Center, Dahlgren, VA, USA, September 8, 2017

- Sang-Hoon Kim, "Popcorn Linux: System Software for Heterogeneous Hardware," 6th Workshop on Runtime and Operating Systems for the Many-core Era (ROME 2018), co-located with IPDPS 2018, May 21 - 25, 2018, Vancouver, British Columbia, Canada
- Pierre Olivier and Sang-Hoon Kim, "Popcorn Linux: Compiler & OS Support for Execution Migration in Heterogeneous ISA Environments," 2017 Linux Plumbers Conference, September 11, 2017, Los Angeles, CA

Website:

a. Title: Popcorn Linux
b. URL: http://www.popcornlinux.org
c. Description (8000) Characters:
    i. The website aggregates all the key information about the project and its results. Moreover, through the website, the software developed by the project is made open-source and can be freely downloaded. Installation guides, development guides, and use guides are also available on the website in order to facilitate reproduction of the scientific results achieved by the project and disseminated within the scientific literature. All scientific papers, posters, slides, tutorials, and video produced are indexed by the website – most of them can be directly downloaded.
    ii. The website is designed around a discrete amount of pages that respectively: 1) give an overview and high-level vision of the project – using a distributed operating system in a single platform (the multi-kernel OS design) to seamlessly exploit heterogeneous cores; 2) give an overview of the heterogeneous-ISA execution problem and how to compile an application in order to run it on such platforms – not just for the classical CPU/GPU configuration but also for emerging CPU/CPU configurations; 3) how we applied the same concept in a Java virtual machine; 4) a series of documents (from conference papers to video tutorials); 5) a list of software products with respective access link; and 6) a list of current and past contributors. The main page succinctly summarizes the project goals and provides the list of latest news.
    iii. Finally, the website hosts multiple software archives. These are specific versions of the software that have been used for specific publications or when we reached a milestone. The website has multiple links to a sister-website hosted on sourceforge.net that hosts our version-controlled software. We used an external service in order to reduce the maintenance cost of our website. The sourceforge.net website is regularly updated with our periodic releases.