Systems & Software

March 2014

Dr. Kathleen M. Kaplan, MIT Fellow
*Principal Computer Scientist*
Air Force Office of Scientific Research

Integrity ★ Service ★ Excellence
• AF mission: “fly, fight, and win in air, space, and cyberspace”

• Thus, the AF must invest in Systems & Software, which is the keystone of all advanced technology

• Discussion:
  – Basic Research
  – Future portfolio areas
Basic Research wrt Systems & Software

• The delineation of basic research (6.1) versus other research (6.2, etc.) is as follows:
  
  — 6.1: "Supports research that produces new knowledge in a scientific or technological area of interest to the military. Areas include ... computer sciences”
  
  — 6.2: "Supports the exploratory development and initial maturation of new technologies for specific military application (or further developing existing technology for new military applications)"

Quotes from RDT&E Congressional Report

• None of the research directions presented involve specific military applications – all areas are studies directed toward greater knowledge
Systems & Software: 2 Directions

1. Legacy System Research
   i. The AF’s use of legacy systems is well known, along with the detrimental issues of legacy system use
   ii. Many AF systems (such as combat system software) have extremely long life cycles
   iii. In order to ensure that legacy systems are up-to-date, new system infrastructures must be investigated

2. New Technology Research
   i. AF must continue to be on the cutting-edge of technology
   ii. Studying operating systems, compilers, virtual memory, Multi/Many-core platforms, and the like, will drastically improve current AF systems and help to develop new S&T for the benefit of the nation
A core = CPU that reads & executes program instructions; multicore = single computing component with >1 indep. cores

There can be numerous multicore processors in systems of chips (SoCs), so multicore is quickly becoming Many-core – rising exponentially – 2, 4, 8, 16, etc. processor cores in SoCs

Multicore & Many-core systems rapid expansion
  - Ex: Intel’s Core Duo (2006) & Intel’s 2012 Xeon Phi (or Knight’s Corner) have > 50 cores/chip; that’s an increase of 2500% in 6 years!

A 2500% change to the existing state of the art actually constitutes a completely new technology

Due to the rapid change of Systems & Software technology, annual review is needed

Photo from Intel.com
New Support for FY14

• Multicore Real-Time Mixed-Criticality Framework for Avionics
• Scalable, Fault-Tolerant Operating System for Many-core and Multicore Platforms
• Fully Automated Concurrency Refactoring of Legacy Sequential Programs on Chip Multiprocessors: Techniques and Compiler/Virtual Machine
• Adaptive Runtime Verification and Recovery for Mission-Critical Software
• Virtual Machine-level Software Transactional Memory
• Integrated Isogeometric Approach to the Engineering Design and Optimization of Aircraft Structures
• Interframe JPEG-2000 for Motion Imagery Browsing
• Megacore Operating System and Chip Architecture Co-Design
• An Integrated Management Layer to Administer Heterogeneous Resources in Dynamic Workflow Driven Clusters
Multicore Real-Time Mixed-Criticality Framework for Avionics

- **Research Issue:** How to implement & validate real-time avionics software on multicore platforms so that the underlying hardware platform is efficiently utilized

- **Proposed Solution:**
  - Extend mixed-criticality scheduling from uniprocessor platforms to multicore platforms
  - Research resource allocation infrastructures

**PI:** Dr. James Anderson, Professor, University of North Carolina at Chapel Hill
Multicore Real-Time Mixed-Criticality Framework for Avionics

• Approach (overall): Use pessimistic allocation techniques only for software at the highest criticality levels (to make more efficient use of the underlying platform); software at lower levels will be more optimistically provisioned.

CE = Cyclic Executive (table-driven scheduling)
EDF = Earliest-Deadline-First
Approach (continued): Devise techniques based on formal analysis wrt mixed-criticality to:

- Limit the impacts of deviations
- Realize functionality
- Enable mode changes and enable highly dynamic runtime
- Strengthen scheduling guarantees
- Determine an appropriate factoring of the overall resulting framework into real-time operating system and middleware components
- Analyze real-time constraints
Multicore Real-Time Mixed-Criticality Framework for Avionics

• Benefits:
  – Enable certifiable multicore-based flight software in the nation’s next generation of unmanned air vehicles (UAVs)
  – Future UAVs will have far greater autonomous capabilities, be significantly better equipped to adapt to changing environmental conditions, have intensive computational workloads, have system components of varying criticalities, and be subject to stringent certification requirements
  – Expand the state of the art by developing new techniques for multicore, real-time, and/or mixed-criticality frameworks
Scalable, Fault-Tolerant Operating System for Many-core and Multicore Platforms

- Research Issue: How to transparently tolerate operating system (OS) failures on emerging multicore architectures

- Proposed Solution:
  - Research a multikernel-based OS system, where multiple OS kernel instances are booted on multicore hardware
  - Note, Kernel instances:
    - Coordinate to maintain the abstraction of a single-image OS; thus legacy applications can run transparently
    - Can tolerate OS failures: if one instance fails, other instances can continue to serve application requests
    - Can improve performance: if one instance is a bottleneck, other instances can process application requests and improve performance

(Simplified case)
Replica synchronization on OS syscall to ensure consistency; research will study when reduced fault tolerance is acceptable and when faulty behavior is worth checking to improve performance

PI: Dr. Binoy Ravindran, Professor, Virginia Polytechnic Institute & State University
Scalable, Fault-Tolerant Operating System for Many-core and Multicore Platforms

• Approach:
  – Study and experiment with the Linux operating system as a multikernel Linux OS
  – Examine the boot layer, memory management subsystem, and page management subsystem of the Linux OS
  – Develop page coherence and high performance inter-kernel communication techniques, so that different instances of the Linux kernel can run concurrently on different "nodes" of a multicore hardware
Scalable, Fault-Tolerant Operating System for Many-core and Multicore Platforms

• Benefits from outcome:
  – Study will enable future applications to:
    • Run on emerging high core count multicore architectures
    • Transparently tolerate OS failures
    • Achieve high performance
  – New techniques for:
    • Multi-kernel OS design
    • Kernel address space replication
    • Optimistic replication protocols
    • Deterministic thread execution
    • Fast reboot and rapid recovery
    • Fault tolerance algorithms
  – Lead to future research areas in the fields of
    • Petascale and exascale computing
    • Design of fault-tolerant OSs for distributed shared memory environments
• Research Issue: How to automatically optimize large legacy code to exploit the parallelism of emerging multicore hardware

• Proposed Solution:
  – Develop methods to reduce development costs with respect to legacy software and hardware technology research
  – Create basic research tools and techniques for concurrency refactoring
  – Study speculative concurrent execution, in particular using Transactional Memory (TM), augmented with static and run-time program analysis

PI: Dr. Binoy Ravindran, Professor, Virginia Polytechnic Institute & State University
Fully Automated Concurrency Refactoring of Legacy Sequential Programs on Chip Multiprocessors: Techniques and Compiler/Virtual Machine

• Approach:
  – Analyze application programs using compiler and run-time program analysis techniques to automatically understand code segments that are potentially independent
  – Experiment with such code segments under the control of TM
  – If speculation is not correct, the TM system's inherent saving of execution states is leveraged to roll-back and re-execute to ensure memory consistency
  – Studying compiler/run-time infrastructure wrt automatic concurrency refactoring
  – Study a hybrid TM system, with virtual machine-level software TM (STM) and LLVM compiler infrastructure
• Benefits from outcome:
  
  ─ Optimization techniques for large legacy systems software
  
  ─ Improved performance for legacy systems using methods for emerging high core count multicore architectures
  
  ─ Novel automatic refactoring concurrency techniques for a compiler/virtual machine infrastructure
  
  ─ New methods to ensure functional correctness wrt parallelism of chip multiprocessor hardware
  
  ─ New methods for automatic code parallelization
• Research Issue: How to verify mission-critical software to ensure safe execution and completion of mission goals

PI: Dr. Scott Smolka, Professor, State University of New York at Stony Brook
Adaptive Runtime Verification and Recovery for Mission-Critical Software

• Proposed Solution:
  – Development of powerful formal techniques to monitor, analyze, and guide run-time program execution
  – Development of a novel extension of runtime verification in which the runtime verification itself is adaptive

Proposed Simplex architecture for cyber-physical systems
Adaptive Runtime Verification and Recovery for Mission-Critical Software

• Approach:
  
  – Investigate overhead control, incomplete monitoring, predictive analysis, & Simplex architecture for cyber-physical systems
  
  – Evaluate framework performance and utility through significant case studies, including the runtime monitoring of the command-and-control and energy-management infrastructure of a fleet of UAVs
  
  – Study the dynamic allocation of additional resources to the monitoring of high-criticality objects with the goal of increasing the probability of detecting imminent property violations
  
  – Research targeting factors leading to imminent violations:
    • Study the effects of a temporary transfer of control to a baseline system
    • Investigate and activate a repair module to perform online diagnosis and repair of the advanced system
  
  – Research complex adaptive software for improvements to reliability; monitor and analyze the SW’s behavior, environment, and interaction in order to trigger adaptive responses
Adaptive Runtime Verification and Recovery for Mission-Critical Software

• Benefits from outcome:
  – New research wrt high-assurance mission-critical software, especially for software on autonomous unmanned vehicles, where mission success depends on adaptive responses to changing conditions
  – New research in system behavior during testing, debugging, and deployment
  – New research in recovery framework to ensure mission completion in the face of runtime violations
  – New research in flexible and adaptive runtime verification
• Research Issue: How to overcome the scalability and performance limitations of current software transactional memory (STM) systems

• Proposed Solution: To mitigate many of the overhead factors of STMs, such as interference from the garbage collector and high cost of managing transactional meta-data, by investigating and implementing as much of the STM components "inside" a virtual machine, as opposed to a user space library

PI: Dr. Binoy Ravindran, Professor, Virginia Polytechnic Institute & State University
Virtual Machine-level Software Transactional Memory

• Approach:
  – Research the critical components of an STM system, including:
    • Contention management
    • Meta-data
    • Conflict detection
    • Abort/roll-back
    • Closed nesting
    • Open nesting
    • Strong atomicity
    • Conditional variables
  – Design algorithms for a virtual machine-level STM
  – Study advanced programmability wrt STM, such that a transaction will surround any block of code, and will not be restricted to methods
  – Develop operating system-level scheduling algorithms that are transaction-aware for improved STM performance
  – Evaluate application workloads
Virtual Machine-level Software Transactional Memory

• Benefits from outcome:
  – New algorithms wrt a Java VM-based STM
  – New methods to enable programmers to easily program emerging high core count multicore architectures, and achieve high performance
  – New methods for synchronization abstraction to alleviate the difficulties of lock-based concurrency control on emerging multicore architectures – e.g., scalability, programmability, composability
  – New operating system-level scheduling algorithms that are transaction-aware for improved STM performance
Integrated Isogeometric Approach to the Engineering Design and Optimization of Aircraft Structures

- **Research Issue**: To develop fundamentally new approaches to integrated and geometrically exact engineering design & optimization

- **Proposed Solution**: A new isogeometric approach to design, analysis, and optimization utilizing reduced-order models based on T-splines w/ aircraft wing structures as data

PI: Dr. Michael Scott, Assistant Professor, Brigham Young University
Integrated Isogeometric Approach to the Engineering Design and Optimization of Aircraft Structures

• Approach:

  – Leverage recent advances in computer aided design (CAD), analysis, and optimization

  – Develop fundamentally new approaches to the integrated design, analysis, and optimization of multi-physics systems

  – Investigate parametric design of T-spline shell models of aircraft wing structures

  – Analyze, via T-spline-based isogeometric analysis, structural systems composed of beams, plates, and shells

  – Analyze unsteady aerodynamic loads on T-spline shells

  – Research via geometrically exact multi-objective isogeometric optimization of aircraft wing structures
Integrated Isogeometric Approach to the Engineering Design and Optimization of Aircraft Structures

• Benefits from outcome:
  – Elimination of all mesh generation and geometry clean-up steps for the proposed class of problems
  – Introduction of exact geometry to every stage of the engineering design and optimization process
  – Novel and efficient design approaches applicable to conceptual, preliminary, and detail phases
  – Improve state of the art in multi-physics and computer systems
• Research Issue: How to obtain more efficient motion imagery compression as frame rates increase

• Proposed Solution: New motion imagery compression approach that facilitates efficient video browsing

PI: Dr. John Woods, Professor, Rensselaer Polytechnic Institute
Interframe JPEG-2000 for Motion Imagery Browsing

• Approach:
  – Research algorithms to improve Joint Photographic Experts Group (JPEG) 2000 to take advantage of interframe dependence, while preserving JPEG 2000 Interactive Protocol (JPIP) interactivity
  – Research new algorithms to create a new coder will be designed to fully exploit the features of JPEG 2000 interactive protocol for highly efficient motion imagery browsing
  – Research a novel JPEG-2000 coder to support efficient video browsing by maintaining a cache at the user device containing tagged video elements for future use in browsing, as does JPIP for images, thereby eliminating the need for their retransmission, and thus ensuring highly efficient network use
Interframe JPEG-2000 for Motion Imagery Browsing

• Benefits from outcome:
  – More efficient communication and storage for motion imagery browsing at higher frame rates
  – Improvement in intelligence gathering for science, government, and defense wrt massive amounts of motion imagery
  – Improvement in efficiency for future intelligent software agents to troll through massive video data sets
Megacore Operating System and Chip Architecture Co-Design

- **Research Issue:** How to manage Many-core chips and data centers

- **Proposed Solution:** To investigate how to create single system image OS which scales across future Many-core chips & data centers

Multi-core processor with shared multi-tenant accelerator cores connected to on-chip network

PI: Dr. David Wentzlaff, Assistant Professor, Princeton University
• **Approach (some goals listed):**
  
  – **Research software algorithms:**
    
    • Investigate scalability of internal data structures of an OS to Ks of cores
    
    • Investigate abstractions which provide a single-system abstraction across VMs
  
  – **Research hardware improvements:**
    
    • Investigate adding protection hardware into Many-core processors in order to enable ease of OS scaling
    
    • Investigate scalable multi-tenant I/O devices to enable fast access to I/O devices
  
  – **Research Cross-layer Integration:**
    
    • Investigate OS driver capabilities to take advantage of additional OS-specific hardware scalability
    
    • Evaluate the scalability benefits of co-optimized hardware and software
Megacore Operating System and Chip Architecture Co-Design

• Benefits from outcome:
  – Scalability OS characterization on current and future architectures
  – Investigation of the expansion of a scalable OS to the data center
  – Investigation of the commonality between applications and workloads that run in a common data center
  – Research in Many-core specific architectural protection systems
  – Creation of hardware strategies to scale multi-tenant I/O to Ks of cores
  – Creation of hardware strategies to scale reliable memory coherence across a data center
  – Creation of hardware strategies to take advantage of commonality between different applications executing in a data center
  – An evaluation of the benefits of co-optimizing hardware and software
• Research Issue: How to determine the best application platforms
• Proposed Solution: To investigate a new resource management layer between diverse applications and heterogeneous servers

PI: Dr. Ningfang Mi, Assistant Professor, Northeastern University
Integrated Management Layer Research to Administer Heterogeneous Resources in Dynamic Workflow Driven Clusters

• Approach:
  
  – Develop a new capacity modeling methodology to accurately predict application performance & reliability on a given platform
  
  – Develop a new automatic statistic-based maintenance mechanism to assist cluster managers to best utilize idle resources to meet the desired reliability target & maximize energy efficiency
  
  – Design new pattern-based consolidation and burstiness-aware load balancing algorithms to enable cluster managers to efficiently allocate server resources to diverse applications

Markov chain of a model capturing system workloads & failure events

i: number of customers in the system
#: state of the system: Up or Down
Integrated Management Layer Research to Administer Heterogeneous Resources in Dynamic Workflow Driven Clusters

• Benefits from outcome:
  – Development of advanced models for accurately predicting performance & reliability:
    • To help select the best configuration for deployment, and
    • To enable system managers to optimize the performance, reliability & efficiency of the entire data center infrastructure
  – Research in new resource utilization algorithms to best utilize:
    • “Idle” resources for maintenance tasks &
    • “Busy” resources for regular user tasks
  – Improvement of system reliability & energy optimization
  – Avoidance of service level agreement violations under bursty workloads
  – Development of benchmarks to evaluate cluster performance, dependability & efficiency
Conclusion

• Systems & Software is the cornerstone of all S&T

• Basic research in Systems & Software is critical as improvements to legacy systems and exploration in cutting-edge future research are both needed

• In order to maximize the Systems & Software funding, a two-directional approach was implemented – support both legacy system research and cutting-edge research

• As such, areas such as multicore, operating systems, concurrency refactoring, VMs, TM, software verification & recovery, run-time aspects, and compression algorithms were selected for support for FY14-FY16
Back-Up
Glossary

• Consistency Check: also called synchronization with consistency check is the process by which the data protection manager checks for and corrects inconsistencies between a protected volume and its replica; performs block-by-block verification to ensure that all the data on the replica is consistent with the protected data

• JPEG: Joint Photographic Experts Group – method of lossy image compression (discrete cosine transform-based standard)

• JPEG 2000: improved image compression standard based on wavelet method

• Kernel: computer program that manages input/output requests from software and translates the requests into data processing instructions for the CPU, etc.

• Operating System (OS): software that manages hardware resources and provides common services for computer programs

• Optimistic techniques: process a transaction prior to consistency checking

• Pessimistic techniques: consistency check prior to processing a transaction

• Platform: crucial element in software development; includes hardware architecture and software framework needed to run software; typical platforms include a computer architecture, operating system and runtime libraries

• Transactional Memory: concurrency control mechanism to simplify programming and manage conflicts with respect to atomic execution of groups of load and store instructions

• Virtual Machine (VM): software implementation that executes programs like a physical machine
Photos and references

• Cyber dominance photo from AF website

• Intel Xeon Phi Coprocessor Block Diagram: http://www.intel.com/content/www/us/enprocessors/xeon/xeon-phi-coprocessor-block-diagram.html

• All project photos were supplied by the respective PIs