OPERATING SYSTEMS AND SYSTEMS SOFTWARE

Operating System Kernels

University of California – Berkeley; Thomas E Anderson; *Presidential Faculty Fellow (PFF)*; (CCR-9453532 A001); $100,000; 12 months.

The project will (1) investigate novel approaches to providing extremely efficient communication between protected domains running on the same machine, (2) develop tools for helping programmers tune the performance of applications limited by cache and virtual memory behavior, (3) develop new approaches to building ATM network switches to make them both easy to implement yet capable of delivering high performance, (4) investigate file system designs to improve scalability and to reduce power consumption, and (5) produce software for teaching undergraduate operating systems.

University of Washington; Brian N Bershad; *Presidential Faculty Fellow (PFF)*; (CCR-9453330 A001); $100,000; 12 months.

The project develops an operating system called SPIN that provides fine-grained control to applications over physical resources in ways that are safe, predictable and efficient. The SPIN system is structured around an extensible microkernel architecture. The system is extensible in the sense that application programs can dynamically install restricted program sequences that execute in the kernel in response to a set of predefined events allowing for flexible and rapid response. For example, an application program could install a scheduling filter that runs each time a thread from that program's address space is preempted. The notion of event filtering can be applied to a wide class of events and services, including virtual memory, scheduling, network communication, local interprocess communication and disk management. By viewing system resource management as a process by which physical resources are reassigned across applications, and interpreting each reassignment as a discreet event, SPIN defines critical control points for application feedback, enabling dynamic resource allocation policies. By associating application-defined control sequences with these control points, SPIN enforces a wide range of policies at each event, enabling a single operating system to work well across many types of applications. By executing the control sequences within the kernel, the dynamic control is achieved with very low overhead.

Rice University; Peter Druschel; *CAREER: A Communication-Oriented Operating Systems for the National Information Infrastructure* ; (CCR-9503098 A000); $133,650; 36 months.

This project will design, implement and experiment with an operating system that satisfies the requirements of emerging applications such as multimedia, parallel computing, and distributed information retrieval. These performance critical applications can be characterized by their intense use of communication, and by their need for tight control of the machine's resources. Existing operating systems often limit the performance of such applications, thereby preventing users from reaping the power of high-performance computer systems. As a result, lagging operating system technology threatens to delay the acceptance of parallel computing, multimedia, and other services in the emerging national information infrastructure. The major themes of the project are (1) designing OS communications support that delivers performance close to that provided by the network; (2) achieving OS performance that scales with the underlying computer system's performance; and, (3) investigating open service implementations that allow customization of OS services to match precisely an application's requirements. The goal is to deliver the resources provided by high-performance computer systems to applications, without sacrificing functionality, fairness, safety, and portability. Methods include the re-evaluation of current operating systems in light of changing application requirements and hardware performance characteristics; design, implementation, and experimentation with appropriate solutions; and, dissemination of results through publication and distpibution of software artifacts. Education plans of this CAREER project include the development of an interactive teaching style, introduction of a hands-on course in parallel programming using networks of workstations, and the organization of a summer program designed to expose junior high students to computer and information technology.

Carnegie Mellon University; Elmootazabella N Elnozahy; *CAREER: System Software Design for Energy Conservation*; (CCR-9502933 A000); $130,392; 36 months.

Energy conservation is an important problem in battery powered portable computers. The objective of this project is to investigate software system techniques for conserving energy, using the knowledge about current workload and application requirements. More specifically, the project will: (1) Develop a systematic approach for evaluating energy conservation techniques whether they are implemented in hardware or software. The evaluation will guide the energy driven hardware and software design. (2) At the operating system level, develop techniques to control the devices using the mechanisms provided by the hardware to modify the behavior of common services for lower energy consumption. (3) At the architecture level, design memory architectures to permit trade-off between performance and energy consumption. (4) Develop file systems that exploit information about applications' usage patterns and the availability of low- power solid-state memory. (5) At the application level, design program interfaces for conveying information to operating system about the usage patterns and performance requirements. A combination of simulation and actual implementations will be used to evaluate the techniques described above. The educational component of this CAREER project will include the development of laboratory-oriented operating systems courses for both undergraduate and graduate students. In addition, undergraduate students enrolled in independent studies will be encouraged to participate in the research and development of software tools.

Massachusetts Institute of Technology; M. Frans Kaashoek; *NSF Young Investigator (NYI) Award:: Operating Systems for Multiscale Computers*; (CCR-9457791 A001); $75,000; 12 months.

Advances in VLSI technology have made distributed systems (e.g., clusters of workstations) suitable for parallel computing. Today's software, however, limits the functionality of distributed systems. The goal of this research is to discover and evaluate techniques for (1) automatic management of locality and work distribution, (2) fault isolation, (3) high-performance communication, and (4) fair sharing in a cost-effective distributed system. The approach is based on shared objects, an exokernel-based operating systems, a user-level communication unit, and gang scheduling. The project builds an experimental distributed system that delivers performance comparable to custom-designed supercomputers, while maintaining fault isolation and fair sharing.

University of Washington; Edward D Lazowska, Henry Levy, and John Zahorjan; *System Support for High Performance Computing*; (CCR-9200832 A003); $408,796; 12 months. (Joint support with the New Technologies Program, the CISE Institutional Infrastructure Program, the Software Engineering Program, and the Computer Systems Architecture Program)

The fundamental issue addressed in this work is the structuring of operating systems to take advantage of multiprocessors so as to yield the best possible application performance. The division of labor between the kernel and the user level is different on a multiprocessor than on a uniprocessor. On a uniprocessor, the kernel is responsible for address spaces, communication, resource allocation, threads, and thread scheduling. On a shared memory multiprocessor, the kernel iis left in charge of address spaces and resource allocation, but communication, threads, and thread scheduling are implemented at user level through runtime support in each address space. Furthermore, each address space is in direct control over certain key aspects of virtual memory management and file management. It is important to understand that while "micro-kernel" designs such as Mach 3.0 take the important step of moving the file system, the virtual memory system, etc., out of the kernel to user-level servers, they leave cross-address space communication, threads, and thread scheduling in the kernel. There are structural differences between these monolithic kernels, micro-kernel designs and the "nano-kernel" advocated here. In experimental work over the past several years, the researchers have constructed proof-of-concept prototypes in most of the essential areas that must be addressed: locking protocols, user-level threads, cross-address space communication, the kernel interface, processor allocation policies, and user-level virtual memory management. The objectives in this work are of two kinds, direct and indirect. Direct objectives include: (1) integrating these proof-of-concept prototypes into a single system, convincingly demonstrating the viability of the overall approach; and (2) solving the problems that stand in the way of this integration, some of which are known and some of which can only be discovered by undertaking the task. Indirect objectives include: (a) using the "core" of the resulting system to study a number of important policy questions (thread scheduling, processor allocation, virtual memory management) that can only be explored given such an experimental testbed, and (b) using this core as a base for exploring high performance system support for more advanced architectures, specifically large scale machines with distributed memory, distributed/parallel programming using high band-width low-latency networks, and wide-address machines.

University of Arizona; Sean O'Malley, Todd A Proebsting, and Larry L Peterson; *Scout: A Communications-Oriented, Compiler-Assisted Operating System*; (CCR-9415932 A000); $280,444; 36 months.

This project deals with the design, implementation and evaluation of a new operating system called Scout, which is designed to meet the requirements of the systems connected to the National Information Infrastructure. The high level goals include, reliable delivery of data to applications at high bandwidths, and scalability with increasing number of processors. Scout is based on abstraction called path which extends the network communication into the host operating system: resource allocation, scheduling and fault-isolation are done on a path basis. Scout will address the issue of scalability by developing operating system specific compiler technology to optimize applications to specific operating systems. Two main deliverables of the project are the Scout operating system itself and the set of tools for the development of new operating systems.

Parallel and Distributed Systems

University of Arizona; Gregory R Andrews; *Software Support for Portable Parallel Computing*; (CCR-9415303 A001); $115,950; 0 months.

The project addresses the problem of writing parallel programs that are clear, portable, and efficient. The three common approaches to this problem are parallelizing compiler, very high-level languages and imperative languages or software systems with a single communication model. This project develops a common base of programming abstractions and underlying software that simplifies and supports all of these approaches. The centerpiece of the research is the Filaments package, which supports efficient execution of fine-grain parallelism and a shared communication on both shared and distributed memory multiprocessors. The specific topics to be investigated are: (1) expanding and refining Filaments package, (2) designing and implementing MIF, a multi-paradigm intermediate for compilers that will be both language and architecture independent; and (3) designing and implementing a simple and efficient programming language HPSR (high performance SR). Filaments and MIF address the problems of portability and efficiency, independent of the language used to write parallel programs, and HPSR addresses the problem of program clarity.

Stanford University; Mary G Baker; *CAREER: The MosquitoNet Mobile Computing Project*; (CCR-9501799 A000); $135,000; 36 months.

The research project will investigate operating system and application issues in a network that includes mobile and wireless hosts. Recent advances in wireless technologies and lightweight computers make it attractive to build networks that combine traditional workstations and servers with portable computers. But portables will not achieve equivalence to stationary hosts unless the portables can remain continuously connected to the network, even when changing locations or communication devices. The project, called MosquitoNet, will increase the connectivity of mobile hosts using the following approaches. (1) Enable portable computers to move seamlessly from one communications medium to another; for example, from an Ethernet connection to a wireless modem, without rebooting or restarting applications. (2) Determine the feasibility of hiding the resulting changes in throughput, latency and other network characteristics from applications. (3) Design a simple interface to expose these changing characteristics to applications as necessary. (4) Examine communication device usage patterns for use in implementing battery life extension policies. (5) Develop new file data consistency algorithms for use in a distributed mobile environment where disconnection is allowed but not mandatory. The educational component of the CAREER project will modernize and vitalize the teaching of operating systems and distributed systems at the Stanford University. In particular a new course on wireless and mobile computing will be introduced along with the introduction of new teaching tools, topics and projects into the current curriculum on opepating systems and distributed systems.

Ohio State University ; Gregory D Burns and Charles F Bender; *Software Capitalization Funds for a Cluster Implementation of MPI*; (CCR-9510016 A000); $84,959; 12 months. (Joint support with the New Technologies Program)

The Ohio Supercomputer Center (OSC), under the direction of the PI's, will develop comprehensive documentation for MPI (Message- Passing Interface) cluster software called LAM. LAM is currently supported on networks of workstations. The documentation will enable other researchers in academia to utilize LAM for building parallel processing applications quickly. The project will also port the software to low-end UNIX platforms and high- end shared memory multiprocessors systems supporting Symmetric Multiprocessing (SMP) UNIX. Small business users will benefit from the availability of the software on low-end workstations. The project is supported under the Software Capitalization Grants program.

University of Illinois – Urbana; Roy H Campbell, Andrew A Chien, Simon Kaplan, and Daniel A Reed; *Broadband Network Infrastructure and Data Storage for Research and Education*; (CDA-9401124 A001); $448,850; 12 months. (Joint support with the CISE Institutional Infrastructure Program)

This award provides support for an experimental, high-speed network infrastructure that interconnects parallel systems, large-capacity secondary and tertiary storage devices, and muli-media workstations capable of supporting high-quality digital audio and video. The network architecture is based on a broadband ISDN backbone of Asynchronous Transfer Mode (ATM) switches and fiber optics links. A high-bandwidth, gigabit/second HIPPI network interconnects an Intel Paragon distributed memory parallel system and a tertiary storage system. The research spans three basic areas: system software infrastructure for managing high-speed networks, data pre-fetching and staging policies for high-latency, high-capacity secondary and tertiary data stores, and information access and multimedia collaboration software.

Syracuse University; Alok Choudhary; ***NSF Young Investigator (NYI) Award: Compiler and Runtime Optimization Techniques for Parallel Programming on Distributed Memory Machines*****; (CCR-9357840 A002); $62,500; 12 months.**

The objectives of this research are to develop compiler and runtime optimization techniques for parallel programming on homogeneous and heterogeneous high-performance parallel computers. Specifically, this research addresses the following issues. (1) Scalable Parallel I/O: Scalable massively parallel computers must include scalable I/O systems to provide a balance between computing power and I/O. The thrust of this research is to investigate compiler transformation for parallelizing I/O operations, stripmining and prefetching data, overlapping computation and I/O, generate efficient I/O access schedules, develop runtime support that can recognize and use access patterns to perform efficient parallel I/O, and develop language primitives that allow users to specify parallel I/O operations. (2) Automatic Data Distribution for Homogeneous and Heterogeneous Systems: The goals are to develop algorithms to automatically generate data distributions that incorporate global analysis, use realistic cost models, and that allow dynamic alignments, distributions and data replication. (3) Runtime Support for Dynamic Data Redistribution: Dynamic data redistribution may be necessary due to explicit program constructs, compiler generated distributions, across procedure boundaries, or adaptive nature of a problem. The goals are to develop parallel algorithms (including communication scheduling, index conversion, and space optimization) for runtime support for dynamic data redistribution.

Rice University; Alan Cox; ***NSF Young Investigator (NYI) Award:: High Performance Distributed Shared Memory Multicomputers*****; (CCR-9457770 A001); $62,500; 12 months.**

The goal of this research is to make parallel computing simple and efficient on general-purpose, commodity processors and networks. Recent work has focused on algorithms for the efficient implementation of software distributed shared memory (DSM) on low-cost, high-performance workstation clusters. This work has shown that software DSM can achieve performance close to a hardware implementation of shared memory on a variety of scientific applications that synchronize with moderate frequency. The objective is to improve this work in two ways; (1) to achieve the efficient execution of applications that synchronize with greater frequency and, (2) to support languages ordinarily used to solve symbolic problems. Specifically, the following approaches are evaluated: (a) run-time support for programming high-level synchronization operations, such as task queues, that reduces the amount of communication by the DSM system, (b) compiler support to reduce the amount and/or hide the latency of communication by the DSM system, and (c) garbage collection algorithms integrated with the DSM system to enable the efficient execution of advanced programming languages.

University of Tennessee – Knoxville; Jack J Dongarra, Michael R Leuze, Michael A Langston, Michael D Vose, and Michael G Thomason; ***An Experimental Research Facility for Parallel Computing*****; (CDA-9115428 A004); $275,000; 12 months. (Joint support with the CISE Institutional Infrastructure Program and the ARPA-MESSAGE PASSING INTERFACE Program)**

The advanced Computing Laboratory in the Computer Science Department at the University of Tennessee has access to a number of new architecture high performance computers. This award is to upgrade the facility by the addition of a mixed mode Single Instruction Multiple Data (SIMD)/Multiple Instruction Multiple Data (MIMD) computer, increasing the number of processors in a shared memory MIMD computer, and providing partial support for a technician. The research supported by these computers include the design and development of portable, efficient numerical linear algebra algorithms; the development of non-numeric algorithms for parallel architectures; the development of parallel algorithms for genome sequencing; and the development of parallel discrete optimization algorithms using the genetic algorithm method.

Princeton University; Edward W Felten; ***NSF Young Investigator (NYI) Award:: Efficient Communication for Distributed Parallel Processing Applications*****; (CCR-9457813 A001); $62,500; 12 months.**

The research has two parts: systems software and applications. In the systems software area, the inherent complexity and cost of communication and synchronization between distributed processors is studied. This work includes advanced implementations of communication software in experimental parallel architectures, including tools for generating specially-tailored communication mechanisms for particular parallel applications. The goal is to understand how best to support communication in future systems. Collaboration with researchers in other scientific fields to develop application programs that effectively use the capabilities of today's high-performance parallel computers is the goal of the applications aspects of this project.

Washington University; Kenneth Goldman; ***High-Level Communication for Distributed Multimedia Applications*****; (CCR-9412711 A001); $10,000; 0 months.**

The project takes a new approach to simplifying the construction of distributed multimedia applications. The I/O abstractions put forth separate the description of the communication structures from the computational components of a system, such that software modules written in existing programming languages are integrated flexibly and dynamically. The logical connections among the data interfaces of independent software modules provide a uniform high-level view of communication for both discrete and continuous data. The high-level view of communication is not only intuitive for users, but it also offers a means to establish protection for data and applications and provides opportunities for low-level optimizations. The project develops the software support needed for implementing the I/O abstractions, and evaluates using applications in four domains: digital libraries, collaborative work, simulation and medical imaging. The Washington University's high speed ATM network is used as a testbed for the development work.

Dartmouth College; David F Kotz; ***High-Performance File Systems for Scientific Multiprocessing*****; (CCR-9404919 A001, A002); $24,286; 6 months.**

In this project, an extensible and portable framework for a file system to make high-level I/O operations feasible is developed and implemented. The project evaluates the policies and mechanism using both analytical models and simulations of a variety of parallel architectures. Massively parallel computations on today's parallel computers are limited by their file systems. The performance of the file and disk systems can be improved by clever hardware and software support. I/O performance can be improved by making it possible for the programmer to specify I/O request at a high level. This is similar to how the matrix operations in many parallel languages specify the required semantics of high level operations to compilers, such that the compilers, run-time systems, operating system and the hardware coordinate the I/O to optimize the number and the order of accesses to disks, usage of buffers, layout of files on disks, and so on. These techniques of specifying at higher level can be generalized to other forms of I/O like the network I/O.

University of Wisconsin – Madison; James R Larus; ***NSF Young Investigator (NYI) Award: Programming Massively Parallel Computers*****; (CCR-9357779 A002); $87,069; 12 months.**

The goal of this research is to make massively-parallel computers easy to program. This means that the natural users of these machines; scientists and engineers working on large-scale, state-of-the-art research, as exemplified by the Grand Challenge Problems routinely exploit these computers as a research tool. The problem of programming massively-parallel computers is attacked from three directions: (1) collaborating with computer architects on shared-memory architectures that facilitate programming and compilation, (2) developing new parallel languages that permit more natural expression of parallel algorithms and the techniques to implement them, and (3) building performance evaluation and debugging tools to help programmers to understand and increase their program's performance and correctness.

Clark Atlanta University; Alvin S Lim; ***CAREER: Operating System Support and Programming Environment for Evolutionary Parallel and Distributed Applications*****; (CCR-9502345 A000); $102,181; 36 months.**

The project will create operating system support and a programming environment for developing and maintaining long-running parallel and distributed applications that are continually evolving, such as automated manufacturing and scientific computation. Supporting changes at runtime requires an efficient, dynamic reconfiguration facility and an extensible programming environment. The scope of the research includes building toolkit at the system level, end-user environment for developing applications including algorithm design, prototyping, and evaluation of the efficiency and usefulness of the facility. The toolkit will utilize partial-order semantics and is based on the hierarchical state machine formalism. The programming environment is based on a scalable software architecture for specifying and verifying application behavior. The educational component of the CAREER project will develop advanced and introductory courses, and computer laboratories on parallel processing. These developments, present effective venue for providing advanced education and research training to graduate and undergraduate minority students and equipping them to meet challenges in a world that is increasingly dependent on distributed computer systems.

Bradley University; Jiang B Liu; ***Research at Undergraduate Institutions (RUI): Clustered Computer Network Control, Scheduling, and Management*****; (CCR-9504137 A000); $104,981; 36 months.**

Distributed computer networks control, scheduling and management are well-studied topics in scientific and engineering applications, yet the problem of finding optimal approaches that are not too complex to solve or implement on practical systems remain unsolved. Many computer networks are naturally clustered, and imbalance of load usually occurs among the clusters rather than inside the clusters. This project will develop clustered computer networks' control, scheduling and management policies for allocating computational resources across network clusters and for managing network operations based on simple network management protocol(SNMP). This project will first develop clustering model based on the data collected from monitoring Bradley network loads from real applications. Then the project will investigate and develop adaptive computer algorithms and heuristic schemes for developing the control policies. Finally, the project will develop network load monitoring tools based on X- window and Tcl/Tk for network manager interface. The control schemes developed are expected to be applicable to network load balancing and management in many other clustered distributed systems. The project is funded under the RUI program.

University of Texas – Austin; Gregory J Rodin, J. Tinsley Oden, Robert A van de Geijn, and James C Browne; ***Application of High Performance Computational Methods to Micromechanical Characteristics of Composite Material*****; (ECS-9422707 A000); $422,825; 12 months. (Joint support with the New Technologies Program, the Software Engineering Program, the Numeric, Symbolic & Geometric Computation Program, the Computational Engineering Program, and the Solid Mechanics Program)**

Design of composite materials to have desired properties is a task of great engineering significance. Presently this is usually accomplished through trial and error laboratory synthesis and ad hoc analysis with the result that non-optimal design is commonplace. The dramatic increase in computational power available for mathematical modeling and simulation raises the possibility that computationally-based design can play a significant and complementary role to laboratory development of compmsite materials, as has been the case in other engineering design problems. The computational problem is to determine the overall properties (not necessarily linear elastic) of the composite material from the constituent properties. The goal of the proposed research is to create a toolbox of computational components with which engineers can design and analyze composite materials directly form the microstructural level. The approach taken in this proposal is the adapt the emerging methods for high performance parallel computation, being developed for computational modeling in other domains, to micromechanical analysis of composite materials. The application of large-scale computation to this calls of problems has been impeded by the fact that neither available computation power nor available computational methods were adequate for realistic micromechanical models. The methods to be adapted in this project are: (i) parallel adaptive finite element and boundary element methods for computational fluide dynamics and acoustics, and (ii) parallel fast multipole methods emerging form many-particle dynamics. The most significant properties of these methods are: Superalgebraic convergence rates, permitting effective simulations with models requiring a minimum computational effort to achieve acceptable accuracy's. Amenability to common infrastructure for efficient parallel implementation. Development of an initial version of this infrastructure under pervious research is one critical enabling factor for this proposal. Natural applicability to micromechanical modeling of composite materials since they perform similar roles in their current domains. The methods are complementary in their domains of application to composite materials. The development of the methods and of the toolbox will be empirical and example driven.

Security, Reliability and Fault Tolerance

University of Iowa; Sukumar Ghosh; ***Dynamic Distributed Systems*****; (CCR-9402050 A000); $135,000; 36 months.**

The objective of this project is to develop a hierarchical design methodology for dynamic distributed systems, and investigate its applicability to several application areas. One such application is the design of gracefully degradable systems that guarantee acceptably good performance in the presence of failures. Another application that will be investigated focuses on the design of distributed algorithms for dynamic graphs and computational geometry. The project will demonstrate how the hierarchical design helps synthesizing provably correct systems in the presence of failures, reconfigurations and how existing algorithms can be reused at appropriate layers.

University of Michigan – Ann Arbor; Farnam Jahanian; ***CAREER: Probing and Fault Injection of Distributed Systems*****; (CCR-9502341 A000); $128,299; 36 months.**

This research will investigate fault injection techniques for the evaluation and validation of dependable protocols. The project will focus on techniques that can be used to demonstrate three aspects of a target protocol: (i) detection of design or implementation errors, (ii) identification of violations of protocol specifications, and (iii) insight into design decisions made by the implementors. The end goal of the project is the investigation of techniques for testing the fault-tolerance capabilities of dependable protocols and the development of a tool for the experimental evaluation of commercial and prototype protocols. The approach will center around a technique called"script-driven probing and fault injection." A distributed protocol can be viewed as an abstraction through which a collection of participants communicate by exchanging a set of messages. Hence, no distinction is made between application-level protocols, communication protocols, or device layer protocols. A probe/fault injection (PFI) layer is inserted below a target protocol to filter and to manipulate the messages that are exchanged between participants. The fault injection layer supports the execution of deterministic or randomly-generated test scripts to probe the participants and inject various faults into the system. In particular, by intercepting messages between two layers in a protocol stack, the fault injection layer can delay, drop, reorder, duplicate, and modify messages. Furthermore, it can introduce spontaneous messages into the system to probe the participants and to orchestrate the system execution into a particular path. Since timing constraints are often an important component of the dependability requirements on a system, the intrusiveness of a fault injection mechanism on a target system is a key issue which will be addressed by this research. The educational component of this CAREER project will include "undergraduate mentoring" and "learning via experimentation." Mentoring exposes students to one-on-one technical interactions with instructors and more senior students. Learning via experimentation has a two pronged effect. It allows students to "practice" how to apply knowledge and to"learn" how to obtain new information. These two aspects of the plan will be formalized into the curriculum with the following educational activities: (1) Refocus the Undergraduate Directed Study to involve junior and senior-level students in some of the ongoing prototyping activities in research projects and in various laboratories on campus; (2) Develop a curriculum track that emphasizes experimentation to enhance learning of software systems concepts.

University of Pittsburgh; Daniel Mosse; ***Research Initiation Award (RIA): An Integrated Framework for Fault-Tolerant Applications in Real-Time and Non-Real-Time Systems*****; (CCR-9308886 A002); $4,950; 0 months.**

This research studies a new model of computation, which extends the object-oriented model with timing constraints and resource requirements specifications. A set of scenarios (a set of applications executing at a time) and a scenario change mechanism that allows for different sets of applications to be executed according to the situation are defined. These scenario changes are triggered by an error handler, by an operator or by a user. Scenario changes also allow uniformity in multiple types of fault-tolerant behavior such as graceful degradation, load-shedding, etc.. The latency of these scenario changes and quantify their applicability to real-time and non-real-time systems are studied. Methods are developed for the detection of situations in which these scenario changes can be applied, and develop a theory for the adaptation of these scenario changes to non-time-constrained applications and systems. More accurate models are developed based on real-life workloads and situations, to model realistic fault models, encompassing transient as well as permanent faults. These are of temporal or physical nature (e.g., transient overloads, site crashes or link failures). The implementation of a prototype of a fault-tolerant real-time system that supports this model of computation follows.

Winston – Salem State University; Atanas A Radenski; ***Research at Undergraduate Institutions (RUI): Parallel Object-Oriented Programming with Multiple Inheritance*****; (CCR-9509223 A000); $36,257; 12 months.**

The transition from sequential object-oriented programming to parallelism has been the focus of research over the past decade. Experimental languages that try to integrate objects and parallelism often compromise the inheritance capability for parallel objects. Even the languages that permit some parallelism and inheritance support only single-class inheritance. In this project support for multiple-class inheritance for parallel objects will be investigated. The technique employed is based on "separate" methods which generate processes and provide rendezvous type coordination. The project will apply this technique to Oberon language. The research will be performed under the RUI program.

George Mason University; Ravinderpal S Sandhu; ***Role-Based Access Control - Models, Mechanisms and Tools*****; (CCR-9503560 A000); $150,000; 36 months.**

This project will pursue a combination of basic and applied research on role-based control (RBAC), using a three-pronged approach, by developing 1) models, 2) mechanisms and 3) tools, to support the effective deployment of RBAC in future systems. The basic concept of RBAC is that users are not directly granted access permissions. Instead permissions are granted based on roles, and users are made members of appropriate roles. This simplifies management of authorizations. RBAC owes its origins to multi-user and multi-application on-line systems of the early 1970's. There has been a recent resurgence of interest in RBAC stemming from its use at application level. This project will provide a sound scientific and engineering foundation for providing general purpose facilities for using RBAC, both at the operating systems level and applications level.

Cornell University; Sam Toueg; ***Unreliable Failure Detectors for Reliable Distributed Systems*****; (CCR-9402896 A000); $90,000; 12 months. (Joint support with the Computer Systems Architecture Program and the Theory of Computing Program)**

The starting point for this research is a fundamental problem in fault-tolerant distributed computing: reaching agreement among processes in a system that is subject to failures. It is well-known that this problem, called Consensus, has no deterministic solution in asynchronous systems, even it is assumed that communication is reliable and no more than one process may fail. The impossibility of solving Consensus (and other related problems such as Atomic Broadcast) is one of the most severe obstacles to implementing fault-tolerant applications in asynchronous systems. In recent work the PI has introduced a novel approach to circumvent such impossibility results: he showed that unreliable failure detectors can be used to solve Consensus (and Atomic Broadcast), even if the information that they provide about failures is highly inaccurate, e.g., even if they make an infinite number of mistakes. Since such failure detectors can be implemented in realistic distributed systems, and since various considerations make the asynchronous models especially attractive, this work suggests an approach to fault-tolerance that is viable in practice. The objectives of this research are to broaden the applicability of this approach by removing the limitations of the earlier work, and to explore in more concrete terms its practicability. Specific goals include: tolerating communication failures, including network partitions (the earlier work assumed reliable links); tolerating process failures of various types (the earlier work dealt with crash failures only); considering shared-memory systems (the earlier work dealt with message-passing systems); and solving other problems that are central to fault-tolerant distributed computing, including Group Membership and Group Multicasts (the earlier work solved Consensus and Atomic Broadcast). In order to assess the cost and benefit of using unreliable failure detectors complexity questions will also be explored. Finally, the practicality if this approach will be validated by implementation on an experimental platform.

Michigan State University; I-Ling Yen; *Research Planning Grant (RPG): Systematic Integration of Fault Tolerance in High Performance Parallel Programs*; (CCR-9521419 A000); $17,985; 18 months.

In this Research Planning Grant, the PI will develop a research plan in the area of systematic fault-tolerant design. The research is based on the idea to progressively relax the system invariant assertion in the presence of processor failures. When there are no failures, a strong assertion will be established and the system will operate at peak performance. As processors fail, weaker assertions are ensured, and the system may operate at a reduced performance. During the planning period algorithms to implement the multiple invariant assertions approach to fault- tolerance in multiple server systems will be developed. A successful completion will lead to an expansion of the research to permit a wider class of failures, as well as techniques for more complex transactions supported by the multiple server systems.

Performance Evaluation, Scheduling, and Resource Management

University of Baltimore; Hossein Arsham; *Research at Undergraduate Institutions (RUI): Optimization of Discrete Event Simulation Models: Single Run Based Approaches*; (CCR-9505732 A000); $23,587; 12 months. (Joint support with the Theory of Computing Program)

The aim of this project is to obtain new results which will help reduce the time and computational effort required when using the computer simulation models for optimization found in many discrete-event systems. The focus is on the enhancement of the Score Function and Likelihood Ratio methods to provide accurate estimates for derivative and perturbation, respectively. The goals are improvements the existing variance reduction techniques, theoretical unification, some extensions, and advancements in the practical implementation of these "single run" methods. The project will develop procedures and algorithms, investigate their accuracy versus execution time behavior, and present numerical results for a wide variety of queuing, inventory, and reliability models.

Harvard University; J. Bradley Chen; *CAREER: Understanding Behavior in Commodity Software Systems*; (CCR-9501365 A000); $109,048; 36 months.

Commodity software differs in fundamental ways from the workloads typically used in academic research to measure hardware and software behavior in computer systems research. In this project the behavior of commodity software systems will be studied. The project will develop tools for measuring behavior in commodity software, then apply them tools to important problems in computer systems design.

University of Illinois – Urbana; J.W S Liu; *Meeting End-To-End Deadlines in Distributed Real-Time Systems*; (CCR-9224269 A002); $120,204; 12 months.

A comprehensive set of algorithms is developed for end-to-end scheduling in distributed real-time systems of different sizes, complexity, and dynamics. These algorithms are thoroughly analyzed and evaluated in order to determine their capabilities and limitations. In addition, rigorous design rules and selection criteria governing their use are derived. This should help to develop a basic framework in which strategies for end-to-end scheduling, load balancing, concurrency control, and synchronization are integrated to enhance the responsiveness and robustness of the overall system.

Stanford University; Mendel Rosenblum; *NSF Young Investigator (NYI) Award: Storage Management for High Performance Computer Architecture*; (CCR-9257104 A003); $62,500; 12 months.

This project addresses research in storage management for high performance computer architectures. There is need for technology for comparing storage managers relative to a given workload characterization to determine which will perform best. Research techniques to accomplish this are based on a combination of analysis, simulation, and prototype implementations. Applications such as audio and video are imposing new requirements for performance and guaranteed response times. The introduction of object oriented systems introduces new and different request patterns. This illustrates the direction of new hardware technology and new application demands that will provide a focus for the research.

State University of New York – Buffalo; Wei Wennie Shu; *Runtime Incremental Parallel Scheduling for Solving Irregular and Dynamic Applications*; (CCR-9505300 A000); $94,956; 36 months.

In global parallel scheduling all processors cooperate to schedule work. This should be contrasted with dynamic scheduling since it generates a well-balanced load without incurring a large overhead. This project will explore a new method, calling Runtime Incremental Parallel Scheduling (RIPS). In RIPS, the system scheduling activity alternates with the underlying computation work. Tasks are incrementally scheduled in parallel by the cooperating processors. This project aims at a framework for study of global parallel scheduling and its related techniques. The project includes the development of parallel scheduling algorithms, the control of low parallelism, the design and implementation of a preemptive scheduling system, and solutions to irregular applications. The objective is to show that runtime global parallel scheduling, as a new scheduling approach, can adapt to irregular and dynamic problems and produce high-quality scheduling.

Application of Formal Techniques

Cornell University; Bard Bloom; *Operational and Equational Semantics of Process Algebras*; (CCR-9223183 A003); $87,282; 12 months.

Process algebras are a basic tool in the theory of concurrency: a mathematical model simple enough to analyze and understand, yet complex enough to illustrate many of the subtleties and problems of concurrency. They have been used for theory, programming language design, and specification and verification languages. For sequential languages, a few basic models of computation suffice. However, there are a vast variety of forms of concurrency, and hence a vast variety of process algebras. It if therefore essential to: (1) Understand the connections between the many schools of process algebra. (2) Have theoretical and computer tools for designing and analyzing process algebras. (3) Understand commonalties and metatheory of process algebra, to avoid unnecessarily repeating the same work in many settings. A general framework for investigating process algebras in terms of their structured operational semantics (SOS) has be formulated by the investigator and a surprising connection between SOS and equational definitions was established. Briefly, it states: Given a SOS specification of a process algebra, a complete equational axiom system for that algebra may be mechanically derived. Furthermore, the derived axiomatization is frequently comparable to the best equations discovered by humans. This result covers the one of the purest cases (technically, "GSOS languages and strong bisimulation"). However, the principles should apply to almost all process algebras and notions of equivalence. In particular, operations which respect the most commonly-used process equivalences should be axiomatizable from their SOS specifications. It is believed that this connection is theoretically fundamental and of substantial practical value to researchers. A full elaboration of this paradigm should unify two major communities in process algebra, and provide valuable tools to researchers in concurrency.

University of Texas – Austin; Vijay K Garg; *Detection of Global Predicates in Distributed Programs*; (CCR-9520540 A000); $57,115; 12 months.

The objective of this project is to study theoretical and practical issues in the evaluation of general globel predicates and functions of distributed software. As an application of the techniques of the research, the project will focus on distributed monitoring systems for fault tolerance. An example of detecting a global predicate arises in implementing the most basic command of a monitoring system: "stop the program when the predicate is true". To stop the program it is necessary to detect the predicate q, which is not a trivial task if q requires access to the global state. There are three components to the project: theory, algorithms and implementation. There are two questions that will be investigated in terms of theory. First, the class of global predicates that are efficient to detect will be classified. This will also entail studying the limits of efficient detection of global predicates. Second, a predicate logic which is suitable for efficient detection of global predicates will be developed. After developing the logic, efficient decentralized algorithms for detecting occurrence of logic formulas will be designed. This work will be based on the use of a centralized checker process developed by the PI. Finally, the decentralized algorithm will be used to design an experimental distributed monitoring system.

Massachusetts Institute of Technology; Nancy Lynch; *A Unified Framework for Verification and Complexity Analysis of Real-Time and Distributed Systems*; (CCR-9225124 A003); $113,679; 12 months. (Joint support with the Software Engineering Program)

The goal of this project is to define a general formal model for reasoning about real-time and distributed systems, and to establish its value by using it for a variety of purposes. Many models have been proposed for such systems; this project is unique in that it attempts to define a single model that can be used for many purposes. The uses for such a model include: (1) Proving fundamental results about the capabilities of real-time and distributed systems. These include upper and lower bound complexity results (for time, space and communication complexity) and impossibility results; (2) Describing real systems, using appropriate programming languages; (3) Specifying allowable system behavior; (4) Verifying (by hand or by machine) the correctness of algorithms and systems. There are different styles of verification, of which assertional, algebraic and temporal logic methods are probably the most significant; (5) Analyzing system performance. Traditionally, research in each of these areas has been based on a semantic model designed specifically for that area. In contrast, this project aims to conduct work in these areas in terms of a common model. The project focuses particularly on real-time and other timing-based systems (e.g., communication systems and real-time process control systems).

University of Texas – Austin; Jayadev Misra; *Modular Multiprogramming*; (CCR-9504190 A000); $225,413; 36 months. (Joint support with the Software Engineering Program)

Two major goals in multiprogramming research are: (1) to design and understand the modules (e.g., processes or data objects) of a program in isolation, without considerations of interference by the other modules, and (2) to implement the modules on separate processors with a fine grain of interleaving so that no processor is ever locked out of accessing common data for long periods of time. Further, it is essential to devise a model in which the distinction between computation and communication is removed; in particular, the methods for designing and reasoning about the interfaces should be no different from those employed for the computations at the nodes of the network. The project is based on the hypothesis that many of the programming concepts developed for databases, object-oriented programming and designs of reactive systems can be unified into a compact model of distributed programs that meets the desired goals. In the past, much effort has gone into limiting interference among the modules by employing a variety of synchronization mechanisms: locks or semaphores, critical regions, monitors and communication. In this project, no specific communication or synchronization mechanism, except procedure call, is built into the model. Yet, the traditional schemes for communication using bounded or unbounded channels, semaphores, and accesses to shared memory can be encoded as modules. This project uses two distinct kinds of procedures, to model sequential and concurrent aspects of programming. The programming model allows for disciplined interactions between these two types of procedures. The research will: (1) develop the model, in particular for dynamic creations of modules (2) implement the model on top of an existing language, and (3) develop the appropriate formalism to specify and verify the properties of our programs.

University of California – Santa Barbara; Ambuj Singh; *Interprocess Communication and Synchronization in Parallel and Distributed Systems*; (CCR-9223094 A003); $83,622; 12 months.

Communication and synchronization among processes is of fundamental importance in parallel and distributed systems. Together, they have an impact on the efficiency, the design, and the understanding of concurrent programs. This research aims at examining shared variable and message passing systems with respect to mechanisms for interprocess communication and synchronization. In the case of shared variable systems, the question of cache coherence and the appropriateness of non-atomic shared memory is considered. Proof techniques and programming methodologies are developed for dealing with non-atomic memories. In the case of message passing systems, the question of locality of computations is investigated. Complexity measures that measure locality and efficient algorithms for commonly occurring synchronization problems are developed. The impact of failures and dynamic networks on algorithms and tight lower bounds on various performance measures are explored. Finally, the research examines mixed programming paradigms that support both shared memory and message passing. As a part of this research, efficient simulations of shared memory on an underlying message passing architecture are designed.

University of California – Santa Barbara; Ambuj Singh and Divyakant Agrawal; *Distributed Shared Memories: Specification, Programming and Implementation*; (CCR-9505807 A000); $211,613; 36 months.

The project will focus on the specification, programming, and implementation of different kinds of distributed shared memories. Proof techniques and programming methodologies will be developed for dealing with the complexities in these definitions. Conditions will be developed under which programming with these weak memories has the same net effect as programming in the more familiar sequential domain. The developed methodologies will support both a posteriori as well as a priori approaches towards program design and verification. A testbed for distributed shared memories will be implemented. This testbed will function in a heterogeneous environment of a parallel machine and a high- performance local area network, and will be used for experimenting with protocols and applications from the scientific, database, and digital libraries domain. The testbed will also support distributed simulation of shared memories and architectures.

Kansas State University; Gurdip Singh; *Distributed Composite Protocols*; (CCR-9211621 A002); $15,640; 0 months.

This research is concerned with techniques for designing and verifying distributed protocols. It develops a framework for parallel composition of protocols, applying it to designing adaptive protocols and building a software tool supporting this methodology. It shows that parallel composition allows modular development and reuse of existing protocols and demonstrate its applicability by deriving several protocols. The research involves developing constructs for specifying interactions between the component protocols and deriving proof rules to infer properties of the composite protocol from properties of the component protocols. Results are developed about techniques to modularize distributed programs. Adaptive protocols significantly improve performance by taking advantage of the changing environment and adapting accordingly. The goal is to develop a framework for deriving adaptive protocols using parallel composition.

Kansas State University; Gurdip Singh; ***CAREER: Modular Design of Protocols*****; (CCR-9502506 A000); $101,432; 36 months.**

The difficulty in developing reliable software has been an obstacle in using distributed systems. This project will study and advance the use of formal techniques in distributed protocol design. The objective is to develop a framework which allows a complex protocol to be structured as a composition of simpler protocols. The project will develop notations for specifying the interactions between the component protocols, develop a proof system, and build a software tool supporting our methodology. The framework will be applied to various domains such as design of adaptive protocols, reconfigurable protocols, and structuring of control protocols. The educational component of the CAREER prooject will involve students in the protocol development. This will include emphasis of formal techniques in courses, increasing laboratory component to provide experience and forming alliances with the industry to re-train computer professionals (via television courses), and involving undergraduates in research.