02 INFORMATION ABOUT PRINCIPAL INVESTIGATORS/PROJECT DIRECTORS(PI/PD) and co-PRINCIPAL INVESTIGATORS/co-PROJECT DIRECTORS

Submit only ONE copy of this form **for each PI/PD and co-PI/PD** identified on the proposal. The form(s) should be attached to the original proposal as specified in GPG Section II.C.a. Submission of this information is voluntary and is not a precondition of award. This information will not be disclosed to external peer reviewers. ***DO NOT INCLUDE THIS FORM WITH ANY OF THE OTHER COPIES OF YOUR PROPOSAL AS THIS MAY COMPROMISE THE CONFIDENTIALITY OF THE INFORMATION.***

PI/PD Name: Amer S Diwan

Gender: ☒ Male ☐ Female

Ethnicity: (Choose one response) ☐ Hispanic or Latino ☒ Not Hispanic or Latino

Race:
(Select one or more)

☐ American Indian or Alaska Native
☒ Asian
☐ Black or African American
☐ Native Hawaiian or Other Pacific Islander
☐ White

Disability Status:
(Select one or more)

☐ Hearing Impairment
☐ Visual Impairment
☐ Mobility/Orthopedic Impairment
☐ Other
☒ None

Citizenship: (Choose one) ☒ U.S. Citizen ☐ Permanent Resident ☐ Other non-U.S. Citizen

Check here if you do not wish to provide any or all of the above information (excluding PI/PD name): ☒

REQUIRED: Check here if you are currently serving (or have previously served) as a PI, co-PI or PD on any federally funded project ☒

Ethnicity Definition:
Hispanic or Latino. A person of Mexican, Puerto Rican, Cuban, South or Central American, or other Spanish culture or origin, regardless of race.
Race Definitions:
American Indian or Alaska Native. A person having origins in any of the original peoples of North and South America (including Central America), and who maintains tribal affiliation or community attachment.
Asian. A person having origins in any of the original peoples of the Far East, Southeast Asia, or the Indian subcontinent including, for example, Cambodia, China, India, Japan, Korea, Malaysia, Pakistan, the Philippine Islands, Thailand, and Vietnam.
Black or African American. A person having origins in any of the black racial groups of Africa.
Native Hawaiian or Other Pacific Islander. A person having origins in any of the original peoples of Hawaii, Guam, Samoa, or other Pacific Islands.
White. A person having origins in any of the original peoples of Europe, the Middle East, or North Africa.

WHY THIS INFORMATION IS BEING REQUESTED:

The Federal Government has a continuing commitment to monitor the operation of its review and award processes to identify and address any inequities based on gender, race, ethnicity, or disability of its proposed PIs/PDs. To gather information needed for this important task, the proposer should submit a single copy of this form for each identified PI/PD with each proposal. Submission of the requested information is voluntary and will not affect the organization's eligibility for an award. However, information not submitted will seriously undermine the statistical validity, and therefore the usefulness, of information recieved from others. Any individual not wishing to submit some or all the information should check the box provided for this purpose. (The exceptions are the PI/PD name and the information about prior Federal support, the last question above.)

Collection of this information is authorized by the NSF Act of 1950, as amended, 42 U.S.C. 1861, et seq. Demographic data allows NSF to gauge whether our programs and other opportunities in science and technology are fairly reaching and benefiting everyone regardless of demographic category; to ensure that those in under-represented groups have the same knowledge of and access to programs and other research and educational oppurtunities; and to assess involvement of international investigators in work supported by NSF. The information may be disclosed to government contractors, experts, volunteers and researchers to complete assigned work; and to other government agencies in order to coordinate and assess programs. The information may be added to the Reviewer file and used to select potential candidates to serve as peer reviewers or advisory committee members. See Systems of Records, NSF-50, "Principal Investigator/Proposal File and Associated Records", 63 Federal Register 267 (January 5, 1998), and NSF-51, "Reviewer/Proposal File and Associated Records", 63 Federal Register 268 (January 5, 1998).

02 INFORMATION ABOUT PRINCIPAL INVESTIGATORS/PROJECT DIRECTORS(PI/PD) and co-PRINCIPAL INVESTIGATORS/co-PROJECT DIRECTORS

Submit only ONE copy of this form **for each PI/PD and co-PI/PD** identified on the proposal. The form(s) should be attached to the original proposal as specified in GPG Section II.C.a. Submission of this information is voluntary and is not a precondition of award. This information will not be disclosed to external peer reviewers. ***DO NOT INCLUDE THIS FORM WITH ANY OF THE OTHER COPIES OF YOUR PROPOSAL AS THIS MAY COMPROMISE THE CONFIDENTIALITY OF THE INFORMATION.***

PI/PD Name: Elizabeth Bradley

Gender: ☐ Male ☒ Female

Ethnicity: (Choose one response) ☐ Hispanic or Latino ☒ Not Hispanic or Latino

Race:
(Select one or more)
☐ American Indian or Alaska Native
☐ Asian
☐ Black or African American
☐ Native Hawaiian or Other Pacific Islander
☒ White

Disability Status:
(Select one or more)
☐ Hearing Impairment
☐ Visual Impairment
☐ Mobility/Orthopedic Impairment
☐ Other
☒ None

Citizenship: (Choose one) ☒ U.S. Citizen ☐ Permanent Resident ☐ Other non-U.S. Citizen

Check here if you do not wish to provide any or all of the above information (excluding PI/PD name): ☒

REQUIRED: Check here if you are currently serving (or have previously served) as a PI, co-PI or PD on any federally funded project ☒

Ethnicity Definition:
Hispanic or Latino. A person of Mexican, Puerto Rican, Cuban, South or Central American, or other Spanish culture or origin, regardless of race.
Race Definitions:
American Indian or Alaska Native. A person having origins in any of the original peoples of North and South America (including Central America), and who maintains tribal affiliation or community attachment.
Asian. A person having origins in any of the original peoples of the Far East, Southeast Asia, or the Indian subcontinent including, for example, Cambodia, China, India, Japan, Korea, Malaysia, Pakistan, the Philippine Islands, Thailand, and Vietnam.
Black or African American. A person having origins in any of the black racial groups of Africa.
Native Hawaiian or Other Pacific Islander. A person having origins in any of the original peoples of Hawaii, Guam, Samoa, or other Pacific Islands.
White. A person having origins in any of the original peoples of Europe, the Middle East, or North Africa.

WHY THIS INFORMATION IS BEING REQUESTED:

The Federal Government has a continuing commitment to monitor the operation of its review and award processes to identify and address any inequities based on gender, race, ethnicity, or disability of its proposed PIs/PDs. To gather information needed for this important task, the proposer should submit a single copy of this form for each identified PI/PD with each proposal. Submission of the requested information is voluntary and will not affect the organization's eligibility for an award. However, information not submitted will seriously undermine the statistical validity, and therefore the usefulness, of information recieved from others. Any individual not wishing to submit some or all the information should check the box provided for this purpose. (The exceptions are the PI/PD name and the information about prior Federal support, the last question above.)

Collection of this information is authorized by the NSF Act of 1950, as amended, 42 U.S.C. 1861, et seq. Demographic data allows NSF to gauge whether our programs and other opportunities in science and technology are fairly reaching and benefiting everyone regardless of demographic category; to ensure that those in under-represented groups have the same knowledge of and access to programs and other research and educational oppurtunities; and to assess involvement of international investigators in work supported by NSF. The information may be disclosed to government contractors, experts, volunteers and researchers to complete assigned work; and to other government agencies in order to coordinate and assess programs. The information may be added to the Reviewer file and used to select potential candidates to serve as peer reviewers or advisory committee members. See Systems of Records, NSF-50, "Principal Investigator/Proposal File and Associated Records", 63 Federal Register 267 (January 5, 1998), and NSF-51, "Reviewer/Proposal File and Associated Records", 63 Federal Register 268 (January 5, 1998).

List of Suggested Reviewers or Reviewers Not To Include (optional)

SUGGESTED REVIEWERS:

Nonlinear Dynamics: Eric Kostelich (Arizona State; kostelich@asu.edu), Alfred Hubler (Illinois; a-hubler@uiuc.edu), Louis Pecora (NRL; pecora@nrl.navy.mil).

REVIEWERS NOT TO INCLUDE:

COVER SHEET FOR PROPOSAL TO THE NATIONAL SCIENCE FOUNDATION

PROGRAM ANNOUNCEMENT/SOLICITATION NO./CLOSING DATE/if not in response to a program announcement/solicitation enter NSF 04-23	FOR NSF USE ONLY
NSF 07-504 01/17/07	NSF PROPOSAL NUMBER
FOR CONSIDERATION BY NSF ORGANIZATION UNIT(S) (Indicate the most specific unit known, i.e. program, division, etc.) CNS - COMPUTER SYSTEMS	0720692

DATE RECEIVED	NUMBER OF COPIES	DIVISION ASSIGNED	FUND CODE	DUNS# (Data Universal Numbering System)	FILE LOCATION
01/17/2007	2	05050000 CNS	7354	007431505	10/06/2010 3:54pm S

EMPLOYER IDENTIFICATION NUMBER (EIN) OR TAXPAYER IDENTIFICATION NUMBER (TIN)	SHOW PREVIOUS AWARD NO. IF THIS IS	IS THIS PROPOSAL BEING SUBMITTED TO ANOTHER FEDERAL AGENCY?
846000555	☐ A RENEWAL ☐ AN ACCOMPLISHMENT-BASED RENEWAL	YES ☐ NO ☒ IF YES, LIST ACRONYM(S)

NAME OF ORGANIZATION TO WHICH AWARD SHOULD BE MADE	ADDRESS OF AWARDEE ORGANIZATION, INCLUDING 9 DIGIT ZIP CODE
University of Colorado at Boulder	3100 Marine Street, Room 481 572 UCB Boulder, CO 80309-0572
AWARDEE ORGANIZATION CODE (IF KNOWN) 0013706000	
NAME OF PERFORMING ORGANIZATION, IF DIFFERENT FROM ABOVE	ADDRESS OF PERFORMING ORGANIZATION, IF DIFFERENT, INCLUDING 9 DIGIT ZIP CODE
PERFORMING ORGANIZATION CODE (IF KNOWN)	

IS AWARDEE ORGANIZATION (Check All That Apply) (See GPG II.C For Definitions)
☐ SMALL BUSINESS ☐ MINORITY BUSINESS
☐ FOR-PROFIT ORGANIZATION ☐ WOMAN-OWNED BUSINESS
☐ IF THIS IS A PRELIMINARY PROPOSAL THEN CHECK HERE

TITLE OF PROPOSED PROJECT **CSR---SMA: Validating Architectural Simulators Using Non-Linear Dynamics Techniques**

REQUESTED AMOUNT	PROPOSED DURATION (1-60 MONTHS)	REQUESTED STARTING DATE	SHOW RELATED PRELIMINARY PROPOSAL NO. IF APPLICABLE
$ 577,730	36 months	09/01/07	

CHECK APPROPRIATE BOX(ES) IF THIS PROPOSAL INCLUDES ANY OF THE ITEMS LISTED BELOW
☐ BEGINNING INVESTIGATOR (GPG I.A)
☐ DISCLOSURE OF LOBBYING ACTIVITIES (GPG II.C)
☐ PROPRIETARY & PRIVILEGED INFORMATION (GPG I.B, II.C.1.d)
☐ HISTORIC PLACES (GPG II.C.2.j)
☐ SMALL GRANT FOR EXPLOR. RESEARCH (SGER) (GPG II.D.1)
☐ VERTEBRATE ANIMALS (GPG II.D.5) IACUC App. Date ________
☐ HUMAN SUBJECTS (GPG II.D.6)
Exemption Subsection ________ or IRB App. Date ________
☐ INTERNATIONAL COOPERATIVE ACTIVITIES: COUNTRY/COUNTRIES INVOLVED (GPG II.C.2.j)
____
☐ HIGH RESOLUTION GRAPHICS/OTHER GRAPHICS WHERE EXACT COLOR REPRESENTATION IS REQUIRED FOR PROPER INTERPRETATION (GPG I.G.1)

PI/PD DEPARTMENT	PI/PD POSTAL ADDRESS
Computer Science	430 UCB University of Colorado at Boulder Boulder, CO 803090430 United States
PI/PD FAX NUMBER 303-492-2844	

NAMES (TYPED)	High Degree	Yr of Degree	Telephone Number	Electronic Mail Address
PI/PD NAME Amer S Diwan	PhD	1996	303-492-5172	diwan@colorado.edu
CO-PI/PD Elizabeth Bradley	PhD	1992	303-492-5355	lizb@cs.colorado.edu
CO-PI/PD				
CO-PI/PD				
CO-PI/PD				

Electronic Signature

CERTIFICATION PAGE

Certification for Authorized Organizational Representative or Individual Applicant:

By signing and submitting this proposal, the individual applicant or the authorized official of the applicant institution is: (1) certifying that statements made herein are true and complete to the best of his/her knowledge; and (2) agreeing to accept the obligation to comply with NSF award terms and conditions if an award is made as a result of this application. Further, the applicant is hereby providing certifications regarding debarment and suspension, drug-free workplace, and lobbying activities (see below), as set forth in Grant Proposal Guide (GPG), NSF 04-23. Willful provision of false information in this application and its supporting documents or in reports required under an ensuing award is a criminal offense (U. S. Code, Title 18, Section 1001).

In addition, if the applicant institution employs more than fifty persons, the authorized official of the applicant institution is certifying that the institution has implemented a written and enforced conflict of interest policy that is consistent with the provisions of Grant Policy Manual Section 510; that to the best of his/her knowledge, all financial disclosures required by that conflict of interest policy have been made; and that all identified conflicts of interest will have been satisfactorily managed, reduced or eliminated prior to the institution's expenditure of any funds under the award, in accordance with the institution's conflict of interest policy. Conflicts which cannot be satisfactorily managed, reduced or eliminated must be disclosed to NSF.

Drug Free Work Place Certification

By electronically signing the NSF Proposal Cover Sheet, the Authorized Organizational Representative or Individual Applicant is providing the Drug Free Work Place Certification contained in Appendix C of the Grant Proposal Guide.

Debarment and Suspension Certification

(If answer "yes", please provide explanation.)

Is the organization or its principals presently debarred, suspended, proposed for debarment, declared ineligible, or voluntarily excluded from covered transactions by any Federal department or agency? Yes ☐ No ☒

By electronically signing the NSF Proposal Cover Sheet, the Authorized Organizational Representative or Individual Applicant is providing the Debarment and Suspension Certification contained in Appendix D of the Grant Proposal Guide.

Certification Regarding Lobbying

This certification is required for an award of a Federal contract, grant, or cooperative agreement exceeding $100,000 and for an award of a Federal loan or a commitment providing for the United States to insure or guarantee a loan exceeding $150,000.

Certification for Contracts, Grants, Loans and Cooperative Agreements

The undersigned certifies, to the best of his or her knowledge and belief, that:

(1) No federal appropriated funds have been paid or will be paid, by or on behalf of the undersigned, to any person for influencing or attempting to influence an officer or employee of any agency, a Member of Congress, an officer or employee of Congress, or an employee of a Member of Congress in connection with the awarding of any federal contract, the making of any Federal grant, the making of any Federal loan, the entering into of any cooperative agreement, and the extension, continuation, renewal, amendment, or modification of any Federal contract, grant, loan, or cooperative agreement.

(2) If any funds other than Federal appropriated funds have been paid or will be paid to any person for influencing or attempting to influence an officer or employee of any agency, a Member of Congress, an officer or employee of Congress, or an employee of a Member of Congress in connection with this Federal contract, grant, loan, or cooperative agreement, the undersigned shall complete and submit Standard Form-LLL, ''Disclosure of Lobbying Activities,'' in accordance with its instructions.

(3) The undersigned shall require that the language of this certification be included in the award documents for all subawards at all tiers including subcontracts, subgrants, and contracts under grants, loans, and cooperative agreements and that all subrecipients shall certify and disclose accordingly.

This certification is a material representation of fact upon which reliance was placed when this transaction was made or entered into. Submission of this certification is a prerequisite for making or entering into this transaction imposed by section 1352, Title 31, U.S. Code. Any person who fails to file the required certification shall be subject to a civil penalty of not less than $10,000 and not more than $100,000 for each such failure.

AUTHORIZED ORGANIZATIONAL REPRESENTATIVE	SIGNATURE	DATE
NAME Joyce W Kroll	Electronic Signature	Jan 17 2007 3:42PM

TELEPHONE NUMBER	ELECTRONIC MAIL ADDRESS	FAX NUMBER
303-735-3118	Joyce.Kroll@colorado.edu	303-492-6421

*SUBMISSION OF SOCIAL SECURITY NUMBERS IS VOLUNTARY AND WILL NOT AFFECT THE ORGANIZATION'S ELIGIBILITY FOR AN AWARD. HOWEVER, THEY ARE AN INTEGRAL PART OF THE INFORMATION SYSTEM AND ASSIST IN PROCESSING THE PROPOSAL. SSN SOLICITED UNDER NSF ACT OF 1950, AS AMENDED.

Hardware simulators are invaluable for experimenting with hardware innovations and for getting deep insight into the performance of complex applications. However, results from a simulator are useful only if that simulator accurately models the microprocessor that it is supposed to model. For example, let's suppose an architect proposes extensions to the memory system of an existing microprocessor and finds, using simulations, that the extensions improve performance. If the simulator does not correctly model the microprocessor, the improvements due to the extension may not materialize on real hardware. Even seemingly innocuous or unrelated inaccuracies—e.g., in the speculation mechanisms—can cause the simulator to differ wildly from real hardware.

Since architectural simulators are critical for architecture research, it is surprising that only a handful of published papers try to validate simulators. Worse, all prior work on validating simulators use a single-number model for validation: i.e., they compare summary statistics from a simulator to summary statistics from the hardware. Since program performance does not stay constant over the run but instead varies, sometimes wildly, using a single-number model is inadequate: a simulator and a hardware run may yield same single-number model but their time-varying behavior may be drastically different. The proposed work will design, implement, and evaluate models that capture time-varying behavior and thus support a thorough validation of simulators. Since computer systems exhibit both linear and nonlinear behavior, these models will capture both linear and nonlinear aspects of modern systems.

Specifically, this proposal addresses the following challenges:

1. How to compare the time varying behavior of the simulator and the hardware? Meaningfully comparing two time series is non-trivial since even slight perturbations (e.g., context switch on the hardware) can cause the time series to look very different.
2. What is the nature of the difference between time varying behavior of the simulator and the hardware? Some differences affect performance linearly while other differences affect performance nonlinearly.
3. How to handle chaotic behavior? This proposal presents data that demonstrates that computer systems are chaotic: a small change in the starting condition can dramatically affect the evolution of the behavior. Thus, it is unlikely that one will ever end up with perfect validation even if the simulator correctly models the hardware.
4. How to handle missing information? Hardware manufacturers do not reveal all details of their microprocessors. Thus, a validation methodology should be able to handle this missing information.

Intellectual merit. This proposal brings novel techniques from nonlinear dynamics and data mining to bear on the problem of validating simulators. This work will extend the techniques in prior work and move to a deeper understanding of these techniques by using them to validate simulators used by current researchers.

Broader impact. The quality of architectural simulators to a great extent affects the correctness and quality of much research in the architecture community. This research will improve the quality of simulators and thus directly impact much of the work done by the architecture community.

The PIs teach courses in computer architecture and nonlinear dynamics and have a significant prior track record of educational innovations. They will incorporate class projects in computer architecture courses where students undertake small validation projects; these will give students an appreciation for the complexity of modern hardware. They will incorporate class projects in nonlinear dynamics where students will use textbook techniques for comparing and modeling the time series; these will give students a deeper understanding of the relative strengths and weaknesses of different time-series analysis algorithms.

TABLE OF CONTENTS

For font size and page formatting specifications, see GPG section II.B.2.

*Proposers may select any numbering mechanism for the proposal. The entire proposal however, must be paginated. Complete both columns only if the proposal is numbered consecutively.

1 Motivation

Architects use simulators extensively to evaluate new hardware features; for example, in 2002 one-third of all papers in the top architecture conferences used SimpleScalar simulators [1]. Simulators enable architects to try out new ideas without the often-prohibitive cost of building prototype hardware.

Performance analysts often find it easier to understand the performance of a program running on a simulator than the performance of a program running on hardware. On hardware, the performance analyst needs to contend with issues such as perturbation; on a simulator it is easy to avoid perturbation.

However, any research with a simulator is valid only if the simulator itself is validated. For example, consider a simulator that assumes that all loads and stores to main memory take 350 cycles. An application with a large working set will run slowly on this simulator since it pays a penalty of 350 cycles every time it misses in the cache. In response to this, an architect may add a stride prefetcher to the simulator, which significantly speeds up the application on the simulator. However, this improvement and indeed the original slowness of the application is not real: it happens because the simulator does not model page mode writes and reads: i.e., it does not model that if a memory access is to the same DRAM page as the previous access, it is much faster than 350 cycles. If the simulator had modeled this memory feature, the program would have been fast even on the simulator without the prefetcher, thus rendering the prefetcher unnecessary.

> Unfortunately, even carefully designed and detailed simulators are often wrong [28]. The focus of our work is to come up with a procedure for determining whether or not a simulator accurately models a real machine.

Prior work has used two techniques for validating simulators. First, one can inspect or walk through the simulator code using a debugger and ensure that the simulator does what it is supposed to do [7] (*validate-inspect*). Second, one can simulate a program and also run it on real hardware, then compare the outcomes (*validate-compare*). Since *validate-compare* approach is commonly used and it is the one we improve upon, we describe it in some detail.

Figure 1 illustrates the *validate-compare* approach. All implementations of this technique run a program on a simulator and real hardware, and from each run extract a model. Diwan et al. [20] use overall cycles-per-instruction, Desikan et al. [18] use overall instructions-per-cycle, Black and Shen [7] use cycle counts, and Gibson et al. [28] use relative execution time as models. In other words, all prior work uses a *single number* (which may be a count, such as count of cycles, or a computed value, such as instructions-per-cycle) as the model. This makes comparing models easy: one simply compares the numbers from the simulation run and the hardware run and if they are the same or sufficiently close to each other then the simulation is valid.

Unfortunately, modeling an entire run with a single temporally lumped number is inadequate, for two reasons.

First, even a validated simulator may be totally inaccurate if we are interested in exploring time varying behavior since the single-number validation model does not capture that behavior. Figure 2 demonstrates this. Figure 2 (a) shows the IPC over time for a run of the `mcf` benchmark from the SPEC CPU2006 [49] on a CoreDuo workstation. Figure 2 (b) gives the IPC over time for the same benchmark on `sim-outorder` simulator in the SimpleScalar toolkit [1]. We used the same sampling interval for the simulator and for the real hardware. For the SimpleScalar run we configured the simulator to be as similar to the CoreDuo as we could manage. In order to enhance the readability of the graphs, we truncated the graphs to show only the

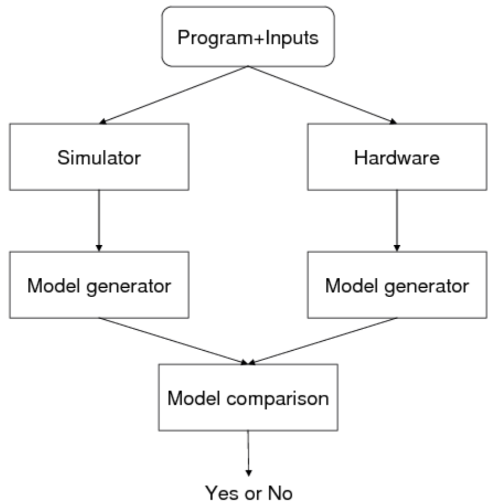


Figure 1: Process for *validate-compare*

prefix of the runs. If we consider a single-number model (as prior work on validation has done) we would end up with an IPC of 0.77 on the real hardware and 0.97 on the simulator, yielding a difference of 26% from the hardware's performance. However, the time varying behavior of the real hardware and simulator are dramatically different: for example, the real hardware exhibits an IPC of around 4 for the first 2B cycles; we do not see this behavior in the simulated run.

Our prior work shows [29, 30] that dramatic time-varying behavior is common: every Java program we looked at (which included the commonly used Java benchmarks [50, 51]) has interesting and significant time varying behavior. If a simulator does not capture this behavior then it will be inadequate for understanding the performance of these programs.

Second, Gibson et al. [28] find that validation is not a monotonic process: sometimes improving the accuracy of a simulator may actually make it appear worse with respect to the single-number model. While Gibson et al. do not provide a reason for this, one possible explanation is that different inaccuracies in a simulator may cancel each other out. In other words, even if we validate a simulator for a given program, it may actually be quite inaccurate even for that program. For these reasons, this proposal explores richer models that capture much more temporal information than the single-number models do. These models, which we describe in Sections 3 and 4, enable us to compare the time-varying behavior of applications running on hardware to the time-varying behavior on simulators, and thus is a far better way than single-number models to elucidate similarities and differences.

Modeling computer performance with linear techniques, which the single-number models do, is also inadequate. Many aspects of computer systems are nonlinear in nature: for example, it is incorrect to assume that each load that misses in the L1 cache takes the same time (i.e., the relationship between a load miss and its cost is not linear); instead, some cache misses may cost hundreds of cycles (if the needed location must

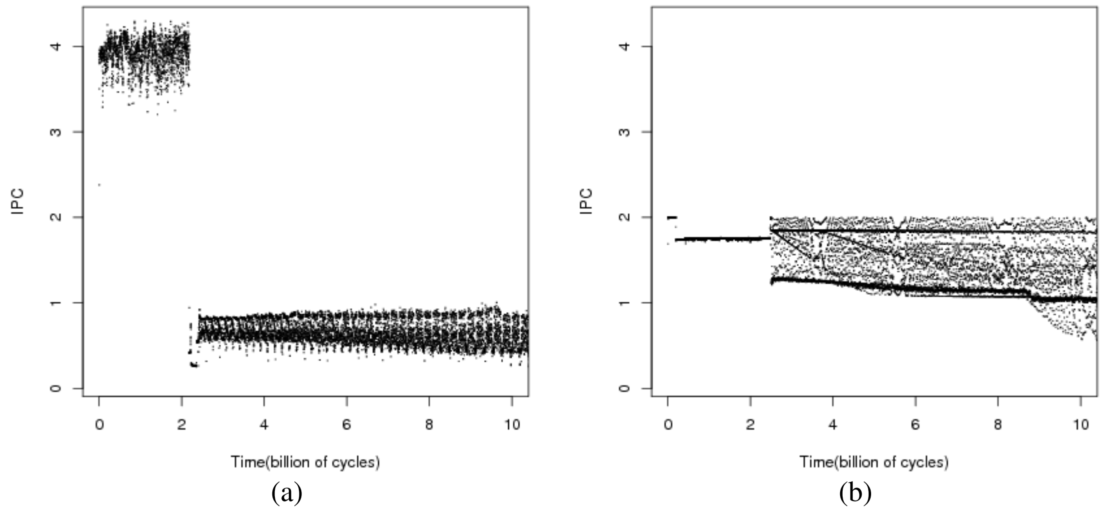


Figure 2: Time varying behavior on real machine and simulator

be fetched from main memory) while others may cost nothing (if the program does not immediately use the loaded value). Intuitively, modern hardware uses "ifs" extensively: *if the needed location is in the L1 cache and it is marked as valid then* $\cdots$ *else if* $\cdots$. This nonlinearity makes the behavior complicated and can even produce dynamical chaos [6]. Linear techniques cannot capture these effects or enable comparisons of the associated behaviors. For these reasons, the techniques covered in Section 4 draw heavily upon ideas and tools from the nonlinear dynamics literature to model and compare computer behavior.

2 Challenges

We first discuss challenges in modeling for validation and then discuss techniques that we will try.

Challenge 1: How do we produce a model that captures the time varying behavior while still enabling comparisons? As discussed earlier, using a single aggregate number as a model is inadequate; instead we need to capture the changing behavior of the simulator and hardware during the program run.

Challenge 2: How do we handle linear *and* non-linear behavior? Microprocessors exhibit significant nonlinear behavior. For example, the relationship between a load and its cost is not linear: it depends on many conditions, such as which level of the memory hierarchy can satisfy the load. Linear models cannot capture this behavior, and applying a linear analysis technique to nonlinear data can produce incorrect results. In order to capture and enable comparisons of computer behavior, our models must be able to handle nonlinear effects.

Challenge 3: How do we handle chaotic behavior? A hallmark of chaos is when two states that are close to each other diverge exponentially over time. We expect that such behavior will become increasingly common as multi-core systems become more prevalent: a minuscule perturbation of a thread running on one core can dramatically change how that thread interacts with threads on other cores. Such behavior is often confused with non-determinism, but strictly speaking it is not: if we run a program twice using *exactly* the same setup (input, memory contents, temperature, etc.) we *will* get the exact same behavior. It is just that it is nearly impossible to run an application twice under the exact same conditions: even a slight different in room temperature, for example, may cause the machine to go into a low power-dissipation mode in one run but not in another.

Challenge 4: How do we validate against something we cannot fully know? While hardware manufacturers publish many internal details of their microprocessors, there are some details that they do not expose. For example, modern Intel Pentium processors convert x86 instructions into microcode before executing them. The microcode instruction set is microprocessor specific and Intel does not reveal it. Thus, what looks like a single instruction at the x86 level may be multiple instructions at lower levels. As another example, Intel does not fully describe how the trace caches in the Pentium 4 work; without knowing this we cannot possible simulate the hardware with complete accuracy.

Broadly speaking, we investigate two kinds of approaches: approaches that use time-based representations for modeling and approaches that use a state-space representation for modeling. The following sections describe both of these approaches and discuss their strengths and weaknesses, particularly in regard to how they are able to address the above challenges.

3 Time-Based Tools

We now describe a simple time-based model that we will extract from our simulator and hardware runs, describe techniques for comparing two such models, and finally discuss the extent to which these models meet the challenges in Section 2

3.1 Modeling

Time-based tools extend a single-number model in an obvious way: they use a time series. For example, instead of using the aggregate instructions-per-cycle, time-based tools divide up an execution into intervals and record instructions-per-cycle at the end of each interval. More generally, time-based tools can record more than one value at the end of each interval and thus end up with multiple time series. For example, one time series may capture instructions-per-cycle over time and another may capture L1-load-misses-per-cycle over time.

Two key parameters for time-series modeling are the interval size and selection of metrics to be recorded. Using an overly small interval will excessively perturb the hardware run; using an overly large interval will average things out and cause our methodology to degenerate into a single-number model as used in prior work. Using hardware supported metrics (e.g., the hardware performance monitoring system) causes minimal perturbation; using software metrics (which require additional instructions to be executed) may perturb behavior, particularly in the hardware run, but may provide richer information.

3.2 Comparing Models

The difficulty with time-based models is to come up with a comparator: how do we compare two time series, one from the simulator and one from hardware? Requiring two series to be exactly the same is unrealistic and probably unrealizable: in our experience it is impossible to get two “identical” runs (i.e., same benchmark, inputs, local disk, single-user mode) on the hardware to produce identical traces. Even if we use hardware performance monitors to collect these series on the hardware, there is still some perturbation from writing the values in a buffer or a file at the end of each time interval. In addition, the OS behaves differently for different runs, for example, accessing different physical pages and thus resulting in different memory system behavior. Moreover, as discussed earlier, computer systems can exhibit chaotic behavior, where a small difference in starting conditions diverges into a dramatic difference over time. Thus, either we need to abstract away from the concrete series or use more sophisticated comparison mechanisms. The techniques in Section 4 abstract away from time series into a state-space representation. In this section we will focus primarily on comparison mechanisms.

More concretely consider the time series (A, B, C, D) from the simulator, where A, B, C, and D are distinct values. Moreover assume that the simulator is fully validated. Even though the simulator is fully validated, the time series from the hardware may look different in many ways. For example, it may look like:

- (A, B, B, C, D) if something (e.g., OS activity) causes the hardware to run slowly for a brief period, thus elongating the one "B" into "B, B".
- (A, C, B, D) if something (e.g., OS activity) changes the timing of the system and thus causes B and C to be reordered with respect to the simulation.
- (W, X, Y, Z) if a slight perturbation causes the system to exhibit chaotic behavior

We have seen many examples of the first two cases in practice; we have seen cases that look like the last one but have not yet determined whether it was caused by chaotic behavior or by something else. For every one of the above cases, standard statistical techniques (such as Spearman's or Pearson's correlation coefficient or Euclidean or Manhattan distance) will find the simulator and hardware to be very different even though in reality the simulator perfectly models the hardware. For this reason, we need a more powerful mechanism for comparing time series.

Levenshtein's distance [37] computes the edit distance between two sequences. An edit is one of insertion, deletion, or substitution. Thus, it would probably work appropriately at least for the first case.

Another technique that is specially strong at the first case is dynamic time warping (DTW) [5]. Indeed we have used this technique for aligning traces which is related to the problem we are trying to solve here [29]. Also prior work has used DTW extensively for comparing various kinds of time series such as spoken utterances and gene sequences [2,41,53].

Given two sequences, X and Y, of lengths $|X|$ and $|Y|$,

$$X = \left(x_1, x_2, \ldots, x_i, \ldots, x_{|X|}\right)$$
$$Y = \left(y_1, y_2, \ldots, y_j, \ldots, y_{|Y|}\right)$$

dynamic time warping constructs a warp path W (w_1, w_2, $\cdots$, $w_{|W|}$) such that each w_k is a pair (x_i, y_j). The warp path satisfies the following constraints:

1. For every element x_i of X, there is at least one element in W that is $(x_i, *)$, and for every element y_j of Y, there is at least one element in W that is $(*, y_j)$, i.e., no element of X or Y is omitted

2. $w_1 = (x_1, y_1)$ and $w_{|W|} = (x_{|X|}, y_{|Y|})$, i.e., the end points of the two sequences are aligned;

3. $w_k \lhd w_{k+1}$, where $w_k = (x_i, y_j)$ and $w_{k+1} = (x_m, y_n)$ if ($i = m$ or $i+1 = m$) and ($j = n$ or $j+1 = n$), but not $i = m$ and $j = n$; i.e., the warp path respects the order of both sequences and each w_k consumes at least one element from one of the sequences.

DTW uses a dynamic programming algorithm to construct a warp path. The algorithm minimizes the *DTWError*:

$$\text{DTWError} = \sum_{k=1}^{|W|} |x_i - y_j| \text{ where } w_k = (x_i, y_j) \tag{1}$$

The *DTWError* of running DTW on two time series indicates the similarity of the two time series, much like a correlation score would. However, unlike traditional correlation techniques, DTW is more forgiving: it recognizes that there might be some perturbation which may cause the hardware run to occasionally run faster and occasionally slower.

DTW, as described in the literature, has some significant shortcomings for our domain. Specifically, while DTW allows dilation (i.e., A, B, C, D can match A, B, B, C, D), it does not allow reordering of events (i.e., A, B, C, D cannot match A, C, B, D) which is a prerequisite for our needs. As part of this work, we will investigate extensions to DTW that enable it to reorder events; obviously if we give it too much flexibility it will warp every time series to every other time series. So we will need to develop a cost model that allows the reordering when it makes sense and disallows it or penalizes it otherwise. Some researchers in bioinformatics have been investigating similar extensions to DTW: we will start with their ideas and extend them as needed.

3.3 How Time-Based Tools Meet the Challenges

In this section we discuss how our proposed time-based approach addresses the challenges laid out at the beginning of Section 2.

Challenge 1. Our time-based tools naturally capture time varying behavior in the time series and we can use DTW to compare time series.

Challenge 2. The essence of dynamic time warping is linear: DTWError is a sum of linear distances. So if the simulator and hardware differ in a nonlinear fashion, most likely, DTWError will detect that as a very large difference but it does not have the ability to determine whether the hardware and simulator differ with respect in a linear or nonlinear fashion. The techniques in Section 4 capture and work with nonlinear effects, and thus we can combine the time-based and state-space based techniques to determine the nature of the difference.

Challenge 3. Chaotic behavior will cause DTWError to be excessively large. However, this is not always desirable: the simulator may actually be an accurate model of the hardware but the conditions under which we generated the time series from the simulator and the hardware may be slightly different; chaotic behavior will exponentially amplify this difference causing us to conclude that the simulator and hardware are very different.

Our state-space techniques (Section 4) specialize in chaotic behavior. However, we will extend DTW so that it can handle some kinds of chaotic behavior (e.g., our above mentioned extensions for allowing reordering of events in DTW).

Challenge 4. If there are aspects of the hardware that are hidden and the simulator differs in those aspects, then the time-based techniques discussed here will not help: they will simply show that the simulator and hardware are different but it will not provide any further insight into how they are different. The state-space techniques handle this naturally.

To summarize, the time-based techniques handle one of the challenges well (Challenge 1) and partly handle two of the others. The state-space techniques described next address many of the situations where the time-based techniques are weak. Working together, these two complimentary approaches will enable us to validate simulators and thus improve their credibility and usefulness for architectural research.

4 State-Space Tools

The state-space tools model a system as an *n* dimensional state space. Each dimension corresponds to a *state variable*. If we had a full understanding of every factor that governed a computer's behavior, we could identify these variables. If we could measure every one of those *n* state variables, x_i, we could understand the system's behavior in great depth. This is the approach usually taken by nonlinear dynamicists, and it has great power. This community plots system behavior as a series of points in the *n*-dimensional space. These *state-space trajectories* represent the behavior of the system in a compact, useful way—and one that is amenable to a host of powerful analysis tools that have been developed over the past few decades [3, 34]. These techniques handle nonlinear behavior smoothly and are very useful for comparing systems and distinguishing between them, which makes them ideal for the purposes this proposal.

Unfortunately, we cannot use such a state-space for our model: it is impractical to measure everything that governs a computer's performance. Moreover, as discussed in Section 2, there are some aspects of a computer that are hidden: we simply cannot get to them. However, we *can* collect time series of various metrics such as instructions per cycle (IPC) or L1 cache misses. These metrics are influenced by all the state variables of the system including variables that may be hard to get at (e.g., the state of the TLB at the start of the program run) or unknown (e.g., internal behavior of the Pentium 4 trace cache)[1]. This poses a conundrum: we cannot simply apply the powerful nonlinear dynamics-based comparison tools mentioned above to time series of IPC or L1 cache misses because they are not in the state-space format, but we cannot measure the right kind of data because it is impractical.

Delay-coordinate embedding is a way to solve this problem. It is a mathematical transformation that lets one reconstruct the internal dynamics of a complicated system from a single time series. In our case, this tool will allow us to transform a time series series (e.g., of IPC) into a full reconstruction of the state-space dynamics of the computer under study. Then we can use the tools from nonlinear dynamics to compare state-space dynamics in novel and powerful ways. No one has applied these kinds of techniques to real computers, and doing so is an important contribution of this proposal. Since this approach explicitly handles Challenge 4–that information may be missing–it can actually enable us make progress in validing simulators for hardware where aspects of the hardware are unknown.

[1]This is somewhat simplistic: these dimensions form a basis and thus rather than one dimension for each hardware feature there will be one for each independent combinations of features.

The following sections discuss these nonlinear dynamics-based state-space modeling and comparison techniques in more detail and present preliminary data that demonstrates that they have merit for our problem domain.

4.1 State-Space Modelling Tools

Intuitively, delay-coordinate embedding [39, 48, 52] takes a sequence of scalar values and generates an embedding from it. This embedding itself is a sequence of m-dimensional values. Each m-dimensional value is an m-tuple of values from the original sequence separated by a time delay, τ. As an example of how this works, consider the sequence of IPC values taken at regular intervals $\{i_1, ..., i_k\}$. If $\tau = 3$ and $m = 2$, we get the embedding $\{(i_1, i_4), (i_2, i_5), (i_3, i_6), \cdots, (i_{k-3}, i_k)\}$. Providing that our IPC measurements are evenly spaced in time and that we have picked reasonable values for τ and m, *the embedding theorems guarantee that the embedded IPC data is topologically equivalent to the system's true trajectory in its n-dimensional state space*. In other words, the embedding enables us to construct a state space even when we cannot measure many of the state variables.

This is a surprising result. Even though we know nothing about the internal state variables x_i, the embedding process, working on a single stream of data, produces a curve that has the same shape as the curve that the system follows in that state space. Moreover, the n state variables need not be linearly related: the embedding transformation can handle any kind of relationship. This ability to reconstruct hidden variables is the true power of embedding, and the reason for us to use it here. Specifically, this power gives us the ability to compare and validate simulators to real hardware *even if there are aspects of the real hardware that we do not or cannot know (Challenge 4)!*

The underlying theorems have a variety of conditions, of course, and the reconstruction is not completely equivalent to the internal dynamics in all situations. However, embedding can still be extremely useful because its results are guaranteed to be *topologically* (i.e., qualitatively) identical to the internal dynamics. This is an extremely powerful correspondence; it implies that we can use IPC or L1 data to effectively reconstruct the detailed internal dynamics of a modern multi-CPU computer executing a complex piece of software—and then use powerful dynamics-based tools to analyze it.

The nonlinear dynamics community uses a variety of heuristics to pick good values for τ and m. These methods are the subject of an extremely active literature; we will not cover that literature here in any detail, but simply point out the ones that we will try in this work. One common heuristic for choosing τ is the average mutual information [27]. In this method, one computes the average autocorrelation of successive points in the time series, for a range of τs, and then chooses the τ corresponding to the first minimum of that curve. The intuition here is that this lag allows new information to appear between samples. There are many other ways to use this curve (or others) to choose a τ value if mutual information proves to be inadequate for our purposes [3, 34].

Estimating the embedding dimension m is harder and more critical. Intuitively, m needs to be large enough to allow the dynamics—which have effectively been projected onto a single axis via the sampling process that measures only IPC—to "re-inflate." The false near neighbor (FNN) algorithm [35] is based on the observation that neighboring points may be projections of points that in reality are very far apart. The FNN algorithm starts with $m = 1$, finds each point's nearest neighbor, and then embeds the data with $m = 2$. If the neighboring points' separations change abruptly between the one- and two-embeddings, then the points were *false* neighbors. FNN continues adding dimensions and re-embedding until an acceptably small number of false near neighbors remain, and returns the last m-value as the estimated dimension. There are dozens of other heuristics for estimating m; again, see [3,34] for good surveys. In the worst case, successful reconstruction may require an m of up to twice the number n of internal state variables. In practice, though, one can often get a good reconstruction using a much lower m, which is an indication that the system's dynamics are governed by fewer factors. We exploit that conjecture in this proposal.

To explore the relationship between embedding dimension and system complexity, we ran a number of Java applications on two versions of sim-outoforder from the SimpleScalar toolkit [1]. We configured the first version ("Power4") to be similar to a Power4 system. We configured the second version ("simpleMem") similarly to Power4 except that it had an unrealistically optimistic model of the memory system (i.e., L1 cache, L2 cache, and memory all had an access latency of 1 cycle). The m values we got using the false near neighbor algorithm on the Power4 for a number of the SPECjvm98 benchmarks [51] were 11 (`compress` benchmark), 8 (`jess` benchmark), and >15 (`db` benchmark)[2]. In contrast, simpleMem gave m values of 7 (`compress`), 7 (`jess`), and 11 (`db`). Thus, the comparison of m appears to reflect that simpleMem is simpler than the Power4 configuration of SimpleScalar. Moreover, it is telling that the biggest difference in m values occurs for `db`: of the three benchmarks, `db` is the most memory intensive program and the L2 cache performance is the primary factor that governs its performance [30].

The embedding theorems do *not* guarantee these correspondences; the overall dynamics in the embedding space are guaranteed to be diffeomorphic to the dynamics in the true state space, but the form of the diffeomorphism is not specified, and is in general impossible to obtain. This means that the correspondence between the true state variables (e.g., parameters of computer subsystems) and the embedded variables $\{(i_1,i_4),(i_2,i_5),(i_3,i_6),\cdots,(i_{k-3},i_k)\}$ is not known. In practice, however, a large change in m often means that the dimensionality of the true dynamics changed as well, so we have some reason to believe that the one with the smaller m is simpler than the one with the larger m. Thus, it is useful as a quick check but one needs better techniques to provide richer information.

Another important caveat in the embedding theorems is that they strictly only apply if the data variable that one has measured is a state variable of the system. Neither IPC nor L1 misses are state variables; rather, they are complex nonlinear functions of multiple internal state variables. Nonetheless, our preliminary experiments (reported in section 4.2, below) suggest that we can indeed use these data to obtain successful embeddings.

Once the internal dynamics of a real or simulated computer have been reconstructed by successfully embedding the IPC or some other time series, we can use the arsenal of powerful nonlinear dynamics tools to analyze the resulting data. Our specific objective is to compare real and simulated computers; to do this, we will compare two embeddings, one from the simulation run and one from the hardware run. We will first assess the topology of the two embeddings using the Lyapunov exponent. We will also analyze the level of determinism in the two different embeddings using a technique called the recurrence plot. We cover both techniques in the following section. If the topology or the determinism levels of the two embeddings is different, we conjecture, there is reason to believe that the simulation run behaves differently from the hardware run.

4.2 State-Space Comparator Tools

Formally, the Lyapunov exponents of a dynamical system, λ_i, measure how fast two neighboring state-space trajectories diverge. These λs are the nonlinear generalizations of the real parts of the system's eigenvalues,

[2] The computation for the `db` benchmark took too long so we cut it off when it reached 15; the actual m is some value greater than 15.

and there are as many of them as there are state variables in the system. A positive λ implies exponential growth of a perturbation along the *unstable manifold*, the nonlinear generalization of the eigenvector associated with a positive-real-part eigenvalue. A positive λ is also part of the definition of dynamical chaos, where a small perturbation has a large effect on the system's future evolution. A negative λ implies exponential shrinkage of a perturbation along the *stable manifold* that is the nonlinear analog of the stable eigenvector. This represents an element of convergence and stability.

For our purposes, the important point is that these λ_i are one measure of the fundamental topology of the system's state-space behavior. This means that can use them to compare trajectories from different computer runs and identify any differences. Because the λ_i are exponential growth factors, the largest positive one obviously dominates the long-term behavior, and that is what most algorithms calculate[3]. Most of these algorithms find and follow pairs of points over time, monitoring how fast they separate and often renormalizing to avoid numerical issues. We will use the algorithm of Kantz *et al.*, a particularly stable version that uses ε-diameter clouds of points—instead of pairs—in order to smooth out noise effects [33]. This algorithm outputs the logarithm of the stretching factor that distorts the ε-cloud of points, plotted against the time interval over which the stretching was computed. If that curve increases linearly over some range, that slope is a reasonable estimate of λ.

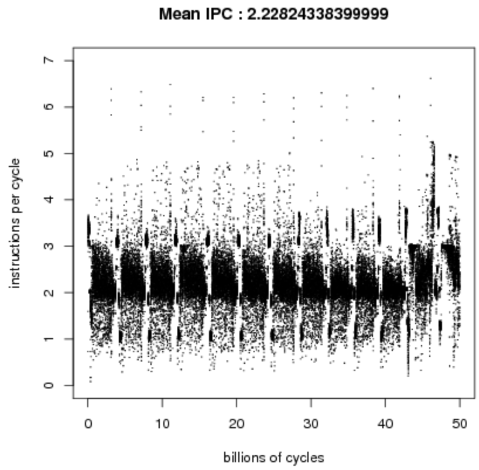
A *recurrence plot* [26] is a two-dimensional representation of a single trajectory. The trajectory to be analyzed spans both ordinate and abscissa, and each point (i, j) on the plane is shaded according to the distance between the i^{th} and j^{th} trajectory points. If one shades only points that are close to one another, this is an effective way to analyze recurrences—hence the name of the technique. Diagonal lines on such a plot are particularly meaningful, as they correspond to periodicities in the time series: different time intervals where the trajectory is tracing out exactly the same progression of states. The %determinism metric [54], which computes the percentage of recurrent points that are in these diagonal lines, is a good measure of the determinism of the trajectory—and an effective way to compare two different embeddings.

To evaluate how well these state-space tools work on computer performance data, we applied them to an IPC trace from a `gcc`-compiled `bzip2` running on a Core Duo machine with input from the SPEC2000 benchmark suite `(input.combined)`. The left-hand column of Figure 3 shows the results.

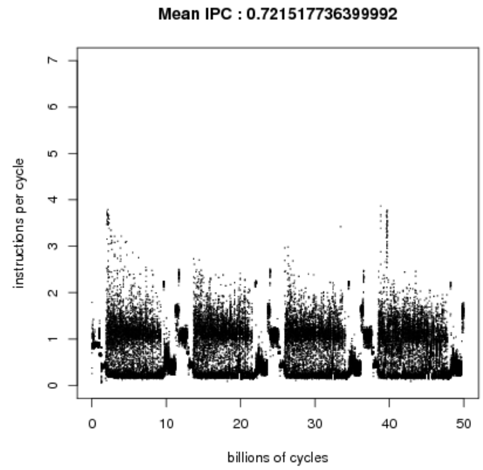
The IPC time series—part (a) of the Figure—was measured every 100,000 cycles. We first embedded this time series, following the process covered in Section 4.1, then computed the recurrence plot and the Lyapunov exponent using the algorithms described at the beginning of this section. The first minimum in the mutual-information curve for this time series occurred at a time lag of 19.4 million cycles, so we chose that as the τ for the embedding process. The false-near neighbor (FNN) algorithm returned a dimension m of 10 for this time series. We then embedded the IPC data using these τ and m values and computed the recurrence plot, which is shown in Figure 3(c). The %determinism value for this plot was 0.49, reflecting the fact that roughly half of the recurrent points in the plot (those that are within one standard deviation of the mean distance of any two points) appear in lines that parallel the main diagonal. Using the same embedded trajectory, we then computed the Lyapunov exponent λ using Kantz's algorithm, varying the window size parameter ε from 1/1000 of the data interval to 1/100 of the data interval . We also varied m as a check on FNN. The results are shown in Figure 3(e). Recall that λ is defined as the slope of the linear part (if any) of these curves. In Figure 3(e), this slope is 1.06, indicating that the behavior of the system is chaotic[4].

[3]For the rest of this proposal, we will refer to this largest λ_i as λ, or as "the" Lyapunov exponent.

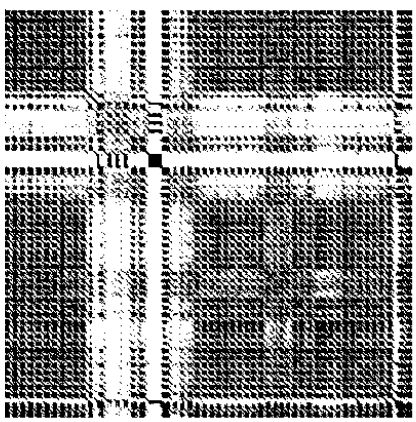
[4]There are two linear parts of these curves, but the flattening at high time interval is a saturation effect, not a property of the dynamics, so we take the slope in the region [0,2.5] million instructions using a simple linear regression fit to the data, shown as



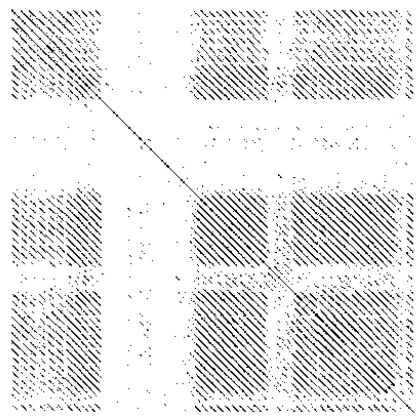
(a) IPC over time for CoreDuo



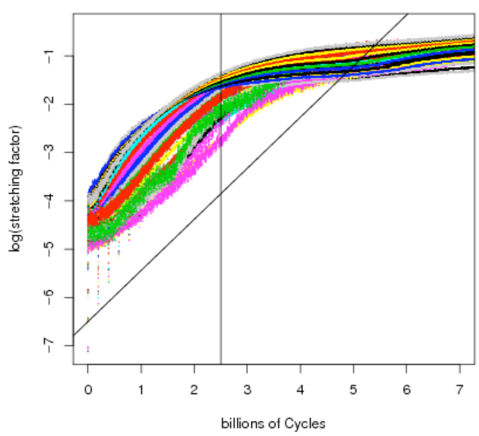
(b) IPC over time for Pentium 4



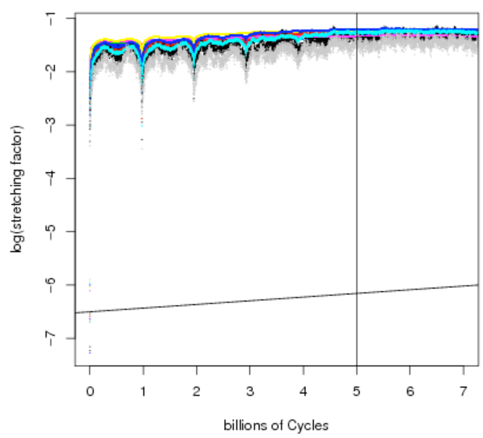
(c) Recurrence plot for CoreDuo



(d) Recurrence plot for Pentium 4



(e) λ plot for CoreDuo



(f) λ plot for Pentium 4

Figure 3: Nonlinear dynamics analysis of IPC data from `bzip2` running on two different architectures: the Core Duo (left column) and the Pentium IV (right column). Top row: time series. Middle row: recurrence plot. Bottom row: Lyapunov exponent. The different colors in the recurrence plots show different interpoint distances; in the λ calculations, different colors identify different parameters (ε and m) passed to the λ algorithm.

Note that this does not mean that the results of the computation will be unpredictable or nondeterministic, just that the time progression of IPC (and possibly the elapsed time of the run) are sensitively dependent to small perturbations in the system's internal state variables. A series of simulation experiments reported in the nonlinear dynamics literature in 2006 [6] also found evidence of chaos in `bzip2`, but the results of Figure 3 are the first observation, to our knowledge, of chaos in real computer hardware.

To see whether our nonlinear tools can distinguish one computer architecture from another, we repeated the same experiment and analysis on a Pentium 4. Comparing Figures 3(a) and (b), which show IPC vs. time for a segment of `bzip`'s run on a CoreDuo and Pentium 4 workstation respectively, we see that the benchmark exhibits periodic behavior on both workstations. If we use aggregate metrics, such as overall IPC, to validate a simulator against these runs (as prior work as done) we may end up validating a simulator that does not exhibit this periodic pattern but has similar IPC. We also see that on the CoreDuo most of the points are at the high end (yielding an average IPC of 2.2) whereas on the Pentium 4 most of the points are at the low end (yielding an average IPC of 0.7). We analyzed these runs using our state-space methods. The embedding parameters were $\tau = 97.3$ million cycles and $m = 12$. The %determinism of the Pentium 4 recurrence plot was 0.38 and the λ was 0.07—compared to 0.49 and 1.06, respectively, on the CoreDuo. Clearly, these state-space tools can pick up differences between two microprocessor architectures.

Strictly speaking, as mentioned in the previous section, one can only embed *state variable* data, and IPC is not a state variable. In order to explore whether embedding this data makes any sense at all, we repeated the same `bzip2`/Core Duo experiment, but using another metric: L1 cache misses. The results were $\tau = 2.8$ million cycles, $m = 11$, %determinism $= 0.34$, and $\lambda = 1.07$. The strong similarity to the IPC results (19.4, 10, 0.39, and 1.06 respectively) suggests that applying these nonlinear dynamics tools to computer performance data has real merit.

There are many other nonlinear dynamics tools that measure different aspects of state-space behavior and some of these may be very useful for our purposes. λ is only one way to measure topology, for instance, and a fairly coarse-grained way at that. If we find that we need a finer-grained technique, we can explore *local* λs—the same kind of stretching-factor idea, but calculated in *patches* of the state space [4]. There are other ways to measure topology besides how fast things stretch, some of which were developed by co-PI Bradley's group [42, 43, 45].

4.3 How State-Space Tools Meet the Challenges

In this section, we summarize the results above and discuss how our proposed state-space approach addresses the challenges laid out at the beginning of Section 2.

Challenge 1. Though the time-based tools of Section 3 are the source of the true power to meet this challenge, the recurrence plot captures some temporal detail that compliments those techniques very well. These state-space tools allow us to focus in quickly on time intervals where two traces are behaving identically, and the associated %determinism calculation makes comparisons very easy (e.g., 0.49 for `bzip2` on the Core Duo versus 0.34 on the Pentium 4).

Challenge 2. Nonlinearity is built into these tools from the ground up, so they handle those effects naturally. Their results bring out differences between architectures very clearly (e.g., a Lyapunov exponent of 1.06 for `bzip2` on the Core Duo versus 0.07 on the Pentium 4).

the diagonal line in the figure. In part d, we fit a similar regression over the interval [0.2, 5] million instructions.

Challenge 3. The power of these nonlinear state-space tools allowed us to identify chaotic behavior in `bzip2` on the Core Duo—a result that has been predicted from simulations, but never observed on a real machine, and one that would be impossible to make with linear tools.

Challenge 4. Delay-coordinate embedding is a way to reconstruct the internal dynamics of a complicated system from a single time series. This is a partial solution to control theory's *observer problem*: how to identify all of the internal state variables of a system and infer their values from the signals that *can* be observed. It is also an effective way to address our fourth challenge, which demands that we model a complicated system that has many hidden but important state variables. To explore this, we embedded data from various Java applications running on different SimpleScalar configurations and found that simpler configurations required a lower-dimensional reconstruction. We also constructed embeddings of IPC and L1 data from the same experiment and obtained essentially identical answers. These results suggest that embedding is an effective way to work with our target systems.

These state-space techniques handle three of this proposal's challenges (2–4) very well and partly handle the other (Challenge 1). It is worth mentioning, in light of our point about a single-number model being inadequate, that the methods in this section also produce single numbers. Those numbers, however, like DTW described in Section 3, capture far more temporal and spatial detail about the system under examination than end-to-end IPC counts over a whole run do. And we do not simply rely on a single number (λ, %determinism, ...) but rather use several of them together in order to understand the system's behavior. Working together, the time-based techniques of Section 3 and the state-space tools described in this section will constitute a novel and powerful approach for modelling, comparing, and validating computer systems.

5 Educational goals

Liz Bradley teaches a course an upper-level undergraduate course in non-linear dynamics which covers some of the techniques in Section 4. For this class, our research will provide concrete and novel applications of the techniques covered in class. By incorporating projects where students analyze time series from simulators and hardware, students will get an appreciation for the utility and power of the techniques.

Amer Diwan teaches a graduate level seminar in computer architecture and will teach the upper-level undergraduate computer organization course in the near future. These courses will benefit from projects where students try to improve architectural simulators or try to find inaccuracies in them. Such projects will give students a keen appreciation of the complexity of modern hardware and in particular the difficulty of collecting and understanding reliable and credible data from computer systems. Amer Diwan has a history of incorporating educational innovations in the classroom. For example Amer's PL-Detective system [19, 21] is a novel infrastructure for teaching programming languages and is now being used at several institutions outside of Amer's home institution.

6 Results from prior NSF support

6.1 Diwan

Diwan's NSF grant titled "Understanding the Performance of Modern Systems" (CNS-0509521: $400K from 09/01/05 to 08/30/09) is closest to this proposal. The goal of that proposal is to develop statistical, machine learning, and visualization techniques to help understand the performance of modern machines.

This grant is in its early stages. So far we have accomplished the following:

- Two workshop papers (First Workshop on Software Tools for Multi-Core Systems, 2006, and Ninth Workshop on Computer Architecture Evaluation Using Commercial Workloads,2006) that describe the key issues in our research.
- A technique for aligning traces, which is necessary for understanding the performance of modern systems. We demonstrate that our technique performs well: using this technique performance analysts can do correlations across traces and expect to get very similar results to if they had correlated within a trace. This technique has resulted in a workshop paper (Next Generation Systems workshop, 2006) and a conference submission to International Sympsium on Computer Architecture.
- A tool-supported methodology for understanding the performance of systems running on modern machines. Our work so far focuses on single-core microprocessors. This work has resulted in a submission to the journal ACM Transactions on Computer Systems.
- An investigation into using Grainger causality to automatically infer causal relationships from trace data. This work is in its preliminary stages and we do not have any papers on it yet.

While working on this project we realized that it is much easier to understand the performance of systems running on a simulator than on a real machine. Unfortunately, when we tried the commonly-used existing simulator infrastructures (SimpleScalar and Liberty) we were disappointed that the behavior on the simulators was often very different from that on real hardware. In other words, understanding the performance of a system on a simulator was worthless since the simulators did not match what the hardware was doing (which is what we really cared about). Moreover, getting the simulator to match the real hardware is usually not as simple as tuning simulator parameters: oftentimes there are significant hardware subsystems that the the system models differently or not at all. Thus the current proposal is a direct outgrowth of the work discussed in this section.

6.2 Bradley

Bradley has had five NSF grants and three REU supplements in the past ten years. Of these, the work proposed here is most closely related to her NYI award (CCR-9357740; $287K; 1993–1998), which supported work on modelling tools [12, 13, 15, 22–25, 31], practical uses of chaos [8, 16, 17], and a variety of nonlinear dynamics techniques, including the recurrence plot [14, 32], topological methods [45–47], and other data-analysis techniques [9–11, 44].

Bradley's research group is deeply interdisciplinary and vertically integrated. She has been lead PI on two ITR grants in the past five years, one to do experimental fluid dynamics modeling and MEMS-based flow control (ACI-0083004; $497K; 2000–2003) and one to apply artificial intelligence techniques to a problem in paleoclimatology (ATM-0325812; $922K; 2003–2008). Publications from these grants have reported novel fluid dynamics experiments [40], novel MEMS designs [38], a novel non-invasive technique for the measurement of blood velocity fields [36], and a novel software framework for radioisotope dating [55]. These projects have been successful in other ways as well. The fluids lab work was conducted largely by undergraduates, three of whom moved into graduate programs at Colorado and Stanford; the postdocs on the project moved on to tenure-track positions at Gyeongsang National University (Korea) and MIT. The laboratory apparatus was also used in a groundbreaking course on the art and physics of flow visualization (`http://www.colorado.edu/MCEN/flowvis/`).

7 Conclusions

Research in computer architecture employ simulators extensively. For such research to be applicable to real hardware, it is critical for the simulators to be valid: if the simulators do not correctly model the actual hardware, then any results from the simulator may be simply invalid. Unfortunately, simulators are incredibly difficult to build because modern computers are extremely complicated systems. Their dynamic behavior is governed by an increasingly sophisticated set of architectural features, which can make the time evolution of program performance very difficult to understand and simulate correctly. Computers are also nonlinear: for example, a small change in the state of the cache can dramatically affect the cost of an instruction.

Existing efforts to validate simulators use a single aggregate number. For example, they compare the average IPC over a run for the simulator with the same for the hardware; if these average numbers are similar between the simulator and the hardware for a number of benchmarks, then the simulator is considered to be validated. Unfortunately, our experiments show this is not enough: aggregate numbers ignore time-varying behavior; indeed our data shows that just because the aggregate IPC is similar for the simulator and hardware, it does not mean that they exhibit similar time-varying behavior.

This proposal uses rich models to capture the temporal details of a computer's performance and powerful tools to analyze and compare the time-varying behavior on the hardware and the simulator. The time-based techniques of Section 3 handle changing time evolution. They excel when the differences between the simulator and the hardware are small; for example the simulator (incorrectly) charges 1 cycle for a cache hit instead of 2 cycles. The state-space-based techniques of Section 4 handle nonlinearity and hidden variables. They excel when the simulator and hardware are very different: for example, they have different dynamics. These two tool sets are complimentary.

This project will broadly impact three communities. First it will make a contribution to the nonlinear dynamics literature. The tools described in Section 4 have been applied to a wide variety of physical systems, ranging from lasers and neutron stars to EKG data and the currencies market, but never to computer hardware. Our application will give us deeper insights into the strengths and weaknesses of the tools and possibly lead to improvements to these tools. Second, it will make a contribution to time-series data analysis literature. Specifically, our work will lead us to extending dynamic-time warping which is a key technique in that area. Third, and most importantly, our work will dramatically improve the state of the art in computer simulation and thus benefit most research into computer architecture.

[1] Simplescalar llc. `http://simplescalar.com`.

[2] John Aach and George Church. Aligning gene expression time series with time warping algorithms. *Bioinformatics*, Vol. 17:495–508, June 2001.

[3] H. Abarbanel. *Analysis of Observed Chaotic Data*. Springer, 1995.

[4] H. Abarbanel, R. Brown, and M. B. Kennel. Variation of Lyapunov exponents on a strange attractor. *J. Nonl. Sci.*, 1:175, 1991.

[5] Donald J. Berndt and James Clifford. Using dynamic time warping to find patterns in time series. In *Working Notes of the Knowledge Discovery in Databases Workshop*, pages 359–370, July 1994.

[6] H. Berry, D. Perez, and O. Temam. Chaos in computer performance. *Chaos*, 16, 2006.

[7] Bryan Black and John Paul Shen. Calibration of microprocessor performance models. *j-COMPUTER*, 31(5):59–65, 1998.

[8] E. Bradley. Using chaos to broaden the capture range of a phase-locked loop. *IEEE Transactions on Circuits and Systems*, 40:808–818, 1993.

[9] E. Bradley. Analysis of time series. In *Intelligent Data Analysis: An Introduction*, pages 199–226. Springer-Verlag, 2 edition, 2000. M. Berthold and D. Hand, eds.

[10] E. Bradley, N. Collins, and W. Kegelmeyer. Feature characterization in scientific data. In *Proceedings of the Fourth International Symposium on Intelligent Data Analysis (IDA-01)*, pages 1–12, 2001.

[11] E. Bradley and M. Easley. Reasoning about sensor data for automated system identification. In *Proceedings of the Second International Symposium on Intelligent Data Analysis (IDA-97)*, pages 561–572, 1997.

[12] E. Bradley and M. Easley. Reasoning about sensor data for automated system identification. *Intelligent Data Analysis: An International Journal*, 2(2):123–138, 1998.

[13] E. Bradley, M. Easley, and R. Stolle. Reasoning about nonlinear system identification. *Artificial Intelligence*, 133:139–188, December 2001.

[14] E. Bradley and R. Mantilla. Recurrence plots and unstable periodic orbits. *Chaos*, 12:596–600, 2002.

[15] E. Bradley, A. O'Gallagher, and J. Rogers. Global solutions for nonlinear systems using qualitative reasoning. *Annals of Mathematics and Artificial Intelligence*, 23:211–228, 1998.

[16] E. Bradley and D. Straub. Using chaos to broaden the capture range of a phase-locked loop: Experimental verification. *IEEE Transactions on Circuits and Systems*, 43:914–822, 1996.

[17] E. Bradley and J. Stuart. Using chaos to generate variations on movement sequences. *Chaos*, 8:800–807, 1998.

[18] Rajagopalan Desikan, Doug Burger, and Stephen W. Keckler. Measuring experimental error in microprocessor simulation. In *SSR '01: Proceedings of the 2001 symposium on Software reusability*, pages 266–277, New York, NY, USA, 2001. ACM Press.

[19] Amer Diwan, Michele H. Jackson, William M. Waite, and Jacob Dickerson. PL-Detective: experiences and results. In *SIGCSE '05: Proceedings of the 36th SIGCSE technical symposium on Computer science education*, pages 221–225, New York, NY, USA, 2005. ACM Press.

[20] Amer Diwan, David Tarditi, and Eliot Moss. Memory system performance of programs with intensive heap allocation. *ACM Trans. Comput. Syst.*, 13(3):244–273, 1995.

[21] Amer Diwan, William M. Waite, and Michele H. Jackson. PL-Detective: a system for teaching programming language concepts. In *SIGCSE '04: Proceedings of the 35th SIGCSE technical symposium on Computer science education*, pages 80–84, New York, NY, USA, 2004. ACM Press.

[22] M. Easley and E. Bradley. Generalized physical networks for model building. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 1999.

[23] M. Easley and E. Bradley. Reasoning about input-output modeling of dynamical systems. In *Proceedings of the Third International Symposium on Intelligent Data Analysis (IDA-99)*, pages 343–355, 1999.

[24] M. Easley and E. Bradley. Meta-domains for automated system identification. In C. Dagli, editor, *Intelligent Engineering Through Artificial Neural Networks*, volume 10. ASME Press, 2000.

[25] M. Easley and E. Bradley. Information granulation in automated modeling. In W. Pedrycz, editor, *Granular Computing: An Emerging Paradigm*. Physica-Verlag, 2001.

[26] J.-P. Eckmann, S. Kamphorst, and D. Ruelle. Recurrence plots of dynamical systems. *Europhysics Letters*, 4:973–977, 1987.

[27] A. M. Fraser and H. L. Swinney. Independent coordinates for strange attractors from mutual information. *Physical Review A*, 33(2):1134–1140, 1986.

[28] Jeff Gibson, Robert Kunz, David Ofelt, Mark Horowitz, John Hennessy, and Mark Heinrich. Flash vs. (simulated) flash: closing the simulation loop. *SIGPLAN Not.*, 35(11):49–58, 2000.

[29] Matthias Hauswirth, Amer Diwan, Peter F. Sweeney, and Michael C. Mozer. Automating vertical profiling. In *OOPSLA '05: Proceedings of the 20th annual ACM SIGPLAN conference on Object oriented programming, systems, languages, and applications*, pages 281–296, New York, NY, USA, 2005. ACM Press.

[30] Matthias Hauswirth, Peter F. Sweeney, Amer Diwan, and Michael Hind. Vertical profiling: Understanding the behavior of object-oriented applications. In *Proceedings of the 19th Annual ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA)*. ACM Press, October 2004.

[31] A. Hogan, R. Stolle, and E. Bradley. Agenda control for heterogeneous reasoners. *Journal of Logic Programming and Applications*, 62:41–69, 2005.

[32] J. Iwanski and E. Bradley. Recurrence plots of experimental data: To embed or not to embed? *Chaos*, 8:861–871, 1998.

[33] H. Kantz. A robust method to estimate the maximal lyapunov exponent of a time series. *Phys. Lett. A*, 185:77, 1994.

[34] H. Kantz and T. Schreiber. *Nonlinear Time Series Analysis*. Cambridge University Press, Cambridge, 1997.

[35] M. B. Kennel, R. Brown, and H. D. I. Abarbanel. Determining minimum embedding dimension using a geometrical construction. *Physical Review A*, 45:3403–3411, 1992.

[36] H-B. Kim, J. Hertzberg, and R. Shandas. In vitro study of wall shear stress measurement using Echo-PIV technique. In *ASME Summer Bioengineering Conference*, 2003. submitted to Annals BioMed Eng.

[37] V. I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. *Sov. Phys. Dokl.*, 6:707–710, 1966.

[38] Z. Ma, E. Bradley, T. Peacock, J. Hertzberg, and Y-C. Lee. Solder-assembled large MEMS flaps for fluid mixing. *IEEE Transactions on Advanced Packaging*, 26:268–276, 2003.

[39] N. Packard, J. Crutchfield, J. Farmer, and R. Shaw. Geometry from a time series. *Physical Review Letters*, 45:712, 1980.

[40] T. Peacock, J. Hertzberg, Y.-C. Lee, and E. Bradley. Forcing a planar jet flow using MEMS. *Experiments in Fluids*, 37:22–28, 2004.

[41] Chotirat Ann Ratanamahatana and Eamonn Keogh. Everything you know about dynamic time warping is wrong. In *Third Workshop on Mining Temporal and Sequential Data*, August 2004.

[42] V. Robins. Towards computing homology from finite approximations. *Topology Proceedings*, 24:503–532, 1999.

[43] V. Robins, J. Abernethy, N. Rooney, and E. Bradley. Topology and intelligent data analysis. In *Proceedings of the Fifth International Symposium on Intelligent Data Analysis (IDA-03)*, pages 111–122, 2003.

[44] V. Robins, J. Abernethy, N. Rooney, and E. Bradley. Topology and intelligent data analysis. *Intelligent Data Analysis*, 8:505–515, 2004.

[45] V. Robins, J. Meiss, and E. Bradley. Computing connectedness: An exercise in computational topology. *Nonlinearity*, 11:913–922, 1998.

[46] V. Robins, J. Meiss, and E. Bradley. Computing connectedness: Disconnectedness and discreteness. *Physica D*, 139:276–300, 2000.

[47] V. Robins, N. Rooney, and E. Bradley. Topology-based signal separation. *Chaos*, 14:305–316, 2004.

[48] T. Sauer, J. Yorke, and M. Casdagli. Embedology. *Journal of Statistical Physics*, 65:579–616, 1991.

[49] Standard performance evaluation corporation. SPEC CPU2006 benchmarks. http://www.spec.org/cpu2006/.

[50] Standard performance evaluation corporation. SPECjbb2000 (java business benchmark). http://www.spec.org/jbb2000.

[51] Standard performance evaluation corporation. SPECjvm98 benchmarks. http://www.spec.org/jvm98.

[52] F. Takens. Detecting strange attractors in fluid turbulence. In D. Rand and L.-S. Young, editors, *Dynamical Systems and Turbulence*, pages 366–381. Springer, Berlin, 1981.

[53] Byoung-Kee Yi, H. V. Jagadish, and Christos Faloutsos. Efficient retrieval of similar time sequences under time warping. In *Proceedings of the International Conference on Data Engineering*, pages 23–27, February 1998.

[54] J. Zbilut and C. Webber. Embeddings and delays as derived from recurrence quantification analysis. *Physics Letters A*, 171:199–203, 1992.

[55] C. Zweck, M. Zreda, K. Anderson, E. Bradley, and L. Rassbach. icronus: A computational tool for cosmogenic nuclide dating. In *American Geophysical Union*, 2006.

Amer S. Diwan

Department of Computer Science
University of Colorado
Boulder, CO 80309-0430
(303) 492-5172; diwan@cs.colorado.edu

Education

Ph.D.	Computer Science, University of Massachusetts at Amherst, October 1996
	Thesis: *Understanding and improving the performance of modern programming langauges*
	Committee: Eliot Moss (chair), Kathryn McKinley, John Stankovic, Peter Lee, and Mani Krishna
M.Sc.	Computer Science, University of Massachusetts, Amherst, 1992
	Thesis: *Compiler support for garbage collection*
B.A.	Middlebury College, Middlebury, VT, 1989

Appointments

1999–	Assistant Professor, Computer Science, University of Colorado, Boulder, Colorado
1996–99	Engineering Research Associate, Computer Science, Stanford Univerisity
1989–99	Research Assistant, Computer Science, University of Massachusetts at Amherst

Five papers related to proposed work

1. Han B. Lee, Amer Diwan, and J. Eliot B. Moss, *Design, Implementation, and Evaluation of a Compilation Server*. To appear in ACM Transactions on Programming Languages and Systems.

2. Han Lee, Daniel von Dincklage, Amer Diwan, J. Eliot B. Moss, *Understanding the behavior of compiler optimizations*. To appear in Software Practice and Experience.

3. Matthias Hauswirth, Amer Diwan, Peter Sweeney, and Michael Mozer, *Automating vertical profiling*. Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA) 2005 (25 of 142 accepted)

4. Matthias Hauswirth, Peter Sweeney, Amer Diwan, Michael Hind, *Vertical Profiling: Understanding the Behavior of Object-Oriented Applications*. Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA) 2004 (27 out of 173 accepted).

5. Amer Diwan, David Tarditi, and J. Eliot B. Moss, *Memory system performance of programs with intensive heap allocation*. ACM Transactions on Computer Systems, 13(3), August 1995, pp. 244–273.

Five other papers

1. Martin Hirzel, Daniel vonDincklage, Amer Diwan, and Michael Hind, *Fast pointer analysis in the presence of dynamic class loading*. To appear in in ACM Transactions on Programming Languages and Systems.

2. Peter F. Sweeney, Matthias Hauswirth, Brendon Cahoon, Perry Cheng, Amer Diwan, David Grove, and Michael Hind, *Using hardware performance monitors to understand the behavior of Java applications*. USENIX 3rd Virtual Machine Research and Technology Symposium (VM'04) (acceptance rate unknown).

3. Martin Burtscher, Amer Diwan, and Matthias Hauswirth, *Compiler support for load-value prediction*, ACM Conference on Programming Language Design and Implementation, 2002, pp. 222–233 (28 out of 169 accepted).

4. Martin Hirzel, Amer Diwan, and Johannes Henkel, *On the Usefulness of Type and Liveness Accuracy for Garbage Collection and Leak Detection*, ACM Transactions on Programming Languages and Systems, 24(6), November 2002, pp. 593–624.

5. Amer Diwan, Kathryn McKinley, and J. Eliot B. Moss, *Using types to analyze and optimize object-oriented programs*, ACM Transactions on Programming Languages and Systems, 23(1), January 2001, pp. 30–72.

Synergistic Activities

1. Developed an eclipse plugin for supporting API evolution. Paper appeared in ICSE 2005.

2. Developed the PL-Detective, an innovative tool based on a modular-compiler, for teaching programming languages to undergraduates. Papers appeared in SIGCSE 2004 and SIGCSE 2005.

3. Developed the concept of a "Conversational Classroom" for teaching computer science. Paper appeared in SIGCSE 2004.

4. Developed a garbage collector simulator infrastructure (with student Martin Hirzel), which is now available under GPL for other researchers.

5. Developed an open-source performance visualizer (with student Matthias Hauswirth and collaborators at IBM) which receives significant use at in our research and within IBM.

Recent Collaborators

Martin Burtscher, Cornell University; **Keith Farkas**, COMPAQ Western Research Laboratory; **Dirk Grunwald**, University of Colorado; **David Heine**, Stanford University; **Michael Hind**, IBM Research; **Tony Hosking**, Purdue University; **Richard Hudson**, Intel Corporation; **Michele Jackson**, University of Colorado; **Monica Lam**, Stanford University; **Shih-Wei Liao**, Intel Corporation; **Kathryn McKinley**, University of Texas at Austin; **Eliot Moss**, University of Massachusetts; **Peter Sweeney**, IBM Research; **David Tarditi**, Microsoft Research; **William Waite**, University of Colorado.

Biographical Sketch

Elizabeth BRADLEY

a. Professional Preparation: M.I.T.; S.B. in Electrical Engineering (1983), S.M. in Computer Science (1986), Ph.D. in Electrical Engineering and Computer Science (1992).

b. Appointments:

• Fellow, Radcliffe Institute for Advanced Study, AY 2006/2007.
• Professor, University of Colorado, Department of Computer Science, May 2004 to present. Associate Professor, August 1999 to May 2004. Assistant Professor, January 1993 to August 1999. Chair, July 2003 to March 2006. Joint appointment in the Department of Electrical and Computer Engineering.
• Visiting Scholar, Harvard University, Division of Engineering and Applied Sciences, Spring 1997 and AY 1999/2000.

c. Publications: `www.cs.colorado.edu/~lizb/publications.html`

- **most closely related to this proposal:**
  - E. Bradley, "Causes and Effects of Chaos," *Computers and Graphics*, **19**:755-778 (1995)
  - E. Bradley, "Time-Series Analysis," in M. Berthold and D. Hand, editors, *Intelligent Data Analysis: An Introduction*, Springer Verlag, 2000; second edition, 2003 *[27 pages]*
  - E. Bradley, M. Easley, and R. Stolle, "Reasoning About Nonlinear System Identification," *Artificial Intelligence* **133**:139-188 (2001)
  - E. Bradley and R. Mantilla, "Recurrence Plots and Unstable Periodic Orbits," *Chaos* **12**:596-600 (2002)
  - V. Robins, J. Abernethy, N. Rooney, and E. Bradley, "Topology and Intelligent Data Analysis," *Intelligent Data Analysis* **8**:505-515 (2004)
- **other significant:**
  - E. Bradley and M. Easley, "Reasoning About Sensor Data for Automated System Identification," *Intelligent Data Analysis* **2**:123-138 (1998)
  - E. Bradley and D. Straub, "Using Chaos to Improve the Capture Range of a Phase-Locked Loop: Experimental Verification," *IEEE Transactions on Circuits and Systems*, **43**:914-922 (1996)
  - V. Robins, N. Rooney, and E. Bradley, "Topology-Based Signal Separation," *Chaos* **14**:305-316 (2004)
  - V. Robins, J. Meiss, and E. Bradley, "Computing Connectedness: Disconnectedness and Discreteness," *Physica D* **139**:276-300 (2000)
  - V. Robins, J. Meiss, and E. Bradley, "Computing Connectedness: an Exercise in Computational Topology," *Nonlinearity*, **11**:913-922 (1998)

d. Synergistic Activities:

- I serve on the external faculty and editorial board of the Santa Fe Institute, and I teach at their summer school every year to a broad audience from across the country and across all disciplines of science, engineering, and the social sciences. I also teach interdisciplinary courses at Colorado, I arrange to have them cross-listed in other departments, and I actively recruit students for these courses from all over the university.

- I collaborate with students and faculty in eight departments, ranging from Astrophysics to Dance. My students and I present the resulting work at conferences that cross the traditional discipline boundaries (e.g., the *International Dance and Technology* workshop, where we presented a paper on the mathematics of movement sequences).

- I review articles and programs for a broad selection of journals, conferences, and organizations: the *International Journal of Parallel Programming*, *IEEE Control Systems Magazine*, *Chaos*, *IEEE Transactions on Automatic Control*, *Physica D*, the *IEEE American Control Conference*, *IEEE Transactions on Circuits and Systems*, *Computers and Electrical Engineering*, *Physical Review*, *Physical Review Letters*, *Physics Letters A*, the NSF, the USGS, the DoD (panelist for the the IS&T TARA), and the International (*IJCAI*) and National (*AAAI*) conferences on Artificial Intelligence. I am an editor of *Chaos*, the American Institute of Physics's interdisciplinary journal of nonlinear science, and I serve on the steering committee of the *Intelligent Data Analysis* organization, which oversees a biannual conference and a semiannual journal.

- I am consistently involved in research fairs, professional panels, industrial liason activities, and the planning and execution of other events designed to encourage high-school, women, and minority students in their pursuit of science and engineering. The group of graduate research students with whom I am currently working includes two men and four women.

- As department chair, two of my primary goals were to revamp the undergraduate teaching laboratories and to institute a junior-faculty mentoring program. Both projects were judged successful by our external review committee in the spring of 2006.

- In 1999, I received the yearly student-voted College of Engineering "Innovation in Teaching" Award, largely for my work in coupling research and undergraduate teaching.

e. Collaborators & Other Affiliations:

- **collaborators and co-editors:** Ken Anderson, Computer Science, UColorado; Michael Berthold, ICSI, Berkeley; David Campbell, Boston University; Jerome Dorignac, Boston University. David Hand, Open Univ (UK); Jean Hertzberg, Mechanical Engineering, UColorado; Alain Karma, Northeastern University; Y-C Lee, Mechanical Engineering, UColorado; Marek Zreda, Hydrology, UArizona.

- **graduate and postdoctoral advisors:** H. Abelson, MIT; G. J. Sussman, MIT.

- **thesis advisor and postgraduate-scholar sponsor:**

  – *undergraduate research/thesis students:* Patrick Clary (Sun), Matt Culbreth (Ph.D. student, Stanford), Mark Eret, Apollo Hogan (Ph.D. student, Berkeley), Asim Khwaja (Professor, Kennedy School of Govt, Harvard), Sebastian Kuzminsky (UColorado BioServe Center), Evan Sheehan (UColorado PRA), Josh Stuart (Professor, UC Santa Cruz).

  – *graduate students:* Jenny Abernethy, Stephanie Boyles (local industry), Nancy Collins (NCAR), Jon Dixon (Lincoln Labs), Matt Easley (Rockwell), James Garnett (Secure64), John Giardino (Tennesse Valley Infrastructure Group) Kyle Gilbertson, Joe Iwanski (Dwight Englewood School), Zhichun Ma, Damon McCoy, Todd Mytkowicz, Serguei Ovtchinnikov, Laura Rassbach, Vanessa Robins (Postdoc, Australian Natl U.), Natalie Ross, Andrey Smirnov (Sun), Matt Schwieterman (local industry), Reinhard Stolle (BMW Research), Doug Straub (local industry), Elizabeth White.

  – *postdoctoral associates:* Nic Brummell, Hyoung-Bum Kim (Professor, Gyeongsang National University), Brian LaMacchia (Microsoft), Markus Sarkis (Professor, Old Dominion U.), Tom Peacock (Professor, MIT).

People whose affiliations are unspecified are still at CU.

SUMMARY PROPOSAL BUDGET

YEAR 1

ORGANIZATION **University of Colorado at Boulder**			FOR NSF USE ONLY PROPOSAL NO.	DURATION (months) Proposed	Granted
PRINCIPAL INVESTIGATOR / PROJECT DIRECTOR **Amer S Diwan**			AWARD NO.		

A. SENIOR PERSONNEL: PI/PD, Co-PI's, Faculty and Other Senior Associates (List each separately with title, A.7. show number in brackets)	NSF Funded Person-months CAL	ACAD	SUMR	Funds Requested By proposer	Funds granted by NSF (if different)
1. Amer S Diwan - PI	0.00	0.00	1.00	$ 11,139	$
2. Elizabeth Bradley - Co-PI	0.00	0.00	1.00	13,951	
3.					
4.					
5.					
6. (0) OTHERS (LIST INDIVIDUALLY ON BUDGET JUSTIFICATION PAGE)	0.00	0.00	0.00	0	
7. (2) TOTAL SENIOR PERSONNEL (1 - 6)	0.00	0.00	2.00	25,090	
B. OTHER PERSONNEL (SHOW NUMBERS IN BRACKETS)					
1. (0) POST DOCTORAL ASSOCIATES	0.00	0.00	0.00	0	
2. (0) OTHER PROFESSIONALS (TECHNICIAN, PROGRAMMER, ETC.)	0.00	0.00	0.00	0	
3. (2) GRADUATE STUDENTS				60,255	
4. (0) UNDERGRADUATE STUDENTS				0	
5. (0) SECRETARIAL - CLERICAL (IF CHARGED DIRECTLY)				0	
6. (0) OTHER				0	
TOTAL SALARIES AND WAGES (A + B)				85,345	
C. FRINGE BENEFITS (IF CHARGED AS DIRECT COSTS)				10,331	
TOTAL SALARIES, WAGES AND FRINGE BENEFITS (A + B + C)				95,676	
D. EQUIPMENT (LIST ITEM AND DOLLAR AMOUNT FOR EACH ITEM EXCEEDING $5,000.)					
TOTAL EQUIPMENT				0	
E. TRAVEL 1. DOMESTIC (INCL. CANADA, MEXICO AND U.S. POSSESSIONS)				5,800	
2. FOREIGN				0	
F. PARTICIPANT SUPPORT COSTS 1. STIPENDS $ 0 2. TRAVEL 0 3. SUBSISTENCE 0 4. OTHER 0					
TOTAL NUMBER OF PARTICIPANTS (0) TOTAL PARTICIPANT COSTS				0	
G. OTHER DIRECT COSTS					
1. MATERIALS AND SUPPLIES				100	
2. PUBLICATION COSTS/DOCUMENTATION/DISSEMINATION				500	
3. CONSULTANT SERVICES				0	
4. COMPUTER SERVICES				9,420	
5. SUBAWARDS				0	
6. OTHER				17,222	
TOTAL OTHER DIRECT COSTS				27,242	
H. TOTAL DIRECT COSTS (A THROUGH G)				128,718	
I. INDIRECT COSTS (F&A)(SPECIFY RATE AND BASE) 51.5% MTDC (Rate: 51.5000, Base: 111496)					
TOTAL INDIRECT COSTS (F&A)				57,420	
J. TOTAL DIRECT AND INDIRECT COSTS (H + I)				186,138	
K. RESIDUAL FUNDS (IF FOR FURTHER SUPPORT OF CURRENT PROJECTS SEE GPG II.C.6.j.)				0	
L. AMOUNT OF THIS REQUEST (J) OR (J MINUS K)				$ 186,138	$

M. COST SHARING PROPOSED LEVEL $ 0	AGREED LEVEL IF DIFFERENT $		
PI/PD NAME **Amer S Diwan**	FOR NSF USE ONLY INDIRECT COST RATE VERIFICATION		
ORG. REP. NAME* **Joyce Kroll**	Date Checked	Date Of Rate Sheet	Initials - ORG

 *ELECTRONIC SIGNATURES REQUIRED FOR REVISED BUDGET

SUMMARY PROPOSAL BUDGET   YEAR 2

ORGANIZATION **University of Colorado at Boulder**				FOR NSF USE ONLY PROPOSAL NO.	DURATION (months) Proposed	Granted
PRINCIPAL INVESTIGATOR / PROJECT DIRECTOR **Amer S Diwan**				AWARD NO.		

A. SENIOR PERSONNEL: PI/PD, Co-PI's, Faculty and Other Senior Associates (List each separately with title, A.7. show number in brackets)	NSF Funded Person-months CAL	ACAD	SUMR	Funds Requested By proposer	Funds granted by NSF (if different)
1. Amer S Diwan - PI	0.00	0.00	1.00	$ 11,518	$
2. Elizabeth Bradley - Co-PI	0.00	0.00	1.00	14,425	
3.					
4.					
5.					
6. (0) OTHERS (LIST INDIVIDUALLY ON BUDGET JUSTIFICATION PAGE)	0.00	0.00	0.00	0	
7. (2) TOTAL SENIOR PERSONNEL (1 - 6)	0.00	0.00	2.00	25,943	
B. OTHER PERSONNEL (SHOW NUMBERS IN BRACKETS)					
1. (0) POST DOCTORAL ASSOCIATES	0.00	0.00	0.00	0	
2. (0) OTHER PROFESSIONALS (TECHNICIAN, PROGRAMMER, ETC.)	0.00	0.00	0.00	0	
3. (2) GRADUATE STUDENTS				62,304	
4. (0) UNDERGRADUATE STUDENTS				0	
5. (0) SECRETARIAL - CLERICAL (IF CHARGED DIRECTLY)				0	
6. (0) OTHER				0	
TOTAL SALARIES AND WAGES (A + B)				88,247	
C. FRINGE BENEFITS (IF CHARGED AS DIRECT COSTS)				10,680	
TOTAL SALARIES, WAGES AND FRINGE BENEFITS (A + B + C)				98,927	
D. EQUIPMENT (LIST ITEM AND DOLLAR AMOUNT FOR EACH ITEM EXCEEDING $5,000.)					
TOTAL EQUIPMENT				0	
E. TRAVEL 1. DOMESTIC (INCL. CANADA, MEXICO AND U.S. POSSESSIONS)				5,800	
2. FOREIGN				0	
F. PARTICIPANT SUPPORT COSTS 1. STIPENDS $ 0 2. TRAVEL 0 3. SUBSISTENCE 0 4. OTHER 0					
TOTAL NUMBER OF PARTICIPANTS (0) TOTAL PARTICIPANT COSTS				0	
G. OTHER DIRECT COSTS					
1. MATERIALS AND SUPPLIES				100	
2. PUBLICATION COSTS/DOCUMENTATION/DISSEMINATION				500	
3. CONSULTANT SERVICES				0	
4. COMPUTER SERVICES				9,506	
5. SUBAWARDS				0	
6. OTHER				18,514	
TOTAL OTHER DIRECT COSTS				28,620	
H. TOTAL DIRECT COSTS (A THROUGH G)				133,347	
I. INDIRECT COSTS (F&A)(SPECIFY RATE AND BASE) **51.5% MTDC (Rate: 51.5000, Base: 114833)**					
TOTAL INDIRECT COSTS (F&A)				59,139	
J. TOTAL DIRECT AND INDIRECT COSTS (H + I)				192,486	
K. RESIDUAL FUNDS (IF FOR FURTHER SUPPORT OF CURRENT PROJECTS SEE GPG II.C.6.j.)				0	
L. AMOUNT OF THIS REQUEST (J) OR (J MINUS K)				$ 192,486	$
M. COST SHARING PROPOSED LEVEL $ 0	AGREED LEVEL IF DIFFERENT $				

PI/PD NAME **Amer S Diwan**	FOR NSF USE ONLY INDIRECT COST RATE VERIFICATION		
ORG. REP. NAME* **Joyce Kroll**	Date Checked	Date Of Rate Sheet	Initials - ORG

 ***ELECTRONIC SIGNATURES REQUIRED FOR REVISED BUDGET**

SUMMARY PROPOSAL BUDGET

YEAR 3

FOR NSF USE ONLY		
PROPOSAL NO.	DURATION (months)	
	Proposed	Granted
AWARD NO.		

ORGANIZATION
University of Colorado at Boulder

PRINCIPAL INVESTIGATOR / PROJECT DIRECTOR
Amer S Diwan

A. SENIOR PERSONNEL: PI/PD, Co-PI's, Faculty and Other Senior Associates (List each separately with title, A.7. show number in brackets)	NSF Funded Person-months CAL	ACAD	SUMR	Funds Requested By proposer	Funds granted by NSF (if different)
1. Amer S Diwan - PI	0.00	0.00	1.00	$ 11,910	$
2. Elizabeth Bradley - Co-PI	0.00	0.00	1.00	14,915	
3.					
4.					
5.					
6. (0) OTHERS (LIST INDIVIDUALLY ON BUDGET JUSTIFICATION PAGE)	0.00	0.00	0.00	0	
7. (2) TOTAL SENIOR PERSONNEL (1 - 6)	0.00	0.00	2.00	26,825	
B. OTHER PERSONNEL (SHOW NUMBERS IN BRACKETS)					
1. (0) POST DOCTORAL ASSOCIATES	0.00	0.00	0.00	0	
2. (0) OTHER PROFESSIONALS (TECHNICIAN, PROGRAMMER, ETC.)	0.00	0.00	0.00	0	
3. (2) GRADUATE STUDENTS				64,423	
4. (0) UNDERGRADUATE STUDENTS				0	
5. (0) SECRETARIAL - CLERICAL (IF CHARGED DIRECTLY)				0	
6. (0) OTHER				0	
TOTAL SALARIES AND WAGES (A + B)				91,248	
C. FRINGE BENEFITS (IF CHARGED AS DIRECT COSTS)				11,044	
TOTAL SALARIES, WAGES AND FRINGE BENEFITS (A + B + C)				102,292	
D. EQUIPMENT (LIST ITEM AND DOLLAR AMOUNT FOR EACH ITEM EXCEEDING $5,000.)					
TOTAL EQUIPMENT				0	
E. TRAVEL 1. DOMESTIC (INCL. CANADA, MEXICO AND U.S. POSSESSIONS)				5,800	
2. FOREIGN				0	
F. PARTICIPANT SUPPORT COSTS					
1. STIPENDS $ 0					
2. TRAVEL 0					
3. SUBSISTENCE 0					
4. OTHER 0					
TOTAL NUMBER OF PARTICIPANTS (0) TOTAL PARTICIPANT COSTS				0	
G. OTHER DIRECT COSTS					
1. MATERIALS AND SUPPLIES				100	
2. PUBLICATION COSTS/DOCUMENTATION/DISSEMINATION				500	
3. CONSULTANT SERVICES				0	
4. COMPUTER SERVICES				9,594	
5. SUBAWARDS				0	
6. OTHER				19,903	
TOTAL OTHER DIRECT COSTS				30,097	
H. TOTAL DIRECT COSTS (A THROUGH G)				138,189	
I. INDIRECT COSTS (F&A)(SPECIFY RATE AND BASE) 51.5% MTDC (Rate: 51.5000, Base: 118286)					
TOTAL INDIRECT COSTS (F&A)				60,917	
J. TOTAL DIRECT AND INDIRECT COSTS (H + I)				199,106	
K. RESIDUAL FUNDS (IF FOR FURTHER SUPPORT OF CURRENT PROJECTS SEE GPG II.C.6.j.)				0	
L. AMOUNT OF THIS REQUEST (J) OR (J MINUS K)				$ 199,106	$

M. COST SHARING PROPOSED LEVEL $ 0 AGREED LEVEL IF DIFFERENT $

PI/PD NAME
Amer S Diwan

ORG. REP. NAME*
Joyce Kroll

FOR NSF USE ONLY		
INDIRECT COST RATE VERIFICATION		
Date Checked	Date Of Rate Sheet	Initials - ORG

 *ELECTRONIC SIGNATURES REQUIRED FOR REVISED BUDGET

SUMMARY PROPOSAL BUDGET

Cumulative

FOR NSF USE ONLY		
PROPOSAL NO.	DURATION (months)	
	Proposed	Granted
AWARD NO.		

ORGANIZATION
University of Colorado at Boulder

PRINCIPAL INVESTIGATOR / PROJECT DIRECTOR
Amer S Diwan

A. SENIOR PERSONNEL: PI/PD, Co-PI's, Faculty and Other Senior Associates (List each separately with title, A.7. show number in brackets)	NSF Funded Person-months CAL	ACAD	SUMR	Funds Requested By proposer	Funds granted by NSF (if different)
1. Amer S Diwan - PI	0.00	0.00	3.00	$ 34,567	$
2. Elizabeth Bradley - Co-PI	0.00	0.00	3.00	43,291	
3.					
4.					
5.					
6. ( ) OTHERS (LIST INDIVIDUALLY ON BUDGET JUSTIFICATION PAGE)	0.00	0.00	0.00	0	
7. (2) TOTAL SENIOR PERSONNEL (1 - 6)	0.00	0.00	6.00	77,858	
B. OTHER PERSONNEL (SHOW NUMBERS IN BRACKETS)					
1. (0) POST DOCTORAL ASSOCIATES	0.00	0.00	0.00	0	
2. (0) OTHER PROFESSIONALS (TECHNICIAN, PROGRAMMER, ETC.)	0.00	0.00	0.00	0	
3. (6) GRADUATE STUDENTS				186,982	
4. (0) UNDERGRADUATE STUDENTS				0	
5. (0) SECRETARIAL - CLERICAL (IF CHARGED DIRECTLY)				0	
6. (0) OTHER				0	
TOTAL SALARIES AND WAGES (A + B)				264,840	
C. FRINGE BENEFITS (IF CHARGED AS DIRECT COSTS)				32,055	
TOTAL SALARIES, WAGES AND FRINGE BENEFITS (A + B + C)				296,895	
D. EQUIPMENT (LIST ITEM AND DOLLAR AMOUNT FOR EACH ITEM EXCEEDING $5,000.)					
TOTAL EQUIPMENT				0	
E. TRAVEL 1. DOMESTIC (INCL. CANADA, MEXICO AND U.S. POSSESSIONS)				17,400	
2. FOREIGN				0	
F. PARTICIPANT SUPPORT COSTS 1. STIPENDS $ 0 2. TRAVEL 0 3. SUBSISTENCE 0 4. OTHER 0					
TOTAL NUMBER OF PARTICIPANTS (0) TOTAL PARTICIPANT COSTS				0	
G. OTHER DIRECT COSTS					
1. MATERIALS AND SUPPLIES				300	
2. PUBLICATION COSTS/DOCUMENTATION/DISSEMINATION				1,500	
3. CONSULTANT SERVICES				0	
4. COMPUTER SERVICES				28,520	
5. SUBAWARDS				0	
6. OTHER				55,639	
TOTAL OTHER DIRECT COSTS				85,959	
H. TOTAL DIRECT COSTS (A THROUGH G)				400,254	
I. INDIRECT COSTS (F&A)(SPECIFY RATE AND BASE)					
TOTAL INDIRECT COSTS (F&A)				177,476	
J. TOTAL DIRECT AND INDIRECT COSTS (H + I)				577,730	
K. RESIDUAL FUNDS (IF FOR FURTHER SUPPORT OF CURRENT PROJECTS SEE GPG II.C.6.j.)				0	
L. AMOUNT OF THIS REQUEST (J) OR (J MINUS K)				$ 577,730	$

M. COST SHARING PROPOSED LEVEL $ 0 AGREED LEVEL IF DIFFERENT $

PI/PD NAME
Amer S Diwan

ORG. REP. NAME*
Joyce Kroll

FOR NSF USE ONLY		
INDIRECT COST RATE VERIFICATION		
Date Checked	Date Of Rate Sheet	Initials - ORG

C *ELECTRONIC SIGNATURES REQUIRED FOR REVISED BUDGET

PROPOSED BUDGET DETAILS

Institution: The Regents of the University of Colorado, 572 UCB, Boulder, CO 80309-0572

Title: CSR--SMA: Validating Architectural Simulators Using Non-linear Dynamics Techniques

Principal Investigator: Amer Diwan

Co-Principal Investigator: Elizabeth Bradley

Duration: 9-1-07 - 8-31-10

	Year 1	Year 2	Year 3
A. Salaries and Wages			
Principal Investigator: Amer Diwan			
100% time, 1 mo Summer	11,139	11,518	11,910
Co-Principal Investigator: Elizabeth Bradley			
100% time, 1 mo Summer	13,951	14,425	14,915
Graduate Research Assistants: Post-comp--2			
50% time, 9 mos. AY	35,828	37,046	38,306
100% time, 3 mos. Summer	24,427	25,258	26,117
Total Salaries and Wages	85,345	88,247	91,248
B. Fringe Benefits			
PI: 23.4%	2,607	2,695	2,787
Co-PI: 23.4%	3,265	3,375	3,490
GRAs: 7.4%	4,459	4,610	4,767
Total Fringe Benefits	10,331	10,680	11,044
C. Permanent Equipment			
None requested	0	0	0
Total Permanent Equipment	0	0	0
D. Travel			
Domestic:			

E. Other Direct Costs			
1) Materials and Supplies	100	100	100
2) Publication Costs	500	500	500
3) Computer Services			
Services	3,420	3,506	3,594
Computer workstations	6,000	6,000	6,000
4) Other:			
Tuition Remission: Resident, 2, 2, 2	17,222	18,514	19,903
Total Other Direct Costs	27,242	28,620	30,097
F. Total Direct Costs	128,718	133,347	138,189
G. Indirect Costs			
Base:			
On Campus: 51.5% of MTDC, predetermined for the period 7/1/06-6/30/10, provisional thereafter. Per HHS agreement dated 10/30/2006.	57,420	59,139	60,917
H. Total Costs	186,138	192,486	199,106
Total requested for three years:		**577,730**	

Current and Pending Support

(See GPG Section II.C.2.h for guidance on information to include on this form.)

The following information should be provided for each investigator and other senior personnel. Failure to provide this information may delay consideration of this proposal.

Investigator: Amer Diwan

Other agencies (including NSF) to which this proposal has been/will be submitted.

Support: ☒ Current ☐ Pending ☐ Submission Planned in Near Future ☐ *Transfer of Support

Project/Proposal Title: CAREER: Compiler and Run-time systems for modern hardware

Source of Support: NSF

Total Award Amount: $ 400,000 Total Award Period Covered: 08/01/02 - 08/01/07

Location of Project: University of Colorado

Person-Months Per Year Committed to the Project. Cal:0.00 Acad: 0.00 Sumr: 1.00

Support: ☒ Current ☐ Pending ☐ Submission Planned in Near Future ☐ *Transfer of Support

Project/Proposal Title: Understanding the performance of modern systems

Source of Support: NSF

Total Award Amount: $ 400,000 Total Award Period Covered: 09/01/05 - 08/30/09

Location of Project: University of Colorado

Person-Months Per Year Committed to the Project. Cal:0.00 Acad: 0.00 Sumr: 1.00

Support: ☒ Current ☐ Pending ☐ Submission Planned in Near Future ☐ *Transfer of Support

Project/Proposal Title: Algorithmic optimizations in dynamic programming environments

Source of Support: NSF

Total Award Amount: $ 350,000 Total Award Period Covered: 08/01/06 - 07/31/09

Location of Project: University of Colorado

Person-Months Per Year Committed to the Project. Cal:0.00 Acad: 0.00 Sumr: 1.00

Support: ☐ Current ☒ Pending ☐ Submission Planned in Near Future ☐ *Transfer of Support

Project/Proposal Title: Improving on Refactorings

Source of Support: NSF

Total Award Amount: $ 257,000 Total Award Period Covered: 08/01/07 - 07/31/10

Location of Project: University of Colorado

Person-Months Per Year Committed to the Project. Cal:0.00 Acad: 0.00 Sumr: 1.00

Support: ☐ Current ☐ Pending ☐ Submission Planned in Near Future ☐ *Transfer of Support

Project/Proposal Title:

Source of Support:

Total Award Amount: $ Total Award Period Covered:

Location of Project:

Person-Months Per Year Committed to the Project. Cal: Acad: Summ:

*If this project has previously been funded by another agency, please list and furnish information for immediately preceding funding period.

Current and Pending Support

(See GPG Section II.C.2.h for guidance on information to include on this form.)

The following information should be provided for each investigator and other senior personnel. Failure to provide this information may delay consideration of this proposal.

Investigator: Elizabeth Bradley

Other agencies (including NSF) to which this proposal has been/will be submitted.

Support: ☒ Current ☐ Pending ☐ Submission Planned in Near Future ☐ *Transfer of Support

Project/Proposal Title: ITR: Collaborative Research: Software for Interpretation of Cosmogenic Isotope Inventories - A Combination of Geology, Modeling, Software Engineering, and Artificial Intelligence

Source of Support: NSF

Total Award Amount: $ 921,622 Total Award Period Covered: 09/15/03 - 08/31/08

Location of Project: University of Colorado

Person-Months Per Year Committed to the Project. Cal: 0.00 Acad: 0.00 Sumr: 1.00

Support: ☒ Current ☐ Pending ☐ Submission Planned in Near Future ☐ *Transfer of Support

Project/Proposal Title: REU supplement to ITR listed above

Source of Support: NSF

Total Award Amount: $ 12,000 Total Award Period Covered: 09/01/04 - 08/31/08

Location of Project: University of Colorado

Person-Months Per Year Committed to the Project. Cal: 0.00 Acad: 0.00 Sumr: 0.00

Support: ☐ Current ☒ Pending ☐ Submission Planned in Near Future ☐ *Transfer of Support

Project/Proposal Title: A Data-Adaptive Fluids Solver

Source of Support: NSF

Total Award Amount: $ 419,353 Total Award Period Covered: 08/01/07 - 07/31/10

Location of Project: University of Colorado

Person-Months Per Year Committed to the Project. Cal: 0.00 Acad: 0.00 Sumr: 1.00

Support: ☐ Current ☐ Pending ☐ Submission Planned in Near Future ☐ *Transfer of Support

Project/Proposal Title:

Source of Support:

Total Award Amount: $ Total Award Period Covered:

Location of Project:

Person-Months Per Year Committed to the Project. Cal: Acad: Sumr:

Support: ☐ Current ☐ Pending ☐ Submission Planned in Near Future ☐ *Transfer of Support

Project/Proposal Title:

Source of Support:

Total Award Amount: $ Total Award Period Covered:

Location of Project:

Person-Months Per Year Committed to the Project. Cal: Acad: Summ:

---

*If this project has previously been funded by another agency, please list and furnish information for immediately preceding funding period.

FACILITIES, EQUIPMENT & OTHER RESOURCES

FACILITIES: Identify the facilities to be used at each performance site listed and, as appropriate, indicate their capacities, pertinent capabilities, relative proximity, and extent of availability to the project. Use "Other" to describe the facilities at any other performance sites listed and at sites for field studies. USE additional pages as necessary.

Laboratory:

Clinical:

Animal:

Computer: We already have significant computing infrastructure for our simulations. We will use the equipment funds in the proposal to buy new machines so we can investigate issues with validating simulators for those machines.

Office:

Other:

MAJOR EQUIPMENT: List the most important items available for this project and, as appropriate identifying the location and pertinent capabilities of each.

OTHER RESOURCES: Provide any information describing the other resources available for the project. Identify support services such as consultant, secretarial, machine shop, and electronics shop, and the extent to which they will be available for the project. Include an explanation of any consortium/contractual arrangements with other organizations.