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: Luke-jon Tierney

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:

Andreas Buja,
Dianne Cook,
Jan DeLeeuw,
James Gentle,
Michael M. Meyer,
Deborah Nolan,
Balasubramanian Narasimhan,
David W. Scott,
Antony Unwin,
Edward Wegman

REVIEWERS NOT TO INCLUDE:

Not Listed

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 08-1

NSF 08-1 **11/07/08**

FOR CONSIDERATION BY NSF ORGANIZATION UNIT(S) (Indicate the most specific unit known, i.e. program, division, etc.)

DMS - STATISTICS

FOR NSF USE ONLY

NSF PROPOSAL NUMBER

0906398

DATE RECEIVED	NUMBER OF COPIES	DIVISION ASSIGNED	FUND CODE	DUNS# (Data Universal Numbering System)	FILE LOCATION
11/04/2008	**2**	**03040000 DMS**	**1269**	**062761671**	**09/29/2009 11:02am S**

EMPLOYER IDENTIFICATION NUMBER (EIN) OR TAXPAYER IDENTIFICATION NUMBER (TIN)

426004813

SHOW PREVIOUS AWARD NO. IF THIS IS
☒ A RENEWAL
☐ AN ACCOMPLISHMENT-BASED RENEWAL

0604593

IS THIS PROPOSAL BEING SUBMITTED TO ANOTHER FEDERAL AGENCY? YES ☐ NO ☒ IF YES, LIST ACRONYM(S)

NAME OF ORGANIZATION TO WHICH AWARD SHOULD BE MADE

University of Iowa

AWARDEE ORGANIZATION CODE (IF KNOWN)

0018929000

ADDRESS OF AWARDEE ORGANIZATION, INCLUDING 9 DIGIT ZIP CODE

University of Iowa
2 Gilmore Hall
Iowa City, IA. 522421320

NAME OF PERFORMING ORGANIZATION, IF DIFFERENT FROM ABOVE

PERFORMING ORGANIZATION CODE (IF KNOWN)

ADDRESS OF PERFORMING ORGANIZATION, IF DIFFERENT, INCLUDING 9 DIGIT ZIP CODE

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 **Computing Environments for Stastistics**

REQUESTED AMOUNT	PROPOSED DURATION (1-60 MONTHS)	REQUESTED STARTING DATE	SHOW RELATED PRELIMINARY PROPOSAL NO. IF APPLICABLE
$ **390,965**	**36** months	**06/01/09**	

CHECK APPROPRIATE BOX(ES) IF THIS PROPOSAL INCLUDES ANY OF THE ITEMS LISTED BELOW

☐ BEGINNING INVESTIGATOR (GPG I.G.2)
☐ DISCLOSURE OF LOBBYING ACTIVITIES (GPG II.C)
☐ PROPRIETARY & PRIVILEGED INFORMATION (GPG I.D, 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 _____
PHS Animal Welfare Assurance Number _____

☐ HUMAN SUBJECTS (GPG II.D.6) Human Subjects Assurance Number _____
Exemption Subsection _____ or IRB App. Date _____
☒ INTERNATIONAL COOPERATIVE ACTIVITIES: COUNTRY/COUNTRIES INVOLVED (GPG II.C.2.j)
AU NZ SZ UK
☐ HIGH RESOLUTION GRAPHICS/OTHER GRAPHICS WHERE EXACT COLOR REPRESENTATION IS REQUIRED FOR PROPER INTERPRETATION (GPG I.G.1)

PI/PD DEPARTMENT
Statistics and Actuarial Science

PI/PD FAX NUMBER
319-335-3017

PI/PD POSTAL ADDRESS
241 Schaeffer Hall

Iowa City, IA 52242
United States

NAMES (TYPED)	High Degree	Yr of Degree	Telephone Number	Electronic Mail Address
PI/PD NAME **Luke-jon Tierney**	**PhD**	**1980**	**319-335-3386**	**luke@stat.uiowa.edu**
CO-PI/PD				
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 Authorized Organizational Representative or Individual Applicant 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), nondiscrimination, and flood hazard insurance (when applicable) as set forth in the NSF Proposal & Award Policies & Procedures Guide, Part I: the Grant Proposal Guide (GPG) (NSF 08-1). 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).

Conflict of Interest Certification

In addition, if the applicant institution employs more than fifty persons, by electronically signing the NSF Proposal Cover Sheet, the Authorized Organizational Representative 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 the NSF Proposal & Award Policies & Procedures Guide, Part II, Award & Administration Guide (AAG) Chapter IV.A; 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 dislosed 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 Exhibit II-3 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 Exhibit II-4 of the Grant Proposal Guide.

Certification Regarding Lobbying

The following 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.

Certification Regarding Nondiscrimination

By electronically signing the NSF Proposal Cover Sheet, the Authorized Organizational Representative is providing the Certification Regarding Nondiscrimination contained in Exhibit II-6 of the Grant Proposal Guide.

Certification Regarding Flood Hazard Insurance

Two sections of the National Flood Insurance Act of 1968 (42 USC §4012a and §4106) bar Federal agencies from giving financial assistance for acquisition or construction purposes in any area identified by the Federal Emergency Management Agency (FEMA) as having special flood hazards unless the:

(1) community in which that area is located participates in the national flood insurance program; and
(2) building (and any related equipment) is covered by adequate flood insurance.

By electronically signing the NSF Proposal Cover Sheet, the Authorized Organizational Representative or Individual Applicant located in FEMA-designated special flood hazard areas is certifying that adequate flood insurance has been or will be obtained in the following situations:

(1) for NSF grants for the construction of a building or facility, regardless of the dollar amount of the grant; and
(2) for other NSF Grants when more than $25,000 has been budgeted in the proposal for repair, alteration or improvement (construction) of a building or facility.

AUTHORIZED ORGANIZATIONAL REPRESENTATIVE	SIGNATURE	DATE
NAME **John S Massa**	**Electronic Signature**	**Nov 4 2008 6:51PM**

TELEPHONE NUMBER	ELECTRONIC MAIL ADDRESS	FAX NUMBER
319-335-2123	**john-massa@uiowa.edu**	**319-335-2130**

*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.

A Project Summary

The objective of the proposed research is to improve the ability of researchers, instructors, and other users of statistical methodology to apply statistical methods in new problem areas. The primary intellectual contribution of the proposed research will be the exploration and development of new principles for the design of statistical software to take advantage of modern computing power. This work will explore a range of issues, including the use of parallel computing, compilation and static code analysis in statistical computing environments. The broader impact resulting from the proposed research is to improve the ability of researchers, instructors, and other users of statistical methodology to effectively apply this methodology in scientific research and teaching. This is accomplished by making available results of the research as enhancements and additions to the open source statistical software system R, and by influencing the design of other statistical software through the R system and through publications resulting from the proposed research.

TABLE OF CONTENTS

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

	Total No. of Pages	Page No.* (Optional)*
Cover Sheet for Proposal to the National Science Foundation		
Project Summary (not to exceed 1 page)	1	
Table of Contents	1	
Project Description (Including Results from Prior NSF Support) (not to exceed 15 pages) **(Exceed only if allowed by a specific program announcement/solicitation or if approved in advance by the appropriate NSF Assistant Director or designee)**	15	
References Cited	4	
Biographical Sketches (Not to exceed 2 pages each)	2	
Budget (Plus up to 3 pages of budget justification)	5	
Current and Pending Support	2	
Facilities, Equipment and Other Resources	2	
Special Information/Supplementary Documentation	0	
Appendix (List below.) **(Include only if allowed by a specific program announcement/ solicitation or if approved in advance by the appropriate NSF Assistant Director or designee)**		
Appendix Items:		

*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.

C Project Description

The proposed research represents a continuation of research supported under grant DMS-0604593 "Computing Environments for Statistics" for the period from 06/01/06 to 05/31/09 for a total award to the University of Iowa of $295,663.

C.1 Results from Prior NSF Support

C.1.1 Statistical Computing Environments

Research under DMS-0604593 resulted in the following formal publications related to computing environments.

Rossini, A. J., Tierney, L., and Li, N., "Simple parallel statistical computing in R," *Journal of Computational and Graphical Statistics*, 16 (1), 399–420, 2007.

This paper outlines how a simple framework for parallel computing within a high level statistical language can be used to effectively improve performance for a range of problems.

Tierney, L., Rossini, A. J., and Li, N., "`snow`: A parallel computing framework for the R system," *International Journal of Parallel Programming*, to appear.

This paper presents the current design and implementation of the parallel computing framework `snow` for a computer science audience.

Tierney, L., "R: An overview and some current directions," *Journal of the Korean Statistical Society*, 36 (1), 31–58, 2007.

This paper provides an overview of R and outlines efforts towards compilation and parallelization.

Tierney, L., "Code analysis and parallelizing vector operations in R," *Computational Statistics*, to appear.

This paper provides an overview of the current state of code analysis tools for R and presents some preliminary results on parallelizing basic vectorized mathematical functions.

Tierney, L., "Implicit and explicit parallel computing in R," in "COMPSTAT 2008: Proceedings in Computational Statistics," Paula Brito (ed.), Physica Verlag, 43–51, 2008.

This paper outlines explicit parallelization based on the `snow` framework and further developments in parallelizing vectorized mathematical functions.

Feng, D., and Tierney, L., "Computing and Displaying Isosurfaces in R," *Journal of Statistical Software*, 28 (1), 1–24, 2008.

This paper presents the design and outlines the implementation of the 3D contouring facilities provided by the `misc3d` package.

Further work in this area was made public through less traditional means, in particular through the releases of new versions of the R system and of packages for R.

Continuing work on compilation has emphasized some foundational issues, including name space management and static code analysis. These work together to obtain a clearer understanding of

the semantics of the code to be compiled. The refined code analysis work has resulted in a separate package `codetools` that became a recommended package included with the basic R distribution starting with R 2.5. This package is now used as part of the package checking process by R package developers and the CRAN archive maintainers.

A package `misc3d`, originally released in 2005, has been substantially revised and extended. Rendering of 3D contours now also supports standard and grid graphics in addition to interactive `rgl` graphics. This is joint work with Dai Feng and forms part of his Ph. D. thesis.

The `proftools` package for processing and visualizing R profiling output was released in 2007. This package helps identifying computational bottlenecks and directing effort for performance improvement of R code.

New versions of the `snow` package for parallel computing were released in 2007 and 2008. These add support for NWS as a fourth low-level communications mechanism, and the ability to run `snow` clusters on Windows and mixed Windows/Linux/UNIX environments. A version with support for collecting and visualizing timing information for parallel computations is in preparation.

Experimental packages `pnmath` and `pnmath0` have been made available on the PI's web site. These implement a strategy for parallelizing basic vectorized mathematical functions. Refining the strategy and integration into the standard R distribution are part of the proposed research.

C.1.2 Other Areas

Work on Markov chain Monte Carlo methods resulted in one new submission:

Haran, M. and Tierney, L., "Exact and approximate samplers for spatial generalized linear models," submitted for publication.

Collaboration with Dr. Haran, who was supported during his graduate studies under an earlier NSF grant, is ongoing and will likely lead to further publications.

Dr. Dai Feng, who received support from this grant, completed his Ph. D. in May 2008. His thesis developed a method for brain tissue classification based on MRI images. A joint paper on this work is in preparation and should be submitted within a few months.

Dr. Jung-Eun Song, who also received some support on this grant, completed her Ph. D. in June 2008. Her work involved the study of Bayesian partition models as a non-parametric method for fitting curves and surfaces. A package implementing the methodology and a paper are in preparation.

C.2 Proposed Research

The objective of this research is to develop design principles for the next generation of statistical software systems that will allow statisticians to effectively take advantage of computational resources that are becoming available. These resources include fast multi-core processors and bitmapped displays, distributed data bases, and distributed computing facilities, among others.

Modern data analysis environments used by practitioners, methodological researchers, and instructors in statistics include S, R, MATLAB, among others. The fact that these environments incorporate or are based on programming languages is crucial to their ability to implement new approaches and to support adapting analyses of scientific problems to the requirements of a problem, rather than adapting the problem to available analysis tools. Experience with these systems has underscored the value of the language-based approach, but has also revealed many weaknesses that need to be addressed in future software generations. In addition, new hardware and operating system capabilities have opened up new opportunities that can only be exploited if they are accessible from statistical environments. Much can be learned by drawing on results and experience in Computer Science research on high level languages, but there are features, and combinations of features, of statistical computing that are not well addressed by standard language research. These include the need for vector and array operations, the need for frameworks for data management that are integrated with the language, and the need to support interactive and experimental programming. The challenge to research on statistical computing environments and languages is to take advantage of progress made in Computer Science, and perhaps contribute to that progress, while at the same time addressing these unique needs of statistical computing.

The proposed research will involve a number of different but related areas and approaches. Some of these are outlined below. The primary framework for experimenting with the proposed ideas and for disseminating the results of this work will be the R system. Due to the massive increase in popularity of R in recent years this ensures both a thorough testing of ideas and a substantial impact of successful approaches. The development of R is a collaborative effort among 19 researchers throughout the world, with significant contributions from the user community. Incorporation of new ideas will involve coordination with many members of the R core development group. Where more extensive collaboration with specific individuals is in progress or anticipated this is noted in the discussion below.

C.2.1 Parallel and Distributed Statistical Computations

The ability to take advantage of opportunities for parallel computation in scientific computing is becoming increasingly important. Improvements in speed of individual processors have leveled off, but multiple processor cores are becoming more widely available, both as multi-core CPUs in individual machines and as collections of workstations in departments or rack mounted clusters. Over the next few years it is expected that in scientific computing in general significant performance improvements will have to come primarily from taking advantage of these opportunities for parallel computing [8]. Parallel programming is however quite difficult, so it is important to provide tools and frameworks to assist users of scientific software in taking advantage of these opportunities.

This research will focus on two different approaches: creating frameworks for explicit parallel computing that are easy to use but nevertheless effective for a wide range of problems, and implicit approaches in which the internals of a system like R are modified to carry some computations out in parallel without the need for any explicit user specification. Work on explicit parallelization will

be further divided into high-level tools that are intended to be mastered very easily and lower level tools that require more experience but can express a richer set of parallel computations.

High-Level Explicit Parallelization

Previous research has lead to the development of the `snow` parallel computing framework [41, 48]. The framework is designed to implement simple scatter-compute-gather computations on top of a distributed memory message passing framework. A set of worker R processes is started on one or more computers by a master R process, and functions are provided for partitioning jobs, sending the jobs from the master to the workers, and retrieving results in the master proces. Four different low level message passing mechanisms are supported: MPI [34], PVM [20], NWS [39], and a basic socket mechanism. The computational functions are modeled on the `apply` family of functions familiar to R programmers. The `snow` package has been quite successful and is now used in a range of packages available from the CRAN and Bioconductor repositories. Nevertheless, there is still considerable room for further progress, and the proposed research will take advantage of experience gained so far in moving the framework forward.

The original framework was easiest to deploy in a UNIX/Linux framework. With increasing availability of Windows clusters, along with a quite common scenario of researchers using Windows workstations to access Linux clusters, there is a strong need to better support Windows and mixed Windows/Linux settings. For use on stand-alone multi-core computers it is also useful to be able to avoid the need for additional software, such as message-passing libraries that may be difficult to set up for either technical or political reasons. The current development version of the `snow` package includes experimental enhancements to the socket-based framework that provide much enhanced support for Windows operating systems and for handling mixed environments consisting of Windows, Linux, and Mac OS X computers. Further work is needed in testing and refining these enhancements to improve their reliability and ease of use.

The highest level interface of the `snow` framework includes functions like `parLapply`, a parallel version of the standard R function `lapply` that applies a specified function to each element of a generic vector and returns a vector of the results. The parallel version partitions the vector into chunks, one for each worker, applies `lapply` on the worker processes, and assembles the results. Experience with the interface provided by the Open MP [15] framework suggests that it would be useful to provide some additional controls to the partitioning process. A minimal chunk size could be used to specify that the `lapply` should be done sequentially if the vector is not long enough for the benefits of parallelizeiton to outweigh the cost of synchronization. Also useful would be the ability to request that a larger number of smaller chunks be used in a load-balancing fashion, i.e. assigning additional chunks of work to workers that finish first. Open MP provides scheduling directives for this purpose; we will explore whether similar approaches can be used effectively with the higher-level `snow` functions.

Another area in need of further exploration is the handling of errors. Currently if any of the workers in a scatter-compute-gather computation produces an error then the entire computation will signal an error on the master process. This is often the right result, but there are situations where one might want to filter out computations that produce an error and return the non-error results. It is possible to program alternative strategies using the error handling mechanisms provided by R, but this can be quite tedious and we will therefore explore approaches for more easily specifying alternative error handling strategies. The ability to cleanly interrupt parallel computations is also limited at best; improvements will need to rest on improved control of asynchronous error handling

as mentioned in Section C.2.3 below.

Profiling tools are a valuable aid in developing parallel programs. For message-passing systems in particular it is useful to have tools for visualizing how much of the run time of a parallel computation is spent actually performing computations as opposed to communicating or waiting for results. Visualization tools are available for some message-passing back ends, such as the `xpvm` tool for PVM and the `xmpi` tool for the LAM implementation of MPI. But not all versions of MPI offer such tools, and no such tools are available for use with a socket-based communication mechanism. Research will explore whether the `snow` implementation can be modified to collect relevant data on timing and communication and then display those using R's own graphics facilities. An initial experiment along these lines is available in the development version of `snow`; further refinements and enhancements will be explored.

A high-level paradigm for parallel computations that has received considerable attention in recent years is Google's Map/Reduce approach [16]. A group of researchers at Duke University lead by Ricardo Pietrobon is currently investigating ways of taking advantage of this framework within R, and they are initially looking at possible extensions of the `snow` framework as a way of implementing this. It is expected that collaboration with this effort will lead to useful enhancements to the `snow` framework.

Intermediate-Level and Low-Level Explicit Parallelization

The basic model for the `snow` toolkit is a single scatter-compute-gather operation. Many interesting algorithms can be expresses as a single initial scatter of data, a sequence of scatter-compute-gather operations in which communication is limited, and ending with a larger gather operation. This can be handled by the current package if data is stored in global variables on the worker nodes across calls to the parallel execution functions. While feasible, this has all the usual drawbacks of using global state. A research objective will be to develop a clean mechanism for specifying a session that manages these persistent values, ensures that the values are released once the computation is complete, and possibly ensures that intermediate values can be garbage collected. A related issue is to integrate distributed arrays for use with the ScaLapack parallel linear algebra package into the framework. Also useful in this context would be a means of loading such arrays from disk and storing distributed arrays to disk; this would help in situations where an array is too large for the memory on a single machine but can fit in the collective memory of nodes in a cluster. The RScaLapack package, which unfortunately is no longer under active development, did not address the issue of maintaining distributed arrays across calls, though other aspects of this package may prove useful. The design of the MATLAB distributed computing toolbox as described in [43] and the PyACTS system described in [17] may also provide useful guides.

The possibility of smoothly integrating the basic `snow` functionality with other richer parallel programming paradigms will also be explored. Because of its avoidance of deadlock, the BSP model [9] represents a particularly attractive approach. Implementations of this model within Python [22] and Objective Caml [31, 29] may serve as useful starting points.

At times it can be useful to have available a lower level parallel computing framework that allows creation of individual processes and communication among any of these processes. Such a framework could be useful for creating servers providing statistical functionality as a web service, for implementing responsive graphical user interfaces to one or more computational engines, and for implementing higher level mechanisms like `snow`. An interesting possibility that will be explored is whether a useful framework can be built on the model provided by the functional language Erlang

[7], which was originally designed for creating highly reliable software for the telecommunications industry. The basic Erlang parallel communication framework is built around three primitives: `spawn` for creating a new process, `send` for asynchronously sending a message to a process, and `recv` for waiting for a message, or until an optional timeout expires. The processes do not share memory — all information needed by cooperating processes is transmitted using messages. These high level language processes can be implemented as separate operating system processes, possibly running on separate computers connected by a network, or as operating system threads running within a single operating system process, or as a combination. An additional possibility that only provides concurrency rather than true parallelism is to interleave concurrent processes within an R evaluator running in a single thread.

A preliminary implementation of this mechanism in which each R process is mapped to a separate R session and operating system process shows considerable promise. This approach uses a manager process that acts as a server for exchanging messages; this design is similar to the design used in the iPython framework [35]. An interesting question is whether this model can effectively be used with operating system threads within a single process or a collection of operating system processes to provide a form of multi-threading for R. Previous efforts to add thread support to R have stumbled over issues related to the heavy use of global variables in many R packages. The fact that in this model the R processes would not share memory may make it possible to use a design in which "safe" R processes that do not use packages making use of globals can run on threads within a single operating system process, while "unsafe" R processes run in separate OS processes. Use of separate OS processes for less trusted code can also provide some protection from crashes caused by errors in package C code. This approach to using OS processes in place of threads for increased reliability of an overall application has recently been introduced into development versions of several web browsers to protect against faults in browser plug-ins.

Implicit Parallelization

The basic idea in implicit parallelization of a high level language like R is to identify operations that are computationally intensive and to arrange for the work of these operations to be divided up among several processors without requiring explicit intervention on the part of the user. This involves the use of multiple threads. In some cases libraries using this approach are readily available and can be used by linking them into R. For example, most linear algebra computations are based on the basic linear algebra subroutines library, or BLAS. R provides its own basic implementation derived from an open source reference implementation, but makes it easy to substitute an alternate implementation, such as a hardware vendor library or one from the ATLAS project [49]. A number of BLAS implementations provide threaded versions that try to improve performance by using multiple threads. A major challenge is that there is overhead associated with synchronization of threads, among other things, that can result in threaded versions running slower than non-threaded ones. This has been observed in the use of threaded BLAS libraries.

Another candidate for implicit parallelization is R's vectorized arithmetic operations. The R math library includes many special functions, densities, cumulative distribution functions, and quantile functions. R level versions of these functions apply the functions to all elements of vector arguments. This is currently implemented by a simple loop. If the work of this loop is divided among several processors then the resulting computation may run faster. However, care is needed as there is synchronization overhead, and shared resources (memory, bus, etc.) impose bottlenecks. As a result, while parallelization of vectorized operations will be beneficial for large vectors, it can

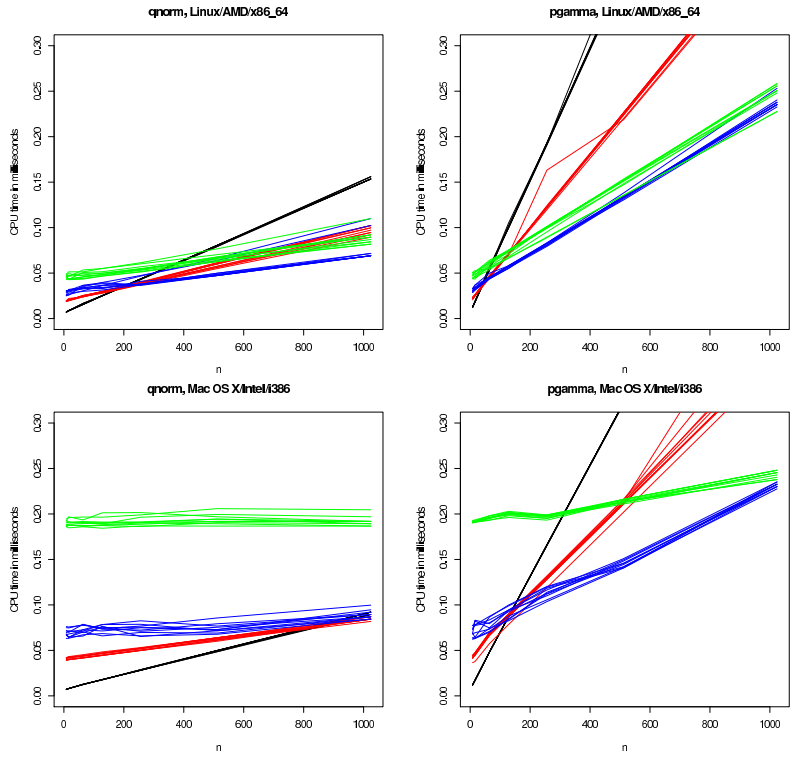

Figure 1: Timings of vectorized function evaluations for `qnorm` and `pgamma` as a function of vector length for two 8-processor systems. Plots for 10 replications are shown. Black: single thread; red: 2 threads; blue: 4 threads; green: 8 threads.

be harmful for short ones. Careful tuning is needed to ensure that parallelization is only used if it will be helpful.

Figure 1 shows performance measurements for a range of vector sizes and two functions on two 8-processor systems. Some simple empirical observations from these and similar plots for other functions: Times are roughly linear in vector length for each function/OS/thread number combination, the intercepts are roughly the same for all functions on a given platform, and the relative slopes for different functions appear to be roughly independent of OS/architecture. These observations motivate a simple cost model for determining the appropriate number of threads to use given the vector size, the number of available processors, and information on OS/architecture performance encoded in pre-determined slope and intercept values. Slopes can be determined once and updated only when implementations change or new functions are added; intercepts can be determined using a calibration experiment run at installation time, perhaps with an option of using pre-computed values for common OS/architecture combinations. The slope for a simple function, like the normal density function `dnorm`, can be used as a base line; thus timings are computed in units of single element `dnorm` calculations. Experiments suggest that the two-processor intercept for Linux/AMD/x86_64 is approximately 200 `dnorm` evaluations; for Mac OS X 10.4/Intel/i386 it is around 500. Thus for Mac OS X parallel evaluation of `dnorm` calls only pays for vectors of length 500 or more. For simpler functions like `sqrt` the cutoff is even higher, but there are some very computationally intensive functions, such as `qtukey`, where parallel evaluation is useful for vectors

of two elements.

Implementing the approach outlined above involves using threads to evaluate different parts of the basic vectorization loops. One possibility is to directly use a basic threading API, such as `pthreads`, but a better choice is to use Open MP (Chandra et al. 2000). Many commercial compilers as well as gcc 4.2 support Open MP. Open MP uses compiler directives (`#pragma` statements in C; FORTRAN uses structured comments) to request parallel implementation of a loop. A compiler that does not support Open MP will ignore the Open MP directives. Use of Open MP eliminates the need to manually manage threads, but some effort is still needed. Only loops with simple control structure can be parallelized by Open MP, which requires rewriting some of the loops used in the standard R code. Also, it is essential that the functions being called are safe to call from multiple threads. For this to be true these functions cannot use read/write global variables, call R's memory manager, signal warnings or errors, or check for user interrupts. Almost all functions in the basic R math library are either thread-safe or easily modified to be thread-safe. Exceptions are the Bessel functions and the Wilcoxon and signed rank functions — these use global variables to store allocated memory for intermediate results and will need to be rewritten to make them thread-safe.

A preliminary implementation of the approach outlined here is available as a package `pnmath` [46]. Loading this package replaces the standard vectorized functions in R by parallelized ones. The next step is to integrate this work into the R distribution and resolve some configuration issues related to detecting libraries and compiler flags needed for Open MP support. Once this is complete, other functions will be examined for possible parallelization. Obvious candidates include calculations of matrix row and column sums and means and some simple applications of the `sweep` and `outer` functions. More extensive options may become available with the support of compilation as discussed in the following section.

C.2.2 Compilation and Code Analysis for Statistical Languages

Traditionally, statistical and numerical languages such as S [13], R [26], MATLAB [44], or Lisp-Stat [47] have been designed around an interpreter. This works well as long as the languages are used primarily as a simple mechanism for linking together basic numerical routines written in Fortran or C. But as more sophisticated programs are developed in the languages themselves, the need for greater speed becomes more important, as does the need for tools to aid in ensuring code correctness. Common current practice is to develop prototypes in the high-level statistical languages and then manually translate critical portions into a lower level language. This is error prone and time consuming. In Computer Science, new languages are almost always designed with compilation to a lower level target language in mind. Sometimes this target language is C, sometimes the machine language of a particular computer, but often it is the machine language for a conceptual, or virtual, machine. This machine language is then either interpreted by a virtual machine implementation or translated further to executable code. The advantage of compilation even to an interpreted byte code is that many operations, such as type checking, identification of the locations of local variables, or establishing targets for non-local exits, that have to be done by an interpreter at run time can safely be carried out at compile time or eliminated entirely as the result of static code analysis. This can result in very substantial performance improvements with no user intervention. It may not be reasonable to hope that the need to code some portions in C or Fortran can be eliminated entirely, but even modest performance improvements can reduce the number of situations where manual translation is needed. In addition, speed improvements through compilation can allow code

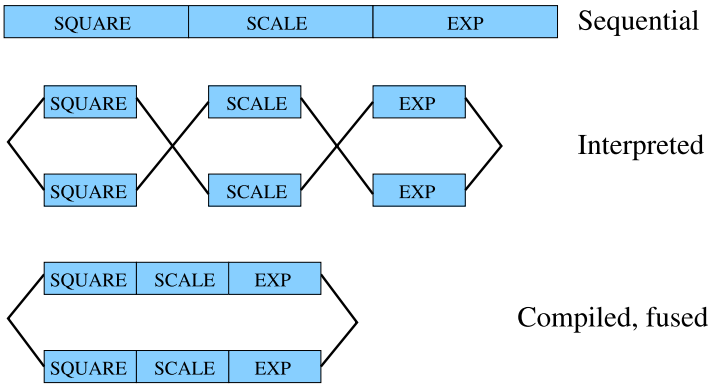

Figure 2: An illustration of the reduced synchronization in compound vector operations made possible by compilation.

implementing the statistical language itself that is currently implemented in C to be moved to the higher level language, thus making the code easier to maintain.

As described in Section C.1, work under the previous grant made significant progress on code analysis tools, and the results have been incorporated into the standard R distribution. Less progress was made on compilation itself, in large part because it has become clear that some changes in direction are needed from the initial approach first outlined in [45] and leading to the currently available experimental compiler. Three major issues need to be addressed: compiler-based support for parallel computing, improvements in loop performance for scalar data, and mantainability within the context of an open source project.

With the increasing availability of multi-core processors the need to take advantage of parallelism to improve performance has come further to the forefront. The implicit parallelization approach outlined in the previous section can be effective for large vectors, but because of overhead involved in setting up and synchronizing parallel computations, parallel computation will not help for modest size vectors when the cost of this overhead has to be paid for each vector operation. With compilation it is possible to take advantage of the observation that most arithmetic operations occur in larger expressions which the compiler can fuse to amortize the overhead and thus make parallelization pay off for smaller vector sizes. As a simple example, consider the expression

```
exp(-0.5 * x ^ 2)
```

applied to a vector `x`. Figure 2 illustrates how sequential, interpreted parallel, and compiled parallel approaches might work. The computation consists essentially of three steps: squaring the values in the vector, scaling the results, and exponentiating the scaled results. An interpreted parallel approach would need to synchronize its threads around each step. A compiler could fuse the steps and synchronize only at the beginning and end of the fused operation. A cost model along the lines described in the previous section can be used to compute at compile time for each fused expression a threshold for determining when the sizes of vectors involved are sufficient to warrant use of multiple threads. In addition to recognizing compound expressions that can be parallelized as a unit, compilation can also recognize simple applications of the `apply` family of functions that can be performed in parallel, for example by analyzing the functions applied to determine that they are free of side effects.

While R and other vector-based languages perform best when code it written to take advantage

of vectorized operations, there are situations where scalar loops are difficult to avoid. Improving the performance of such loops is one of the goals of any compilation project. While the initial R byte code compiler made some good progress in this direction, recent results obtained by the Psyco [3] specializing compiler for Python and also Stephen Milborrow's `jit` compiler for R loops [32] suggest that much more progress can be made within even a byte compiled framework. Research will explore, in collaboration with Milborrow, whether ideas from the R `jit` framework can be effectively incorporated into a compiler and runtime system for the full R language. Ideas from the Psyco and PyPy [4] projects may also prove useful. The initial compilation target will remain a specialized virtual machine designed specifically for R, but other options will be considered as well. These include the LLVM framework that can generate native machine code, and other byte code systems, such as the JVM and CIL [2] as used in the Microsoft .Net framework. Other possible approaches include compilation to C code, which has been considered in the RCC project [18, 19]. Ihaka and Temple Lang [25] are also exploring the possibility of using Common Lisp as a low level implementation base, in particular Common Lisp implementations with strong compilation support. This approach may require more changes to the R language than seems desirable at this time but is worth examining.

An important aspect of the success of R is its open source nature and the fact that a significant number of researchers, 19 currently, participate in its maintenance and evolution. One consequence, however, that has become apparent is that having two evaluation mechanisms, one based on interpreting source code and one based on byte code, creates maintenance problems: it is too easy to make changes in one without making the corresponding change in the other, resulting in the two mechanisms drifting apart. It is therefore important to move as soon as possible to a framework in which all code is compiled. It will also be necessary to extensively document the working of the virtual machine and the compiler so all members of the development group can participate in its maintenance if needed. It is important to emphasize that a compilation-based approach in no way inhibits interactive use of the language. It merely means that the evaluation mechanism compiles an expression and then executes the compiled code instead of directly interpreting the internal representation of the source language.

Once an effective virtual machine design has been identified and an initial compiler targeting the machine has been constructed it will be necessary to explore opportunities for compiler optimizations. A number of additional opportunities for optimization are available. One costly operation in tight numerical loops is the allocation of heap storage for intermediate results. The interpreter cannot afford to do very much about this, but a compiler, perhaps aided by suitable assertions in the code, can determine when storage can be safely reused immediately without going through the memory manager. In some cases it may be possible to allocate the storage in a particularly efficient way, say using a stack discipline. Work on compiling APL [12] and NESL [10] may provide useful approaches. These optimizations will work hand in hand with the efforts to parallelize vector operations as discussed above.

The amount of process stack available is typically rather limited by the operating system. The R interpreter, like many interpreters, implements R function calls with a series of nested C function calls, which uses the process stack and thus limits the possible depth of recursion in R. A virtual machine can be designed to reduce process stack usage by a combination of heap allocation of the R call stack and tail call optimizations to eliminate unnecessary stack growth. R semantics place some limits on such optimizations; research will explore what is possible and how making available user declarations might help enable more aggressive optimization.

As R is often used in a functional style, results on compiling Lisp [27], ML [5] and other mostly functional languages [6] should also prove helpful in developing optimization strategies for compiling R. The R language also uses a limited form of lazy evaluation that applies only to function arguments. Since R, unlike fully lazy languages like Haskell [36], does not use lazy data structures, lazy evaluation has less impact on performance, but there still is some impact and it is worth exploring whether techniques such as conservative strictness analysis can be used to improve performance and perhaps enable other optimizations. Research will also examine whether small changes in R semantics may help in enabling optimizations. One example is to reduce the requirement to provide access to the full call stack — the immediate caller is sufficient in almost all cases. This would allow tail call optimization, an important technique for compiling functional languages.

One area in which languages to support statistical computing differ from general purpose languages is in the need for a much wider range of basic mathematical functions. In addition to standard arithmetic operations and transcendental functions there is a need for functions to evaluate densities, cumulative distribution functions, and quantile functions for standard distributions. Furthermore, users need to be able to add support for nonstandard distributions and other special functions. Ideally it should be possible to allow user packages to add support for such functions in a way that integrates them with the compilation system and allows them to take advantage of the parallel evaluation mechanism just described. Research will explore the development of an effective and disciplined approach to allow the compiler and virtual machine to be extended in this way.

Initially the compiler will be designed for the R language and the Virtual machine will be closely tied to the current R runtime system. A longer term goal is to produce an abstract virtual machine that can serve as a compilation target for several different statistical languages, including Lisp-like, R-like, and graphical languages, and that in turn can be implemented by direct interpretation of virtual machine code, or by translation to C code, or to Java JVM or CIL byte code.

Work on code analysis tools will continue as part of the overall efforts on compilation of R. In particular with dynamic languages like R code analysis can identify some bugs with certainty, but in many cases it can only identify a situation that represents a possible bug that merits further investigation. A balance must be struck between reporting too many and too few possible bugs by providing control over classes of issues that are shown, and if possible a useful ranking within these classes. Code analysis for error detection has been the subject of recent research in Computer Science where some strategies have uncovered a number of previously unknown bugs in large code bases, including the Linux kernel [28, 24]. The possibility of transferring results from this research to the R context will be explored.

C.2.3 Other Areas of Research

This section describes a few additional areas that will be addressed as time permits or the need arises to support other aspects of the investigation.

Exception Handling in Statistical Languages

Structured exception handling is an important component of all modern programming languages. It is also becoming more important to statistical languages as the opportunity and need to access resources over the internet from within a computation has increased, and these opportunities bring with them a host of new types of possible failures that need to be addressed in different ways. A

powerful exception handling system supporting both calling handler semantics (in the spirit of Unix signal handlers) and exiting handler semantics (along the lines of mechanisms provided by Java or C++) has been developed and incorporated into R. The mechanism also includes a separate mechanism for managing restarts, points at which a computation can be resumed. The previous simple mechanisms have been reimplemented on top of the new mechanism. The mechanism is strongly influenced by the Common Lisp model [38], with some simplifications taken from the Dylan model [42]. Experience so far has shown this approach to be successful in practice, but several issues remain to be addressed in continued research.

One important task is to develop a hierarchy of exception conditions to represent the many possible error conditions signaled by internal R code. Currently these all produce the same type of error condition. For effective programmatic error handling it is important to allow code to distinguish, for example, between the error of providing an improperly formatted URL and an error due to a failed connection. A refined set of error conditions will provide a means for making this distinction programmatically.

Another area that needs attention is the possibility of integrating R with structured exception handling mechanisms in other software. Many languages that are now used to write code that may need to be interfaced to R provide structured exception handling mechanisms and may also assume that this code is only exited through ordinary returns or use of these exception handling mechanisms. C++ is one example; the operating system structured exception handling mechanism on the MS Windows platform is another. It would be very useful, if not essential, for R exceptions that occur in code called from C++ to be converted to C++ exceptions and vice versa. Means for accomplishing this, to the extent possible, will be investigated.

A further issue related to exception handling is the handling of asynchronous exceptions, in particular user interrupts. Providing a means for users to interrupt computations that take too long or were started in error is essential for any interactive statistical system, and R provides such a mechanism on all platforms. However, certain operations involving the use of resources that need to be properly closed should not be interrupted at arbitrary points. The structured exception handling mechanism provides the basic means for controlling *what* is done in response to an interrupt, but additional means are needed to control exactly *when* the interrupt can be delivered and when it must be deferred until a critical operation is complete. Parallel computations as discussed in Section C.2.1 provide a good example: If the user on a master process interrupts a parallel computation, then the interrupt must be caught, propagated to the worker nodes to ensure that these are properly terminated, and only then continue and terminate the computation on the master. Research on asynchronous exception handling in other languages, in particular Haskell [30], will provide valuable guidance to possible approaches in the R context. An experimental mechanism for controlling when interrupts may occur has been implemented but further testing and refinement is needed.

Once these issues have been addressed an important next step is to preparate a document describing the effective use of powerful exception handling in a range of statistical applications. This will encourage use of the approach and also help identify weaknesses. This document, to be developed with Robert Gentleman, will be submitted for publication in a statistical computing journal.

Graphical User Interfaces

A recurring theme in discussions about R is graphical user interfaces. This can have a number of different meanings, such as: a limited interface designed to bring a particular analysis to a

specialized audience; a framework to make a wide range of basic procedures more easily available to a broad audience; or a means for expressing a rich set of computations through a graphical language. All are challenging problems. From a system design perspective, a common theme is that they require a structure that allows user input through user interface events from a user interface toolkit to work seamlessly with executing R language code. Since the ability to display graphics with a limited amount of interaction is a basic component of the simplest R session, this involves the interplay of two user interface frameworks. This is a major challenge and is in large part responsible for the difficulty in experimenting with alternative graphical interfaces to R. Some preliminary work to develop an event handling infrastructure that can allow multiple toolkits to cooperate within a single R session has been carried out. If time permits, a prototype implementation will be constructed and used to implement a set of example interfaces. Work in this area will most likely be joint with Duncan Temple Lang and Simon Urbanek.

Memory Management and Serialization Issues

With increasing use of R in cooperation with other systems that also use memory management mechanisms it is becoming more important to have ways of integrating R's memory management system with other systems. External references and weak pointers based on the design of a similar system for Haskell [37] have been added to R. There is a strong need to for more extensive documentation of these mechanisms and to review and revise the mechanism in light of experience to date. For example, the need to be able to customize the serialization mechanism for external pointers when these are transmitted to other applications over a network or stored on disk has recently arisen in the context of recreating Java objects used by the RWeka package [23]. A possible approach has been designed and will be implemented and tested in the near future. Once these revisions have been completed a paper for a statistical software journal will be written to provide documentation of the mechanisms and their uses. Some of this work will be in collaboration with Kurt Hornik and Simon Urbanek.

Object Oriented Programming Support

Object-oriented programming has proven to be a very valuable paradigm in all areas, including statistical computing. Exactly what is meant by object oriented programming can be somewhat unclear. Historically there have been many variations on the theme, though in recent years a common claim is that only the form of object-oriented programming available in C++ or Java is "true" object oriented programming. R provides two systems for object-oriented programming, the S3 system and the S4 system [13, 14]. Both are based on a generic function approach, a quite different model from the one used by C++ and Java. The S4 approach supports multiple inheritance, multiple dispatch, and several other advanced concepts. The design is very ambitious and very powerful. It is also often perceived as being very complex and, in some respects, rigid. It has been adopted enthusiastically by some projects, notably the BioConductor suite [1, 21], but has been avoided by others.

It may be useful to begin a re-examination of the object-oriented programming support in R. One direction to consider is whether it is time to prepare more extensive educational material on proper and effective use of the S4 approach, in particular in the context of the modularity provided by the name space mechanism. A second option is to explore alternate approaches. For example, predicate based dispatch is an approach that allows methods to be defined for objects, or sets of

objects, that satisfy a predicate. A simple example is a method for a square matrix, i.e. a matrix for which the number of rows equals the number of columns. A system supporting predicate dispatch would allow a method for a square matrix to be defined without the need to introduce a formal square matrix subclass. Some recent work in this direction is discussed in [33].

Persistent Storage and Checkpointing

Integration with persistent storage systems is another area where further work is needed. R assumes that working data is maintained in memory, but requires a form of specialized storage for preserving data across sessions. This storage mechanism is also used for managing the definitions of system functions. The mechanism used by R for serializing objects into a stream of bytes for storage has been made explicit and has been used to allow the implementation of a deferred loading mechanism for function definitions [40]. This can be viewed as an ad hoc persistent storage interface. A more formal structure in which different back end data bases can be used individually or in combination is worth exploring. A system that supports persistent storage of the state of a computation will provide a valuable mechanism for checkpointing and possibly migrating long-running computations. A careful design, integrated well with existing SQL data base access approaches, could lead to a significant improvement in both maintainability and usability. Experience with the Root system [11] may provide valuable guidance in this work. Efforts in this direction will likely be in collaboration with Brian Ripley and Duncan Temple Lang.

Applications to Brain Imaging

Collaboration with imaging researchers at the University of Iowa has resulted in one Ph.D. dissertation and was a major motivation for the work on visualization incorporated in the `misc3d` package. This collaboration is expected to continue. In addition to providing many interesting challenges and opportunities for developing new statistical methodology, such as the tissue classification method developed by Dr. Dai Feng in his thesis research, the very large data sets arising in medical image analysis provide a good framework for exploring the effectiveness of parallelization strategies and other strategies for improving the performance of R on large data sets.

C.2.4 Research Approach and Progress to Date

Exploring different approaches to the design of statistical software requires that they be implemented and tested in the context of a statistical computing environment. The R system will serve as the primary basis for the initial explorations. It is important to note that the objective of this work is not just to design a specific system but to also develop principles of system design that can be adopted by others designing very different systems. But principles can only be explored and tested in the context of a complete system with a reasonable user base. R provides such a framework.

Results of this research will initially be made available as software implementations in revisions of R and in separate R packages. After some experience has been gained from this exposure, results will be submitted for publication as articles in journals on statistical computing.

Work on extensions to the `snow` framework for parallel computing will continue throughout the project. The experimental work on parallel evaluation of vectorized mathematical functions will be folded into R and extended to functions not yet covered once some configuration issues have

been resolved, hopefully by mid 2009. Work will then be able to proceed on parallelizing other internal functions, such as row and column summaries and other basic variants of the `apply` family of functions.

A first generation experimental implementation of a compiler for R has been completed and further refined and has been made available in a limited way for testing. This design needs to be revised to allow incorporation of parallel execution instructions and to support more efficient representation of scalar data and more efficient variable access. This will hopefully be completed in time for an initial public release by late 2009. At that point experimentation with more significant optimizations and with the development of annotation mechanisms for guiding the compiler will begin; this should lead to a release of an improved compiler towards the middle of 2010. If these approaches are successful, a conversion to a pure compiled runtime may be possible by early 2011.

Several portions of the research project may benefit significantly from attending Computer Science conferences on language design and implementation. Work on several implementation aspects of the proposed research will proceed more quickly if a graduate assistant with sufficient programming skills is available to assist with some of the coding. Experience working on system implementation may also result in new design ideas that can form the basis of a Ph. D. dissertation.

C.2.5 Broader Impact of Proposed Research

Data analysis is an integral part of most if not all scientific research. Good data analysis practice, and the teaching of good data analysis practice, require the use of good software, software that allows the data analytic tasks to be matched to the needs of the basic scientific problems investigated. The proposed research contributes to this goal in a very direct way by making available its results in the form of enhancements and additions to the R statistical software system. R is already used extensively in both research and teaching, and the proposed research will lead to improvements that make it considerably more valuable for both purposes.

One area where R has become particularly valuable is bioinformatics through the BioConductor project [1, 21]. Compilation and the support of parallel computing will allow the current tool set to be applied more effectively and will aid in supporting the development of more computationally intensive techniques. Further improvements to error handling mechanisms will aid areas such as the increasingly important activity of reliably incorporating bioinformatics and other information available on the internet into statistical computations.

The longer term effect of this research will be to make contributions to the principles of statistical software design that are applicable to all forms of statistical software. This will help enhance productivity of research and improve instruction that use data analysis tools of any kind.

D References Cited

[1] BioConductor: Software for bioinformatics. World Wide Web, URL = `http://www.bioconductor.org/`.

[2] Standard ECMA-335: Common Language Infrastructure (CLI). World Wide Web, URL = `http://www.ecma-international.org/publications/standards/Ecma-335.htm`.

[3] Psyco. World Wide Web, URL=`http://psyco.sourceforge.net/`, 2008.

[4] PyPy. World Wide Web, URL=`http://codespeak.net/pypy/dist/pypy/doc/home.html`, 2008.

[5] Andrew W. Appel. *Compiling with Continuations*. Cambridge University Press, 1992.

[6] Andrew W. Appel. *Modern Compiler Implementation in C: Basic Techniques*. Cambridge University Press, 1997.

[7] Joe Armstrong. *Programming Erlang: Software for a Concurrent World*. Pragmatic Bookshelf, 2007.

[8] K. Asanovic, R. Bodik, B. C. Catanzaro, J. J. Gebis, P. Husbands, K Keutzer, D. A. Patterson, L. W. Plishker, J. Shalf, S. W. Williams, and K. A Yelick. The landscape of parallel computing research: a view from Berkeley. Technical Report UCB/EECS-2006-183, EECS Department, University of California, Berkeley, 2006.

[9] Rob H. Bisseling. *Parallel Scientific Computation: A Structured Approach using BSP and MPI*. Oxford, 2004.

[10] Guy E. Blelloch, Siddhartha Chatterjee, Jonathan C. Hardwick, Jay Sipelstein, and Marco Zagha. Implementation of a portable nested data-parallel language. *Journal of Parallel and Distributed Computing*, 21(1):4–14, April 1994.

[11] Rene Brun and Fons Rademakers. The ROOT system home page. World Wide Web, URL=`http://root.cern.ch/`, 2008.

[12] Timothy Budd. *An APL Compiler*. Springer-Verlag, 1988.

[13] John M. Chambers. *Programming with Data: A Guide to the S Language*. Springer, 1998.

[14] John M. Chambers. *Software for Data Analysis: Programming with R*. Springer-Verlag, 2008.

[15] R. Chandra, R. Menon, and D. Dagum, L. Kohr. *Parallel Programming in OpenMP*. Morgan Kaufmann, San Francisco, 2000.

[16] J. Dean and S. Ghemawat. Mapreduce: Simplified data processing on large clusters. *Operating Systems Design and Implementation*, pages 137–149, 2004.

[17] L. Antony Drummond, Vicente Galiano, Violeta Migallón, and Jose Penadés. PyACTS: A Python based interface to ACTS tools and parallel scientific applications. *International Journal of Parallel Programming*, to appear.

[18] John Garvin. RCC. World Wide Web, URL=`http://hipersoft.cs.rice.edu/rcc/`.

[19] John Garvin. RCC: A compiler for the R language for statistical computing. Master's thesis, Rice University, 2004.

[20] A. Geist, A. Beguelin, J. Dongarra, W. Jiang, R. Manchek, and V. Sunderam. *PVM: Parallel Virtual Machine. A User's Guide and Tutorial for Networked Parallel Computing*. MIT Press, 1994.

[21] R. Gentleman, V. Carey, D. Bates, B. Bolstad, M. Dettling, S. Dudoit, B. Ellis, L. Gautier, Y. Ge, J. Gentry, K. Hornik, T. Hothorn, W. Huber, S. Iacus, R. Irizarry, F. Leisch, C. Li, M. Maechler, A. J. Rossini, G. Sawitzki, C. Smith, G. K. Smyth, L. Tierney, J. Y. H. Yang, and J. Zhang. Bioconductor: Open software development for computational biology and bioinformatics. *Genome Biology*, 5(10), 2004.

[22] Konrad Hinsen, Hans Petter Langtangen, Ola Skavhaug, and Åsmund Ødegård. Using BSP and Python to simplify parallel programming. *Future Gener. Comput. Syst.*, 22(1):123–157, 2006.

[23] Kurt Hornik. *`RWeka`: R/Weka Interface*, 2008.

[24] David Hovemeyer and William Pugh. Finding bugs is easy. In *OOPSLA '04: Companion to the 19th annual ACM SIGPLAN conference on Object-oriented programming systems, languages, and applications*, pages 132–136, New York, NY, USA, 2004. ACM Press.

[25] R. Ihaka and D. Temple Lang. Back to the future: Lisp as a base for a statistical computing system. In Paula Brito, editor, *COMPSTAT 2008: Proceedings in Computational Statistics*, pages 21–34. Physica Verlag, 2008.

[26] Ross Ihaka and Robert Gentleman. R: A language for data analysis and graphics. *Journal of Computational and Graphical Statistics*, 5:299–314, 1996.

[27] David Kranz, Richard Kelsey, Jonathan Rees, Paul Hudak, James Philbin, and Norman Adams. ORBIT: An optimizing compiler for scheme. *ACM SIGPLAN Notices*, pages 219–233, 1986.

[28] Ted Kremenek, Ken Ashcraft, Junfeng Yang, and Dawson Engler. Correlation exploitation in error ranking. In *SIGSOFT '04/FSE-12: Proceedings of the 12th ACM SIGSOFT twelfth international symposium on Foundations of software engineering*, pages 83–93, New York, NY, USA, 2004. ACM Press.

[29] F. Loulergue, R. Benheddi, F. Gava, and D. Louis-Régis. Bulk synchronous parallel ML: Semantics and implementation of the parallel juxtaposition. In Dima Grigoriev, John Harrison, and Edward A. Hirsch, editors, *Computer Science - Theory and Applications, First International Computer Science Symposium in Russia, CSR 2006, St. Petersburg, Russia, June 8-12, 2006, Proceedings*, volume 3967 of *Lecture Notes in Computer Science*, pages 475–486. Springer, 2006.

[30] Simon Marlow, Simon Peyton Jones, Andrew Moran, and John Reppy. Asynchronous exceptions in Haskell. In *PLDI '01: Proceedings of the ACM SIGPLAN 2001 conference on Programming language design and implementation*, pages 274–285, New York, NY, USA, 2001. ACM Press.

[31] A. Merlin, G. Hains, and F. Loulergue. An SPMD environment machine for functional BSP programs. In *SFP'2001 Scottish Functional Programming Workshop*, Stirling, August 2001.

[32] Stephen Milborrow. `jit`: Just-in-Time Compiler for the R Language. World Wide Web, URL=`http://www.milbo.users.sonic.net/ra`.

[33] Todd Millstein. Practical predicate dispatch. In *OOPSLA 2004: Proceedings of the 2004 ACM Conference on Object-Oriented Programming, Languages, and Applications*, pages 354–364. ACM, 2004.

[34] The message passing interface (MPI) standard. World Wide Web, URL = `http://www-unix.mcs.anl.gov/mpi/`.

[35] F Pérez and B. Granger. IPython: A system for interactive scientific computing. *Computing in Science and Engineering*, 9(3):21–29, 2007.

[36] Simon L. Peyton Jones, Cordy Hall, Kevin Hammond, Will Partain, and Phil Wadler. The Glasgow Haskell compiler: a technical overview. In *Proceedings of the UK Joint Framework for Information Technology (JFIT) Thechical Conference*, Keele, UK, 1993.

[37] Simon L. Peyton Jones, Simon Marlow, and Conal Elliott. Stretching the storage manager: Weak pointers and stable names in Haskell. In *Implementation of Functional Languages*, pages 37–58, 1999.

[38] Kent M. Pitman. *Condition Handling in the Lisp Language Family*, volume 2022 of *Springer Lecture Notes in Computer Science*, pages 39–58. Springer-Verlag, 2001.

[39] REvolution Computing. *NetWorkSpaces for R*, 2008. R package version 1.6.3.

[40] Brian D. Ripley. Lazy loading and packages in R 2.0.0. *R News*, 4(2):2–4, September 2004.

[41] A. J. Rossini, L. Tierney, and N. Li. Simple parallel statistical computing in R. *Journal of Computational and Graphical Statistics*, 16(1):399–420, 2007.

[42] Andrew Shalit, David Moon, and Orca Starbuck. *The Dylan Reference Manual: The Definitive Guide to the New Object-Oriented Dynamic Language*. Addison-Wesley, 1996.

[43] Gaurav Sharma and Jos Martin. MATLAB: A language for parallel computing. *International Journal of Parallel Programming*, to appear.

[44] The MathWorks. MATLAB. World Wide Web, URL = `http://www.mathworks.com/`.

[45] L. Tierney. Compiling R: A preliminary report. In *Proceedings of the 2nd International Workshop on Distributed Statistical Computing*, 2001.

[46] L. Tierney. Implicit and explicit parallel computing in R. In Paula Brito, editor, *COMPSTAT 2008: Proceedings in Computational Statistics*, pages 43–51. Physica Verlag, 2008.

[47] Luke Tierney. *LISP-STAT: An Object-Oriented Environment for Statistical Computing and Dynamic Graphics*. Wiley, NewYork, NY, 1990.

[48] Luke Tierney, Anthony J. Rossini, and Na Li. `snow`: A parallel computing framework for the R system. *International Journal of Parallel Programming*, to appear.

[49] R. C. Whaley and A. Petitet. Minimizing development and maintenance costs in supporting persistently optimized BLAS. *Software: Practice and Experience*, 35(2):101–121, 2005.

E Biographical Sketch for Luke Tierney

Biographical Data

Mailing Address

Luke Tierney, Department of Statistics and Actuarial Science, University of Iowa, 241 Schaeffer Hall, Iowa City, IA 52242.

Date/Place of Birth

October 26, 1954 New York City, NY

Education

Ph. D. 1980, Operations Research, Cornell University, Ithaca, NY.

M. A. 1979, Operations Research, Cornell University, Ithaca, NY.

B. A., M. A. 1977, Mathematical Sciences, Johns Hopkins University, Baltimore, Md.

Employment

Chair, Department of Statistics and Actuarial Science, University of Iowa, since August 2004.

Professor, Department of Statistics and Actuarial Science, University of Iowa, August 2002.

Professor, School of Statistics, University of Minnesota, September 1991.

Associate Professor, School of Statistics, University of Minnesota, September 1985.

Assistant Professor, School of Statistics, University of Minnesota, January 1984.

Assistant Professor, Department of Statistics, Carnegie-Mellon University, September 1980 - December 1983.

Publications

Five Related Publications

- Tierney, L. *LISP-STAT: An Object-Oriented Environment for Statistical Computing and Dynamic Graphics*, New York: Wiley, 1990.
- Tierney, L. "Name Space Management for R," *R News: The Newsletter of the R Project 3(1)*, 2–6, 2003.
- Rossini, A. J., Tierney, L., and Li, N., "Simple parallel statistical computing in R," *Journal of Computational and Graphical Statistics*, 16 (1), 399–420, 2007.
- Tierney, L., "Implicit and explicit parallel computing in R," in "COMPSTAT 2008: Proceedings in Computational Statistics," Paula Brito (ed.), Physica Verlag, 43–51, 2008.
- Tierney, L., Rossini, A. J., and Li, N., "`snow`: A parallel computing framework for the R system," *International Journal of Parallel Programming*, to appear.

Five Other Publications

- Tierney, L. "Markov chains for exploring posterior distributions," with discussion, *Ann. Statist.* **22**, 1701–1762, 1994.
- Tierney, L. and Mira, A., "Some Adaptive Monte Carlo Methods for Bayesian Inference," *Statistics in Medicine 18*, 2507–2515, 1999.
- Mira, A. and Tierney, L. "On the use of auxiliary variables in Markov chain Monte Carlo sampling," *Scandinavian Journal of Statistics 29*, 1–12, 2002.
- R. Gentleman, V. Carey, D. Bates, B. Bolstad, M. Dettling, S. Dudoit, B. Ellis, L. Gautier, Y. Ge, J. Gentry, K. Hornik, T. Hothorn, W. Huber, S. Iacus, R. Irizarry, F. Leisch, C. Li, M. Maechler, A. J. Rossini, G. Sawitzki, C. Smith, G. K. Smyth, L. Tierney, J. Y. H. Yang, J. Zhang, "Bioconductor: Open Software Development for Computational Biology and Bioinformatics," *Genome Biology 5 (10)*, 2004.
- Feng, D., and Tierney, L., "Computing and Displaying Isosurfaces in R," *Journal of Statistical Software*, 28 (1), 1–24, 2008.

Collaborators During Past 48 Months

D. M. Bates (U. of Wisconsin), J. M. Chambers (Lucent), R. Gentleman (U. of Washington), M. Haran (Penn State U.), K. Hornik (WU VIenna), J. Huang (U. of Iowa) S. Iacus (U. of Milan, Italy), R. Ihaka (Auckland U., New Zealand), F. Leisch (TU VIenna), N. Li (U. of Minnesota), S. Ma (Yale U.) M. Maechler (ETH Zurich, Switzerland), R. Pietrobon (Duke U.), A. Rossini (Novartis), H. Sevcikova (U. of Washington) D. Temple Lang (UC Davis).

Theses Supervised

Synian Hwang, Ph. D. Thesis, Carnegie-Mellon University, August 1984 (supervised jointly with S. Fienberg). Affiliation: State of New York.

YuTing Cheng, Ph. D. Thesis, University of Minnesota, October 1996. Affiliation: Cheng Chi University, Taiwan.

Antonietta Mira, Ph. D. Thesis, University of Minnesota, October 1998. Affiliation: Insubria University, Varese, Italy.

Murali Haran, Ph. D. Thesis, University of Minnesota, May 2003 (supervised jointly with B. Carlin). Affiliation: Pennsylvania State University.

Day Feng, Ph. D. Thesis, University of Iowa, May 2008. Affiliation: Merck & Co, New Jersey.

Jung-Eun Song, Ph. D. Thesis, University of Iowa, June 2008, Affiliation: Electrical Geodesics, Eugene, OR.

Thesis Advisor

Howard M. Taylor (U. of Delaware, retired).

SUMMARY PROPOSAL BUDGET

YEAR 1

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

ORGANIZATION: **University of Iowa**

PRINCIPAL INVESTIGATOR / PROJECT DIRECTOR: **Luke-jon Tierney**

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. **Luke-jon Tierney - none**	0.00	0.00	2.00	$ **33,899**	$
2.					
3.					
4.					
5.					
6. (**0**) OTHERS (LIST INDIVIDUALLY ON BUDGET JUSTIFICATION PAGE)	0.00	0.00	0.00	**0**	
7. (**1**) TOTAL SENIOR PERSONNEL (1 - 6)	0.00	0.00	2.00	**33,899**	
B. OTHER PERSONNEL (SHOW NUMBERS IN BRACKETS)					
1. (**0**) POST DOCTORAL SCHOLARS	0.00	0.00	0.00	**0**	
2. (**0**) OTHER PROFESSIONALS (TECHNICIAN, PROGRAMMER, ETC.)	0.00	0.00	0.00	**0**	
3. (**0**) GRADUATE STUDENTS				**20,258**	
4. (**0**) UNDERGRADUATE STUDENTS				**0**	
5. (**0**) SECRETARIAL - CLERICAL (IF CHARGED DIRECTLY)				**0**	
6. (**0**) OTHER				**0**	
TOTAL SALARIES AND WAGES (A + B)				**54,157**	
C. FRINGE BENEFITS (IF CHARGED AS DIRECT COSTS)				**13,748**	
TOTAL SALARIES, WAGES AND FRINGE BENEFITS (A + B + C)				**67,905**	
D. EQUIPMENT (LIST ITEM AND DOLLAR AMOUNT FOR EACH ITEM EXCEEDING $5,000.)					
HP 9400 Workstation $ **8,654**					
TOTAL EQUIPMENT				**8,654**	
E. TRAVEL 1. DOMESTIC (INCL. CANADA, MEXICO AND U.S. POSSESSIONS)				**2,500**	
2. FOREIGN				**2,500**	
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				**2,000**	
2. PUBLICATION COSTS/DOCUMENTATION/DISSEMINATION				**2,000**	
3. CONSULTANT SERVICES				**0**	
4. COMPUTER SERVICES				**2,400**	
5. SUBAWARDS				**0**	
6. OTHER				**4,954**	
TOTAL OTHER DIRECT COSTS				**11,354**	
H. TOTAL DIRECT COSTS (A THROUGH G)				**92,913**	
I. INDIRECT COSTS (F&A)(SPECIFY RATE AND BASE)					
Indirect cost (base excludes equipment and stipend) (Rate: 50.0000, Base: 79305)					
TOTAL INDIRECT COSTS (F&A)				**39,653**	
J. TOTAL DIRECT AND INDIRECT COSTS (H + I)				**132,566**	
K. RESIDUAL FUNDS				**0**	
L. AMOUNT OF THIS REQUEST (J) OR (J MINUS K)				$ **132,566**	$

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

PI/PD NAME: **Luke-jon Tierney**

ORG. REP. NAME*: **John Massa**

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

YEAR 2

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

ORGANIZATION
University of Iowa

PRINCIPAL INVESTIGATOR / PROJECT DIRECTOR
Luke-jon Tierney

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. **Luke-jon Tierney - none**	0.00	0.00	2.00	$ **34,916**	$
2.					
3.					
4.					
5.					
6. (**0**) OTHERS (LIST INDIVIDUALLY ON BUDGET JUSTIFICATION PAGE)	0.00	0.00	0.00	**0**	
7. (**1**) TOTAL SENIOR PERSONNEL (1 - 6)	0.00	0.00	2.00	**34,916**	
B. OTHER PERSONNEL (SHOW NUMBERS IN BRACKETS)					
1. (**0**) POST DOCTORAL SCHOLARS	0.00	0.00	0.00	**0**	
2. (**0**) OTHER PROFESSIONALS (TECHNICIAN, PROGRAMMER, ETC.)	0.00	0.00	0.00	**0**	
3. (**0**) GRADUATE STUDENTS				**20,866**	
4. (**0**) UNDERGRADUATE STUDENTS				**0**	
5. (**0**) SECRETARIAL - CLERICAL (IF CHARGED DIRECTLY)				**0**	
6. (**0**) OTHER				**0**	
TOTAL SALARIES AND WAGES (A + B)				**55,782**	
C. FRINGE BENEFITS (IF CHARGED AS DIRECT COSTS)				**14,593**	
TOTAL SALARIES, WAGES AND FRINGE BENEFITS (A + B + C)				**70,375**	
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)				**2,500**	
2. FOREIGN				**2,500**	
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				**2,000**	
2. PUBLICATION COSTS/DOCUMENTATION/DISSEMINATION				**2,000**	
3. CONSULTANT SERVICES				**0**	
4. COMPUTER SERVICES				**2,400**	
5. SUBAWARDS				**0**	
6. OTHER				**4,954**	
TOTAL OTHER DIRECT COSTS				**11,354**	
H. TOTAL DIRECT COSTS (A THROUGH G)				**86,729**	
I. INDIRECT COSTS (F&A)(SPECIFY RATE AND BASE) **Indirect costs (base excludes tuition stipend) (Rate: 50.0000, Base: 81775)**					
TOTAL INDIRECT COSTS (F&A)				**40,888**	
J. TOTAL DIRECT AND INDIRECT COSTS (H + I)				**127,617**	
K. RESIDUAL FUNDS				**0**	
L. AMOUNT OF THIS REQUEST (J) OR (J MINUS K)				$ **127,617**	$

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

PI/PD NAME
Luke-jon Tierney

ORG. REP. NAME*
John Massa

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

YEAR 3

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

ORGANIZATION
University of Iowa

PRINCIPAL INVESTIGATOR / PROJECT DIRECTOR
Luke-jon Tierney

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. **Luke-jon Tierney - none**	0.00	0.00	2.00	$ **35,963**	$
2.					
3.					
4.					
5.					
6. (**0**) OTHERS (LIST INDIVIDUALLY ON BUDGET JUSTIFICATION PAGE)	0.00	0.00	0.00	**0**	
7. (**1**) TOTAL SENIOR PERSONNEL (1 - 6)	0.00	0.00	2.00	**35,963**	
B. OTHER PERSONNEL (SHOW NUMBERS IN BRACKETS)					
1. (**0**) POST DOCTORAL SCHOLARS	0.00	0.00	0.00	**0**	
2. (**0**) OTHER PROFESSIONALS (TECHNICIAN, PROGRAMMER, ETC.)	0.00	0.00	0.00	**0**	
3. (**0**) GRADUATE STUDENTS				**21,492**	
4. (**0**) UNDERGRADUATE STUDENTS				**0**	
5. (**0**) SECRETARIAL - CLERICAL (IF CHARGED DIRECTLY)				**0**	
6. (**0**) OTHER				**0**	
TOTAL SALARIES AND WAGES (A + B)				**57,455**	
C. FRINGE BENEFITS (IF CHARGED AS DIRECT COSTS)				**15,030**	
TOTAL SALARIES, WAGES AND FRINGE BENEFITS (A + B + C)				**72,485**	
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)				**2,500**	
2. FOREIGN				**2,500**	
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				**2,000**	
2. PUBLICATION COSTS/DOCUMENTATION/DISSEMINATION				**2,000**	
3. CONSULTANT SERVICES				**0**	
4. COMPUTER SERVICES				**2,400**	
5. SUBAWARDS				**0**	
6. OTHER				**4,954**	
TOTAL OTHER DIRECT COSTS				**11,354**	
H. TOTAL DIRECT COSTS (A THROUGH G)				**88,839**	
I. INDIRECT COSTS (F&A)(SPECIFY RATE AND BASE) **Indirect costs (base excludes tuition stipend) (Rate: 50.0000, Base: 83885)**					
TOTAL INDIRECT COSTS (F&A)				**41,943**	
J. TOTAL DIRECT AND INDIRECT COSTS (H + I)				**130,782**	
K. RESIDUAL FUNDS				**0**	
L. AMOUNT OF THIS REQUEST (J) OR (J MINUS K)				$ **130,782**	$

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

PI/PD NAME
Luke-jon Tierney

ORG. REP. NAME*
John Massa

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 Iowa**

PRINCIPAL INVESTIGATOR / PROJECT DIRECTOR: **Luke-jon Tierney**

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. **Luke-jon Tierney - none**	0.00	0.00	6.00	$ **104,778**	$
2.					
3.					
4.					
5.					
6. ( ) OTHERS (LIST INDIVIDUALLY ON BUDGET JUSTIFICATION PAGE)	0.00	0.00	0.00	**0**	
7. (**1**) TOTAL SENIOR PERSONNEL (1 - 6)	0.00	0.00	6.00	**104,778**	
B. OTHER PERSONNEL (SHOW NUMBERS IN BRACKETS)					
1. (**0**) POST DOCTORAL SCHOLARS	0.00	0.00	0.00	**0**	
2. (**0**) OTHER PROFESSIONALS (TECHNICIAN, PROGRAMMER, ETC.)	0.00	0.00	0.00	**0**	
3. (**0**) GRADUATE STUDENTS				**62,616**	
4. (**0**) UNDERGRADUATE STUDENTS				**0**	
5. (**0**) SECRETARIAL - CLERICAL (IF CHARGED DIRECTLY)				**0**	
6. (**0**) OTHER				**0**	
TOTAL SALARIES AND WAGES (A + B)				**167,394**	
C. FRINGE BENEFITS (IF CHARGED AS DIRECT COSTS)				**43,371**	
TOTAL SALARIES, WAGES AND FRINGE BENEFITS (A + B + C)				**210,765**	
D. EQUIPMENT (LIST ITEM AND DOLLAR AMOUNT FOR EACH ITEM EXCEEDING $5,000.) $ **8,654**					
TOTAL EQUIPMENT				**8,654**	
E. TRAVEL 1. DOMESTIC (INCL. CANADA, MEXICO AND U.S. POSSESSIONS)				**7,500**	
2. FOREIGN				**7,500**	
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				**6,000**	
2. PUBLICATION COSTS/DOCUMENTATION/DISSEMINATION				**6,000**	
3. CONSULTANT SERVICES				**0**	
4. COMPUTER SERVICES				**7,200**	
5. SUBAWARDS				**0**	
6. OTHER				**14,862**	
TOTAL OTHER DIRECT COSTS				**34,062**	
H. TOTAL DIRECT COSTS (A THROUGH G)				**268,481**	
I. INDIRECT COSTS (F&A)(SPECIFY RATE AND BASE)					
TOTAL INDIRECT COSTS (F&A)				**122,484**	
J. TOTAL DIRECT AND INDIRECT COSTS (H + I)				**390,965**	
K. RESIDUAL FUNDS				**0**	
L. AMOUNT OF THIS REQUEST (J) OR (J MINUS K)				$ **390,965**	$

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

PI/PD NAME: **Luke-jon Tierney**

ORG. REP. NAME*: **John Massa**

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

C ***ELECTRONIC SIGNATURES REQUIRED FOR REVISED BUDGET**

Budget Justification

Domestic Travel

To support the proposed research it would be helpful to be able to attend one national statistical meeting and one computer science meeting each year.

Foreign Travel

To support the proposed research and present results of the research it would be helpful to be able to attend one international statistics or computer science meeting each year, and to be able to meet with international collaborators on the R project at least once every two years.

Workstation

The proposal includes a request for $8,654 for a dual quad-core processor HP 9400 AMD `x86_64` Linux workstation with 32 GB of memory. This is needed to provide the processing power required for the project, to provide a test bed for parallel and threaded computation, and for exploring the impact of using very large data sets as arise, for example, in neuroimaging applications.

Materials and Supplies

Materials and supplies required by this project include books and manuals, computer science journals, and computer software (in particular statistical software systems and compilers).

Graduate Student Support

A graduate assistant will help with programming tasks, and experience with such tasks may lead to a dissertation in this area. I have not yet identified a particular individual; I hope to do so during the coming academic year. Requested support is for a graduate RA stipend for 12 months and a tuition fellowship, in accordance with the University of Iowa gradate employee contract.

Computer Services

The Computer Support Group provides storage space, backup, and software support for the workstations that will be used in this project.

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: Luke-jon Tierney	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: Computing Environments for Statistics

Source of Support: NSF
Total Award Amount: $ 295,663 Total Award Period Covered: 06/01/06 - 05/31/09
Location of Project: University of Iowa
Person-Months Per Year Committed to the Project. Cal: 0.00 Acad: 0.00 Sumr: 2.00

Support: ☒ Current ☐ Pending ☐ Submission Planned in Near Future ☐ *Transfer of Support
Project/Proposal Title: Regularized Classification & Survival Analysis for Expression Profiling of Cancer

Source of Support: NIH
Total Award Amount: $ 1,161,536 Total Award Period Covered: 01/01/08 - 12/31/11
Location of Project: University of Iowa (PI: J. Huang)
Person-Months Per Year Committed to the Project. Cal: 0.90 Acad: 0.00 Sumr: 0.00

Support: ☒ Current ☐ Pending ☐ Submission Planned in Near Future ☐ *Transfer of Support
Project/Proposal Title: Travel Support for the Directions in Statistical Computing (DSC) 2007 Meeting

Source of Support: NSF
Total Award Amount: $ 10,000 Total Award Period Covered: 03/01/07 - 02/28/09
Location of Project: University of Iowa
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: ParallelR: Parallel Computing for Biomedical Research

Source of Support: NIH
Total Award Amount: $ 42,462 Total Award Period Covered: 12/01/08 - 11/30/11
Location of Project: Duke University (PI: R. Pietrobon)
Person-Months Per Year Committed to the Project. Cal: 0.45 Acad: 0.00 Sumr: 0.00

Support: ☐ Current ☐ Pending ☒ Submission Planned in Near Future ☐ *Transfer of Support
Project/Proposal Title: Travel Support for the Directions in Statistical Computing (DSC) 2009 Meeting

Source of Support: NSF
Total Award Amount: $ 15,000 Total Award Period Covered: 07/01/09 - 06/30/11
Location of Project: University of Iowa
Person-Months Per Year Committed to the Project. Cal: 0.00 Acad: 0.00 Summ: 0.00

*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: Luke-jon Tierney

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: Computing Environments for Statistics

Source of Support: NSF
Total Award Amount: $ 390,965 Total Award Period Covered: 06/01/09 - 05/31/12
Location of Project: University of Iowa
Person-Months Per Year Committed to the Project. Cal: 0.00 Acad: 0.00 Sumr: 2.00

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: 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: 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: Available equipment: Approx. 30 Linux x86_64 workstations, 1 Mac OS X workstation, and a Beowulf cluster of 21 quad-core Linux x86_64 nodes. File server, backup, and software support provided by Division of Mathematical Sciences Computer Support Group.

Office: Shaeffer Hall 209; standard office furnishings.

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.

FACILITIES, EQUIPMENT & OTHER RESOURCES

Continuation Page:

COMPUTER FACILITIES (continued):