Preview of Award 0916027 - Annual Project Report

Cover

Federal Agency and Organization Element to Which Report is Submitted:	4900
Federal Grant or Other Identifying Number Assigned by Agency:	0916027
Project Title:	III: Small: RIOT: Statistical Computing with Efficient, Transparent I/O
PD/PI Name:	Jun Yang, Principal Investigator
Submitting Official (if other than PD\PI):	Jun Yang Principal Investigator
Submission Date:	07/22/2013
Recipient Organization:	Duke University
Project/Grant Period:	09/01/2009 - 08/31/2014
Reporting Period:	09/01/2012 - 08/31/2013
Signature of Submitting Official (signature shall be submitted in accordance with agency specific instructions)	Jun Yang

Accomplishments

* What are the major goals of the project?

Recent technological advances have enabled collection of massive amounts of data in science, commerce, and society. These large datasets have brought us closer than ever before to solving important problems such as decoding human genomes and coping with climate changes. Meanwhile, the exponential growth in the amount of data has created an urgent challenge. Today, much of advanced analysis is done with programs custom-developed by statisticians. Unfortunately, progress has been hindered by the lack of easy-to-use statistical computing environments that scale to large datasets. Many existing tools assume that datasets fit in main memory; when applied to large datasets, they are unacceptably slow because of excessive disk input/output (I/O) operations. There have been many approaches toward I/O-efficiency, but none has gained traction with the statistical computing community. Disk-based storage engines and I/O-efficient function libraries provide only a partial solution, because many sources of I/O-inefficiency in a program remain at a higher, inter-operation level: e.g., how large intermediate results are passed between operations, how much performance can be gained by deferring and reordering operations, etc. Database systems seem to be a natural solution, with I/O-efficiency and a high-level language (SQL) enabling many high-level optimizations. However, work in integrating databases and statistical computing has mostly remained database-centric, forcing statisticians to learn unfamiliar languages and deal with their impedance mismatch with the host language.

To make a practical impact on the statistical computing community, the

RIOT project seeks to extend R---an open-source statistical computing environment widely used by statisticians---to transparently provide scalability over large datasets. Transparency means no SQL, or any new language to learn. Transparency means that existing code should run without modification, and automatically gain efficiency. RIOT is developing an end-to-end solution that addresses issues on all fronts: I/O-efficient and parallel algorithms, deferred evaluation, pipelined execution, cost-driven optimization, smart storage and materialization options, and seamless integration with database systems and the interpreted host language.

Over the course of the project, we have also expanded the scope of RIOT to consider issues such as extending database systems to support matrix storage and computation, parallelization using GPU and cloud, and leveraging SSDs for better performance. Description of our work and contributions in previous years can be found in older project reports; this report focuses on the progress made in Year 4.

*** What was accomplished under these goals (you must provide information for at least one of the 4 categories below)?**

Major Activities:

In Year 4 (2012-2013) of the project, we have worked on the following specific tasks.

A) Parallelization with Cloud. A main motivation behind the RIOT project has been improving the usability of statistical computing platforms. The rise of cloud computing in recent years, exemplified by the popularity of services such as Amazon EC2, offers a promising possibility for supporting big-data analytics. Its "pay-as-you-go" business model is especially attractive: users gain on-demand access to computing resources while avoiding hardware acquisition and maintenance costs. Support for data-parallel computing in the cloud provides a natural opportunity to scale up beyond I/O-efficiency.

However, it remains frustratingly difficult for many scientists and statisticians to use the cloud for any nontrivial statistical analysis of big data. First, developing efficient statistical computing programs requires a great deal of expertise and effort. Popular cloud programming platforms, such as Hadoop, require users to code and think in low-level, platform-specific ways, and, in many cases, resort to extensive manual tuning to achieve acceptable performance. Second, deploying such programs in the cloud is hard. Users are faced with a maddening array of choices, ranging from resource provisioning (e.g., type and number of machines to request on Amazon EC2), software configuration (e.g., number of parallel execution slots per machine for Hadoop), to execution parameters and implementation alternatives. Some of these choices can be critical to meeting deadlines and staying within budget, but current systems offer little help to users in making such choices.

In Year 4, we developed a system called Cumulon, as a natural extension of RIOT to the cloud, aimed at helping users rapidly develop and intelligently deploy matrix-based big-data analysis programs in the cloud. Cumulon features a extensible execution model and new operators especially suited for such workloads. We have implemented Cumulon on top of Hadoop/HDFS while avoiding limitations of MapReduce,

and demonstrated Cumulon's performance advantages over existing Hadoop-based systems for statistical data analysis. To support intelligent deployment in the cloud according to time/budget constraints, Cumulon goes beyond database-style optimization to make choices automatically on not only physical operators and their parameters, but also hardware provisioning and configuration settings. We apply a suite of benchmarking, simulation, modeling, and search techniques to support effective cost-based optimization over this rich space of deployment plans.

B) Working with Solid-State Drives (SSDs). Solid-state drives are becoming a viable alternative to magnetic disks for many workloads. We have been studying how to use them effectively for various database and linear algebra workloads. The first problem, which we began to investigate in Year 1, is indexing. We designed, implemented, and evaluated an index structure called FD+tree and an associated concurrency control scheme called FD+FC.

In Years 3 and 4, we continued this line of work by studying the problem of permuting (and re-sorting) on SSDs. A main application of these problems in RIOT is data layout conversion (e.g., converting a matrix from column-layout to block-layout). Conventional I/O-efficient algorithms for these problems assume that disk reads and writes have equal costs, and that random accesses cost a lot more than sequential ones. However, these assumptions are not valid with SSDs. We found that the trade-off between reads and writes and fast random accesses enable interesting new algorithms for permuting on SSDs. While external merge sort has often been used for permutation, it is an overkill that fails to exploit the property of permutation fully and carries unnecessary overhead in storing and comparing keys. We proposed faster algorithms with lower memory requirements for a large, useful class of permutations. We also tackled practical challenges that traditional permutation algorithms have not dealt with, such as exploiting random block accesses more aggressively, considering the cost asymmetry between reads and writes, and handling arbitrary data dimension sizes (as opposed to perfect powers often assumed by previous work). As a result, our algorithms are faster and more broadly applicable.

C) RIOT Execution and Optimization Issues. Database queries exhibit a limited number of fairly simple data access patterns. However, linear algebra operations common in statistical computing give rise to a much larger space of complex access patterns typically expressed using nested loops. We have developed a framework that allows these access patterns to be captured---either through code analysis or user specification---in a form amenable to automatic optimization. Specifically, the first problem that we tackled with this framework is exploiting I/O sharing opportunities. Most analysis tasks consist of multiple steps, each making one or multiple passes over arrays to be analyzed and generating intermediate results. Existing database techniques fall short of solving the I/O sharing problem in this setting. A database-like, operator-based approach does not allow full-fledged inter-operator optimization: when putting operators together for co-optimization, they cannot be treated as black boxes

but need to be 'opened up' so that the optimizer can tweak their inner workings further. Our framework captures a broad range of analysis tasks expressible in nested-loop forms, represents them in a declarative way, and optimizes their I/O by identifying sharing opportunities. Experiment results show that our optimizer is capable of finding execution plans that exploit nontrivial I/O sharing opportunities with significant savings.

The next problem we considered is the joint optimization of I/O and data layouts. Our PVLDB 2012 paper did not consider array layouts as one dimension of its optimization space; instead, array layouts were assumed to be predetermined by the programmer or some other software component. However, the choice of array layouts can dramatically affect a program's overall I/O performance, and thus needs to be determined in a cost-based manner. Furthermore, the choice of array layout may influence how to share I/Os optimally. Therefore, we consider how layout choices can be co-optimized with I/O sharing to yield further I/O savings. We focus on the commonly used block-based layout (which subsumes column- and row-major layouts).

D) Educational Activities. In terms of educational activities in Year 4, the PI has continued to closely supervise and train graduate and undergraduate researchers. Risi Thonangi and Botong Huang continue to make solid progress in their PhD study. Andrew Shim is an undergraduate student who started working with the PI in Spring 2013, under the CSURF program at Duke Computer Science. The PI taught an undergraduate database course in Fall 2012, and won the David and Janet Vaughan Brooks Teaching Award in 2013.

Specific Objectives: The specific objectives for the project in Year 4 correspond to the major activities described above, including A) parallelization with cloud, B) working with solid-state drives (SSDs), C) RIOT execution and optimization issues, and D) educational activities. For detailed descriptions of these objectives, see above.

Significant Results: Below is a brief summary of the significant results obtained in Year 4:

A) Parallelization with Cloud. We have completed a prototype implementation of Cumulon on top of Hadoop and Amazon EC2, which significantly improves the ease and efficiency of development and deployment of matrix-based statistical computing workloads in the cloud.

B) Working with Solid-State Drives (SSDs). We have developed more efficient, concurrent index structures as well as permutation/resorting algorithms for SSDs.

C) RIOT Execution and Optimization Issues. We have developed a powerful framework for optimizing I/O sharing (and data layout) when computing with big matrices, which goes well beyond database-style query optimization.

See other parts of this report for more details on these results as well as where they have been published.

Key outcomes or Below is a summary of the research accomplishments to date. In Year

Other achievements: 1, we invested most of our effort in prototyping and development activities, which provided us with much experience and insights. In Year 2, we expanded into several focused problems, including building a better storage engine and designing a better execution and optimization frameworks for RIOT, extending database systems to support matrix storage and computation, parallelization using GPU and cloud, and leveraging SSDs for better performance. In Years 3 and 4, we continued to make progress on these problems, and further investigated the problem of optimizing and provisioning for RIOT workloads in cloud and that of permuting and sorting data on SSDs. Published results so far include: 1) the RIOT 'vision' paper in CIDR 2009, which also covers RIOT-DB; 2) a demonstration of a prototype for the next generation of RIOT in ICDE 2010; 3) a paper on the RIOT storage engine in PVLDB 2011; 4) a paper on optimizing I/O sharing in RIOT in PVLDB 2012; 5) a paper on efficient indexing on SSDs in CIKM 2012; 6) a paper on Cumulon, which pushes RIOT to the cloud, in SIGMOD 2013; and 7) a paper on permuting/reorganizing data on SSDs. A provisional patent has been filed (with HP Labs) on GPU-based matrix computation inside a database system. We have released the code for RIOT-DB on the project website, and plan to make additional releases in the last year of the project.

*** What opportunities for training and professional development has the project provided?**

During Year 4 of the project, Risi Thonangi and Botong Huang continue to make solid progress in their PhD study under the PI's supervision. Andrew Shim is an undergraduate student who started working with the PI in Spring 2013, under the CSURF program at Duke Computer Science.

*** How have the results been disseminated to communities of interest?**

Below is a brief summary of how results have been disseminated in Year 4:

A) Parallelization with RIOT. Our paper on Cumulon appeared in SIGMOD 2013. We have already presented Cumulon at the IBM Workshop on Big Data Analytics in June 2013.

B) Working with Solid-State Drives (SSDs). A paper describing our FD+tree index and associated concurrency control scheme appeared in CIKM 2012. A paper describing our work on permuting (and resorting) data has been accepted to PVLDB 2013.

C) RIOT Execution and Optimization Issues. Our work on exploiting I/O sharing opportunities was published in PVLDB 2012. Our work on joint optimization of I/O data layouts became part of Yi Zhang's doctoral dissertation.

D) Educational Activities. PI's course materials are posted online and open to the general public.

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

A) Parallelization with RIOT. We are currently working making Cumulon more elastic so it can take advantages of the so-called "spot instances," which are cheap but only available for limited amount of

time.

B) Working with Solid-State Drives (SSDs). We have begun working on optimizing sorting on SSDs by reducing writes. We expect to complete this study in Year 5.

D) Educational Activities. The PI will continue to supervise PhD students Thonangi and Huang, as well as undergraduate Shim, to work on the project. Also, the PI is working on designing a new course on "introduction to data" (as opposed to databases), to be offered in Spring 2014. We hope that this course will attract undergraduates both inside and outside computer science, and help them learn how to work with data in this age of "big data."

E) RIOT code release. We plan to make a code release for RIOT in Year 5, for the main purpose of facilitating future research. In addition, we will invest in making some "core" parts of the code high-quality and usable by the public. Limited by resources and manpower, however, we will not make a full, production-quality release.

Products

Journals

Yi Zhang, Kamesh Munagala, and Jun Yang (2011). Storing matrices on disk: theory and practice revisited. *Proceedings of the VLDB Endowment*. 4 (11), 1075-1086.

Status = PUBLISHED; Acknowledgment of Federal Support = Yes ; Peer Reviewed = Yes

Yi Zhang and Jun Yang (2012). Optimizing I/O for big array analytics. *Proceedings of the VLDB Endowment*. 5 (8), 764-775.

Status = PUBLISHED; Acknowledgment of Federal Support = Yes ; Peer Reviewed = Yes

Risi Thonangi and Jun Yang (2013). Permuting data on random-access block storage. *Proceedings of the VLDB Endowment*. ?? (??), ????-????.

Status = ACCEPTED; Acknowledgment of Federal Support = Yes ; Peer Reviewed = Yes

Books

Book Chapters

Thesis/Dissertations

Yi Zhang. *Transparent and efficient I/O for statistical computing*. (2013). Duke University.

Acknowledgment of Federal Support = No

Conference Papers and Presentations

Yi Zhang, Herodotos Herodotou, and Jun Yang (2009). *RIOT: I/O-efficient numerical computing without SQL*. 2009 Conference on Innovative Data Systems Research. Asilomar, California, USA.

Status = PUBLISHED; Acknowledgement of Federal Support = Yes

Yi Zhang, Weiping Zhang, and Jun Yang (2010). *I/O-efficient statistical computing with RIOT*. 2010 International Conference on Data Engineering. Long Beach, California, USA.

Status = PUBLISHED; Acknowledgement of Federal Support = Yes

Risi Thonangi, Shivnath Babu, and Jun Yang (2012). *A practical concurrent index for solid-state drives*. 2012 International Conference on Information and Knowledge Management. Maui, Hawaii, USA.

Status = PUBLISHED; Acknowledgement of Federal Support = Yes

Botong Huang, Shivnath Babu, and Jun Yang (2013). *Cumulon: optimizing statistical data analysis in the cloud*. 2013 ACM SIGMOD International Conference on Management of Data. New York City, New York, USA.

Status = PUBLISHED; Acknowledgement of Federal Support = Yes

Other Publications

Technologies or Techniques
Nothing to report.

Patents

Patent Abstract:	N/A
Patent Title:	Two-Level Chunking for Data Analytics
Patent Number:	UNITED STATES
Country:	03/16/2012
Application Date:	Submitted
Patent Status:	
Date Issued:	

Inventions

Licenses

Websites

Title:	RIOT project website
URL:	http://db.cs.duke.edu/projects/riot
Description:	

Other Products
Nothing to report.

Participants

Research Experience for Undergraduates (REU) funding

How many REU applications were received during this reporting period? 1

How many REU applicants were selected and agreed to participate during this reporting period? 1

What individuals have worked on the project?

Name	Most Senior Project Role	Nearest Person Month Worked
Risi, Thonangi	Graduate Student (research assistant)	12
Jun Yang	PD/PI	6
Botong Huang	Graduate Student (research assistant)	9

Andrew Shim	Research Experience for Undergraduates (REU) Participant	3

What other organizations have been involved as partners?

Name	Location
HP	HP Labs in Beijing, China

Have other collaborators or contacts been involved? Y

Impacts

What is the impact on the development of the principal discipline(s) of the project?

At the end of Year 4 of the project, we have made a series of solid contributions toward enabling efficient statistical analysis over massive datasets. We have built two functional prototypes, RIOT-DB and RIOT, and published in CIDR 2009, ICDE 2010, PVLDB 2011, PVLDB 2012, CIKM 2012, SIGMOD 2013, and PVLDB 2013. For detailed descriptions of these contributions, please refer to the section of this report on research and education activities.

What is the impact on other disciplines?

The PI has been part of two other interdisciplinary projects---one funded by NSF, which studies how to collect and analyze ecological data from a sensor network, and another funded by NIH, which develops analytical and modeling tools for immunology. The PI is also starting a new project on computational journalism, in collaboration with journalists and social scientists. Some of the work in this project is motivated by the problems faced in the other projects, which will in turn benefit from our research results. As RIOT matures, it will be made available to the general statistical computing community for broader impact.

What is the impact on the development of human resources?

Ph.D. students: Yi Zhang, Risi Thonangi, Botong Huang.

Undergraduate students: Weiping Zhang, Jiaqi Yan, Andrew Shim.

What is the impact on physical resources that form infrastructure?
Nothing to report.

What is the impact on institutional resources that form infrastructure?
Nothing to report.

What is the impact on information resources that form infrastructure?
Nothing to report.

What is the impact on technology transfer?

The PI received an HP Labs Innovation Research Award for work on RIOT. The work on integrating RIOT within GPU-enabled DBMS was presented at

HP Tech Con 2011, and a provisional patent has been filed.

What is the impact on society beyond science and technology?

Nothing to report.

Changes

Changes in approach and reason for change

The vision of the RIOT project is to make statistical analysis scalable on big data in a way that is easy and natural to statisticians. It started out as platform for automatically enabling I/O-efficiency to code written in R (a matrix-based, high-level language popular among statisticians and data analysts). Over the course of the project, however, our vision has become broader, as we began to recognize the potential of emerging hardware and infrastructure in enabling big-data analytics. Hence, beyond I/O-efficiency in the traditional setting of a single machine with a hard drive, we have studied how to make matrix-based statistical computing easier and more efficient on solid-state drives and data-parallel cloud computing platforms. We have made good progress on the problems that we originally set out to investigate, as well as new ones that emerged during our course of investigation. Please see the rest of this project report for details.

Actual or Anticipated problems or delays and actions or plans to resolve them

Despite the nice array of results we have obtained, our project team has actually been overworked and undermanned for the last two years. An important reason is the demanding nature of the research tasks, but another practical reason is the unfortunate budget crunch faced by my research group over the past two years. Thus, I requested a no-cost extension for this grant, which has already been approved by NSF. This extension allows the lead PhD student on the RIOT project, Risi Thonangi, to be funded during the 2013-2014 academic year. More importantly, it will give us the time needed to bring a nice conclusion to our research tasks, and to make a final release of the RIOT code. For more discussion of the ongoing and remaining tasks, please refer to other parts of this project report.

Changes that have a significant impact on expenditures

Nothing to report.

Significant changes in use or care of human subjects

Nothing to report.

Significant changes in use or care of vertebrate animals

Nothing to report.

Significant changes in use or care of biohazards

Nothing to report.