Proceedings of
IEEE 28th International
Parallel and Distributed Processing Symposium

IPDPS 2014 Advance Program Abstracts

Los Alamitos, California

Washington • Tokyo

IEEE Computer Society Order Number P5207
BMS Part Number CFP14023-PRT
ISBN 978-0-7695-5207-1

Additional copies may be ordered from:

IEEE Computer Society
Customer Service Center
10662 Los Vaqueros Circle
P.O. Box 3014
Los Alamitos, CA 90720-1314
Tel: + 1 800 272 6657
Fax: + 1 714 821 4641
http://computer.org/cspress
csbooks@computer.org

IEEE Service Center
445 Hoes Lane
P.O. Box 1331
Piscataway, NJ 08855-1331
Tel: + 1 732 981 0060
Fax: + 1 732 981 9667
http://shop.ieee.org/store/
customer-service@ieee.org

IEEE Computer Society
Asia/Pacific Office
Watanabe Bldg., 1-4-2
Minami-Aoyama
Minato-ku, Tokyo 107-0062
JAPAN
Tel: + 81 3 3408 3118
Fax: + 81 3 3408 3553
tokyo.ofc@computer.org

Individual paper REPRINTS may be ordered at: <reprints@computer.org>

Editorial production by Lisa O'Conner
Cover art production by Mark Bartosik

IEEE Computer Society
***Conference Publishing Services* (CPS)**
http://www.computer.org/cps

2014 IEEE 28th International Parallel & Distributed Processing Symposium

IPDPS 2014

Table of Contents

Session 3: GPU

Session 4: I/O, Storage, and Networking

Session 5: Multi-core Algorithms

Session 6: Computational Biology

Session 7: Interconnection Network

Session 8: System-Level Resource Management

Session 9: GPU Algorithms

Session 10: Graph and Network Processing

Session 11: Modeling, Simulation, and Reliability

Session 12: Accelerator Application Development and Optimization

Keynote Speaker 2

Session 13: Combinatorial Algorithms

Session 14: Large Scale Scientific Applications

Session 15: Multicore and Transactional Memory

Session 16: HPC Operating Systems and Runtime Systems

Session 17: Algorithms for Distributed Computing

Session 18: Milestones at the Petascale

Session 19: Storage and Reliability

Session 20: Map/Reduce and Big Data

Session 21: Network Algorithms

Keynote Speaker 3

Best Papers Session

Session 22: Performance Characterization and Optimization

Session 23: Multithreading and Concurrency

Session 24: Numerical Algorithms

Session 25: Performance Impacts of Hardware Acceleration

Session 26: Programming Models and Tools

Session 27: Algorithms for High Performance Computing

Session 28: Scalable Algorithms

Session 29: Resilience and Reliability

Keynote Speaker 1

IPDPS 2014

Scalability-Centric HPC System Design

Yutong Lu

National University of Defense Technology, China

Since scalability is one of the major challenges for advanced HPC systems in the post-petascale and exascale era, innovative integrated technology designs are needed for new architecture as well as associated software stacks. We need to explore the capability of cpu, accelerator, interconnection, I/O storage system, and till whole system. This talk will discuss the way of scalability-centric HPC system hardware and software design related to the computation, communication, data procession, and fault tolerance. The experiences on the design and implementation of Tianhe systems will also be given. Furthermore, some investigations on architecture and software design for the next generation HPC system will be presented. In general, a co-design approach should be followed throughout the research and development activities to deliver a whole system for scalable computing, to support the large-scale domain applications efficiently.

Session 1: Algorithms for Resource Management and Awareness

IPDPS 2014

Cost-Optimal Execution of Boolean Query Trees with Shared Streams

Henri Casanova[1], Lipyeow Lim[1], Yves Robert[2,3], Frédéric Vivien[2], and Dounia Zaidouni[2]

[1]University of Hawai'i at Manoa, Honolulu, USA
{henric|lipyeow}@hawaii.edu
[2]´École Normale Supérieure de Lyon & INRIA, France
{Yves.Robert|Frederic.Vivien|Dounia.Zaidouni}@ens-lyon.fr
[3]University of Tennessee, Knoxville, USA

The processing of queries expressed as trees of boolean operators applied to predicates on sensor data streams has several applications in mobile computing. Sensor data must be retrieved from the sensors, which incurs a cost, e.g., an energy expense that depletes the battery of a mobile query processing device. The objective is to determine the order in which predicates should be evaluated so as to shortcut part of the query evaluation and minimize the expected cost. This problem has been studied assuming that each data stream occurs at a single predicate. In this work we remove this assumption since it does not necessarily hold in practice. Our main results are an optimal algorithm for single-level trees and a proof of NP-completeness for DNF trees. For DNF trees, however, we show that there is an optimal predicate evaluation order that corresponds to a depth-first traversal. This result provides inspiration for a class of heuristics. We show that one of these heuristics largely outperforms other sensible heuristics, including a heuristic proposed in previous work.

It's About Time: On Optimal Virtual Network Embeddings under Temporal Flexibilities

Matthias Rost[*], *Stefan Schmid*[*]†, *and Anja Feldmann*[*]

∗Internet Network Architectures, Technische Universität Berlin, Germany
†Telekom Innovation Laboratories, Germany

Distributed applications often require high-performance networks with strict connectivity guarantees. For instance, many cloud applications suffer from today's variations of the intra-cloud bandwidth, which leads to poor and unpredictable application performance. Accordingly, we witness a trend towards virtual networks (VNets) which can provide resource isolation. Interestingly, while the problem of where to embed a VNet is fairly well-understood today, much less is known about when to optimally allocate a VNet. This however is important, as the requirements specified for a VNet do not have to be static, but can vary over time and even include certain temporal flexibilities. This paper initiates the study of the temporal VNet embedding problem (TVNEP). We propose a continuous-time mathematical programming approach to solve the TVNEP, and present and compare different algorithms. Based on these insights, we present the CSM-Model which incorporates both symmetry and state-space reductions to significantly speed up the process of computing exact solutions to the TVNEP. Based on the CSM-Model, we derive a greedy algorithm OGA to compute fast approximate solutions. In an extensive computational evaluation, we show that despite the hardness of the TVNEP, the CSM-Model is sufficiently powerful to solve moderately sized instances to optimality within one hour and under different objective functions (such as maximizing the number of embeddable VNets). We also show that the greedy algorithm exploits flexibilities well and yields good solutions. More generally, our results suggest that already little time flexibilities can improve the overall system performance significantly.

Exploiting Geometric Partitioning in Task Mapping for Parallel Computers

Mehmet Deveci, Sivasankaran Rajamanickam†, Vitus J. Leung†, Kevin Pedretti†, Stephen L. Olivier†, David P. Bunde‡, Umit V. Çatalyürek*, and Karen Devine†*

∗The Ohio State University, Columbus, Ohio
† Sandia National Laboratories, Albuquerque, New Mexico
‡Knox College, Galesburg, IL

We present a new method for mapping applications' MPI tasks to cores of a parallel computer such that communication and execution time are reduced. We consider the case of sparse node allocation within a parallel machine, where the nodes assigned to a job are not necessarily located within a contiguous block nor within close proximity to each other in the network. The goal is to assign tasks to cores so that interdependent tasks are performed by "nearby" cores, thus lowering the distance messages must travel, the amount of congestion in the network, and the overall cost of communication. Our new method applies a geometric partitioning algorithm to both the tasks and the processors, and assigns task parts to the corresponding processor parts. We show that, for the structured finite difference mini-app Mini Ghost, our mapping method reduced execution time 34% on average on 65,536 cores of a Cray XE6. In a molecular dynamics mini-app, Mini MD, our mapping method reduced communication time by 26% on average on 6144 cores. We also compare our mapping with graph-based mappings from the LibTopoMap library and show that our mappings reduced the communication time on average by 15% in MiniGhost and 10% in MiniMD.

Communication-Efficient Distributed Variance Monitoring and Outlier Detection for Multivariate Time Series

Moshe Gabel and Assaf Schuster

Department of Computer Science
Technion – Israel Institute of Technology
Haifa, Israel
{mgabel,assaf}@cs.technion.ac.il

Daniel Keren

Computer Science Department
University of Haifa
Haifa, Israel
dkeren@cs.haifa.ac.il

Modern scale-out services are comprised of thousands of individual machines, which must be continuously monitored for unexpected failures. One recent approach to monitoring is latent fault detection, an adaptive statistical framework for scale-out, load-balanced systems. By periodically measuring hundreds of performance metrics and looking for outlier machines, it attempts to detect subtle problems such as misconfigurations, bugs, and malfunctioning hardware, before they manifest as machine failures. Previous work on a large, real-world Web service has shown that many failures are indeed preceded by such latent faults. Latent fault detection is an offline framework with large bandwidth and processing requirements. Each machine must send all its measurements to a centralized location, which is prohibitive in some settings and requires data-parallel processing infrastructure. In this work we adapt the latent fault detector to provide an online, communication- and computation-reduced version. We utilize stream processing techniques to trade accuracy for communication and computation. We first describe a novel communication-efficient online distributed variance monitoring algorithm that provides a continuous estimate of the global variance within guaranteed approximation bounds. Using the variance monitor, we provide an online distributed outlier detection framework for non-stationary multivariate time series common in scale-out systems. The adapted framework reduces data size and central processing cost by processing the data in situ, making it usable in wider settings. Like the original framework, our adaptation admits different comparison functions, supports non-stationary data, and provides statistical guarantees on the rate of false positives. Simulations on logs from a production system show that we are able to reduce bandwidth by an order of magnitude, with below 1% error compared to the original algorithm.

Session 2: Big Data Processing

IPDPS 2014

MobiStreams: A Reliable Distributed Stream Processing System for Mobile Devices

Huayong Wang and Li-Shiuan Peh

MIT
Massachusetts, USA
huayongw@smart.mit.edu
peh@csail.mit.edu

Multi-core phones are now pervasive. Yet, existing applications rely predominantly on a client-server computing paradigm, using phones only as thin clients, sending sensed information via the cellular network to servers for processing. This makes the cellular network the bottleneck, limiting overall application performance. In this paper, we propose Mobi Streams, a Distributed Stream Processing System (DSPS) that runs directly on smartphones. Mobi Streams can offload computing from remote servers to local phones and thus alleviate the pressure on the cellular network. Implementing DSPS on smartphones faces significant challenges: 1) multiple phones can readily fail simultaneously, and 2) the phones' ad-hoc WiFi network has low bandwidth. Mobi Streams tackles these challenges through two new techniques: 1) token-triggered check pointing, and 2) broadcast-based check pointing. Our evaluations driven by two real world applications deployed in the US and Singapore show that migrating from a server platform to a smartphone platform eliminates the cellular network bottleneck, leading to 0.78~42.6X throughput increase and 10%~94.8% latency decrease. Also, Mobi Streams' fault tolerance scheme increases throughput by 230% and reduces latency by 40% vs. prior state-of-the-art fault-tolerant DSPSs.

MapReuse: Reusing Computation in an In-Memory MapReduce System

Devesh Tiwari

Oak Ridge National Laboratory
tiwari@ornl.gov

Yan Solihin

North Carolina State University
solihin@ncsu.edu

MapReduce programming model is being increasingly adopted for data intensive high performance computing. Recently, it has been observed that in data-intensive environment, programs are often run multiple times with either identical or slightly-changed input, which creates a significant opportunity for computation reuse. Recognizing the opportunity, researchers have proposed techniques to reuse computation in disk-based MapReduce systems such as Hadoop, but not for in-memory MapReduce (IMMR) systems such as Phoenix. In this paper, we propose a novel technique for computation reuse in IMMR systems, which we refer to as MapReuse. MapReuse detects input similarity by comparing their signatures. It skips re-computing output from a repeated portion of the input, computes output from a new portion of input, and removes output that corresponds to a deleted portion of the input. MapReuse is built on top of an existing IMMR system, leaving it largely unmodified. MapReuse significantly speeds up IMMR, even when the new input differs by 25% compared to the original input.

PAGE: A Framework for Easy PArallelization of GEnomic Applications

Mucahid Kutlu and Gagan Agrawal

Department of Computer Science and Engineering
Ohio State University
Columbus, OH, 43210
Email: kutlu@cse.ohio-state.edu
Email: agrawal@cse.ohio-state.edu

With the availability of high-throughput and low-cost sequencing technologies, an increasing amount of genetic data is becoming available to researchers. There is clearly a potential for significant new scientific and medical advances by analysis of such data, however, it is imperative to exploit parallelism and achieve effective utilization of the computing resources to be able to handle massive datasets. Thus, frameworks that can help researchers develop parallel applications without dealing with low-level details of parallel coding are very important for advances in genetic research. In this study, we develop a middleware, PAGE, which supports 'map reduce-like' processing, but with significant differences from a system like Hadoop, to be useful and effective for parallelizing analysis of genomic data. Particularly, it can work with map functions written in any language, thus allowing utilization of existing serial tools (even those for which only an executable is available) as map functions. Thus, it can greatly simplify parallel application development for scenarios where complex data formats and/or nuanced serial algorithms are involved, as is often the case for genomic data. It allows parallelization by partitioning by-locus or partitioning by-chromosome, provides different scheduling schemes, and execution models, to match the nature of algorithms common in genetic research. We have evaluated the middleware system using four popular genomic applications, including VarScan, Unified Genotyper, Realigner Target Creator, and Indel Realigner, and compared the achieved performance against with two popular frameworks (Hadoop and GATK). We show that our middleware outperforms GATK and Hadoop and it is able to achieve high parallel efficiency and scalability.

Pythia: Faster Big Data in Motion through Predictive Software-Defined Network Optimization at Runtime

Marcelo Veiga Neves and César A.F. De Rose

Pontifical Catholic University of Rio Grande do Sul
Porto Alegre, Brazil
Email: marcelo.neves@acad.pucrs.br

Kostas Katrinis

IBM Research – Ireland
Dublin, Ireland
Email: katrinisk@ie.ibm.com

Hubertus Franke

IBM TJ Watson Research Center
Yorktown Heights, NY, US
Email: frankeh@us.ibm.com

The rise of Internet of Things sensors, social networking and mobile devices has led to an explosion of available data. Gaining insights into this data has led to the area of Big Data analytics. The MapReduce framework, as implemented in Hadoop, is one of the most popular frameworks for Big Data analysis. To handle the ever-increasing data size, Hadoop is a scalable framework that allows dedicated, seemingly unbound numbers of servers to participate in the analytics process. Response time of an analytics request is an important factor for time to value/insights. While the compute and disk I/O requirements can be scaled with the number of servers, scaling the system leads to increased network traffic. Arguably, the communication-heavy phase of MapReduce contributes significantly to the overall response time, the problem is further aggravated, if communication patterns are heavily skewed, as is not uncommon in many MapReduce workloads. In this paper we present a system that reduces the skew impact by transparently predicting data communication volume at runtime and mapping the many end-to-end flows among the various processes to the underlying network, using emerging software-defined networking technologies to avoid hotspots in the network. Dependent on the network oversubscription ratio, we demonstrate reduction in job completion time between 3% and 46% for popular MapReduce benchmarks like Sort and Nutch.

Session 3: GPU

IPDPS 2014

A Case for a Flexible Scalar Unit in SIMT Architecture

Yi Yang[1], Ping Xiang[2], Michael Mantor[3], Norman Rubin[3], Lisa Hsu[3], Qunfeng Dong[4], and Huiyang Zhou[2]

[1]NEC Labs
Princeton, NJ
yyang@nec-labs.com

[2]NCSU
Raleigh, NC
{pxiang, hzhou}@ncsu.edu

[3]AMD Inc.
Orlando, FL
{Michael.Mantor, norman.rubin, Lisa.Hsu}@amd.com

[4]Dept of Central Hardware Huawei
Shanghai, China
Qunfeng.dong@ieee.org

The wide availability and the Single-Instruction Multiple-Thread (SIMT)-style programming model have made graphics processing units (GPUs) a promising choice for high performance computing. However, because of the SIMT style processing, an instruction will be executed in every thread even if the operands are identical for all the threads. To overcome this inefficiency, the AMD's latest Graphics Core Next (GCN) architecture integrates a scalar unit into a SIMT unit. In GCN, both the SIMT unit and the scalar unit share a single SIMT style instruction stream. Depending on its type, an instruction is issued to either a scalar or a SIMT unit. In this paper, we propose to extend the scalar unit so that it can either share the instruction stream with the SIMT unit or execute a separate instruction stream. The program to be executed by the scalar unit is referred to as a scalar program and its purpose is to assist SIMT-unit execution. The scalar programs are either generated from SIMT programs automatically by the compiler or manually developed by expert developers. We make a case for our proposed flexible scalar unit through three collaborative execution paradigms: data prefetching, control divergence elimination, and scalar-workload extraction. Our experimental results show that significant performance gains can be achieved using our proposed approaches compared to the state-of-art SIMT style processing.

Scalar Waving: Improving the Efficiency of SIMD Execution on GPUs

Ayse Yilmazer, Zhongliang Chen, and David Kaeli

Electrical and Computer Engineering Department
Northeastern University
Boston, USA
yilmazer@ece.neu.edu, zhonchen@ece.neu.edu, kaeli@ece.neu.edu

GPUs take advantage of uniformity in program control flow and utilize SIMD execution to obtain execution efficiency. In SIMD execution, threads are batched into SIMD groups to share a common program counter and execute identical instructions on SIMD pipelines. Previous research has shown that there is a significant number of scalar instructions - instructions where different threads in a SIMD group execute using the same input operands and generate the exact same output - present in a range of applications. GPUs eliminate redundant fetches and decodes by utilizing a shared common pipeline front-end. However, most GPUs do not handle scalar instruction efficiently, allowing these instructions to be redundantly executed by the threads in a SIMD group. In this paper, we propose to use scalar execution to eliminate redundant execution of scalar instructions. We introduce scalar waving as a mechanism to batch scalar operations possessing the same PC and execute them as a group on SIMD lanes for efficiency. We also propose simultaneous execution of dynamically-formed scalar waves with SIMD groups to overcome the under-utilization of SIMD lanes when encountering divergence. We evaluate our work using 22 different GPU benchmarks taken from 4 different benchmark suites. We evaluate a range of configurations using timing simulation. Our results show that scalar waving can obtain up to a 25% improvement in performance on average. Our experiments also provide insight into the amount of performance gain that we can expect with scalar waving as a function of the scalar content, occupancy, and memory characteristics of the target application.

Power and Performance Characterization and Modeling of GPU-Accelerated Systems

Yuki Abe[1], Hiroshi Sasaki[1], Shinpei Kato[2], Koji Inoue[1], Masato Edahiro[2], and Martin Peres[3]

[1]Kyushu University
Fukuoka, Japan
abe@soc.ait.kyushu-u.ac.jp
sasaki@soc.ait.kyushu-u.ac.jp
inoue@ait.kyushu-u.ac.jp

[2]Nagoya University
Nagoya, Japan
shinpei@is.nagoya-u.ac.jp
eda@is.nagoya-u.ac.jp

[3]Laboratoire Bordelais de Recherche en Informatique
Bordeaux, France
martin.peres@labri.fr

Graphics processing units (GPUs) provide an order-of-magnitude improvement on peak performance and performance-per-watt as compared to traditional multicore CPUs. However, GPU-accelerated systems currently lack a generalized method of power and performance prediction, which prevents system designers from an ultimate goal of dynamic power and performance optimization. This is due to the fact that their power and performance characteristics are not well captured across architectures, and as a result, existing power and performance modeling approaches are only available for a limited range of particular GPUs. In this paper, we present power and performance characterization and modeling of GPU-accelerated systems across multiple generations of architectures. Characterization and modeling both play a vital role in optimization and prediction of GPU-accelerated systems. We quantify the impact of voltage and frequency scaling on each architecture with a particularly intriguing result that a cutting-edge Kepler-based GPU achieves energy saving of 75% by lowering GPU clocks in the best scenario, while Fermi- and Tesla-based GPUs achieve no greater than 40% and 13%, respectively. Considering these characteristics, we provide statistical power and performance modeling of GPU-accelerated systems simplified enough to be applicable for multiple generations of architectures. One of our findings is that even simplified statistical models are able to predict power and performance of cutting-edge GPUs within errors of 20% to 30% for any set of voltage and frequency pair.

Energy Efficient HPC on Embedded SoCs: Optimization Techniques for Mali GPU

Ivan Grasso†, Petar Radojković*, Nikola Rajović*, Isaac Gelado*, and Alex Ramirez*‡*

∗ Barcelona Supercomputing Center, Barcelona, Spain
† Institute of Computer Science, University of Innsbruck, Austria
‡ Universitat Politècnica de Catalunya, Barcelona, Spain

A lot of effort from academia and industry has been invested in exploring the suitability of low-power embedded technologies for HPC. Although state-of-the-art embedded systems-on-chip (SoCs) inherently contain GPUs that could be used for HPC, their performance and energy capabilities have never been evaluated. Two reasons contribute to the above. Primarily, embedded GPUs until now, have not supported 64-bit floating point arithmetic - a requirement for HPC. Secondly, embedded GPUs did not provide support for parallel programming languages such as OpenCL and CUDA. However, the situation is changing, and the latest GPUs integrated in embedded SoCs do support 64-bit floating point precision and parallel programming models. In this paper, we analyze performance and energy advantages of embedded GPUs for HPC. In particular, we analyze ARM Mali-T604 GPU - the first embedded GPUs with OpenCL Full Profile support. We identify, implement and evaluate software optimization techniques for efficient utilization of the ARM Mali GPU Compute Architecture. Our results show that, HPC benchmarks running on the ARM Mali-T604 GPU integrated into Exynos 5250 SoC, on average, achieve speed-up of 8.7X over a single Cortex-A15 core, while consuming only 32% of the energy. Overall results show that embedded GPUs have performance and energy qualities that make them candidates for future HPC systems.

Session 4: I/O, Storage, and Networking

IPDPS 2014

Bursting the Cloud Data Bubble: Towards Transparent Storage Elasticity in IaaS Clouds

Bogdan Nicolae

IBM Research, Ireland
bogdan.nicolae@ie.ibm.com

Pierre Riteau

University of Chicago, USA
priteau@uchicago.edu

Kate Keahey

Argonne National Laboratory, USA
keahey@mcs.anl.gov

Storage elasticity on IaaS clouds is an important feature for data-intensive workloads: storage requirements can vary greatly during application runtime, making worst-case over-provisioning a poor choice that leads to unnecessarily tied-up storage and extra costs for the user. While the ability to adapt dynamically to storage requirements is thus attractive, how to implement it is not well understood. Current approaches simply rely on users to attach and detach virtual disks to the virtual machine (VM) instances and then manage them manually, thus greatly increasing application complexity while reducing cost efficiency. Unlike such approaches, this paper aims to provide a transparent solution that presents a unified storage space to the VM in the form of a regular POSIX file system that hides the details of attaching and detaching virtual disks by handling those actions transparently based on dynamic application requirements. The main difficulty in this context is to understand the intent of the application and regulate the available storage in order to avoid running out of space while minimizing the performance overhead of doing so. To this end, we propose a storage space prediction scheme that analyzes multiple system parameters and dynamically adapts monitoring based on the intensity of the I/O in order to get as close as possible to the real usage. We show the value of our proposal over static worst-case over-provisioning and simpler elastic schemes that rely on a reactive model to attach and detach virtual disks, using both synthetic benchmarks and real-life data-intensive applications. Our experiments demonstrate that we can reduce storage waste/cost by 30-40% with only 2-5% performance overhead.

Scibox: Online Sharing of Scientific Data via the Cloud

Jian Huang†, Xuechen Zhang†, Greg Eisenhauer†, Karsten Schwan†, Matthew Wolf†, Stephane Ethier§, and Scott Klasky**

† Georgia Institute of Technology, § Princeton Plasma Physics Laboratory,
* Oak Ridge National Laboratory

{jhuang95, xczhang, eisen, schwan, mwolf}@cc.gatech.edu, {ethier}@pppl.gov,
{klasky}@ornl.gov

Collaborative science demands global sharing of scientific data. But it cannot leverage universally accessible cloud-based infrastructures like Drop Box, as those offer limited interfaces and inadequate levels of access bandwidth. We present the Scibox cloud facility for online sharing scientific data. It uses standard cloud storage solutions, but offers a usage model in which high end codes can write/read data to/from the cloud via the APIs they already use for their I/O actions. With Scibox, data upload/download volumes are controlled via Data Reduction-functions stated by end users and applied at the data source, before data is moved, with further gains in efficiency obtained by combining DR-functions to move exactly what is needed by current data consumers. We evaluate Scibox with science applications and their representative data analytics - the GTS fusion and the combustion image processing - demonstrating the potential for ubiquitous data access with substantial reductions in network traffic.

CALCioM: Mitigating I/O Interference in HPC Systems through Cross-Application Coordination

Matthieu Dorier, Gabriel Antoniu†, Rob Ross‡, Dries Kimpe‡, and Shadi Ibrahim†*

∗ENS Cachan Brittany
IRISA, Rennes, France
matthieu.dorier@irisa.fr

†INRIA Rennes Bretagne-Atlantique
Rennes, France
{gabriel.antoniu, shadi.ibrahim}@inria.fr

‡Argonne National Laboratory
IL 60439, USA
{rross, dkimpe}@mcs.anl.gov

Unmatched computation and storage performance in new HPC systems have led to a plethora of I/O optimizations ranging from application-side collective I/O to network and disk-level request scheduling on the file system side. As we deal with ever larger machines, the interference produced by multiple applications accessing a shared parallel file system in a concurrent manner becomes a major problem. Interference often breaks single-application I/O optimizations, dramatically degrading application I/O performance and, as a result, lowering machine wide efficiency. This paper focuses on CALCioM, a framework that aims to mitigate I/O interference through the dynamic selection of appropriate scheduling policies. CALCioM allows several applications running on a supercomputer to communicate and coordinate their I/O strategy in order to avoid interfering with one another. In this work, we examine four I/O strategies that can be accommodated in this framework: serializing, interrupting, interfering and coordinating. Experiments on Argonne's BG/P Surveyor machine and on several clusters of the French Grid'5000 show how CALCioM can be used to efficiently and transparently improve the scheduling strategy between two otherwise interfering applications, given specified metrics of machine wide efficiency.

Active Measurement of the Impact of Network Switch Utilization on Application Performance

Marc Casas

Barcelona Supercomputing Center
Jordi Girona, 29. Nexus II Building
08034 Barcelona

Greg Bronevetsky

Lawrence Livermore National Laboratory
7000 East Avenue
Livermore, CA, 94550

Inter-node networks are a key capability of High-Performance Computing (HPC) systems that differentiates them from less capable classes of machines. However, in spite of their very high performance, the increasing computational power of HPC compute nodes and the associated rise in application communication needs make network performance a common performance bottleneck. To achieve high performance in spite of network limitations application developers require tools to measure their applications' network utilization and inform them about how the network's communication capacity relates to the performance of their applications. This paper presents a new performance measurement and analysis methodology based on empirical measurements of network behavior. Our approach uses two benchmarks that inject extra network communication. The first probes the fraction of the network that is utilized by a software component (an application or an individual task) to determine the existence and severity of network contention. The second aggressively injects network traffic while a software component runs to evaluate its performance on less capable networks or when it shares the network with other software components. We then combine the information from the two types of experiments to predict the performance slowdown experienced by multiple software components (e.g. multiple processes of a single MPI application) when they share a single network. Our methodology is applied to individual network switches and demonstrated taking 6 representative HPC applications and predicting the performance slowdowns of the 36 possible application pairs. The average error of our predictions is less than 10%.

Session 5: Multi-core Algorithms

IPDPS 2014

Multi-resource Real-Time Reader/Writer Locks for Multiprocessors

Bryan C. Ward and James H. Anderson

Dept. of Computer Science
The University of North Carolina at Chapel Hill

A fine-grained locking protocol permits multiple locks to be held simultaneously by the same task. In the case of real-time multiprocessor systems, prior work on such protocols has considered only mutex constraints. This unacceptably limits concurrency in systems in which some resource accesses are read-only. To remedy this situation, a variant of a recently proposed fine-grained protocol called the real-time nested locking protocol (RNLP) is presented that enables concurrent reads. This variant is shown to have worst-case blocking no worse (and often better) than existing coarse-grained real-time reader/writer locking protocols, while allowing for additional parallelism. Experimental evaluations of the proposed protocol are presented that consider both schedulability (i.e., the ability to validate timing constraints) and implementation-related overheads. These evaluations demonstrate that the RNLP (both the mutex and the proposed reader/writer variant) provides improved schedulability over existing coarse-grained locking protocols, and is practically implementable.

Remote Invalidation: Optimizing the Critical Path of Memory Transactions

Ahmed Hassan, Roberto Palmieri, and Binoy Ravindran

Electrical and Computer Engineering Department
Virginia Tech
Blacksburg, Virginia, USA
hassan84@vt.edu, robertop@vt.edu, binoy@vt.edu

Software Transactional Memory (STM) systems are increasingly emerging as a promising alternative to traditional locking algorithms for implementing generic concurrent applications. To achieve generality, STM systems incur overheads to the normal sequential execution path, including those due to spin locking, validation (or invalidation), and commit/abort routines. We propose a new STM algorithm called Remote Invalidation (or RInval) that reduces these overheads and improves STM performance. RInval's main idea is to execute commit and invalidation routines on remote server threads that run on dedicated cores, and use cache-aligned communication between application's transactional threads and the server routines. By remote execution of commit and invalidation routines and cache-aligned communication, RInval reduces the overhead of spin locking and cache misses on shared locks. By running commit and invalidation on separate cores, they become independent of each other, increasing commit concurrency. We implemented RInval in the Rochester STM framework. Our experimental studies on micro-benchmarks and the STAMP benchmark reveal that RInval outperforms InvalSTM, the corresponding non-remote invalidation algorithm, by as much as an order of magnitude. Additionally, RInval obtains competitive performance to validation-based STM algorithms such as NOrec, yielding up to 2x performance improvement.

Revisiting Asynchronous Linear Solvers: Provable Convergence Rate through Randomization

H. Avron and A. Gupta

IBM T.J. Watson Research Center
haimav@us.ibm.com
anshul@us.ibm.com

A. Druinsky

Tel Aviv University
alexdrui@post.tau.ac.il

Asynchronous methods for solving systems of linear equations have been researched since Chazan and Mir Anker's pioneering 1969 paper. The underlying idea of asynchronous methods is to avoid processor idle time by allowing the processors to continue to make progress even if not all progress made by other processors has been communicated to them. Historically, work on asynchronous methods for solving linear equations focused on proving convergence in the limit. Comparison of the asynchronous convergence rate with its synchronous counterpart and its scaling with the number of processors were seldom studied, and are still not well understood. Furthermore, the applicability of these methods was limited to restricted classes of matrices, such as diagonally dominant matrices. We propose a randomized shared-memory asynchronous method for general symmetric positive definite matrices. We rigorously analyze the convergence rate and prove that it is linear, and is close to that of the method's synchronous counterpart if the processor count is not excessive relative to the size and sparsity of the matrix. Our work presents a significant improvement in convergence analysis as well as in the applicability of asynchronous linear solvers, and suggests randomization as a key paradigm to serve as a foundation for asynchronous methods.

Accelerating MPI Collective Communications through Hierarchical Algorithms Without Sacrificing Inter-Node Communication Flexibility

Benjamin S. Parsons and Vijay S. Pai

Purdue University - West Lafayette, IN
bsparson,vpai@purdue.edu

This paper presents and evaluates a universal algorithm to improve the performance of MPI collective communication operations on hierarchical clusters with many-core nodes. This algorithm exploits shared-memory buffers for efficient intra-node communication while still allowing the use of unmodified, hierarchy-unaware traditional collectives for inter-node communication (including collectives like Alltoallv). This algorithm improves on past works that convert a specific collective algorithm into a hierarchical version and are generally restricted to fan-in, fan-out, and All gather algorithms. Experimental results show impressive performance improvements utilizing a variety of collectives from MPICH as well as the closed-source Cray MPT for the inter-node communication. The experimental evaluation tests the new algorithms with as many as 65536 cores and sees speedups over the baseline averaging 14.2x for Alltoallv, 26x for All gather, and 32.7x for Reduce-Scatter. The paper further improves inter-node communication by utilizing multiple senders from the same shared memory buffer, achieving additional speedups averaging 2.5x. The discussion also evaluates special-purpose extensions to improve intra-node communication by returning shared memory or copy-on-write protected buffers from the collective.

Session 6: Computational Biology

IPDPS 2014

Enabling In-Situ Data Analysis for Large Protein-Folding Trajectory Datasets

Boyu Zhang, Trilce Estrada†, Pietro Cicotti‡, and Michela Taufer**

∗University of Delaware
{bzhang, taufer}@udel.edu

†University of New Mexico
{estrada}@cs.unm.edu

‡San Diego Supercomputer Center
{pcicotti}@sdsc.edu

This paper presents a one-pass, distributed method that enables in-situ data analysis for large protein folding trajectory datasets by executing sufficiently fast, avoiding moving trajectory data, and limiting the memory usage. First, the method extracts the geometric shape features of each protein conformation in parallel. Then, it classifies sets of consecutive conformations into meta-stable and transition stages using a probabilistic hierarchical clustering method. Lastly, it rebuilds the global knowledge necessary for the intraand inter-trajectory analysis through a reduction operation. The comparison of our method with a traditional approach for a villin headpiece sub domain shows that our method generates significant improvements in execution time, memory usage, and data movement. Specifically, to analyze the same trajectory consisting of 20,000 protein conformations, our method runs in 41.5 seconds while the traditional approach takes approximately 3 hours, uses 6.9MB memory per core while the traditional method uses 16GB on one single node where the analysis is performed, and communicates only 4.4KB while the traditional method moves the entire dataset of 539MB. The overall results in this paper support our claim that our method is suitable for in-situ data analysis of folding trajectories.

Overcoming the Limitations Posed by TCRβ Repertoire Modeling through a GPU-Based In-Silico DNA Recombination Algorithm

Gregory Striemer, Harsha Krovi†, Ali Akoglu*, Benjamin Vincent§, Ben Hopson¶, Jeffrey Frelinger‡, and Adam Buntzman‡*

∗Department of Electrical and Computer Engineering
University of Arizona, Tucson, AZ 85721
Email:gmstrie@email.arizona.edu, akoglu@ece.arizona.edu

† Integrated Department of Immunology
University of Colorado, Denver, CO, 80206
Email: shk8f@virginia.edu

‡Department of Immunobiology
University of Arizona, Tucson, AZ, 85719
Email: buntzman@email.arizona.edu,jfrelin@email.arizona.edu
§Department of Medicine, Division of Hematology-Oncology University of North Carolina-Chapel Hill, NC, 27599
Email: BGVincen@unch.unc.edu

¶School of Engineering,
The University of Edinburgh, Edinburgh, EH9 3JL, UK
Email: B.Hopson@ed.ac.uk

The DNA recombination process known as V(D)J recombination is the central mechanism for generating diversity among antigen receptors such as T-cell receptors (TCRs). This diversity is crucial for the development of the adaptive immune system. However, modeling of all the alpha beta TCR sequences is encumbered by the enormity of the potential repertoire, which has been predicted to exceed 10^15 sequences. Prior modeling efforts have, therefore, been limited to extrapolations based on the analysis of minor subsets of the overall TCRbeta repertoire. In this study, we map the recombination process completely onto the graphics processing unit (GPU) hardware architecture using the CUDA programming environment to circumvent prior limitations. For the first time, we present a model of the mouse TCRbeta repertoire to an extent which enabled us to evaluate the Convergent Recombination Hypothesis (CRH) comprehensively at peta-scale level on a single GPU.

Parallel Mutual Information Based Construction of Whole-Genome Networks on the Intel® Xeon Phi ™ Coprocessor

Sanchit Misra, Kiran Pamnany*, and Srinivas Aluru†‡*

∗Parallel Computing Lab
Intel Corporation, Bangalore, India,
Email: {sanchit.misra,kiran.pamnany}@intel.com

†School of Computational Science and Engineering
Georgia Institute of Technology, Atlanta, USA
Email: aluru@cc.gatech.edu

‡Dept. of Computer Science and Engineering
Indian Institute of Technology Bombay, Mumbai, India

Construction of whole-genome networks from large-scale gene expression data is an important problem in systems biology. While several techniques have been developed, most cannot handle network reconstruction at the whole-genome scale, and the few that can, require large clusters. In this paper, we present a solution on the Intel (R) Xeon Phi (TM) coprocessor, taking advantage of its multi-level parallelism including many x86-based cores, multiple threads per core, and vector processing units. We also present a solution on the Intel (R) Xeon (R) processor. Our solution is based on TINGe, a fast parallel network reconstruction technique that uses mutual information and permutation testing for assessing statistical significance. We demonstrate the first ever inference of a plant whole genome regulatory network on a single chip by constructing a 15,575 gene network of the plant Arabidopsis thaliana from 3,137 microarray experiments in only 22 minutes. In addition, our optimization for parallelizing mutual information computation on the Intel Xeon Phi coprocessor holds out lessons that are applicable to other domains.

cuBLASTP: Fine-Grained Parallelization of Protein Sequence Search on a GPU

Jing Zhang, Hao Wang*, Heshan Lin*, and Wu-Chun Feng*†*

∗Dept. of Computer Science and †Dept. of Electrical & Computer Engineering
Virginia Tech
Email: {zjing14, hwang121, hlin2, wfeng}@vt.edu

BLAST, short for Basic Local Alignment Search Tool, is a fundamental algorithm in the life sciences that compares biological sequences. However, with the advent of next-generation sequencing (NGS) and increase in sequence read-lengths, whether at the outset or downstream from NGS, the exponential growth of sequence databases is arguably outstripping our ability to analyze the data. Though several recent studies have utilized the graphics processing unit (GPU) to speedup the BLAST algorithm for searching protein sequences (i.e., BLASTP), these studies used coarse-grained parallel approaches, where one sequence alignment is mapped to only one thread. Moreover, due to the irregular memory access patterns in BLASTP, there remain significant challenges to map the most time-consuming phases (i.e., hit detection and ungapped extension) to the GPU using a fine-grained multithreaded approach. To address the above issues, we propose cuBLASTP, an efficient fine-grained BLASTP implementation for the GPU using CUDA. Our cuBLASTP realization encompasses many research contributions, including (1) memory-access reordering to reorder hits from column-major order to diagonal-major order, (2) position-based indexing to map a hit with a packed data structure to a bin, (3) aggressive hit filtering to eliminate hits beyond the threshold distance along the diagonal, (4) diagonal-based parallelism and hit-based parallelism for ungapped extension to extend sequences with different lengths in databases, and (5) hierarchical buffering to reduce memory-access overhead for the core data structures. The experimental results show that on a NVIDIA Kepler GPU, cuBLASTP delivers up to a 5.0-fold speedup over sequential FSA-BLAST and a 3.7-fold speedup over multithreaded NCBI-BLAST for the overall program execution. In addition, compared with GPU-BLASTP (the fastest GPU implementation of BLASTP to date), cuBLASTP achieves up to a 2.8-fold speedup for the kernel execution on the GPU and a 1.8-fold speedup for the overall program execution.

Session 7: Interconnection Network

IPDPS 2014

Skywalk: A Topology for HPC Networks with Low-Delay Switches

Ikki Fujiwara and Michihiro Koibuchi*†*

*National Institute of Informatics / JST
†The Graduate University for Advanced Studies (SOKENDAI)
2-1-2 Hitotsubashi, Chiyoda-ku,
Tokyo, JAPAN 101-8430
Email: fikki, koibuchig@nii.ac.jp

Hiroki Matsutani

Keio University
3-14-1 Hiyoshi, Kohoku-ku, Yokohama,
Kanagawa, JAPAN 223-8522
Email: matutani@arc.ics.keio.ac.jp

Henri Casanova

University of Hawai‘i at Manoa
1680 East-West Road, Honolulu,
HI, U.S.A. 96822
Email: henric@hawaii.edu

With low-delay switches on the horizon, end-to-end latency in large-scale High Performance Computing (HPC) interconnects will be dominated by cable delays. In this context we define a new network topology, Skywalk, for deploying low-latency interconnects in upcoming HPC systems. Skywalk uses randomness to achieve low latency, but does so in a way that accounts for the physical layout of the topology so as to lead to further cable length and thus latency reductions. Via graph analysis and discrete-event simulation we show that Skywalk compares favorably (in terms of latency, cable length, and throughput) to traditional low-degree torus and moderate-degree hypercube topologies, to high-degree fully-connected Dragonfly topologies, to the HyperX topology, and to recently proposed fully random topologies.

LFTI: A New Performance Metric for Assessing Interconnect Designs for Extreme-Scale HPC Systems

Xin Yuan and Santosh Mahapatra

Department of Computer Science, Florida State University
Tallahassee, FL 32312
Email: {xyuan,mahapatr}@cs.fsu.edu

Michael Lang and Scott Pakin

Los Alamos National Laboratory
Los Alamos, New Mexico
Email: {mlang,pakin}@lanl.gov

Traditionally, interconnect performance is either characterized by simple topological parameters such as bisection bandwidth or studied through simulation that gives detailed performance information for the scenarios simulated. Neither of these approaches provides a good performance overview for extreme-scale interconnects. The topological parameters are not directly related to application level communication performance while the simulation complexity limits the number of scenarios that can be investigated. In this work, we propose a new performance metric, called LANL-FSU Throughput Indices (LFTI), for characterizing the throughput performance of interconnect designs. LFTI combines the simplicity of topological parameters and the accuracy of simulation: like topological parameters, LFTI can be derived from interconnect specification, at the same time, it directly reflects the application level communication performance. Moreover, in cases when the theoretical throughput for each communication pattern can be modeled efficiently for an interconnect, LFTI for the interconnect can be computed efficiently. These features potentially allow LFTI to be used for rapid and comprehensive evaluation and comparison of extreme-scale interconnect designs. We demonstrate the effectiveness of LFTI by using it to evaluate and explore the design space of a number of large-scale interconnect designs.

An Improved Router Design for Reliable On-Chip Networks

Pavan Poluri and Ahmed Louri

Department of Electrical and Computer Engineering
University of Arizona
Tucson, USA
pavanp@email.arizona.edu
louri@email.arizona.edu

Aggressive technology scaling into the deep nanometer regime has made the Network-on-Chip (NoC) in multicore architectures increasingly vulnerable to faults. This has accelerated the need for designing reliable NoCs. To this end, we propose a reliable NoC router architecture capable of tolerating multiple permanent faults. The proposed router achieves a better reliability without incurring too much area and power overhead as compared to the baseline NoC router or other fault-tolerant routers. Reliability analysis using Mean Time to Failure (MTTF) reveals that our proposed router is six times more reliable than the baseline NoC router (without protection). We also compare our proposed router with other existing fault-tolerant routers such as Bullet Proof, Vicis and RoCo using Silicon Protection Factor (SPF) as a metric. SPF analysis shows that our proposed router is more reliable than the mentioned existing fault tolerant routers. Hardware synthesis performed by Cadence Encounter RTL Compiler using commercial 45nm technology library shows that the correction circuitry incurs an area overhead of 31% and power overhead of 30%. Latency analysis on a 64-core mesh based NoC simulated using GEM5 and running SPLASH-2 and PARSEC benchmark application traffic shows that in the presence of multiple faults, our proposed router increases the overall latency by only 10% and 13% respectively while providing better reliability.

Energy-Efficient Time-Division Multiplexed Hybrid-Switched NoC for Heterogeneous Multicore Systems

Jieming Yin, Sachin S. Sapatnekar, and Antonia Zhai

University of Minnesota, Twin Cities
Minneapolis, Minnesota 55455, USA
{jyin, zhai}@cs.umn.edu, sachin@umn.edu

Pingqiang Zhou

ShanghaiTech University
Shanghai 200031, China
zhoupq@shanghaitech.edu.cn

NoCs are an integral part of modern multicore processors, they must continuously support high-throughput low-latency on-chip data communication under a stringent energy budget when system size scales up. Heterogeneous multicore systems further push the limit of NoC design by integrating cores with diverse performance requirements onto the same die. Traditional packet-switched NoCs, which have the flexibility of connecting diverse computation and storage devices, are facing great challenges to meet the performance requirements within the energy budget due to latency and energy consumption associated with buffering and routing at each router. In this paper, we take advantage of the diversity in performance requirements of on-chip heterogeneous computing devices by designing, implementing, and evaluating a hybrid-switched network that allows the packet-switched and circuit-switched messages to share the same communication fabric by partitioning the network through time-division multiplexing (TDM). In the proposed hybrid-switched network, circuit-switched paths are established along frequently communicating nodes. Our experiments show that utilizing these paths can improve system performance by reducing communication latency and alleviating network congestion. Furthermore, better energy efficiency is achieved by reducing buffering in routers and in turn enabling aggressive power gating.

Session 8: System-Level Resource Management

IPDPS 2014

Heterogeneity-Aware Workload Placement and Migration in Distributed Sustainable Datacenters

Dazhao Cheng and Xiaobo Zhou

Department of Computer Science
University of Colorado, Colorado Springs, USA

Changjun Jiang

Department of Computer Science & Technology
Tongji University, Shanghai, China
Email addresses: dcheng@uccs.edu, cjjiang@tongji.edu.cn, xzhou@uccs.edu

While major cloud service operators have taken various initiatives to operate their sustainable data enters with green energy, it is challenging to effectively utilize the green energy since its generation depends on dynamic natural conditions. Fortunately, the geographical distribution of data enters provides an opportunity for optimizing the system performance by distributing cloud workloads. In this paper, we propose a holistic heterogeneity-aware cloud workload placement and migration approach, sCloud, that aims to maximize the system good put in distributed self-sustainable data enters. sCloud adaptively places the transactional workload to distributed data enters, allocates the available resource to heterogeneous workloads in each data enter, and migrates batch jobs across data enters, while taking into account the green power availability and QoS requirements. We formulate the transactional workload placement as a constrained optimization problem that can be solved by nonlinear programming. Then, we propose a batch job migration algorithm to further improve the system good put when the green power supply varies widely at different locations. We have implemented sCloud in a university cloud test bed with real-world weather conditions and workload traces. Experimental results demonstrate sCloud can achieve near-to-optimal system performance while being resilient to dynamic power availability. It outperforms a heterogeneity-oblivious approach by 26% in improving system good put and 29% in reducing QoS violations.

Online Server and Workload Management for Joint Optimization of Electricity Cost and Carbon Footprint Across Data Centers

Zahra Abbasi, Madhurima Pore, and Sandeep K.S. Gupta

IMPACT Lab (http://impact.asu.edu/)
Arizona State University, Tempe, AZ
zahra.abbasi, madhurima.pore, and sandeep.gupta@asu.edu

Internet data centers, typically distributed across the world in order to provide timely and reliable Internet service, have been increasingly pressurized to reduce their carbon footprint and electricity cost. Particularly, data centers will soon be required to abide by carbon capping polices which impose carbon footprint limits to encourage brown energy conservation. We propose an online algorithm, called OnlineCC, for minimizing the operational cost while satisfying the carbon footprint reduction target of a set of geo-distributed data centers. OnlineCC makes use of Lyapunov optimization technique while operating without long-term future information, making it attractive in the presence of uncertainties associated with data center information e.g., input workload. We prove that OnlineCC achieves a near optimal operational cost (electricity cost) compared to the optimal algorithm with future information, while bounding the potential violation of carbon footprint target, depending on the Lyapunov control parameter, namely V. We also give a heuristic for finding V which significantly shortens the search space to adjust its value. Finally, we perform a trace-based simulation study and a small scale experiment to complement the analysis. The results show that OnlineCC reduces cost by more than 18% compared to a prediction-based online solution while resulting in equal or smaller carbon footprint.

Cost-Efficient and Resilient Job Life-Cycle Management on Hybrid Clouds

Hsuan-Yi Chu

Computer Science Department
University of Southern California
Los Angeles, CA 90089, U.S.A.
Email: hsuanyi@usc.edu

Yogesh Simmhan

Supercomputer Education and Research Center
Indian Institute of Science
Bangalore 560012, India
Email: simmhan@serc.iisc.in

Cloud infrastructure offers democratized access to on-demand computing resources for scaling applications beyond captive local servers. While on-demand, fixed-price Virtual Machines (VMs) are popular, the availability of cheaper, but less reliable, spot VMs from cloud providers presents an opportunity to reduce the cost of hosting cloud applications. Our work addresses the issue of effective and economic use of hybrid cloud resources for planning job executions with deadline constraints. We propose strategies to manage a job's life-cycle on spot and on on-demand VMs to minimize the total dollar cost while assuring completion. With the foundation of stochastic optimization, our reusable table-based algorithm (RTBA) decides when to instantiate VMs, at what bid prices, when to use local machines, and when to checkpoint and migrate the job between these resources, with the goal of completing the job on time and with the minimum cost. In addition, three simpler heuristics are proposed as comparison. Our evaluation using historical spot prices for the Amazon EC2 market shows that RTBA on an average reduces the cost by 72%, compared to running on only on-demand VMs. It is also robust to fluctuations in spot prices. The heuristic, H3, often approaches RTBA in performance and may prove adequate for ad hoc jobs due to its simplicity.

A Coprocessor Sharing-Aware Scheduler for Xeon Phi-Based Compute Clusters

Giuseppe Coviello, Srihari Cadambi, and Srimat Chakradhar

NEC Laboratories America, Inc.
4 Independence Way, Suite 200
Princeton NJ 08540, USA
{giuseppe.coviello, cadambi, chak}@nec-labs.com

We propose a cluster scheduling technique for compute clusters with Xeon Phi coprocessors. Even though the Xeon Phi runs Linux which allows multiprocessing, cluster schedulers generally do not allow jobs to share coprocessors because sharing can cause oversubscription of coprocessor memory and thread resources. It has been shown that memory or thread oversubscription on a many core like the Phi results in job crashes or drastic performance loss. We first show that such an exclusive device allocation policy causes severe coprocessor underutilization: for typical workloads, on average only 38% of the Xeon Phi cores are busy across the cluster. Then, to improve coprocessor utilization, we propose a scheduling technique that enables safe coprocessor sharing without resource oversubscription. Jobs specify their maximum memory and thread requirements, and our scheduler packs as many jobs as possible on each coprocessor in the cluster, subject to resource limits. We solve this problem using a greedy approach at the cluster level combined with a knapsack-based algorithm for each node. Every coprocessor is modeled as a knapsack and jobs are packed into each knapsack with the goal of maximizing job concurrency, i.e., as many jobs as possible executing on each coprocessor. Given a set of jobs, we show that this strategy of packing for high concurrency is a good proxy for (i) reducing make span, without the need for users to specify job execution times and (ii) reducing coprocessor footprint, or the number of coprocessors required to finish the jobs without increasing make span. We implement the entire system as a seamless add on to Condor, a popular distributed job scheduler, and show make span and footprint reductions of more than 50% across a wide range of workloads.

Session 9: GPU Algorithms

IPDPS 2014

Work-Efficient Parallel GPU Methods for Single-Source Shortest Paths

Andrew Davidson and John D. Owens

University of California, Davis
aaldavidson@ucdavis.edu
jowens@ece.ucdavis.edu

Sean Baxter and Michael Garland

NVIDIA Research
sbaxter@nvidia.com
mgarland@nvidia.com

Finding the shortest paths from a single source to all other vertices is a fundamental method used in a variety of higher-level graph algorithms. We present three parallel friendly and work-efficient methods to solve this Single-Source Shortest Paths (SSSP) problem: Work front Sweep, Near-Far and Bucketing. These methods choose different approaches to balance the trade off between saving work and organizational overhead. In practice, all of these methods do much less work than traditional Bellman-Ford methods, while adding only a modest amount of extra work over serial methods. These methods are designed to have a sufficient parallel workload to fill modern massively-parallel machines, and select reorganizational schemes that map well to these architectures. We show that in general our Near-Far method has the highest performance on modern GPUs, outperforming other parallel methods. We also explore a variety of parallel load-balanced graph traversal strategies and apply them towards our SSSP solver. Our work-saving methods always outperform a traditional GPU Bellman-Ford implementation, achieving rates up to 14x higher on low-degree graphs and 340x higher on scale free graphs. We also see significant speedups (20-60x) when compared against a serial implementation on graphs with adequately high degree.

Efficient Multi-GPU Computation of All-Pairs Shortest Paths

Hristo Djidjev and Sunil Thulasidasan

Los Alamos National Laboratory
Los Alamos, NM, USA
Email: {djidjev,sunil}@lanl.gov

Guillaume Chapuis, Rumen Andonov, and Dominique Lavenier

INRIA/IRISA, University of Rennes 1
Rennes, France
Email: {guillaume.chapuis,rumen.andonov,dominique.lavenier}@irisa.fr

We describe a new algorithm for solving the all-pairs shortest-path (APSP) problem for planar graphs and graphs with small separators that exploits the massive on-chip parallelism available in today's Graphics Processing Units (GPUs). Our algorithm, based on the Floyd-War shall algorithm, has near optimal complexity in terms of the total number of operations, while its matrix-based structure is regular enough to allow for efficient parallel implementation on the GPUs. By applying a divide-and-conquer approach, we are able to make use of multi-node GPU clusters, resulting in more than an order of magnitude speedup over the fastest known Dijkstra-based GPU implementation and a two-fold speedup over a parallel Dijkstra-based CPU implementation.

An Efficient GPU General Sparse Matrix-Matrix Multiplication for Irregular Data

Weifeng Liu and Brian Vinter

Niels Bohr Institute
University of Copenhagen
Copenhagen, Denmark
{weifeng, vinter}@nbi.dk

General sparse matrix-matrix multiplication (SpGEMM) is a fundamental building block for numerous applications such as algebraic multigrid method, breadth first search and shortest path problem. Compared to other sparse BLAS routines, an efficient parallel SpGEMM algorithm has to handle extra irregularity from three aspects: (1) the number of the nonzero entries in the result sparse matrix is unknown in advance, (2) very expensive parallel insert operations at random positions in the result sparse matrix dominate the execution time, and (3) load balancing must account for sparse data in both input matrices. Recent work on GPU SpGEMM has demonstrated rather good both time and space complexity, but works best for fairly regular matrices. In this work we present a GPU SpGEMM algorithm that particularly focuses on the above three problems. Memory pre-allocation for the result matrix is organized by a hybrid method that saves a large amount of global memory space and efficiently utilizes the very limited on-chip scratchpad memory. Parallel insert operations of the nonzero entries are implemented through the GPU merge path algorithm that is experimentally found to be the fastest GPU merge approach. Load balancing builds on the number of the necessary arithmetic operations on the nonzero entries and is guaranteed in all stages. Compared with the state-of-the-art GPU SpGEMM methods in the CUSPARSE library and the CUSP library and the latest CPU SpGEMM method in the Intel Math Kernel Library, our approach delivers excellent absolute performance and relative speedups on a benchmark suite composed of 23 matrices with diverse sparsity structures.

Improving the Performance of CA-GMRES on Multicores with Multiple GPUs

Ichitaro Yamazaki, Hartwig Anzt*, Stanimire Tomov*, Mark Hoemmen§, and Jack Dongarra**

*University of Tennessee, Knoxville, USA

§Sandia National Laboratory, New Mexico, USA
iyamazak@eecs.utk.edu, hanzt@icl.utk.edu, tomov@eecs.utk.edu, mhoemme@sandia.gov, and dongarra@eecs.utk.edu

The Generalized Minimum Residual (GMRES) method is one of the most widely-used iterative methods for solving nonsymmetric linear systems of equations. In recent years, techniques to avoid communication in GMRES have gained attention because in comparison to floating-point operations, communication is becoming increasingly expensive on modern computers. Since graphics processing units (GPUs) are now becoming crucial component in computing, we investigate the effectiveness of these techniques on multicore CPUs with multiple GPUs. While we present the detailed performance studies of a matrix powers kernel on multiple GPUs, we particularly focus on orthogonalization strategies that have a great impact on both the numerical stability and performance of GMRES, especially as the matrix becomes sparser or ill-conditioned. We present the experimental results on two eight-core Intel Sandy Bridge CPUs with three NDIVIA Fermi GPUs and demonstrate that significant speedups can be obtained by avoiding communication, either on a GPU or between the GPUs. As part of our study, we investigate several optimization techniques for the GPU kernels that can also be used in other iterative solvers besides GMRES. Hence, our studies not only emphasize the importance of avoiding communication on GPUs, but they also provide insight about the effects of these optimization techniques on the performance of the sparse solvers, and may have greater impact beyond GMRES.

Session 10: Graph and Network Processing

IPDPS 2014

How Well Do Graph-Processing Platforms Perform? An Empirical Performance Evaluation and Analysis

Yong Guo, Marcin Biczak*, Ana Lucia Varbanescu†, Alexandru Iosup*, Claudio Martella‡, and Theodore L. Willke§*

*TU Delft
The Netherlands
Email: {Yong.Guo, M.Biczak, A.Iosup}@tudelft.nl

†University of Amsterdam
The Netherlands
Email: A.L.Varbanescu@uva.nl

‡VU University Amsterdam
The Netherlands
Email: claudio.martella@vu.nl

§Systems Architecture Lab
Intel Corporation, USA
Email: theodore.l.willke@intel.com

Graph-processing platforms are increasingly used in a variety of domains. Although both industry and academia are developing and tuning graph-processing algorithms and platforms, the performance of graph-processing platforms has never been explored or compared in-depth. Thus, users face the daunting challenge of selecting an appropriate platform for their specific application. To alleviate this challenge, we propose an empirical method for benchmarking graph-processing platforms. We define a comprehensive process, and a selection of representative metrics, datasets, and algorithmic classes. We implement a benchmarking suite of five classes of algorithms and seven diverse graphs. Our suite reports on basic (user-lever) performance, resource utilization, scalability, and various overhead. We use our benchmarking suite to analyze and compare six platforms. We gain valuable insights for each platform and present the first comprehensive comparison of graph-processing platforms.

Complex Network Analysis Using Parallel Approximate Motif Counting

George M. Slota and Kamesh Madduri

Department of Computer Science and Engineering
The Pennsylvania State University
University Park, PA, USA
Email: gms5016@psu.edu, madduri@cse.psu.edu

Subgraph counting forms the basis of many complex network analysis metrics, including motif and anti-motif finding, relative graph let frequency distance, and graph let degree distribution agreements. Determining exact subgraph counts is computationally very expensive. In recent work, we present FASCIA, a shared-memory parallel algorithm and implementation for approximate subgraph counting. FASCIA uses a dynamic programming-based approach and is significantly faster than exhaustive enumeration, while generating high-quality approximations of subgraph counts. However, the memory usage of the dynamic programming step prohibits us from applying FASCIA to very large graphs. In this paper, we introduce a distributed-memory parallelization of FASCIA by partitioning the graph and the dynamic programming table. We discuss a new collective communication scheme to make the dynamic programming step memory-efficient. These optimizations enable scaling to much larger networks than before. We also present a simple parallelization strategy for distributed subgraph counting on smaller networks. The new additions let us use subgraph counts as graph signatures for a large network collection, and we analyze this collection using various subgraph count-based graph analytics.

Finding Motifs in Biological Sequences Using the Micron Automata Processor

Indranil Roy and Srinivas Aluru

School of Computational Science and Engineering
Georgia Institute of Technology, Atlanta, GA 30332
Email: iroy@gatech.edu; aluru@cc.gatech.edu

Finding approximately conserved sequences, called motifs, across multiple DNA or protein sequences is an important problem in computational biology. In this paper, we consider the (l, d) motif search problem of identifying one or more motifs of length l present in at least q of the n given sequences, with each occurrence differing from the motif in at most d substitutions. The problem is known to be NP-hard, and the largest solved instance reported to date is (26, 11). We propose a novel algorithm for the (l, d) motif search problem using streaming execution over a large set of Non-deterministic Finite Automata (NFA). This solution is designed to take advantage of the Micron Automata Processor, a new technology close to deployment that can simultaneously execute multiple NFA in parallel. We estimate the run-time for the (39, 18) and (40, 17) problem instances using the resources available within a single Automata Processor board. In addition to solving larger instances of the (l, d) motif search problem, the paper serves as a useful guide to solving problems using this new accelerator technology.

Traversing Trillions of Edges in Real Time: Graph Exploration on Large-Scale Parallel Machines

Fabio Checconi and Fabrizio Petrini

High Performance Analytics Department
IBM TJ Watson, Yorktown Heights, NY 10598
Email: {fchecco,fpetrin}@us.ibm.com

The world of Big Data is changing dramatically right before our eyes-from the amount of data being produced to the way in which it is structured and used. The trend of "big data growth" presents enormous challenges, but it also presents incredible scientific and business opportunities. Together with the data explosion, we are also witnessing a dramatic increase in data processing capabilities, thanks to new powerful parallel computer architectures and more sophisticated algorithms. In this paper we describe the algorithmic design and the optimization techniques that led to the unprecedented processing rate of 15.3 trillion edges per second on 64 thousand Blue Gene/Q nodes, that allowed the in-memory exploration of a petabyte-scale graph in just a few seconds. This paper provides insight into our parallelization and optimization techniques. We believe that these techniques can be successfully applied to a broader class of graph algorithms.

Session 11: Modeling, Simulation, and Reliability

IPDPS 2014

TBPoint: Reducing Simulation Time for Large-Scale GPGPU Kernels

Jen-Cheng Huang[1], Lifeng Nai[1], Hyesoon Kim[2], and Hsien-Hsin S. Lee[1]

[1]School of Electrical and Computer Engineering
[2]School of Computer Science
Georgia Institute of Technology
{jhuang34, lnai3, hyesoon.kim, leehs}@gatech.edu

Architecture simulation for GPGPU kernels can take a significant amount of time, especially for large-scale GPGPU kernels. This paper presents TBPoint, an infrastructure based on profiling-based sampling for GPGPU kernels to reduce the cycle-level simulation time. Compared to existing approaches, TBPoint provides a flexible and architecture-independent way to take samples. For the evaluated 12 kernels, the geometric means of sampling errors of TBPoint, Ideal-Simpoint, and random sampling are 0.47%, 1.74%, and 7.95%, respectively, while the geometric means of the total sample size of TBPoint, Ideal-Simpoint, and random sampling are 2.6%, 5.4%, and 10%, respectively. TBPoint narrows the speed gap between hardware and GPGPU simulators, enabling more and more large-scale GPGPU kernels to be analyzed using detailed timing simulations.

Algorithmic Time, Energy, and Power on Candidate HPC Compute Building Blocks

Jee Choi, Marat Dukhan, Xing Liu, and Richard Vuduc

School of Computational Science and Engineering
Georgia Institute of Technology
Atlanta, Georgia, 30332-0765, USA
{jee,mdukhan3,xing.liu,richie}@gatech.edu

We conducted a micro benchmarking study of the time, energy, and power of computation and memory access on several existing platforms. These platforms represent candidate compute-node building blocks of future high-performance computing systems. Our analysis uses the "energy roofline" model, developed in prior work, which we extend in two ways. First, we improve the model's accuracy by accounting for power caps, basic memory hierarchy access costs, and measurement of random memory access patterns. Secondly, we empirically evaluate server-, mini-, and mobile-class platforms that span a range of compute and power characteristics. Our study includes a dozen such platforms, including x86 (both conventional and Xeon Phi), ARM, GPU, and hybrid (AMD APU and other SoC) processors. These data and our model analytically characterize the range of algorithmic regimes where we might prefer one building block to others. It suggests critical values of arithmetic intensity around which some systems may switch from being more to less time- and energy-efficient than others, it further suggests how, with respect to intensity, operations should be throttled to meet a power cap. We hope our methods can help make debates about the relative merits of these and other systems more quantitative, analytical, and insightful.

Characterization of Impact of Transient Faults and Detection of Data Corruption Errors in Large-Scale N-Body Programs Using Graphics Processing Units

Keun Soo Yim

Google, Inc.
yim@google.com

In N-body programs, trajectories of simulated particles have chaotic patterns if errors are in the initial conditions or occur during some computation steps. It was believed that the global properties (e.g., total energy) of simulated particles are unlikely to be affected by a small number of such errors. In this paper, we present a quantitative analysis of the impact of transient faults in GPU devices on a global property of simulated particles. We experimentally show that a single-bit error in non-control data can change the final total energy of a large-scale N-body program with ~2.1% probability. We also find that the corrupted total energy values have certain biases (e.g., the values are not a normal distribution), which can be used to reduce the expected number of re-executions. In this paper, we also present a data error detection technique for N-body programs by utilizing two types of properties that hold in simulated physical models. The presented technique and an existing redundancy-based technique together cover many data errors (e.g., >97.5%) with a small performance overhead (e.g., 2.3%).

Analytically Modeling Application Execution for Software-Hardware Co-design

Jichi Guo†, Jiayuan Meng‡, Qing Yi†, Vitali Morozov‡, and Kalyan Kumaran‡

University of Colorado Colorado Springs†
Colorado Springs, CO, USA
{jguo2, qyi}@uccs.edu

Argonne National Laboratory‡
Lemont, IL, USA
{jmeng, morozov, kumaran}@anl.gov

Software-hardware co-design has become increasingly important as the scale and complexity of both are reaching an unprecedented level. To predict and understand application behavior on emerging or conceptual systems, existing research has mostly relied on cycle-accurate micro-architecture simulators, which are known to be time-consuming and are oblivious to workloads' control flow structure. As a result, simulations are often limited to small kernels, and the first step in the co-design process is often to extract important kernels, construct mini-applications, and identify potential hardware limitations. This requires a high level understanding about the full applications' potential behavior on a future system, e.g. the most time-consuming regions, the performance bottlenecks for these regions, etc. Unfortunately, such application knowledge gained from one system may not hold true on a future system. One solution is to instrument the full application with timers and simulate it with a reasonable input size, which can be a daunting task in itself. We propose an alternative approach to gain first-order insights into hardware-dependent application behavior by trading off the accuracy of analysis for improved efficiency. By modeling the execution flows of user applications and analyzing it using target hardware's performance models, our technique requires no cycle-accurate simulation on a prospective system. In fact, our technique's analysis time does not increase with the input data size.

Session 12: Accelerator Application Development and Optimization

IPDPS 2014

Interactive Program Debugging and Optimization for Directive-Based, Efficient GPU Computing

*Seyong Lee†, Dong Li†, and Jeffrey S. Vetter†**

†Oak Ridge National Laboratory

*Georgia Institute of Technology
{lees2, lid1}@ornl.gov, {vetter}@computer.org

Directive-based GPU programming models are gaining momentum, since they transparently relieve programmers from dealing with complexity of low-level GPU programming, which often reflects the underlying architecture. However, too much abstraction in directive models puts a significant burden on programmers for debugging applications and tuning performance. In this paper, we propose a directive-based, interactive program debugging and optimization system. This system enables intuitive and synergistic interaction among programmers, compilers, and runtimes for more productive and efficient GPU computing. We have designed and implemented a series of prototype tools within our new open source compiler framework, called Open Accelerator Research Compiler (Open ARC), Open ARC supports the full feature set of Opencast V1.0. Our evaluation on twelve Open ACC benchmarks demonstrates that our prototype debugging and optimization system can detect a variety of translation errors. Additionally, the optimization provided by our prototype minimizes memory transfers, when compared to a fully manual memory management scheme.

Unified Development for Mixed Multi-GPU and Multi-coprocessor Environments Using a Lightweight Runtime Environment

Azzam Haidar, Chongxiao Cao*, Asim Yarkhan*, Piotr Luszczek*, Stanimire Tomov*, Khairul Kabir*, and Jack Dongarra*†‡*

∗University of Tennessee, Knoxville, USA

†Oak Ridge National Laboratory, Oak Ridge, USA

‡University of Manchester, Manchester M13 9PL, UK

Many of the heterogeneous resources available to modern computers are designed for different workloads. In order to efficiently use GPU resources, the workload must have a greater degree of parallelism than a workload designed for multicore-CPUs. And conceptually, the Intel Xeon Phi coprocessors are capable of handling workloads somewhere in between the two. This multitude of applicable workloads will likely lead to mixing multicore-CPUs, GPUs, and Intel coprocessors in multi-user environments that must offer adequate computing facilities for a wide range of workloads. In this work, we are using a lightweight runtime environment to manage the resource-specific workload, and to control the dataflow and parallel execution in two-way hybrid systems. The lightweight runtime environment uses task superscalar concepts to enable the developer to write serial code while providing parallel execution. In addition, our task abstractions enable unified algorithmic development across all the heterogeneous resources. We provide performance results for dense linear algebra applications, demonstrating the effectiveness of our approach and full utilization of a wide variety of accelerator hardware.

Nitro: A Framework for Adaptive Code Variant Tuning

Saurav Muralidharan, Manu Shantharam, and Mary Hall

University of Utah
Salt Lake City, UT
Email: {sauravm, manushan, mhall}@cs.utah.edu

Michael Garland and Bryan Catanzaro

NVIDIA Corporation
Santa Clara, CA
Email: {mgarland, bcatanzaro}@nvidia.com

Auto tuning systems intelligently navigate a search space of possible implementations of a computation to find the implementation(s) that best meets a specific optimization criteria, usually performance. This paper describes Nitro, a programmer-directed auto tuning framework that facilitates tuning of code variants, or alternative implementations of the same computation. Nitro provides a library interface that permits programmers to express code variants along with meta-information that aids the system in selecting among the set of variants at run time. Machine learning is employed to build a model through training on this meta-information, so that when a new input is presented, Nitro can consult the model to select the appropriate variant. In experiments with five real-world irregular GPU benchmarks from sparse numerical methods, graph computations and sorting, Nitro-tuned variants achieve over 93% of the performance of variants selected through exhaustive search. Further, we describe optimizations and heuristics in Nitro that substantially reduce training time and other overheads.

Keynote Speaker 2

IPDPS 2014

Reading the Tea-Leaves: How Architecture Has Evolved at the High End

Peter Kogge

University of Notre Dame

The 2008 DARPA Exascale study was one of the first in-depth attempts to project ahead key characteristics for high-end massively parallel systems on the basis of technology trends, architectures, and computational kernels, and identified four major challenges for future systems designs. It focused on a single benchmark, Linpack, and identified two distinct classes of architectures: “heavyweight” and “lightweight.” This talk is a continuation of a series of updates to that study, and includes not only the most recent technology projections but also several new benchmarks for which significant multi-year data exists, and new classes of architectures that have emerged since then. The talk will address changes in characteristics (both before and after the seminal year of 2004 where multi-core took over), and how those characteristics are likely to project into the future. A series of vignettes on specific features will provide insight into areas where current design trends are becoming over or under-balanced. Special attention is given to both computational energy and memory.

Session 13: Combinatorial Algorithms

IPDPS 2014

New Effective Multithreaded Matching Algorithms

Fredrik Manne

Department of Informatics
University of Bergen
N-5020 Bergen, Norway
Email: fredrikm@ii.uib.no

Mahantesh Halappanavar

Pacific Northwest National Laboratory
902 Battelle Boulevard
P.O.Box 999, MSIN J4-30
Richland, WA 99352, USA
Email: hala@pnnl.gov

Matching is an important combinatorial problem with a number of applications in areas such as community detection, sparse linear algebra, and network alignment. Since computing optimal matchings can be very time consuming, several fast approximation algorithms, both sequential and parallel, have been suggested. Common to the algorithms giving the best solutions is that they tend to be sequential by nature, while algorithms more suitable for parallel computation give solutions of lower quality. We present a new simple 1/2-approximation algorithm for the weighted matching problem. This algorithm is both faster than any other suggested sequential 1/2-approximation algorithm on almost all inputs and when parallelized also scales better than previous multithreaded algorithms. We further extend this to a general scalable multithreaded algorithm that computes matchings of weight comparable with the best sequential deterministic algorithms. The performance of the suggested algorithms is documented through extensive experiments on different multithreaded architectures.

A Medium-Grain Method for Fast 2D Bipartitioning of Sparse Matrices

Daniël M. Pelt

Scientific Computing Group
Centrum Wiskunde & Informatica
Amsterdam, The Netherlands
D.M.Pelt@cwi.nl

Rob H. Bisseling

Mathematical Institute
Utrecht University
Utrecht, The Netherlands
R.H.Bisseling@uu.nl

We present a new hyper graph-based method, the medium-grain method, for solving the sparse matrix partitioning problem. This problem arises when distributing data for parallel sparse matrix-vector multiplication. In the medium-grain method, each matrix nonzero is assigned to either a row group or a column group, and these groups are represented by vertices of the hyper graph. For an m x n sparse matrix, the resulting hyper graph has m+n vertices and m+n hyper edges. Furthermore, we present an iterative refinement procedure for improvement of a given partitioning, based on the medium-grain method, which can be applied as a cheap but effective post processing step after any partitioning method. The medium-grain method is able to produce fully two-dimensional bipartitionings, but its computational complexity equals that of one-dimensional methods. Experimental results for a large set of sparse test matrices show that the medium-grain method with iterative refinement produces bipartitionings with lower communication volume compared to current state-of-the-art methods, and is faster at producing them.

Bipartite Matching Heuristics with Quality Guarantees on Shared Memory Parallel Computers

Fanny Dufossé

LAAS CNRS - Universit´e de Toulouse,
7 avenue du colonel Roche
F-31400 Toulouse, France
Email:fanny.dufosse@laas.fr

Kamer Kaya

The Ohio State University
Dept. of Biomedical Informatics
Columbus, OH, USA
Email: kamer@bmi.osu.edu

Bora Uçar

LIP (CNRS, ENS Lyon, UCBL,
INRIA, Universit´e de Lyon),
46, all´ee d'Italie, ENS Lyon,
Lyon F-69364, France

We propose two heuristics for the bipartite matching problem that are amenable to shared-memory parallelization. The first heuristic is very intriguing from parallelization perspective. It has no significant algorithmic synchronization overhead and no conflict resolution is needed across threads. We show that this heuristic has an approximation ratio of around 0.632. The second heuristic is designed to obtain a larger matching by employing the well-known Karp-Sipser heuristic on a judiciously chosen subgraph of the original graph. We show that the Karp-Sipser heuristic always finds a maximum cardinality matching in the chosen subgraph. Although the Karp-Sipser heuristic is hard to parallelize for general graphs, we exploit the structure of the selected sub graphs to propose a specialized implementation which demonstrates a very good scalability. Based on our experiments and theoretical evidence, we conjecture that this second heuristic obtains matchings with cardinality of at least 0.866 of the maximum cardinality. We discuss parallel implementations of the proposed heuristics on shared memory systems. Experimental results, for demonstrating speed-ups and verifying the theoretical results in practice, are provided.

BFS and Coloring-Based Parallel Algorithms for Strongly Connected Components and Related Problems

George M. Slota, Sivasankaran Rajamanickam†, and Kamesh Madduri**

∗Computer Science and Engineering, The Pennsylvania State University
Email: gslota@psu.edu, madduri@cse.psu.edu

†Scalable Algorithms Department, Sandia National Laboratories
Email: srajama@sandia.gov

Finding the strongly connected components (SCCs) of a directed graph is a fundamental graph-theoretic problem. Tarjan's algorithm is an efficient serial algorithm to find SCCs, but relies on the hard-to-parallelize depth-first search (DFS). We observe that implementations of several parallel SCC detection algorithms show poor parallel performance on modern multicore platforms and large-scale networks. This paper introduces the Multistep method, a new approach that avoids work inefficiencies seen in prior SCC approaches. It does not rely on DFS, but instead uses a combination of breadth-first search (BFS) and a parallel graph coloring routine. We show that the Multistep method scales well on several real-world graphs, with performance fairly independent of topological properties such as the size of the largest SCC and the total number of SCCs. On a 16-core Intel Xeon platform, our algorithm achieves a 20X speedup over the serial approach on a 2 billion edge graph, fully decomposing it in under two seconds. For our collection of test networks, we observe that the Multistep method is 1.92X faster (mean speedup) than the state-of-the-art Hong et al. SCC method. In addition, we modify the Multistep method to find connected and weakly connected components, as well as introduce a novel algorithm for determining articulation vertices of biconnected components. These approaches all utilize the same underlying BFS and coloring routines.

Session 14: Large Scale Scientific Applications

IPDPS 2014

Large-Scale Hydrodynamic Brownian Simulations on Multicore and Manycore Architectures

Xing Liu and Edmond Chow

School of Computational Science and Engineering
College of Computing, Georgia Institute of Technology
Atlanta, Georgia, 30332, USA
xing.liu@gatech.edu
echow@cc.gatech.edu

Conventional Brownian dynamics (BD) simulations with hydrodynamic interactions utilize 3n×3n dense mobility matrices, where n is the number of simulated particles. This limits the size of BD simulations, particularly on accelerators with low memory capacities. In this paper, we formulate a matrix-free algorithm for BD simulations, allowing us to scale to very large numbers of particles while also being efficient for small numbers of particles. We discuss the implementation of this method for multicore and many core architectures, as well as a hybrid implementation that splits the workload between CPUs and Intel Xeon Phi coprocessors. For 10,000 particles, the limit of the conventional algorithm on a 32 GB system, the matrix-free algorithm is 35 times faster than the conventional matrix based algorithm. We show numerical tests for the matrix-free algorithm up to 500,000 particles. For large systems, our hybrid implementation using two Intel Xeon Phi coprocessors achieves a speedup of over 3.5x compared to the CPU-only case. Our optimizations also make the matrix-free algorithm faster than the conventional dense matrix algorithm on as few as 1000 particles.

Using Load Balancing to Scalably Parallelize Sampling-Based Motion Planning Algorithms

Adam Fidel†, Sam Ade Jacobs, Shishir Sharma‡, Nancy M. Amato†, and Lawrence Rauchwerger†

†Parasol Lab, Dept. of Computer Science and Engineering,
Texas A&M University, 3112 TAMU, College Station, TX, USA
{fidel,amato,rwerger}@cse.tamu.edu

∗ABB Corporate Research, Raleigh, NC, USA
sam.jacobs@us.abb.com

‡Microsoft Corp., Redmond, WA, USA
shishir.sharma@microsoft.com

Motion planning, which is the problem of computing feasible paths in an environment for a movable object, has applications in many domains ranging from robotics, to intelligent CAD, to protein folding. The best methods for solving this PSPACE-hard problem are so-called sampling-based planners. Recent work introduced uniform spatial subdivision techniques for parallelizing sampling-based motion planning algorithms that scaled well. However, such methods are prone to load imbalance, as planning time depends on region characteristics and, for most problems, the heterogeneity of the sub problems increases as the number of processors increases. In this work, we introduce two techniques to address load imbalance in the parallelization of sampling-based motion planning algorithms: an adaptive work stealing approach and bulk-synchronous redistribution. We show that applying these techniques to representatives of the two major classes of parallel sampling-based motion planning algorithms, probabilistic roadmaps and rapidly-exploring random trees, results in a more scalable and load-balanced computation on more than 3,000 cores.

Petascale Application of a Coupled CPU-GPU Algorithm for Simulation and Analysis of Multiphase Flow Solutions in Porous Medium Systems

James E. McClure, Hao Wang†, Jan F. Prins‡, Cass T. Miller§, and Wu-Chun Feng†*

∗Advanced Research Computing
Virginia Tech, Blacksburg, Virginia
Email: mcclurej@vt.edu

‡Dept. of Computer Science
University of North Carolina at Chapel Hill,
Chapel Hill, North Carolina
Email:prins@cs.unc.edu

§Dept. of Environmental Science & Engineering
University of North Carolina at Chapel Hill,
Chapel Hill, North Carolina
Email:casey_miller@unc.edu

†Dept. of Computer Science
Virginia Tech, Blacksburg, Virginia
Email: {hwang121, wfeng}@vt.edu

Large-scale simulation can provide a wide range of information needed to develop and validate theoretical models for multiphase flow in porous medium systems. In this paper, we consider a coupled solution in which a multiphase flow simulator is coupled to an analysis approach used to extract the interfacial geometries as the flow evolves. This has been implemented using MPI to target heterogeneous nodes equipped with GPUs. The GPUs evolve the multiphase flow solution using the lattice Boltzmann method while the CPUs compute up scaled measures of the morphology and topology of the phase distributions and their rate of evolution. Our approach is demonstrated to scale to 4,096 GPUs and 65,536 CPU cores to achieve a maximum performance of 244,754 million-lattice-node updates per second (MLUPS) in double precision execution on Titan. In turn, this approach increases the size of systems that can be considered by an order of magnitude compared with previous work and enables detailed in situ tracking of averaged flow quantities at temporal resolutions that were previously impossible. Furthermore, it virtually eliminates the need for post-processing and intensive I/O and mitigates the potential loss of data associated with node failures.

A Spatio-temporal Coupling Method to Reduce the Time-to-Solution of Cardiovascular Simulations

Amanda Randles

Center for Applied Scientific Computing
Lawrence Livermore National Laboratory
Livermore, CA, USA
randles2@llnl.gov

Efthimios Kaxiras

Department of Physics and
School of Engineering and Applied Sciences
Harvard Universty
Cambridge, MA, USA
kaxiras@seas.harvard.edu

We present a new parallel-in-time method designed to reduce the overall time-to-solution of a patient-specific cardiovascular flow simulation. Using a modified Para real algorithm, our approach extends strong scalability beyond spatial parallelism with fully controllable accuracy and no decrease in stability. We discuss the coupling of spatial and temporal domain decompositions used in our implementation, and showcase the use of the method on a study of blood flow through the aorta. We observe an additional 40% reduction in overall wall clock time with no significant loss of accuracy, in agreement with a predictive performance model.

Session 15: Multicore and Transactional Memory

IPDPS 2014

Mitigating the Mismatch between the Coherence Protocol and Conflict Detection in Hardware Transactional Memory

Lihang Zhao[1], Lizhong Chen[2], and Jeffrey Draper[1]

[1]Information Sciences Institute
[2]Ming Hsieh Department of Electrical Engineering
University of Southern California
Email: flihangzh, lizhongcg@usc.edu, draper@isi.edu

Hardware Transactional Memory (HTM) usually piggybacks onto the cache coherence protocol to detect data access conflicts between transactions. We identify an intrinsic mismatch between the typical coherence scheme and transaction execution, which causes a sizable amount of unnecessary transaction aborts. This pathological behavior is called false aborting and increases the amount of wasted computation and on-chip communication. For the TM applications we studied, 41% of the transactional write requests incur false aborting. To combat false aborting, we propose Predictive Unicast and Notification (PUNO), a novel hardware mechanism to 1) replace the inefficient coherence multicast with a unicast scheme to prevent transactions from being disrupted unnecessarily and 2) restrain transaction polling through proactive notification. PUNO reduces transaction aborts by 61% and network traffic by 32% in workloads representative of future TM applications with a VLSI implementation area overhead of 0.41%.

Performance and Energy Analysis of the Restricted Transactional Memory Implementation on Haswell

Bhavishya Goel, Ruben Titos-Gil, Anurag Negi, Sally A. McKee, and Per Stenstrom

Chalmers University of Technology
Gothenburg, Sweden
{goelb, ruben.titos, negi, mckee, per.stenstrom}@chalmers.se

Hardware transactional memory implementations are becoming increasingly available. For instance, the Intel Core i7 4770 implements Restricted Transactional Memory (RTM) support for Intel Transactional Synchronization Extensions (TSX). In this paper, we present a detailed evaluation of RTM performance and energy expenditure. We compare RTM behavior to that of the TinySTM software transactional memory system, first by running micro benchmarks, and then by running the STAMP benchmark suite. We find that which system performs better depends heavily on the workload characteristics. We then conduct a case study of two STAMP applications to assess the impact of programming style on RTM performance and to investigate what kinds of software optimizations can help overcome RTM's hardware limitations.

Runtime-Guided Cache Coherence Optimizations in Multi-core Architectures

Madhavan Manivannan and Per Stenstrom

Chalmers University of Technology
Gothenburg, Sweden
{madhavan, per.stenstrom}@chalmers.se

Emerging task-based parallel programming models shield programmers from the daunting task of parallelism management by delegating the responsibility of mapping and scheduling of individual tasks to the runtime system. The runtime system can use semantic information about task dependencies supplied by the programmer and the mapping information of tasks to enable optimizations like data-flow based execution and locality-aware scheduling of tasks. However, should the cache coherence substrate have access to this information from the runtime system, it would enable aggressive optimizations of prevailing access patterns such as one-to-many producer-consumer sharing and migratory sharing. Such linkage has however not been studied before. We present a family of runtime guided cache coherence optimizations enabled by linking dependency and mapping information from the runtime system to the cache coherence substrate. By making this information available to the cache coherence substrate, we show that optimizations, such as downgrading and self-invalidation, that help reducing overheads associated with producer-consumer and migratory sharing can be supported with reasonable extensions to the baseline cache coherence protocol. Our experimental results establish that each optimization provides significant performance gain in isolation and can provide additional gains when combined. Finally, we evaluate these optimizations in the context of earlier proposed runtime-guided prefetching schemes and show that they can have synergistic effects.

High Performance Alltoall and Allgather Designs for InfiniBand MIC Clusters

Akshay Venkatesh, Sreeram Potluri, Raghunath Rajachandrasekar, Miao Luo, Khaled Hamidouche, and Dhabaleswar K. Panda

Network-Based Computing Laboratory
The Ohio State University
{akshay, potluri, rajachan, luom, hamidouc, panda}@cse.ohio-state.edu

Intel's Many-Integrated-Core (MIC) architecture aims to provide Teraflop throughput (through high degrees of parallelism) with a high FLOP/Watt ratio and x86 compatibility. However, this two-fold approach to solving power and programmability challenges for Exascale computing is constrained by certain architectural idiosyncrasies. MIC coprocessors have a memory constrained environment and its processors operate at slower clock rates. Also, being PCI devices, the communication characteristics of MIC co-processors are different compared to communication behavior seen in homogeneous environments. For instance, the performance of sending data from the MIC memory to a remote node's memory through message passing routines has 3x-6x higher latency than sending from the host processor memory. Hence communication libraries that do not consider these architectural subtleties are likely to nullify performance benefits or even cause degradation in applications that intend to use MICs and rely heavily on communication routines. The performance of Message Passing Interface (MPI) operations, especially dense collective operations like All-to-all and All gather, strongly affect the performance of many distributed parallel applications. In this paper, we revisit state-of-the-art algorithms commonly used to implement All-to-all collectives and propose adaptations and optimizations to alleviate architectural bottlenecks on MIC clusters. We also propose a few novel designs to improve the communication latency of these operations. Through micro-benchmarks and applications, we substantiate the benefits of incorporating the proposed adaptations to the All-to-All collective operations. At the micro-benchmark level, the proposed designs show as much as 79% improvement for All gather operation and up to 70% improvement for All-to-all and with the P3DFFT application, an improvement of 38% is seen in overall execution time.

Session 16: HPC Operating Systems and Runtime Systems

IPDPS 2014

HPMMAP: Lightweight Memory Management for Commodity Operating Systems

Brian Kocoloski and John Lange

Department of Computer Science
University of Pittsburgh
Pittsburgh, PA 15260
{briankoco,jacklange}@cs.pitt.edu

Linux-based operating systems and runtimes (OS/Rs) have emerged as the environments of choice for the majority of modern HPC systems. While Linux-based OS/Rs have advantages such as extensive feature sets as well as developer familiarity, these features come at the cost of additional overhead throughout the system. In contrast to Linux, there is a substantial history of work in the HPC community focused on lightweight OS/R architectures that provide scalable and consistent performance for tightly coupled HPC applications, but lack many of the features offered by commodity OS/Rs. In this paper, we propose to bridge the gap between LWKs and commodity OS/Rs by selectively providing a lightweight memory subsystem for HPC applications in a commodity OS/R environment. Our system HPMMAP provides isolated and low overhead memory performance transparently to HPC applications by bypassing Linux's memory management layer. Our approach is dynamically configurable at runtime, and adds no additional overheads nor requires any resources when not in use. We show that HPMMAP can decrease variance and reduce application runtime by up to 50%.

Victim Selection and Distributed Work Stealing Performance: A Case Study

Swann Perarnau

RIKEN AICS
Kobe, Japan
perarnau@riken.jp

Mitsuhisa Sato

University of Tsukuba/RIKEN AICS
Tsukuba, Japan
msato@cs.tsukuba.ac.jp

Work stealing is a popular solution to perform dynamic load balancing of irregular computations, both for shared memory and distributed memory systems. While shared memory performance of work stealing is well understood, distributing this algorithm to several thousands of nodes can introduce new performance issues. In particular, most studies of work stealing assume that all participating processes are equidistant from each other, in terms of communication latency. This paper presents a new performance evaluation of the popular UTS benchmark, in its work stealing implementation, on the scale of ten thousands of compute nodes. Taking advantage of the physical scale of the K Computer, we investigate in details the performance impact of communication latencies on work stealing. In particular, we introduce a new performance metric to assess the time needed by the work stealing scheduler to distribute work among all processes. Using this metric, we identify a previously overlooked issue: the victim selection function used by the work stealing application can severely impact its performance at large scale. To solve this issue, we introduce a new strategy taking into account the physical distance between nodes and achieve significant performance improvements.

Power-Efficient Multiple Producer-Consumer

Ramy Medhat

Dept. of Elec. and Comp. Eng.
University of Waterloo, Canada
Email: rmedhat@uwaterloo.ca

Borzoo Bonakdarpour

School of Computer Science
University of Waterloo, Canada
Email: borzoo@cs.uwaterloo.ca

Sebastian Fischmeister

Dept. of Elec. and Comp. Eng.
University of Waterloo, Canada
Email: sfischme@uwaterloo.ca

Power efficiency has been one of the main objectives of hardware design in the last two decades. However, with the recent explosion of mobile computing and the increasing demand for green data centers, software power efficiency has also risen to be an equally important factor. We argue that most classic concurrency control algorithms were designed in an era when power efficiency was not an important dimension in algorithm design. Such algorithms are applied to solve a wide range of problems from kernel-level primitives in operating systems to networking devices and web services. These primitives and services are constantly and heavily invoked in any computer system and by larger scale in networking devices and data centers. Thus, even a small change in their power spectrum can make a huge impact on overall power consumption in long periods of time. This paper focuses on the classic producer-consumer problem. First, we study the power efficiency of different existing implementations of the producer-consumer problem. In particular, we present evidence that these implementations behave drastically differently with respect to power consumption. Secondly, we present a dynamic algorithm for the multiple producer-consumer problem, where consumers in a multicore system use learning mechanisms to predict the rate of production, and effectively utilize this prediction to attempt to latch onto previously scheduled CPU wake-ups. Such group latching results in minimizing the overall number of CPU wakeups and in effect, power consumption. We enable consumers to dynamically reserve more pre-allocated memory in cases where the production rate is too high. Consumers may compete for the extra space and dynamically release it when it is no longer needed. Our experiments show that our algorithm provides up to 40% decrease in the number of CPU wakeups, and 30% decrease in power consumption. We validate the scalability of our algorithm with an increasing number of consumers.

Efficient Data Race Detection for C/C++ Programs Using Dynamic Granularity

Young Wn Song and Yann-Hang Lee

Computer Science and Engineering
Arizona State University
Tempe, AZ, 85281
ywsong@asu.edu, yhlee@asu.edu

To detect races precisely without false alarms, vector clock based race detectors can be applied if the overhead in time and space can be contained. This is indeed the case for the applications developed in object-oriented programming language where objects can be used as detection units. On the other hand, embedded applications, often written in C/C++, necessitate the use of fine-grained detection approaches that lead to significant execution overhead. In this paper, we present a dynamic granularity algorithm for vector clock based data race detectors. The algorithm exploits the fact that neigh boring memory locations tend to be accessed together and can share the same vector clock archiving dynamic granularity of detection. The algorithm is implemented on top of Fast Track and uses Intel PIN tool for dynamic binary instrumentation. Experimental results on benchmarks show that, on average, the race detection tool using the dynamic granularity algorithm is 43% faster than the Fast Track with byte granularity and is with 60% less memory usage. Comparison with existing industrial tools, Val grind DRD and Intel Inspector XE, also suggests that the proposed dynamic granularity approach is very viable.

Session 17: Algorithms for Distributed Computing

IPDPS 2014

Improved Time Bounds for Linearizable Implementations of Abstract Data Types

Jiaqi Wang, Edward Talmage, Hyunyoung Lee, and Jennifer L. Welch

Department of Computer Science and Engineering
Texas A&M University
College Station, Texas, USA
adah1002@gmail.com, {etalmage, hlee, welch}@cse.tamu.edu

Linearizability is a well-known consistency condition for shared objects in concurrent systems. We focus on the problem of implementing linearizable objects of arbitrary data types in message-passing systems with bounded, but uncertain, message delay and bounded, but non-zero, clock skew. We present an algorithm that exploits axiomatic properties of different operations to reduce the running time of each operation below that obtainable with previously known algorithms. We also prove lower bounds on the time complexity of various kinds of operations, specified by the axioms they satisfy, resulting in reduced gaps in some cases and tight bounds in others.

DEX: Self-Healing Expanders

Gopal Pandurangan and Peter Robinson

Div. of Mathematical Sciences
Nanyang Technological University
Singapore 637371
gopalpandurangan@gmail.com
peter.robinson@ntu.edu.sg

Amitabh Trehan

School of Electronics, Electrical
Engineering and Computer Sciences
Queen's University Belfast
Belfast, NI, United Kingdom
a.trehan@qub.ac.uk

We present a fully-distributed self-healing algorithm DEX, that maintains a constant degree expander network in a dynamic setting. To the best of our knowledge, our algorithm provides the first efficient distributed construction of expanders — whose expansion properties hold deterministically — that works even under an all-powerful adaptive adversary that controls the dynamic changes to the network (the adversary has unlimited computational power and knowledge of the entire network state, can decide which nodes join and leave and at what time, and knows the past random choices made by the algorithm). Previous distributed expander constructions typically provide only probabilistic guarantees on the network expansion which rapidly degrade in a dynamic setting, in particular, the expansion properties can degrade even more rapidly under adversarial insertions and deletions. Our algorithm provides efficient maintenance and incurs a low overhead per insertion/deletion by an adaptive adversary: only $O(log n)$ rounds and $O(log n)$ messages are needed with high probability (n is the number of nodes currently in the network). The algorithm requires only a constant number of topology changes. Moreover, our algorithm allows for an efficient implementation and maintenance of a distributed hash table (DHT) on top of DEX, with only a constant additional overhead. Our results are a step towards implementing efficient self-healing networks that have guaranteed properties (constant bounded degree and expansion) despite dynamic changes.

Fair Maximal Independent Sets

Jeremy T. Fineman, Calvin Newport, Micah Sherr, and Tonghe Wang

Department of Computer Science
Georgetown University, Washington, DC, U.S.A.
{jfineman, cnewport, msherr}@cs.georgetown.edu and tw473@georgetown.edu

Finding a maximal independent set (MIS) is a classic problem in graph theory that has been widely studied in the context of distributed algorithms. Standard distributed solutions to the MIS problem focus on time complexity. In this paper, we also consider fairness. For a given MIS algorithm A and graph G, we define the inequality factor for A on G to be the largest ratio between the probabilities of the nodes joining an MIS in the graph. We say an algorithm is fair with respect to a family of graphs if it achieves a constant inequality factor for all graphs in the family. In this paper, we seek efficient and fair algorithms for common graph families. We begin by describing an algorithm that is fair and runs in O(log* n)-time in rooted trees of size n. Moving to unrooted trees, we describe a fair algorithm that runs in O(log n) time. Generalizing further to bipartite graphs, we describe a third fair algorithm that requires O(log2 n) rounds. We also show a fair algorithm for planar graphs that runs in O(log2 n) rounds, and describe an algorithm that can be run in any graph, yielding good bounds on inequality in regions that can be efficiently colored with a small number of colors. We conclude our theoretical analysis with a lower bound that identifies a graph where all MIS algorithms achieve an inequality bound in Ω(n)—eliminating the possibility of an MIS algorithm that is fair in all graphs. Finally, to motivate the need for provable fairness guarantees, we simulate both our tree algorithm and Luby's MIS algorithm [13] in a variety of different tree topologies—some synthetic and some derived from real world data. Whereas our algorithm always yield an inequality factor ≤ 3.25 in these simulations, Luby's algorithms yields factors as large as 168.

Session 18: Milestones at the Petascale

IPDPS 2014

Balancing CPU-GPU Collaborative High-Order CFD Simulations on the Tianhe-1A Supercomputer

Chuanfu Xu, Lilun Zhang*, Xiaogang Deng*, Jianbin Fang†, Guangxue Wang‡, Wei Cao*, Yonggang Che*, Yongxian Wang*, and Wei Liu**

∗College of Computer Science
National University of Defense Technology, Changsha 410073, P.R. China

†Parallel and Distributed Systems Group
Delft University of Technology, Delft 2628CD, The Netherlands

‡State Key Laboratory of Aerodynamics
P.O. Box 211, Mianyang 621000, P.R. China

HOSTA is an in-house high-order CFD software that can simulate complex flows with complex geometries. Large scale high-order CFD simulations using HOSTA require massive HPC resources, thus motivating us to port it onto modern GPU accelerated supercomputers like Tianhe-1A. To achieve a greater speedup and fully tap the potential of Tianhe-1A, we collaborate CPU and GPU for HOSTA instead of using a naive GPU-only approach. We present multiple novel techniques to balance the loads between the store-poor GPU and the store-rich CPU, and overlap the collaborative computation and communication as far as possible. Taking CPU and GPU load balance into account, we improve the maximum simulation problem size per Tianhe-1A node for HOSTA by 2.3X, meanwhile the collaborative approach can improve the performance by around 45% compared to the GPU-only approach. Scalability tests show that HOSTA can achieve a parallel efficiency of above 60% on 1024 Tianhe-1A nodes. With our method, we have successfully simulated China's large civil airplane configuration C919 containing 150M grid cells. To our best knowledge, this is the first paper that reports a CPUGPU collaborative high-order accurate aerodynamic simulation result with such a complex grid geometry.

Shedding Light on Lithium/Air Batteries Using Millions of Threads on the BG/Q Supercomputer

Valéry Weber, Costas Bekas, Teodoro Laino, and Alessandro Curioni

IBM Research – Zurich, Säumerstrasse 4, 8803 Rüschlikon, Switzerland
Email: {vwe,teo,bek,cur}@zurich.ibm.com

Adam Bertsch and Scott Futral

Lawrence Livermore National Laboratory,
7000 East Avenue, Livermore, CA 94550, USA
Email: {bertsch2,futral2}@llnl.gov

In this work, we present a novel parallelization scheme for a highly efficient evaluation of the Hartree-Fock exact exchange (HFX) in ab initio molecular dynamics simulations, specifically tailored for condensed phase simulations. Our developments allow one to achieve the necessary accuracy for the evaluation of the HFX in a highly controllable manner. We show here that our solutions can take great advantage of the latest trends in HPC platforms, such as extreme threading, short vector instructions and highly dimensional interconnection networks. Indeed, all these trends are evident in the IBM Blue Gene/Q supercomputer. We demonstrate an unprecedented scalability up to 6,291,456 threads (96 BG/Q racks) with a near perfect parallel efficiency, which represents a more than 20-fold improvement as compared to the current state of the art. In terms of reduction of time to solution, we achieved an improvement that can surpass a 10-fold decrease in runtime with respect to directly comparable approaches. We exploit this development to enhance the accuracy of DFT based molecular dynamics by using the PBE0 hybrid functional. This approach allowed us to investigate the chemical behavior of organic solvents in one of the most challenging research topics in energy storage, lithium/air batteries, and to propose alternative solvents with enhanced stability to ensure an appropriate reversible electrochemical reaction. This step is key for the development of a viable lithium/air storage technology, which would have been a daunting computational task using standard methods. Recent research has shown that the electrolyte plays a key role in non-aqueous lithium/air batteries in producing the appropriate reversible electrochemical reduction. In particular, the chemical degradation of propylene carbonate, the typical electrolyte used, by lithium peroxide has been demonstrated by molecular dynamics simulations of highly realistic models. Reaching the necessary high accuracy in these simulations is a daunting computational task using standard methods.

Enabling and Scaling a Global Shallow-Water Atmospheric Model on Tianhe-2

Wei Xue, Chao Yang†, Haohuan Fu‡, Xinliang Wang*‡, Yangtong Xu*‡, Lin Gan*‡, Yutong Lu¶, and Xiaoqian Zhu¶*

∗Dept. of Computer Science & Technology
Tsinghua University, Beijing 100084, China

†Institute of Software
Chinese Academy of Sciences, Beijing 100190, China

‡Ministry of Education Key Laboratory for Earth System Modeling,
and Center for Earth System Science
Tsinghua University, Beijing 100084, China

§State Key Laboratory of Computer Science
Chinese Academy of Sciences, Beijing 100190, China

¶Dept. of Computer Science & Technology
National University of Defense Technology, Changsha, Hunan 410073, China

This paper presents a hybrid algorithm for the petascale global simulation of atmospheric dynamics on Tianhe-2, the world's current top-ranked supercomputer developed by China's National University of Defense Technology (NUDT). Tianhe-2 is equipped with both Intel Xeon CPUs and Intel Xeon Phi accelerators. A key idea of the hybrid algorithm is to enable flexible domain partition between an arbitrary number of processors and accelerators, so as to achieve a balanced and efficient utilization of the entire system. We also present an asynchronous and concurrent data transfer scheme to reduce the communication overhead between CPU and accelerators. The acceleration of our global atmospheric model is conducted to improve the use of the Intel MIC architecture. For the single-node test on Tianhe-2 against two Intel Ivy Bridge CPUs (24 cores), we can achieve 2.07x, 3.18x, and 4.35x speedups when using one, two, and three Intel Xeon Phi accelerators respectively. The average performance gain from SIMD vectorization on the Intel Xeon Phi processors is around 5x (out of the 8x theoretical case). Based on successful computation-communication overlapping, large-scale tests indicate that a nearly ideal weak-scaling efficiency of 93.5% is obtained when we gradually increase the number of nodes from 6 to 8,664 (nearly 1.7 million cores). In the strong-scaling test, the parallel efficiency is about 77% when the number of nodes increases from 1,536 to 8,664 for a fixed 65,664 × 65, 664 × 6 mesh with 77.6 billion unknowns.

Overcoming the Scalability Challenges of Epidemic Simulations on Blue Waters

Jae-Seung Yeom[1, 2], Abhinav Bhatele[3], Keith Bisset[2], Eric Bohm[4], Abhishek Gupta[4], Laxmikant V. Kale[4], Madhav Marathe[1, 2], Dimitrios S. Nikolopoulos[5], Martin Schulz[3], and Lukasz Wesolowski[4]

[1]Department of Computer Science
Virginia Tech, Blacksburg, VA 24061 USA

[2]Virginia Bioinformatics Institute
Virginia Tech, Blacksburg, VA 24061 USA
{jyeom, kbisset, mmarathe}@vbi.vt.edu

[3]Center for Applied Scientific Computing
Lawrence Livermore National Laboratory, Livermore, CA 94551 USA
{bhatele, schulzm}@llnl.gov

[4]Department of Computer Science
University of Illinois at Urbana-Champaign, Urbana, IL 61801 USA
{ebohm, gupta59, kale, wesolwsk}@illinois.edu,

[5]School of EEECS
Queen's University of Belfast, Belfast, Northern Ireland BT9 5BN UK
d.nikolopoulos@qub.ac.uk

Modeling dynamical systems represents an important application class covering a wide range of disciplines including but not limited to biology, chemistry, finance, national security, and health care. Such applications typically involve large-scale, irregular graph processing, which makes them difficult to scale due to the evolutionary nature of their workload, irregular communication and load imbalance. EpiSimdemics is such an application simulating epidemic diffusion in extremely large and realistic social contact networks. It implements a graph-based system that captures dynamics among co-evolving entities. This paper presents an implementation of EpiSimdemics in Charm++ that enables future research by social, biological and computational scientists at unprecedented data and system scales. We present new methods for application-specific processing of graph data and demonstrate the effectiveness of these methods on a Cray XE6, specifically NCSA's Blue Waters system.

Session 19: Storage and Reliability

IPDPS 2014

POD: Performance Oriented I/O Deduplication for Primary Storage Systems in the Cloud

Bo Mao†, Hong Jiang‡, Suzhen Wu, and Lei Tian‡*

†Software School of Xiamen University, China

‡Department of Computer Science and Engineering
University of Nebraska-Lincoln, USA

∗Corresponding author, Computer Science Department of Xiamen University, China
{maobo, suzhen}@xmu.edu.cn, {jiang, tian}@cse.unl.edu

Recent studies have shown that moderate to high data redundancy clearly exists in primary storage systems in the Cloud. Our experimental studies reveal that data redundancy exhibits a much higher level of intensity on the I/O path than that on disks due to the relatively high temporal access locality associated with small I/O requests to redundant data. On the other hand, we also observe that directly applying data deduplication to primary storage systems in the Cloud will likely cause space contention in memory and data fragmentation on disks. Based on these observations, we propose a Performance-Oriented I/O Deduplication approach, called POD, rather than a capacity-oriented I/O deduplication approach, represented by iDedup, to improve the I/O performance of primary storage systems in the Cloud without sacrificing capacity savings of the latter. The salient feature of POD is its focus on not only the capacity-sensitive large writes and files, as in iDedup, but also the performance-sensitive while capacity-insensitive small writes and files. The experiments conducted on our lightweight prototype implementation of POD show that POD significantly outperforms iDedup in the I/O performance measure by up to 87.9% with an average of 58.8%. Moreover, our evaluation results also show that POD achieves comparable or better capacity savings than iDedup.

Pipelined Compaction for the LSM-Tree

Zigang Zhang†§, Yinliang Yue†, Bingsheng He‡, Jin Xiong†, Mingyu Chen†, Lixin Zhang†, and Ninghui Sun†

†SKL Computer Architecture, ICT, CAS

§University of Chinese Academy of Sciences

‡Nanyang Technological University

Write-optimized data structures like Log-Structured Merge-tree (LSM-tree) and its variants are widely used in key-value storage systems like Big Table and Cassandra. Due to deferral and batching, the LSM-tree based storage systems need background compactions to merge key-value entries and keep them sorted for future queries and scans. Background compactions play a key role on the performance of the LSM-tree based storage systems. Existing studies about the background compaction focus on decreasing the compaction frequency, reducing I/Os or confining compactions on hot data key-ranges. They do not pay much attention to the computation time in background compactions. However, the computation time is no longer negligible, and even the computation takes more than 60% of the total compaction time in storage systems using flash based SSDs. Therefore, an alternative method to speedup the compaction is to make good use of the parallelism of underlying hardware including CPUs and I/O devices. In this paper, we analyze the compaction procedure, recognize the performance bottleneck, and propose the Pipelined Compaction Procedure (PCP) to better utilize the parallelism of CPUs and I/O devices. Theoretical analysis proves that PCP can improve the compaction bandwidth. Furthermore, we implement PCP in real system and conduct extensive experiments. The experimental results show that the pipelined compaction procedure can increase the compaction bandwidth and storage system throughput by 77% and 62% respectively.

EDM: An Endurance-Aware Data Migration Scheme for Load Balancing in SSD Storage Clusters

Jiaxin Ou, Jiwu Shu, Youyou Lu, Letian Yi, and Wei Wang*

Department of Computer Science and Technology
Tsinghua University, Beijing, China

Tsinghua National Laboratory for Information Science and Technology, Beijing, China
*Corresponding author: shujw@tsinghua.edu.cn
{ojx11, luyy09, yilt09, wangwei11}@mails.tsinghua.edu.cn

Data migration schemes are critical to balance the load in storage clusters for performance improvement. However, as NAND flash based SSDs are widely deployed in storage systems, extending the lifespan of SSD storage clusters becomes a new challenge for data migration. Prior approaches designed for HDD storage clusters, however, are inefficient due to excessive write amplification during data migration, which significantly decrease the lifespan of SSD storage clusters. To overcome this problem, we propose EDM, an endurance aware data migration scheme with careful data placement and movement to minimize the data migrated, so as to limit the worn-out of SSDs while improving the performance. Based on the observation that performance degradation is dominated by the wear speed of an SSD, which is affected by both the storage utilization and the write intensity, two complementary data migration policies are designed to explore the trade-offs among throughput, response time during migration, and lifetime of SSD storage clusters. Moreover, we design an SSD wear model and quantitatively calculate the amount of data migrated as well as the sources and destinations of the migration, so as to reduce the write amplification caused by migration. Results on a real storage cluster using real-world traces show that EDM performs favorably versus existing HDD based migration techniques, reducing cluster-wide aggregate erase count by up to 40%. In the meantime, it improves the performance by 25% on average compared to the baseline system which achieves almost the same effectiveness of performance improvement as previous migration techniques.

Session 20: Map/Reduce and Big Data

IPDPS 2014

Characterization and Optimization of Memory-Resident MapReduce on HPC Systems

Yandong Wang, Robin Goldstone†, Weikuan Yu*, and Teng Wang**

Auburn University*
Lawrence Livermore National Laboratory†
fwangyd,wkyu,tzw0019g@auburn.edu fgoldstone1g@llnl.gov

MapReduce is a widely accepted framework for addressing big data challenges. Recently, it has also gained broad attention from scientists at the U.S. leadership computing facilities as a promising solution to process gigantic simulation results. However, conventional high-end computing systems are constructed based on the compute-centric paradigm while big data analytics applications prefer a data-centric paradigm such as MapReduce. This work characterizes the performance impact of key differences between compute- and data-centric paradigms and then provides optimizations to enable a dual-purpose HPC system that can efficiently support conventional HPC applications and new data analytics applications. Using a state-of-the-art MapReduce implementation Spark and the Hyperion system at Lawrence Livermore National Laboratory, we have examined the impact of storage architectures, data locality and task scheduling to the memory-resident MapReduce jobs. Based on our characterization and findings of the performance behaviors, we have introduced two optimization techniques, namely Enhanced Load Balancer and Congestion-Aware Task Dispatching, to improve the performance of Spark applications.

MIC-SVM: Designing a Highly Efficient Support Vector Machine for Advanced Modern Multi-core and Many-Core Architectures

Yang You||, Shuaiwen Leon Song†, Haohuan Fu*, Andres Marquez†, Maryam Mehri Dehnavi‡, Kevin Barker†, Kirk W. Cameron§, Amanda Peters Randles¶, and Guangwen Yang**

*Ministry of Education Key Laboratory for Earth System Modeling, Center for Earth Science, Tsinghua University, Beijing, China

||Tsinghua National Laboratory for Information Science and Technology (TNList), Tsinghua University, Beijing, China
you-y12@mails.tsinghua.edu.cn, fhaohuan, ygwg@tsinghua.edu.cn

†Pacific Northwest National Lab, Richland, WA, USA

‡Computer Science and Artificial Intelligence Laboratory, MIT, Cambridge, MA, USA

§Virignia Tech, Blacksburg, VA, USA

¶Lawrence Livermore National Lab, Livermore, CA, USA

Support Vector Machine (SVM) has been widely used in data-mining and Big Data applications as modern commercial databases start to attach an increasing importance to the analytic capabilities. In recent years, SVM was adapted to the field of High Performance Computing for power/performance prediction, auto-tuning, and runtime scheduling. However, even at the risk of losing prediction accuracy due to insufficient runtime information, researchers can only afford to apply offline model training to avoid significant runtime training overhead. Advanced multi- and many-core architectures offer massive parallelism with complex memory hierarchies which can make runtime training possible, but form a barrier to efficient parallel SVM design. To address the challenges above, we designed and implemented MIC-SVM, a highly efficient parallel SVM for x86 based multi-core and many-core architectures, such as the Intel Ivy Bridge CPUs and Intel Xeon Phi co-processor (MIC). We propose various novel analysis methods and optimization techniques to fully utilize the multilevel parallelism provided by these architectures and serve as general optimization methods for other machine learning tools. MIC-SVM achieves 4.4-84x and 18-47x speedups against the popular LIBSVM, on MIC and Ivy Bridge CPUs respectively, for several real-world data-mining datasets. Even compared with GPUSVM, run on a top of the line NVIDIA k20x GPU, the performance of our MIC-SVM is competitive. We also conduct a cross-platform performance comparison analysis, focusing on Ivy Bridge CPUs, MIC and GPUs, and provide insights on how to select the most suitable advanced architectures for specific algorithms and input data patterns.

BigKernel—High Performance CPU-GPU Communication Pipelining for Big Data-Style Applications

Reza Mokhtari and Michael Stumm

Department of Electrical and Computer Engineering
University of Toronto
Toronto, Canada
{mokhtari, stumm}@eecg.toronto.edu

GPUs offer an order of magnitude higher compute power and memory bandwidth than CPUs. GPUs therefore might appear to be well suited to accelerate computations that operate on voluminous data sets in independent ways, e.g., for transformations, filtering, aggregation, partitioning or other "Big Data" style processing. Yet experience indicates that it is difficult, and often error-prone, to write GPGPU programs which efficiently process data that does not fit in GPU memory, partly because of the intricacies of GPU hardware architecture and programming models, and partly because of the limited bandwidth available between GPUs and CPUs. In this paper, we propose Big Kernel, a scheme that provides pseudo-virtual memory to GPU applications and is implemented using a 4-stage pipeline with automated prefetching to (i) optimize CPU-GPU communication and (ii) optimize GPU memory accesses. Big Kernel simplifies the programming model by allowing programmers to write kernels using arbitrarily large data structures that can be partitioned into segments where each segment is operated on independently, these kernels are transformed into Big Kernel using straight-forward compiler transformations. Our evaluation on six data-intensive benchmarks shows that Big Kernel achieves an average speedup of 1.7 over state-of-the-art double-buffering techniques and an average speedup of 3.0 over corresponding multi-threaded CPU implementations.

DataMPI: Extending MPI to Hadoop-Like Big Data Computing

Xiaoyi Lu, Fan Liang, Bing Wang, Li Zha, and Zhiwei Xu

Institute of Computing Technology, Chinese Academy of Sciences
{luxiaoyi, liangfan, wangbing, char, zxu}@ict.ac.cn

MPI has been widely used in High Performance Computing. In contrast, such efficient communication support is lacking in the field of Big Data Computing, where communication is realized by time consuming techniques such as HTTP/RPC. This paper takes a step in bridging these two fields by extending MPI to support Hadoop-like Big Data Computing jobs, where processing and communication of a large number of key-value pair instances are needed through distributed computation models such as MapReduce, Iteration, and Streaming. We abstract the characteristics of key-value communication patterns into a bipartite communication model, which reveals four distinctions from MPI: Dichotomic, Dynamic, Data-centric, and Diversified features. Utilizing this model, we propose the specification of a minimalistic extension to MPI. An open source communication library, DataMPI, is developed to implement this specification. Performance experiments show that DataMPI has significant advantages in performance and flexibility, while maintaining high productivity, scalability, and fault tolerance of Hadoop.

Session 20: Map/Reduce and Big Data

IPDPS 2014

An Efficient Method for Stream Semantics over RDMA

Patrick MacArthur and Robert D. Russell

Department of Computer Science
University of New Hampshire
Durham, New Hampshire 03824-3591, USA
{pio3,rdr}@cs.unh.edu

Most network applications today are written to use TCP/IP via sockets. Remote Direct Memory Access (RDMA) is gaining popularity because its zero-copy, kernel-bypass features provide a high throughput, low latency reliable transport. Unlike TCP, which is a stream-oriented protocol, RDMA is a message-oriented protocol, and the OFA verbs library for writing RDMA application programs is more complex than the TCP sockets interface. UNH EXS is one of several libraries designed to give applications more convenient, high-level access to RDMA features. Recent work has shown that RDMA is viable both in the data center and over distance. One potential bottleneck in libraries that use RDMA is the requirement to wait for message advertisements in order to send large zero-copy messages. By sending messages first to an internal, hidden buffer and copying the message later, latency can be reduced at the expense of higher CPU usage at the receiver. This paper presents a communication algorithm that has been implemented in the UNH EXS stream-oriented mode to allow dynamic switching between sending transfers directly to user memory and sending transfers indirectly via an internal, hidden buffer depending on the state of the sender and receiver. Based on preliminary results, we see that this algorithm performs well under a variety of application requirements.

Collaborative Network Configuration in Hybrid Electrical/Optical Data Center Networks

Zhiyang Guo and Yuanyuan Yang

Department of Electrical and Computer Engineering
Stony Brook University
Stony Brook, NY 11794, USA

Recently, there has been much effort on introducing optical fiber communication to data center networks (DCNs) because of its significant advantage in bandwidth capacity and power efficiency. However, due to limitations of optical switching technologies, optical networking alone has not yet been able to accommodate the volatile data center traffic. As a result, hybrid packet/circuit (Hypac) switched DCNs, which argument the electrical packet switched (EPS) network with an optical circuit switched (OCS) network, have been proposed to combine the strengths of both types of networks. However, one problem with current Hypac DCNs is that the EPS network is shared in a best effort fashion and is largely oblivious to the accompanying OCS network, which results in severe drawbacks, such as degraded network predictability and deficiency in handling correlated traffic. Since the OCS/EPS networks have unique strengths and weaknesses, and are best suited for different traffic patterns, coordinating and collaborating the configuration of both networks is critical to reach the full potential of Hypac DCNs, which motivates the study in this paper. First, we present a network model that accurately abstracts the essential characteristics of the EPS/OCS networks. Second, considering the recent advances in network control technology, we propose a time-efficient algorithm called Collaborative Bandwidth Allocation (CBA) that configures both networks in a complementary manner. Finally, we conduct comprehensive simulations, which demonstrate that CBA significantly improves the performance of Hypac DCNs in many aspects.

Optimizing Bandwidth Allocation in Flex-Grid Optical Networks with Application to Scheduling

Hadas Shachnai, Ariella Voloshin, and Shmuel Zaks

Department of Computer Science
Technion
Haifa, ISRAEL
Email: [hadas,variella,zaks]@cs.technion.ac.il

All-optical networks have been largely investigated due to their high data transmission rates. In the traditional Wavelength-Division Multiplexing (WDM) technology, the spectrum of light that can be transmitted through the optical fiber has been divided into frequency intervals of fixed width, with a gap of unused frequencies between them. Recently, an alternative emerging architecture was suggested which moves away from the rigid Dense WDM (DWDM) model towards a flexible model, where usable frequency intervals are of variable width (even within the same link). Each light path has to be assigned a frequency interval (sub-spectrum), which remains fixed through all of the links it traverses. Two different light paths using the same link must be assigned disjoint sub-spectra. This technology is termed flex-grid (or, flex-spectrum), as opposed to fixed-grid (or, fixed-spectrum) current technology. In this work we study a problem of optimal bandwidth allocation arising in the flex-grid technology. In this setting, each light path has a lower and upper bound on the width of its frequency interval, as well as an associated profit, and we want to find a bandwidth assignment that maximizes the total profit. This problem is known to be NP-Complete. We observe that, in fact, the problem is inapproximable within any constant ratio even on a path network. We further derive NP-hardness results and present approximation algorithms for several special cases of the path and ring networks, which are of practical interest. Finally, while in general our problem is hard to approximate, we show that an optimal solution can be obtained by allowing resource augmentation. Our study has applications also in real time scheduling.

Balancing On-Chip Network Latency in Multi-application Mapping for Chip-Multiprocessors

Di Zhu, Lizhong Chen, Siyu Yue, Timothy M. Pinkston, and Massoud Pedram

Ming Hsieh Department of Electrical Engineering
University of Southern California
Email: {dizhu, lizhongc, siyuyue, tpink, pedram}@usc.edu

As the number of cores continues to grow in chip multiprocessors (CMPs), application-to-core mapping algorithms that leverage the non-uniform on-chip resource access time have been receiving increasing attention. However, existing mapping methods for reducing overall packet latency cannot meet the requirement of balanced on-chip latency when multiple applications are present. In this paper, we address the looming issue of balancing minimized on-chip packet latency with performance-awareness in the multi-application mapping of CMPs. Specifically, the proposed mapping problem is formulated, its NP-completeness is proven, and an efficient heuristic-based algorithm for solving the problem is presented. Simulation results show that the proposed algorithm is able to reduce the maximum average packet latency by 10.42% and the standard deviation of packet latency by 99.65% among concurrently running applications and, at the same time, incur little degradation in the overall performance.

Keynote Speaker 3

IPDPS 2014

Astrophysical Applications of Machine Learning at Scale and under Duress

Joshua Bloom

University of California, Berkeley

The universe is teeming with change on timescales from billions of years to milliseconds. A major goal of modern synoptic imaging surveys is to categorize this change over the entire sky to infer the diverse physical origins of variability. However, event discovery is only the beginning in the quest to extract the deepest insights: expensive follow-up resources (telescopes and people) are required, often in a time constrained environment. Viewing discovery and scientific insight through a resource-maximization lens, I discuss how machine learning is being applied to some modern astrophysics challenges. Here, the surfacing of parallelized feature engineering and machine learning into production-quality (scalable and fault tolerant) frameworks is the frontier for our field.

Best Papers Session

IPDPS 2014

Scalable Single Source Shortest Path Algorithms for Massively Parallel Systems

Venkatesan T. Chakaravarthy, Fabio Checconi†, Fabrizio Petrini†, and Yogish Sabharwal**

* IBM Research - India, New Delhi. {vechakra,ysabharwal}@in.ibm.com

† IBM T J Watson Research Center, USA. {fchecco,fpetrin}@us.ibm.com

In the single-source shortest path (SSSP) problem, we have to find the shortest paths from a source vertex v to all other vertices in a graph. In this paper, we introduce a novel parallel algorithm, derived from the Bellman-Ford and Delta-stepping algorithms. We employ various pruning techniques, such as edge classification and direction-optimization, to dramatically reduce inter-node communication traffic, and we propose load balancing strategies to handle higher-degree vertices. The extensive performance analysis shows that our algorithms work well on scale-free and real-world graphs. In the largest tested configuration, an R-MAT graph with 238 vertices and 242 edges on 32,768 Blue Gene/Q nodes, we have achieved a processing rate of three Trillion Edges Per Second (TTEPS), a four orders of magnitude improvement over the best published results.

A New Scalable Parallel Algorithm for Fock Matrix Construction

Xing Liu, Aftab Patel, and Edmond Chow

School of Computational Science and Engineering
College of Computing, Georgia Institute of Technology
Atlanta, Georgia, 30332, USA
xing.liu@gatech.edu, aypatel@gatech.edu, echow@cc.gatech.edu

Hartree-Fock (HF) or self-consistent field (SCF) calculations are widely used in quantum chemistry, and are the starting point for accurate electronic correlation methods. Existing algorithms and software, however, may fail to scale for large numbers of cores of a distributed machine, particularly in the simulation of moderately-sized molecules. In existing codes, HF calculations are divided into tasks. Fine-grained tasks are better for load balance, but coarse-grained tasks require less communication. In this paper, we present a new parallelization of HF calculations that addresses this trade-off: we use fine grained tasks to balance the computation among large numbers of cores, but we also use a scheme to assign tasks to processes to reduce communication. We specifically focus on the distributed construction of the Fock matrix arising in the HF algorithm, and describe the data access patterns in detail. For our test molecules, our implementation shows better scalability than NWChem for constructing the Fock matrix.

ReDHiP: Recalibrating Deep Hierarchy Prediction for Energy Efficiency

Xun Li[1], Diana Franklin[2], Ricardo Bianchini[3], Frederic T. Chong[2]

[1]Facebook
Menlo Park, CA, USA
xun@fb.com

[2]University of California, Santa Barbara
Santa Barbara, CA, USA
{franklin, chong}@cs.ucsb.edu

[3]Rutgers University
Piscataway, NJ, USA
ricardob@cs.rutgers.edu

Recent hardware trends point to increasingly deeper cache hierarchies. In such hierarchies, accesses that lookup and miss in every cache involve significant energy consumption and degraded performance. To mitigate these problems, in this paper we propose Recalibrating Deep Hierarchy Prediction (ReDHiP), an architectural mechanism that predicts last-level cache (LLC) misses in advance. An LLC miss means that all cache levels need not be accessed at all. Our design for ReDHiP focuses on a simple, compact prediction table that can be efficiently recalibrated over time. We find that a simpler scheme, while sacrificing accuracy, can be more accurate per bit than more complex schemes through recalibration. Our evaluation shows that ReDHiP achieves an average of 22% cache energy savings and 8% performance improvement for a wide range of benchmarks. ReDHiP achieves these benefits at a hardware cost of less than 1% of the LLC. We also demonstrate how ReDHiP can be used to reduce the energy overhead of hardware data prefetching while being able to further improve the performance.

F2C2-STM: Flux-Based Feedback-Driven Concurrency Control for STMs

Kaushik Ravichandran

College of Computing
Georgia Institute of Technology
Atlanta, USA
kaushikr@gatech.edu

Santosh Pande

College of Computing
Georgia Institute of Technology
Atlanta, USA
santosh@cc.gatech.edu

Software Transactional Memory (STM) systems provide an easy to use programming model for concurrent code and have been found suitable for parallelizing many applications providing performance gains with minimal programmer effort. With increasing core counts on modern processors one would expect increasing benefits. However, we observe that running STM applications on higher core counts is sometimes, in fact, detrimental to performance. This is due to the larger number of conflicts that arise with a larger number of parallel cores. As the number of cores available on processors steadily rise, a larger number of applications are beginning to exhibit these characteristics. In this paper we propose a novel dynamic concurrency control technique which can significantly improve performance (up to 50%) as well as resource utilization (up to 85%) for these applications at higher core counts. Our technique uses ideas borrowed from TCP's network congestion control algorithm and uses self-induced concurrency fluctuations to dynamically monitor and match varying concurrency levels in applications while minimizing global synchronization. Our flux-based feedback-driven concurrency control technique is capable of fully recovering the performance of the best statically chosen concurrency specification (as chosen by an oracle) regardless of the initial specification for several real world applications. Further, our technique can actually improve upon the performance of the oracle chosen specification by more than 10% for certain applications through dynamic adaptation to available parallelism. We demonstrate our approach on the STAMP benchmark suite while reporting significant performance and resource utilization benefits. We also demonstrate significantly better performance when comparing against state of the art concurrency control and scheduling techniques. Further, our technique is programmer friendly as it requires no changes to application code and no offline phases.

Session 22: Performance Characterization and Optimization

IPDPS 2014

Identifying Code Phases Using Piece-Wise Linear Regressions

Harald Servat†, Germán Llort*†, Juan González*†, Judit Giménez*†, and Jesús Labarta*†*

∗Computer Sciences Department, Barcelona Supercomputing Center
c/Jordi Girona, 31 - 08034 Barcelona, Catalunya, Spain
†Department of Computer Architecture, Universitat Polit`ecnica de Catalunya
c/Jordi Girona, 1-3 - 08034 Barcelona, Catalunya, Spain
name.surname@bsc.es

Node-level performance is one of the factors that may limit applications from reaching the supercomputers' peak performance. Studying node-level performance and attributing it to the source code results into valuable insight that can be used to improve the application efficiency, albeit performing such a study may be an intimidating task due to the complexity and size of the applications. We present in this paper a mechanism that takes advantage of combining piece-wise linear regressions, coarse-grain sampling, and minimal instrumentation to detect performance phases in the computation regions even if their granularity is very fine. This mechanism then maps the performance of each phase into the application syntactical structure displaying a correlation between performance and source code. We introduce a methodology on top of this mechanism to describe the node-level performance of parallel applications, even for first-time seen applications. Finally, we demonstrate the methodology describing optimized in-production applications and further improving their performance applying small transformations to the code based on the hints discovered.

Auto-Tuning Dedispersion for Many-Core Accelerators

Alessio Sclocco and Henri E. Bal
Faculty of Sciences
Vrije Universiteit Amsterdam
Amsterdam, the Netherlands
a.sclocco@vu.nl, h.e.bal@vu.nl

Jason Hessels and Joeri van Leeuwen
ASTRON
Netherlands Institute for Radio Astronomy
Dwingeloo, the Netherlands
hessels@astron.nl, leeuwen@astron.nl

Rob V. van Nieuwpoort
NLeSC
Netherlands eScience Center
Amsterdam, the Netherlands
r.vannieuwpoort@esciencecenter.nl

Dedispersion is a basic algorithm to reconstruct impulsive astrophysical signals. It is used in high sampling-rate radio astronomy to counteract temporal smearing by intervening interstellar medium. To counteract this smearing, the received signal train must be dedispersed for thousands of trial distances, after which the transformed signals are further analyzed. This process is expensive on both computing and data handling. This challenge is exacerbated in future, and even some current, radio telescopes which routinely produce hundreds of such data streams in parallel. There, the compute requirements for dedispersion are high (petascale), while the data intensity is extreme. Yet, the dedispersion algorithm remains a basic component of every radio telescope, and a fundamental step in searching the sky for radio pulsars and other transient astrophysical objects. In this paper, we study the parallelization of the dedispersion algorithm on many-core accelerators, including GPUs from AMD and NVIDIA, and the Intel Xeon Phi. An important contribution is the computational analysis of the algorithm, from which we conclude that dedispersion is inherently memory-bound in any realistic scenario, in contrast to earlier reports. We also provide empirical proof that, even in unrealistic scenarios, hardware limitations keep the arithmetic intensity low, thus limiting performance. We exploit auto-tuning to adapt the algorithm, not only to different accelerators, but also to different observations, and even telescopes. Our experiments show how the algorithm is tuned automatically for different scenarios and how it exploits and highlights the underlying specificities of the hardware: in some observations, the tuner automatically optimizes device occupancy, while in others it optimizes memory bandwidth. We quantitatively analyze the problem space, and by comparing the results of optimal auto-tuned versions against the best performing fixed codes, we show the impact that auto-tuning has on performance, and conclude that it is statistically relevant.

RCMP: Enabling Efficient Recomputation Based Failure Resilience for Big Data Analytics

Florin Dinu

EPFL, Lausanne, Switzerland

T. S. Eugene Ng

Rice University, Houston, TX, USA

Data replication, the main failure resilience strategy used for big data analytics jobs, can be unnecessarily inefficient. It can cause serious performance degradation when applied to intermediate job outputs in multi-job computations. For instance, for I/O-intensive big data jobs, data replication is especially expensive because very large datasets need to be replicated. Reducing the number of replicas is not a satisfactory solution as it only aggravates a fundamental limitation of data replication: its failure resilience guarantees are limited by the number of available replicas. When all replicas of some piece of intermediate job output are lost, cascading job recomputations may be required for recovery. In this paper we show how job recomputation can be made a first-order failure resilience strategy for big data analytics. The need for data replication can thus be significantly reduced. We present RCMP, a system that performs efficient job recomputation. RCMP can persist task outputs across jobs and leverage them to minimize the work performed during job recomputations. More importantly, RCMP addresses two important challenges that appear during job recomputations. The first is efficiently utilizing the available compute node parallelism. The second is dealing with hot-spots. RCMP handles both by switching to a finer-grained task scheduling granularity for recomputations. Our experiments show that RCMP's benefits hold across two different clusters, for job inputs as small as 40GB or as large as 1.2TB. Compared to RCMP, data replication is 30%-100% worse during failure-free periods. More importantly, by efficiently performing recomputations, RCMP is comparable or better even under single and double data loss events.

A Step towards Energy Efficient Computing: Redesigning a Hydrodynamic Application on CPU-GPU

Tingxing Dong, Veselin Dobrev†, Tzanio Kolev†, Robert Rieben†, Stanimire Tomov*, and Jack Dongarra**

∗Innovative Computing Laboratory, University of Tennessee, Knoxville

†Lawrence Livermore National Laboratory
∗tdong, tomov, dongarra@eecs.utk.edu
†dobrev1,kolev1,rieben1@llnl.gov

Power and energy consumption are becoming an increasing concern in high performance computing. Compared to multi-core CPUs, GPUs have a much better performance per watt. In this paper we discuss efforts to redesign the most computation intensive parts of BLAST, an application that solves the equations for compressible hydrodynamics with high order finite elements, using GPUs BLAST, Dobrev. In order to exploit the hardware parallelism of GPUs and achieve high performance, we implemented custom linear algebra kernels. We intensively optimized our CUDA kernels by exploiting the memory hierarchy, which exceed the vendor's library routines substantially in performance. We proposed an auto tuning technique to adapt our CUDA kernels to the orders of the finite element method. Compared to a previous base implementation, our redesign and optimization lowered the energy consumption of the GPU in two aspects: 60% less time to solution and 10% less power required. Compared to the CPU-only solution, our GPU accelerated BLAST obtained a 2.5× overall speedup and 1.42× energy efficiency (green up) using 4th order (Q_4) finite elements, and a 1.9× speedup and 1.27× green up using 2nd order (Q2) finite elements.

Session 23: Multithreading and Concurrency

IPDPS 2014

Using Multiple Threads to Accelerate Single Thread Performance

Zehra Sura, Kevin O'Brien, and Jose Brunheroto

IBM T.J. Watson Research Center
Yorktown Heights, New York, USA
{zsura,caomhin,brunhe}@us.ibm.com

Computing systems are being designed with an increasing number of hardware cores. To effectively use these cores, applications need to maximize the amount of parallel processing and minimize the time spent in sequential execution. In this work, we aim to exploit fine-grained parallelism beyond the parallelism already encoded in an application. We define an execution model using a primary core and some number of secondary cores that collaborate to speed up the execution of sequential code regions. This execution model relies on cores that are physically close to each other and have fast communication paths between them. For this purpose, we introduce dedicated hardware queues for low-latency transfer of values between cores, and define special "enque" and "deque" instructions to use the queues. Further, we develop compiler analyses and transformations to automatically derive fine-grained parallel code from sequential code regions. We implemented this model for exploiting fine-grained parallelization in the IBM XL compiler framework and in a simulator for the Blue Gene/Q system. We also studied the Sequoia benchmarks to determine code sections where our techniques are applicable. We evaluated our work using these code sections, and observed an average speedup of 1.32 on 2 cores, and an average speedup of 2.05 on 4 cores. Since these code sections are otherwise sequentially executed, we conclude that our approach is useful for accelerating single thread performance.

Active Measurement of Memory Resource Consumption

Marc Casas

Barcelona Supercomputing Center
Jordi Girona 29, Nexus II Building
08034 Barcelona

Greg Bronevetsky

Lawrence Livermore National Laboratory
7000 East Avenue
Livermore, CA, 94550

Hierarchical memory is a cornerstone of modern hardware design because it provides high memory performance and capacity at a low cost. However, the use of multiple levels of memory and complex cache management policies makes it very difficult to optimize the performance of applications running on hierarchical memories. As the number of compute cores per chip continues to rise faster than the total amount of available memory, applications will become increasingly starved for memory storage capacity and bandwidth, making the problem of performance optimization even more critical. We propose a new methodology for measuring and modeling the performance of hierarchical memories in terms of the application's utilization of the key memory resources: capacity of a given memory level and bandwidth between two levels. This is done by actively interfering with the application's use of these resources. The application's sensitivity to reduced resource availability is measured by observing the effect of interference on application performance. The resulting resource-oriented model of performance both greatly simplifies application performance analysis and makes it possible to predict an application's performance when running with various resource constraints. This is useful to predict performance for future memory-constrained architectures.

Locating Parallelization Potential in Object-Oriented Data Structures

Korbinian Molitorisz, Thomas Karcher, Alexander Bieleš, and Walter F. Tichy

Institute for Program Structures and Data Organization (IPD)
Karlsruhe Institute of Technology (KIT)
Am Fasanengarten 5, 767131 Karlsruhe, Germany
Email: molitorisz, thomas.karcher, walter.tichy@kit.edu, alexander.bieles@student.kit.edu

The free lunch of ever increasing single-processor performance is over. Software engineers have to parallelize software to gain performance improvements. But not every software engineer is a parallel expert and with millions of lines of code that have not been developed with multicore in mind, we have to find ways to assist in identifying parallelization potential. This paper makes three contributions: 1) An empirical study of more than 900,000 lines of code reveals five use cases in the runtime profile of object-oriented data structures that carry parallelization potential. 2) The study also points out frequently used data structures in realistic software in which these use cases can be found. 3) We developed DSspy, an automatic dynamic profiler that locates these use cases and makes recommendations on how to parallelize them. Our evaluation shows that DSspy reduces the search space for parallelization by up to 77% and engineers only need to consider 23% of all data structure instances for parallelization.

Session 24: Numerical Algorithms

IPDPS 2014

An Accelerated Recursive Doubling Algorithm for Block Tridiagonal Systems

Sudip K. Seal

Oak Ridge National Laboratory
Oak Ridge, TN 37831, USA
sealsk@ornl.gov

Block tridiagonal systems of linear equations arise in a wide variety of scientific and engineering applications. Recursive doubling algorithm is a well-known prefix computation-based numerical algorithm that requires O(M3(N/P + logP)) work to compute the solution of a block tridiagonal system with N block rows and block size M on P processors. In real-world applications, solutions of tridiagonal systems are most often sought with multiple, often hundreds and thousands, of different right hand sides but with the same tridiagonal matrix. Here, we show that a recursive doubling algorithm is sub-optimal when computing solutions of block tridiagonal systems with multiple right hand sides and present a novel algorithm, called the accelerated recursive doubling algorithm, that delivers O(R) improvement when solving block tridiagonal systems with R distinct right hand sides. Since R is typically ~ 102 – 104, this improvement translates to very significant speedups in practice. Detailed complexity analyses of the new algorithm with empirical confirmation of runtime improvements are presented. To the best of our knowledge, this algorithm has not been reported before in the literature.

Designing LU-QR Hybrid Solvers for Performance and Stability

Mathieu Faverge[1], Julien Herrmann[2], Julien Langou[3], Bradley R. Lowery[3], Yves Robert[2,4,] and Jack Dongarra[4]

[1] Laboratoire LaBRI, IPB ENSEIRB-MatMeca, Bordeaux, France

[2]Laboratoire LIP, École Normale Supérieure de Lyon, France

[3] University Colorado Denver, USA

[4] University of Tennessee Knoxville, USA

This paper introduces hybrid LU-QR algorithms for solving dense linear systems of the form Ax = b. Throughout a matrix factorization, these algorithms dynamically alternate LU with local pivoting and QR elimination steps, based upon some robustness criterion. LU elimination steps can be very efficiently parallelized, and are twice as cheap in terms of operations, as QR steps. However, LU steps are not necessarily stable, while QR steps are always stable. The hybrid algorithms execute a QR step when a robustness criterion detects some risk for instability, and they execute an LU step otherwise. Ideally, the choice between LU and QR steps must have a small computational overhead and must provide a satisfactory level of stability with as few QR steps as possible. In this paper, we introduce several robustness criteria and we establish upper bounds on the growth factor of the norm of the updated matrix incurred by each of these criteria. In addition, we describe the implementation of the hybrid algorithms through an extension of the Parsec software to allow for dynamic choices during execution. Finally, we analyze both stability and performance results compared to state-of-the-art linear solvers on parallel distributed multicore platforms.

Effectively Exploiting Parallel Scale for All Problem Sizes in LU Factorization

Md Rakib Hasan and R. Clint Whaley

Computer Science and Engineering Division/
Center for Computation & Technology
Louisiana State University
Baton Rouge, LA, USA
Email: rhasan@cct.lsu.edu
Email: rcwhaley@lsu.edu

LU factorization is one of the most widely-used methods for solving linear equations, and thus its performance underlies a broad range of scientific computing. As architectural trends have replaced clock rate improvements with increases in parallel scale, library writers have responded by using tiled algorithms, where operand size is constrained in order to maximize parallelism, as seen in the well-known PLASMA library. This approach has two main drawbacks: (1) asymptotic performance is reduced due to limited operand size, (2) performance of small to medium sized problems is reduced due to unnecessary data motion in the parallel caches. In this paper we introduce a new approach where asymptotic performance is maximized by using special low-overhead kernel primitives that are auto-generated by the ATLAS framework, while unnecessary cache motion is minimized by using explicit cache management. We show that this technique can outperform all known libraries at all problem sizes on commodity parallel Intel and AMD platforms, with asymptotic LU performance of roughly 91% of hardware theoretical peak for a 12-core Intel Xeon, and 87% for a 32-core AMD Opteron.

Anatomy of High-Performance Many-Threaded Matrix Multiplication

Tyler M. Smith, Robert van de Geijn*, Mikhail Smelyanskiy†, Jeff R. Hammond‡, and Field G. Van Zee**

∗Institute for Computational Engineering and Sciences and Department of Computer Science
The University of Texas at Austin, Austin TX, 78712
Email: tms,rvdg,field@cs.utexas.edu

†Parallel Computing Lab Intel Corporation
Santa Clara, CA 95054
Email: mikhail.smelyanskiy@intel.com

‡Leadership Computing Facility Argonne National Lab
Argonne, IL 60439
Email: jhammond@alcf.anl.gov

BLIS is a new framework for rapid instantiation of the BLAS. We describe how BLIS extends the "GotoBLAS approach" to implementing matrix multiplication (GEMM). While GEMM was previously implemented as three loops around an inner kernel, BLIS exposes two additional loops within that inner kernel, casting the computation in terms of the BLIS micro-kernel so that porting G E M M becomes a matter of customizing this micro-kernel for a given architecture. We discuss how this facilitates a finer level of parallelism that greatly simplifies the multithreading of GEMM as well as additional opportunities for parallelizing multiple loops. Specifically, we show that with the advent of many-core architectures such as the IBM PowerPC A2 processor (used by Blue Gene/Q) and the Intel Xeon Phi processor, parallelizing both within and around the inner kernel, as the BLIS approach supports, is not only convenient, but also necessary for scalability. The resulting implementations deliver what we believe to be the best open source performance for these architectures, achieving both impressive performance and excellent scalability.

Session 25: Performance Impacts of Hardware Acceleration

IPDPS 2014

Comparative Performance Analysis of Intel Xeon Phi, GPU, and CPU: A Case Study from Microscopy Image Analysis

George Teodoro[1], Tahsin Kurc[2,3], Jun Kong[4], Lee Cooper[4], and Joel Saltz[2]

[1]Department of Computer Science
University of Brasília, Brasília, DF, Brazil

[2]Department of Biomedical Informatics
Stony Brook University, Stony Brook, NY, USA

[3]Scientific Data Group
Oak Ridge National Laboratory, Oak Ridge, TN, USA

[4]Department of Biomedical Informatics
Emory University, Atlanta, GA, USA
teodoro@unb.br,{tkurc,lee.cooper,jun.kong,jhsaltz}@emory.edu

We study and characterize the performance of operations in an important class of applications on GPUs and Many Integrated Core (MIC) architectures. Our work is motivated by applications that analyze low-dimensional spatial datasets captured by high resolution sensors, such as image datasets obtained from whole slide tissue specimens using microscopy scanners. Common operations in these applications involve the detection and extraction of objects (object segmentation), the computation of features of each extracted object (feature computation), and characterization of objects based on these features (object classification). In this work, we have identify the data access and computation patterns of operations in the object segmentation and feature computation categories. We systematically implement and evaluate the performance of these operations on modern CPUs, GPUs, and MIC systems for a microscopy image analysis application. Our results show that the performance on a MIC of operations that perform regular data access is comparable or sometimes better than that on a GPU. On the other hand, GPUs are significantly more efficient than MICs for operations that access data irregularly. This is a result of the low performance of MICs when it comes to random data access. We also have examined the coordinated use of MICs and CPUs. Our experiments show that using a performance aware task strategy for scheduling application operations improves performance about 1.29× over a first-come-first-served strategy. This allows applications to obtain high performance efficiency on CPU-MIC systems - the example application attained an efficiency of 84% on 192 nodes (3072 CPU cores and 192 MICs).

A Framework for Lattice QCD Calculations on GPUs

F.T. Winter, M.A. Clark†, R.G. Edwards*, and B. Joó**

∗Thomas Jefferson National Accelerator Facility, Newport News, VA, USA
{fwinter,edwards,bjoo}@jlab.org

†NVIDIA Corporation, 2701 San Tomas Expressway, Santa Clara, CA 95050, USA
mclark@nvidia.com

Computing platforms equipped with accelerators like GPUs have proven to provide great computational power. However, exploiting such platforms for existing scientific applications is not a trivial task. Current GPU programming frameworks such as CUDA C/C++ require low-level programming from the developer in order to achieve high performance code. As a result porting of applications to GPUs is typically limited to time-dominant algorithms and routines, leaving the remainder not accelerated which can open a serious Amdahl's law issue. The Lattice QCD application Chroma allows us to explore a different porting strategy. The layered structure of the software architecture logically separates the data-parallel from the application layer. The QCD Data-Parallel software layer provides data types and expressions with stencil-like operations suitable for lattice field theory. Chroma implements algorithms in terms of this high-level interface. Thus by porting the low-level layer one effectively ports the whole application layer in one swing. The QDP-JIT/PTX library, our reimplementation of the low-level layer, provides a framework for Lattice QCD calculations for the CUDA architecture. The complete software interface is supported and thus applications can be run unaltered on GPU-based parallel computers. This reimplementation was possible due to the availability of a JIT compiler which translates an assembly language (PTX) to GPU code. The existing expression templates enabled us to employ compile-time computations in order to build code generators and to automate the memory management for CUDA. Our implementation has allowed us to deploy the full Chroma gauge-generation program on large scale GPU-based machines such as Titan and Blue Waters and accelerate the calculation by more than an order of magnitude.

Improving Communication Performance and Scalability of Native Applications on Intel® Xeon Phi™ Coprocessor Clusters

Karthikeyan Vaidyanathan, Kiran Pamnany*, Dhiraj D. Kalamkar*, Alexander Heinecke†, Mikhail Smelyanskiy‡, Jongsoo Park‡, Daehyun Kim‡, Aniruddha Shet G*, Bharat Kaul*, Balint Joo§, and Pradeep Dubey‡*

∗Parallel Computing Lab
Intel Corporation, Bangalore, India

†Department of Informatics
Technische Universität München, Munich, Germany

‡Parallel Computing Lab
Intel Corporation, Santa Clara, CA, U.S.A

§Thomas Jefferson National Accelerator Facility, Newport News, VA, U.S.A

Intel Xeon Phi coprocessor-based clusters offer high compute and memory performance for parallel workloads and also support direct network access. Many real world applications are significantly impacted by network characteristics and to maximize the performance of such applications on these clusters, it is particularly important to effectively saturate network bandwidth and/or hide communications latency. We demonstrate how to do so using techniques such as pipelined DMAs for data transfer, dynamic chunk sizing, and better asynchronous progress. We also show a method for, and the impact of avoiding serialization and maximizing parallelism during application communication phases. Additionally, we apply application optimizations focused on balancing computation and communication in order to hide communication latency and improve utilization of cores and of network bandwidth. We demonstrate the impact of our techniques on three well known and highly optimized HPC kernels running natively on the Intel Xeon Phi coprocessor. For the Wilson-Dslash operator from Lattice QCD, we characterize the improvements from each of our optimizations for communication performance, apply our method for maximizing concurrency during communication phases, and show an overall 48% improvement from our previously best published result. For HPL/LINPACK, we show 68.5% efficiency with 97 TFLOPs on 128 Intel Xeon Phi coprocessors, the first ever reported native HPL efficiency on a coprocessor-based supercomputer. For FFT, we show 10.8 TFLOPs using 1024 Intel Xeon Phi coprocessors on the TACC Stampede cluster, the highest reported performance on any Intel Architecture-based cluster and the first such result to be reported on a coprocessor-based supercomputer.

Computational Co-design of a Multiscale Plasma Application: A Process and Initial Results

Joshua Payne, Dana Knoll, Allen McPherson, William Taitano, Luis Chacon, Guangye Chen, and Scott Pakin

Los Alamos National Laboratory

As computer architectures become increasingly heterogeneous the need for algorithms and applications that can exploit these new architectures grows more pressing. This paper demonstrates that co-designing a multi-architecture, multi-scale, highly optimized framework with its associated plasma-physics application can provide both portability across CPUs and accelerators and high performance. Our framework utilizes multiple abstraction layers in order to maximize code reuse between architectures while providing low-level abstractions to incorporate architecture-specific optimizations such as vectorization or hardware fused multiply-add. We describe a co-design process used to enable a plasma physics application to scale well to large systems while also improving on both the accuracy and speed of the simulations. Optimized multi-core results will be presented to demonstrate ability to isolate large amounts of computational work with minimal communication.

Session 26: Programming Models and Tools

IPDPS 2014

UPC++: A PGAS Extension for C++

Yili Zheng, Amir Kamil*, Michael B. Driscoll*†, Hongzhang Shan*, and Katherine Yelick*†*

∗Lawrence Berkeley National Laboratory
†Department of EECS, University of California, Berkeley

Partitioned Global Address Space (PGAS) languages are convenient for expressing algorithms with large, random-access data, and they have proven to provide high performance and scalability through lightweight one-sided communication and locality control. While very convenient for moving data around the system, PGAS languages have taken different views on the model of computation, with the static Single Program Multiple Data (SPMD) model providing the best scalability. In this paper we present UPC++, a PGAS extension for C++ that has three main objectives: 1) to provide an object-oriented PGAS programming model in the context of the popular C++ language, 2) to add useful parallel programming idioms unavailable in UPC, such as asynchronous remote function invocation and multidimensional arrays, to support complex scientific applications, 3) to offer an easy on-ramp to PGAS programming through interoperability with other existing parallel programming systems (e.g., MPI, OpenMP, CUDA). We implement UPC++ with a "compiler-free" approach using C++ templates and runtime libraries. We borrow heavily from previous PGAS languages and describe the design decisions that led to this particular set of language features, providing significantly more expressiveness than UPC with very similar performance characteristics. We evaluate the programmability and performance of UPC++ using five benchmarks on two representative supercomputers, demonstrating that UPC++ can deliver excellent performance at large scale up to 32K cores while offering PGAS productivity features to C++ applications.

An Evaluation of One-Sided and Two-Sided Communication Paradigms on Relaxed-Ordering Interconnect

Khaled Z. Ibrahim, Paul H. Hargrove, Costin Iancu, and Katherine Yelick

Lawrence Berkeley National Laboratory, Berkeley, USA
Email: {kzibrahim, phhargrove, cciancu, kayelick}@lbl.gov

The Cray Gemini interconnect hardware provides multiple transfer mechanisms and out-of-order message delivery to improve communication throughput. In this paper we quantify the performance of one-sided and two-sided communication paradigms with respect to: 1) the optimal available hardware transfer mechanism, 2) message ordering constraints, 3) per node and per core message concurrency. In addition to using Cray native communication APIs, we use UPC and MPI micro-benchmarks to capture one- and two-sided semantics respectively. Our results indicate that relaxing the message delivery order can improve performance up to 4.6x when compared with strict ordering. When hardware allows it, high-level one-sided programming models can already take advantage of message reordering. Enforcing the ordering semantics of two-sided communication comes with a performance penalty. Furthermore, we argue that exposing out-of-order delivery at the application level is required for the next-generation programming models. Any ordering constraints in the language specifications reduce communication performance for small messages and increase the number of active cores required for peak throughput.

Scaling Irregular Applications through Data Aggregation and Software Multithreading

Alessandro Morari, Antonino Tumeo, and Daniel Chavarría-Miranda

Pacific Northwest National Laboratory
Richland, WA, USA
{alessandro.morari, antonino.tumeo,
daniel.chavarria}@pnnl.gov

Oreste Villa

NVIDIA
Santa Clara, CA, USA
ovilla@nvidia.com

Mateo Valero

Universitat Politecnica de Catalunya
Barcelona Supercomputing Center
Barcelona, Spain
mateo@bsc.es

Emerging applications in areas such as bioinformatics, data analytics, semantic databases and knowledge discovery employ datasets from tens to hundreds of terabytes. Currently, only distributed memory clusters have enough aggregate space to enable in-memory processing of datasets of this size. However, in addition to large sizes, the data structures used by these new application classes are usually characterized by unpredictable and fine-grained accesses: i.e., they present an irregular behavior. Traditional commodity clusters, instead, exploit cache-based processor and high-bandwidth networks optimized for locality, regular computation and bulk communication. For these reasons, irregular applications are inefficient on these systems, and require custom, hand-coded optimizations to provide scaling in both performance and size. Lightweight software multithreading, which enables tolerating data access latencies by overlapping network communication with computation, and aggregation, which allows reducing overheads and increasing bandwidth utilization by coalescing fine-grained network messages, are key techniques that can speed up the performance of large scale irregular applications on commodity clusters. In this paper we describe GMT (Global Memory and Threading), a runtime system library that couples software multithreading and message aggregation together with a Partitioned Global Address Space (PGAS) data model to enable higher performance and scaling of irregular applications on multi-node systems. We present the architecture of the runtime, explaining how it is designed around these two critical techniques. We show that irregular applications written using our runtime can outperform, even by orders of magnitude, the corresponding applications written using other programming models that do not exploit these techniques.

Generalizing Run-Time Tiling with the Loop Chain Abstraction

Michelle Mills Strout, Fabio Luporini†, Christopher D. Krieger*, Carlo Bertolli,‡*
Gheorghe-Teodor Bercea†, Catherine Olschanowsky, J. Ramanujam§, and Paul H.J. Kelly†*

∗Colorado State University
∗{mstrout|krieger|cathie}@cs.colostate.edu,

†Imperial College London
†{f.luporini12|gheorghe-teodor.bercea08|p.kelly}@imperial.ac.uk

‡IBM T.J. Watson Research
‡cbertol@us.ibm.com

§Louisiana State University
§jxr@ece.lsu.edu

Many scientific applications are organized in a data parallel way: as sequences of parallel and/or reduction loops. This exposes parallelism well, but does not convert data reuse between loops into data locality. This paper focuses on this issue in parallel loops whose loop-to-loop dependence structure is data-dependent due to indirect references such as A[B[i]]. Such references are a common occurrence in sparse matrix computations, molecular dynamics simulations, and unstructured-mesh computational fluid dynamics (CFD). Previously, sparse tiling approaches were developed for individual benchmarks to group iterations across such loops to improve data locality. These approaches were shown to benefit applications such as moldyn, Gauss-Seidel, and the sparse matrix powers kernel, however the run-time routines for performing sparse tiling were hand coded per application. In this paper, we present a generalized full sparse tiling algorithm that uses the newly developed loop chain abstraction as input, improves inter-loop data locality, and creates a task graph to expose shared-memory parallelism at runtime. We evaluate the overhead and performance impact of the generalized full sparse tiling algorithm on two codes: a sparse Jacobi iterative solver and the Airfoil CFD benchmark.

Session 27: Algorithms for High Performance Computing

IPDPS 2014

s-Step Krylov Subspace Methods as Bottom Solvers for Geometric Multigrid

Samuel Williams, Mike Lijewski, Ann Almgren, and Brian Van Straalen

Lawrence Berkeley National Lab
SWWilliams@lbl.gov

Erin Carson, Nicholas Knight, and James Demmel

University of California at Berkeley
ecc2z@eecs.berkeley.edu

Geometric multigrid solvers within adaptive mesh refinement (AMR) applications often reach a point where further coarsening of the grid becomes impractical as individual sub domain sizes approach unity. At this point the most common solution is to use a bottom solver, such as BiCGStab, to reduce the residual by a fixed factor at the coarsest level. Each iteration of BiCGStab requires multiple global reductions (MPI collectives). As the number of BiCGStab iterations required for convergence grows with problem size, and the time for each collective operation increases with machine scale, bottom solves in large-scale applications can constitute a significant fraction of the overall multigrid solve time. In this paper, we implement, evaluate, and optimize a communication-avoiding s-step formulation of BiCGStab (CABiCGStab for short) as a high-performance, distributed-memory bottom solver for geometric multigrid solvers. This is the first time s-step Krylov subspace methods have been leveraged to improve multigrid bottom solver performance. We use a synthetic benchmark for detailed analysis and integrate the best implementation into BoxLib in order to evaluate the benefit of a s-step Krylov subspace method on the multigrid solves found in the applications LMC and Nyx on up to 32,768 cores on the Cray XE6 at NERSC. Overall, we see bottom solver improvements of up to 4.2x on synthetic problems and up to 2.7x in real applications. This results in as much as a 1.5x improvement in solver performance in real applications.

Reconstructing Householder Vectors from Tall-Skinny QR

Grey Ballard, James Demmel†, Laura Grigori‡, Mathias Jacquelin§, Hong Diep Nguyen†, and Edgar Solomonik†*

∗Sandia National Laboratories, Livermore, USA

†University of California, Berkeley, Berkeley, USA

‡INRIA Paris - Rocquencourt, Paris, France

§Lawrence Berkeley National Laboratory, Berkeley, USA
gmballa@sandia.gov, demmel@cs.berkeley.edu, laura.grigori@inria.fr,
mjacquelin@lbl.gov, hdnguyen@cs.berkeley.edu, solomon@cs.berkeley.edu

The Tall-Skinny QR (TSQR) algorithm is more communication efficient than the standard Householder algorithm for QR decomposition of matrices with many more rows than columns. However, TSQR produces a different representation of the orthogonal factor and therefore requires more software development to support the new representation. Further, implicitly applying the orthogonal factor to the trailing matrix in the context of factoring a square matrix is more complicated and costly than with the Householder representation. We show how to perform TSQR and then reconstruct the Householder vector representation with the same asymptotic communication efficiency and little extra computational cost. We demonstrate the high performance and numerical stability of this algorithm both theoretically and empirically. The new Householder reconstruction algorithm allows us to design more efficient parallel QR algorithms, with significantly lower latency cost compared to Householder QR and lower bandwidth and latency costs compared with Communication-Avoiding QR (CAQR) algorithm. As a result, our final parallel QR algorithm outperforms ScaLAPACK and Elemental implementations of Householder QR and our implementation of CAQR on the Hopper Cray XE6 NERSC system. We also provide algorithmic improvements to the ScaLAPACK and CAQR algorithms.

Petascale General Solver for Semidefinite Programming Problems with Over Two Million Constraints

Katsuki Fujisawa[1], Toshio Endo[2], Yuichiro Yasui[1], Hitoshi Sato[2], Naoki Matsuzawa[3], Satoshi Matsuoka[4], and Hayato Waki[4]

[1]Chuo University, Tokyo, Japan
fujisawa@indsys.chuo-u.ac.jp
yasui@indsys.chuo-u.ac.jp

[2]Tokyo Institute of Technology, Tokyo, Japan
endo@is.titech.ac.jp
hitoshi.sato@gsic.titech.ac.jp
matsu@is.titech.ac.jp

[3]University of Tokyo
Tokyo, Japan
naoki matsuzawa@ipc.i.u-tokyo.ac.jp

[4]Kyushu University, Fukuoka, Japan
waki@imi.kyushu-u.ac.jp

The semi definite programming (SDP) problem is one of the central problems in mathematical optimization. The primal-dual interior-point method (PDIPM) is one of the most powerful algorithms for solving SDP problems, and many research groups have employed it for developing software packages. However, two well-known major bottlenecks, i.e., the generation of the Schur complement matrix (SCM) and its Cholesky factorization, exist in the algorithmic framework of the PDIPM. We have developed a new version of the semi definite programming algorithm parallel version (SDPARA), which is a parallel implementation on multiple CPUs and GPUs for solving extremely large-scale SDP problems with over a million constraints. SDPARA can automatically extract the unique characteristics from an SDP problem and identify the bottleneck. When the generation of the SCM becomes a bottleneck, SDPARA can attain high scalability using a large quantity of CPU cores and some processor affinity and memory interleaving techniques. SDPARA can also perform parallel Cholesky factorization using thousands of GPUs and techniques for overlapping computation and communication if an SDP problem has over two million constraints and Cholesky factorization constitutes a bottleneck. We demonstrate that SDPARA is a high-performance general solver for SDPs in various application fields through numerical experiments conducted on the TSUBAME 2.5 supercomputer, and we solved the largest SDP problem (which has over 2.33 million constraints), thereby creating a new world record. Our implementation also achieved 1.713 PFlops in double precision for large-scale Cholesky factorization using 2,720 CPUs and 4,080 GPUs.

Optimization of Multi-level Checkpoint Model for Large Scale HPC Applications

Sheng Di[1, 2], Mohamed Slim Bouguerra[1, 2], Leonardo Bautista-Gomez[2], and Franck Cappello[2]

[1]INRIA, France
sheng.di@inria.fr, slim.bouguerra@imag.fr

[2]Argonne National Laboratory, USA
{leobago, cappello}@mcs.anl.gov

HPC community projects that future extreme scale systems will be much less stable than current Petascale systems, thus requiring sophisticated fault tolerance to guarantee the completion of large scale numerical computations. Execution failures may occur due to multiple factors with different scales, from transient uncorrectable memory errors localized in processes to massive system outages. Multi-level checkpoint/restart is a promising model that provides an elastic response to tolerate different types of failures. It stores checkpoints at different levels: e.g., local memory, remote memory, using a software RAID, local SSD, remote file system. In this paper, we respond to two open questions: 1) how to optimize the selection of checkpoint levels based on failure distributions observed in a system, 2) how to compute the optimal checkpoint intervals for each of these levels. The contribution is three-fold. (1) We build a mathematical model to fit the multi-level checkpoint/restart mechanism with large scale applications regarding various types of failures. (2) We theoretically optimize the entire execution performance for each parallel application by selecting the best checkpoint level combination and corresponding checkpoint intervals. (3) We characterize checkpoint overheads on different checkpoint levels in a real cluster environment, and evaluate our optimal solutions using both simulation with millions of cores and real environment with real-world MPI programs running on hundreds of cores. Experiments show that optimized selections of levels associated with optimal checkpoint intervals at each level outperforms other state-of-the-art solutions by 5-50 percent.

Session 28: Scalable Algorithms

IPDPS 2014

Evaluating the Impact of SDC on the GMRES Iterative Solver

James Elliott†, Mark Hoemmen†, and Frank Mueller**

∗ Computer Science Department
North Carolina State University, Raleigh, NC

† Sandia National Laboratories
Albuquerque, NM

Increasing parallelism and transistor density, along with increasingly tighter energy and peak power constraints, may force exposure of occasionally incorrect computation or storage to application codes. Silent data corruption (SDC) will likely be infrequent, yet one SDC suffices to make numerical algorithms like iterative linear solvers cease progress towards the correct answer. Thus, we focus on resilience of the iterative linear solver GMRES to a single transient SDC. We derive inexpensive checks to detect the effects of an SDC in GMRES that work for a more general SDC model than presuming a bit flip. Our experiments show that when GMRES is used as the inner solver of an inner-outer iteration, it can "run through" SDC of almost any magnitude in the computationally intensive orthogonalization phase. That is, it gets the right answer using faulty data without any required roll back. Those SDCs which it cannot run through, get caught by our detection scheme.

A Multi-core Parallel Branch-and-Bound Algorithm Using Factorial Number System

Mohand Mezmaz, Rudi Leroy†, Nouredine Melab†, and Daniel Tuyttens**

∗ University of Mons
Mons, Belgium
Email: {Mohand.Mezmaz,Daniel.Tuyttens}@umons.ac.be

† Inria Lille - Nord Europe, LIFL-Université Lille1
Villeneuve d'Ascq, France
Email: {Rudi.Leroy,Nouredine.Melab}@lifl.fr

Many real-world problems in different industrial and economic fields are permutation combinatorial optimization problems. Solving to optimality large instances of these problems, such as flowshop problem, is a challenge for multi-core computing. This paper proposes a multi-threaded factoradic-based branch-and-bound algorithm to solve permutation combinatorial problems on multi-core processors. The factoradic, called also factorial number system, is a mixed radix numeral system adapted to numbering permutations. In this new parallel algorithm, the B&B is based on a matrix of integers instead of a pool of permutations, and work units exchanged between threads are intervals of factoradics instead of sets of nodes.

Compared to a conventional pool-based approach, the obtained results on flowshop instances demonstrate that our new factoradic-based approach, on average, uses about 60 times less memory to store the pool of subproblems, generates about 1.3 times less page faults, waits about 7 times less time to synchronize the access to the pool, requires about 9 times less CPU time to manage this pool, and performs about 30,000 times less context switches.

Optimizing Sparse Matrix-Multiple Vectors Multiplication for Nuclear Configuration Interaction Calculations

Hasan Metin Aktulga, Aydin Buluç, Samuel Williams, and Chao Yang

Computational Research Division
Lawrence Berkeley National Lab
{hmaktulga, abuluc, swwilliams, cyang}@lbl.gov

Obtaining highly accurate predictions on the properties of light atomic nuclei using the configuration interaction (CI) approach requires computing a few extremal Eigen pairs of the many-body nuclear Hamiltonian matrix. In the Many-body Fermion Dynamics for nuclei (MFDn) code, a block Eigen solver is used for this purpose. Due to the large size of the sparse matrices involved, a significant fraction of the time spent on the Eigen value computations is associated with the multiplication of a sparse matrix (and the transpose of that matrix) with multiple vectors (SpMM and SpMM_T). Existing implementations of SpMM and SpMM_T significantly underperform expectations. Thus, in this paper, we present and analyze optimized implementations of SpMM and SpMM_T. We base our implementation on the compressed sparse blocks (CSB) matrix format and target systems with multi-core architectures. We develop a performance model that allows us to understand and estimate the performance characteristics of our SpMM kernel implementations, and demonstrate the efficiency of our implementation on a series of real-world matrices extracted from MFDn. In particular, we obtain 3-4 speedup on the requisite operations over good implementations based on the commonly used compressed sparse row (CSR) matrix format. The improvements in the SpMM kernel suggest we may attain roughly a 40% speed up in the overall execution time of the block Eigen solver used in MFDn.

Session 29: Resilience and Reliability

IPDPS 2014

FMI: Fault Tolerant Messaging Interface for Fast and Transparent Recovery

Kento Sato
Dep. of Mathematical and Computing Science
Tokyo Institute of Technology
2-12-1-W8-33, Ohokayama,
Meguro-ku, Tokyo 152-8552 Japan
Email: kent@matsulab.is.titech.ac.jp

Adam Moody, Kathryn Mohror, Todd Gamblin and Bronis R. de Supinski
Center for Applied Scientific Computing
Lawrence Livermore National Laboratory
Livermore, CA 94551 USA
Email: fmoody20, kathryn, tgamblin, bronisg@llnl.gov

Naoya Maruyama
Advanced Institute for Computational Science
RIKEN
7-1-26, Minatojima-minami-machi,
Chuo-ku, Kobe, Hyogo, 650-0047 Japan
Email: nmaruyama@riken.jp

Satoshi Matsuoka
Global Scientific Information and Computing Center
Tokyo Institute of Technology
2-12-1-W8-33, Ohokayama,
Meguro-ku, Tokyo 152-8552 Japan
Email: matsu@is.titech.ac.jp

Future supercomputers built with more components will enable larger, higher-fidelity simulations, but at the cost of higher failure rates. Traditional approaches to mitigating failures, such as checkpoint/restart (C/R) to a parallel file system incur large overheads. On future, extreme-scale systems, it is unlikely that traditional C/R will recover a failed application before the next failure occurs. To address this problem, we present the Fault Tolerant Messaging Interface (FMI), which enables extremely low-latency recovery. FMI accomplishes this using a survivable communication runtime coupled with fast, in-memory C/R, and dynamic node allocation. FMI provides message-passing semantics similar to MPI, but applications written using FMI can run through failures. The FMI runtime software handles fault tolerance, including check pointing application state, restarting failed processes, and allocating additional nodes when needed. Our tests show that FMI runs with similar failure-free performance as MPI, but FMI incurs only a 28% overhead with a very high mean time between failures of 1 minute.

Designing Bit-Reproducible Portable High-Performance Applications

Andrea Arteaga[1], Oliver Fuhrer[2], and Torsten Hoefler[1]

[1]ETH Zurich, Switzerland
andrea.arteaga@env.ethz.ch
htor@ethz.ch

[2]Federal Office for Meteorology and Climatology
MeteoSwiss, Zurich, Switzerland
oliver.fuhrer@meteoswiss.ch

Bit-reproducibility has many advantages in the context of high-performance computing. Besides simplifying and making more accurate the process of debugging and testing the code, it can allow the deployment of applications on heterogeneous systems, maintaining the consistency of the computations. In this work we analyze the basic operations performed by scientific applications and identify the possible sources of non-reproducibility. In particular, we consider the tasks of evaluating transcendental functions and performing reductions using non-associative operators. We present a set of techniques to achieve reproducibility and we propose improvements over existing algorithms to perform reproducible computations in a portable way, at the same time obtaining good performance and accuracy. By applying these techniques to more complex tasks we show that bit-reproducibility can be achieved on a broad range of scientific applications.

F-SEFI: A Fine-Grained Soft Error Fault Injection Tool for Profiling Application Vulnerability

Qiang Guan, Nathan DeBardeleben, and Sean Blanchard

Ultrascale Systems Research Center
Los Alamos National Laboratory1
{qguan,ndebard,seanb}@lanl.gov

Song Fu

Department of Computer Science and Engineering
University of North Texas
Song.Fu@unt.edu

As the high performance computing (HPC) community continues to push towards exascale computing, resilience remains a serious challenge. With the expected decrease of both feature size and operating voltage, we expect a significant increase in hardware soft errors. HPC applications of today are only affected by soft errors to a small degree but we expect that this will become a more serious issue as HPC systems grow. We propose F-SEFI, a Fine-grained Soft Error Fault Injector, as a tool for profiling software robustness against soft errors. In this paper we utilize soft error injection to mimic the impact of errors on logic circuit behavior. Leveraging the open source virtual machine hypervisor QEMU, F-SEFI enables users to modify emulated machine instructions to introduce soft errors. F-SEFI can control what application, which sub-function, when and how to inject soft errors with different granularities, without interference to other applications that share the same environment. F-SEFI does this without requiring revisions to the application source code, compilers or operating systems. We discuss the design constraints for F-SEFI and the specifics of our implementation. We demonstrate use cases of F-SEFI on several benchmark applications to show how data corruption can propagate to incorrect results.

Author Index

Author Index

Author Index

Author Index

Author Index