

Sudheer Kumar

RESEARCH INTERESTS	High Performance Computing, Performance Optimization on Multi-cores and Accelerators
EDUCATION	<p>Ph.D., Computer Science, Sri Sathya Sai Institute of Higher Learning, April 2013</p> <ul style="list-style-type: none">• Advisor: <i>Prof. Ashok Srinivasan, Florida State University</i>• Thesis: <i>Topology and Routing Aware Mapping on Parallel Processors</i> <p>M.Tech., Computer Science, Sri Sathya Sai Institute of Higher Learning, March 2006</p> <ul style="list-style-type: none">• Advisor: <i>Mr. Shakti Kapoor, STSM, IBM Austin</i> <p>B.Tech., Information Technology, RVR&JC College of Engineering, April 2004</p>
WORK EXPERIENCE	Assistant Professor , Department of Mathematics & Computer Science, Sri Sathya Sai Institute of Higher Learning, July 2011 - Present
JOURNAL PUBLICATIONS	<i>Dynamic Load Balancing for Petascale Quantum Monte Carlo Applications: The Alias Method</i> , C.D. Sudheer , S. Krishnan, A. Srinivasan, and P. R. C. Kent. Computer Physics Communications, Feb 2013, Impact Factor : 3.268, (5-year Impact Factor: 2.812).
CONFERENCE PUBLICATIONS	<p><i>Optimization of the Hop-Byte Metric for Effective Topology Aware Mapping</i>, C.D. Sudheer, Ashok Srinivasan, Proceedings of the 19th IEEE International Conference on High Performance Computing (HiPC), 2012, (Acceptance rate: 25%).</p> <p><i>Optimizing Assignment of Threads to SPEs of the Cell BE Processor</i>, C.D. Sudheer, T. Nagaraju, P.K. Baruah, Ashok Srinivasan, 10th IEEE International Workshop on Parallel and Distributed Scientific and Engineering Computing (PDSEC), Proceedings of the 23rd International Parallel and Distributed Processing Symposium, IEEE, 2009, (Citations: SG - 5).</p> <p><i>High Throughput Compression of Floating point numbers in GPUs</i>, Ajith Padyana, C.D. Sudheer, P.K. Baruah, Ashok Srinivasan, 2nd IEEE International Conference on Parallel, Distributed and Grid Computing - 2012 Himachal Pradesh, December 2012.</p> <p><i>Investigating Algorithmic Techniques for Enhancing Application Performance on Multicore Processors</i>, C.D. Sudheer (Advisor: Ashok Srinivasan), PhD Forum at IEEE International Parallel and Distributed Processing Symposium, 2009.</p> <p><i>A Communication Model for Determining Optimal Affinity on the Cell BE processor</i>, C.D. Sudheer, Sriram, S.: In: Proc. 16th IEEE International Conference on High Performance Computing (HiPC), Student Research Symposium, Dec 2009.</p>
INVITED PRESENTATIONS	<p><i>An Overview of the Global Arrays Toolkit</i>, Five-days Technology Workshop on Heterogeneous Computing - Many Core/ Multi GPU - Performance of Algorithms, Application Kernels (HeMPa), 2011, at CMSD, UoHYD by C-DAC Pune & CMSD.</p> <p><i>Programming for Performance on Cell BE processor</i>, Performance Enhancement on Emerging Parallel Processing Platforms Workshop (PEEP), 2008, jointly organized by C-DAC and IUCAA.</p>
PROFESSIONAL SERVICE	Technical Program Committee member, International Conference on High Performance Computing and Communications (HPCC), 2011, 2012.

TPC member, The 11th IEEE International Symposium on Parallel and Distributed Processing with Applications (ISPA), 2013.

TPC member, The 12th International Conference on Algorithms and Architectures for Parallel Processing (ICA3PP), 2012.

TPC member, Student Research Symposium, IEEE International Conference on High Performance Computing (HiPC), 2012.

Reviewer, Computing, Springer Journal.

COMPUTER ACCESS TIME GRANTS *XSEDE Research Allocation: Scaling Communication Performance for Massively Parallel Applications*, 800,000 SUs, PI: Prof. Ashok Srinivasan, Florida State University.

Teragrid Startup Allocation, PI: Prof. P. Sadayappan, Ohio State University.

XSEDE Education Allocation: Programming for Performance on multicore and many-core processor, PI: Prof. Ravi Mukkamala, Old Dominion University. Have used this allocation effectively for teaching graduate level courses. Had access to the following supercomputers: TACC systems Ranger, Lonestar and Longhorn, SDSC systems Gordon and Trestles, PSC Blacklight.

TECHNICAL EXPERTISE

Familiar with the usage of massively parallel machines such as NICS Kraken and TACC Stampedede.

Experience in programming Cell BE, CUDA, OpenCL, Cilk, MPI, OpenMP and Global Arrays.

Programming Languages: C, PowerPC and x86 assembly and C++.

Conversant with architectures Cell BE, PowerPC 405 Embedded, ARM, MIPS and x86.

KEY RESEARCH PROJECTS

Topology and routing aware mapping tool for massively parallel processors

- Developed general mapping techniques by posing the hop-byte metric as a quadratic assignment problem (QAP). Rather than using the metric just for evaluation of the mapping quality, the idea is intuitive in optimizing the metric itself.
- A metric based on the idea of minimizing the number of bottleneck links, called the maximum contention metric, requires the routing information along with the topology details. We showed that our heuristics for optimizing this metric are more effective in reduction communication costs.
- Tools/programming methods used: C, C++, Cray MPI, Cray Scheduler, Python, OSU MPI benchmarks and large scale code profiler - HPC toolkit.

Topology aware implementation of Global Arrays data management for QWalk

- An automated data management approach is generally used to enable existing QWalk applications using the GA library to significantly enhance the range of problem sizes that can be handled.
- In this work, the topology and routing aware grouping of GA groups was implemented such that the contention created by the communication in the GA calls is minimized.
- Tools/programming methods used: C, C++, Global Arrays and MPI.

Optimal dynamic load balancing algorithm for large scale codes involving near identical computational tasks

- A new load balancing algorithm for Quantum Monte Carlo and showed that it scales well, especially when compared with existing implementations.
- We also theoretically analyze its performance characteristics under a variety of metrics, in order to provide more insight into its strengths and limitations.
- An important feature of the new algorithm is that the load can be perfectly balanced with each process receiving at most one message. It is also optimal in the maximum size of messages received by any process.

- Tools/programming methods used: C, MPI, OSU MPI benchmarks and profilers such TAU, IPM.

Optimizing assignment of threads to SPEs on the Cell BE Processor

- The actual bandwidth obtained for inter-SPE communication is strongly influenced by the assignment of threads to SPEs (thread-SPE affinity) in many realistic communication patterns. We identify the bottlenecks to optimal performance and use this information to determine good affinities for common communication patterns.
- Our solutions improve performance by up to a factor of two over the default assignment. We also discussed the optimization of affinity on a Cell blade consisting of two Cell processors, and provided a software tool to help with this.
- Tools/programming methods used: Cell BE programming involving signals, DMAs and mailboxes, C and MPI.

Reducing the disk IO bandwidth bottleneck through fast floating point compression using accelerators

- We proposed a compression technique based on time-series analysis, and investigate its effectiveness on floating point data from a variety of applications.
- We show that significant reduction in IO time can be achieved, even accounting for the compression overhead, on a Cell BE processor. In our experiments, the typical improvement was around 30%, varying from a slight loss in performance to a factor of seven improvement, depending on the type of application.
- The contribution of this work lies in demonstrating the potential of floating point compression in reducing the IO bandwidth bottleneck for important classes of scientific applications.
- Tools/programming methods used: Cell BE programming, GPU CUDA programming and C.

Design and implementation of an optimized Proc file system for a message passing operating system

- The aim is to design a process(proc) file system in the Minix OS by first understanding the implementation of it in Linux and then implement it for Minix OS running on PowerPC 405GP board.
- Implementation involved the debugging of Minix on PowerPC using IBM RISCWatch debugger.
- Tools/programming methods used: MINIX OS, PowerPC ISA and C.

Mini software projects

- Implementation of TFTP (Trivial File Transfer Protocol) in Java. It was also provided with a GUI.
- Implementation of a primitive web server / proxy / browser in UNIX.
- Tools/programming methods used: Java, C and UNIX programming internals.

TEACHING

High Performance Computing with Accelerators, 2012. <http://dmacssite.github.com>

- Students of this course learn skills to produce scalable parallel programs targeting the unique requirements for obtaining high performance on GPUs and multi-cores. We will compare and contrast parallel programming style for GPUs and conventional multi-cores.
- *This course was highly successful resulting in 9 student papers, more than one third of the total papers, accepted for Student Research Symposium, at HiPC 2012. And also, 3 out of the 4 awards constituted for Best Presentation and Best Poster were secured by the students of this course.*

Programming for Performance, 2011, 2012 and 2013. <http://progforperf.github.com>

- To obtain the high level of performance needed in scientific computing, it is necessary for programs to exploit many of the features of modern computer architectures. In this course, we study the performance-critical features of state-of-the-art architectures such as Intel MIC, Intel SandyBridge and AMD Opetron, and discuss how applications can take advantage of them to obtain high performance.
- This is not a course on software tricks; rather, the emphasis is on abstractions of computer architecture, understanding performance, and obtaining performance when you need it.

Parallel Computing, 2013. <http://parallelcomp.github.com>

- This course is a comprehensive exploration of parallel programming paradigms, examining core concepts, focusing on a subset of widely used contemporary parallel programming models, and providing context with a small set of parallel algorithms.

Computer Organization and Design, 2010, 2011 and 2012.

- This course provides a programmer's view of how computer systems execute programs, store information, and communicate. It enables students to become more effective programmers, especially in dealing with issues of performance, portability and robustness.
- It also serves as a foundation for courses on compilers, networks, operating systems, and computer architecture, where a deeper understanding of systems-level issues is required.
- Topics covered include: machine-level code and its generation by optimizing compilers, performance evaluation and optimization, computer arithmetic, memory organization and management and IO.

Processor Architecture and its Applications, 2008.

- The objective of this course is to teach intricate details of processor system Architecture. The idea is to design and develop an instruction set simulator for a simple in-order single issue RISC processor.
- It involves designing an orthogonal ISA for a hypothetical architecture, simulation routines for pipeline functional units, assembler, physically indexed cache, and then translation.

Operating Systems Design and Implementation, 2007.

Systems Programming using MINIX Operating System, 2006.

- The aim is to explain the interaction between the Operating System and the Architecture. MINIX OS code for the PowerPC architecture is used in the course.
- Various topics covered in the course: context switching (system call interrupt), OS message passing mechanism, keyboard interrupt, Programmable Interrupt Timer interrupt, reset interrupt, Address Translation (TLB interrupt) etc.

Course management tools

- Maintained course websites locally over e-learning portal and also in public domain over Git.
- Incorporated the use of Piazza, a web based tool for an efficient way to manage class Q&A.

ACADEMIC ACHIEVEMENTS

TCPD PhD Forum Travel Grant for attending the IPDPS 2009 conference in Italy.

Secured 96.84 percentile in Graduate Aptitude Test in Engineering (GATE-04) in Computer Science stream.

Achieved University first rank in C programming theory and laboratory exam in the 1st year of B.Tech.

REFERENCES

Prof. Ashok Srinivasan, *Florida State University* (asriniva@cs.fsu.edu)

Prof. P. Sadayappan, *Ohio State University* (saday@cse.ohio-state.edu)

Mr. Shakti Kapoor, *STSM, IBM Austin* (skapoor@us.ibm.com)

Prof. Pallav Kumar Baruah, *Sri Sathya Sai Institute of Higher Learning* (pkbaruah@sssihl.edu.in)