**Sandeep K. Gupta**

San Diego Supercomputer Center
9500 Gilman Drive, MC0505
La Jolla, CA 92093.

e-mail: sandeep@sdsc.edu
http://users.sdsc.edu/∼sandeep
phone: (714) 720-9458

## INTERESTS

- Data intensive high performance computing, massive scale graph processing and its applications.
- Parallel data engines for ontology, informatics, and semantic-web.
- Hardware based acceleration of data intensive applications for large scale SMPs, x86 clusters, and scratchpad (embedded) systems.
- Energy efficient architectures, characterization and profiling of applications for novel architectures
- Programming paradigm, domain specific languages, declarative query language.
- Message passing interface (MPI), OpenMP, explicit multi-threading (XMT), map-reduce, concurrent collections (CnC).
- Bioinformatics, machine learning, data mining, indexing, computational geometry, spatial and temporal databases, computational complexity.

## EMPLOYMENT EXPERIENCE

**Research Scientist** May 2009 - present
San Diego Super Computing Center,
University of California San Diego, CA.

– Principle Investigator, 'Processing Massive Graphs on Triton: Performance, Optimizations, and Applications':

○ *Developed a massively parallel engine for processing large scale informatics data and scaling the entire informatics processing workflow from ingesting flat data, representing relations as graphs, and performing operation over billions of records. The engine exploits the unique capabilities of modern clusters and supercomputers: fast interconnect, SSDs, multi-cores, atomics, and deep pipeline to process terabyte scale datasets. It can ingest RDF triples in parallel and construct very large graphs in near real time. It supports joins and other complex graph operations such as bfs, connected component, reachability, and pattern query.*

○ *Developed a hybrid distributed- and shared- memory algorithm for breadth-first-search. The algorithm shows scalability uptil* 2046 *cores distributed over* 128 *compute nodes and has performance rate of 2 billion edges per second (TEPS).*

○ *Developed parallel and distributed graph generators that, using 32–8192 cores, perform generation, permutation, join, sort, and storage of billions of integer pairs. Total storage space ranges up to several terabytes.*

○ *Developed an algorithm to process reachability queries over ontological (acyclic graphs) using massive multi-cores. The algorithm utilizes atomics and various contention avoidance strategies to achieve scalability upto* 256 *cores. The parallel version is more than two orders of magnitude faster than sequential on Altrix UV shared memory supercomputer on a graph with 64M nodes and 320M edges.*

○ *Developed concurrent* $B_{dh}$*-trees. Using double word compare-and-swap atomic instruction (cmpchx16b) available on modern x86 architectures it allows multiple threads to insert, delete, and search in the same tree structure simultaneously without introducing any race conditions. It also avoids the need for locking and mutual exclusion is scalable with respect to number of threads.*

○ *Developed concurrent graph data-structure using* $B_{dh}$*-trees and parallel algorithms to track popular/highly ranked nodes in the Twitter as the network changes over time.*

– Application Centric Hardware Design:

○ *Developed algorithms for large scale graph exploration on scratchpad memory (embedded) systems. The algorithm has been demonstrated to scale upto* 64 *cores.*

○ *Profiled performance of various numerical (shock hydrodynamics, molecular dynamics, sensors) and graph applications using cycle-accurate simulator (SESC) by varying hardware parameters such as L1/L2 cache sizes, cache-line size, pipeline depth, CPU-frequency and voltage.*

○ *Proposed architectural and computational innovations to accelerate connected component and modularity finding on dynamic graphs.*

○ *Developed a dynamic graph kernel and characterized its performance for modularity finding over social networks using SESC simulator.*

○ *Compared and contrasted synchronization costs for atomics on x86 architecture vs. full/empty bits on XMT and*

*its impact on dynamic graph problems.*
◦ *Wrote the social network modularity finding algorithm using the concurrent collections language(CnC). Proposed constructs to the CnC language for an efficient implementation of the same.*
– Participated in Graph500 Benchmark:
◦ *The graph500 competition ranks machine by their performance on graph application. Our entry using 128 nodes on graph of 8 billion nodes and 64 billion edges (scale 33) was ties at rank 20 out of 50 entries. Our method was able to perform breadth-first-search in less than 190 secs.*

**Post Doctoral Fellow** September 2006 - April 2009
San Diego Super Computing Center,
University of California San Diego, CA.
– Neuroscience Information Framework:
*The NIF project is an NIH initiative to integrate and query all of neuroscience relevant data. My responsibilities for this project included designing a unified data model for experimental and literature curated data. Developed benchmarks for informatics query processing. The benchmark comprises an acyclic graph generator, reachability queries, and path pattern queries. Developed a data model and a query language based upon acyclic structures so as to succinctly express life-sciences queries.*

**Graduate Student Research Assistant** August 2000 - August 2006
University of California, Riverside.
Worked on my thesis entitled 'External Memory Algorithms for Shortest Distance and Spatio-Temporal Queries on Road Networks'.
– Disk Based Shortest Distances on Large Graphs:
*Developed two data structures and associated external memory algorithms to answer shortest distances over large graphs efficiently. Implementation using LEDA and Chaco graph libraries demonstrate excellent performance on real data sets (Los Angeles and Chicago road networks). For a graph admitting $O(\sqrt{n})$ recursive vertex separators, the first data structure requires $O(n\sqrt{n}\log n)$ space, $O(\log n)$ time and $O(1)$ disk I/O, while the second data structure requires $O(n\sqrt[4]{n}\log n)$ space, $O(\sqrt[4]{n}\log^2 n)$ time, and $O(\sqrt[4]{n})$ disk I/O.*
– Framework for Moving Object Databases:
*Developed a model for representing trajectories and queries (e.g. range, join, pursuit) over objects moving along road network and distances measured along the road. Devised a route hashing technique based on hypercube embeddings of graph to expedite query processing.*
– Advanced Query Processing in Location Based Services:
*Developed a prototype engine for efficient indexing of spatial and moving objects on roads where distance is measure along the road. Features efficient techniques for clustering, intercept, range, and join query processing and incorporates both of the aforementioned work.*
– Almost Optimal Parallelization of 2D k-Nearest Neighbor Query:
*The $k$-nearest neighbor parallelization (or discrepancy) problem is to color a given set of 2-D data points with a given set of $c$ colors so that $k$-nearest neighbor of any point on the plane is not dominated by a single color. Developed a polynomial-time algorithm that colors with almost optimal discrepancy. In addition, developed a simple linear-time randomized algorithm that colors almost all points yet achieves near optimal discrepancy with high probability.*
– Dissemination of Location-tagged Data:
*Developed theory and techniques for reducing datasets that exhibit spatial correlations (e.g. traffic) such that if the reduced dataset is disseminated it reduces the bandwidth consumption yet has guaranteed error tolerance and low latency for any mobile client.*
– Online Detection and Control of SMS spams:
*Built a framework to accurately identify SMS spams. This work models a SMS message as a point in high dimensional space and then applies powerful yet simple techniques of random projections to perform a fast, scalable clustering and identification of SMS spams.*
– Joins on Historical Trajectory Data:
*Developed an index structure and associate cost models over historical trajectories that is optimized for join queries. The index structure is simple, fast to construct and update. It demonstrates excellent performance on wide variety of datasets for range and join queries.*
– Finding Similar Web Pages:
*Designed and implemented a system for finding pages similar to a given web page using Bayesian based classifier*

*over the DMOZ web hierarchy. Implemented a focused distributed and multi-threaded crawler, a classification scheme and a relational storage mechanism in MySQL.*

**Software Engineer** May 2000 - August 2000
Hughes Software Systems, Gurgaon (Delhi), India.
– Developed a multi-threaded testing suite for real-time protocols.

## EDUCATION

| | |
|---|---|
| 2006 | **Ph. D.** Computer Science & Engineering,<br>Department of Computer Science & Engineering,<br>University of California, Riverside, CA.<br>Thesis Advisor: Dr. Chinya V. Ravishanakar.<br>Thesis title: External Memory Algorithms for Shortest-Distance and Spatio-Temporal Queries on Road Network. |
| 2002 | **M.S.** Computer Science & Engineering,<br>University of California, Riverside, CA. |
| 2000 | **B. Tech.** Computer Science & Engineering<br>Indian Institute of Technology, Guwahati, India. |

### Working Papers

1. S. Gupta, *Parallel Operators for XML and acyclic graph structured databases*, in preparation.
2. S. Gupta, A. Agarwal, A. Snavely, and J. Torrellas *Energy efficient architectures for exploration over static and dynamic social networks*, in preparation.
3. S. Gupta, A. Agarwal, K. Seager, A. Snavely, and J. Torrellas *Power Efficient Breadth First Search Using Scratchpad Memory*, in preparation.
4. S. Gupta, *Parallel Primitives for Graph Processing in Hybrid Distributed+Shared memory environment*, in preparation.
5. S. Gupta, A. Agarwal, C. Liao, J. Torrellas, A. Snavely, and J. Torrellas *Memory and Energy Behavior of the DARPA Ubiquitous High Performance Computing (UHPC) Challenge Applications* , in preparation

## PUBLICATIONS

1. S. Gupta, *Answering Reachability Queries over Massive Acyclic Graphs Using Multi Cores and Flash*, Techincal Report, San Diego Supercomputer Center, TR-2012-1, 2012
2. S. Gupta, *A Unified Data Model and Declarative Query Language for Heterogenous Life Sciences Data*, Techincal Report, San Diego Supercomputer Center, TR-2011-3, 2011
3. S. Gupta, D. Gunopulos, C. V. Ravishankar *On the Complexity of 2-Dimensional K-Nearest Neighbors Query in Parallel* submitted to Computational Geometry- Theory and Applications, 2011
4. S. Gupta and C. V. Ravishankar, *PaL: Partial Labelings For Efficient Spatial and Spatiotemporal Queries on Road Networks,* Transactions of Knowledge Engineering and Databases, IEEE 2009, under major revisions.
5. J. He, A. Jagatheesan, S. Gupta, J. Bennett, A. Snavely, *DASH: A Recipe for a Flash-based Data Intensive Supercomputer,* Proceedings of the ACM/IEEE Conference on High Performance Computing, SC, 2009.
6. S. Gupta, C. Condit, and A. Gupta, *Graphitti: An Annotation Management System for Heterogeneous Objects*, Proc. of the 2008 IEEE 24th International Conference on Data Engineering, Vol. 5, No. 2-4, pp. 694-711, 2008.
7. S. Gupta, and C. V. Ravishankar, *Spatio-temporal Queries on Road Networks, Coding Based Methods,* Encyclopedia of GIS, pp. 1122-1125, Springer, 2008.
8. S. Gupta, J. Ni, and C. V. Ravishankar, *Efficient Data Dissemination Using Locale Covers,* Pervasive and Mobile Computing, Vol 14, No.2, pp. 254-275, 2008.
9. A. Gupta, S. D. Larson, C. Condit, S. Gupta, L. Fong, L. Chen, and M. E. Martone, *Toward an Ontological Database for Subcellular Neuroanatomy,* Int'l Conf. on Conceptual Modeling, Vol. LNCS, 4802, pp. 64-73, Springer, 2007.
10. S. Gupta, X. Qian, and A. Gupta, *OntoQueL: A Query Language for Ontological Database,* SWDB-ODBIS07: Joint ODBIS & SWDB Workshop on Semantic Web, Ontologies, Databses, collocated with VLDB, Vienna Austria, 2007.

11. S. Gupta, D. Gunopulos and C. V. Ravishankar, *Randomize, Refine, and Ignore: Efficient Near-Optimal Parallelization of 2-Dimensional Nearest Neighbor Queries*, Techinal Report, Department of Computer Science & Engineering, University of California, Riverside, 2006.
12. S. Dixit, S. Gupta, and C. V. Ravishankar, *Lohit: An Online Detection & Control System For Cellular SMS Spam,* Proc. IASTED International Conference on Network Security Phoenix, AZ, 2005.
13. Sandeep Gupta, Jinfeng Ni and Chinya V. Ravishankar, *Efficient Data Dissemination Using Locale Covers,* Proceedings of ACM Fourteenth Conference on Information and Knowledge Management (CIKM), Bremen, Germany, October 2005.
14. S. Gupta, S. Kopparty, and C. V. Ravishankar, *Roads, Codes and Spatiotemporal Queries,* Proc. 23rd ACM SIGMOD-SIGACT Symposium on Principles of Database Systems (PODS), Paris, France, June 2004. pp. 115-124, ACM 2004.
15. S. Gupta and C. V. Ravishankar, *Using vTree Indices for Queries over Objects with Complex Motions,* IEEE Int'l Conf. on Data Engineering (ICDE), Vol., pp. 831, 2004.

**CONFERENCES/ PRESENTATIONS/ POSTERS**

1. "Roads, Codes and Spatiotemporal Queries," PODS, Paris, France, June 2004.
2. "Using vTrees Indices for Queries over Objects with Complex Motions," ICDE, Boston, MA, July 2004.
3. "Scaling Reachability Queries on Directed Acyclic Graphs," Database Talks, CSE Department, UCSD, San Diego, October 2010.

**PROFESSIONAL ACTIVITIES**

- Fall 2000- Spring 2002 Teaching Assistant, Department of Computer Science & Engineering, UCR. Duties included grading & tutoring students in the laboratory.
- Professional Affiliations:
  - **Association for Computing Machinery,** Member, 2004 - present.
- External Reviewer for TKDE, ICDE, VLDB, INFOCOM, dates.

**AWARDS AND HONORS**

- Ranked $24th$ at the Graph500 Supercomputing Challenge Competition, May 2011.
- Won the SuperComputing, May 2009 Storage Challenge for Data Intensive Science: Solving Scientific Unknowns by Solving Storage Problems.
- Chancellor's distinguished fellowship from University Of California, Riverside.

**COMPUTER SKILLS**

- **Languages:** C, C++, Java, Perl, Python, Sql, MPI, OpenMP, CnC.
- **Packages:** MySQL, Postgres, LEDA, CGAL, Boost, Matlab, R.
- **Python skills:** Scipy, Statpy, Numpy, Matplotlib.

**PROPOSAL GRANTS:** Processing Massive Graphs on Triton: Performance, Optimizations, and Applications. Funding Program: Triton Research Opportunity (TRO).

**REFERENCES:** Available upon request.