Sanjeev Kulkarni R
Computer Sciences Dept., University of Wisconsin-Madison
sanjeevk@cs.wisc.edu

Office Address:
1210, W. Dayton Street
Madison WI 53706
(608) 265 3151

Home Address:
#209, 2110 University Av.
Madison WI 53705
(608) 231 0705

Objective: Seeking fulltime job in Networking/Distributed systems/Databases.

Education

- **M.S. in Computer Science**, May 2001.
  University of Wisconsin - Madison.
  GPA = 3.917/4.0
- **Bachelor of Technology in Computer Science and Engg(B.TECH)**, May 1999
  Indian Institute of Technology, Guwahati.
  GPA = 9.17/10.0

Skills
Languages: C, C++, Lisp and Java(to some extent).
Platforms: UNIX (Solaris, HP-UX, Linux, AIX).

Status F-1 visa.
Geographical Preferences: None.

Experience

- **Research Assistant under Prof. Miron Livny**, Univ. of Wisconsin-Madison, Fall99-Present.
- **Summer Internship at Desana Systems**, Summer 2000.
- **Summer Internship at Novell IDC**, Summer 1998.

Projects

- **Webfs: A file system interface to the world wide web**
  We modified the Linux kernel to add a new file system that traps the requests for WWW documents and reroutes them to an external HTTP agent that fetches the documents from the net. The file system was developed as an external kernel module that could be plugged into the kernel at run time. Webfs provided caching for recently accessed documents to speed up accesses to the repeatedly accessed documents.
- **High Level Messaging: A Customised Transport Level Protocol**
  Designed and developed a customized transport level communication protocol. This protocol exposed an event handling mechanism by associating an application specific type to each packet and invoking the corresponding handler on packet arrival. It provided a reliable and sequenced communication library to the applications. It also provided multicasting capabilities. This project was part of my summer internship at Desana Systems.
- **Fault Tolerant Grid computing in an Unreliable Computational Cluster**
  As part of my research with the Condor group under Prof. Livny, I am working on MW Driver which provides a robust, fault tolerant, object oriented platform for developing Master-Worker applications. The applications written on top of MW can run either on Condor-PVM, wherein Condor does the resource allocation-deallocation and PVM provides the communication, or on Condor-Files wherein Condor does the resource allocation-deallocation and communication is achieved via a shared file infrastructure(like afs or nfs). MW acts as the resource manager for applications and takes care of hostadds, hostdeletes and other events that can occur in a distributed environment. It is also capable of intelligent scheduling wherein it

can assign tough tasks to fast machines and simple tasks to the other slower machines. Thus with MW, an application developer has just to concentrate on the task at hand instead of worrying about system events. Also the layered design of MW allows application writers to just relink their programs while migrating from one resource management environment to another.

- **A new Packet Scheduling Algorithm to provide QoS**
  Developed, analyzed and simulated a new DiffServ based packet scheduling algorithm for providing Quality of Service. The new algorithm bounds the interpacket delay between any two packets and is computationally inexpensive. This work was done during my summer internship at Novell.
- **Quilt Query Interface for Niagara XML Query Engine**
  Niagara is a research project in Wisconsin that can be used to query XML data. Our project involved creating a Quilt query interface to this engine so that Niagara can accept Quilt queries and process them. Apart from simple path expressions our implemenation supports complex predicates, FOR, WHERE and RETURN clauses.
- **Reliable Computing in a cluster environment using Worms**
  We developed an "internet worm" like infrastructure and used it to do reliable computing. The main idea was to use the replication capabilities of a worm to ensure the existance of atleast one copy of the computation. We also encapsulated the computation with modified condor's checkpointing libraries that took care of multiple processes reading and writing from the same file. The ultimate goal was to see that the computation terminates successfully amid system crashes and/or concerted attacks by a malicous user. We tested our implementation on a 64 node dual processor P-III linux machines.
- **Study of Split/Coalescing techniques for register coloring**
  In our Advanced Compiler project, we studied register coloring algorithms that split/coalesce nodes of the interference graph. Although a lot of study has been done in this area, none, to our knowledge, explore the possibility of combining both techniques at the same time. In this project we came up with a heuristic to split/coalesce the nodes based on certain conditions.
- **Study of hardware/software prefetching techniques**
  As our Advanced Architecture course project we studied a number of hardware/software based prefetching mechanisms for processor architecture. We also came up with a new technique that improves over the existing LookAhead PC mechanism. We simulated the various hardware schemes and compared their performance.

Publications

- **An Enabling Framework for Master-Worker Applications on the Computational Grid**, written with Jean-Pierre Goux, Jeff Linderot and Michael Yoder, HPDC'2000.
- **Flow Service Order: A Computationally inexpensive Packet Scheduling Algorithm to support QoS guarantees for Real-Time Traffic**, written with Krishanu Seal and Dinakar Sitaram, ICATM99.

Masters Courses

- **Advanced Operating Systems** (CS736) (Fall 99)
- **Advanced Architecture-I** (CS752) (Fall 99)
- **Advanced Computer Networks** (CS740) (Spring 2000)
- **Distributed Systems** (CS739) (Spring 2000)
- **Advanced Topics in Databases** (CS764) (Fall 2000)
- **Construction of Compilers** (CS701) (Fall 2000)

References: Available upon request.