Sai Rama Krishna Susarla
223 Medical Plaza
Salt Lake City, UT 84112
(801) 581-9807
E-mail: sai@cs.utah.edu
WWW Home Page: http://www.cs.utah.edu/~sai/

Education

Degree	Year	Institution	Remarks
Bachelor of Technology (Computer Sc. & Engg.)	1988-92	Regional Engg. College Warangal, INDIA	Percentage: 74% (4th in class out of 65)
Master of Technology (Computer Sc. & Engg.)	1992-94	IIT Kanpur, INDIA	GPA: 9.14/10
Doctor of Philosophy (Computer Science)	1995-present	Univ. of Utah, Salt Lake City, USA	in-progress

Work Experience

Aug 94 - Sep 95: Worked at IBM Corporation, Austin TX in their microkernel development team. Participated in IBM Austin's Embedded Microkernel Project. The work involves tailoring various components of the IBM microkernel for embedded applications.

April 94 - July 94: Software development at Tata Information Systems Ltd., Bangalore INDIA, IBM's (50-50 joint) venture in India with Tata Industries Ltd, one of the largest private sector groups in India.

July 96 - September 2000: Worked at Novell Inc., as a regular full-time employee (Software Engineer II) in their Novell Storage Services group, developing their next-generation file system.

September 95 - todate: Currently a full-time PhD student (and Research Assistant) in the University of Utah School of Computer Science, and leading member of its Computer Systems Laboratory's **Khazana** Distributed Services Infrastructure Project.

Publications and Patents

- Co-authored a paper entitled "*CPU-inheritance scheduling*" which appeared in the Proceedings of the Usenix 2nd Symposium on Operating Systems Design and Implementation (OSDI '96).
- Co-authored a paper entitled "*Khazana: An Infrastructure for Building Distributed Services*" which appeared in the Proceedings of the International Conference on Distributed Computing Systems (ICDCS 1998).
- Co-authored a paper entitled "*Supporting Persistent C++ Objects in a Distributed Storage System*" which appeared in the Proceedings of the ACM SIGPLAN Workshop on Compiler Support for System Software (WCSSS'99).
- Filed a patent (as co-inventor) for inventing a scalable and efficient *Modified Files List* facility to help quickly find the list of files worthy of backup during incremental backup sessions of large file system volumes. Patent filed while at Novell Inc. and is currently pending.

Significant Projects

Doctoral Study, Univ. of Utah

1. Developed a novel thread-scheduling paradigm called CPU-inheritance scheduling that allows thread scheduling policies to be customized by users, significantly improves monitoring and control of threads' CPU usage and helps avoid priority inversion in real-time environments. Published a paper on this research at OSDI'96.

2. As part of my PhD thesis research, currently developing **Khazana**, a novel infrastructure to ease the development of (clustered as well as wide-area) distributed services by providing much of the common functionality needed by them behind a familiar shared state abstraction. This helps rapidly develop distributed services and reduce their time-to-market while enabling reuse. Published 2 papers on this research.

Novell Provo, Utah

1. Designed and implemented the complete File Naming subsystem for Novell's journaled file system NSS (Novell Storage Services) as a single recoverable B-tree per file system volume. It maintains its integrity in the face of file system crashes by journaling/transactioning its operations. It is currently part of Novell's NSS product (July end 1996 - May end 1997).

2. Designed and implemented the file purging and salvaging subsystem for Novell's new journaled file system called NSS (Novell Storage Services) It keeps deleted files around until space is needed; helps recover deleted files). It is currently part of Novell's NSS product (June 1997 - July 1998).

3. Designed and implemented the background file compression subsystem for NSS (Both whole-file as well as chunk-based compression). It is backwards compatible with Novell's previous file system product and also exports a modular and efficient streaming interface for plugging in new compression algorithms into NSS. It is now part of Novell's NSS product (August 1998 - August 1999).

4. Designed and implemented Novell's Smart Backup Services support in NSS that enables scalable and high speed incremental and full-backups of NSS volumes. This is part of Novell's next release of NSS product.

   This includes a **patent-pending** Modified Files List facility that helps quickly find the list of files worthy of backup during an incremental backup session. Patent filed (as co-inventor) with the NSS group's Chief Architect.

   This also includes a new interface that enables efficient high-speed and parallelized streaming of pre-flattened file system data between an NSS file server and third-party backup applications with minimal data copying. (September 1999 - September 2000).

IBM Austin

1. Provided the Intel physical map support for a new virtual memory feature called VM Regions in the IBM microkernel (July to August beginning, 1995).

2. Designed and implemented a mechanism called "in-kernel compression pager" for compressing infrequently used parts of a task's (including kernel's) text/read-only segments and uncompressing them on a page-by-page basis, on demand at run-time. This is for use in IBM embedded microkernel (May to June end, 1995).

3. Modified the Intel pmap to do away with kernel's mapping of entire physical memory into its virtual space during initialization. Instead, implemented a mechanism for mapping physical memory pages into kernel's virtual space on a temporary basis, as and when required (August 1995).

4. Reimplemented the kernel virtual memory mapping mechanism on PowerPC, replacing the use of Block Address Translation with segmented address translation using PowerPC's inverted page table. This was for use in ROM-based microkernel for embedded applications (Jan beginning to March beginning 1995).

5. Made the entire IBM microkernel product (the bootloader, microkernel, personality-neutral servers, User-level device drivers) bi-endian (i.e., able to run in both big- and little-endian modes). This was done for use on PowerPC platform. (Oct beginning to Dec end, 1994).

6. Worked on making the Programmable Interrupt Controller (PIC) driver, a loadable microkernel extension, and wrote the driver for a multiprocessor PIC (March to May end 1995).

7. Fixed many defects in the IPC, VM, and pmap subsystem (of both Intel and PowerPC) of the IBM microkernel, for about two months from August 4, 1994.

Tata Information Systems Ltd, Bangalore, INDIA

1. Worked on porting 4.3BSD file system (UFS, NFS, etc) as a user-level file server on CMU Mach kernel, exporting a VFS-level interface. Completed the design. Almost completed the implementation, at the time of coming to IBM Austin. Worked in a team of two.

2. Provided a lot of technical support to another 2-member team (though not as an official member) in the porting of 4.3BSD sockets (supporting TCP/IP layers) as a user-level "socket server" on Mach for providing network access to user-level Mach tasks.

3. Set up a simple Mach programming environment and conducted training classes on Mach concepts.

Graduate study (3 semesters)

1. **Master's thesis**

   As Master's thesis titled *"Real-time extensions to the Mach kernel"*, incorporated many features of state-of-the-art real-time kernels viz., guarantee-based thread-scheduling for predictable execution times, scheduling policy-mechanism separation for flexibility, user-notification of device interrupts, etc, into Mach 3.0. All basic features and abstractions of Mach were retained. The system supports coexistence of real-time and nonreal-time activities.

2. Implemented a VLSI circuit fault-simulator as a term project as part of "VLSI Testing and Fault tolerance" course, in C++ (during the II-semester).

3. Replaced TFTP's stop-and-wait protocol with sliding-window protocol as the main project of "Computer Networks" course (II-semester).

4. Attended a full-semester course on "Software Engineering" during III-semester. As semester project, developed a package for automation of a national-level undergraduate entrance exam process, using Foxbase database package. Worked in a team of two and followed the standard software engineering processes.

Undergraduate study

1. Designed and partially implemented a C++-to-C precompiler as my **bachelor's degree project**. In this process, studied C++ extensively - both usage and implementation.

2. Developed a full-fledged IBM/370 assembler and its machine code emulator (in software) to run on 32-bit UNIX machines. These are being used for writing and testing IBM/370 assembly language programs on UNIX machine itself. Completed the project in 3 months. That was in II-semester of III-year after taking Systems Programming course.

3. Developed an interactive graphic icon-editor on Apollo workstation using the PHIGS package, during Computer graphics course in I-semester of III-year. Programmed in C.

Computer Experience

Languages: C++, C, Java, Pascal, Fortran, COBOL, LISP, Prolog, VHDL, Intel 80x86 assembly, Motorola 680x0 assembly, PowerPC 60x assembly, IBM/370 assembly.

Operating Systems: Various UNIX flavors (Linux, 4.x BSD, System V,SunOS4.1, AIX, HP-UX, Apollo Domain/IX, 386BSD), OSF/1, IBM's WPX, MS-DOS, OS/2, MS Windows 2000, Novell NetWare.

Hardware platforms: HP minicomputers, Sun 3, Sun Sparc 1, Sun/386, IBM RS/6000, AS/400, S/370, Apollo workstations, PowerPC machines (internal to IBM).

Protocols and Packages: TCP/IP, SUN/RPC, X11, Microsoft WINDOWS, FoxBase, Microsoft Access, Microsoft Word, Synopsys tools, Viewlogic.

Graphics Packages and Windowing environments: PHIGS, GKS, Suntools, Openlook, X/Motif, Gnome, KDE.