Krishna Kunchithapadam

krishna_kunchith@yahoo.com
http://www.oocities.com/krishna kunchith/

Home Address
1615 SW Morrison, #419
Portland, OR 97205
Phone: (503) 228-8000

Current Employment

Software Engineer, Principal Member of Technical Staff, Portland Development Center, Oracle Corporation (Portland, OR) since 8th February 1999.

Objective

Research and Development work in the areas of Systems-Software, Software Tools, and Software Engineering.

Interests

Operating Systems, Programming Languages, Software Engineering, Databases, Architecture.

Personal Information

Date of birth:	2 September, 1970.
Citizenship:	Republic of India.
Visa/work status:	LPR (Legal Permanent Resident) of USA, (green card, full employment authorization).

Education

Master of Science, Computer Sciences (Aug. 1991 - Dec. 1998)
University of Wisconsin-Madison. *GPA*: 3.9/4.0
Passed Qualifying and PhD Preliminary Exams (Operating Systems).

Bachelor of Technology, Computer Sciences (Aug. 1987 - Jul. 1991)
Indian Institute of Technology-Madras, India. *GPA*: 9.5/10.0

Skills

Programming Languages:	C++,C,Java,Standard ML/NJ,Fortran,Pascal.
Scripting Languages:	Perl,Tcl/Tk,XML,SQL,PL/SQL,HTML.
Systems:	Unix (Solaris,SunOs,HPUX),Windows/NT,MS-DOS.
Architectures:	SPARC,HPPA,i860,Intel x86.
System Administration:	Unix, Oracle 8i/9i RDBMS.

Industry Work Experience

Employment (Feb. 1999 - present)
Principal Member of Technical Staff, Oracle Corporation,

Principal engineer in the Oracle Data, Indexes, and LOBs group.

Implemented J2EE 1.3/1.4 compliant JMS provider for Oracle Containers for Java (OC4J).

Implemented J2EE 1.3 compliance for Oracle's Java Messaging Service (OJMS).

Participated in the Architecture Task Force for Oracle's Unified Application-to-Application and Business-to-Business Integration Solution.

Designed a C++ version of the Java Messaging Service (JMS) API for access to the Oracle AQ queuing system.

Implemented extensions to the Oracle JMS API to allow browsing on JMS Topics (multi-consumer queues).

Architected a Messaging Gateway integrating Oracle AQ with non-Oracle queueing systems (MQSeries, TIBCO, MS/MQ) using the Procedural Gateway of the Oracle RDBMS and the Oracle Message Broker drivers.

Architected, designed and implemented the BPC (Business Process Coordinator), an execution engine for high-level, declarative business models describing enterprise-wide application integration.

Coordinated proof-of-concept demonstrations of the BPC at McDonalds, Agra Systems, and Envera.

Provided QA, configuration management, build, and installation support for BPC and OMB (Oracle Message Broker).

Student Intern (Aug. 1997 - Oct. 1997)
Architecture Group, Sun Microsystems, Palo Alto, CA.
Worked on the feasibility of executable-editing techniques to study the performance of the Oracle and Informix database servers.

Graduate School Work Experience

Research Assistant
Executable Editing and Binary Rewriting (*advisor*: James Larus)
Paradyn and IPS-2 projects (*advisor*: Barton Miller)

Teaching Assistant
Unix consultant, CS302, CS536, CS701

List of publications and significant projects attached. Copies of publications, course lists, transcripts, and references available upon request.

Publications

Coping with Type Casts in C
Michael Siff, Satish Chandra, Thomas Ball, Krishna Kunchithapadam and Thomas Reps.
(submitted to the *7th ACM SIGSOFT International Symposium on the Foundations of Software Engineering*, September 1999).

Using Lightweight Procedures To Improve Instruction Cache Performance
Krishna Kunchithapadam and James R. Larus.
University of Wisconsin-Madison Computer Sciences Technical Report 1390, January 1999.

The Paradyn Parallel Performance Measurement Tools
Barton Miller, Mark Callaghan, Jonathan Cargille, Jeffrey Hollingsworth, Bruce Irvin, Karen

Karavanic, Krishna Kunchithapadam, Tia Newhall.
IEEE Computer 28, 11, November 1995.

Integrating a Debugger and a Performance Tool for Steering
Krishna Kunchithapadam and Barton Miller.
Proceedings of the Workshop on Debugging and Performance Tuning for Parallel Computing Systems, October 1994.

Optimizing Array Distributions in Data-Parallel Programs
Krishna Kunchithapadam and Barton Miller.
Proceedings of the 7th International Workshop on Languages and Compilers for Parallel Computing, August 1994.

Projects

Implemented a collection of optimization algorithms, based on executable rewriting (using the EEL/PP toolkit), to improve the instruction cache performance of programs on a superscalar, pipelined architecture (UltraSPARC in particular). The collection of programs studies ranged from the SPEC benchmark suite (integer/floating point), to X-windows applications, to large database servers (Oracle/Informix/DB2).

Implemented a source-to-source translator from Java to C++. The translator (consisting of a parser, AST, and code generator) was used to study the performance of (interpreted) Java programs compared to (compiled) C++ equivalents. The runtime support included a reference-counting garbage collector.

Implemented a collection of algorithms, data structures, and animation frameworks for Computational Geometry in 2 and 3 dimensions.

Designed and implemented a framework (based on the Paradyn Performance Measurement System) for performance data collection and semi-automated, self-tuning algorithms for programs based on dynamic performance information.

Implemented a portable, synchronously-scheduled, lightweight thread library (in C). The library is currently used extensively in the Paradyn Tool.

Implemented a type inference algorithm for a Object-Oriented, functional programming language. Also implemented a code-generator, exception architecture, and memoization for the language targeted at a RISC-like virtual machine.

Implemented a code-generator backend for the SPARC architecture and to be used as part of the LCC compiler.

Implemented a syntax-macro system to extend the syntax of the C language with a wide-range of arbitrary extensions. Demonstrated the use of the syntax macros to implement exceptions, function closures, and new statement types in C.

Implemented a file-access library based on B-tree and Hash indices for use in a *toy* database system. The library was designed to be highly configurable in terms of buffer cache sizes, file page sizes, buffer cache page replacement policies, indexing and hashing algorithms and used in studying the effects of these parameters on the performance of a database system. The library is in use in the *Minibase* database package developed at the University of Wisonsin-Madison.

Implemented a connection-oriented, reliable, TCP/IP stack on top of a connectionless, unreliable, unordered messaging layer.

Ported the IPS2 Performance Analysis tool to the Solaris-2.x system. This involved instrumenting the C library and the MT lightweight threads library to collect detailed performance information for use in subsequent analysis.

Implemented a process migration system for checkpointing the state of a running Unix process on one machine and restarting it on another. Demonstrated the system by implementing a *WORM* capable of reliably completing an application in multiple copies across a network of workstations, with the ability to tolerate failure of hosts and processes.