Stephen A. Broeker
steve_broeker@yahoo.com
www.linkedin.com/in/stevebroeker

SYNOPSIS

Senior Software Architect.
Expert in Unix, embedded OS, storage systems, high performance file systems, high performance embedded databases, real time applications, online analytical processing, web services, and cloud storage and services.

OPERATING SYSTEMS

Apache
Linux
VxWorks
Cisco IOS
AT&T Plan9
Tandem Guardian
AT&T UNIX
BSD UNIX

EDUCATION

Santa Clara University, Computer Engineering PhD Candidate, 9/2007 – Present.
Thesis: A Stream Star Schema and Data Value Cube for Data Stream Analysis.

Aurora University, M.S. Computer Science, 2/1989.
Thesis: An Analysis of Polynomial and Generalized Learning as Applied to the Game of Score-Four.

Pacific Lutheran University, B.S. Mathematics, 5/1982.
Thesis: Chebyshev's Approximation to Polynomials.

PATENTS

A Stream Star Schema and Data Value Cube for Data Stream Analysis.
A High Speed Packet Capture File System – Reconnex.
A High Speed Packet Capture Data Base – Reconnex.
A Linux C Apache Web Server – YouSendIt.
A Reliable Shared Memory File System for Inter-Processor Communications – Rasvia.

EXPERIENCE

Internap, Santa Clara, CA
Cloud Architect
April 2011 to Present

Cloud Architect for the Cloud Products group. Cloud Products are a new Internap offering in the data center. Cloud Products are based on OpenStack Compute (Nova) and Storage (Swift). I am primarily working on Swift. I am currently building Swift Monitor daemons that monitor the state and performance of Swift. This monitoring includes the Swift proxies and zones. I am also investigating Swift performance improvements. These include database schema changes and storing Objects in raw disk partitions as opposed to file systems. Since OpenStack is written in Python, I am implementing my work in Python.

Quantum, San Jose, CA
Architect
December 2009 to February 2011

Architect for the infrastructure group, in the storage division. Quantum products are based on 64-bit Linux platforms that provide data movement – data duplication, data replication, and archives.

Architect for system upgrade, web services, cloud services, system database, and system configuration.

Changed system upgrade to use dual boot partitions to enable remote upgrades and enable upgrade rollback.
Converted system database from Linter to MySQL.
Changed system configuration to use a database repository via SQL and XPATH using MySQL. This provided configuration transaction capability.
Converted system management from a CLI to Web Services. We started with SOAP and ended up with REST via Apache.
Created the Quantum Cloud API, which is a superset of the Amazon S3, Microsoft Azure, Google Storage, and Iron Mountain cloud APIs. Used Apache to process HTTP requests. Employed MySQL to handle name space mapping and to store metadata.
Designed the POSIX to Cloud Storage interface. This allowed a Cloud Storage Account to be mounted as a POSIX like file system. Used libcURL to manage connections. Employed MySQL to handle name space mapping and to store metadata.

McAfee, Santa Clara, CA
Senior Architect
September 2008 to December 2009

McAfee acquired Reconnex in September 2008.
Chief Architect for the Reconnex product line.

Reconnex, Mountain View, CA
Senior Architect
November 2003 to September 2008

Designed and implemented Internet Security Systems. These products are based on a real-time object capture engine. As flows come in/out over the Ethernet, they are cataloged (in a database) and stored (in a file system). These products primarily consist of applications code running over Linux on 64 bit Intel Systems.

Platform Team Manager.

Responsible for the file system, database, search control, system configuration, and the process model.
My work was written in C and KSH.
Designed and implemented a process model that is responsible for application control. Includes startup, shutdown, memory dumps, logging, status, statistics, IPC, and alarms.
Designed and implemented an IPC mechanism that is based on shared memory. Includes queues, statistics, control, and a debugger.
Designed and implemented a file system that is modular ande handles a sustained write throughput of a gigabit per second. This file system is used as the object store on the local disk array and on external storage.
Designed and implemented a search control mechanism that covers the object store. Includes job control and results store.
Designed and implemented a star schema to facilitate search results performance when viewed by the GUI.
Designed and implemented a TCP/IP based message system that is used as a reliable, persistent communication path between systems.
Designed and implemented the system configuration subsystem that is based on MySQL.

Awarded Outstanding Engineer, 3rd Quarter 2004.

YouSendIt, Mountain View, CA
Consultant
February 2006 to May 2006

YouSendIt builds Web Caches - both Windows and Linux based.
I implemented a Linux C Apache Module that supports downloads and uploads. The Linux Apache Module was originally in Perl. I improved system performance from 200 simultaneous requests to over 2,000 simultaneous requests. This module includes debugging and statistics.

Rasvia Systems, Sunnyvale, CA
Senior Software Engineer
July 2001 to October 2002

Rasvia built multi-processor network storage systems.
They employed embedded OS (VxWorks 5.4) on the PowerPC cpu (750).
Designed and implemented an OS API layer between VxWorks and Rasvia applications. This work covered the following subsystems: task, malloc, mutex, binary semaphores, counting semaphores, message queues, print, and symbol table. Included a full set of diagnostics, regression, and display for each subsystem.
Designed and implemented the shared memory manager. Included a mutex facility with full debugging support and a commit to disk feature.
Designed and implemented a highly reliable file system. This was used to manage the system area in shared memory and all disk labels.

Member of the Logical Volume Manager Group.

Helped design and implement the logical volume manager admin layer.
Designed and implemented a kernel debugger that was used on memory dumps. Included OS API support and system area commit support.
Designed and implemented a kernel profiler.
Designed and implemented a system diagnostics cacility. Configurable for stand alone and runtime diagnostics.
Designed and implemented a system event logging mechanism. Employed a single log for multiple cpus. Configurable for compile time and run time.

Cisco Systems, San Jose, CA
Technical Lead
January 1997 to July 2001

My work revolved around Cisco's embedded, real time OS - IOS.

Designed a modularity methodology for IOS. This allowed IOS libraries to be tested on a workstation so that software could be validated prior to being loaded on a router. Significantly reduced lab costs and provided a regression test for development.

Designed and implemented an IOS Kernel Debugger that could be entered via the monitor, the console, or a memory dump. Included full virtual memory support: malloc, memory maps, and memory leak detection. Included full process support, formatted printing of kernel data structures, symbolic disassembler, symbol address translation, and a 64 bit calculator. The kernel debugger was fully configurable and ran on its own stack.

Member of a team that redesigned the port adaptor subsystem to employ a virtual function table. This allowed the platform dependent code to be separated from the platform independent code. This greatly simplified port adaptor bring up.

Designed and implemented a ROMMON flash memory dump mechanism. This allowed system memory to be dumped to flash cards from the boot prom.

Ported IOS ENA (Unix based micro kernel) to the 4700 platform. This required rewriting the following subsystems: NIM, EPROM2444, 4700 PCI, AMD9970 Ethernet Driver, Lance Ethernet Driver, and HD64570 Serial Driver. This was the first non-reference port of IOS ENA.

Member of the team that implemented E1/R2 on the 5300 platform. The work was based on TI/CAS. I was responsible for changes to the VTSP subsystem.

Senior IOS Engineer in the RPM group.
The RPM is a blade version of the 7200 router that acts as a line card in the ATM switch (MGX). The ATMDX (ATM port adaptor) is used to connect the RPM to the MGTX back plane. Hence, communication between the RPM and the switch controller (PXM) is done via ATM.

Responsible for ATM performance on the RPM.
Made changes so that full OC3 line rate could be achieved.
Identified design changes needed to allow IOS to support 4K IDBs.
Enhanced kernel profiling.
Designed and implemented an IOS diagnostics subsystem. Employed both hardware and software diagnostics. Allowed diagnostics to be run at boot time, in the background, and from the CLI. Errors were reported back to the PXM. This subsystem was required for RPM diagnosability and redundancy.
Designed and implemented an ATM DX microcode debugging mechanism. Allows ATM DX debugging from IOS. Includes ASSERTs, event logging, code tracing, and viewing of ATM DX data structures.
Designed and implemented an RPM/PXM memory dump mechanism. Cisco routers rely on IP connectivity to perform memory dumping. The RPM is often times not directly connected to the IP, so obtaining memory dumps was impossible. This mechanism allowed RPM memory dumps to be stored on the PXM hard drive.
Increased RPM diagnosability by adding CLI commands to display the following hardware info: NVRAM, MIPS Cause, Status, Config registers, and TLB.
Held tech talks on the following: IOS ENA, port adaptor subsystem. memory leaks, IOS virtual memory, and NVRAM.
Project lead for RPM-400 bring up. This entailed porting the RPM to the MIPS R5000 cpu. Included 3-way cache, new system controller, new flash, new memory DIMS, no SRAM.
Designed and implemented a mechanism to manager ECC Single Bit Errors.
Designed and implemented an RPM/PXM error message facility. Allows the RPM to report errors to the PXM.
Designed and implemented an RPM/PXM heart beat mechanism. Provided a reliable polling service for PXM to RPM traffic.
Designed and implemented an RPM system image checksum mechanism. Helped prevent boot errors.
Designed a facility to allow the RPM to perform memory dumps to a disk drive port adaptor.

Senior Engineer in the IVR group.

Designed and implemented an IVR malloc mechanism - to track memory usage and help detect memory leaks.

nCUBE, Foster City, CA
Senior Software Engineer
December 1995 to January 1997

nCUBE systems use a proprietary 64 bit processor (running Plan9) in an MPP architecture.
These systems are used as scaleable parallel processing engines and scaleable IO engines - video pumps and web servers.
Responsibilities included kernel development (SCSI, VM, COMM), bug fixing, and performance measurement and improvement.
Designed and implemented 90% of the OptiVision High Speed Quad - a SCSI based MPEG decoder.
Increased performance of the SCSI disk driver by 10%.
Designed and implemented performance measurement tools for the SCSI disk, remote SCSI disk, NCMP, and MDS file system subsystems.
Added performance measurement hooks to the kernel subsystems - devwren, devscsi, qlogic, NCMP, and TCM.
Helped create a microsecond timer for the kernel - accounted for context switching and interrupts.
Measured Video Pump performance and identified bottlenecks for ATM delivery, UDP delivery, SCSI raid set size, and SCSI string size.
Designed and implemented a mechanism to poke a hung CPU into the kernel debugger - required changes to boot code.
Designed and implemented a mechanism to dump memory from a CPU - required a new TCM driver.
Designed and implemented a kernel debugger that could be entered via panic, the console, a user program, or a memory dump. Included full virtual memory support: xalloc, kmalloc, palloc, memory maps, memory leak detection, virtual to physical address translation, and physical to virtual address translation. Included full process support, kernel breakpoints, formatted printing of kernel data structures, symbolic disassembler, symbolic address translation, and a 64 bit calculator. The kernel debugger was fully configurable and ran on it's own stack.

Tandem Computers, Cupertino, CA
Software Designer
July 1994 to December 1995

Senior OS Engineer in the Guardian Development Group.
Tandem systems use multiple processors in a message based OS that is fault tolerant on three platforms: proprietary chip, MIPS T3000, MIPS R4000.
Responsibilities included kernel development and bug fixing.
Designed and implemented a kernel managed swap subsystem that is fault tolerant. Included the ability to add/remove an active swap disk and disk down processing.
Responsible for rewriting the memory initialization subsystem. This consisted of moving code from system to the boot sequence.

Pyramid Technology Corporation, San Jose, CA
Senior Software Engineer
July 1989 to July 1994

Senior kernel engineer in Current Product Development.
Pyramid systems use up to 24 processors (SMP based) on three platforms: a dual port of BSD 4.2 and AT&T V.3 on RISC Chips (OSx), AT&T V.4 on the MIPS R3000 (CPL), and AT&T V.4 on the MIPS R4000 (NILE).
My mentor was Richard Hammons.
Responsibilities included enhancements, performance improvements and bug fixing the kernel.
Other responsibilities included training engineers.
Designed and implemented an expansion of the kernels virtual address space for OSx.
Designed and implemented a kernel diagnostics harness for OSx, CPL, NILE.
Increased the capabilities of the system boot strap for OSx.
Project Lead for the dynamic run time I/O distribution interrupt mechanism for CPL, and NILE.
Project Lead for CPL Large system bring up (13-24cpus).
Project lead for a redwing of the kernel stack and user area for CPL - resulted in a major increase in system diagnosability and reliability.
Project lead for a queued spinlock mechanism which replaces simple spinlocks during high contention - required for large CPL.
Responsible for porting Computed Associates UnixCenter to CPL and NILE.
Responsible for porting Software AG Adabas to CPL and NILE.
Designed and implemented a diagnostics mechanism for kernel malloc space (CPL and NILE).
Designed and implemented a kernel diagnostics harness for OSx, CPL and NILE.
Project Lead for a user level spinlock mechanism which included cache alignment, lock testing queued locks, and statistics gathering.
Project Lead for a redesign of the dispatch queues for large CPL and NILE.
Fixed numerous bugs and added copious debugging capability to the following areas of the kernel; virtual memory, heap and map facilities, setjump, scheduler, debugger, and spinlocks.
Successfully performed emergency on site surgery at the following mission critical sites: O'Conner and O'Conner in Chicago, US West in Seattle, and AT&T Bell Labs in Illinois.

Received the following awards:
Outstanding Software Engineer - 4th Quarter 1989.
Pyramid Quality Award - 1st Quarter 1990.
Current Product Achievement - February 1991.
SMP Significant Contributor - April 1993.
Pyramid Quality Award - July 1993.
Pyramid Builder's Club 1993 - included an all expense paid trip to Cancun.

AT&T Information Systems, Lisle, IL
Tier IV UNIX Hotline Consultant
May 1986 to July 1989

Provided customer hotline support for UNIX (V.2, V.3) the language subsystem (C, C++, FORTRAN) and various add-on products (Documentor's Workbench TUXEDO, Writer's WorkBench) on the 3B20m 3B2, 3B15, 3B4000, and 6386.
Fixed kernel and utilities bugs.
Other duties included system crash recovery, system administration (analyzing sar data and optimizing kernels) and writing articles for the monthly software support newsletter.
My mentor was Paul Jatkowski.

UniSoft Corporation, Berkeley, CA
Programmer
July 1985 to May 1986

My work revolved around porting BSD 4.2 UNIX to 6386 platforms.
Responsible for applications porting.
Included compilers (C, FORTRAN, BASIC, COBOL, ADA), data based, and spread sheets.

BASYS Inc., Mountain View, CA
Programmer
February 1984 to July 1985

My work revolved around a newsroom computer system.
Designed and implemented a generic wire copy program in C. Involved implementing a communications protocol between an X80 and an X8000 and data capture on an RX232 async port.
Designed and implemented a console multiplexer (in C) to monitor I/O ports on an IBM PC. Involved writing a terminal emulator and rewriting and optimizing the ports drive in Intel 8088 assembler.
Revised the file system structure (ISAM) to utilize key compression, resulting in up to 70% space reduction.