Michael Hunter

66405 Cline Falls Road
Bend, OR 97703
tahoemph@gmail.com
504-493-5569

Education

MS Mathematics CSULB, June 1994

BA Mathematics Occidental College, June 1989

Employment

Principal Engineer Say Media October 2015 - Present

Team Member
Built the backend and participated in the frontend implementation of a Facebook posting feature for the Say Media publishing platform. The backend was in Python and the frontend in AngularJS. I was the only member to contribute to all repos involved.

Team Member Migrated a subscription feature from an in house integrated Facebook login to one built on top of Auth0. This involved integration with an AngularJS frontend application and a Nodejs based AWS Lambda backend.

Team Leader Lead the Advertising team at Say Media. This included creating a new advertising product which was able to be trafficked by our customers and run on a variety of different DSPs, migrate our data system from running on top of map/reduce to one built on top of Redshift, Kinesis Firehose, and S3 with some custom ETL scripts running in ECS written in Python, and managed day to day ads and data problems. Other technologies we used for delivering ads included Python for general backend services, AngularJS for internal tools, Nodejs for high performance asset delivery, and various AWS services (CloudFront, S3, EC2) for deployments.

Team Member, Architecture, and Leadership
Helped migrate a number of microservices from our data center to AWS. Initially we were using core services such as EC2, ELB, ASG, S3, and RDS. As time went on we added Redshift, CloudFront, SNS, Dynamodb, API Gateway, and others. I've become one of the AWS infrastructure knowledgable engineers at Say. We used the Hashicorp tools for managing out infrastructure.

Principal Engineer iMatchative January 2015 - June 2015

Leading the migration of a financial analytics website onto a modern stack in order to increase feature velocity. The stack was SQL Server, Python, and AngularJS.

Software Architect Say Media September 2012 - January 2015

Initially my role at Say Media was as a Senior Software Engineer working on an internal sales applications written using Django. I learned the codebase, learned the business, and helped those tools grow with the business.

The business was transitioning away from additional internal tooling to building a custom CMS to support our own sites. Later that grew to supporting an expanding set of sites we brought onto the platform. In the broad sense I was part of a small group of architects responsible for choosing and applying technology. My specific focus was on the backend services (Python, MySQL) and on the CMS admin (composing and other editorial functions in AngularJS). We built an editor using content editable which allowed the user to build a representation of the document. We then convert that into a renderable representation based on the target environment. The user had the ability to do normal editorial operations such as applying various types of styling and putting various embeds into the document.

A peer and I designed the concepts behind grouping and representation of non-article data when faced with a site with a large amount of non-article data (biography.com). The core concept was a type with base properties allowing us to display a basic representation on the site. We are able to (and have) extended that type to support other uses including our basic grouping operation. I helped design an editor around introspection of that type which allows us to quickly bring up the ability to build content.

Some of my coding responsibilities during this time included bringing up the initial site migration system and migrating several sites, designing and implementing the new custom type on the back end including relevant API, extending how we manage cache invalidation, prototyping parts of the introspected editor, root causing many of our deeper problem, and architecting and helping design the majority of the backend and admin software.

Senior Data Engineer Zephyr Associates January 2011 - September 2012

Developed an ETL system to replace an existing data pipeline built on DTS using Python 2.7 and SQL Server. The pipeline is used to process roughly 60 data feeds into more than 100 databases which are shipped to Zephyr's customers. I've reduced the time from getting a data feed to shipping a database by a factor of 2. Am currently working on building a production system so that we can further increase our rate and reliability of database production.

Helped replace an aging database install mechanism with one built on C# and Install Shield.

Helped replace an older ISAM database format with SQLite.

At the end of my tenure Zephyr's database schema was denormalized and hard to grow. My research included mechanisms to be used for reducing the amount of curating we need to do and mechanisms to be used for scaling our databases in size without breaking backwards compatibility.

Staff Engineer Sun Microsystems, Inc (Oracle) June 2002 - July 2010

Updated the IPv6 basic API (RFC 3493) and implemented the IPv6 advanced API (RFC 3542).

Worked as an integral part of the IPv6 team working to deliver significant upgrades and bug fixes for the Solaris 10 release.

Worked on NWAM (Network Automagic) from conception through several releases. The focus of NWAM is to simplify and automate network configuration. For many cases software should be able to figure out what the user wants to do. For many others the set of choices are fairly small and the user shouldn't have to dig through uninteresting options. By combining a set of daemons which implement policy and a UI which provides the user with the ability to control that policy NWAM is able to significantly simplify network configuration. The team struggled to find the right balance between the common case (a laptop) and wanting to be able to specify more complicated situations (for example, when the user encounters a certain network NWAM should automatically create a tunnel to be used for communication). I provided an early prototype which later grew into an initial delivery vehicle. Later I worked on providing components for a more complete implementation.

Senior Software Engineer Extreme Networks December 2000 - April 2002

Implemented a memory allocator and buffer management library for a MIPS R4000 operating a T3 line card. I used a fixed buffer layout choosing cacheability based upon memory type in order to be able to reach line rate.

Ported the Telenetworks Frame Relay stack to a T1 blade. We were mixing and matching versions of the Telenetworks Frame Relay stack and the Telenetworks PPP stack running under a custom executive. I used a buffer scheme involving variant types of headers and lazily switched between them under application control in order to get differing buffer schemes to work together.

Added Cisco HDLC support to a T1 blade.

Implemented EDP (PPP Control Protocol)

Senior Software Engineer Redback Networks August 2000 - December 2000

Implemented Cisco HDLC for a pre-release optical line card running on custom processors. These processors provided fast access to packet memory but very slow access to the rest of memory. All coding had to be done in assembly.

Designed and implemented an infrastructure to request information from a line card. This included a framework to help the user (developer) implement the CLI and marshall messages on the host processor and a framework on the data processor to parse the requests and return responses.

Technical Lead Cisco System February 1999 - August 2000

Delivered Cisco HDLC to a pre-release system aimed at environments with high reliability and scalability needs. Cisco HDLC was used to provide interoperability with existing platforms. It is a very simple protocol that was used to test most system features such as restartability.

Provided design input, implementation, and debugging support for interface management in the same environment. Interface management was complicated by the need to manage multiple types of interfaces across multiple processors connected by a variety of media from fairly slow to extremely fast.

Designed and implemented BACP and parts of VPDN for the same environment. This was made difficult by needing to support both local and non-local interfaces.

Software Design Engineer QNX Software System Ltd.June 1995 - August 1998
Manager TCP/IP Technologies Augugst 1998 - February 1999

Brought a port of the BSD 4.3 Reno TCP/IP stack to market. Responsible for the TCP/IP stack, utilities, documentation, and support.

Wrote a TCP/IP stack for a new version of the QNX OS (Neutrino). This stack supports a single network interface, IP, ICMP, UDP, and TCP. My stack met its code budget of 40K of x86 code. Later added multiple I/F support.

Ported the BSD 4.4 stack to QNX 4. Since QNX 4 is a microkernel architecture this involved wrapping the BSD 4.4 stack in an emulation of its native kernel environment.

Wrote various utilities for Neutrino including a general system interrogation utility (super ps) and a filesystem manager to support unix pipes.

Select telecom stack vendor and guided port.

Responsible for TCP/IP technology development from choice of which technologies to pursue to detailed design and scheduling of coding tasks.

Wrote HTML to Troff (man macros, tbl) converter.

Senior Software Engineer Air Touch Teletrac January 1993 - May 1995

Developed a distributed real time simulation under QNX 2.x in support of white box testing a RF switch.

Led the development of a transition plan from QNX 2.x to QNX 4.x.

Evaluated and test communication and computing hardware for a multiprocessor small footprint system.

Ported a complex communication simulation from Quick BASIC 4.5 to Visual BASIC 3.0 under very tight schedule.

Participated in the full development of an autonomous RF switch to replace operator intensive one. This switch will be hosted on a STD32 backplane w/80[45]86 class processors running QNX 4.x. Wrote RF scheduling, generic state transition, and telecommunications code (TCP/IP, async). Developed lib code for message handling and error logging in c for team. Many of my own processes are written in C++.

Build a QNX 4 IO Manager that allowed for redirectable and multiplexed I/O to/from a variety of sources.

Worked as QNX 4 Guru and design expert in a team environment.

Software Design Engineer Microcosm, Inc. June 1987 - January 1993

Developed requirements document, helped design, and wrote parts of Space View, a spacecraft geometry visualization tool. Initial version being written on Sun SPARCstation using the OpenWindows GUI in C++. Designed and implemented an inter-program communication scheme utilizing TCP/IP to exchange data with a related product written by a different company.

Designed, coded, and tested real time flight software for an autonomous spacecraft. Wrote math utilities, system executive, and Mil-Std-1553b output handler. Integrated software on a Mil-Std-1750a target and 1553b bus.

Designed, documented, and implemented real time, flight hardware in the loop, simulation engine in Mil-Std-1815a (Ada) under VMS. Wrote orbit propagator to provide a reference trajectory for spacecraft simulation. Presented aspects of the simulation to customer at design reviews.

Co-author of a winner Phase I SBIR proposal applying a rule based system to space attitude determination and telemetry data editing.

Lead programmer of a spherical geometry visualization tool implemented on an IBM PC in standard c under MS-DOS. Worked with Hercules/EGA/VGA graphics and OS interfaces.

Skills

Soft: Architecture, Team Leadership, Software System Design, New technology evaluation, Pragmatic getting it done

Current Languages: Python, Javascript (many past ones waiting to be rebooted)

Systems: Unix, AWS