

Résumé for Ben Cottrell

1. Contact information

Email: tamino@wolfhut.org
Phone: 408-616-8832
USPS: Ben Cottrell
547 W Washington Ave.
Sunnyvale, CA 94086

2. Experience

Nominum Inc. — November 2006 - current

Senior Software Engineer

- With one other senior engineer, designed and implemented various back-end components of Nominum's DNS-in-the-cloud offering, called Skye. The actual DNS requests coming from the Internet are serviced by Nominum's standard DNS server products, but the team I was part of wrote the code that configures those servers based on the more abstract information in the database about how users have configured their accounts. A minimal set of deltas is calculated when preparing an update, so that if one of the DNS servers is offline while the same DNS record is added and removed many times, when the server comes back online the commands sent to it to bring it up to date only include at most one copy of the record. This is done in a way that does not require entire DNS zones (which could be millions of records) to be loaded into memory at once. Potentially large amounts of statistics are also gathered from the DNS servers.
- With one other senior engineer, designed and implemented a new library (in Python) for the Skye front-end to use to retrieve and modify configuration and DNS zones. This library was also used by a customer-facing HTTP (REST) API, and it was designed to work well for both callers. It was designed to be faster and more scalable than the code previously in use, and to allow batches of records to be operated on at once, with as few round-trips to the database as possible.
- With one other engineer, designed a schema language that all of Nominum's products now use to express in abstract form all of their configurable parameters, documentation associated with each parameter, and the data type of the parameter. Data types may be arbitrarily nested, e.g. a stub entry might contain a list of nameserver declarations, and a nameserver declaration might be defined as a tuple consisting of a DNS name and a list of nameserver addresses, each of which is an IPv4 or IPv6 address literal. Implemented a library (in C) for traversal and manipulation of descriptions written in this language. Nominum's "Engine Administration Console" product (which I also contributed to) was able to connect to running servers and download their schema, and present an HTML-based interface for configuring the engine.
- Worked intensively over a period of weeks to solve stubborn stability and deadlocking issues in EAC. This required a large amount of C-level debugging, even though EAC itself is written in Python. This resulted in submitting <http://bugs.python.org/issue3710> and <http://tinyurl.com/26vmooo> (the latter being a PySQLite mailing list post), as well as finding and fixing several other issues traceable to Nominum code.
- Worked on Nominum's Centris product, which is an extension to the Nominum Authoritative Name Server (ANS) which schedules periodic downloads (by rsync or HTTP) of zone data instead of using standard IXFR and AXFR mechanisms. This project was written in a mix of C and Python.
- Worked on other projects at Nominum, in both Python and C, as needed.

IronPort Systems Inc — June 2003 - November 2006

(IronPort was acquired by Cisco Systems in June 2007)

Software Engineer

- Implemented features and fixed bugs in IronPort's Mail Gateway Appliance line of products. The work was done in Python using cooperative multithreading (similar to that of Stackless Python). This involved less overhead than pre-emptive multithreading, but required engineers to maintain a high level of awareness of exactly how their code would be run, and what assumptions about scheduling they could and could not make.
- Worked on the IronPort Anti-Spam product. Designed and implemented tools for increasing the spam catch rate. This included a tool which could run a set of spam rules against hundreds of thousands of sample messages four times faster (in some cases up to ten times faster) than the tool previously in use. It also included tools for heuristically grouping spams into categories (with no programmed-in knowledge of what the categories were) and for reading in large numbers of sample spams and sample legitimate email and automatically searching for patterns with high predictive power for whether messages are spam. For the latter two especially, the code was designed to be easily extensible even by junior engineers.
- Implemented large amounts of the LDAP client code, including load balancing and failover between different LDAP servers.
- Worked as part of a small team which brought the centralized management feature (clusters of appliances sharing configuration data, automatically resynchronizing after network failures, and making administration commands network transparent) to maturity. For the majority of this time I was the primary person fixing bugs, analyzing failures on customer networks, and increasing stability. Performed root cause analysis on distributed lock contention issues and unpredictable failures that occurred as the cluster sizes grew larger.
- Starting with requirements documents from marketing, wrote detailed engineering specifications, some of which I implemented, and some of which were implemented by other engineers.

Mail Abuse Prevention System (MAPS) LLC — January 2003 - June 2003

Software Developer

- Took care of the engineering part of adding a new DNSBL (named Open Proxy Stopper) to the MAPS product line.
- Designed and implemented (in perl) a scalable, extensible tool for running dozens of security tests on hundreds of IP addresses simultaneously using nonblocking I/O. At first the tool was used by MAPS staff members; then it was incorporated into the back end of the Open Proxy Stopper code.
- Sped up incoming nomination processing by a factor of 4 on the production Relay Spam Stopper systems and by a factor of 12 on the new Open Proxy Stopper systems. I achieved this both by making incremental changes where I saw opportunities, and by writing drop-in replacements in perl for some of the existing Bourne shell scripts (bug-for-bug compatible with the Bourne shell code).

Nominum Inc. — December 2000 - January 2003

Software Engineer

- Added features and fixed bugs in Nominum's DHCP server product, DCS. I designed and implemented the processing of options in DHCP packets. The options code allowed the administrator to create their own data types by combining together primitives. This project was written in C and was single-threaded, making extensive use of callbacks to store information about what should be done next after I/O completed or the next packet from a given client was received.
- Worked on Nominum's Global Name Service DNS hosting (GNS). I added features to the web-based GUI (written in PHP), and the back end (written in perl).
- Worked on the Nominum Identity Director project, which produced a prototype of a DNS and DHCP appliance providing a centralized management interface to control DNS and DHCP at thousands of remote sites. I designed and implemented its system configuration subsystem.
- Assisted with QA, including writing automated test cases to verify RFC conformance.

Stockmaster.com / Red Herring Magazine — June 1999 - December 2000

Software Engineer

- Starting from a prototype, designed and implemented a system for receiving real-time stock trading data, aggregating price information on a minute-by-minute basis, and saving it to disk files to be used by the company's web site. This project was written in C and C++, with a nightly processing program written in perl that aggregated older data into progressively coarser-grained time buckets.
- Tracked down a memory corruption bug in Apache/mod_perl, and submitted a patch to the project.
- Maintained legacy systems while new ones were designed and implemented.

Realize Communications — December 1997 - June 1999

(Realize was acquired by Stockmaster in June 1999 — see above)

Software Engineer

- Worked as the lead programmer on the Realize web site, which provided web-based message boards to users on the Internet. This included writing custom code to monitor the health of individual web servers and load-balance requests among the ones that were healthy. The code was in C and C++ and ran on both Windows and Solaris.
- Added the ability to do incremental writes, with journaling, to the in-house-developed database code. After my changes, doing a write to the database caused the on-disk hash table and free block list to be updated. Also added replication.
- Set up FreeBSD to be the mail server and DNS server for the company.

3. Skills

- Good at writing solid, fast Python and C.
- Experience with UNIX system calls and the standard library, including TCP/IP, sockets, Berkeley db, forking and execing processes, signal handling, mmap, etc.
- Experience using profiling tools such as gprof, and acting on the results.
- Experience with both single-threaded and multi-threaded programming. For example, I am equally comfortable writing a select loop, or using blocking I/O in multiple threads.
- Experience at the implementation level with various protocols, including DNS, DHCP, HTTP, SMTP, and SSH.
- Some experience with perl and Bourne shell.
- Some experience with SQL.
- Familiar with UNIX-as-a-philosophy; writing small sharp tools that work well together.

4. References

Marshall Kirk McKusick, coauthor, *The Design And Implementation of The 4.4BSD Operating System*

Ted Lemon, author, Internet Software Consortium DHCP server; coauthor, *The DHCP Handbook*

5. Education

Ending in spring 1997

Computer science major at the University of California at Berkeley. Although I did not complete the BS program, I completed all required courses in the major.