Darren Gilroy - Software Engineer
gilroy@gmail.com

Summary of Qualifications

Technical leadership driven by customer success.

- **C++** – Most recent 5 years, writing distributed services in C++17
- **Java** – 15 years until 2014, in all frameworks and application types
- **Perl, Python, Bash** – supported production services
- **SQL** – expert: 20 years of advanced SQL. PostgreSQL, BigQuery, Oracle
- **Motivated** by getting something shipped that people use and like

Professional Experience

Cedexis, Inc. – August 2014 - present (acquired by Citrix)
Principal Software Engineer. Lead developer of the platform team, responsible for the maintenance and growth of our internet-scale DNS and HTTP services.

When I joined, the codebase was five years old but still in startup mode: cross-dependencies and point solutions. The architecture was great; reliability and agility were suffering.

- Started and maintained the practice of code review, unit testing, and continual refactoring
- Reduced our set of supported languages to just C++, and normalized all the development and build environments
- Managed a zero-defect queue by maintaining low work-in-progress and focusing relentlessly on any customer-impacting bugs
- Drove down our release cycle from quarterly to weekly, leading directly to improved system reliability
- Decomposed a monolithic code base into half a dozen services, enabling faster iteration and safer deployments
- Improved reliability to the point where the CEO stopped showing the "Customer Impacting Downtime" side in all-hands

These services together handle 15 billion requests per day, about evenly split between DNS and HTTP. They run on machines in six cloud providers. My team runs 12 services, which in turn use a similar number of shared code modules.

- Reduced memory requirements by rewriting core data structures, in one case from 8GB/service to 1.5GB/service
- Lead service rewrites resulting in lower machine counts, in one case a reduction from 64 to 18 hosts
- Dropped p99 latency 80% (!) by replacing a central lock with atomic updates plus a form of garbage collection (QSBR)
- Replaced a kafka/spark cluster running on AWS with some SQL scripts, lowering operational costs by 75% and reducing the need for specialized training
- Transitioned our transaction database from PostgreSQL to BigQuery, including a rewrite of the HTTP API to make it possible for portal to remove embedded SQL
- Ported away from bespoke threading, http, JavaScript, and similar infrastructure code to off-the-shelf libraries
- Rewrote global configuration glue code from python to C++, dropping SLOC by 2/3 and reducing update latency from 3 minutes to 20 seconds for our 40 global data centers
- Split a single large common-code library into single-purpose modules, making source-level dependencies clear at build definition time
- I learned C++ for this job

UTi Worldwide – July 2013 - August 2014
Senior Software Engineer. Lead a small team to develop accounting reconciliation software for end-of-month close. Mentored two junior developers.

Jive Software – October 2011 - February 2013
Senior Software Engineer on the architecture team.

- Spent six solid months working a big bug backlog to get the team out from under some debt
- Performance improvements in caching tier

- Designed and implemented a new permission system to allow bulk checks against query results in both SQL and Lucene

Tripwire, Inc. – April 2007 - October 2011
Senior Software Engineer. Lead architecture team at Tripwire.

- Converted load process from a concurrent to a job-parallel system, reducing lock overhead and improving performance by 20%
- Improved in-house ORM tool to generate better SQL queries
- Implemented a tree in SQL to allow bulk loading of arbitrary-depth trees with single statements
- Designed a message-passing architecture so new features could be developed outside of the large central code base. It was successfully implemented by a different team.
- Rewrite RPC system to improve failure conditions and just generally simplify the code managing remote job state
- Designed an asynchronous message passing architecture for the Tripwire agent to replace java RPC, and lead a team team to implement it in C++

Fujitsu America – May 2002 - April 2007
Software Engineer. Designed and implemented software for global website consolidation across all Fujitsu countries. Implemented code review and coding standards, established unit testing and component development.

Consonus, Inc. – July 1996 - April 2002
Lead Developer at Consonus for six years; developed Internet e-commerce applications.

–December 2019–