

Austin hicks Phone: available
 upon request
e-mail: Home page:
ahicks@ahicks.io <https://ahicks.io>
GitHub: camlorn Twitter: camlorn38

Location: Seattle

Programming languages: Python 2 and 3, C/C++, Rust, Javascript, Lua, Go

Skills: Google Cloud Platform, Kubernetes, Django, Jinja2, Django REST Framework, Celery, RabbitMQ, PostgreSQL, Etcd, Redis, Ubuntu, Helm, Git, SIMD and other low-level optimizations (SSE2), compiler development, parallelism, functional programming, algorithms and data structures, CMake, accessibility, Markdown, LaTeX, Windows API and Microsoft COM

CI platforms: Travis CI and Appveyor

Graduated Florida Atlantic University Summa Cum Laude with a BS of Computer Science, class of 2015. One of two students recognized by the president of Florida Atlantic University for outstanding achievement.

Highlights

- Architected an Amazon Web Services to Google Cloud Platform migration for a large production system including live migration of databases.
- Designed and implemented multiple production-grade microservices.
- Designed and implemented a custom microservice broker/monitoring solution capable of handling 10000 requests per second.
- Made significant contributions to the Rust compiler.
- Designed, implemented, and helped to maintain multiple billing systems.
- Have significant experience writing mathematical code in C++, including custom parallelization algorithms and SIMD optimizations.
- Wrote approximately 75% of a WebAudio implementation targeted at desktop apps.
- have used both Travis CI and Appveyor for release management on a complex C++ project with custom build steps and nontrivial dependencies.

FutureTech Industries (Engineering Manager/Senior Software Engineer, 2018-2019 and Software Developer, part time 2016-2017)

- Interfaced directly with clients to determine requirements, refine stories and tickets, and address client concerns.
- Assisted in determining business direction and goals.
- Participated in product development and market research.
- Designed and implemented 2 billing systems. The most complicated of these implemented everything necessary to be a SaaS provider, including automatic tax handling and invoice charging, usage tracking, and full auditability of all decisions.
- Wrote a production-quality Helm chart for PgBouncer, coauthored articles explaining its proper configuration, and assisted in operating a social media campaign garnering 30000 unique reads.
- Advised clients on Amazon Web Services to Google Container Engine (managed Kubernetes) migration.
- Converted a complex, manually provisioned QA environment to a reusable Kubernetes Helm chart as a precursor to automated provisioning/deprovisioning from inside CI workflows.
- Designed and implemented a custom metrics shipping pipeline powered by Go and GRPC, capable of handling multiple metrics sources which achieved ~3000 points/second per deployment of the agent. Metrics were shipped to sharded PostgreSQL databases managed by a Django-powered microservice responsible for assigning customers to shards and offering a custom JSON query/aggregation API.

- Participated in market research for and designed proprietary autoscaling technologies.
- Designed and assisted in the implementation of custom BigQuery ETL pipelines that talked to 3rd-party services. The most notable of these shipped data from 3Scale, a Redhat-owned API gateway and authentication SaaS offering. We achieved more auditability than the official support channels were able to demonstrate without modification of 3Scale or running our own deployment.
- Designed BigQuery schemas and wrote large BigQuery queries for auditing billing and API usage data.
- Prototyped a distributed request throttling solution powered by Redis which could prevent overloading third-party services with API rate limits.

Cielo24 (Research And Development, 2017-2018)

- One of two developers responsible for architecting and executing an Amazon Web Services to Google Container Engine (managed Kubernetes) migration with minimal downtime on a compressed timeline.
- Trusted with direct production access for all system components.
- Assisted in reducing release times from hours to minutes by improving automation.
- Deployed Deis in production environments, a Heroku-like solution for release management on top of Kubernetes.
- Tested an implementation of distributed locks, proved that it was nonfunctional, and replaced it with a Redis-backed solution.
- Implemented a microservice proxy and broker with the following characteristics:
 - Provided a standardized HTTP-based API abstracting over multiple backend calling conventions.
 - used properly configured Kubernetes rolling deployments and health checks, allowing for zero downtime releases and limited self-healing capacity.
 - Recorded all requests to all microservices within a configurable rolling window.
 - Handled up to 10000 requests/second in bursts.
 - Provided a graphical admin for microservice configuration and request history.
 - Worked around HTTP timeouts for requests that would take hours to complete via a heartbeat model.
 - Provided facilities for sending multiple streams of data in both directions where necessary.
- Implemented a massive autoscaling media transcoding cluster powered by the kubernetes jobs API which could run media transcoding on preemptible VMs.
- Wrote a custom extract-transform-load data processing pipeline to move large amounts of data from legacy systems to Google BigQuery to enable exploratory analysis.
- Administered PostgreSQL:
 - Planned and executed a cross-cloud PostgreSQL migration of a large database.
 - Managed a deployment of PostgreSQL on top of Kubernetes using Stolon. Configured proper health checks and read replicas.
 - Performed query optimization through the use of the slow log, explain, and hypothetical indexes.
 - Rolled out and administered PgBouncer, a connection pooling solution for PostgreSQL.
- Administered RabbitMQ:
 - Planned and executed a zero downtime cross-cloud RabbitMQ migration using the shovl plugin.
 - Planned and implemented a high availability RabbitMQ configuration using the federation plugin.
- Implemented a custom priority queue on top of Redis responsible for assigning tasks to human workers, to replace a prior implementation that didn't scale to meet business requirements. This queue supported task priority updates, task cancellation, task preview, and task timeouts, and improved user-facing responsiveness under load by a factor of 10.
- Assisted in the design and implementation of a microservice for progressively streaming long-running reports (~4+ hours) to the browser, primarily related to billing.
- Introduced proper healthchecks to application components allowing for self-healing capacity.

Rust compiler Development (2016 - 2017)

- Implemented an optimization with proven real-world impact that reduces the size of Rust data types by reordering fields to eliminate padding.
- Removed significant duplication in the compiler's internal representation of data types, a precursor to further work such as repr(transparent) and further niche-filling optimizations for enums.
- Introduced the concept of optimization fuel, which can be used to binary search the compiler's decisions to find bugs.

Libaudioverse (open source, C++, author and primary contributor, 2014-2018)

- An open source DSP library written in C++ for audio synthesis, not dissimilar to WebAudio. Design independently converged with the WebAudio spec later in the library's lifecycle.
- Implements approximately 75% of WebAudio's capabilities for desktop apps.
- Builds releases with Appveyor and Travis CI based off git tags.
- Implemented a node execution engine capable of parallelizing audio synthesis work.
- Implemented the basic building blocks necessary for audio synthesis, including optimized waveform generators, convolution nodes, and white/brown/pink noise implementations.
- Wrote optimized mathematical kernels for convolution and other DSP-related tasks.
- Wrote custom build tooling to generate data tables and automate repetitive coding. Notably, these tools took yaml descriptions of audio nodes and generated documentation and C++ boilerplate for implementation.
- Wrote basic optimized IIR filters and a feedback delay network.
- Implemented a basic high quality reverb algorithm.
- Worked on a basic HRTF implementation.
- Wrote audio I/O code with support for multiple backends. Wrote backends for ALSA, Wasapi, and WaveOut.

DictationBridge (Open Source C/C++, developer, 2017)

- Implemented the initial version of a solution to get JAWS for Windows (a screen reader) talking to Dragon Naturally Speaking (a speech recognition solution), enabling visually impaired individuals with additional physical disabilities to more easily utilize computers.

Simplemail (Florida Atlantic University Software Engineering Project, Team Leader/Head Developer, 2015)

- Led 4 other students in the design and development of a prototype-quality mail inbox provider as part of CEN 4010 Principles Of Software Engineering.
- Set up and administered GitHub repositories, domain names, servers, and other infrastructure for my team.