Alexey Yevtushenko

alexey.yevtushenko.work@gmail.com | github.com/isnastish | linkedin.com/in/alexey-yevtushenko *Stockholm, Sweden*

Summary

Software engineer with 3 years of experience writing C/C++, Python and Golang, specializing in backend and microservice development. I have a solid background in algorithms and data structures and enjoy writing good, maintainable code which solves complex problems. I'm most confident working with Golang and Python at a lower level of tech stack, including network communication via different protocols, but I'm agnostic to any programming language and have experience with modern technologies like Redis, PostgreSQL, gRPC, common cloud providers, containerization and best security practices.

Education

Sumy State University *2019/09 – 2022/04 (Paused)*

Bachelor of Science in Computer Science *Sumy, Ukraine*

- **Courses**: Algorithms and data structures in C++, object-oriented programming in Java and Python, relational databases, compilers, computer networks and more.

Ubisoft *2022/06 – 2023/08*

Work under supervision, mentorship. *Ubisoft Stockholm studio*

- **Topics**: C++ pair-programming. Code optimizations, profiling, distributed systems, microservices.

Work Experience

Ubisoft *Since 2022/06*

Junior Software Engineer *On-site/Stockholm, Sweden*

- Core developer of Scalar, a cloud-based game engine for creating massive multiplayer games.
- Maintained and developed internal sdk for building C++ projects, containerizing and deploying Python services.
- Enhanced a distributed entity component system (ECS) for synchronizing entities in the game world over the network, improving the performance, benchmarking.
- Implemented a backend for AWS S3 storage both in Python, using async version of AWS sdk, and in C++, using libcurl. The storage was used for caching http requests made to the services producing multimedia content.
- Maintained and improved custom eventing system service in Golang. The system utilized multiple NoSQL databases for storing events: FaunaDB, Firestore and Redis, depending on the underlying cloud provider.
- Wrote a gRPC python client to interect with the eventing system service for publishing events and listening for updates from multiple event sources.
- JWT authentication, both in python and Golang, established mutual TLS connection between services.
- Wrote technical documentation, constantly participated in code reviews.
- Implemented cloud mutex for mutually exclusive access to cloud resource between services. The lock was acquired by publisning `ACQUIRE` event and released by publishing `RELEASED` event to the earlier mentioned eventing system.
- Wrote python library to interact with cloud buckets regardless of the underlying cloud provider.

Projects

Distributed key-value storage github.com/kvs-storage *Since 2024/05*

- Designed and implemented a service for storing key-value pairs in memory, aka Redis. Wrote both Golang and Python clients to interact with the service over HTTP using REST api.
- The service utilizes gRPC framework for communication with another service, responsible for persisting transactions. Transaction tracking is needed to recreate the state of the storage in case of abnormal shutdown.
- All transactions are persisted in a Postgres (using PGX driver/toolkit) or in a binary file.

Distibuted, multi-client chat github.com/chat *Since 2024/03*

- Wrote a distributed chat application which supports multiple backends for persisting messages from connected clients.
- The backend uses my custom KVS service and Redis (as an alternative), as well as in-memory storage for local development. Service is run inside a docker container listening on TCP/IP protocol to handle incomming connections. Chat server is designed to operate as a finite-state machine (FSM).

...continues on the next page...

Skills

Programming languages: C/C++, Golang, Python, SQL
Technologies: Docker, CI/CD, Git, Linux, AWS S3, gRPC, GTEST, PostgreSQL, SQLite, Redis, FaunaDB, HTTP, REST, Flask/Quart, FastAPI.

Last updated 2024/10