ALES HUZIK

+61 427 990-909 ⋄ me@aguzik.net
Sydney, Australia

EDUCATION

Belarusian State University of Informatics and Radioelectronics, Minsk *July 2013*
Faculty of Computer Systems and Networks
M.S. in Computer Science & Software Engineering

Belarusian State University of Informatics and Radioelectronics, Minsk *July 2012*
Faculty of Computer Systems and Networks
B.S. in Systems Engineering

CERTIFICATION

CO0401EN Beyond the Basics: Istio and IBM Cloud Kubernetes Service *December 2019*

HAVE EXPERIENCE WITH

Operating Systems	Linux (NixOS, Archlinux, Debian, CentOS, Ubuntu), Mac OS X
Programming Languages	Clojure, Ruby, C, Bash, JavaScript, Perl, Common Lisp, Erlang, Maude, Factor, Lua, Terra, C++, Python and some others
Libraries and Frameworks	React, Reagent, core.async, clojure.spec, prismatic/schema, mount-lite, Compojure, Kioo, Enlive
Markup and Typesetting	LaTeX, HTML, Haml, Slim, CSS, SASS/SCSS, Less, Bootstrap
Cloud	AWS (EC2, S3, IAM, RDS, DynamoDB, Route 53, EKS, ECS), Docker, Kubernetes, DigitalOcean, Dokku, Heroku
Server technologies	NixOps, Consul, Chef, Nginx, Apache httpd, statsd Postfix, Prosody, vsftpd, Squid, Corosync, Pacemaker, DRBD
SQL Databases	PostgreSQL, MySQL/MariaDB, SQLite
NoSQL Databases	Datomic, DynamoDB, Neo4j, OrientDB, MongoDB, Cassandra
Programming paradigms	Imperative, Object-oriented (class-based, prototype-based), Functional, Concatenative (stack-based), Logical (predicate logic, rewriting logic).

WORK POSITIONS

Atlassian, Confluence Server January 2018 - Present
Senior Developer (Synchrony, Clojure) *Sydney, Australia*

- Analyzed L1-L2 support tickets related to the project, identified that majority of them are related to a troublesome Synchrony server setup in the Data Center configuration, came up with the solution to implement automatic management of Synchrony by Confluence in DC to eliminate the need for separate Synchrony server setup, and collaborated with Confluence Server Scale team to implement it.
- Fork Synchrony to maintain a version for Confluence server, separate from Confluence Cloud, as teams are very far and don't communicate, and projects have very different infrastructure and needs. Removed cloud-specific build steps, code and configurations (docker cloud tests, building PaaS jar, cloud loadtest, s3 and dynamodb storage backends, redis caching, cloud-specific encryption, ec2 elb node discovery, aws logging, PaaS statsd, cloud metrics, etc.)

- Build and development setup simplifications and improvements (move to just leiningen instead of bash+gulp+leiningen, remove most leiningen profiles, don't compile java, don't preprocess JS)
- Refactored the code from component to mount-lite. This led to significant simplification of the code, with more explicit graph of dependencies, and significantly improved ability to work with code interactively from the repl.
- Collected all configuration decisions in a single namespace. Previously many configuration options existed in a configuration namespace, but a lot of things have been using environ directly or using other means to detect certain runtime configurations (e.g. running in a cloud PaaS environment, on dev machine, in loadtest etc.). This allowed to ensure that differences between code paths in dev, test and prod environments are explicit and kept to a very minimum.
- Implemented generic managed cluster-shared state as atoms on top of hazelcast's IMap. This allowed working with this distributed state the same way as if it were a normal local clojure atom, getting notifications when the state changes, using all clojure standard library functions for atoms, etc.
- Found a workaround for a bug in core.async that causes exceptions to be thrown from an incorrect stack frame (https://clojure.atlassian.net/browse/ASYNC-198)
- Implemented hub locking for the data eviction project.
- Created simple and extendable cli tooling for all development, build, test, release and other tasks.
- Worked with support team to investigate customer issues

Atlassian, Confluence Cloud December 2016 - January 2018
Senior Developer (Synchrony, Clojure) *Sydney, Australia*

- Code health improvements (code reviews, eliminating tech debt, improving development workflow)
- GDPR and data eviction project
- Pushed for changing synchronization data from linear to tree format for the new editor integration, so standard operational transformation logic won't break the document structure. Implemented a tree diffing algorithm to support tree format synchronization
- Proposed and added jvm memory consumption metrics, so we can see when GC happens and can understand it's implications on the dynamic behavior of the system
- Decreased application bundle size twice (from 100mb to 50mb), which made deployments noticeably faster
- Calculated cluster startup dynamodb usage and increased limits accordingly, which allowed deploying during peak hours with no downtime (previously, deploying during peak hours could lead to downtime up to half an hour).
- Designed and implemented an automated versioning project
- Participated in interviewing potential candidates
- Have been helping with onboarding of new team members
- Led synchrony architecture bootcamp
- Participated in the team on-call rotation
- Participated in a company-wide hackathon (Ship-It) and got into finals

Filemporium/Ourmedian July 2015 - December 2016
Lead Clojure Developer *Remote via Upwork*

- Interviewed potential candidates
- Regularly did code reviews
- Documented project structure, project-specific code conventions, technical decisions, troubleshooting, and Amazon S3 project-specific step-by-step configuration guide
- Did pair programming (to assist others with complicated tasks, to share project knowledge, to get back on track when I'm stuck)

- Revised architecture in a way that drastically simplified client-side state management and allowed live page update of all active user sessions
- Reengineered project build system using boot (previously leiningen were used). Fixed project build time (full recompilation now takes just a couple of minutes instead of an hour). Adjusted project code to work with reloaded workflow.
- Refactored most of the project, implemented lots of functionality and fixed lots of bugs (e.g added config schema validation, cleaned up garbage logging (like `(println "!!! FOO:" x)`) and implemented propper configurable logging throughout the system, implemented chunked file upload with an automatic reconnection, etc.)
- Setup temporary deplyoment via docker and dokku
- Implemented production-ready multiserver setup with zero-downtime deployment using NixOS/NixOps and Consul
- Let go of a programmer that have been writing terrible code

Filemporium/Ourmedian May 2015 - July 2015
Clojure Developer *Remote via Upwork*

- Automated design updates
- Added compile-time template checks to kioo templating library to ensure component correctness after a design update
- Added support for using arbitrary npm libraries from ClojureScript code (to be able to utilize existing js React components)
- Started writing project documentation. Documented actions needed to setup a project, update the design, add an npm library

Softswiss Casino Software October 2014 - May 2015
Senior Software Engineer *Minsk, Belarus*

- Implemented integrations with external game providers (CasinoTechnology, Fengaming)
- Implemented completely custom design for a new customer (HTML/CSS)
- Worked on an external wallet api implementation

Rubyroid Labs, LLC April 2014 - September 2014
Senior Software Engineer/Team Leader *Minsk, Belarus*

- Designed application architecture
- Managed project development
- Did code reviews
- Solely implemented some internal services

Intetics Co. July 2013 - April 2014
Senior Software Engineer *Minsk, Belarus*

- Made fully-automated production server setup
- Worked on refactoring legacy codebase
- Worked on security-related features (IP whitelisting, XSS testing)
- Implemented backend service for mobile apps.
- Implemented automatic management of VPN servers DNS rotation
- Did code reviews

Belarusian State University of Informatics and Radioelectronics February 2013 - January 2014
Teaching assistant at Electronic Computing Machines Department (part-time) *Minsk, Belarus*

- Taught first-year students programming in C.
- Taught fourth-year students IP networking.
- Taught students how to use Git and GitHub.
- Together with students formalized grading criteria.
- Formalized some code quality metrics.
- Regularly reviewed students' code.
- Taught Linux for interested students in my spare time.

PowerMeMobile, Inc. January 2013 - February 2013
Problem solver *Minsk, Belarus*

- Gave an idea of automating deployment process (new tier deployments may take up to a month of SysAdmin team work).
- Implemented initial stages of deployment automation (installing base cluster software, configuring corosync/pacemaker, installing and configuring DRBD and nginx as resource agents) using Chef.
- Made entire deployment configurable from a single place (from chef workstation using node attributes).
- Got an agreement on opensourcing this efforts.

Altoros Systems, Inc. October 2011 - September 2012
Software Engineer in Ruby department *Minsk, Belarus*

- Proved that custom multisite functionality is a bad idea. Dropped the hacks and refactored application to use rails 3 engines.
- Participated in porting internal RightScale services (mostly sinatra+cassandra) to JRuby to utilize native Thrift.
- Participated in all stages of design and development on many projects.

Itransition, Inc. February 2011 - October 2011
Junior Developer in Ruby department *Minsk, Belarus*

- Solely ported large social gaming engine from Rails 2 to Rails 3.
- Initiated using SCSS and Compass, which led to stylesheets development and modification speedup.
- Configured production server from scratch and setup automated Capistrano deployment.

SOME FACTS TO BETTER UNDERSTAND WHAT KIND OF PERSON I AM

- I decided to tie my work to computers when I was 5.
- First program in BASIC at age of 11, first HTML and JavaScript at 12, first program in Pascal at 13.
- I started playing with Linux when I was 14 (it was Mandrake 10 in 2005)
- I use Dvorak keyboard layout
- I use NixOS, and maintain a few packages in it. Before NixOS I used to use Archlinux.
- For many years I used to use tiling window managers and a very minimalistic desktop setup. I switched to Gnome Shell just over a year ago though.
- I use Emacs since 2010. Since 2011 I use it with Evil (vim emulation layer)
- I have dozens of personal opensource projects and have contributed to upstream of at least 20 other.
- One of my primary interests is expressivity of programming languages. I hope to study it as a PhD student one day.

GROWTH DIRECTIONS

- Statistics and Machine Learning
- Semantics of programming languages
- Programming music (Overtone, Extempore) and visuals (Quil, Processing, Fluxus)
- Creating electronic music