

Igor Kaplounenko

senior software engineer

author of `pytest-patch` and `modulemeta`

https://github.com/IcarusMJ12	icarus.aquinas.mj12@gmail.com
---	--

Special Qualifications

Languages

- Python with Tornado, Flask + Connexion
- C++17 with STL and Boost
- CoffeeScript, LiveScript with React or Vue

Other Technologies

- TensorFlow, neon, nGraph
- OpenAPI
- Docker, Kubernetes
- elliptic curve cryptography, NaCl
- SQLite, MySQL, PostgreSQL, Redis
- Linux, FreeBSD
- AWS S3, EC2, Lambda
- Git

I am familiar with a variety of other languages and technologies as well, but am primarily a LiveScript, Python, and C++17 aficionado.

Work Experience

2020-2023 Senior Software Engineer at Lambda Labs

- Maintained the web store by adding features and fixing bugs as requested by the sales and finance teams. (*Python, PostgreSQL, JavaScript*)
- Refactored the codebase and extended it with tests, style checks, and type hints. (*Python, pytest, mypy, Nox*)
- Designed and implemented a customizable web-based GUI wrapper for arbitrary command line tools on behalf of **South Park Studios** that was used for their deepfake YouTube series. (*LiveScript, Vue, Pug*)
- Implemented lambda cloud monitoring. (*AWS Lambda, Python*)
- Implemented the DGX store pages. (*JavaScript*)
- Added minor features to the Deep Learning as a Service cloud platform. (*Elixir*)

2016-2020 Cloud Software Engineer at Nervana / Intel

- Designed and implemented a profiler for Intel's proprietary machine learning chip. (*C++17*)
- Designed and implemented a microservice for hosting AI chip profiling data. (*Python, Flask, Connexion*)

- Contributed to the NGraph TensorFlow bridge, including porting to OS X, refactoring, implementation of various operations, pip packaging, and the C/C++ API. (*Python, C++11*)
- Contributed to the Machine Learning Toolkit, primarily by refactoring the CLI to automatically load `argparse` commands from modules via metaclasses. (*Python*)
- Designed and implemented a service and a client library for neural network hyperparameter optimization and experiment tracking, intended to interface with the like of SigOpt and MOE. (*Python, Flask, Connexion*)
- Contributed to the deep learning `neon` framework and associated web services for deep learning in the cloud, particularly CLI refactoring, 100% unit test code coverage, launcher consolidation for different types of jobs, and a multitude of bug fixes. (*Python, Flask, Kubernetes*)

2015-2016 Software Engineer at Hipmunk

- Designed and implemented an admin panel for managing server alerters for PagerDuty. (*Python, Tornado, CoffeeScript, React, Lodash, Bootstrap*)
- Implemented a bot detection system that uses browser profiling and geoIP tracking to tarpit and feed fake data to bots. (*Python, Tornado*)
- Interfaced with Qatar's flight search API. (*Python, Tornado*)
- Implemented the second version of the United Airlines' booking API. (*Python, Tornado*)
- Implemented the generation of booking links from Sabre travel network's API. (*Python, Tornado*)

2012-2015 Python Developer at BitTorrent, Inc.

- Designed and implemented a UDP distributed share server to provide an API for uTorrent clients to help seed pieces of large torrents. (*Python, Tornado*)
- Implemented the distributed share API on the uTorrent side. (*C++*)
- As part of the uTorrent Core team, helped fix bugs in the uTorrent codebase. (*C++*)
- Implemented features for the SoShare product on both client and server-side. (*C++, JavaScript, Python, Tornado*)
- Was one of the core contributors to the daemon component of the chat client Bleep, which functions as a SIP server with a cryptography layer. (*C++11, NaCl, curve25519, ed25519, SQLite*)
- As part of an Agile team, worked on a number of features and bug fixes in Bleep on Windows, Android, OS X, and iOS.

2010-2012 Senior Software Engineer at Sony Network Entertainment, Inc.

- Implemented a simulator of a PSN video store and was responsible for extending it with prototypal functionality. This was showcased at E3. (*Python, Tornado*)

2008-2010 Software Engineer at Skiff, LLC

- Designed and implemented a WebKit plugin for mp3 audio playback on an embedded device. (*ALSA, MPG123, C, C++*)
- Ported `frotz` (a Z-Machine implementation) to an embedded device and implemented a WebKit plugin to interface with it. (*C++*)
- Designed and developed a New York Times crossword engine. (*JavaScript*)
- Engineered a library to dynamically insert ads into publications as they are displayed. (*C++, SQLite*)
- Collaborated with other teams to integrate libraries and plugins.

- Debugged third party Linux drivers that would cause the system to intermittently hang during boot. (C)
- Ported OProfile to a particular architecture and added support for hardware timers. (C)

2008 Lead Software Engineer at Genesis Interactive

- Designed and implemented two server-side applications that poll third party APIs for information and make it accessible to the client-side app over HTTP. (Python)
- Wrote an administrative API plugin for Openfire XMPP server that allows remote user management and administration.
- Designed and implemented a RESTful API that interfaces a client-side application with a variety of third party services.
- Wrote a variety of server administration-related scripts. (bash, Python)
- Wrote an LDAP login module for Bugzilla. (Perl)
- Installed, configured, and administrated sendmail, Subversion, Bugzilla, Twiki, OpenLDAP, and integrated them together.
- Set up Nagios and wrote monitoring scripts for NRPE. (bash, Python)
- Set up and administrated Amazon EC2 servers as well as all in-house servers, running a selection of Fedora Core 7, Gentoo, and FreeBSD.

2007-2008 Software Engineer at Yahoo

Yahoo Messenger

- Designed and implemented an XML API to serve widget metadata to Yahoo Messenger widgets.
- Designed database schema for storing widget metadata. (MySQL)

Yahoo Photos

- Designed and implemented a distributed web API for the transfer of Yahoo Photos metadata to affiliates. (C++, MySQL)
- Implemented the logic for the user landing page for Yahoo Photos migration.
- Troubleshooted a variety of problems with the pre-existing internal photos API. (C++)
- Designed and implemented monitoring solutions for Yahoo Photos migration servers. (bash, MySQL)
- Designed and implemented a customer care tool for YPhotos accounts.
- Designed and implemented a landing page internationalization tool. (Python)
- Designed and implemented bulk account information retrieval tools. (bash, C++)

2006-2007 Software Engineer at AnchorFree

- Worked as an embedded devices developer for mipsel architecture routers running a modified version of OpenWRT.
- Designed and implemented a buffered HTML parser and editor to be used as a proxy module. (C, JavaScript)
- Designed and implemented a URL redirector to be used as a proxy module. (C)
- Tweaked and optimized an existing proxy. (C)
- Maintained a Subversion repository as well as a Bugzilla installation.
- Designed and implemented a collection of device registration and monitoring tools. (C, perl, bash)

- Ported TCL and Expect libraries to the mips/mipsel architecture.
- Designed and developed a remote administration tool based on inverse SSH to enable administration of routers used inside LANs. (*C, bash*)

2005-2006 Software Engineer at Coherent Inc.

- Worked as a developer for a series of tools that were responsible for extracting data from laser test stations and putting it into the database, as well as making the data accessible via Apache/Tomcat.
- Designed and implemented data processing and analysis algorithms for the test stations (*C++*)
- Designed and implemented scripts to perform operations on databases. (*Jython*)
- Constructed and used Hyperion/Brio reports.
- Created scripts for data extraction from the laser testing station. (*Python*)
- Set up a MySQL database.

2004-2005 Software Engineer at UCSD Bioengineering

- Worked as a developer on an application titled Continuity that modeled hearts in 3D using the finite element method. (*C++, C, Python, OpenGL, Fortran*)
- Designed and implemented numerical algorithms emulating heart operation in a computer model. (*C, Fortran*)
- Performed memory usage optimizations for matrix computation algorithms.
- Designed and implemented top-level control structures as well as GUI forms for Continuity. (*Python, Tkinter*)
- Ported Continuity from Linux to Windows.
- Worked on socket level communication between the client and server side of the application. (*Python*)
- Migrated the codebase from CVS to Subversion and restructured the source code.

2003 Software Engineer at Electric Power Research Institute

- Worked as a developer on a climate-modeling application COSMIC.
- Designed and implemented the GUI for the program.
- Troubleshooted and corrected the program's numerical functions.

2003 Technical Writer at Coherent Inc.

- Designed and implemented a help system for a laser testing program written in LabView.

Personal Projects

pytest-patch An automagic patch fixture that can monkeypatch objects directly or by name.

modulemeta a Module metaclass to turn your classes into modules.

For committed developers, or at least those who definitely ought to be.

dualshock.pd A Pure Data extension that captures Sony DualShock input and translates it into human-usable form.

Comes with a rudimentary AM/FM synthesizer, a duophonic synthesizer, a noise synthesizer, and a drumkit. Fun but not especially practical.

ebb the Expressly Better Bencoder! (*C++11, Boost PP*)

Used internally at BitTorrent, this is a heavily template-based header-only library that simplifies encoding data into a JSON-like format. Bdecoding is also supported and is functional for straightforward data structures, but needs more work to be compatible with .torrent files.

bevel A multitool for commandline processing of bencoded entities with grep- and sed-like functionality. (*Python*)

Also used internally at BitTorrent for viewing and editing bencoded entities either manually or in a scripted environment.

Education

Major BA in Computer Science in the Math Department at UCSD

Minor Interdisciplinary Computing and the Arts with Music emphasis

- Completed a Cryptography I course on Coursera as offered by Stanford University.
- Over a year's worth of coursework in Digital Signal Processing and Synthesis as it applies to sound, incl. graduate level work. (*Pure Data, C, C++*)
- Familiarity with computer graphics, esp. raytracing and radiosity techniques. (*C++, OpenGL*)

references available upon request