

## Experience

### BBC

Engineer • Aug. '19 –

Building and maintaining the BIDI CDN for BBC iPlayer.

Engineering leadership aspects within role;

- Took on many operational responsibilities on top of my day-to-day work, including:
  - Became someone that members from other teams approach with questions about BIDI & Radix
  - Became someone that operational teams came to when issues were raised or when a go-ahead needed to be given to perform operational or maintenance tasks.
  - I am generally one of the first people to notice incidents and always work to troubleshoot and resolve them.
  - Trained to be a First Aider and a Mental Health First Aider to support my colleagues.
- Mentoring members of my own team as well as members of other teams.
- Helped apprentices with their coursework - this ended up being a multi-hour, multi-day investment as a colleague and I had to teach them Python basics and the linked list data structure.
- Being part of a team of engineers from multiple teams helping provide experience and expertise to another team while they worked on hiring and training new members.
- Security considerations during development.
- Playing an active role in many design decisions.
- Guiding team members towards possible solutions to problems.
- Actively seeking out and working on prototyping work for new features.

Some features I designed and/or implemented include;

- **Feedback loop system** – Built version 1 and 2 of our feedback loop system which is used by the platform to balance traffic between sites and remove failing servers automatically.
- **Intelligent sharding of content** – this feature directs clients to caching servers more likely to already have a cached copy of the content being requested.
- **Cache tiering** – this feature is an extension of sharding that allows the caching server to request content being requested from a nearby server rather than origin if it is not in the local cache. I did the initial prototyping in Lua and C, as well as simulation work.
- **Origin retrying** – this feature allows for retrying of origins based on various failure scenarios. Retrying based on a configured value or based on the number of origin servers in the pool for that content type. This was implemented in Lua directly in OpenResty.
- **Origin switching** – this feature allowed our caching servers to swap their origins during client requests if an error occurs without interrupting the client. This feature is similar in spirit to Origin retrying but is actually wildly different and overrides the retry behaviour for the origins it is enabled on. This was implemented in Lua directly in OpenResty.
- **Path-based origin switching** – like Origin switching but transparently swapping origin based on the request path rather than when encountering an error. This was implemented in Lua directly in OpenResty.
- **mVIP** – this feature expanded our caching unit design from having n servers behind a switch with a single VIP to each server broadcasting its own VIP and its neighbours using BGP. Allowing individual servers to act independently but also act as a failover for their neighbours. I worked on the initial prototyping and testing the resiliency methodology, as well as some of the implementation.
- **High throughput server performance tuning** – evaluation, implementation and verification of multiple Linux kernel, AMD CPU and PCIe NVMe performance tuning strategies. Increased server throughput by 48%, improving energy efficiency and cost of delivery.
- Moved our entire test and build process from Jenkins over to AWS CodeBuild for all of our projects.
- **Prototype ISP Portal** – allow ISPs to view machines embedded within their network and perform some admin actions, i.e. scheduled withdrawal from service.
- **Prototype ARM machine** – worked with R&D to build, tune and loadtest an ARM prototype for our next generation of server hardware.
- **Prototype measurements integration using InfluxDB** – replaced our old Collectd + Statsd + Graphite measurements system with Telegraf + InfluxDB (querying with Flux), including dual running both solutions until proven good. As part of this I also had to redesign our measurement names and data, recreate all of our graphs, and redo our alerting system.
- New API endpoints and functionality in the centralised control tooling.
- Optimisation and troubleshooting multi-server clustered AWS Aurora databases.

### Contactor/Consultant

Various roles/positions • Jan. '15 – Aug. '19

Working contracts of varying lengths. These contracts involved systems administration, systems engineering, Python, and some information security.

### DreamHost

Cloud Architect Contractor • Oct. '14 – Jan. '15

Contracted to work on OpenStack at DreamHost for their DreamCompute offering. This role involved both developing OpenStack itself within the company and helping to run and debug the infrastructure.

### Tangent Plc

Lead Systems Engineer • Aug. '09 – May '14

- Designed, built and maintained virtualised hosting platform using VMWare.
- Customised Ubuntu heavily to keep system resource usage as low as possible.
- Management, maintenance and troubleshooting of over 250 virtual machines and around 50 physical boxes. Mostly running Ubuntu, and CentOS.
- Installation and configuration of a Nagios cluster for monitoring including writing an SMS alert relay Provisioning using Puppet, later moving to Salt Stack.
- Using VMWare templates and the VMWare API.
- Mentoring, helping and advising the development team of around 30 developers on subjects such as Linux (Debian, Ubuntu, CentOS and RHEL), security, Python, MySQL and PostgreSQL.
- Acting as a sort of technical architect, helping with platform designs including hardware and software.
- Introduced a lot of tech that the company had otherwise not used PostgreSQL instead of MySQL, Pound, HAProxy, Varnish, nginx, Solr, Puppet/Salt Stack and others.

### Tobias & Tobias

Senior PHP & Python Developer • Apr. '07 – Aug. '09

Developing PHP and Python web application and ecommerce platforms. Linux server administration and general Dev Ops tasks.

### ServiceWanted

Developing PHP web applications and general Dev Ops scripts/tasks using Python.

## Projects

---

### wikmd

Contributor •

wikmd is an opensource file-based markdown wiki. I have contributed multiple features to the project including a caching layer and the full-text search feature using the Whoosh pure Python search engine library.

### MAP

Architect, Lead Engineer & Lead Developer • '12 – '14

MAP is a marketing platform that allows clients to create marketing campaigns comprising of print, direct mail, email and SMS.

I architected and designed most of the system and wrote the email and SMS send engines, the tracking system and the analytic system using Python/PyPy, Tornado, Pika, Celery, Redis and RabbitMQ.

I was also responsible for building the server architecture and building many of the proof-of-concepts and finalised modules.

### Blackhole

Lead Developer •

Blackhole is a fake MTA (mail transfer agent) that is designed blindly accept, bounce or reject all inbound and outbound email without actually processing or storing any of the data. This makes it safe for handling large volumes of email messages.

Written to work with Python 3.5 utilising the `asyncio` standard library modules and using the `async`, `def` and `await` statements. Originally written for Python 2.7 and earlier version of Python 3 and PyPy using the Tornado library.

Blackhole and it's precursor were written to help prototype the MAP project outlined above.

### Yarg

Lead Developer •

Yarg is a PyPI client, it was written for pypip.in and can search packages as well as read the RSS feeds from PyPI for new packages and new package version releases.

### Pelican

Contributor •

I am one of the developers of the Pelican static blogging platform. I have also authored many plugins and themes for this platform.

### vagrant-bash-completion

Developer •

Originally a fork of another repository, this completion script was written for the developers on a team I worked on at the time. It is now part of Vagrant itself.

---

## Skills

### Leadership

I am capable of taking on leadership roles and guide development strategies. If someone needs to lead development, or troubleshooting an issue I am happy to take charge.

### Mentoring/teaching

I enjoy mentoring and teaching immensely and relish the chance to impart knowledge.

### Computer hardware

I am very interested in computer hardware, especially server hardware and try to be knowledgeable on the subject. I have a small 15U rack in my office filled with rackmount routers, switches, servers, PDUs, and UPS.

### Linux

Debian/Ubuntu, RHEL/CentOS

### Languages

Python, Lua, Shell scripting, & some C, C++, Java & Go.

### Python libraries/frameworks

asyncio, Tornado, Twisted, Flask, Django, Gevent, Eventlet

### Databases

PostgreSQL, MySQL, some Oracle experience. NoSQL/key-value databases like Redis, Couch & Mongo, Solr, Lucene & Elasticsearch.

### HTTP

nginx/OpenResty, Apache, HAProxy, Varnish.

### Virtualisation/containerisation

AWS, OpenStack, Podman, Docker, VMWare, Xen

---

## Outside Interests

- Programming/engineering
- Linux
- Computer/server hardware
- Video encoding
- Video streaming

## Social Links

---

- Github: [@kura](#)
- Website: [kura.gg](#)