Richard Dawe's Curriculum Vitae

Current role:	Principal Engineer at SparkPost
Nationality:	British
Locality:	Bristol, United Kingdom
LinkedIn:	https://www.linkedin.com/in/richdawe
Web site:	https://www.richdawe.be/
Github:	richdawe and rdawemsys
Contact:	0777-6141675 / cv@richdawe.be

Systems software engineer with 20+ years' experience. My goals:

- Build cloud services that delight customers, and make their lives easier.
- Build reliable, operable, supportable systems that scale out to large numbers of customers/users/jobs/etc.
- Help colleagues in my team and across the business be successful.

I'm customer-focussed and results-driven. I'm always looking for ways to help my colleagues succeed, and to improve the quality of the systems that we're building. I have a history of delivering solid, reliable code.

I like to understand systems inside-out. I love digging in, starting at the high level, diving down into the code, and building up a mental model of a system.

My most recent experience has been with building and supporting the SparkPost e-mail cloud service, for mail generation, delivery and analytics. I have been working as part of a team to break up a large, critical monolithic e-mail platform written in C and Lua into multiple node.js microservices, running in the Amazon Web Services (AWS) cloud. I introduced autoscaling of a key component to reduce costs and to manage capacity automatically.

My background is in highly multi-threaded e-mail server software in C and Lua on Linux, running on-premise and in the AWS cloud. I am interested in distributed systems and systems architecture.

Skills

Languages	Proficient:	C, Lua
	Competent:	Perl 5, bash shell scripting
	Advanced beginner:	node.js (server-side JavaScript)
	Needs updating:	C++ (C++03, STL, boost)
Development technologies	Linux; Amazon Web Services (AWS) cloud (inc. ECS, EC2 Spot instances, SQS, CloudFormatiom, CloudWatch, autoscaling); multi-threaded software; client and server software; async event loop with threadpools	
Databases	Cassandra (intermediate); Postgres (intermediate)	
Development practices	Version control (Mercurial, git, subversion); debugging (gdb, valgrind, strace, lsof); design reviews; code reviews (Crucible, GitHub); Continuous Integration and Deployment (CI/CD) (Bamboo, Ansible); unit/functional/integraiton testing	

Networking protocols	HTTP 1.1, SMTP, SMPP, SNMP; SSL and TLS, including X509 certificates; TCP/IP (IPv4, IPv6, UDP, TCP)
Data formats	MIME, RFC 2822 and related standards for e-mail; JSON; XML (DOM, SAX, schemas); YAML; CSV

Employment

SparkPost and Message Systems (remote working, from home), UK (September 2007 - present)

SparkPost provides email delivery services for companies such as the Financial times and Pinterest, via REST APIs and SMTP (Simple Mail Transfer Protocol). Reporting and metrics are provided via webhooks, APIs and an administrative UI. SparkPost is hosted in the Amazon Web Services (AWS) cloud, and is built by Message Systems.

Message Systems's Momentum software is a high-performance messaging server supporting SMTP for mail reception/delivery, supporting extensions in C or Lua, e.g.: for integrating anti-spam technologies.

SparkPost is a trading name of Message Systems, Inc. I've broken my time at SparkPost / Message Systems down into several "eras" below.

March 2015 - present

The **Cloud era** at SparkPost: Principal Engineer

As part of the Platform Engineering team, I designed and implemented new features in our SparkPost and Momentum products. I provided architectural oversight for the messaging platform and integration with other systems and with internal teams. I provided technical leadership and encouragement to the rest of the team. Some highlights:

1. Introduced autoscaling into the highest-volume and most critical SparkPost API, the Transmissions API. I modelled autoscaling, validated the autoscaling parameters in staging, and rolled it out into the pre-existing service in production with no issues. I also worked through issues with AWS EC2 Spot Fleet and our Puppet configuration scripts, so that we could autoscale using a spot fleet for further cost reductions.

2. Vision and design for taking our single-tenant mail platform and making it support multiple tenants concurrently. This was part of a larger project to reduce costs and allow more rapid delivery of new features. I proposed and sold the approach to internal stakeholders; led the development team; did a good chunk of the development work (mostly in Lua). Migrations to the new architecture were done by a separate team.

3. Investigated and greatly reduced the latency of the Transmission REST API. Documented the behaviour of the garbage collector (GC) in our internal fork of Lua 5.1. Implemented data gathering to help understand slowest steps. Changed the GC behaviour to remove a major bottleneck, reducing P95 latency by 50%. (All C code)

4. Designed and prototyped the Relay Webhooks feature, for converting inbound SMTP traffic into HTTP POST calls (mostly in Lua, with C as necessary). This feature enabled customers to use mail purely with HTTP/REST protocols.

5. Helped diagnose and resolve/mitigate various production issues; provided Root Cause Analysis (RCA). Secondary on-call for production issues.

6. Triaged and resolved escalations from our support team; supported colleagues working on escalations / production issues.

7. Trained my colleagues: multiple event loop architecture in C; extending Momentum in C and Lua; Lua best practices; how Momentum delivers e-mail; TLS encryption.

8. Kept people focussed on delivering the minimum functionality required. This was partly to ensure we could iterate rapidly, but also to ensure that we did not over-build software before we understood what our customers wanted or needed. We used Kanban, and earlier Scrum.

June 2013 - March 2015

The **on-premise product development era** at Message Systems:

Lead Engineer (June 2014 - March 2015),
Senior Software Engineer (July 2013 - June 2014)

I joined the Engineering team part-way through the development of Momentum 3.6, a major update to take full advantage of multi-core systems; and at the beginning of Momentum 4.0, a major update to the Momentum product line, adding scalable, fault-tolerant message generation APIs, and new UI, metrics APIs and webhooks. Some highlights:

1. Designed and built the C and Lua APIs to allow REST APIs to be built using idiomatic Lua on top of Momentum's HTTP server. This allowed faster development and iteration of key APIs than would have been possible with C.

2. Designed the click and open tracking system, and led the team building it. (80% C, 20% Lua)

3. Discovered, investigated and resolved throughput issues with the HTTP server when running in multi-threaded mode. (C code)

4. Tracked down and resolved some infrequent and subtle crashes when reloading the configuration under high load, due to unsafe use of data across multiple threads. Some core C APIs and callers needed updating to be thread-safe. (C code)

5. Triaging and resolving escalations from our support team; supporting colleagues working on escalations / production issues.

6. Subject-matter expert on the internals of our software, Lua, and mail standards.

7. Trained my colleagues: lifecycle of messages in Momentum; how to use 3rd-party anti-virus and anti-spam integrations.

September 2009 - July 2013 (~6 years)

The **"Professional Services" era** at Message Systems:

Principal Solutions Engineer (January 2012 - July 2013)
Client Integration Architect (September 2009 - January 2012)
Professional Services Engineer (May 2009 - September 2009)
Senior Software Engineer (September 2007 - May 2009)

As a part of the Professional Services team, I designed and implemented solutions based on Message Systems' Momentum messaging server, primarily for e-mail. This involved extensive customisation, including extending the platform with C, C++ and Lua.

I was the technical sponsor and development lead for several large Professional Services engagements, including two separate multi-year projects. These involved extensive requirements analysis, design and solution implementation work on behalf of the clients.

Many of the projects were replacements for existing mail systems. Challenges included "hidden requirements" where the functionality of existing systems was undocumented or not well understood; providing feature parity or equivalence, including integrating with other customer systems; in some cases, no access was permitted to customer systems due to regulatory requirements.

One of the largest engagements that I was responsible for involved customising Momentum with ~5,000 lines of C++ code, replacing some of the core codepaths of Momentum with custom behaviour, and driving mail policy based on data from an external system accessed via a C++ library provided by the customer. Message Systems was contracted to develop and support this code. I developed about 50% of the code. I also documented how a complex and poorly understood part of the customer's existing system worked, so that we could build the replacement.

Another large engagement was one of the first large-scale uses of Lua for mail policy with Momentum, driven by data from a Postgres database. As part of this project I and another colleague in Professional Services found, debugged and fixed a number of issues with the Lua bindings, which involved tracing problems from the Lua layer down into the core C code. These ranged from simple functional bugs through to subtle memory corruption or timing bugs.

Other activities included working with the Engineering Team on development of the mail platform, discussing client requirements with Product Management, and working with the Support Team.

MessageLabs, Gloucester, UK (Now Symantec Email Security.cloud) (May 2003 - May 2007)

Senior Software Engineer (October 2005 - May 2007)
Software Engineer (May 2003 - September 2005)

MessageLabs was an example of an early cloud service, hosted in its own data centres. E-mail sent to/by a customer passed through MessageLabs's e-mail infrastructure before being delivered. Services such as anti-virus (AV), anti-spam (AS), Image Control (IC) and Content Control (CC) were applied as the message passed through.

1. Responsible for the SMTP mail server software at the core of MessageLabs's e-mail infrastructure. This accepted, scanned, modified, and delivered messages according to customer-configurable policy. Written in a mix of C and C++ (STL and boost) on Linux.

2. Designed & implemented new features in the mail server to support updated versions of AV/AS/IC/CC services, working closely with other teams (Development, Test, Operations, Client Services, Product Management).

3. Led a small team of 2 - 4 people implementing new features and bugfixes in the mail server.

4. Supported the Operations and Client Services teams in resolving customer issues.

5. Experience of the full software lifecycle (SDLC), from requirements analysis to ongoing maintenance.

6. Ensured quality: peer code reviews; unit testing; end-to-end test suites; memory debugging/valdation tools such as valgrind; investigated and evaluated static analysis tools.

7. Technical Lead for a proof-of-concept (PoC) to evaluate a replacement for the aging SMTP mail server software. A replacement was chosen and successfully implemented.

8. Integrated Symantec Brightmail Anti-Spam into the SMTP mail server. This went live with no major issues.

Riverstone Networks (June 2000 - October 2002)

Software Engineer at **Riverstone Networks**, **Reading, UK**: Note: Riverstone was spun out of Cabletron.

1. Wrote microcode (assembler) for two custom processors, including support for Multi-Protocol Label Switching (MPLS) and IPv6 protocols. Worked closely with the hardware, software and simulation teams in California.
2. Initiated and wrote significant portions of tools in C++ for blackbox testing of microcode. These were used to find several hardware bugs during simulation, avoiding re-spins of hardware, saving significant sums of money.
3. Ported the existing IP, TCP, UDP and PPP MIB support to a new SNMPv3 agent.

Cabletron Systems (August 1999 - June 2000)

Software Engineer at **Cabletron Systems**, **Reading, UK**: Worked on Virtual Private Networking (VPN) software in embedded C for small-office routers:

1. Internet security protocol testing: IKE & IPSec.
2. Worked on support for Public Key Infrastructure (PKI) using digital certificates, which included porting a vendor's PKI toolkit to the router software environment. Liaised closely with the PKI toolkit vendor to report bugs, then test and integrate bugfixes into their toolkit.

Summer Jobs

Summer 1996, 1997, 1998
Programmer at **Alchemetrics Ltd.**, **Wokingham, UK**: Wrote database programs in C; wrote documentation for their custom database library; wrote a caching library for their database library to speed data access.

Summer 1995
Programmer at **Murray Computing Ltd., Wokingham, UK**: Wrote database programs which eliminated invalid data and extracted statistics from the data.

Education

1995 - 1999

University of Bristol, UK

MSci Physics with First Class Honours: Four-year course, combining a BSc and Masters degree. Final year included: 20-week research project with a partner on nanofabrication; 45-minute presentation on Free Electron Lasers. For my elective modules, I focussed on Quantum Mechanics, mathematics and the properties of materials.

1988 - 1995

St. Crispins's Comprehensive School, Wokingham, UK

A-Levels (1995): Chemistry (A), Maths (A), Physics (A); AS-Level (1995): German (A); GCSEs (1993): 8 As, 2 Bs

Interests And Hobbies

- Learning German (intermediate level)
- Cycling
- Learning Python
- Member of the Institute of Physics (MInstP)

HTML and CSS crafted by hand. HTML linted with HTML Tidy. PDF generated using Puppeteer.

Sat, 7 Sep 2019 15:18:09 +0100 (cv.html 228659c3ad05e0203d8f5c38428f7bf971d6ef22)