Contact

tong.clive@gmail.com

www.linkedin.com/in/clivetong (LinkedIn)
clivetong.wordpress.com/ (Blog)
www.simple-talk.com/author/clive-tong/ (Other)
www.reflector.net/author/clive/ (Other)

Top Skills

.NET
Lisp
Compilers

Certifications

Linear and Discrete Optimization
Introduction to Mathematical Philosophy
Cloud Computing Concepts: Part 2
Malicious Software and its Underground Economy: Two Sides to Every Story
Heterogeneous Parallel Programming

Clive Tong

Software Architect at Redgate Software
Greater Cambridge Area

Summary

Experienced Technical Lead with a demonstrated history of working in the computer software industry. Skilled in C# and Visual Studio. Strong engineering professional with qualifications in Mathematics and Computer Science from University of Cambridge. I have a strong interest in functional languages and theoretical computer science outside of my work activities.

Experience

Redgate Software
3 years

Software Architect
February 2023 - Present (11 months)

Technical Lead
August 2021 - January 2023 (1 year 6 months)
Cambridge, England, United Kingdom

I am Technical Lead for a database cloning solution based on containers running in a Kubernetes cluster deployed to a customer site using Replicated. I have gained experience of Kubernetes, Go and Linux, plus some experience of AKS for running a test system. The team also does support for the existing Windows only cloning solution.

Lead Software Engineer
January 2021 - August 2021 (8 months)
Cambridge, England, United Kingdom

I currently work on the Sql Clone team at Redgate. We are working on a Kubernetes based database cloning solution, as well as continuing work on the existing Sql Clone product.

Facebook
Software Engineer
September 2020 - December 2020 (4 months)
London, England, United Kingdom

After Bootcamp, I worked in the Ads Experimentation Infra team

Redgate Software
Technical Lead
April 2019 - September 2020 (1 year 6 months)
Cambridge

In my first six months back at Redgate, I worked as a Tech Lead of the Sql Data Catalog product.
During this period, we productized the code which had been developed over an EAP program that had run for more than a year. We then released the first commercial version.

In January 2020 I returned to the Sql Monitor Product as a Tech Lead of one of the three teams working on the code base. My four person team is looking at ways to make the software more scalable to handle more servers.

In my last two months at Redgate, I moved to Tech Lead on one of the other monitor teams that was adding support for Amazon RDS

Speechmatics
Software Engineer
February 2019 - April 2019 (3 months)
Cambridge, United Kingdom

I worked as a Go and Python programmer working on code that used the C++ machine learning system which implemented the speech recognition. I spend my time working in Linux and gained experience with Docker and Kubernetes as well as the two languages mentioned above.

Redgate Software
11 years 1 month

Technical Lead
June 2016 - January 2019 (2 years 8 months)
Cambridge, United Kingdom

I work as the Technical Lead for the SQL Monitor team.

SQL Monitor monitors an estate of machines and associated SQL Server instances, providing information on the status of the various entities, and alerting when certain conditions are met. I code but also do more management type things such as line management for the team members, organise the planning of the sprints and two weekly releases and try to balance commercial

requirements such as feature development against the maintenance and quality activities of the team.

I still help with the weekly functional programming lunch and also help organise the bi-weekly lightning talks.

Software Developer
January 2008 - June 2016 (8 years 6 months)

My first project was Exchange Server Archiver, an email archiver. I was responsible for the code which scheduled the archiving tasks, the OWA integration for 2007 and 2003, and a web service which allowed access to archived emails.
I then moved to the .NET team where I worked on the .NET profilers (performance and memory). Over the years, this work has included making changes to the C++ core [which is a COM component] as well as the C# code which analyses the results and displays them. Features I added required work on IL instrumentation and hooking Windows heap allocation to allow us to track unmanaged memory allocation.
Red Gate acquired the .NET Reflector IL decompiler, and I worked on this as part of a small team. I did large parts of the work for generating PDB files to accompany the decompiled source and Visual Studio integration to allow users to debug third party decompiled code.
I spent a year away from the .NET team, working on tooling for Azure and supporting and then rewriting a Cloud application that backs up clients' Azure data. As part of this work, we rewrote the application as a number of MicroServices.
I then went back to the .NET team, and for seven months worked as part of a three person .NET team working on the company's .NET products [profilers/ Reflector/SmartAssembly], with an intern over the summer period. As part of this, I did some team management.
In March 2016, I joined a team working on a cloud application for uploading data from an on-premise database to an Azure Dataware House - http://www.dataplatformstudio.com/. The application installed a desktop application on the client which talks to a cloud based service that orchestrates the upload.

I help organise the bi-weekly lightning talks, have given numerous presentations (on C#, F#, Async, memory profiling), and help with the weekly Clojure/Haskell lunch.

Scientia Ltd

Software Developer
January 1999 - January 2008 (9 years 1 month)

The company produced scheduling software which was mainly written in Lisp. The aim was to make the data from the Lisp world available to external tools and applications. I wrote a framework for exposing the applications object model (defined via an XML DSL) as a set of COM objects. This involved low level coding to make up the v-tables, code generation of an interop layer, and at a testing level, auto-generating C++ and VB code to exercise the object model.
We then exposed this object model via OLE-DB, which involved defining a table structure over the object model, and writing an SQL interpreter to interpret the user queries. Again, the translator auto-generated sets of SQL queries by anti-parsing which I ran against the application and an ACCESS database, checking that the results were the same.
We starting using C# for the GUI, wrapping the COM exposed application, at one point deploying both parts using ClickOnce.
I also worked on the database persistence of the object model, wrote a web application which allowed users to schedule individual activities, and also did general coding on the application as well as helping introduce practices like unit testing.

Harlequin Ltd
Software Developer
January 1992 - August 1999 (7 years 8 months)

Worked on Common Lisp and ML runtimes and development environments, including compilers and garbage collectors. The ML compiler from the early 1990s has recently been open sourced https://github.com/Ravenbrook/mlworks. I had the chance to port the Lisp system to the Alpha architecture, and the compiler and GC to run on x86. Lots of experience debugging code generation at the assembly level on many architectures (x86,Sparc,Mips,Alpha).
Developed a CORBA ORB for the Lisp system (and helped with the design of ORBs implemented in DYLAN and ML).
Reponsibility for the cross platform GUI framework (CAPI), where I learned loads about WIN32 and X Windows.

Harlequin Ltd
Software Developer
September 1989 - December 1990 (1 year 4 months)
Cambridge, UK

Worked on additions to the Common Lisp system, including the floating point optimisation of FFT.
Worked on an ML compiler to TEN15 - wrote the code generator and lexer.
Helped with the CHAMELION project, dynamic code migration via the HARP abstract machine

ICL Fujitsu
Programmer
October 1984 - July 1985 (10 months)

Industrial year as part of sponsorship. Worked on team doing FORTRAN on a mini-DAP, a dual array processor machine. My main project was benchmarking various operations running on a simulator [using a mix of csh and FORTRAN]. I spent loads of time in the technical library, taught myself C and wrote a compiler for S-ALGOL and Lisp, and started playing with LispKit Lisp.

Subsequent Xmas and Summer vacations at ICL West Gorton working on the Flagship project, a machine that was programmed in the functional programming language HOPE. I learned the low level assembly language of the machine [DACTL] and spent time implementing test programs at this level

Education

University of Cambridge
Phd (unfinished), Computer Science · (1991 - 1991)

University of Cambridge
Diploma in Computer Science, Distinction · (1988 - 1989)

University of Cambridge
BA Mathematics, 1st · (1985 - 1988)