Tuan Phong Ngo

Stockholm, Stockholm County, Sweden

tuanphong.ngo@gmail.com

linkedin.com/in/tuan-phong-ngo-59959b12

Summary

I am a compiler engineer enthusiastic about code optimizations in all aspects from high levels to target-dependent levels. (i) I have detailed knowledge in developing high-performance applications from both perspectives of a software engineer and a compiler engineer. (ii) I also have a strong background and research experience in concurrent programs, relaxed memory models (e.g., Intel-x86 TSO, IBM Power, ARM), and software model checking.

Experience

Compiler Engineer

IAR Systems

Sep 2020 - Present (7 months +)

I have been doing research and developing optimization techniques for code generation at both high level and target-dependent level in IAR's C/C++ compilers. I have been working closely with generating assembly code for different embedded hardware such as ARM, RISCV, AVR, and Renesas. I gain detailed knowledge in developing high-performance applications from both perspectives of a software engineer and a compiler engineer.

The job uses C/C++, Visual Studio, Subversion (SVN)/ Git. We also use Agile and JIRA for software development.

Keywords: Compiler, Code generation, Assembly code, Optimization, Performance, Code size, Embedded system, C/C++, Agile software development, JIRA

C/C++ Developer - Developing an efficient architecture to render geometries

COMSOL, Inc.

Oct 2019 - Sep 2020 (1 year)

COMSOL Multiphysics® is a simulation software for modeling designs used worldwide in research and computer-aided engineering. Mesh, a discretization form of geometry, is used in all COMSOL's products. The goal of mesh generation is to enable efficient finite-element (FEM) computations based on 3D CAD models and 3D scans.

I have designed a new architecture to effectively generate mesh for rendering geometries inside COMSOL Multiphysics®. The new architecture optimizes computation time by 30% on average for large models that contain virtual geometric operations. The core idea of the architecture is using caches of rendering objects at different detail levels for each virtual operation. When a user creates a geometric object by performing a long sequence of operations, a rendering object will be fetched from an appropriate cache for each operation. Then, the rendering object is updated incrementally. Finally, the rendering object is kept for future operations. Because the architecture keeps all necessary rendering information, it does not waste time to redo the same jobs. Therefore, it can obtain a high-performance rendering.

It is important to note that I am the main developer implementing almost all mesh generation functions inside the new architecture.

The job used C/C++/Java, Visual Studio, and Subversion (SVN).

Keywords: Mesh generation, Graphical rendering, High-performance architecture, Cache system

References: Björn Bretz (Bjorn.Bretz@comsol.com) - Team Leader of Mesh Group, COMSOL AB

Software Verification Engineer

IAR Systems

May 2019 - Oct 2019 (6 months)

Visual State is a product for software design and code generation. I have made several performance improvements for IAR Visual State's verification engine. Especially, I have developed a new high-performance parallel (multi-core) version of the verification engine.

The job used C/C++, Visual Studio, and Subversion (SVN).

Keywords: State machine, Software verification, Symbolic model checking, (parallel) BDD

Researcher

Uppsala University

Mar 2019 - May 2019 (3 months)

I collaborated with other researchers at the IT department, Uppsala University, to develop a high-performance stateless model checking algorithm for checking the correctness of concurrent programs running under the Sequential consistency semantics. We published a paper "Optimal stateless model checking for reads-from equivalence under sequential consistency" at OOPSLA 2019 - a top conference in Object-oriented programming.

Ph.D Student in Computer science

Uppsala University

Nov 2013 - Jan 2019 (5 years 3 months)

Weak memory model, implemented in modern hardware (e.g., Intel x86-TSO, IBM POWER, or ARM) or modern programming languages (e.g., C/C++11 or Java), obtains high performance by executing instructions in a program out-of-order. This work used formal methods to check the correctness of concurrent programs running under weak memory models.

I was the main developer of several efficient tools using C/C++/Python. Publications are in top conferences of software engineering and software verification such as OOPSLA, TACAS, and ESOP.

References: Prof. Parosh Aziz Abdulla (parosh@it.uu.se), Dr. Mohamed Faouzi Atig (mohamed_faouzi.atig@it.uu.se)

Teaching Assistant in Computer Science

Uppsala University

Nov 2013 - Jan 2019 (5 years 3 months)
I have taken care of lab sessions and home assignments in several courses related to Programming and Program verification such as Programming theory (using Dafny from Microsoft), Data structures and algorithms, Automata and logic in IT system modeling (using SPIN from Bell Lab and CBMC from the University of Oxford).

Lecturer In Computer Science
Hanoi University of Science and Technology
Sep 2009 - Aug 2011 (2 years)
I gave lectures in several undergraduate courses such as C Programming and Database Design.

Education

Uppsala University
Doctor of Philosophy (Ph.D), Computer Science
2013 - 2019
Weak memory model, implemented in modern hardware (e.g., Intel x86-TSO, IBM POWER, or ARM) or modern programming languages (e.g., C/C++11 or Java), obtains high performance by executing instructions in a program out-of-order. This thesis applied formal methods to check concurrent programs running under weak memory models. An important part of the thesis was to develop efficient tools in C/C++/Python. Publications are in top conferences of software engineering and software verification such as OOPSLA, TACAS, and ESOP.

References: Prof. Parosh Aziz Abdulla (parosh@it.uu.se), Dr. Mohamed Faouzi Atig (mohamed_faouzi.atig@it.uu.se)

Uppsala University
Master's degree, Computer Science
2011 - 2013
Selected coursework:
* Mathematical logic,
* Artificial intelligence,
* Advanced algorithms,
* Advanced computer architecture,
* Programming theory,
* Constraint programming,
* Optimisation,
* Functional programming,
* Compiler design,
* Computer networks,
* Programming in embedded systems

Hanoi University of Science and Technology
Bachelor's degree, Computer Science
2004 - 2009

I have studied my bachelor in Computer science in a talented engineering class. The class had top-most students in the university.

Skills

Java • C • C++ • Compiler Optimization • Code Generation • Embedded Systems • Software Verification

Honors & Awards

Best paper award nominee - The European Joint Conferences on Theory and Practice of Software (ETAPS) 2017
Apr 2017
Our paper "Context-Bounded Analysis for POWER" was one nominated as one of the best papers at TACAS 2017 (see https://etaps.org/2017/etaps-programme/etaps-2017-preprints-thursday)

PhD research grant - UPMARC research centre
Nov 2013
Receiving a 5-year Ph.D. research grant at UPMARC research center in 2013.

The Uppsala Programming for Multicore Architectures Research Center (UPMARC) is a long-term interdisciplinary research program in parallel computer systems. The mission is to identify and address the fundamental challenges that will enable all programmers to leverage the potential performance from the ongoing shift to universal parallel computing. This transition pushes to the forefront the challenges of enabling new groups of developers, algorithms, and legacy systems to exploit the performance benefits of ubiquitous parallel platforms.

Master scholarship - European Union’s Erasmus Mundus scholarship
Jun 2009

Student grant for Summer of code - Google
May 2012

Third prize in Artificial Intelligence programming competition - Uppsala University
Nov 2019

First prize in Functional programming competition - Uppsala University
Oct 2012

Scholarship for excellent students - Hanoi University of Science and Technology
Semester scholarships each year from 2004 to 2009 .

Best paper award nominee - 27th International Conference on Concurrency Theory (CONCUR 2016)
Aug 2016

Our paper "The Benefits of Duality in Verifying Concurrent Programs under TSO" is invited to submit to a special issue of Logical Methods in Computer Science (LMCS): https://lmcs.episciences.org/4228/