

Software-Design für die AUTOSAR Adaptive Platform

Neuartige Optimierungstools für parallele POSIX-Software

Maximilian Odendahl, CEO, Silexica GmbH

Die Entwicklung von Software für die neue AUTOSAR Adaptive Platform wird Entwickler vor neuartige Herausforderungen stellen. Im Vergleich zu bestehenden Singlecore-Systemen, aber auch im Vergleich zur existierenden AUTOSAR Classic Platform, wird die Verteilung der Software auf die verschiedenen Prozessorkerne der Ziellplattform eine immer grösser werdende Herausforderung. Dabei gilt es nicht nur, zeitliche und funktionale Vorgaben zu berücksichtigen, sondern idealerweise auch den Leistungs- und Energieverbrauch zu minimieren. Selbst für einfachere Algorithmen und Zielsysteme ergibt sich daher eine Vielzahl von möglichen Lösungen, deren Entwicklung und Bewertung bisherige Entwicklungsansätze an ihre Grenzen bringen. Abhilfe können neuartige Algorithmen und Tools aus dem Bereich der Software Design Automation schaffen.

Multicore-Programmierung und ihre Herausforderungen

Die Entwicklung von Software für neuartige Multi- und Manycore-Systeme stellt zahlreiche Industrien vor neue Herausforderungen. Bedingt wird dies durch die steigende Komplexität der zur Verfügung stehenden Hardwaresysteme, wobei gleichzeitig neben hoher Performance für eine Vielzahl von Anwendungsszenarien oft auch eine minimale Leistungsaufnahme gefordert wird. Verschärft wird dieses Problem zusätzlich durch immer kürzer werdende Entwicklungszyklen und gestiegene Sicherheitsanforderungen. Die daraus entstandene „Effizienzlücke“ zwischen den herausragenden Fähigkeiten moderner Hardware und der Fähigkeit, diese effizient durch Software auszunutzen, ist mit der zurzeit üblichen manuellen Systemauslegung und Programmierung nicht wirtschaftlich zu schließen.

Die Herausforderungen sind hierbei neben der Parallelisierung von bestehendem C/C++ Code vor allem eine optimale Verteilung bereits paralleler Software unter Berücksichtigung von zeitlichen und funktionalen Einschränkungen. Hierbei muss nicht nur die benötigte und verfügbare Rechenleistung berücksichtigt werden, sondern in wachsenden Maß auch die Kommunikation zwischen den Prozessorkernen, die zunehmend zum Flaschenhals für die Systemleistung wird. Um alle zeitlichen und funktionalen Einschränkungen zu erfüllen, ist daher eine gemeinsame Berücksichtigung von Rechenleistung und Kommunikation notwendig.

AUTOSAR Adaptive Platform

Um neuartige Anwendungsfälle bedienen zu können, werden vom Auto der Zukunft immer mehr rechenintensive Aufgaben übernommen, unter anderem

- Automatisiertes Fahren
- Vernetzung des Fahrzeugs mit dem umgebenden Straßenverkehr ("V2X")
- Anwendungen mit permanenter "over-the-air"-Aktualisierung

Aus diesem Grund wurde im AUTOSAR-Konsortium gemeinsam die Adaptive Platform spezifiziert, um Software flexibler für neuartige Zentralcomputern entwickeln zu können. Aufgrund der Einführung der AUTOSAR Adaptive Platform kommen aber grundlegende

Änderungen auf die Entwickler zu. Eine kurze technische Übersicht zwischen den beiden Plattformen wird in der folgenden Tabelle gegeben:

Classic Platform	Adaptive Platform
Basierend auf OSEK	Basierend auf POSIX ()
Statische Task-Konfiguration	Unterstützung von dynamischem Scheduling
Gleicher Adressraum für alle Anwendungen	Jede Anwendung besitzt ihren eigenen (virtuellen) Adressraum
Optimiert auf eventbasierte Kommunikation (z.B. CAN, FlexRay)	Serviceorientierte Kommunikation
Ausführung der Anwendung aus dem ROM	Application wird aus dem RAM geladen

POSIX (Portable Operating System Interface) ist eine standardisierte Programmierschnittstelle zwischen der Applikation und dem Betriebssystem und wurde ursprünglich nicht für die Automobilbranche entwickelt. Jedoch machen POSIX-Betriebssysteme die Softwareentwicklung für das Fahrzeug deutlich flexibler und erlauben die Nutzung von zahlreichen bestehenden Bibliotheken aus den Bereichen High Performance Computing, Embedded Vision und Machine Learning.

Ein wesentlicher Unterschied wird die Unterstützung von dynamischem Scheduling sein. Hierbei sind viele unterschiedliche Strategien abhängig vom eingesetzten Betriebssystem denkbar, z.B. Round-Robin oder prioritätsbasiertes Scheduling. Dynamisches Scheduling kann theoretisch die Performance von Anwendungen steigern, die Vorhersehbarkeit des zeitlichen Verhaltens wird dadurch aber deutlich schwieriger. Ein Entwickler kann sehr selten alle potentiellen Thread-Interaktionen in komplexen, parallelen System vorhersagen. Außerdem ist es oftmals schwierig, den wahren Schuldigen bei Systemeffekten und Performance-Engpässen zu identifizieren. Adaptive AUTOSAR erlaubt auch die dynamische Erzeugung von weiteren Threads zur Laufzeit, was die Vorhersehbarkeit weiter beeinträchtigt.

Lösungsansätze durch neuartige Software-Designtools

Für die oben genannten Problemstellungen ist eine manuelle Auslegung und Programmierung unter wirtschaftlichen Aspekten bereits heute nicht mehr möglich. Manuelle Ansätze, wie z.B. die mehrmalige Ausführung einer Anwendung und anschließender Vergleich von detaillierten Traces, wird in der Zukunft aufgrund der steigenden Komplexität nicht mehr möglich sein. Bereits heute sehen wir sowohl erhöhte Kosten für Verifikation und Tests als auch vermehrte Projektverzögerungen.

Eine neue Generation von automatisierten Entwicklertools versucht diese Probleme durch Compiler-basierte statische und dynamische Datenflussanalyse, automatische Laufzeitabschätzung und Berechnung von Affinity-Masken zu lösen. Ziel dabei ist eine automatische Optimierung der Software und eine automatische Verteilung auf homogene und heterogene Zielarchitekturen. Während des gesamten Entwicklungsprozesses wird dabei die Zielplattform direkt mit einbezogen, was eine kombinierte Software- und Hardware-Optimierung ermöglicht.

Für die sequentielle C/C++ Programmierung sowie für die AUTOSAR Classic Platform ist eine solche Toolchain bereits von der Firma Silexica verfügbar. Für die Adaptive Platform ist

diese derzeit in der Entwicklung und besteht, wie in Abbildung 1 angedeutet, aus mehreren Elementen.

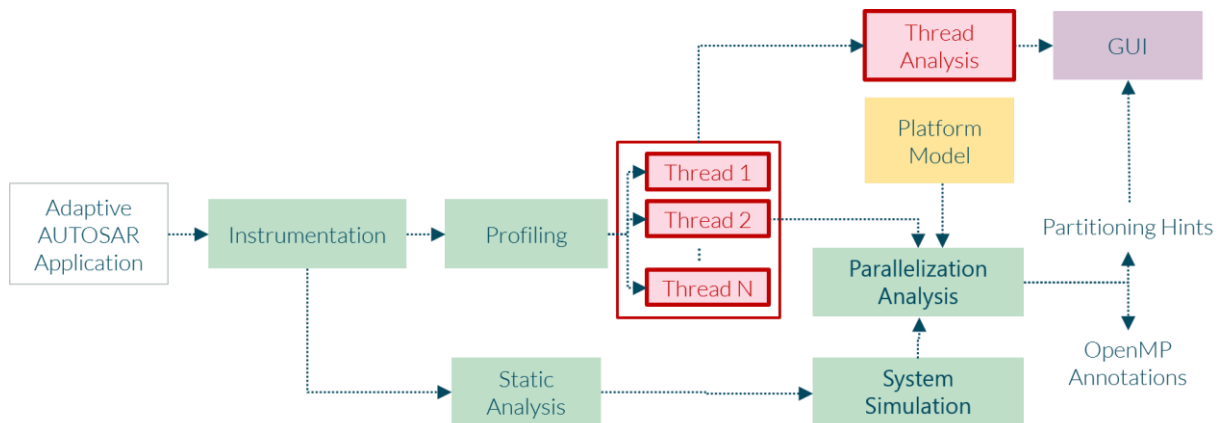


Abbildung 1: Automatische Analyse von Adaptive AUTOSAR Anwendungen

Die Thread-Analyse ist dabei der schwierigste Teil:

- Der Programmfluss muss abstrahiert und durch Ereignisse ersetzt werden, die ihre Source-Code-Eigenschaften beinhalten und semantische Eigenschaften repräsentieren, welche sowohl den Ausführungsfluss als auch den globalen Zustand der Anwendung beeinflussen.
- Schnelle Algorithmen müssen genutzt werden, um die Abhängigkeiten bzgl. Reihenfolge und Datenkonsistenz dieser Events berechnen und extrahieren zu können.

Mögliche Events für die abstrakte Darstellung einer Anwendung sind hierbei unter anderem das Thread Management des Betriebssystems, Process/library loading (z.B. via `vma_link`), signals (z.B. via `SIGSEGV`), Pthreads calls für Mutexes (z.B. `pthread_mutex_lock`) und Barriers und gemeinsame Zugriffe auf den gleichen Speicherbereich. Die semantische Analyse benötigt dann unterschiedliche Heuristiken, um die unterschiedlichen Effekte zu verstehen. So müssen z.B. zwei Abhängigkeiten zwischen zwei Mutex-Ereignissen völlig anders bewertet werden als die Abhängigkeiten zwischen zwei Barrieren-Ereignissen.

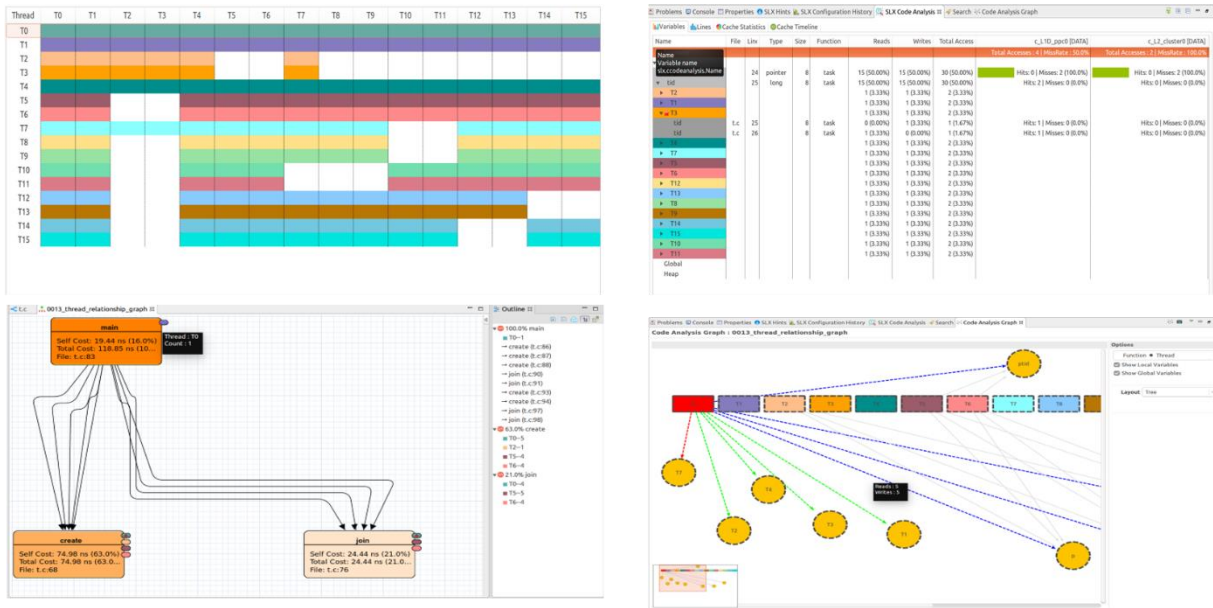


Abbildung 2: Unterschiedliche Visualisierungen der Codeanalyse von POSIX-basierten Anwendungen i) Liveness Ansicht von Threads ii) Call-Graph iii) Zugriffshäufigkeit auf Variablen iii) Code-Visualisierung von Threads, Funktionen und Variablen

Full-System Simulation

Die auf der AUTOSAR Adaptive Plattform basierenden Anwendungen haben den Vorteil, dass sie zur Laufzeit in das System eingebunden werden, wodurch sie unabhängig voneinander entwickelt, getestet und verteilt beziehungsweise aktualisiert werden können. Die serviceorientierte Kommunikation ermöglicht es zudem, Anwendungen zu jedem beliebigen Zeitpunkt in das Gesamtsystem zu integrieren. Sie ist kompatibel mit SOME/IP der AUTOSAR Classic Plattform. So werden im zentralen Supercomputer im Auto der Zukunft mehrere Systeme parallel laufen, angedeutet in Abbildung 3.

Dies erhöht nochmals die zuvor beschriebenen, wachsenden Schwierigkeiten des zeitlichen Verständnisses des Gesamtsystems. Neuartige Tools werden auch hierfür in naher Zukunft notwendig sein um der steigenden Komplexität im Auto der Zukunft Herr zu werden.

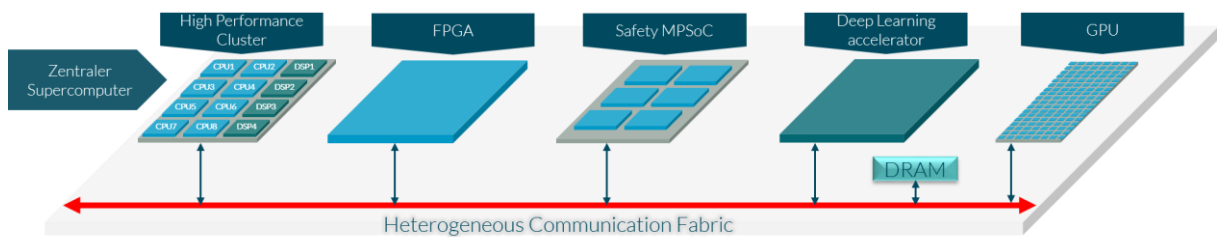


Abbildung 3: Zentraler Supercomputer im Auto der Zukunft

Zusammenfassung

Die zunehmende Komplexität von modernen Multiprozessorsystemen in Kombination mit verkürzten Entwicklungszyklen und steigenden Anwendungsanforderungen haben zu einer Effizienzlücke geführt: Die effiziente Nutzung moderner Multiprozessorsysteme kann mit manueller Systemauslegung und Programmierung nur beschränkt wirtschaftlich bewerkstelligt werden. Mit steigender Plattform- und Software-Komplexität wird dieses Problem zukünftig sogar noch größer.

Aufgrund der Einführung der AUTOSAR Adaptive Platform kommen weitere Herausforderungen auf Softwareentwickler hinzu. Neben dem Wechsel von OSEK zu POSIX ist hier vor allem die Möglichkeit des dynamischen Scheduling hervorzuheben. Diese neu gewonnene Flexibilität ist hierbei Segen und Fluch zugleich, da die Vorhersehbarkeit des zeitlichen Verhaltens des Systems stark eingeschränkt ist.

Neuartige Software Design Automation Tools können genutzt werden, um datengetriebene Entscheidungen zu treffen, sein Gesamtsystem besser verstehen zu können und die Lücke zwischen Software- und Hardwareinnovationen der letzten Jahre zu schließen.

Autor

Maximilian Odendahl studierte Computer Engineering an der RWTH Aachen und war nach seinem Abschluss am Institut for Communication Technologies and Embedded Systems (ICE) als Wissenschaftlicher Mitarbeiter mit dem Schwerpunkt der Toolentwicklung für heterogenen Multicore-Systeme tätig. Von Januar 2013 bis Dezember 2014 war er zusätzlich Oberingenieur des ICE. Heute ist er geschäftsführender Gesellschafter der Silexica GmbH und spricht regelmäßig auf Konferenzen.



Kontakt

Internet: <http://www.silexica.com>

Email: odendahl@silexica.com