DER ZUKUNFT VORAUS

DER ZUKUNFT VORAUS

IM SPANNUNGSFELD VON HARD- UND SOFTWARE

HOLGER FRÖNING

Die Grenze zwischen Soft- und Hardware bestimmt auch die Grenze zwischen der Leistungssteigerung von Computern und deren Programmierbarkeit. Diesem Spannungsfeld, das derzeit eine der grundlegendsten Fragestellungen der Informatik darstellt, widmen sich Forscherinnen und Forscher am Institut für Technische Informatik der Universität Heidelberg.

Will man Computer leistungsfähiger machen, muss man sie spezialisieren. Dazu wird dem Prozessor, der zentralen Recheneinheit des Computers, eine spezielle Hardware – beispielsweise für eine Matrizenmultiplikation – hinzugefügt, die häufig wiederkehrende Aufgaben übernimmt. Man kann für den Prozessor auch eine komplett neue Architektur entwickeln, was eine noch umfassendere Spezialisierung und somit eine noch größere Rechenleistung verspricht. Betrachtet man die Spezialisierung als Aspekt der Hardware und die Programmierbarkeit als Aspekt der Software, dann wird der Verlauf der Grenze zwischen Soft- und Hardware interessant.

Im Falle der Spezialisierung kann man sich diese Grenze bis weit hin zur Software verschoben vorstellen: Eine komplette Spezialisierung wäre unglaublich schnell und effizient – sie ist aber nicht in Einklang zu bringen mit der Programmierbarkeit und mit Innovationen in Algorithmen und Anwendungen. Daraus resultieren weniger Flexibilität und weniger Anwendungsmöglichkeiten. Weil „mehr Spezialisierung" stets mit „weniger Programmierbarkeit" einhergeht, wird aktuell nach Prozessorarchitekturen gesucht, die sowohl den steigenden Anforderungen an Anwendungen als auch der möglichst umfassenden Programmierbarkeit gerecht werden können. Dieses Spannungsfeld zwischen Leistungssteigerung und Programmierbarkeit – und somit die Grenze zwischen Soft- und Hardware – ist eine der derzeit grundlegenden Fragen der Informatik.

Die Komplexität von Milliarden Transistoren

Die meisten Computer basieren auf der „Turingmaschine", so benannt nach dem britischen Mathematiker Alan Turing, der sie in den 1930er-Jahren eingeführt hat. Es handelt sich dabei nicht um eine Maschine im herkömmlichen Sinn, sondern um ein mathematisches Modell der theoretischen Informatik für Aufgaben der Datenverarbeitung wie bei Servern, Workstations und Laptops, bei autonomen Systemen wie Drohnen, selbstfahrenden Autos oder anderen Robotern oder für Computerspiele und sogenannte Wearables-Computersysteme, die während der Anwendung am Körper getragen werden wie beispielsweise Datenbrillen oder Smartwatches. Bei diesem Modell, das Betrachtungen zur Berechenbarkeit ermöglicht, existieren keine Grenzen zwischen Hard- und Software; diese sind lediglich abhängig von der gewählten Implementierung. Heute beruhen nahezu alle Implementierungen auf der digitalen CMOS-Technologie als Bauform für Transistoren. In diesem Kontext ist das „Moore'sche Gesetz" wohlbekannt: Es besagt, dass sich die Anzahl der (CMOS-)Transistoren alle zwei Jahre verdoppelt, was in der Regel zu einer substanziellen Leistungssteigerung der Computer führt, weil immer mehr Transistoren immer komplexere Prozessorarchitekturen ermöglichen. Aktuell haben die kleinsten Transistoren eine Größe von fünf Nanometern, und Prozessoren können in integrierten Schaltkreisen auf einer Fläche von circa 400 mm² bis zu 50 Milliarden Transistoren integrieren.

Es ist illusorisch anzunehmen, dass Anwender diese Komplexität umfassend verstehen können – daran scheitern selbst Experten. Helfen kann einer der wichtigsten Grundpfeiler der Informatik, die Abstraktion. Sie hat zum Ziel, Komplexität zu verstecken und dennoch ein ausreichendes

Institut für Technische Informatik (ZITI)

Die Forschung am Institut für Technische Informatik (ZITI) beschäftigt sich im Bereich „Innovative Computing" mit der Funktionsweise, dem Aufbau und der Nutzung von Hardware- und Softwarearchitekturen zur Erfassung und Verarbeitung von Daten, um Algorithmen an die Stärken und Schwächen der verwendeten Architektur anzupassen. Das Institut ist mit vielen Forschungsbereichen der Universität Heidelberg vernetzt, in denen Daten erfasst oder rechenintensive Aufgaben bewältigt werden. Übergeordnetes Ziel ist in der Regel eine substanzielle Steigerung der Rechenleistung sowie der Energieeffizienz oder eine Reduktion der Leistungsaufnahme. Für die Lösung von Problemstellungen im Bereich von Sensorik oder Datenkommunikation kann das ZITI auch Mikrochips von Grund auf neu entwerfen.

Neben dieser Spezialisierung befasst sich das Institut mit effizienter Programmierung, verfügt über eine Vielzahl von verschiedenen Rechenbeschleunigern und untersucht so Methoden und Arbeitsflüsse für hocheffiziente Datenverarbeitung. Weiterhin werden im Bereich Biorobotik und Biomechanik disziplinenübergreifend neue Technologien entwickelt, die Biologie und Robotik symbiotisch miteinander verbinden, um eine effizientere Kommunikation zwischen Maschinen zu ermöglichen, die mit biologischen Systemen interagieren. Ziel sind robotische Geräte, die beispielsweise in der neurologischen Rehabilitation oder zur Unterstützung des Menschen in der Arbeitswelt eingesetzt werden können.

www.ziti.uni-heidelberg.de

Verständnis der zugrunde liegenden Funktionsweise, Architektur oder Organisation zu ermöglichen. Somit existiert für eine Prozessorarchitektur immer eine zugehörige Abstraktion, fachsprachlich „Instruction Set Architecture" – kurz ISA – genannt. Sie ist eine Art Vertrag zwischen Programmierer und Prozessor und definiert die Grenze zwischen Soft- und Hardware. In der Abstraktion kann ein Prozessor bestimmte Anweisungen – die Instruktionen – schnell und parallel in Hardware ausführen. Die Flexibilität im Hinblick auf die Möglichkeiten der Anweisung – die Parameter der Instruktion – hingegen sind eingeschränkt. Das bedeutet: Innovationen in den Algorithmen müssen entsprechend den gegebenen Instruktionen formuliert werden.

Die „Matrizenmultiplikation" ist ein klassisches Beispiel. Dabei werden die Elemente der Zeilen der ersten Matrize mit Elementen der Spalten der zweiten Matrize paarweise multipliziert und addiert. Bei zweidimensionalen Matrizen ergeben sich drei äußere Schleifen: zwei Schleifen für die Dimensionen und eine Schleife für die Iteration über der Zeile beziehungsweise Spalte. In der innersten Schleife werden Elemente aus dem Speicher in den Prozessor geladen, Elemente paarweise multipliziert und akkumuliert. Für all diese Schritte gibt es entsprechende Instruktionen, ebenso für die Umsetzung der Schleifen bezüglich Indexarithmetik und Kontrollfluss.

Viele Möglichkeiten der Spezialisierung

Einen Algorithmus über solche Instruktionen anzupassen ist sehr einfach. Insgesamt ist dieser konventionelle Ansatz jedoch eher ineffizient, weil für eine Ausführung auf einer komplexen Prozessorarchitektur viele Instruktionen geladen, decodiert und interpretiert sowie der Daten- und Kontrollfluss berücksichtigt werden müssen. Man kann sich für eine Matrizenmultiplikation aber auch eine einzelne, spezialisierte Instruktion vorstellen – zum Beispiel eine

„Wenn sowohl Hardware als auch Software veränderbar sind, wirft dies schnell die Frage auf, wo die Grenzlinie zwischen den beiden Komponenten am besten zu ziehen ist."

Instruktion, die in Bezug auf die Matrixgröße parametrisierbar ist, ansonsten aber alle vorherigen Schritte integriert. Kombiniert mit einer entsprechenden Hardwareeinheit wird eine Ausführung dann um Größenordnungen effizienter – die Flexibilität der Berechnung aber geht verloren.

Es gibt noch viele weitere Möglichkeiten der Spezialisierung. Ein Beispiel sind Grafikprozessoren für nichtgrafische Anwendungen: Sie sind Standardprozessoren sehr ähnlich – beide basieren auf parallelen Vektorarchitekturen –, unterscheiden sich jedoch durch ein grundsätzlich anderes Ausführungsmodell und durch die Dimensionierung einzelner Komponenten. Für Grafikprozessoren ergibt sich dadurch ein deutlicher Vorteil im Hinblick auf viele Anwendungen, insbesondere in den Bereichen maschinelles Lernen und wissenschaftlich-technisches Rechnen. Aus Systemsicht führen die hochgradig unterschiedlichen Ausführungsmodelle jedoch zu Ineffizienzen, vor allem dann, wenn mehrere Komponenten interagieren. Unsere Arbeitsgruppe im Institut für Technische Informatik der Universität Heidelberg hat in den vergangenen Jahren mehrere neuartige Kommunikationsmodelle für verteilte Grafikprozessoren vorgeschlagen, die auf einer Vielzahl von Grafikprozessoren basieren und auch die Aspekte der Programmierbarkeit von Systemen adressieren.

Hardwaresprachen

Grundsätzlich ist auch eine Hardwarekomponente wie ein Prozessor veränderbar. Hierfür existieren Sprachen, die Hardware beschreiben, wobei es sich hauptsächlich um Programmiersprachen für die sogenannte transistorbasierte Logik handelt. In der Praxis ist der Entwurf von Hardwarekomponenten jedoch eingeschränkt, weil jede Veränderung eine neue Produktion mit moderner CMOS-Technologie erfordert, die zu Einmalkosten im Bereich von rund 100 Millionen Euro führen kann. Andere Möglichkeiten bietet die rekonfigurierbare Logik. Sie basiert auf sogenannten FPGAs (integrierte Schaltkreise der Digitaltechnik) und kleinen eingebetteten Systemen, die geringere Einmalkosten haben und aufgrund ihrer geringen Leistungsaufnahme gerade im mobilen Bereich häufig anzutreffen sind.

Wenn sowohl Hardware als auch Software veränderbar sind, wirft dies schnell die Frage danach auf, wo die Grenzlinie zwischen den beiden Komponenten am besten zu ziehen ist. Das genau ist die Frage, mit der sich das „Hardware/Software-Codesign" befasst: Es sucht eine möglichst gute Gesamtarchitektur für ausgewählte Anwendungen. Die Entwurfsprozesse von Soft- und Hardware unterscheiden sich jedoch grundlegend: Ein Softwareentwurf orientiert sich primär an der Zeit und ist hochgradig flexibel; die wesentliche Zielmetrik eines Hardwareentwurfs indes ist die Größe der Chipfläche, und aufgrund des deutlich längeren Entwurfsprozesses zudem relativ unflexibel. Das traditionelle Hardware/Software-Codesign berücksichtigt diese Unterschiede über einen zweigeteilten, üblicherweise wechselseitigen und iterativen Entwurfsprozess. Moderne Codesign-Ansätze verbessern diesen Ansatz mit einer vereinheitlichten Programmierung, mit Modellen, die Leistung und Kosten vorhersagen können sowie mit einer Abkehr vom traditionellen Wasserfallmodell der Softwareentwicklung. Unsere Arbeitsgruppe interpretiert das Hardware/Software-Codesign darüber hinaus so, dass auch mehrere alternative Prozessorarchitekturen Teil des Entwurfsraums sind.

Aufgrund der Spezialisierung existieren heute sehr viele unterschiedliche Prozessorarchitekturen. Das führt zu der Frage, für welche Anwendung oder für welchen Algorithmus welche Prozessorarchitektur am besten geeignet ist. Dieses Problem betrachten wir im „DeepChip-Projekt" für Modelle des maschinellen Lernens, einer der wichtigsten und anspruchsvollsten Anwendungen, sowohl was das Trainieren dieser Modelle angeht als auch deren Nutzung auf kleinen mobilen Geräten. Einige unserer wichtigsten während des DeepChip-Projekts erarbeiteten Erkenntnisse bestehen darin, dass die Modelle des maschinellen Lernens oft deutlich überdimensioniert sind und dass es viele Möglichkeiten gibt, sie zu komprimieren, etwa über dünn besetzte Datenstrukturen oder über andere Datentypen. Wir konnten zudem zeigen, dass auch nicht spezialisierte Prozessorarchitekturen, beispielsweise ARM-Prozessoren, eine gute Wahl sein können, wenn man gut geeignete Methoden zur Ausführung der Modelle wählt. Dünn besetzte Strukturen sind weiterhin ein hervorragendes Beispiel für das Spannungsfeld zwischen reduzierter Komplexität und dem Aufwand für Codierung und Ausführung, insbesondere auf massiv parallelen Prozessoren. Wir plädieren auch deshalb dafür, dass Hardware/Software-Codesign nicht wie bislang als gleichzeitigen Entwurf von Hardware und Software zu interpretieren, sondern alle Schichten eines Rechnermodells einzubeziehen, das System als Ganzes zu betrachten und somit auch vorhandene Komponenten in die Überlegungen einzubeziehen.

Zukunftsbetrachtungen

Die Zukunft der Computertechnologie lässt sich aufgrund vieler Unsicherheiten nicht im Detail verlässlich vorhersagen. Sicher lässt sich aber sagen, dass Diversität und Heterogenität der Prozessorarchitekturen auch künftig weiter zunehmen werden, so dass für eine Anwendung mehr Auswahlmöglichkeiten zur Verfügung stehen werden. Schon seit Jahren skaliert das Moore'sche Gesetz nicht mehr so wie gewohnt, es sei hier aber daran erinnert, dass es sich um ein ökonomisches Gesetz handelt, das somit lediglich ökonomische Auswirkungen beobachten lässt. Wir gehen unabhängig von den ökonomischen Auswirkungen hingegen davon aus, dass die zugehörige CMOS-Technologie noch durchaus bis zu zehn Jahre weiter skaliert und somit weiterhin mehr Transistoren zur Verfügung stehen werden. Als Konsequenz wird der Hardwareent-

„Das Hardware/ Software-Codesign befasst sich mit einer möglichst guten Gesamtarchitektur für ausgewählte Anwendungen.“

PROF. DR. HOLGER FRÖNING ist seit 2019 Professor am Institut für Technische Informatik der Universität Heidelberg (ZITI) und leitet dort die „Computing Systems Group“. Nach dem Studium der Technischen Informatik und der Promotion an der Universität Mannheim kam er 2008 im Rahmen einer Kooperation mit der Universitat Politècnica de València (Spanien) an die Universität Heidelberg und wurde 2011 Juniorprofessor am ZITI. Als Gastwissenschaftler forschte er bei NVIDIA Research im kalifornischen Santa Clara (USA) und an der Technischen Universität in Graz (Österreich). 2020 wurde er als Gastwissenschaftler an die Chinesische Akademie der Wissenschaften berufen. Holger Frönings Forschungsinteressen konzentrieren sich auf Hardware- und Softwarearchitekturen sowie deren Einsatz für maschinelles Lernen und Hochleistungsrechnen. Mehrere seiner wissenschaftlichen Arbeiten haben einen Best-Paper-Award erhalten.

Kontakt: holger.froening@ziti.uni-heidelberg.de

wurf von spezialisierten Prozessoren für immer weniger Unternehmen ökonomisch machbar sein, was hingegen ein weiterer Grund ist, im Hardware/Software-Codesign vorhandene Komponenten mit zu berücksichtigen.

Hingegen ist unseres Erachtens eine viel relevantere Einschränkung der CMOS-Technologie, dass es teurer ist, Daten zu bewegen als Daten zu berechnen. Dies steht im Kontrast zu den grundlegenden Modellen der theoretischen Informatik, die Komplexität über den Aufwand der Berechnung abbilden. Die Betrachtung der Datenbewegung als Komplexitätsmetrik erfordert gänzlich neue Ansätze, um das Bewegen von Daten zu verringern oder gar zu vermeiden. Da die Kosten einer Datenbewegung auch von der zu überwindenden Distanz abhängen, kann es sogar notwendig sein, die Lokalität um ein Distanzmaß zu erweitern.

Wir können also weiterhin mit steigenden Mengen an Transistoren rechnen – allerdings sind integrierte Schaltkreise aus technischen Gründen in der Leistungsaufnahme stark limitiert. Die Fachliteratur bezeichnet dies als „Dark Silicon“: Transistoren sind zwar vorhanden, können aber aufgrund der limitierten Leistungsaufnahme nicht genutzt werden. Die bereits beschriebenen Grafikprozessoren sind auch hier ein gutes Beispiel für Innovationen: Ihre zugrunde liegende massiv parallele Architektur erlaubt es, die Ausführungseinheiten zu vereinfachen und die Operationsfrequenz zu reduzieren. Damit kann bei gleicher Leistungsaufnahme eine höhere Rechenleistung realisiert werden. Die Konsequenz einer derart massiven parallelen Ausführung sind weitere Anforderungen, die an Berechnungsvorschriften gestellt werden müssen, insbesondere dann, wenn es die dafür notwendige Lokalität zu berücksichtigen gilt: Die Berechnungsvorschriften müssen aus vielen unabhängigen Teilen bestehen (Parallelität), die sich größtenteils gleich verhalten sollten (Regularität) und gleichzeitig gut planbar sein müssen (Vorhersagbarkeit). Sonst entstehen auf massiv parallelen Systemen bei der Ausgabe und Ausführung von Anweisungen Ineffizienzen.

Aktuell existieren neben der CMOS-Technologie einige vielversprechende Kandidaten, etwa Quantencomputer oder sogenannte neuromorphe Computer. Es handelt sich jedoch meist um Erweiterungen, keiner der Kandidaten ist ein vollwertiger Ersatz. Grundlegend aber ist: Alle diese Alternativen erfordern neue Ansätze für die Hardware/Software-Schnittstelle; sie sind somit disruptiv in Bezug auf existierende Lösungen. Hingegen gibt es im Kontext von CMOS auch viele Innovationen, beispielsweise neuartige Transistoren, ein extremes Verringern der Versorgungsspannung und nicht zuletzt Speichertechniken, die sich besser mit CMOS kombinieren lassen. Diese Innovationen sind nicht oder deutlich weniger disruptiv als die vorab genannten CMOS-Alternativen.

„Die Grenze zwischen Hard- und Software stellt nicht nur die Grenze zwischen Programmierbarkeit und Leistungsfähigkeit dar – sie ist zugleich eine der wichtigsten Abstraktionen der Informatik.“

AHEAD OF ITS TIME

THE TRADE-OFF BETWEEN HARDWARE AND SOFTWARE

HOLGER FRÖNING

The boundary between software and hardware also defines the boundary between a computer's performance and its programmability. Shifting the boundary towards hardware allows for more flexibility in programming, but performance is diminished as more functionality is implemented in software, which is slow in comparison to hardware. By contrast, shifting the boundary towards software means that more functionality is implemented in fast, efficient, but rigid hardware, creating the opposite effect. While the latter is desirable from a performance point of view, it prevents innovation in applications and algorithms. Researchers at the Institute of Computer Engineering (ZITI) of Heidelberg University are attempting to solve this trade-off, which is currently one of the most fundamental problems in computer science. ●

PROF. DR HOLGER FRÖNING has been a professor at Heidelberg University's Institute of Computer Engineering (ZITI) since 2019, where he heads the "Computing Systems Group". After graduating in computer engineering and obtaining his PhD from the University of Mannheim, he started at Heidelberg University in 2008 in the context of a collaboration with the Universitat Politècnica de València (Spain), and became a junior professor at ZITI in 2011. He was a visiting scientist at NVIDIA Research in Santa Clara, California (USA) and at Graz University of Technology (Austria). In 2020, he was appointed visiting scientist at the Chinese Academy of Sciences. Holger Fröning's research interests are hardware and software architectures and their use for machine learning and high-performance computing. Several of his publications have received a Best Paper Award.

Contact: holger.froening@ziti.uni-heidelberg.de

"If both hardware and software are modifiable, we are soon faced with the question of where to draw the boundary between these two components."

„Die Diversität und Heterogenität der Prozessorarchitekturen wird auch künftig weiter zunehmen, so dass für eine Anwendung mehr Auswahlmöglichkeiten zur Verfügung stehen werden.“

Eine der wichtigsten Abstraktionen der Informatik
Halten wir fest: Die Grenze zwischen Hard- und Software stellt nicht nur die Grenze zwischen Programmierbarkeit und Leistungsfähigkeit dar – sie ist zugleich eine der wichtigsten Abstraktionen der Informatik. Eine Verschiebung der Grenze bis hin zum Extrem einer nicht mehr programmierbaren Hardware würde zwar eine höhere Computerleistung um circa drei Größenordnungen erbringen. Sie steht aber im starken Widerspruch zu kontinuierlichen Innovationen in den Anwendungen, Algorithmen und Methoden. Umgekehrt steht eine hohe Programmierbarkeit im Konflikt mit den stetig steigenden Anforderungen, die an Anwendungen gestellt werden, mit deren Hilfe immer größere Probleme immer schneller bearbeitet werden sollen.

Schließen möchte ich mit einem Vergleich der Architektur von Prozessoren mit der Architektur von Gebäuden. In einem Beitrag in der „Frankfurter Allgemeinen Zeitung“ hieß es vor einigen Jahren, dass sich der architektonische Wert eines Gebäudes erst dann erweise, „wenn es in vollkommen gewandelten ästhetischen und politischen Verhältnissen nicht nur standhält, sondern ihnen sogar entgegenkommt“. Für die Prozessor- und Systemarchitektur gilt Ähnliches: Sie muss gewandelten Verhältnissen bezüglich Anwendungen, Algorithmen und Methoden nicht nur standhalten, sondern solche Veränderungen beziehungsweise Innovationen durch entsprechende Flexibilität sogar fördern. Nur dann kann sich auch der Wert der Prozessorarchitektur über die Zeit bewahren. ●