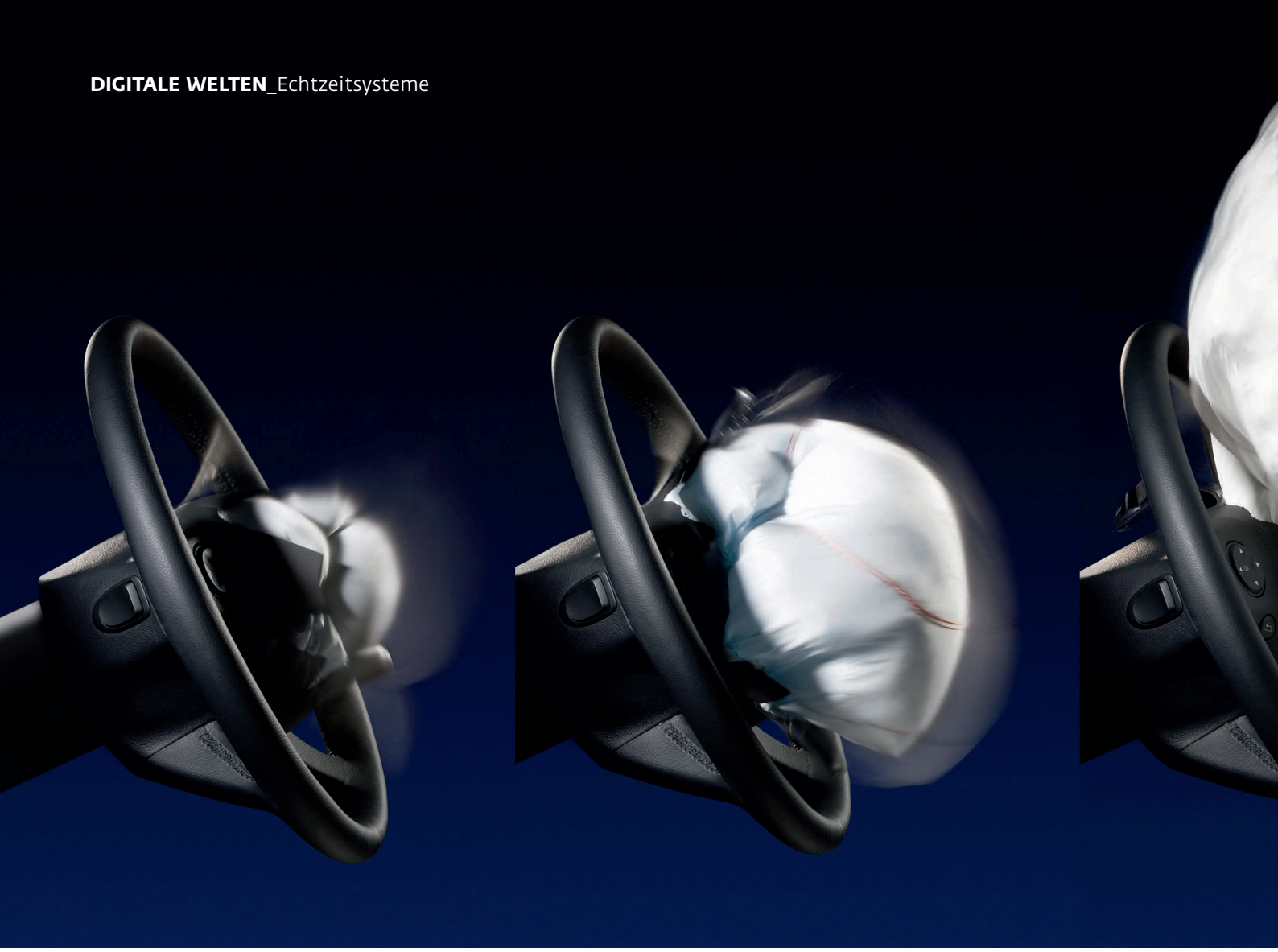


# Mit Sicherheit **pünktlich**

Wenn ein Computer eine Webseite extrem langsam aufbaut, ist das vielleicht ärgerlich, aber nicht mehr. Wenn jedoch die Elektronik im Auto oder in Flugzeugen nicht absolut pünktlich Befehle verarbeitet, dann kann das lebensgefährlich werden. Unter welchen Bedingungen die dort gefragten Echtzeitsysteme zuverlässig funktionieren, untersuchen **Björn Brandenburg** und sein Team am **Max-Planck-Institut für Softwaresysteme** in Kaiserslautern und Saarbrücken.

TEXT **ALEXANDER STIRN**

Foto: Daimler AG

Wenn Bruchteile von Sekunden überlebenswichtig sind: Ehe der Kopf des Fahrers auf das Lenkrad aufprallt, muss der Airbag aufgeblasen sein. Seine Steuerung muss ihn daher absolut pünktlich auslösen.

Pünktlichkeit ist unterbewertet – zumindest in der Informatik. Wenn der Airbag im Auto ein paar Millisekunden zu spät auslöst, weil sein Steuergerät herumtrödelt, wird es gefährlich. Wenn das Handy den winzigen Augenblick verschläft, für den es von seiner Funkzelle die Erlaubnis zur Kommunikation erhalten hat, dann bleiben die Daten stecken. Wenn ein Pilot ein Flugzeug landen will, seine Kommandos aber nicht rechtzeitig bei den Turbinen oder Landeklappen ankommen, kann das fatale Folgen haben.

„Die physikalische Welt hört nicht auf, bloß weil der Rechner nicht hinterherkommt", erklärt Björn Brandenburg, Nachwuchsgruppenleiter am Max-Planck-Institut für Softwaresysteme in Kaiserslautern und Saarbrücken. Dort kümmert sich der Informatiker um sogenannte Echtzeitsysteme – Anwendungen, bei denen ein Computer unter keinen Umständen zu spät dran sein darf.

Die theoretische und praktische Umsetzung solcher Systeme stellt die Informatik vor große Herausforderungen: Normalerweise arbeitet ein Programm korrekt, sobald bestimmte Eingaben zu den logisch korrekten Ergebnissen führen. „Bei uns dagegen ist ein System nur dann korrekt, wenn es die richtigen Ausgaben auch zum richtigen Zeitpunkt macht", sagt Brandenburg.

Als Leiter der Forschungsgruppe für Echtzeitsysteme untersucht der 30-Jährige daher, wie in einer zunehmend komplexen und miteinander vernetzten Umgebung garantiert werden kann, dass solche Systeme sicher und zuverlässig arbeiten. Während in der technischen Praxis Experimente und Intuitionen immer noch eine wichtige Rolle spielen, setzt Brandenburg auf harte Mathematik: „Für sicherheitskritische Anwendungen brauchen wir Analysemethoden, die mathematisch wohlfundiert sind und korrekt beweisen, dass ein System stets wie gewünscht funktioniert", sagt der Max-Planck-Forscher.

Die Anforderungen an Echtzeitsysteme unterscheiden sich damit grundlegend vom Informatikalltag: Ob sich das Programmfenster mit der gerade gelesenen E-Mail sofort nach dem Mausklick oder erst mit etwas Verzögerung schließt – dafür braucht es keine mathematischen Modelle. Selbst wenn manchmal gar nichts passiert, flucht der Anwender vielleicht, aber die Welt geht nicht unter. Sollen derartige Systeme verbessert werden, vertrauen Entwickler folglich auf ihre Erfahrung, auf ihre Ideen. Sie schauen sich den Programmcode an


Anwälte der Pünktlichkeit: Björn Brandenburg (rechts) und Alexander Wieder entwickeln Methoden, um zu beweisen, dass sicherheitsrelevante Systeme den Job, der von ihnen erwartet wird, pünktlich erledigen – immer.

und testen die Neuerungen in groß angelegten Studien. „Wenn das im Schnitt gut funktioniert, ist das für allgemeine Systeme völlig okay“, sagt Brandenburg.

Bei sicherheitskritischen Echtzeitsystemen wie dem Airbag im Auto helfen Durchschnittswerte nicht weiter. Das Produkt muss funktionieren. Punkt. „Im Ganzen sind die Systeme inzwischen allerdings so komplex geworden, dass es in der Praxis nicht mehr reicht, mit dem menschlichen Auge scharf hinzuschauen“, sagt Brandenburg.

## MEHR ALS 100 MIKROCOMPUTER IN EINEM LUXUSAUTO

So sind allein in einem modernen Luxusauto mehr als hundert Mikrocomputer verbaut, die sämtliche Systeme vom Airbag über die Motorsteuerung bis hin zum Radio kontrollieren. Die meisten bearbeiten dabei mehrere Aufgaben – das macht die mathematische Beschreibung so kompliziert: „Wenn ich ausschließen will, dass etwas schiefgehen kann, wird das umso schwieriger, je mehr Dinge miteinander interagieren“, sagt Björn Brandenburg.

Der Informatiker vergleicht die Situation gern mit einem Büro: Der dortige Mitarbeiter, der in diesem Bild einen Prozessor darstellt, steht unter Druck. Immer wieder will der Chef etwas von ihm; dessen Aufgaben müssen sofort abgearbeitet werden. Aber auch die Kollegen haben Fragen und wollen mit ihren Anliegen nicht den ganzen Tag warten.

Das Büro funktioniert nur, wenn alle Aufgaben in der gewünschten Zeit gelöst werden und wenn der arme Angestellte bei Feierabend keinen Berg an Arbeit vor sich herschiebt. Vor allem aber dürfen die einzelnen Wünsche, insbesondere die des Chefs, nicht so lange liegen bleiben, dass schon wieder die nächste Anfrage mit gleicher Priorität eintrudelt. Denn sonst endet all das im Chaos. Die Informatiker sprechen von einem nichtlinearen Verhalten, das urplötzlich zu Sprüngen bei der Reaktionszeit führt.

„Um solche Vorgänge mathematisch beschreiben zu können, brauchen wir Modelle, die die Welt gut abbilden, mit deren Hilfe wir aber gleichzeitig auch beweisen können, dass die Systeme die in sie gesetzten Anforderungen erfüllen“, sagt Brandenburg. Deshalb packen im Fall des Büros – oder eines einfachen Echtzeitsystems – die Informatiker die Häufigkeit der einzelnen Aufgaben, den Arbeitsaufwand für ihre Bearbeitung und die gewünschte Zeit, in der eine Antwort verlangt wird, in mathematische Gleichungen. Oft fällt das Formelsystem dabei so kompliziert aus, dass es sich analytisch nicht mehr lösen lässt. Dann müssen Informatiker durch Annahme eines Startwerts und durch konsequentes Verfeinern Schritt für Schritt eine Lösung suchen – ein Standardverfahren, das Mathematiker Fixpunktiteration nennen.

Ein weiteres Problem besteht darin, realistische Minimal- und Maximalwerte für den Arbeitsaufwand, die Häufigkeit und die erlaubte Antwortzeit zu finden. Nur mit solchen Angaben lassen sich die Formeln auf Herz und Nieren prüfen. „In der Praxis übernimmt das häufig ein Ingenieur mit viel Erfahrung, der ein bisschen testet und dann noch eine Sicherheitsmarge draufschlägt“, sagt Brandenburg. „In vielen Fällen kann das sogar eine hinreichend gute Abschätzung sein.“

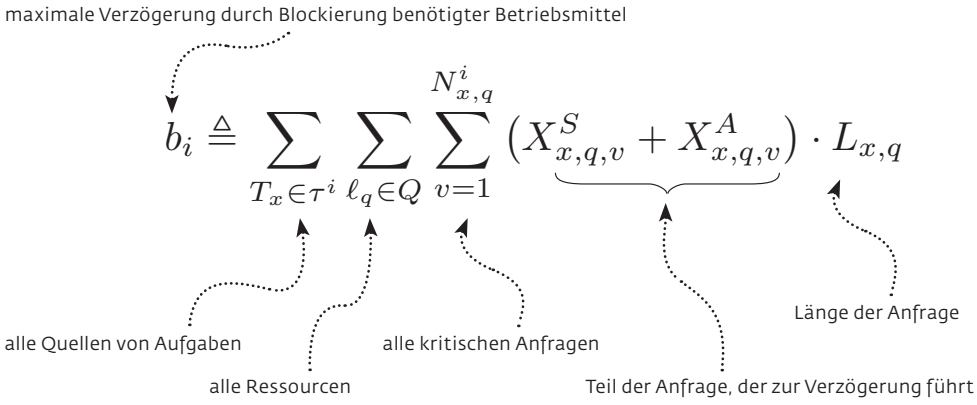
## PROZESSOREN SIND HEUTE UNBERECHENBARER ALS FRÜHER

Beim Airbag ergibt sich die maximal erlaubte Dauer der Berechnung zum Beispiel aus der Zeit, die zwischen dem Crash und dem Aufprall des Fahrerkopfs auf das Lenkrad vergehen würde. Bis dahin muss der Airbag komplett aufgeblasen sein – und daraus lässt sich zurückrechnen, wie schnell die Software reagieren muss. „Letztlich leitet sich die Deadline immer aus den physikalischen Anforderungen des Systems ab“, sagt Brandenburg.

Der Motor macht da keine Ausnahme. Bei ihm diktiert die erlaubte Drehzahl die minimale Zeit zwischen zwei Vorgängen. Muss beispielsweise einmal pro Umdrehung die Abgaskonzentration ausgelesen werden, ergibt sich bei einer Maximaldrehzahl von 6000 Umdrehungen pro Minute (oder 100 Umdrehungen pro Sekunde) eine kleinstmögliche Pause von einer Hundertstelsekunde.

Deutlich komplizierter ist es, die tatsächliche Rechenzeit eines modernen Systems zu ermitteln und so abzuschätzen, ob die Arbeit auch im vorgegebenen Rahmen erledigt werden kann. Bei früheren Prozessorgenerationen war das noch vergleichsweise einfach möglich. Damals konnten Ingenieure im

Foto: Thomas Hartmann




Eine Formel für Verzug: Die Max-Planck-Forscher untersuchen Echtzeitsysteme, deren Prozessoren sich andere Ressourcen teilen. Die Systeme erledigen eine sicherheitsrelevante Aufgabe möglicherweise zu spät, weil sie auf eine Ressource nicht zugreifen können. Die maximale Verzögerung ergibt sich aus der Summe Σ aller Teile von Anfragen, die zu Verzögerungen führen, und der jeweiligen Länge dieser Anfragen. Die Anfragenteile können zu verschiedenen Aufgaben gehören, die aus verschiedenen Quellen stammen, und auf mehrere Ressourcen angewiesen sein. Um die Verzögerung berechnen zu können, ist die Bestimmung der Anfragenteile $X^S_{x,q,v}$ und $X^A_{x,q,v}$, die zur Verzögerung führen, entscheidend. Dafür nutzen Björn Brandenburg und seine Kollegen erstmals ein Verfahren namens lineare Optimierung.

Maschinencode abzählen, wie viele Rechenschritte nötig sind, um eine Aufgabe abzuarbeiten. Da der Arbeitstakt des Prozessors bekannt war, ließ sich daraus die benötigte Zeit ermitteln.

Heutige Prozessoren sind viel unberechenbarer: Sie versuchen zu erraten, welche Aufgabe als nächste ansteht, und bereiten sich darauf vor. Sie können ihre Taktfrequenz regulieren. Sie greifen auf eine Vielzahl von Zwischenspeichern zurück. Inzwischen kümmert sich ein eigenes Forschungsfeld der Informatik, die *Worst-Case Execution Time Analysis*, um nichts anderes, als bei einem vorgegebenen Programm und einer vorgegebenen Hardware die maximale Rechenzeit im schlimmsten aller Fälle zu bestimmen.

„Es ist unheimlich schwer, so etwas exakt zu machen", sagt Brandenburg. „Deshalb kann es sinnvoll sein, echte Systeme zu beobachten und Messwerte zu extrahieren." Eine Software protokolliert dabei zum Beispiel alle Kommandos, die während einer Testfahrt im Auto abgearbeitet werden. Ingenieure können daraus Daten wie die maximale Ausführungszeit ableiten. Danach wird noch ein kleiner Sicherheitspuffer eingebaut, fertig sind die Ausgangswerte für die computergesteuerte Analyse der Programme. „Die Puristen in unserem Feld würden sagen: Da ist ja eine Messung drin. Damit kann man doch nicht streng mathematisch beweisen, dass das wirklich der schlimmste anzunehmende Fall ist", sagt Brandenburg. „Ich bin da pragmatischer. Eine Analyse der Antwortzeiten ist auf jeden Fall besser als irgendeine Tabellenkalkulation, in die ein paar mehr oder weniger willkürliche Zahlen eingetippt worden sind."

Erst nachdem alle Randbedingungen ermittelt werden konnten, kommt die Mathematik zum Zug. „Wenn wir dann zeigen können, dass die Antwortzeit selbst in den allerschlimmsten Fällen niemals größer als die gewünschte Deadline ausfällt, wissen wir: Das passt so, das System ist sicher", sagt Alexander Wieder, Doktorand in Brandenburgs Forschungsgruppe.

## INGENIEURE DEFINIEREN DIE ANFORDERUNGEN

So etwas beweisen Mathematiker meist durch ein Verfahren, das sie „Widerspruchsbeweis" oder auch „indirekten Beweis" nennen. Die Forscher nehmen dabei an, dass die vorgegebene Ausführungszeit überschritten wurde – dass also genau der Fall eingetreten ist, den sie eigentlich ausschließen möchten. Dann schauen sie, welche Schlussfolgerungen sich daraus ableiten lassen: Wurde die Aufgabe nicht wie geplant erfüllt, gibt es zwei Möglichkeiten: Entweder die Bearbeitung einer Aufgabe hat länger gedauert als angenommen, oder der Prozessor muss noch etwas anderes gemacht haben, während er den sicherheitsrelevanten Job erledigen sollte – er war also offenbar mit Anfragen ausgelastet, die er der wichtigsten Aufgabe vorgezogen hatte.

Im nächsten Schritt schauen sich die Informatiker alle Prozesse genauer an, analysieren, wie viel Arbeit diese verursachen. Das Verfahren wird Schritt für Schritt verfeinert, bis am Ende ein paar Prozesse übrig bleiben, die unter den gegebenen Annahmen mehr Arbeit verursachen müssen, als sie laut Modell eigentlich dürften. Ein Widerspruch. „Für uns bedeutet das: Entweder ist unser Modell falsch, oder so etwas ist unmöglich", sagt Brandenburg. Die vorgegebene Ausführungszeit würde somit niemals überschritten. Das System ist hundertprozentig sicher – zumindest unter den gegebenen Modellannahmen.

Die Modelle beruhen auf Anforderungen, die Ingenieure definieren. Jenseits dieser Spezifikationen, wenn der Motor zum Beispiel mit mehr als 6000 Umdrehungen pro Minute läuft, gibt es keine mathematische Sicherheit mehr. Die Software kann noch funktionieren, sie muss aber nicht. Die Ingenieure müssen daher ihr Fachwissen einbringen, um sichere und vollständige Anforderungen an das System zu stellen.

Häufig simuliert wird dagegen eine Art sprunghafter Chef, der viele Aufgaben innerhalb kurzer Zeit verteilt und sich dann erst einmal in sein Büro zurückzieht. *Busty arrivals*, „stoßweise Eingänge", nennen Informatiker das Phänomen. Es lässt sich recht gut in Algorithmen packen, auch wenn der Aufwand für die Beweisführung dadurch etwas steigt.

Wirklich kompliziert werden Echtzeitanwendungen allerdings, sobald sie auf einem Prozessor bearbeitet werden, der sich mit weiteren Prozessoren andere Ressourcen teilen muss. So können die

Grafik: MPI für Softwaresysteme

links: Dolmetscher zwischen zwei Fachsprachen: Björn Brandenburg versteht sowohl die Ingenieure, die Echtzeitsysteme entwickeln, als auch die Theoretiker, die deren Zuverlässigkeit beweisen wollen. Daher möchte er zwischen beiden Disziplinen vermitteln.

rechte Seite: Damit ein Flugzeug sicher landet, muss die Elektronik die Steuerbefehle des Piloten unbedingt in der vorgegebenen Zeit an die Turbinen oder Landeklappen weitergeben.

Schalteinheiten etwa gemeinsam auf einen Nachrichtenpuffer zugreifen, der Sensordaten speichert, solange sie nicht weiterverarbeitet werden können. Systeme mit geteilten Ressourcen bilden einen Schwerpunkt der Forschung am Max-Planck-Institut für Softwaresysteme.

Im Bild des Büros, in dem ein Mitarbeiter für einen Prozessor steht, hieße das: Den Bürokräften, die normalerweise ungestört nebeneinanderher arbeiten, steht zum Beispiel nur ein Kopierer oder ein Telefon zur Verfügung. Da kommt es unweigerlich zu Diskussionen. Einige Mitarbeiter müssen mit einer Aufgabe dringend fertig werden, andere hätten eigentlich mehr Zeit, wollen aber trotzdem nicht warten. „Je nachdem, in welcher Reihenfolge man die Leute ranlässt, kann das entscheiden, ob alle Deadlines eingehalten werden“, sagt Björn Brandenburg.

In der Praxis verfolgen Informatiker unterschiedliche Ansätze: Die Prozessoren können fein säuberlich in eine Warteschlange eingereiht werden, es lassen sich Prioritäten verteilen, das Los kann entscheiden, oder es kann eine bunte Mischung aus sämtlichen Möglichkeiten zum Einsatz kommen. Auch bei der Frage, wie sich die einzelnen Prozessoren die Wartezeit vertreiben sollten, führen mehrere Wege zum Ziel: Sie können pausenlos fragen: „Kann ich? Kann ich? Kann ich?“, bis sie endlich an der Reihe sind. Oder sie können darauf warten, an den Kopierer gebeten zu werden.

Theoretisch ist das Warten die bessere Lösung, da die Mitarbeiter während dieser Zeit andere, nicht ganz so wichtige Aufgaben anpacken können. In der Praxis ist der ständige Wechsel zwischen Kopieren und Telefonieren aber mit beträchtlichem Mehraufwand verbunden – nichts kommt wirklich voran. In kritischen Systemen wie dem Auto erhalten daher meistens die quengelnden Prozessoren den Vorzug.

Brandenburg und seine Gruppe sind nicht die Ersten, die sich dieses Problems annehmen. Bislang haben Informatiker die geteilten Ressourcen aber meist von Hand analysiert: Sie haben über mögliche Verzögerungen nachgedacht und daraus einen Maximalwert für die Antwortzeit ermittelt. Dabei genügt allerdings eine falsche Annahme, um zu einem Ergebnis zu kommen, das nicht mehr sicher ist. Zudem fallen die Abschätzungen unglaublich pessimistisch aus. Entsprechend unrealistisch sind die Ergebnisse. „Von den Ingenieuren hört man dann: Schön, dass ihr nun eine sichere Antwort habt, aber in der Praxis wird der Wert nie so hoch sein. Das ist nutzlos“, erzählt Brandenburg.

## EINE OBERGRENZE FÜR MÖGLICHE BLOCKADEN

Die Max-Planck-Forscher wählen deshalb einen anderen Weg. Sie erstellen zunächst eine Gesamtmenge von Arbeitsabläufen, die in einem Büro – oder einem Echtzeitsystem – theoretisch nicht unmöglich sind, die also erst einmal nicht ausgeschlossen werden können. Anschließend blenden sie nach und nach all die Fälle aus, die in der Praxis nicht passieren können. Zum Beispiel ist es undenkbar, dass eine Bürokraft alle Geräte gleichzeitig benutzen will.

Letztlich kommen auf diese Weise Dutzende oder gar Hunderte Beschränkungen zusammen. All das lässt sich in Gleichungen packen. Schritt für Schritt schrumpft somit die Menge aller nicht ausgeschlossenen Aufgaben zusammen – so lange, bis eine Obergrenze gefunden ist für all die Prozesse, die die geplanten Abläufe im Büro blockieren können. Dieser Wert kann dann als weiterer Faktor in jene Algorithmen einfließen, mit denen die Pünktlichkeit der Echtzeitsysteme bewiesen wird.

„Lineare Optimierung“ heißt das zugrunde liegende Verfahren, das in der Mathematik seit mehr als sechzig Jahren angewandt wird. Inzwischen haben Informatiker daraus sehr viele, sehr schnelle Analysesysteme entwickelt, deren Algorithmen die Max-Planck-Forscher auf Echtzeitsysteme anpassen können. Brandenburg sagt: „Wir können nun deutlich kompliziertere Zusammenhänge analysieren und haben dadurch unheimliche Fortschritte bei der Genauigkeit gemacht.“

All das ist in der Praxis vor allem deshalb wichtig, weil viele Echtzeitsysteme heutzutage sogenannte Multicore-Prozessoren verwenden. Mehrere voneinander unabhängige Rechenkerne teilen sich dabei die Arbeit, sie greifen auf gemeinsame Ressourcen zurück, sie werkeln nebeneinanderher, ohne sich großartig abzusprechen. „Für Echtzeitsysteme stellen Multicore-Prozessoren eine extreme Herausforderung dar“, sagt Björn Brandenburg.

Die Informatik hinkt da hinterher. Lange Zeit galten die komplexen Prozessoren als theoretisches Problem, als et-

Foto: Thomas Hartmann

was, das bei zeitkritischen Anwendungen nicht verbaut wird. Heute sind die Anforderungen selbst im Auto so hoch, dass Multicore-Prozessoren fast schon Standard geworden sind. Die bewährten Algorithmen funktionieren hier nicht mehr, alles muss neu entwickelt werden. „Das bislang verwendete Argument: Wenn ein Prozessor einer Aufgabe nachgeht, dann können andere, potenziell störende Aktivitäten zur gleichen Zeit nicht stattfinden, ist ein unheimlich mächtiges Element in der Beweisführung“, sagt Brandenburg. „Bei Multicore-Systemen funktioniert das nicht mehr, da weiß ich nie, was die anderen Kerne gerade machen.“

Vor allem die maximal benötigte Rechenzeit, die sich bereits auf modernen Einkernprozessoren kaum ermitteln lässt, stellt Informatiker vor große Herausforderungen. Auch die Frage, wie stark ein Multicore-Prozessor ausgelastet ist und wie viel Zeit er sich mit seinen Antworten lässt, ist ungeklärt. Bislang konnten die Forscher lediglich zeigen, dass der schlimmste anzunehmende Fall, in dem sämtliche Prozesse mit hoher Priorität den Rechner gleichzeitig auslasten, unter gewissen Umständen existieren muss. Wie er aussieht, wie er in Formeln gepackt werden kann und wie sich dadurch die Analyse verbessern lässt, bleibt offen.

Noch ein weiteres – wesentlich menschlicheres – Problem macht den Max-Planck-Forschern zu schaffen. Viele Theoretiker, die an neuen Algorithmen für Mehrkernprozessoren tüfteln, sind zwar mathematisch gut geschult, sie sitzen jedoch selten im Labor und schreiben Programme. Die Ingenieure in der Industrie dagegen sind hochspezialisiert und bringen mit ihren Tricks sogar die obskursten Systeme zum Laufen. Sie sind aber nicht unbedingt die besten Mathematiker.

Stattdessen versuchen sie, ihre Erfahrung in mehrere Tausend Seiten umfassende Regelwerke zu pressen, die – wie beispielsweise im Automobilbau – die Anforderungen an jedes System im Detail vorgeben. Das wiederum missfällt den Theoretikern, die für ihre Sicherheitsbeweise so wenige Vorgaben wie möglich bevorzugen. „Ein Teil meiner Forschungsarbeit besteht darin, in der Mitte zu sitzen und zu sagen: Ich verstehe die Mathematiker, ich verstehe die Praktiker, ich versuche da mal zu dolmetschen“, sagt Brandenburg. Gerade in der Echtzeitforschung ist es dafür höchste Zeit. ◂

## AUF DEN PUNKT GEBRACHT

- **Wenn es etwa um die Sicherheit von Autos oder Flugzeugen geht, muss Elektronik Daten absolut zuverlässig in einer festgelegten Zeit verarbeiten. Ob solche Echtzeitsysteme tatsächlich pünktlich ihre Aufgaben erledigen, können Experimente nicht zweifelsfrei belegen. Dafür braucht es mathematische Beweise.**
- **Moderne elektronische Prozessoren besitzen mehrere Rechenkerne, bearbeiten mehrere Aufgaben parallel und variieren ihre Taktfrequenz. Das macht die mathematische Beschreibung solcher Systeme und die Beweise, dass sie funktionieren, sehr kompliziert.**
- **Max-Planck-Forscher gehen bei der Beweisführung teilweise neue Wege, etwa indem sie auf die lineare Optimierung setzen. So können sie kompliziertere Zusammenhänge genauer analysieren als bisher.**

## GLOSSAR

**Echtzeitsystem:** Ein System, wie etwa eine elektronische Steuerung oder ein Mikrocomputer, das einen Prozess zwingend innerhalb einer vorgegebenen Zeit abarbeiten muss. Echtzeitsysteme kommen bei sicherheitsrelevanten Aufgaben, etwa der Steuerung eines Airbags, zur Anwendung.

**Lineare Optimierung:** Sucht für ein Problem, das als lineare mathematische Gleichung oder Ungleichung formuliert werden kann, die optimale Lösung unter gegebenen Randbedingungen, die sich ebenfalls in linearen Gleichungen oder Ungleichungen ausdrücken lassen.

**Widerspruchsbeweis:** Ein indirekter Beweis, der eine Aussage belegt, indem er ihr Gegenteil widerlegt. Die zu widerlegende Aussage wird dabei so weit reduziert, dass sich ein Widerspruch zu einer als gesichert angenommenen These ergibt.