Eine *success story* mit TRACE32

Auf der Jagd nach Linux-Zeitfressern mit hardwarebasiertem Tracing

Die Fähigkeit zur Analyse des Laufzeitverhaltens eines Zielsystems ist in vielen Fällen ein außerordentlich wichtiges Element des Debugging-Prozesses, das aber häufig übersehen wird. Bei Echtzeitsystemen ist eine verspätete Antwort oft genauso fatal wie eine falsche Antwort. Zwar existieren insbesondere für Linux verschiedene Softwaretools, die die Leistung eines eingebetteten Systems messen, doch können diese das Problem in manchen Fällen sogar noch verschärfen. Dieser Artikel erzählt, wie es der Advanced Driver Information Technology GmbH (ADIT) in Hildesheim gelungen ist, dieses Problem mit dem nicht-intrusiven Tracing-Tool TRACE32 von Lauterbach zu meistern.

Bei ADIT läuft unter anderem Linux sowohl auf Arm- als auch auf Intel-Prozessoren. Die allgemeine Systemleistung wird dabei mit SystemTap [1] gemessen, damit man Engpässe aufspüren und beseitigen kann. SystemTap bedient sich uprobe und kprobe, eines hilfreichen Linux-Features, mit dem der Benutzer ein dynamisches Tracing der Funktionen auf Benutzer- und Kernel-Ebene vornehmen kann.

Bei leichten bis moderaten Systembelastungen traten keine besonderen Probleme auf. Man ging davon aus, dass sich ein Software-Tool wie SystemTap kaum auf die Echtzeit-Performance des Gesamtsystems auswirken würde. Was man aber nicht erwartet hatte: Auf den Arm-basierten Plattformen wurde das System deutlich langsamer als auf deren Intel-Gegenstücken. Zur Bestätigung schuf ADIT eine Pseudofunktion und nahm Messungen mit uprobe vor. Dabei zeigte sich, dass ein einziger Aufruf von uprobe auf der Arm-Plattform zweimal so lange dauerte. Da uprobe intern auf kprobe zurückgreift, bestand zunächst der Verdacht, dass das Problem bei letzterem zu suchen sei. Das erwies sich als falsch, da kprobe auf dem Arm-Prozessor sogar schneller lief als auf Intel: Das Problem lag also eindeutig im uprobe-Code.

Da das Problem im Code des Software-Tracing lag, konnte es auch nicht durch Software-Tracing gefunden werden.

"Ich hatte keine Ahnung, wie ich weitermachen sollte, weil der Kernel-uprobe-Code nicht gerade unkompliziert ist. Daher beschloss ich, es mit TRACE32 zu versuchen, um mir ein Bild darüber zu machen, was vor sich geht. Manchmal ist es hilfreich, eine nette Graphik zu haben. Anhand des Diagramms konnte ich verschiedene Codebereiche auswählen und dann ausführlicher betrachten", erklärt ADIT-Entwickler Frederic Berat.

Also entschied man sich bei ADIT, TRACE32 PowerTrace mit Hardware-Tracing zu verwenden. Der hardwarebasierte Trace hat keinerlei Einfluss auf den Zeitverlauf des Ziels und ermöglicht so eine Tiefenanalyse auch kleinster Codeabschnitte.

Sowohl Arm- als auch Intel-Geräte können nicht-intrusive Informationen über Programmabläufe liefern. Bei Arm heißt dies Embedded Trace Macrocell (ETM), das Intel-Äquivalent heißt Intel Processor Trace (IPT). Informationen über die Ausführung des Codes werden über fest zugeordnete Pins übertragen. Die TRACE32-Tools erfassen die Daten über diese Pins und werten sie dann aus. So entsteht

Lauterbach GmbH • Altlaufstraße 40 • D-85635 Höhenkirchen-Sieg. • Tel. +49 8102 9876-0 • marketing@lauterbach.com • www.lauterbach.com

ein Funktionsablaufdiagramm über die Anwendung mit ausführlichem Zeitverlauf der einzelnen Funktionen.

Selbst in komplexen Umgebungen vermag TRACE32 noch den kompletten Programmablauf zu erfassen und auszuwerten, einschließlich der Anwendungen auf Benutzerebene und des Kernel-Codes. Der Funktionsablauf des kompletten Systems wird rekonstruiert und als Zeitdiagramm bzw. Funktionshierarchie statistisch dargestellt. Die Darstellung des Laufs eines vollständigen Linux-Systems einschließlich Kernel und Prozessen ergibt ein sehr umfangreiches Diagramm. Aber TRACE32 hilft bei der Auswertung der entscheidenden Teile tatkräftig mit – so gelang es den ADIT-Technikern, sich strikt auf die kprobe- und uprobe-Anteile des Kernels zu konzentrieren.

Mithilfe der extrem leistungsfähigen Analyse-funktion von TRACE32 wurde schnell klar, dass zwei Engpässe vorlagen (Bild 1 und 2). Besonders bemerkenswert war, dass uprobe auf der Arm-Plattform viermal preempt_disable() und preempt_enable() aufruf, wobei jedes Mal der Stack-Frame geprüft wurde, was ca. 0,6 µs, insgesamt also 2,4 µs kostete. Bei den Intel-Prozessoren war das nicht der Fall. Ein einziger Unterschied von nur 2,4 µs mag sich nach nicht viel anhören. Aber wenn es viele uprobe-Anforderungen gibt, führt das schnell zu einer deutlichen Verzögerung. Eine noch tiefergehende Analyse enthüllte einen Unterschied bei den String-Operationen, die ein notwendiger Bestandteil von uprobe sind. Dies konnte aber dem architektonischen Unterschied zwischen Arm und Intel zugeschrieben werden.
Ohne Echtzeit-Tracing wäre es praktisch unmöglich gewesen, dem auf den Grund zu gehen – mit Echtzeit-Tracing dagegen war es ein Kinderspiel. Da man nun wusste, wo man zu suchen hatte, fand man bei ADIT das Problem in der Kernel-Konfiguration. Bei der Migration von einer anderen Plattform hatte man eine temporäre Einstellung von

CONFIG_PREEMPT_TRACE

versehentlich aktiviert gelassen. Das Tracing zeigte, dass es hierdurch bei Arm zu einer Stack-Abwicklung, bei Intel dagegen aber zu einem "no-op" kam. Und das war auch ist die Erklärung für den extremen Leistungsunterschied zwischen den beiden Plattformen.

[1] https://en.wikipedia.org/wiki/SystemTap

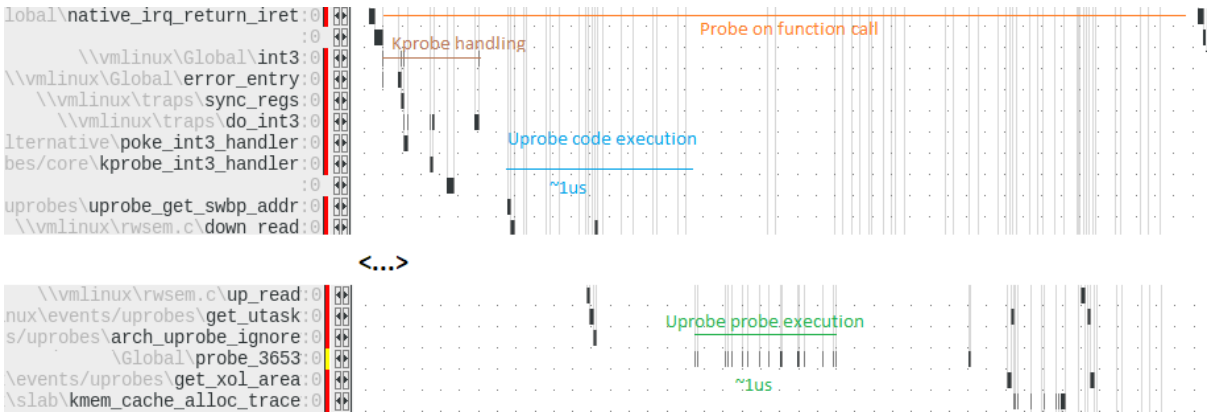

Bild 1: uprobe-Aufrufe bei Intel-Prozessoren

Lauterbach GmbH • Altlaufstraße 40 • D-85635 Höhenkirchen-Sieg. • Tel. +49 8102 9876-0 • marketing@lauterbach.com • www.lauterbach.com

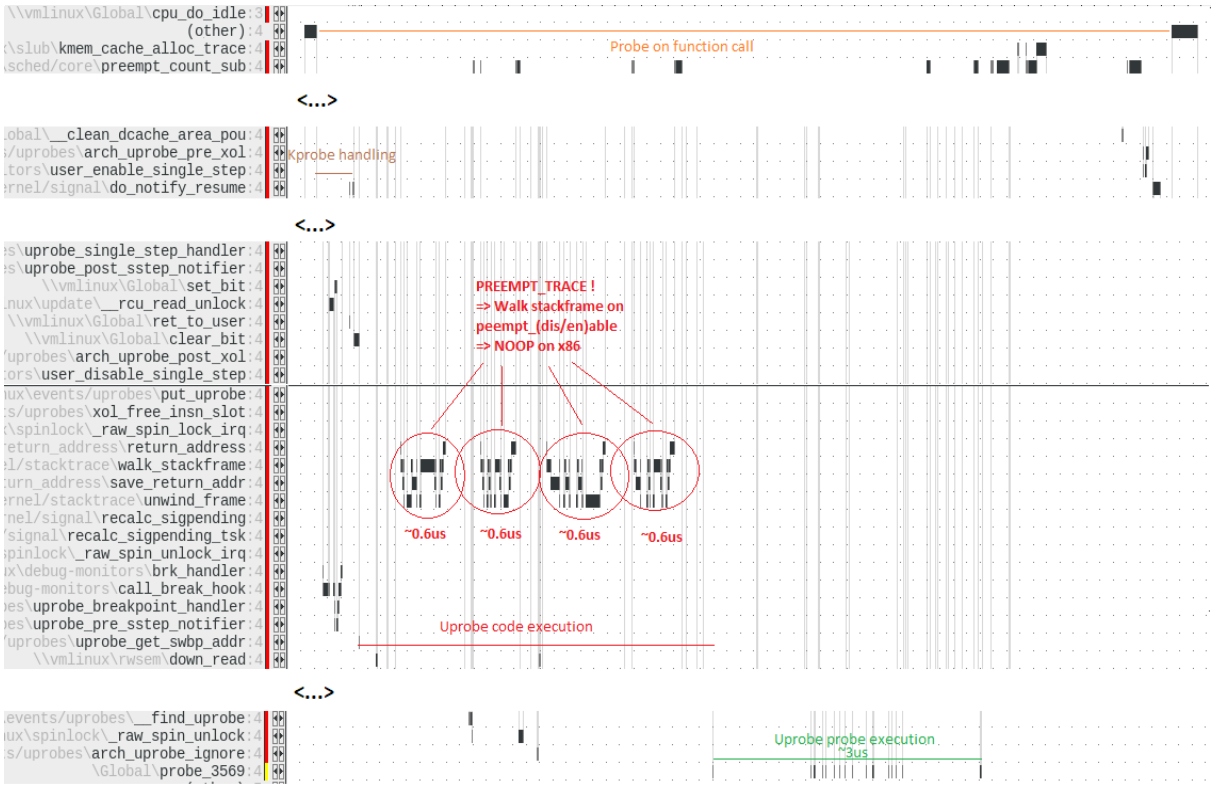

Bild 2: uprobe-Aufrufe bei Arm

Veröffentlichung in Heft Nr. 12
der Design & Elektronik
vom 19. November 2018

Lauterbach GmbH • Altlaufstraße 40 • D-85635 Höhenkirchen-Sieg. • Tel. +49 8102 9876-0 • marketing@lauterbach.com • www.lauterbach.com