

Der Pinguin und das große Blech

VON MICHAEL STÖRCHLE,
IBM DEUTSCHLAND GMBH

Zu Beginn dieses Jahrtausends ergab sich in der IT-Szene eine zunächst seltsam anmutende Liaison: Der Mainframe und Linux fanden zueinander. Dinosaurier und Pinguin, eine auf den ersten Blick eigenartige Mischung. Wir werfen einen zweiten Blick darauf.

Linux auf dem Mainframe, offiziell »Linux on IBM System z« genannt, ist die Verbindung der Stärken von Linux und des Mainframes. Es ist kein Versuch, eines von beiden neu zu erfinden. Für Linux – ebenso wie für andere Mainframe-Betriebssysteme – sind die systemimmanenten Vorteile des Mainframes wie Stabilität, Zuverlässigkeit und Skalierbarkeit ein direkter Mehrwert. Die besondere Attraktivität von Linux auf dem IBM System z zeigt sich vor allem in Verbindung mit den exzellenten Virtualisierungsmechanismen des Mainframes. Hier wird die symbiotische Beziehung zwischen Dinosaurier und Pinguin besonders klar: Hunderte Linux-Instanzen, die in virtuellen Maschinen auf einem real vorhandenen Mainframe betrieben werden. Sozusagen eine Pinguinfarm. Oder, um bei dem Begriff Dinosaurier zu bleiben, eine agile Raptorenherde.

Von Dinosauriern und Pinguinen

Linux on IBM System z gliedert sich bezüglich der Installationsmöglichkeiten nahtlos in die Reihe der anderen Mainframe-Betriebssysteme ein. Es lässt sich in einer LPAR (Logical Partition) oder unter z/VM betreiben. Die Anzahl der Linux-Instanzen ist prinzipiell unbegrenzt, wird in der Realität jedoch durch eine begrenzte Zahl der zur Verfügung stehenden LPARS beziehungsweise

durch die unter z/VM zur Verfügung stehenden Ressourcen limitiert. Generell ist der Einsatz unter z/VM empfohlen, da hier eine effizientere Virtualisierung und damit eine höhere Systemauslastung realisiert wird: Ressourcen können granularer als auf der LPAR-Ebene zwischen den einzelnen Linux-Instanzen verteilt und geteilt werden, ebenso sind die Linux-Instanzen dank zahlreicher Tools des Hypervisors leichter administrierbar. Die Hardware des Mainframes wird auch unter Linux on IBM System z im vollen Umfang genutzt. OSA-Adapter, Hipersockets, Crypto-Karte oder die IEEE-Floating-Point-Instruktion sind nur einige Beispiele hierfür. Die Architektur des IFL (Integrated Facility for Linux), der oft irreführend als »Spezialprozessor für

Linux« bezeichnet wird, ist identisch mit normalen Mainframeprozessoren und nicht speziell für Linux gebaut. Vielmehr handelt es sich um eine kostengünstige Alternative zu normalen Mainframe-Prozessoren, da – bedingt durch eine Microcode-Restriktion – nur Linux-Workload und gegebenenfalls das zugehörige z/VM abgehandelt werden kann. Dieser Effekt ist lediglich auf der Kostenseite sichtbar, weshalb man hier auch von einer »kaufmännischen Lösung« spricht.

Linux ist Linux ist Linux – dieser Grundsatz, nämlich eine möglichst identische Betriebssystemumgebung für viele Hardware-Plattformen, trifft auch auf die Mainframe-Variante zu. Linux on IBM System z ist eine vollständige Portierung von Linux auf

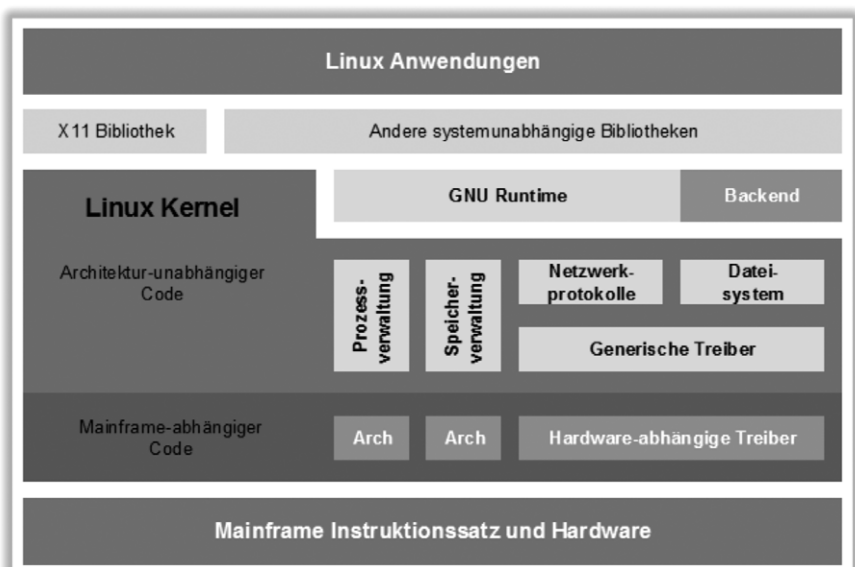


Bild 1: Aufbau des Linuxkerns (Mainframe)

die Mainframe-Hardware ohne jegliche API-Emulation. Daher basiert die komplette Betriebssystemumgebung auch auf ASCII, die sonst auf dem Mainframe übliche EBCDIC-Codepage sucht man vergebens. Ebenso gibt es keine ASCII/EBCDIC-Konversionen wie sie unter dem z/OS-Service USS (Unix System Services), einer kompletten Unix-Umgebung unter z/OS, notwendig ist. Bei der Portierung des Linuxkernel-Codes auf die Mainframe-Architektur mußte naturgemäß der Quelltext angepaßt werden. Die Unterschiede im Sourcecode von Linux on IBM System z zur IA32-Version bestehen jedoch lediglich auf der untersten Ebene der Linux-Kernelarchitektur, die direkt mit der Hardware kommuniziert. Diese Besonderheiten der Architektur werden zum Zeitpunkt der Kernel-Kompilierung eingebaut. IBM hat hierzu mainframe-eigene Kernel-»Ergänzungen« entwickelt, auf die der verwendete C-Compiler aufgrund des Arch-Flags verzweigt. Beispiele hierfür sind bestimmte Hardwaretreiber und spezielle Instruktionen. Änderungen im allgemeinen Linux-Sourcecode oder der Kernelstruktur gibt es nicht. Daher ist es auch keine Überraschung, daß diese systemspezifischen »Lines of Code« nur in etwa 4,1 Prozent des gesamten Quelltexts ausmachen. Bei den wichtigen Systemkomponenten wie gcc, glibc oder binutils sind es unter einem Prozent.

Zwangsläufig stellt sich die Frage, was die Mainframe-Variante grundlegend von der IA32-Version auf der untersten Ebene der Linux-Kernelarchitektur unterscheidet. Für Anwendungsentwickler sind diese Spezifika in der Regel von geringer Bedeutung und müssen lediglich bei hardwarenahen Anwendungen berücksichtigt werden. Aus diesem Grund ist eine echte Cross-Plattform-Entwicklung unter Linux möglich. Linux on IBM System z gibt es als 64-Bit und als 31-Bit-Versionen. Letztere Variante steht einer 32-Bit-Version auf der IA32-Architektur gegenüber. Diesen Unterschied in der Adressierbarkeit des Speichers bedingt die

Historie des Mainframe. Adressen wurden schon immer in 32-Bit-Wörtern hinterlegt, allerdings wurden bis zum Jahr 1983 lediglich 24 Bit genutzt. In diesem Jahr wurde die 31-Bit-Adressierung eingeführt. Das »verlorene« Bit, auch »most significant bit« genannt, wird als Flag dafür benutzt, ob die folgenden 24 oder 31 Bit als Adresse interpretiert werden sollen. Aufgrund des »verlorenen« Bits ergibt sich zwangsläufig ein geringerer Speicherbereich, den Linux on IBM System z (zwei Gigabyte) im Vergleich zu seinem IA32-Pendant (vier Gigabyte ohne PAE) adressieren kann. Eine weitere Folge ist, daß die Speicheradressen, in denen Executables und Libraries geladen werden, auf den beiden Architekturen unterschiedlich sind. Auch hier gilt der Grundsatz, daß diese Umstände Anwendungsentwickler in der Regel nicht berühren. Sie können aber bei der Verwendung von Pointern bedeutsam werden, ebenso falls mmap() in Verbindung mit dem Parameter MAP_FIXED genutzt wird. Momentan löst die 64-Bit-Version immer mehr 31-Bit-Installationen ab, weshalb diese Betrachtungen in der Praxis immer weiter an Bedeutung verlieren.

Anpassungen am Kernel

Ein weiterer grundlegender Unterschied besteht in der Art und Weise, wie Zahlen, die aus mehreren Byte bestehen, im Speicher abgelegt werden, der sogenannten Byte-Order. Während auf der IA32-Architektur Little Endian (das niederwertigste Byte wird an der niedrigsten Speicheradresse abgelegt) als Byte-Order verwendet wird, ist beim Mainframe Big Endian die Byte-Order. Dieser Unterschied ist für Anwendungsentwickler allerdings nur bei Pointermanipulationen relevant.

Architekturbedingte Unterschiede zwischen verschiedenen Plattformen sind oft die Knackpunkte bei Anwendungsmigrationen von der Entwicklungsplattform auf andere Hardware-Architekturen. Genau hier ist eine der großen Stärken von Linux,

denn es gestattet eine flexible Cross-Plattform-Entwicklung von Anwendungen: Die plattformspezifischen Anpassungen im Linux-Kernel sind, wie bereits aufgezeigt, marginal und für Anwendungsentwickler in der Regel irrelevant. Daher gibt es einen Codestream für mehrere Plattformen.

Der Einsatz von Linux auf einem Mainframe macht vor allem dann Sinn, wenn Anwendungen die klassischen Stärken des Mainframes benötigen: Durch Virtualisierungstechniken stehen Mechanismen zur Verfügung, die eine maximale Nutzung der vorhandenen Hardwareressourcen im Sinne einer Serverkonsolidierung gestatten. Cloning- und Verwaltungstechniken gestatten darüber hinaus einen sehr kurzfristigen Rollout bei minimalem administrativen Aufwand. Im Bereich Sicherheit ist der Mainframe unangefochten am Markt. Durch Funktionen wie Cryptohardware oder Resource Access Control Facility wird einem Linux-System auf dieser Architektur eine breite Palette an Sicherheitslösungen und -zertifizierungen angeboten, was einen Einsatz in sensiblen, unternehmenskritischen Bereichen legitimiert. Verfügbarkeit, Verlässlichkeit und Ausfallsicherheit des Systems sind zwei weitere wichtige Punkte – auch in diesem Bereich ist die Mainframe-Hardware herausragender Marktführer. Für den Produktivbetrieb ist darüber hinaus die Performance entscheidend. Die Mainframe-Hardware punktet in diesem Bereich vor allem dann, wenn es nicht nur auf reine Prozessorleistung, sondern auf ein ausgewogenes Gesamtsystem ankommt. Denn die Leistungsfähigkeit eines Systems im Produktivbetrieb ist nicht über die Taktung des Prozessors, sondern über die Antwortzeit der Anwendung definiert. ♦

Literatur

Linux on IBM System z:
<http://www-03.ibm.com/systems/z/os/linux/>

Linux for Big Iron:
<http://linuxvm.org/>