Kurz notiert: TCP Offload Engines

Entlaster

Uwe Schulze

Immer schnellere Ethernet-Schnittstellen beanspruchen die Rechenzeit der CPU von Servern und Appliances. Eine TCP Offloading genannte Technik bringt Entlastung, Hersteller implementieren sie jedoch unterschiedlich.

Der TCP/IP-Protokoll-Stack ist direkt in den Serverbetriebssystemen implementiert, und die CPU führt überwiegend die immer gleichen Algorithmen aus: Bilden des IP-Headers und Berechnen der Prüfsummen für die Datenpakete. Zwar werden moderne Multicore-CPUs leistungsfähiger, die Belastung wächst jedoch auf zwei Ebenen: erhöhtes Tempo der Ethernet-Schnittstellen und steigendes Datenvolumen. Außerdem eignen sich CPUs besser für arithmetische Operationen als für die Behandlung von Ein-/Ausgabevorgängen (I/O).

So liegt der Gedanke nahe, diese Aufgaben in die Hardware zu verlagern – ein Verfahren, das man als TCP Offload Engine (TOE) bezeichnet. Damit lassen sich höherer Datendurchsatz, geringere CPU-Belastung und geringere Latenz erzielen. Auf Ethernet-Adaptern oder Mainboards integrieren Hersteller hierfür Chips, die wesentliche Teile des TCP/IP-Protokollstandards umsetzen. Welche dies sind, hängt von der konkreten Implementierung ab. Im einfachsten Fall handelt es sich nur um das Handling der Datenpakete (Data Path Offloading oder Partial Offload), während die Verbindungssteuerung und Fehlerbehandlung im Betriebssystem verbleibt. Komplexere Umsetzungen verlagern auch diese Funktionen in die Netzhardware (Full Offload).

Neben dem Verbindungsauf- und -abbau betrifft dies vor allem die Steuerung der Paketreihenfolge und der Pufferauslastung (Flow oder Congestion Control). Funktionen, die nicht direkt mit der Übertragung von Nutzdaten in Verbindung stehen, verbleiben fast immer in der Betriebssystemsoftware (wie ARP, ICMP oder DHCP).

Viele Implementierungen beherrschen nur TCP Offload (TOE), einige nur UDP Offload (UOE) oder beides. Eine Vorstufe von TOE war das Checksum Offload, bei dem nur die Prüfsummenbildung auf die Netzkarte verlagert wurde. In aktuellen TOE-Adaptern ist diese Funktion meist einzeln aktivierbar.

TCP Offload etablierte sich in Servern mit dem zunehmenden Einsatz von Gigabit-Ethernet und dem Übergang zu 10GE. Inzwischen gibt es Implementationen für 25GE und 40GE. Produkte für 100GE befinden sich in Entwicklung. Da es sich bei TOE um herstellerspezifische Add-ons handelt, fehlt es oft auf Netzkarten der ersten Generation eines neuen Ethernet-Standards.

Schnell oder flexibel

TCP Offload Engines werden als unveränderliche anwendungsspezifische (ASIC) oder programmierbare Schaltung (FPGA) ausgeführt. Sie kommen in Servern und Netz-Appliances zum Einsatz – in Endgeräten stehen die Mehrkosten nicht in sinnvollem Verhältnis zum Tempogewinn. Router können entsprechende Techniken ebenfalls nutzen, allerdings ist dies meist nicht transparent, da Hard- und Software aus einer Hand stammen. In der Abwägung von Geschwindigkeit und Flexibilität eignen sich ASICs eher für Partial Offload, während Firmware-Lösungen Updates gestatten, was fürs Implementieren höherer Protokollschichten von Vorteil ist.

TCP Offloading ist kein Standard wie TCP/IP. Damit existiert auch keine einheitliche Softwareschnittstelle zum Betriebssystem, vielmehr muss der Hardwarehersteller entsprechende Treiber liefern. Microsoft etwa implementiert Partial Offload unter dem Namen „TCP Chimney Offload“ direkt in seinen Serverprodukten, allerdings muss die Netzkarte dazu kompatibel sein.

Wer in Support-Foren nach TCP Offloading sucht, findet vor allem die Frage, wie man es abschaltet. Insbesondere in Virtualisierungsumgebungen können einzelne TCP/IP-Funktionen zueinander inkompatibel werden. Nach dem Deaktivieren von TOE wird wieder der Standard-TCP-Stack des Betriebssystems genutzt. Teilweise ist es auch möglich, einzelne Offload-Funktionen ein- oder auszuschalten, beispielsweise Large Segment Offload (LSO) oder Large Receive Offload (LRO) – eine zentrale TOE-Technik, bei der die Umsetzung der vom Betriebssystem genutzten Datenblöcke (üblicherweise 64 KByte) in meist 1448 Byte große IP-Pakete (und umgekehrt) komplett auf der Netzkarte erfolgt.

Schwierigkeiten mit TOEs beobachtet man insbesondere beim Full Offload, denn die Protokolle der TCP-Familie werden ständig weiterentwickelt und an höheres Leitungstempo angepasst – Änderungen im Betriebssystem erfolgen in der Regel rasch, für Hardwareimplementierungen gelten jedoch längere Entwicklungszyklen. Liegen Unix-Treiber im Quelltext vor, sind Änderungen unabhängig vom Hardwarehersteller möglich.

Neben besserer Kompatibilität spricht für Partial Offload: Verbleiben die höheren Protokollschichten im Betriebssystem, können Sicherheitsrisiken genauer überwacht und Security-Funktionen rascher aktualisiert werden. Andererseits kann Full Offload die Latenz deutlich reduzieren.

Besonders zugute kommt dies der Datenübertragung in Speichernetzen (SANs). Hier treffen große Datenmengen und Anforderungen nach möglichst geringen Verzögerungen aufeinander. Hinzu kommt im Falle von iSCSI der TCP-Protokoll-Overhead. Werden nicht nur die IP-Protokolle in Hardware implementiert, sondern auch Teile von SCSI, bewerben Hersteller dies als iSCSI Offload. Ähnliches bieten sie für Fibre Channel in Form von FCoE-Offload-Adaptern an.

TCP Offloading kann entscheidende Vorteile beim Tempo bringen, etwa in einem SAN. Das Fehlen eines Standards und die unterschiedlichen Implementierungen und Weiterentwicklungen legen jedoch intensive Tests nahe, bevor man sich für ein Produkt entscheidet. (tiw)

Uwe Schulze

ist Fachautor in Berlin.

Alle Links: www.ix.de/ix1609090