

1 Ziel

Das Ziel dieser Übung ist die Verwendung der RDMA Socket API (rsocket), die den Aufwand der Portierung von herkömmlichen Anwendungen auf z. B. die IB Verbs Kommunikationsschnittstelle vermeiden kann.

Am konkreten Beispiel von InfiniBand soll der Vorteil dieser API untersucht werden. InfiniBand stellt mit dem IPOIB-ULP (Internet Protocol over InfiniBand - Upper Layer Protocol) zwar bereits eine allgemein verwendbare Schnittstelle zur Verfügung, diese hat mit der starken Abstraktion der besonderen Merkmale von InfiniBand einige Nachteile. Die Verarbeitung des IP-Stacks im Betriebssystemkern erfordert z. B. viel CPU-Zeit, die InfiniBand durch Offloading eigentlich reduzieren möchte. Die rsocket-API versucht durch eine engere Verknüpfung der RDMA-Semantik diese Nachteile zu umgehen.

An einem einfachen Beispiel bestehend aus zwei Teilen (client und server) sollen zum einen Übertragungsraten auf Basis von IPOIB untersucht und zum anderen diese Messwerte mit einer rsocket-basierten Kommunikation verglichen werden. In einem letzten Schritt soll geprüft werden, ob weitere Verbesserungen durch eine direkte Portierung auf die rsocket-API zu erreichen sind.

2 rsocket

Die RDMA Socket API (rsocket) [1] ist eine Entwicklung von Intel's Sean Hefty. Sie kann als Nachfolger des SDP-ULP (Sockets Direct Protocol) [3, 4] eingeordnet werden, welches nicht mehr weiterentwickelt wird.

Motivation für die Entwicklung war der notwendige Aufwand Socket-basierte Anwendungen auf RDMA-Semantik zu portieren. Die Komplexität wird etwa bei der Verwendung der IB Verbs-Schnittstelle offensichtlich [5].

Die Implementierung der rsocket-API ist im Quellcode der *librdmacm* [6] zu finden.

Die Header-Datei `<rdma/rsocket.h>` beinhaltet die Schnittstellendefinitionen der rsocket-API für den Fall der direkten Verwendung [2]. Unter direkter Verwendung verstehen wir in diesem Zusammenhang den Aufruf von rsocket-API-Funktionen. Außerdem steht eine dynamisch ladbare Bibliothek zur Verfügung, die den transparenten (indirekten) Zugriff auf die rsocket-Funktionalität ohne Änderung von Anwendungscode ermöglicht.

3 Beispielprogramm

Das Beispielprogramm besteht aus zwei Teilen. Der erste Teil ist eine Server-Anwendung (`server.c`), die zunächst eine TCP-Socket erstellt und konfiguriert. Dieser wartet auf eine Verbindung durch einen Klienten. Sobald die Verbindung zu diesem hergestellt ist, wird im Standardfall ein exemplarisches Datenvolumen von 1 Gibibyte gesendet. Im Anschluss wird die Verbindung beendet. Die Anwendung wartet danach auf die nächste Verbindung mit einem Klienten.

Der zweite Teil besteht aus der Client-Anwendung (`client.c`), die eine TCP-Verbindung an die als Kommandozeilenparameter angegebene IP-Adresse und die fest-kodierten Portnummer aufbaut. Danach wird analog versucht ein Datenvolumen von 1 Gibibyte zu empfangen. Die dafür benötigte Zeit wird gemessen und die daraus ermittelte Datenübertragungsrate ausgegeben. Danach beendet sich dieser Klient.

4 Aufgaben

1. Übersetzen Sie die Quellcodes `client.c` und `server.c` mit Hilfe des `Makefile`. Bestimmen Sie die vorhandenen (IP-)Schnittstellen Ihres Testsystems durch das Linux-Kommando `ip addr`. Führen Sie `server` und `client` mit geeigneten Parametern aus und notieren Sie die Ausgaben/Messwerte von mehreren Ausführungen. Beachten Sie zusätzlich die verwendete Systemzeit (`time`-Werkzeug)!
2. Führen Sie Messungen unter Zuhilfenahme der `rsocket`-Bibliothek durch ohne die Programme zu ändern! Erfassen Sie mehrere Messwerte.
3. Kopieren Sie die Vorgabedateien und modifizieren Sie den Quellcode für die direkte Nutzung der `rsocket`-Bibliothek. Führen Sie die geänderten und übersetzten Programme aus um mehrere Messwerte zu erhalten.
4. *Zusatzaufgabe:* Untersuchen Sie den Einfluss der übertragenen Menge an Daten. Erstellen Sie ein Diagramm zu den Übertragungsraten von `IPoIB` und `rsockets`.

5 Auswertung

1. Welche Hardware-seitigen und Software-seitigen Voraussetzungen sind für `rsockets` zu erfüllen?
2. Welche Übertragungsraten haben Sie erwartet und welche haben Sie gemessen?
3. Welchen Vorteil bietet die direkte Nutzung der `rsocket`-API in einer Anwendung?

Literatur und wichtige Links

- [1] *Sean Hefty: RSOCKETS*
<http://downloads.openfabrics.org/downloads/rdmacm/rsockets-ofa12.pptx>
http://downloads.openfabrics.org/Media/Monterey_2012/2012_Workshop_Mon_Rsockets.pdf
- [2] *Manpage rsocket librdmacm Programmer's Manual* (`man rsocket`)
- [3] *RDMA Consortium: Sockets Direct Protocol (SDP) for iWARP over TCP (v1.0)*
<http://www.rdmacconsortium.org/home/draft-pinkerton-iwarp-sdp-v1.0.pdf>
- [4] *InfiniBand Trade Association: InfiniBand Architecture Release v1.2.1 Volume 1, ANNEX A4: Sockets Direct Protocol (SDP)*
<https://cw.infinibandta.org/document/dl/7143>
- [5] *RDMA Aware Networks Programming User Manual*
http://www.mellanox.com/related-docs/prod_software/RDMA_Aware_Programming_user_manual.pdf
- [6] *General RDMA communication manager (librdmacm)*
<http://www.openfabrics.org/downloads/rdmacm/>