

# Parallelisierung eines NVIDIA GPGPU Client Server Services basierend auf TCP-Sockets

Til Koke

Institut für Betriebssysteme und

Rechnernetze

TU Braunschweig

Mühlenpfordstr.23, 38106 Braunschweig

Email: til.koke@tu-bs.de

Maximilian Wesche

Institut für Betriebssysteme und

Rechnernetze

TU Braunschweig

Mühlenpfordstr.23, 38106 Braunschweig

Email: maximilian.wesche@tu-bs.de

David Winterland

Institut für Betriebssysteme und

Rechnernetze

TU Braunschweig

Mühlenpfordstr.23, 38106 Braunschweig

Email: david.winterland@tu-bs.de

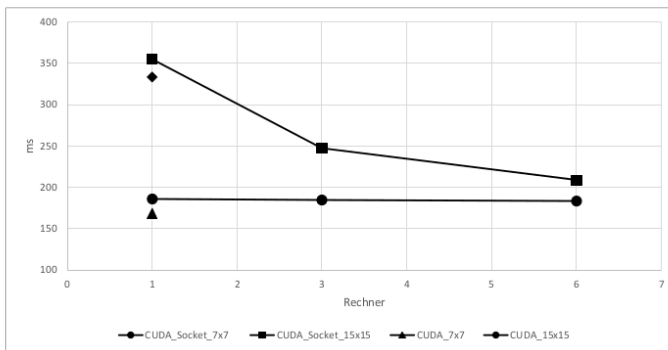


Fig. 1. Vergleich der Socket Implementation gegenüber dem CUDA-Algorithmus auf einer Grafikkarte.

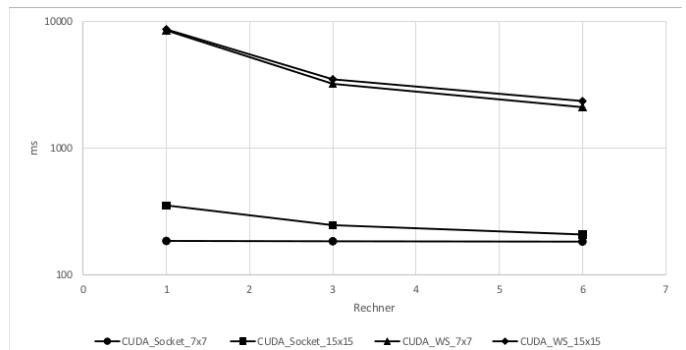


Fig. 2. Vergleich der Socket Implementation gegenüber der Webservice Variante.

**Abstract**—The abstract goes here.

## I. EINLEITUNG

Algorithmus mit CUDA SOAP hat Overhead Motiv für C-Sockets: weniger Overhead (Performance)-Vergleich mit aktuellem Soap Ansatz

November 24, 2018

## II. VERWANDTE ARBEITEN

## III. ANSATZ

Implementiert mit nativen C-Client-Server Sockets nachher Rechnerverbund um Aufgabe verteilt berechnen zu können.

### A. Unterkapitel xy

Subsection text here.

## IV. IMPLEMENTIERUNG & VALIDIERUNG

HW Beschreibung: 1080GTX Titan, x Anzahl an Shader TCP Stream Sockets, Datenstruktur Beschreibung, welche Daten werden gesendet?

## V. ZUSAMMENFASSUNG UND WEITERE ARBEITEN

Evaluation welches Ergebnis schneller war und warum. TCP Sockets aktuell verwendet, man könnte es auch mit UDP Sockets (Datagram Sockets) machen. Oder eine andere Middleware bspw. JAVA RMI

## REFERENCES

- [1] H. Kopka and P. W. Daly, *A Guide to L<sup>A</sup>T<sub>E</sub>X*, 3rd ed. Harlow, England: Addison-Wesley, 1999.