TECHNISCHE UNIVERSITÄT DRESDEN

ANALYSE DER LATENZEN IM KOMMUNIKATIONSSTACK EINES PCIE-GEKOPPELTEN FPGA-BESCHLEUNIGERS

Sascha Kath

Dresden, 09.03.2017

DRESDEN concept
Exzellenz aus Wissenschaft und Kultur

Gliederung

1. Motivation & Zielstellung
2. Systembeschreibung
3. Implementierung und Messungen
4. Auswertung
5. Zusammenfassung & Ausblick

1. MOTIVATION & ZIELSTELLUNG

Motivation

- Beschleunigung von Anwendungen
  - Parallelisierbare Algorithmen auf Hardwarebeschleuniger ausführen
  - Ausführungszeit und Energiebedarf verringern → Effizientere Berechnung

  → Ausreichend schnelle Übertragungsraten nötig

Zielstellung

- Kommunikationsstack analysieren

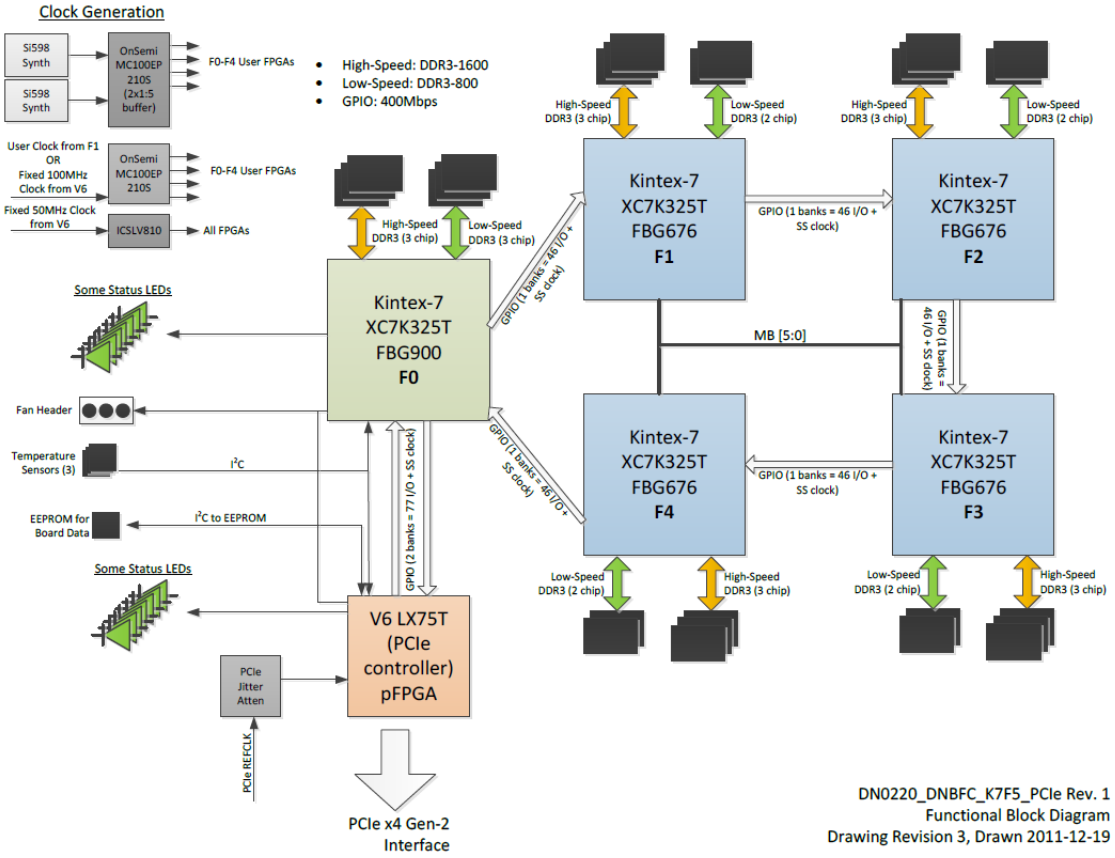
→ Flaschenhals identifizieren

- Möglichkeiten analysieren, um möglichen Flaschenhals zu umgehen

→ Bandbreite besser ausnutzen

2. SYSTEMBESCHREIBUNG

FPGA Board - Übersicht



[1]

Kintex 7K325T – Übersicht in Zahlen

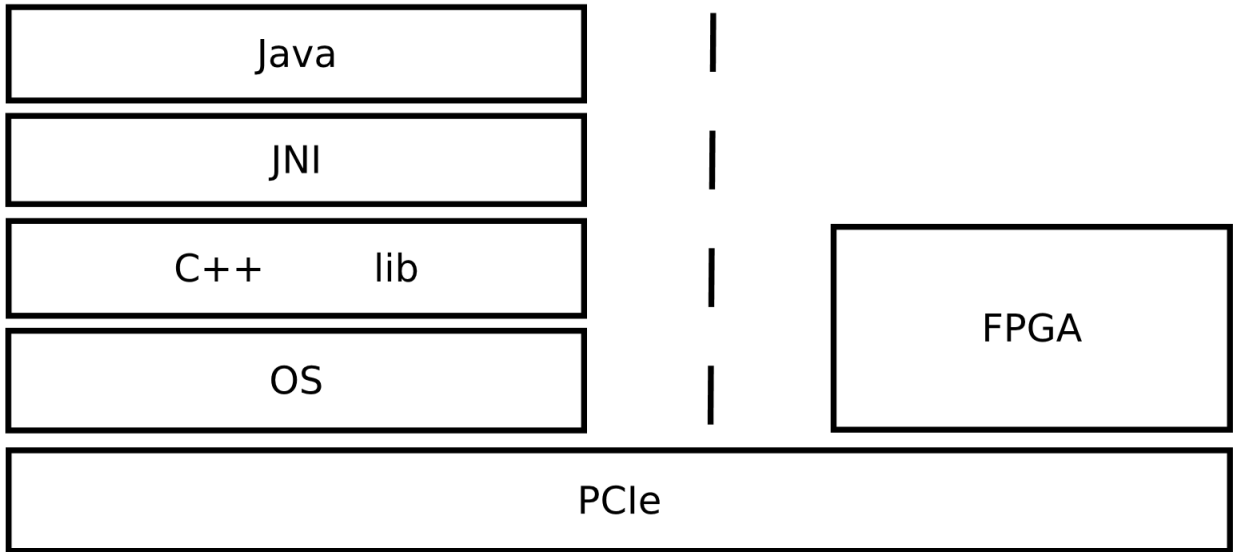
Block RAM	Total Block RAM	PCIe	Lanes	Max. Bandbreite
445 x 36 Kbit	16 020 Kbit	Gen 2	4	1 GB/s

$$MaxBandwidth = \frac{5Gb/s * LaneWidth}{10bit/Byte}$$

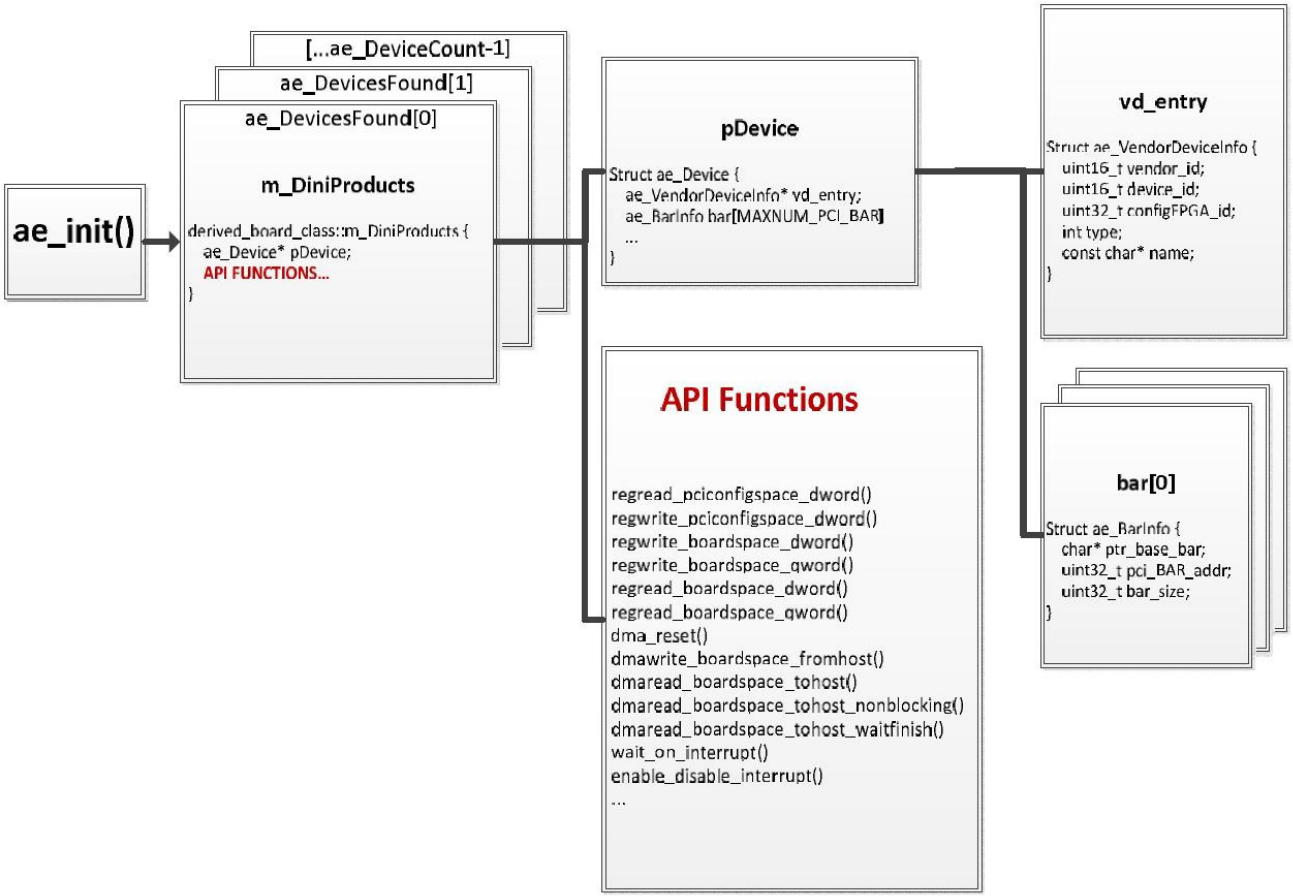
- Gen 2: 5 Gbit/s
- Je 2 Lanes für Hin- und Rückrichtung

→ LaneWidth = 2

Kommunikationsstack

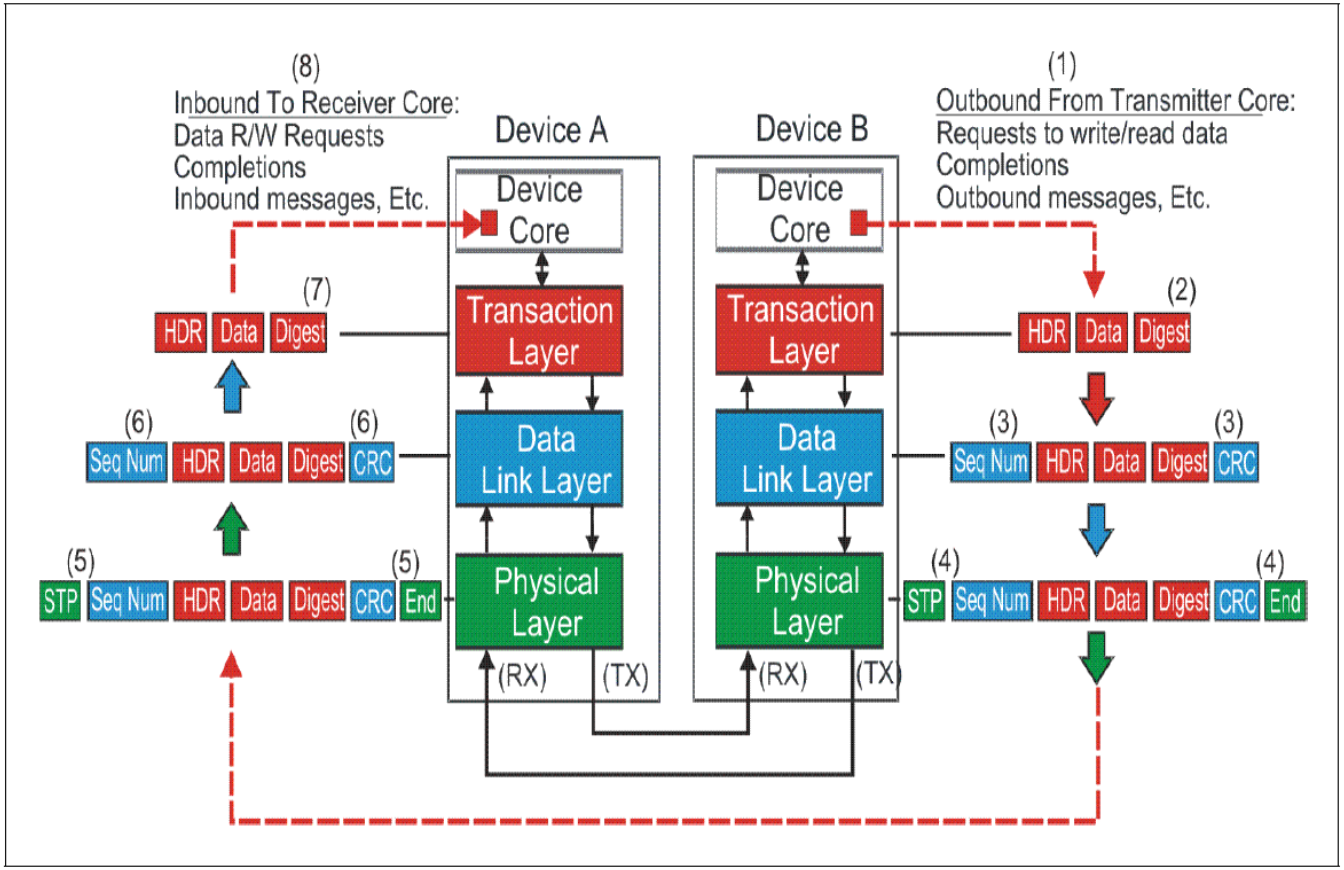


AETest Software



[4]

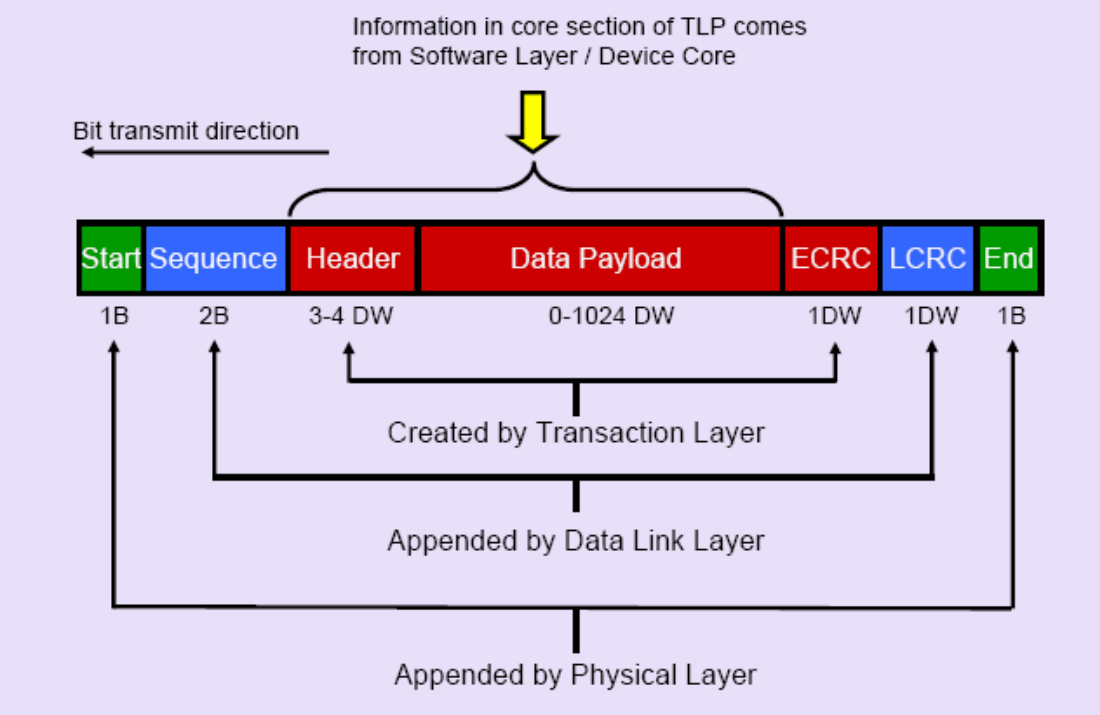
PCIe OSI Schichten



[2]

Paket Overhead

- 256 Byte Payload mit 24 Byte Overhead
- Maximale Effizienz von 91.4 %



[3]

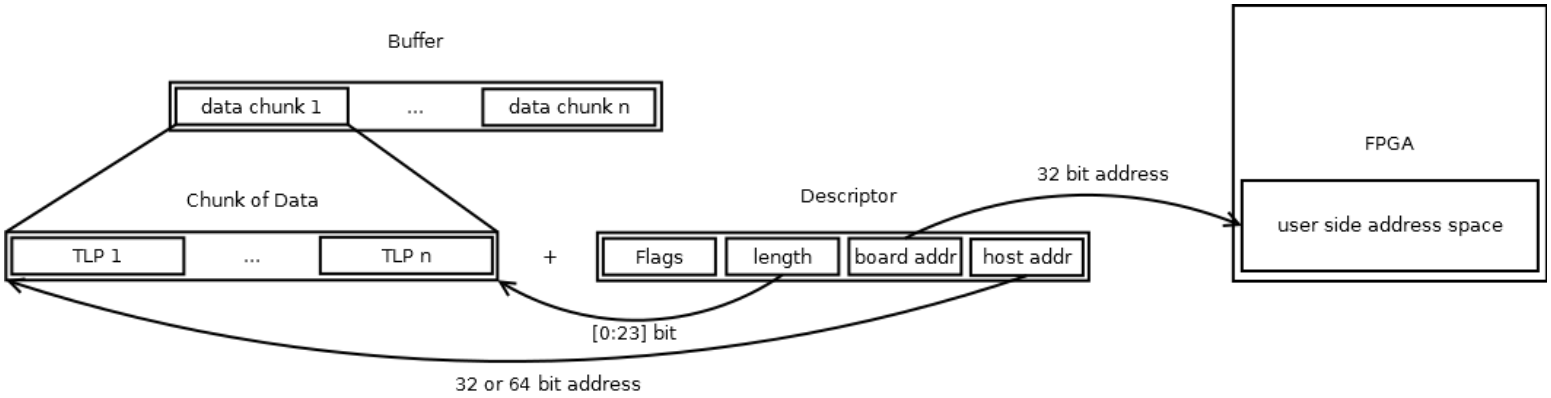
PCIe - Deskriptoren

- Deskriptor:
  - Beschreibt Daten-Chunk
  - Dword Länge: (Anzahl der 32-Bit-Worte)
  - Board Adress: Byte-Adresse des Nutzer-Adressraums auf dem FPGA-Board
  - PCI Adresse: physische Adresse im Host-Memory

31	30	29	28	27	26	25	24	23 ... 2	1	0	
V	D	I	HA	RSVD		O	C	DWord Length [23:0]			0
Board Address [31:0]											1
PCI Address [31:2]									0	0	2
PCI Address [63:32] - HA											3

PCIe - Deskriptoren

- Adresse in Deskriptor muss auf kontinuierlichen Speicherbereich zeigen
- 64-Bit-Wort Alignment wichtig



3. IMPLEMENTIERUNG UND MESSUNGEN

Hardware-Design

- FPGA-Aufteilung: 4 Chips mit je 4 Parts
  - 445 Block RAMs je FPGA / 16

  → 27 Block RAMs / Part
  - Jeder Part hat Ein- und Ausgabepuffer

  → 13 Block RAMs / Puffer * 4,5 KB= 58,5 KB

Hardware-Design

- 2 Varianten für Messungen verwendet:

1. Gegebene Implementierung mit 4-KiByte-Puffern

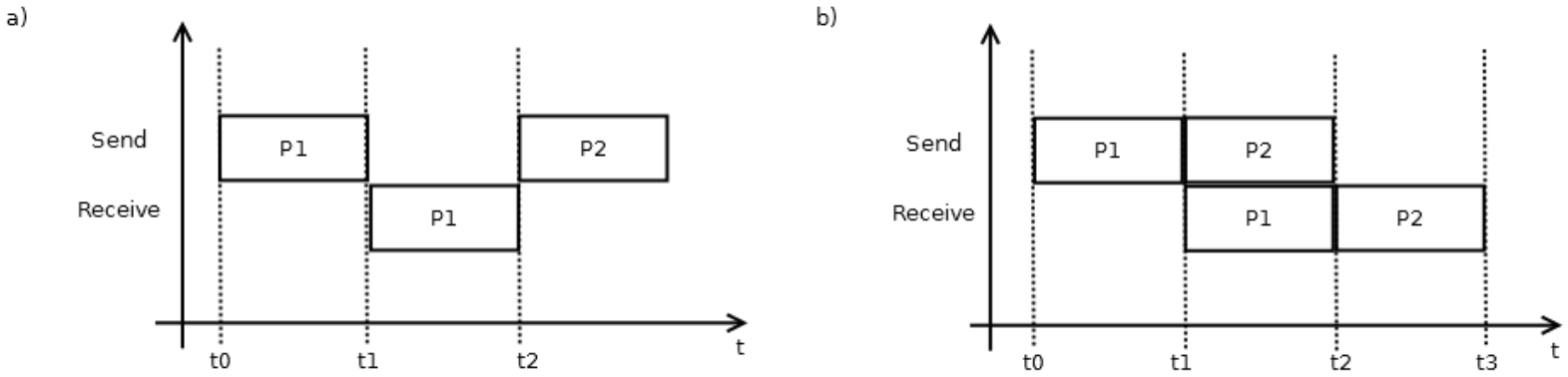
2. Verändertes Design mit 72-KiByte-Puffern:
   - Nutzung von LUT RAMs
   - Senden von 64-KiByte-Daten-Chunks möglich

Software

- Java DmaLoop:
  - schreibt Sequenznummer, liest diese zurück und vergleicht
  - 3 Varianten:
    1. Synchrones Senden
    2. Asynchrones Senden über synchronen Kanal
    3. Asynchrones Senden über asynchronen Kanal
- Java DmaEngine: mit und ohne Flusskontrolle

Synchrones vs. Asynchrones Senden

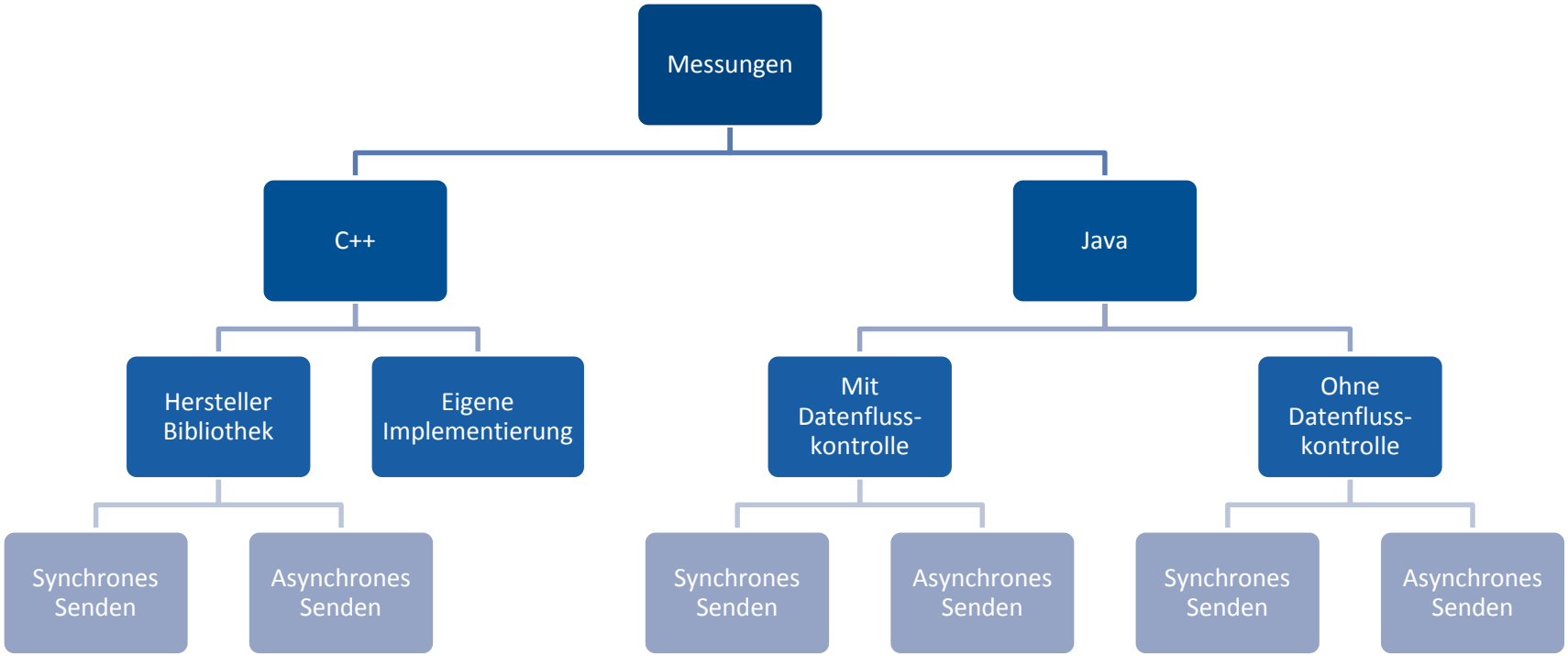
- Höhere Bandbreite durch paralleles Senden



Messungen

- Vorgehen:
  1. Messungen in C++ (Vergleichswerte / praktische Möglichkeiten)
  2. Messungen in Java
  3. Quelle der Zeitdifferenzen finden
  4. Parameteranpassungen, um Bandbreite besser auszunutzen

Messungen


Messungen
C++
Java
Hersteller Bibliothek
Eigene Implementierung
Mit Datenfluss-kontrolle
Ohne Datenfluss-kontrolle
Synchrones Senden
Asynchrones Senden
Synchrones Senden
Asynchrones Senden
Synchrones Senden
Asynchrones Senden

4. AUSWERTUNG

C++

	Asynchrones Senden (t_{CPU})	Synchrones Senden (t_{CPU})	Synchrones Senden (t_{WCT})	DmaEngine (t_{WCT})
Bandbreite	980 MB/s	430 MB/s	100 MB/s	80 MB/s

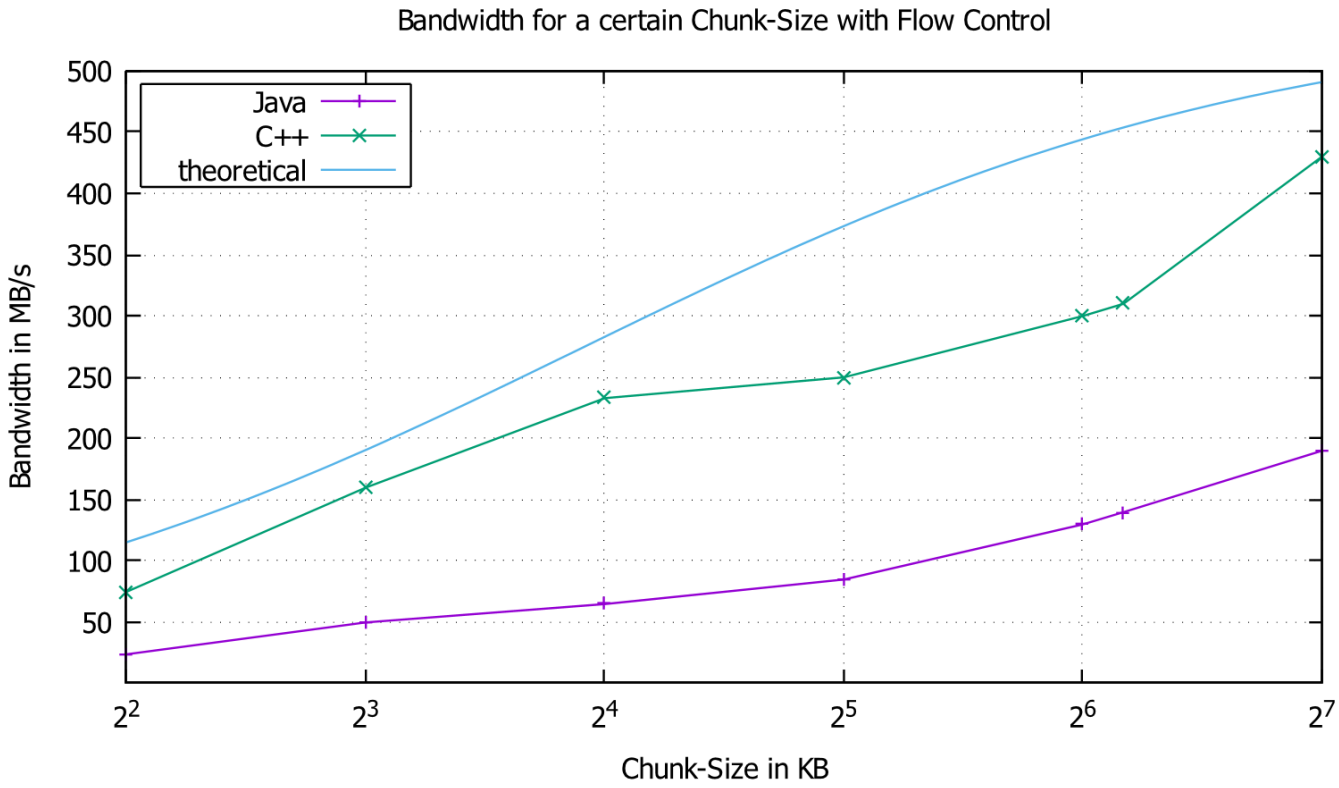
- Aetest-Funktionen nutzen CPU-Zeit (t_{CPU})
- Aber Wall-Clock-Time (t_{WCT}) interessant für Applikation
- Eigene DmaEngine nutzt Funktionen die const char* zurückgeben

Java mit Flusskontrolle (je 72-KiByte-Puffer)

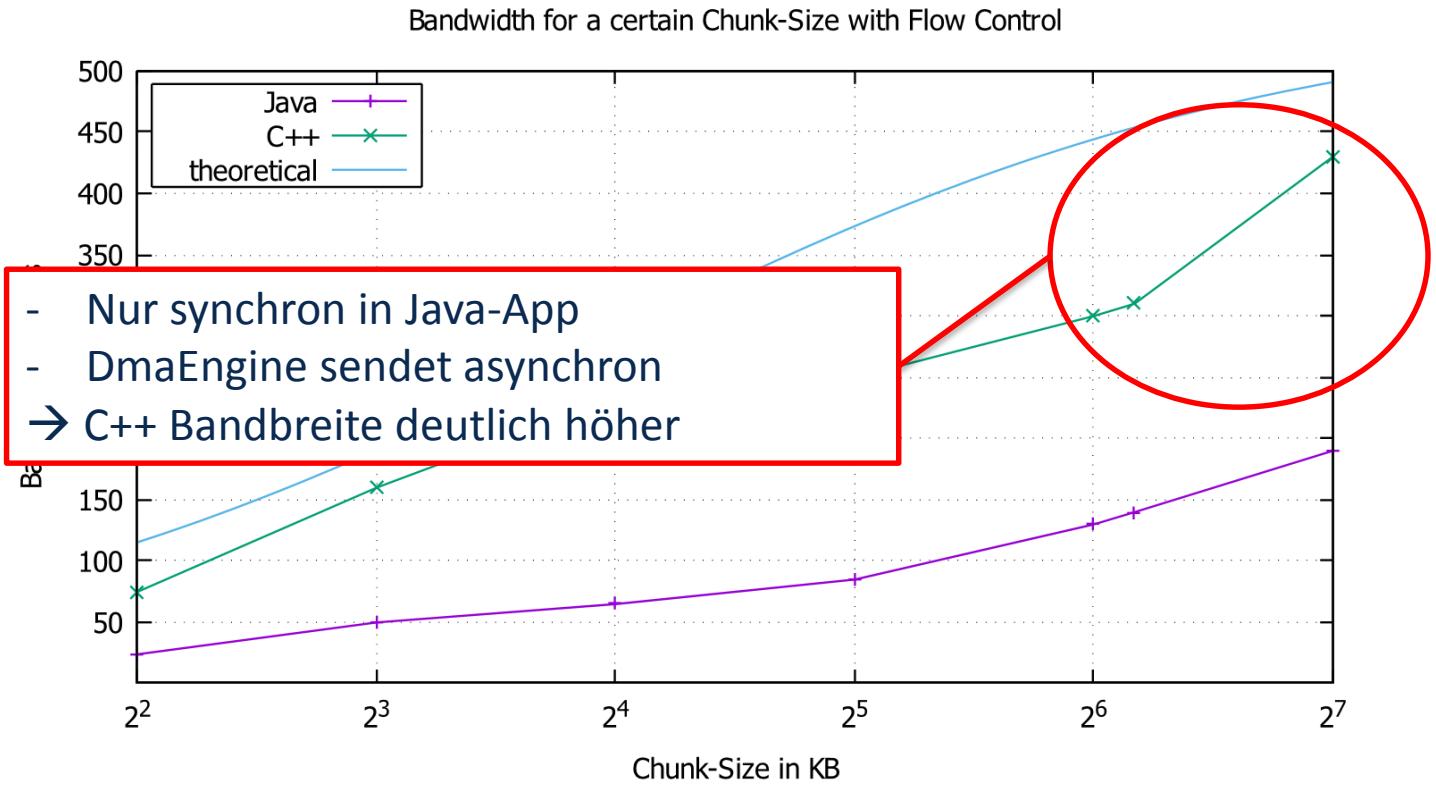
Buffer-Size	Synchrones Senden	Asynchron (ByteChannel)	Asynchron (Asynchronous ByteChannel)
4 KB	25 MB/s	42 MB/s	46 MB/s
8 KB	50 MB/s	70 MB/s	72 MB/s
16 KB	65 MB/s	*	*
32 KB	85 MB/s	*	*
64 KB	130 MB/s	*	*
128 KB	190 MB/s	*	*

* Momentan maximal 8 KB Chunk-Sizes unterstützt

Synchrones Senden mit Flusskontrolle



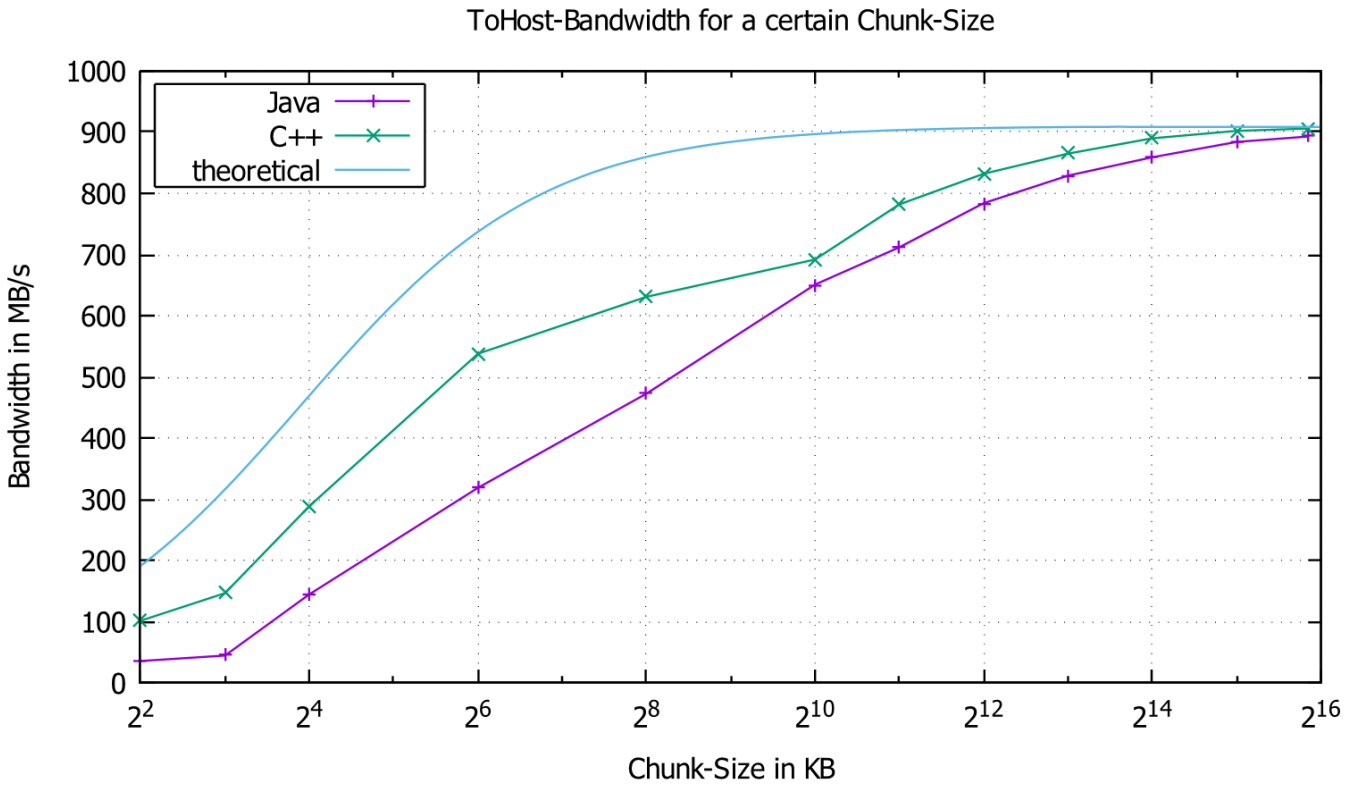
Synchrones Senden mit Flusskontrolle



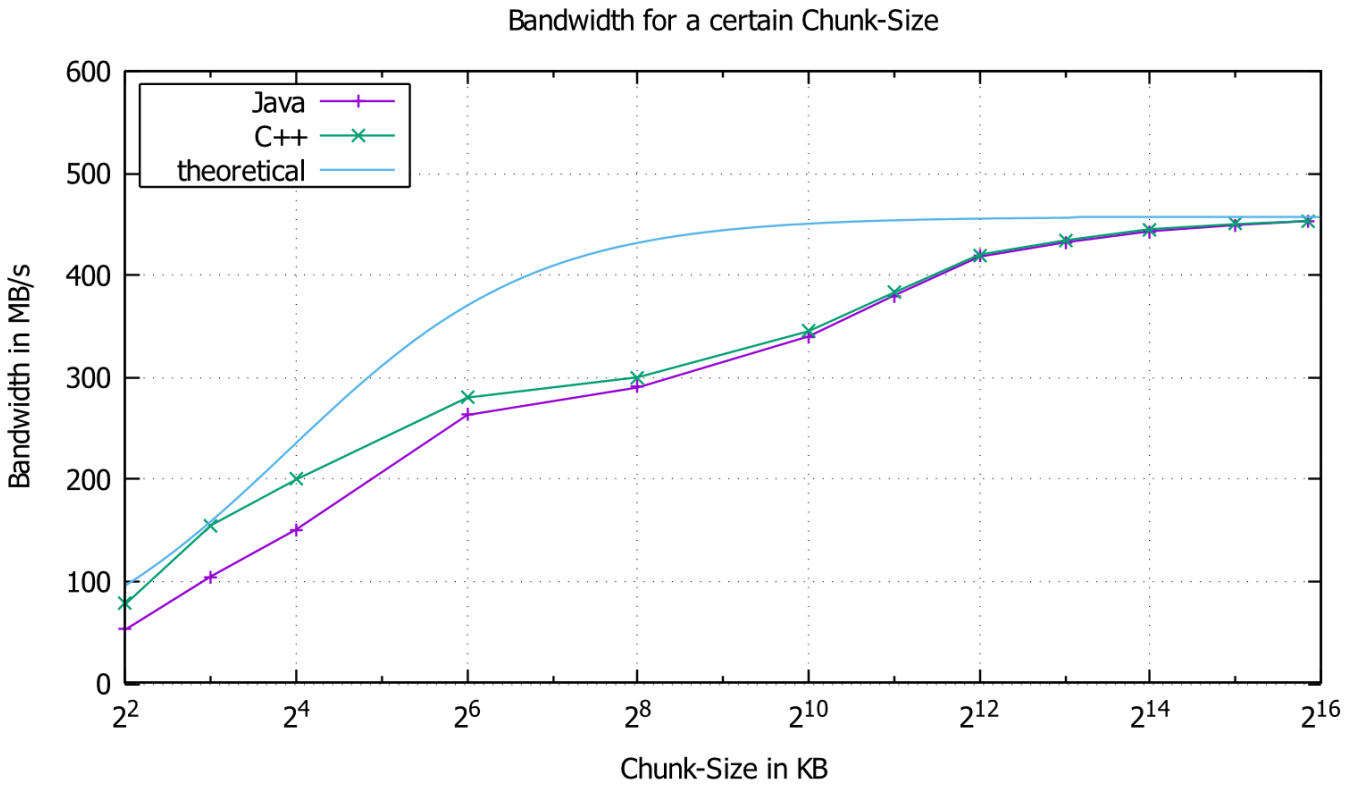
Zwischenfazit

- DMA Initialisierung erzeugt hohe Latenz
  - Größere Chunk-Size = höhere Bandbreite
- Asynchrones Senden erwartungsgemäß schneller, aber zu kleine Chunk-Sizes
- Ab jetzt Untersuchung der Möglichkeiten
  - Ohne Flusskontrolle
  - Buffer-Size = Chunk-Size
  - Physischer Speicher ungenügend → Richtige Datenmenge, aber falsche Daten

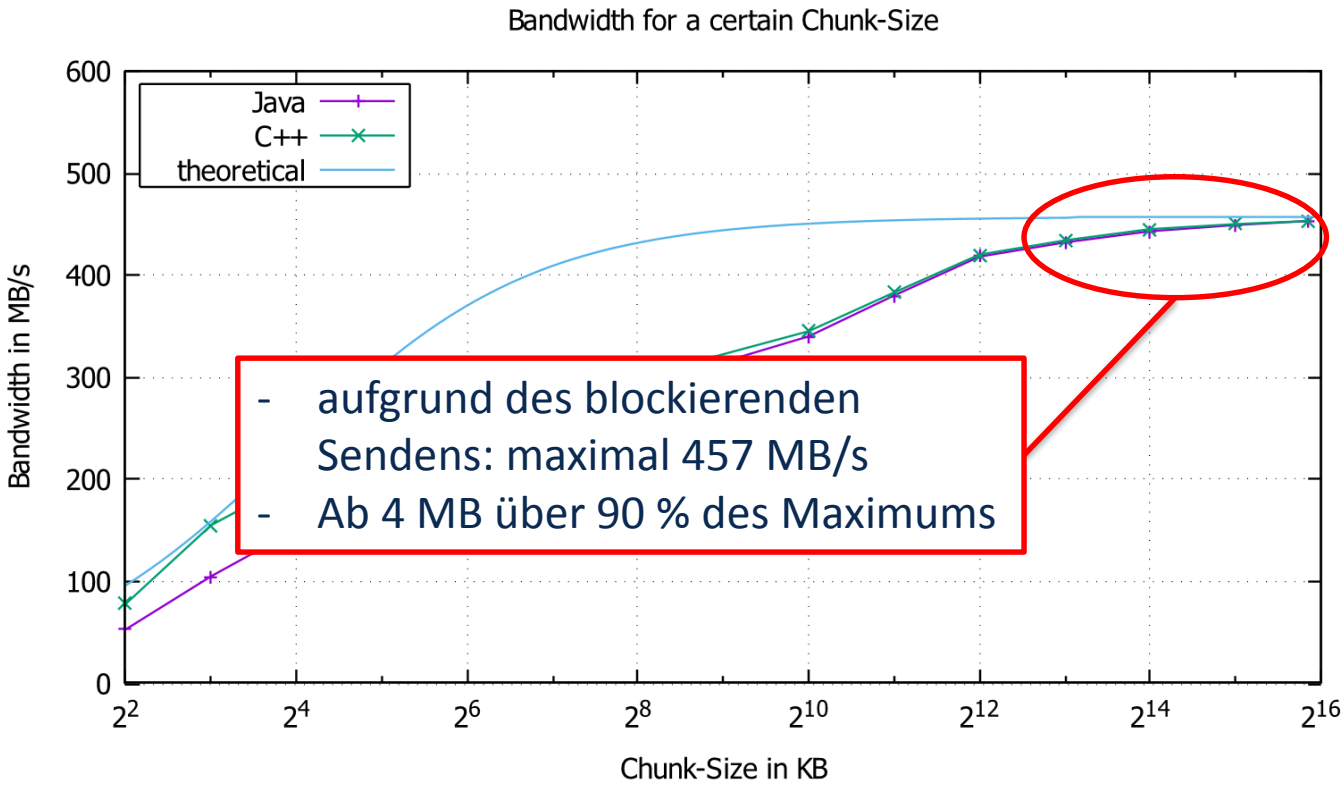
Asynchrones Senden ohne Flusskontrolle



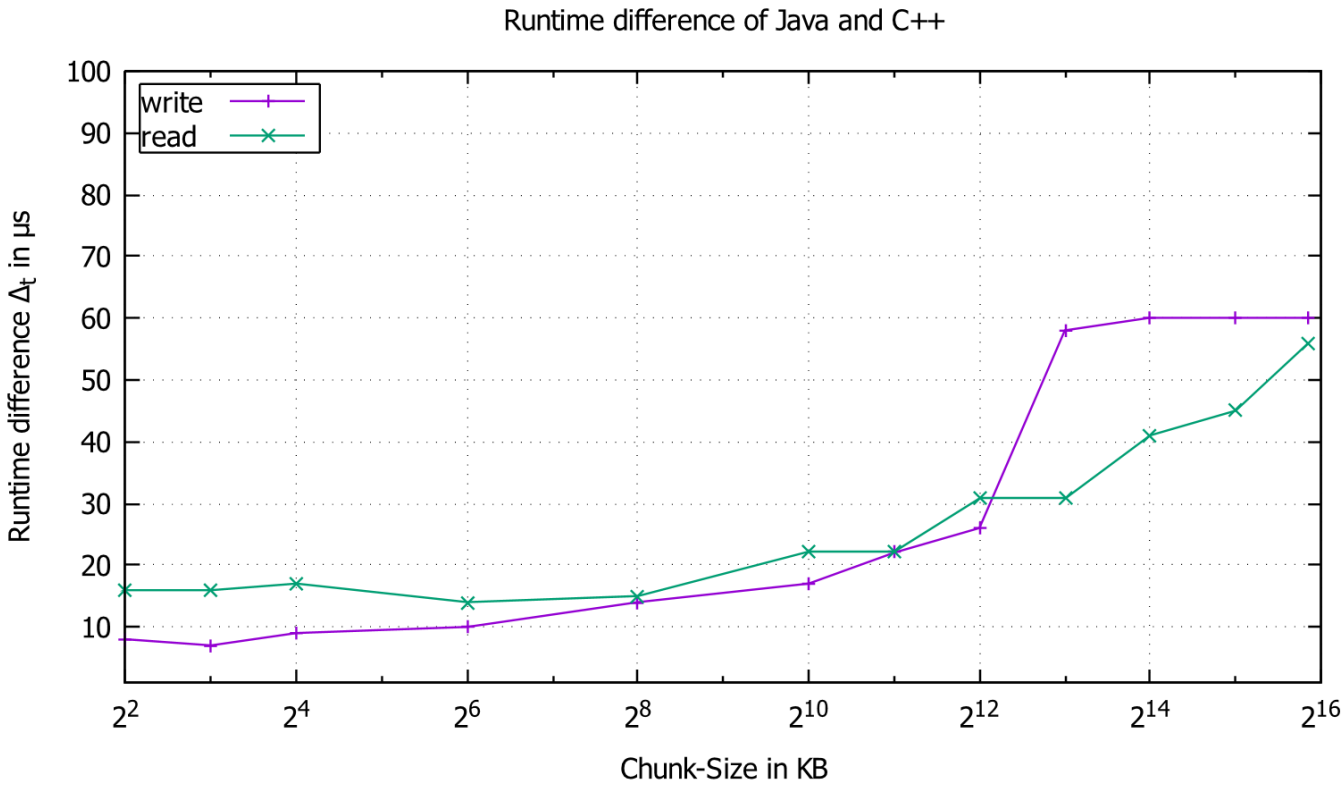
Synchrones Senden ohne Flusskontrolle



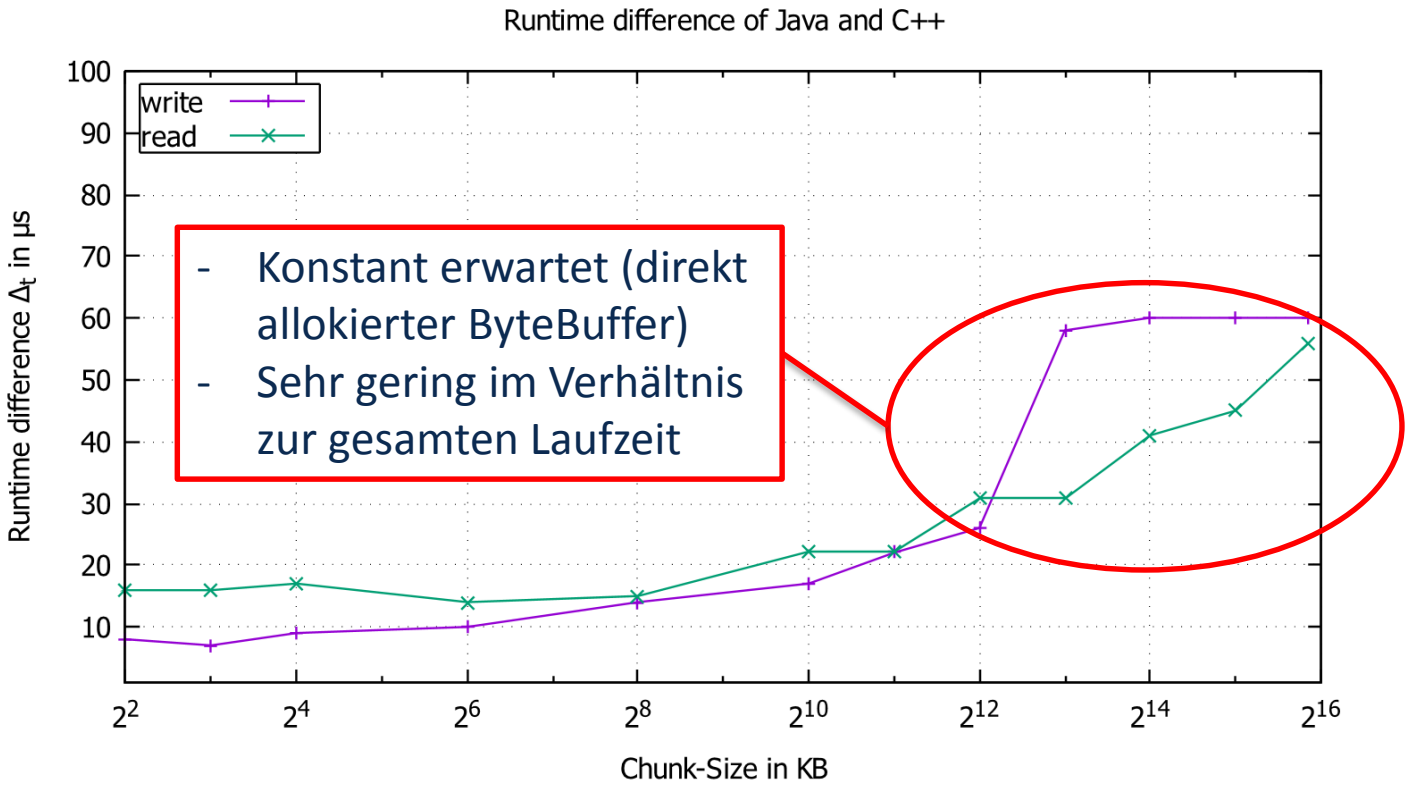
Synchrones Senden ohne Flusskontrolle



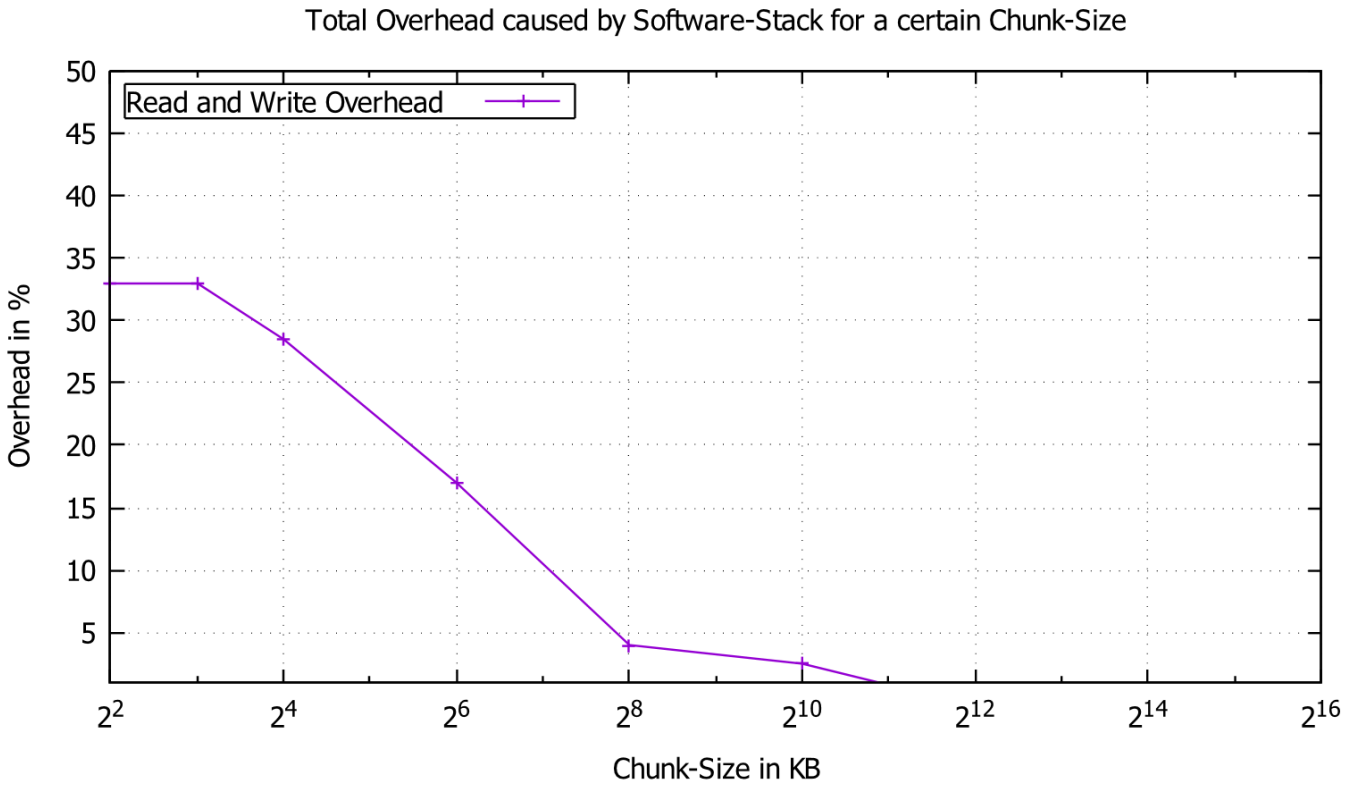
Laufzeitdifferenz – Java vs. C++



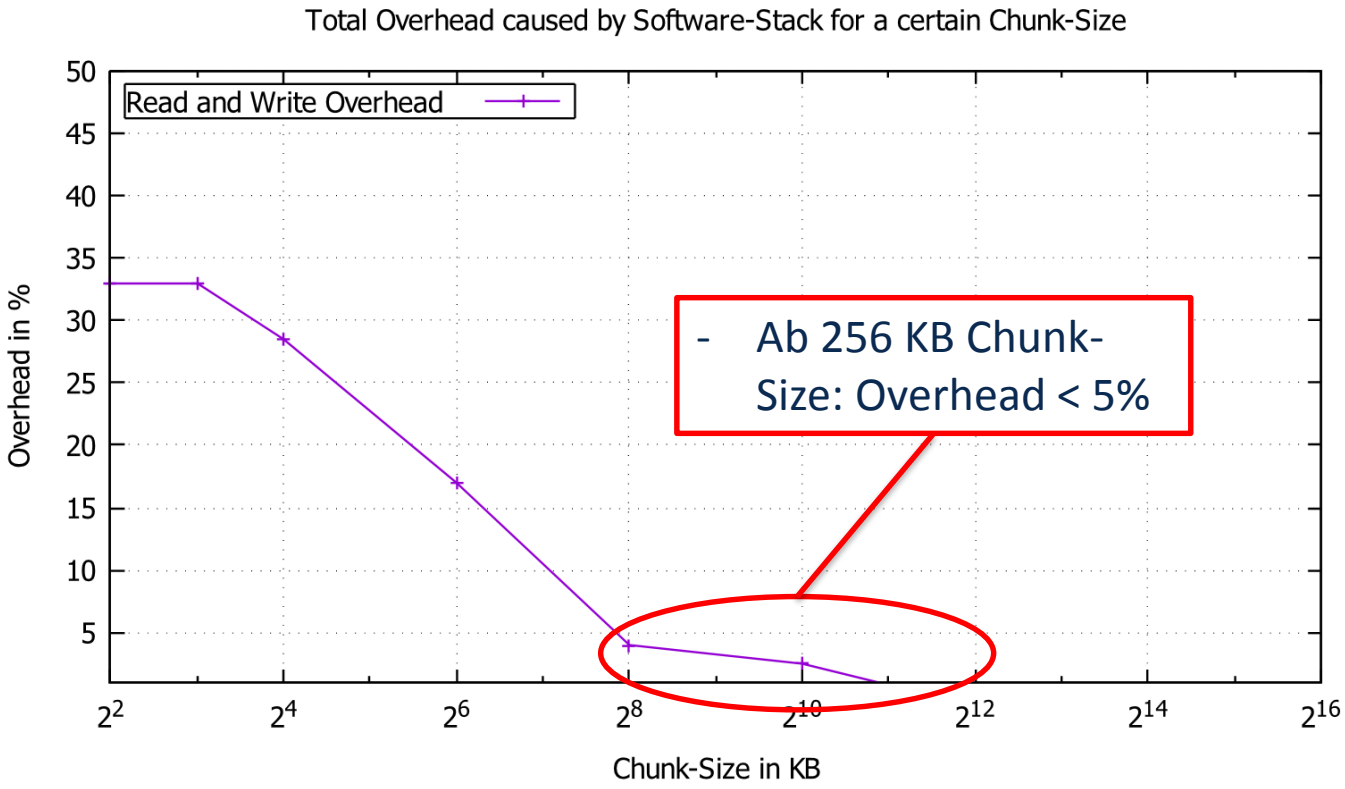
Laufzeitdifferenz – Java vs. C++



Aufruf-Overhead im Kommunikationsstack



Aufruf-Overhead im Kommunikationsstack



5. ZUSAMMENFASSUNG & AUSBLICK

5. Zusammenfassung & Ausblick

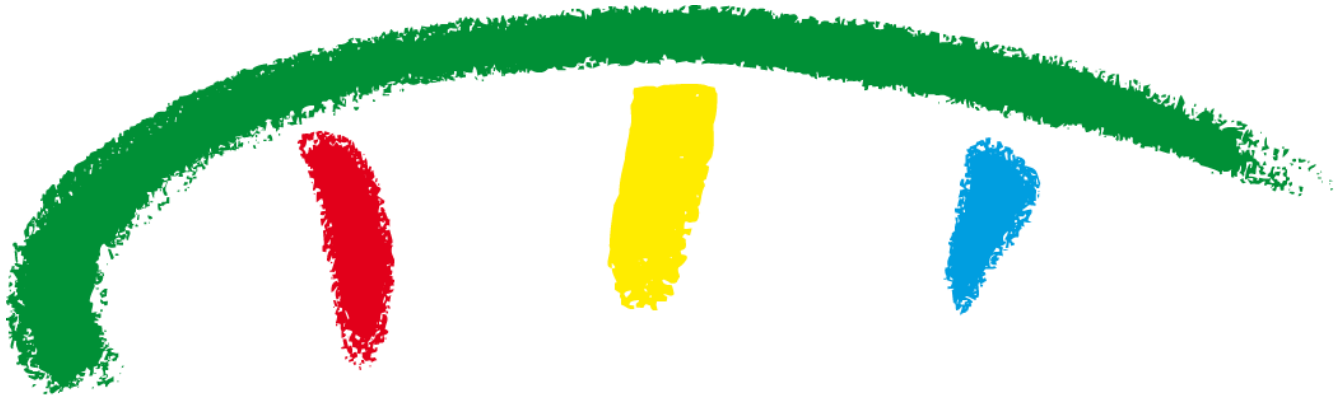
- 4 PCIe-Lanes (max. 1 GB/s pro Richtung)
- Konstanter Paket-Overhead
- Aufteilung in Daten-Chunks
  - Deskriptoren zur Beschreibung
  - Nach Empfang des Deskriptors DMA
  - Größere Chunks = geringere Latenz = höhere Bandbreite

5. Zusammenfassung & Ausblick

- Synchrones Senden
  - Bandbreite maximal 457 MB/s
  - Ab 4-MB-Chunk-Size über 90% des Maximums, aber unter theoretischen Möglichkeiten:
    - Bessere Ausnutzung der Bandbreite durch paralleles (asynchrones) Senden: z.Z. nur maximal 8-KB-Chunks unterstützt (auf Java-Ebene)

5. Zusammenfassung & Ausblick

- Puffern mehrerer großer Chunks:
  - 256 KB: Bandbreite > 50%
  - Größere Puffer nötig → externer Speicher
  - 3 x 4 GB DDR3-1600 verfügbar



»Wissen schafft Brücken.«

Quellen

[1] DINI Group. *DNK7_F5_PCIe Hardware Manual*. 2014. URL: http://www.dinigroup.com/product/data/DNK7_F5PCIe/files/Hardware_Manual_DNK7_F5_PCIe_REV4.pdf (besucht am 27. 02. 2017).

[2] Ravi Budruk, Don Anderson und Tom Shanley. *PCI express system architecture*. Addison-Wesley Professional, 2004.

[3] Verien Design Group, LLC. *PCIe Packet Encapsulation*. 2012. URL: http://www.verien.com/picts/pcie_packet_encapsulation.png (besucht am 27. 02. 2017).

[4] DINI Group. *AETest Software Manual*. 2017. URL: http://www.dinigroup.com/files/web_packs/Aetest.zip (besucht am 27. 02. 2017).