

Result Set Serialization

Ferdinand List

Technische Universität München

Fakultät für Informatik

Lehrstuhl für Datenbanksysteme

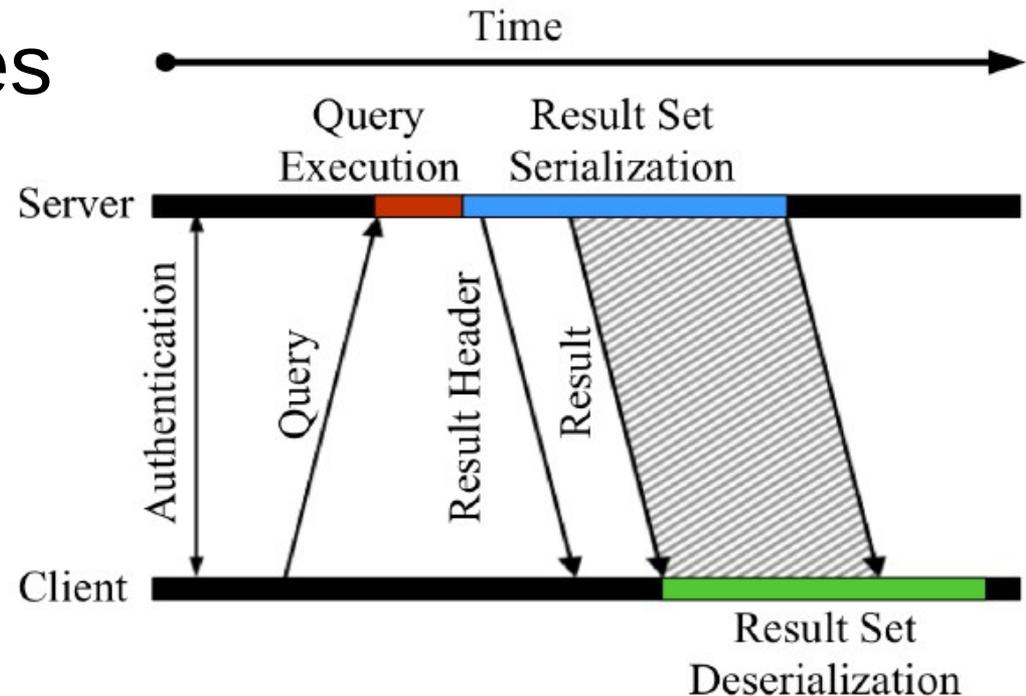
10. Dezember 2018



Uhrenturm der TUM

Result Set Serialization - Idee

- Serialisierung des Results zur Übertragung



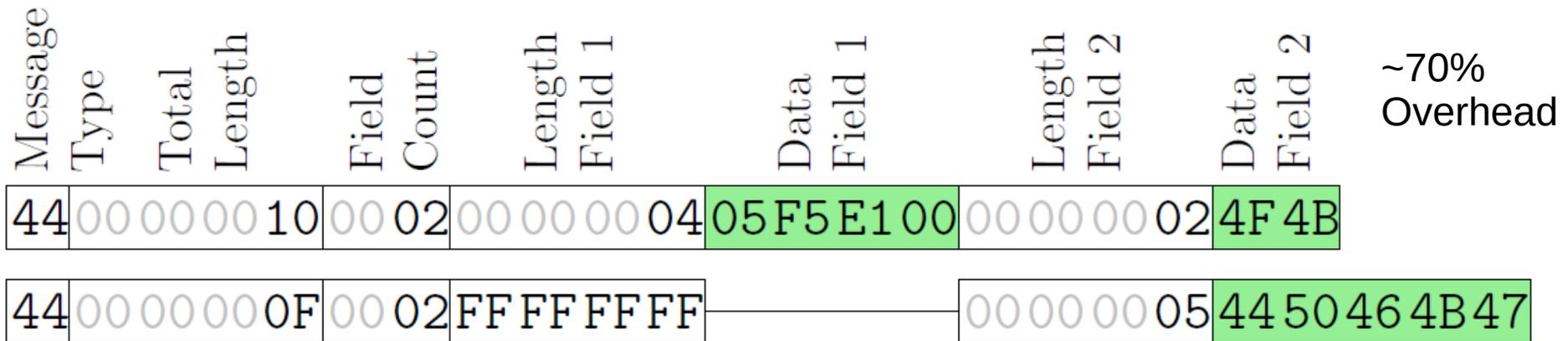
Nachteile aktueller Systeme

- $\frac{3}{4}$ für Kompression verwendet
=> rentabel bei langsamer Netzwerkanbindung oder großer Datenmenge
- schlechte Protokolle in allen gängigen Datenbanksystemen

System	Zeit(s)	Größe(GB)
Netcat (CSV)	10.25	7.19
MySQL	101.25	7.44
MySQL + GZIP	391.27	2.85
MonetDB	209.02	8.97

Abfrage der LineItem Tabelle
(Werte aus [1])

Aktuelles Übertragungsformat von Postgres



INTEGER32	VARCHAR10
100.000.000	OK
NULL	DPFKG

[1]:Raasveldt et al.

Protokoll

- Datenbanksysteme häufig Zeilenbasiert
 - Komprimierung schwierig
- => Übertragung Spaltenbasiert

Chunks

	Chunksize	Rows	Time	Size (GB)	C. Ratio
Lineitem	2KB	1.4×10^1	55.9	6.56	1.38
	10KB	7.1×10^1	15.2	5.92	1.80
	100KB	7.1×10^2	10.9	5.81	2.12
	1MB	7.1×10^3	10.0	5.80	2.25
	10MB	7.1×10^4	10.9	5.80	2.26
	100MB	7.1×10^5	13.3	6.15	2.23
ACS	2KB	1.0×10^0	281.1	11.36	2.06
	10KB	8.0×10^0	46.7	9.72	3.18
	100KB	8.5×10^1	16.2	9.50	3.68
	1MB	8.5×10^2	11.9	9.49	3.81
	10MB	8.5×10^3	15.3	9.50	3.86
	100MB	8.5×10^4	17.9	10.05	3.84
Ontime	2KB	1.0×10^0	162.9	8.70	2.13
	10KB	8.0×10^0	27.3	4.10	4.15
	100KB	8.5×10^1	7.6	3.47	8.15
	1MB	8.6×10^2	6.9	3.42	9.80
	10MB	8.6×10^3	6.2	3.42	10.24
	100MB	8.6×10^4	11.9	3.60	10.84

Protokoll - Komprimierung

- Komprimierung
wichtig bei
langsamer
Übertragungsrate

	Lokal (s)	1 Gbit/s (s)	10 Mbit/s (s)	MB
Ohne	1.5	10.4	848	1012
Snappy	3.3	3.8	373	447
GZIP	59.8	60.4	226	272

Abfrage Teil der Lineltem Tabelle (Werte aus [1])

Protokoll - Komprimierung

	Lokal	LAN	WAN
None	1.3	5.8	54.4
Binpack	1.4	6.2	44.4
PFOR	1.6	5.8	44.4
Snappy	1.4	1.4	3.2
Binpack+Sy	1.8	1.9	5.7
PFOR+Sy	1.8	1.9	5.9

OnTime

	Lokal	LAN	WAN
None	15.2	78.6	800.6
Binpack	120.5	133.9	421.2
PFOR	166.8	170.1	300.9
Snappy	20.5	22.8	204.5
Binpack+Sy	152.6	160.9	190
PFOR+Sy	165,8	168.4	185.4

ACS

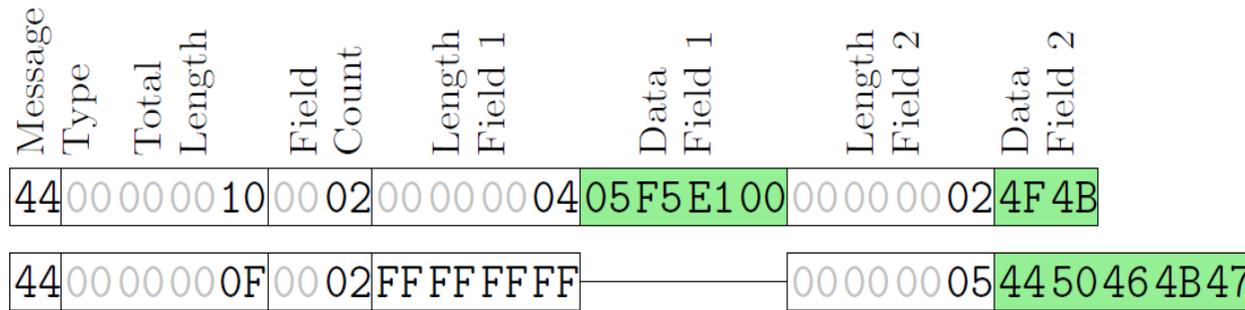
Abfrage Integer Spalten (Werte aus [1])

Protokoll - Stringbehandlung

- Null – Termination => Null-Byte am Ende jedes Strings
- Length-Prefixing => Vorausgehend Länge des kommenden Strings
- Fixed Width => Stringrepräsentation hat immer fixe Länge

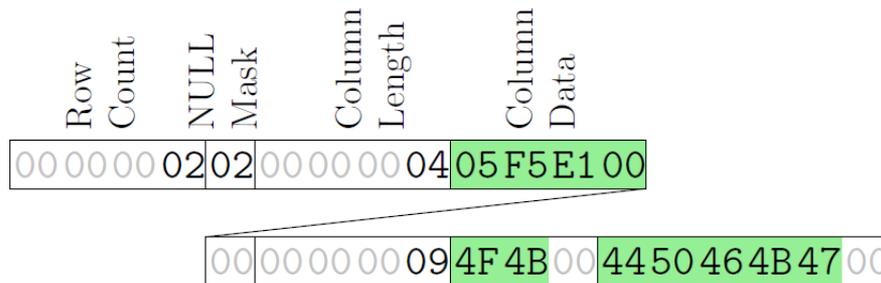
Implementierung der Autoren - Postgres

Postgre Protokoll



~70%
Overhead

Postgre++ Protokoll

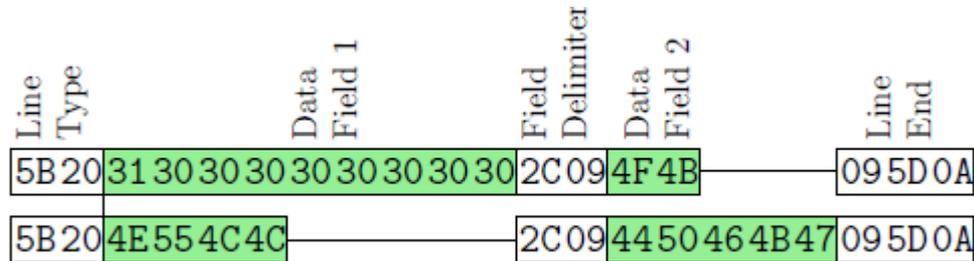


~50%
Overhead

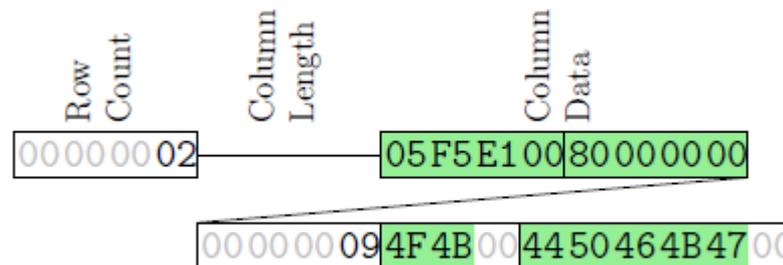
[1]:Raasveldt et al.

Implementierung der Autoren - MonetDB

MonetDB Protokoll

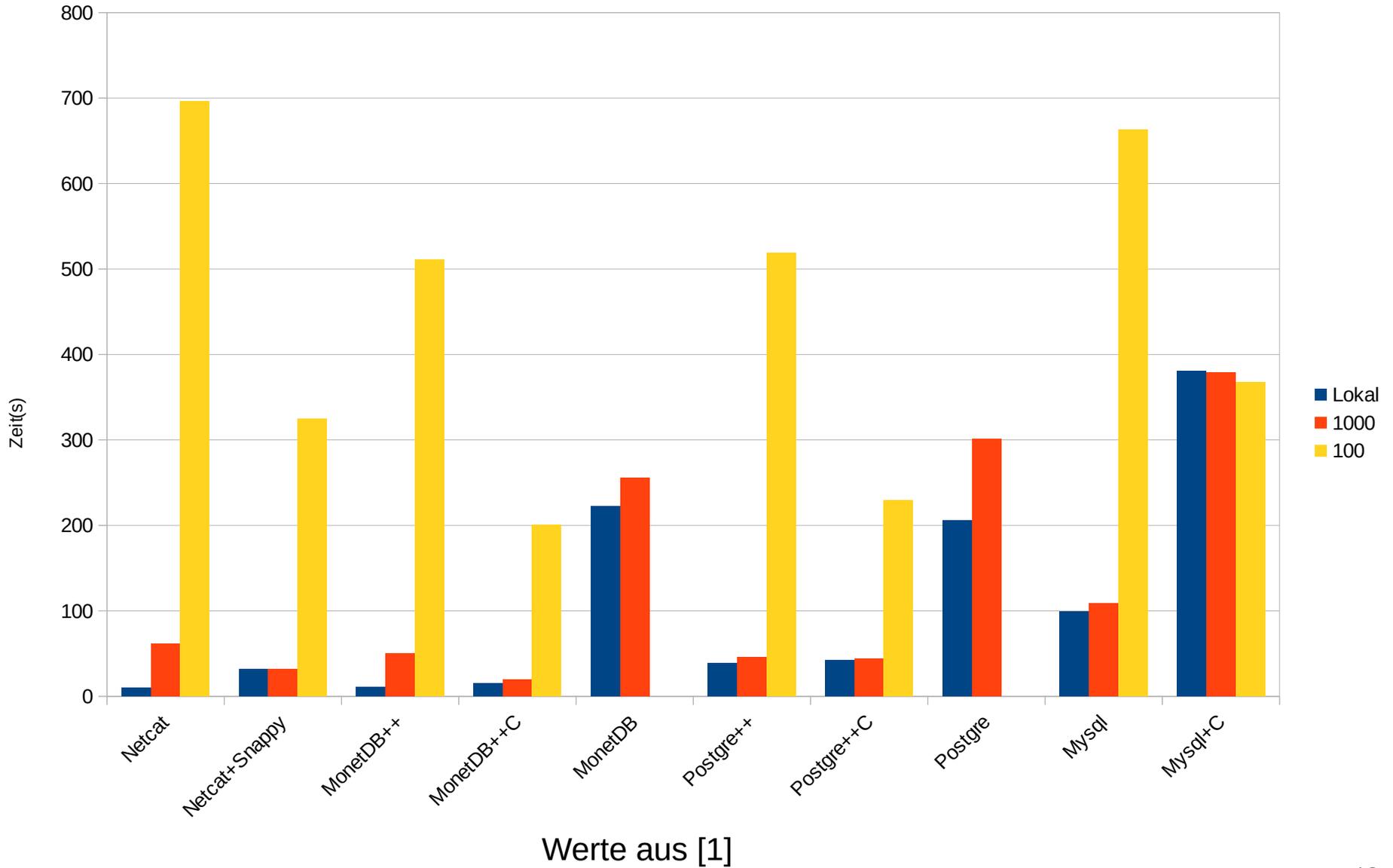


MonetDB++ Protokoll

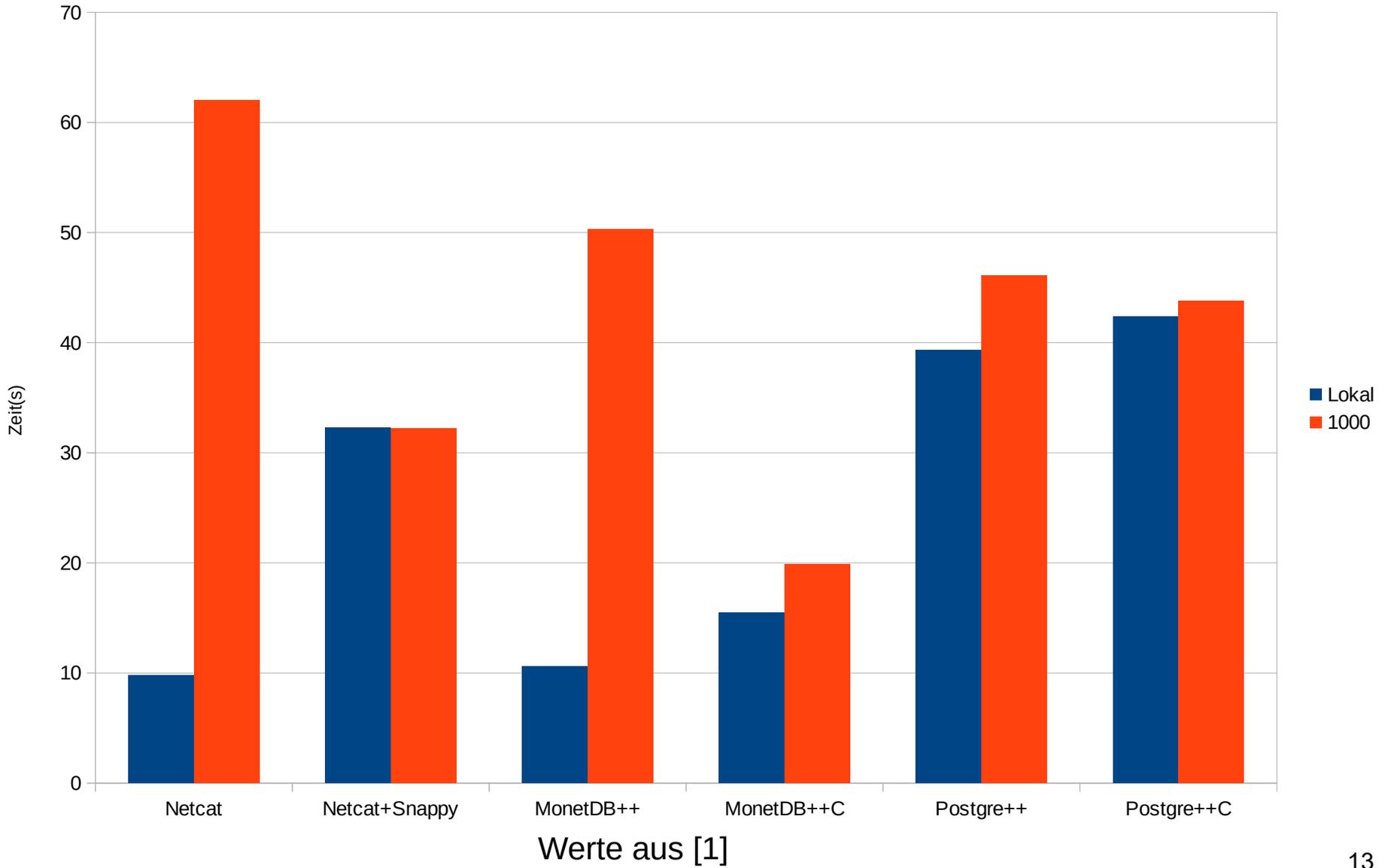


[1]:Raasveldt et al.

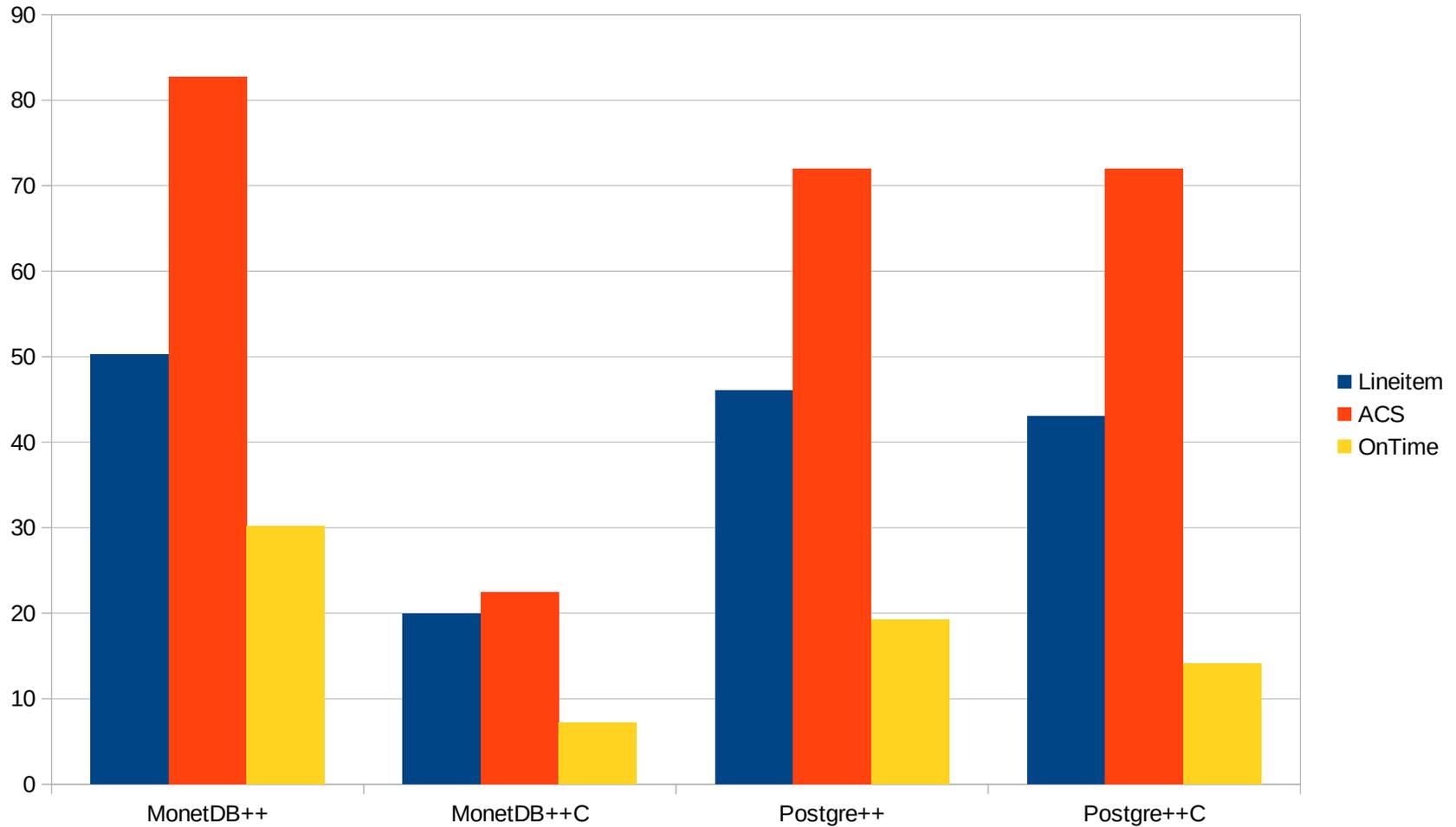
Transferring Lineitemtable



Transferring Linitemtable

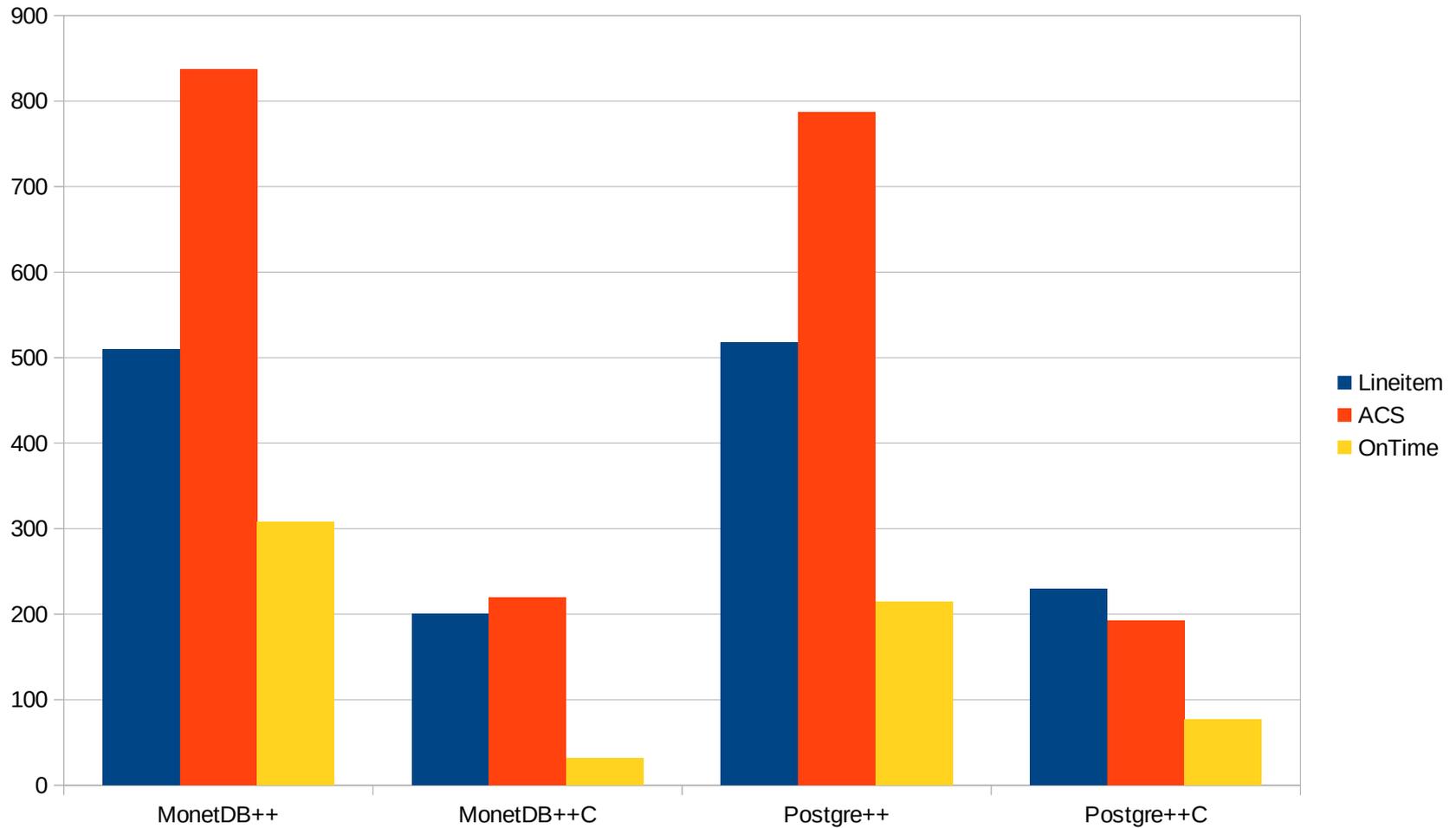


Verschiedene Tabellen über Lan übertragen



Werte aus [1]

Verschiedene Tabellen über WAN übertragen



Werte aus [1]

Zusammenfassung

Verbesserungen

- Kompression Spaltenbasiert
- Übertragung in Chunks
=> Benutzung des L3 Cache
- String Format
- => Reduzierung des Overheads

Steht noch aus

- Parallelisierung
=> jedoch immer Serialisierung
- Auswahl der Kompressionsmethode verbessern
- Bessere Komprimierung für unterschiedliche Anwendungsfälle

Vielen Dank für eure Aufmerksamkeit

Quellen

[1]: Raasveldt, Mühleisen - Don't Hold My Data Hostage – A Case For Client Protocol Redesign