



Friedrich Schiller Universität/Jena  
Lehrstuhl für Rechnerarchitektur

# **Entwicklung einer abstrakten Speicherkomponente für eine verteilte heterogene dynamische Infrastruktur in Java/CORBA**

Torsten Fink   Michael M. Gutzmann  
Torsten Wolf   Werner Erhard

# Überblick

1. Ansätze, Ziele, Werkzeuge
2. Überblick über das Gesamtsystem
3. Anforderungen an eine Speicherkomponente
4. Aufbau der Speicherkomponente
  - Datenspeicher und Datenobjekte
  - Verbindung von Datenspeichern
  - Dynamische Unterstützung mehrerer Protokolle
5. Messung der Übertragungsleistung
6. Erfahrungen mit Java und CORBA
7. Zusammenfassung und Ausblick

# Ansätze, Ziele, Werkzeuge

## Motivation:

- Per Internet sind leistungsfähige Rechenressourcen kombiniert einsetzbar.
- Dieses System (Metacomputer) ist heterogen und sehr dynamisch.  
⇒ schwer programmierbar

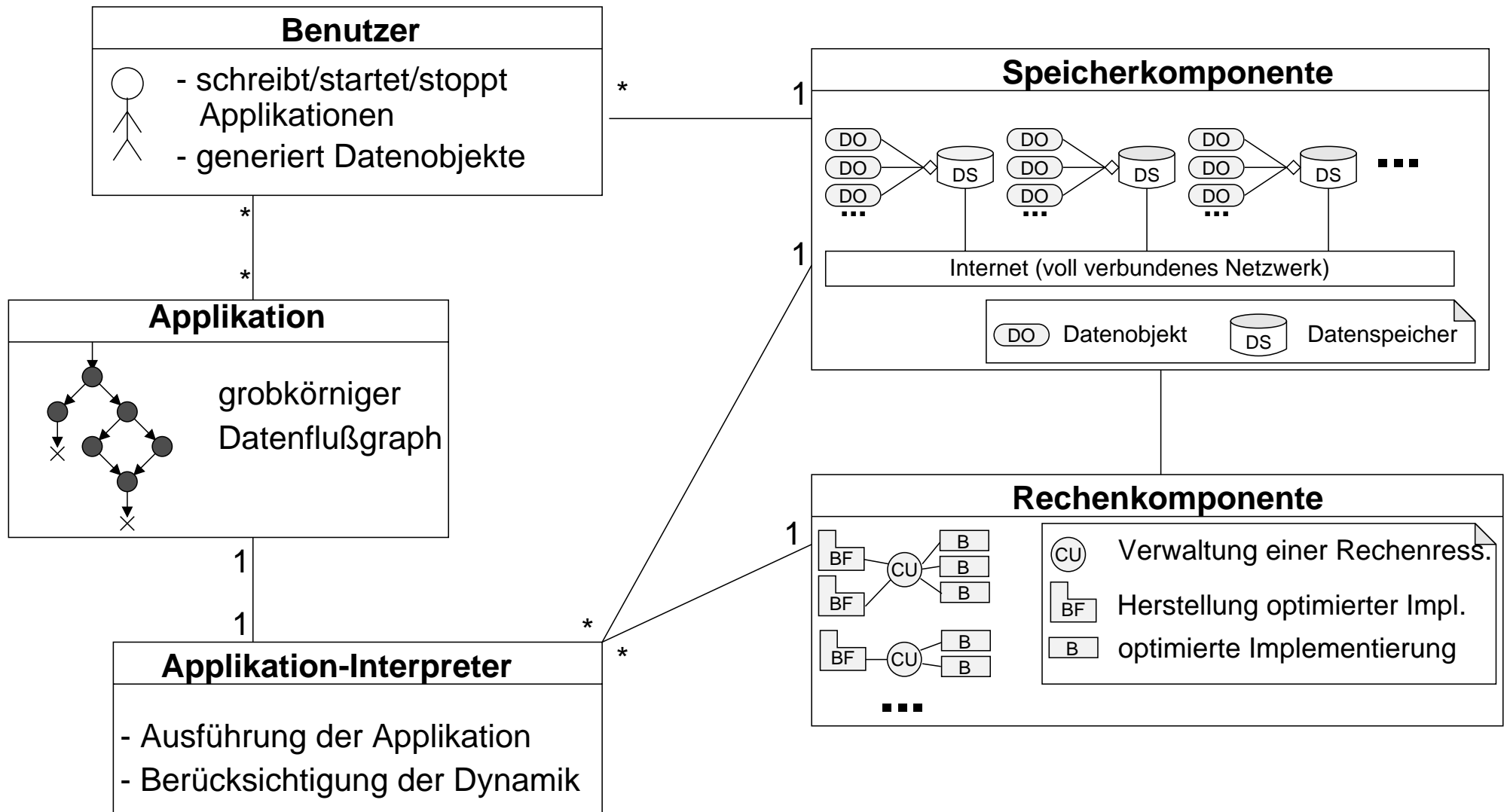
## Zielsetzung unseres Systems:

- Unterstützung von Applikationen mit grobkörniger Parallelität
- problem- und architektureoptimierte Nutzung von Rechenressourcen
- dynamische Ausführung einer Applikation

## Werkzeuge:

- für architekturunabhängige Komponenten: Java
- für architekturabhängige Komponenten: C, C++, MPL etc.
- als Middleware CORBA

# Überblick über das Gesamtsystem



# Anforderungen an eine Speicherkomponente

## **Abstraktion:**

Verdeckung der Implementierungsdetails

Berücksichtigung des Datentyps durch Selektoren/Modifikatoren

## **Erweiterbarkeit:**

dynamische Erweiterung um neue Datentypen/Datenspeicher

## **Ortstransparenz:**

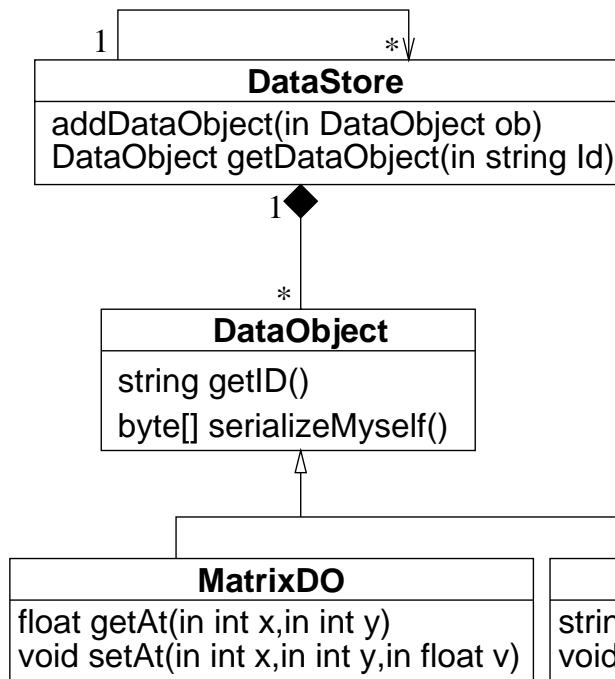
kein Wissen bzgl. des Ortes der Lagerung seitens des Benutzers

Zugriff auf Daten mittels eines eindeutigen Bezeichners

## **Zugriffseffizienz:**

dynamische Datenumverteilung  
Erstellung lokaler Kopien (Caching) }  $\Rightarrow$  protokollspezifische  
Datenübertragung

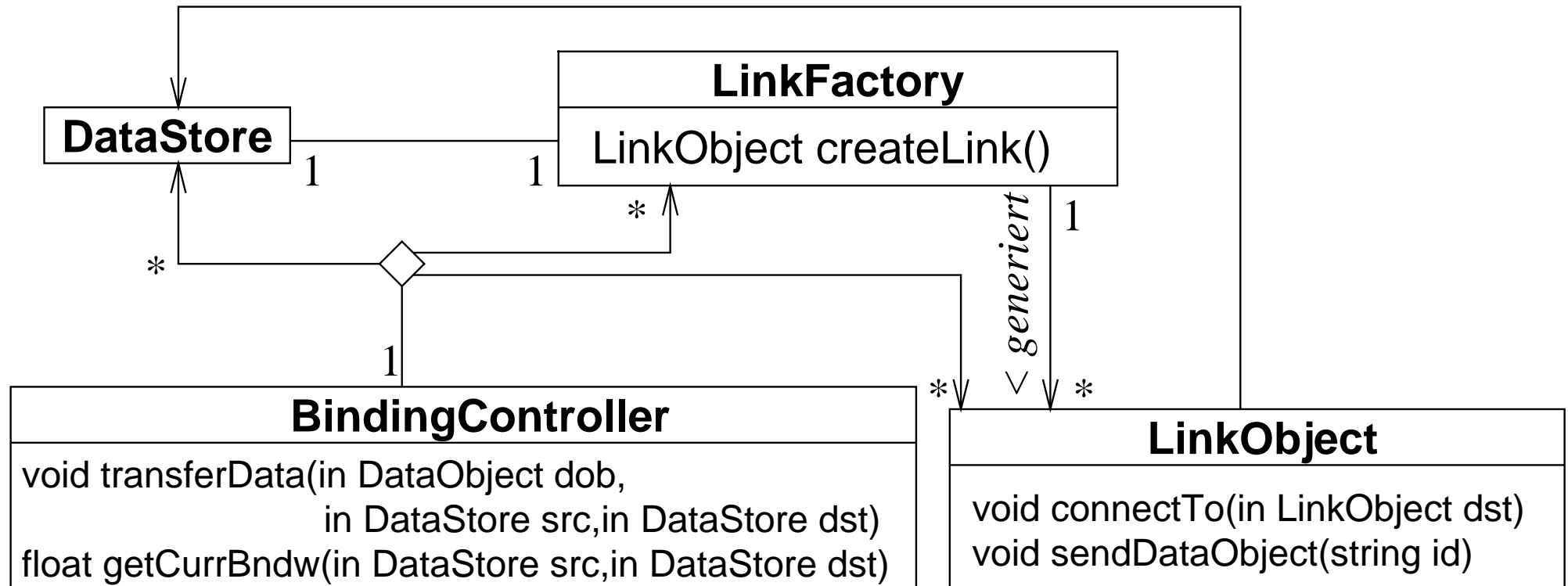
# Datenspeicher und Datenobjekte



- Datenspeicher (DS) verwalten physikalischen Speicher.
- DS bilden ein transparentes Netzwerk  
Jeder DS ist ein Repräsentant der Speicherkomponente.
- Unterstützung spezieller Datentypen durch eigene Klassen.

- Zur Optimierung können Datenobjekte kopiert werden.
- Die Konsistenzhaltung wird von den Datenobjekten autonom verwaltet.

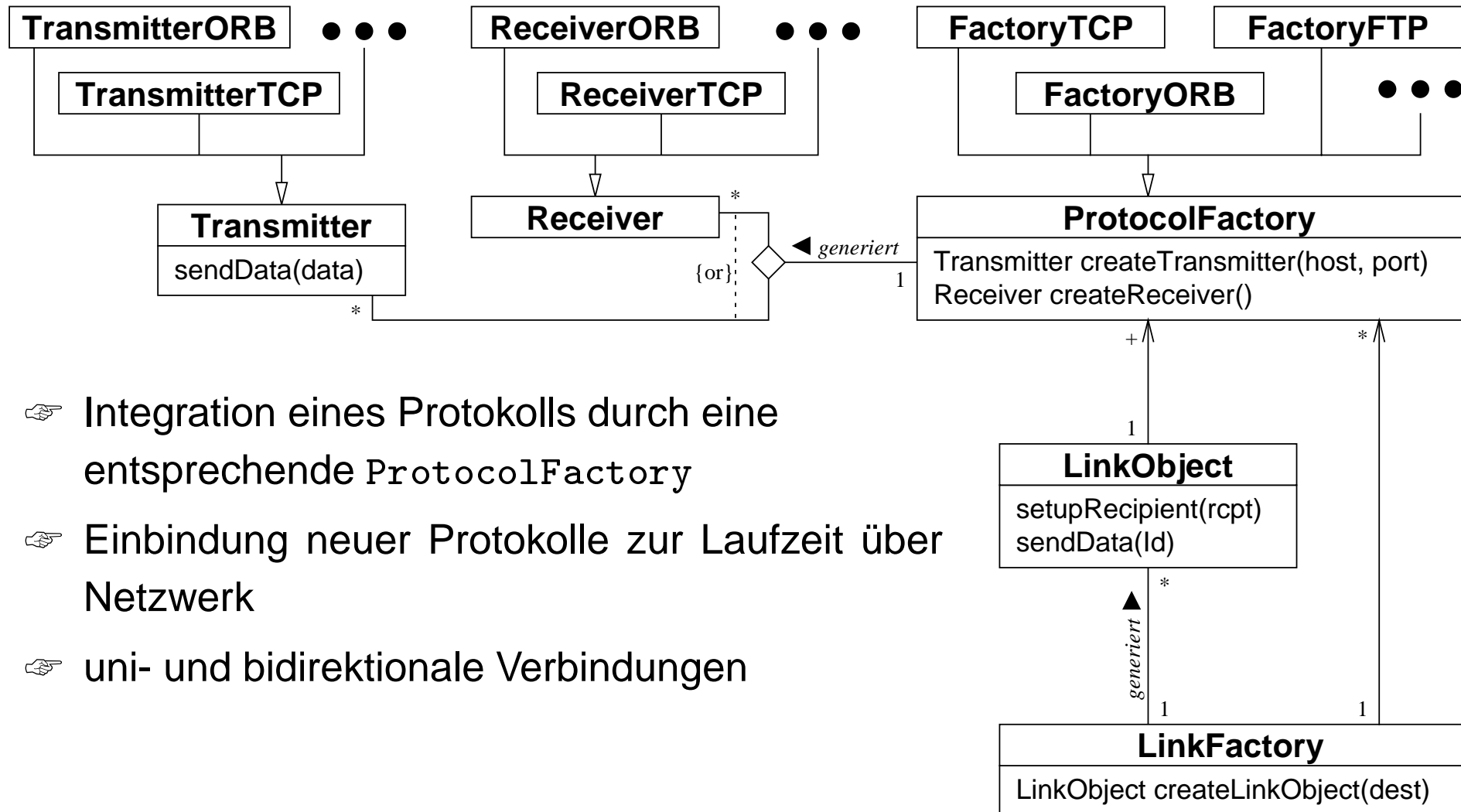
## Verbindung von Datenspeichern



Aufgaben des BindingController:

- Verwaltung der Übertragung von DatenObjekten, wobei evtl. neue Verbindungen aufgebaut werden müssen.
- Auskunft über der Bandbreiten zwischen einzelnen Datenspeichern.

# Dynamische Unterstützung mehrerer Protokolle



- ➡ Integration eines Protokolls durch eine entsprechende `ProtocolFactory`
- ➡ Einbindung neuer Protokolle zur Laufzeit über Netzwerk
- ➡ uni- und bidirektionale Verbindungen



# Messung der Übertragungsleistung

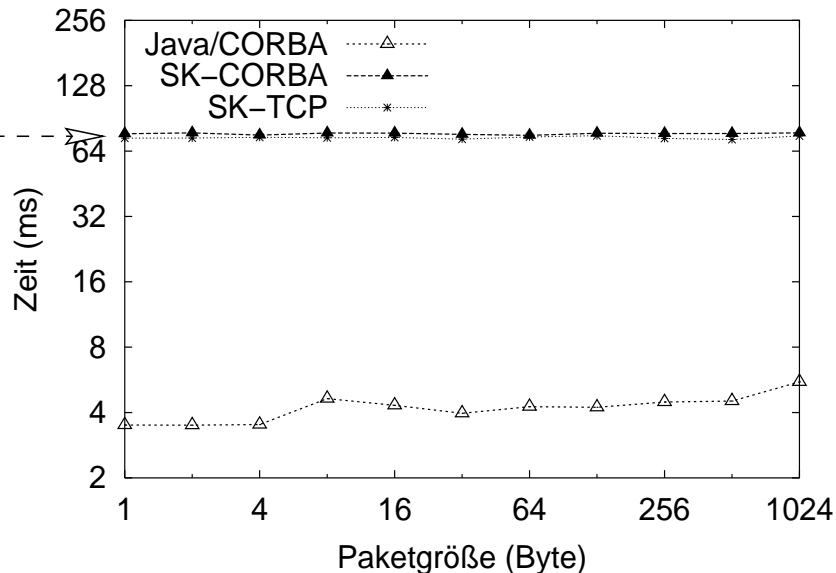
Messungen der Übertragungszeiten in 3 Szenarien (Sun Ultra-1, Ethernet 10MBit):

- direkt als Parameter mit Java/CORBA
- zwischen zwei DS-Objekten verschickt per CORBA
- zwischen zwei DS-Objekten per TCP

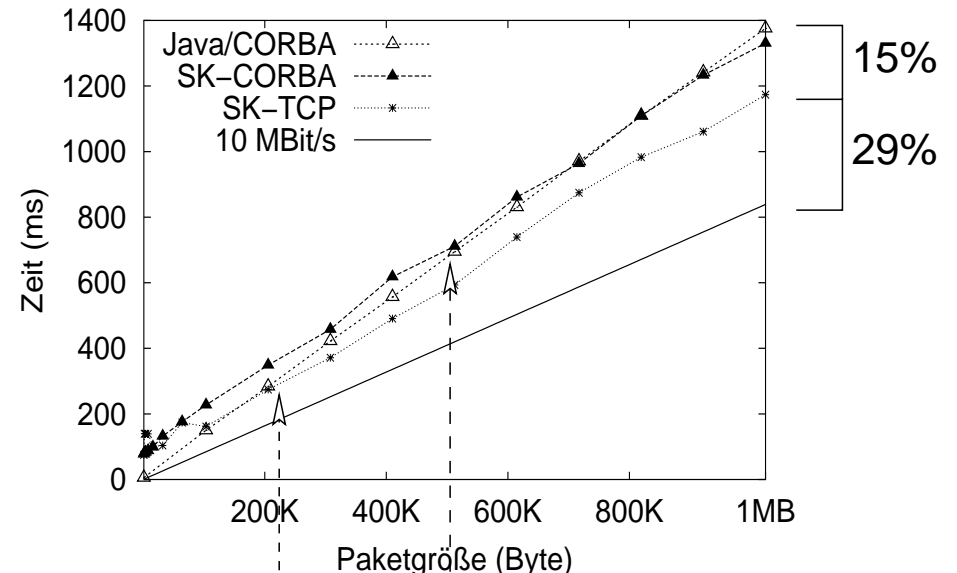
*Java/CORBA*

*SK-CORBA*

*SK-TCP*



--- Aufsetzzeit: 75 ms



--- vernachlässigbarer Overhead

--- SK-TCP < Java/CORBA

- ⇒ Bei 1 MB liegt die Übertragungsrate 29% unter maximaler Leistung.  
Grund dafür ist wahrscheinlich die schlechte Leistung von Java.

## Erfahrungen mit Java und CORBA

- Werkzeuge (JDK 1.1.5, ORBacus):
    - ⊕ hohe Stabilität
    - ⊖ geringer Komfort
      - keine grafischer Debugger, kein Profiling
  - Sprachumfang von Java:
    - ⊕ Serialisierungskonzept
      - Erlaubt einfache Übertragung beliebiger Datenobjekte.
    - ⊕ Threads
    - ⊖ fehlende Mehrfachvererbung
      - CORBA-Objekte werden von automatisch generiertem Code abgeleitet.
  - ⊖ geringe Geschwindigkeit
- ⇒ Die mächtigen Konzepte von Java unterstützen prototypische Entwicklungen.  
Bei realistischen Anwendungen müssen die Geschwindigkeitsanforderungen genau geprüft werden.

## Zusammenfassung

- Entwicklung einer Speicherkomponente einer umfassenden MC-Infrastruktur mit den Zielen: *Abstraktion, Erweiterbarkeit, Ortstransparenz, Zugriffseffizienz*.
- Effizienz durch Datenreplikation und Unterstützung unterschiedlicher Protokolle.
- Messungen auf einem 10 MBit Ethernet ergaben Organisationsaufwand von  $\approx 75ms$  und eine 71%ige Ausnutzung.

## Ausblick

- Konkreter Einsatz der Speicherkomponente in Applikationen
- Untersuchung unterschiedlicher Strategien für Datenlagerung, Datentypen und Konsistenzprotokolle