

# Projektskizze

**Norman Nabhan**

Datum: 27. November 2020



## Inhaltsverzeichnis

<b>1 Einführung</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Ziel . . . . .	1
<b>2 Projektstufen</b>	<b>2</b>
2.1 Theoretischer Teil . . . . .	2
2.2 Praktischer Teil . . . . .	2
<b>Literatur</b>	<b>3</b>
<b>Abbildungsverzeichnis</b>	<b>4</b>
<b>Tabellenverzeichnis</b>	<b>4</b>

# 1 Einführung

## 1.1 Motivation

Im Rahmen der Projektarbeit soll ein verteilter *in-memory storage* auf Basis von Apache Plasma entwickelt werden. Dieser verteilte Speicher soll in der Lage sein, große Datenmengen im Sinne von *Big Data* zu speichern und zur Verfügung zu stellen.

Plasma ist Teil des Apache Arrow Projekts. Dieses Projekt beschreibt sein Einsatzgebiet wie folgt:

“Apache Arrow defines a language-independent columnar memory format for flat and hierarchical data, organized for efficient analytic operations on modern hardware like CPUs and GPUs. The Arrow memory format also supports zero-copy reads for lightning-fast data access without serialization overhead.”[1]

Der Zugriff erfolgt hierbei über Bibliotheken in diversen Programmiersprachen. Hierbei sei zu beachten, dass Plasma selbst nicht als verteilte Anwendung konzipiert ist. Zwar können mehrere Instanzen auf einem Computer gestartet werden, jedoch verfügt Plasma über kein Protokoll, um mehrere Instanzen über ein Netzwerk zu verbinden. Somit ist auch der Speicher nicht verteilt und jeweils nur auf einem Computer verfügbar.

## 1.2 Ziel

Um die oben genannte Problematik zu lösen, soll ein Protokoll entwickelt werden, welches es ermöglicht mehrere Instanzen zu einem Netzwerk Cluster zu verbinden. Dieses Overlay-Netzwerk wird in Java entwickelt. Wie in Abbildung 1 dargestellt, sind alle Komponenten miteinander verbunden. Wobei jeder Node und der Dispatcher jeweils eine direkte Verbindung zu allen anderen Komponenten des Netzwerks besitzt. Wenn Daten gespeichert werden sollen, entscheidet der Dispatcher, welcher Node für die Daten verantwortlich ist. Der Dispatcher nutzt Redis zur Datenhaltung.

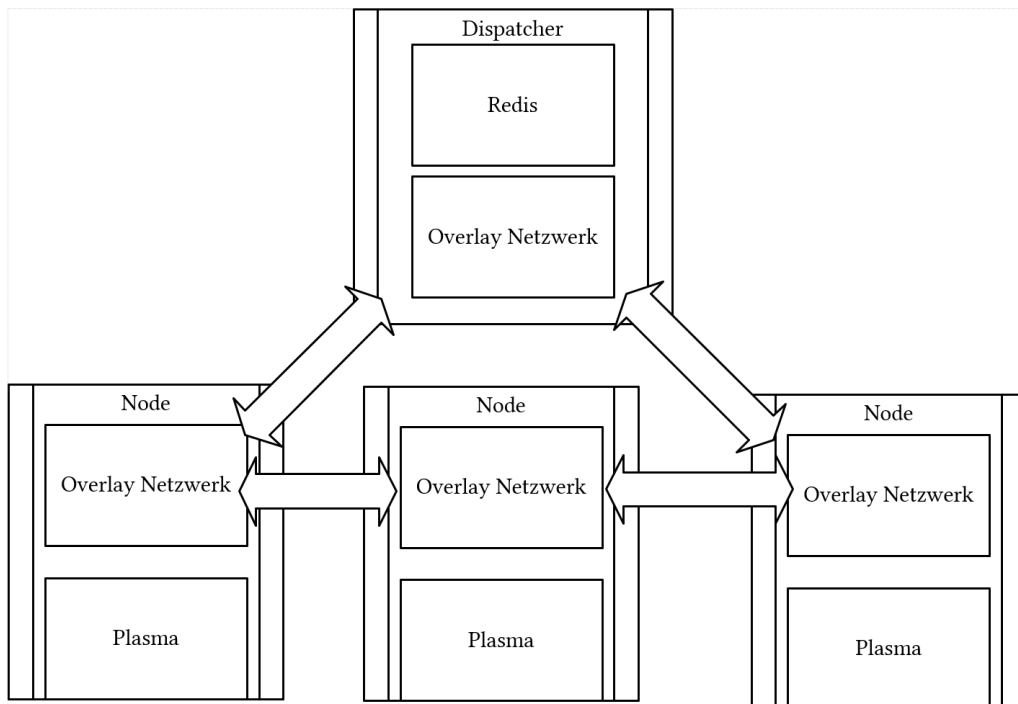


Abbildung 1: Skizze der angedachten Architektur.

## 2 Projektstufen

Die Projektarbeit gliedert sich in verschiedene, aufeinander folgenden Phasen. Die Projektarbeit beginnt mit einem theoretischen Teil und endet mit einem praktischen Teil.

### 2.1 Theoretischer Teil

Dieser Teil beginnt mit der Anforderungsanalyse. Die hier beschriebene Software soll in ein bereits existierendes System integriert werden. Somit muss die Kompatibilität zu den bereits existierenden Komponenten in Form von Soft- und Hardware gewährleistet sein. Darauf folgt der Entwurf der Anwendungsfälle. Hierbei gilt es Funktionalitäten zu beschreiben, die zwingend notwendig sind. Ergänzend dazu können noch weitere Funktionalitäten festgelegt werden, die zwar wünschenswert sind, aber nicht zwingend erforderlich sind. Danach muss die Architektur festgelegt und spezifiziert werden, um die Anwendungsfälle zu erfüllen. Verschiedene Lösungsansätze sollen beschrieben und verglichen werden.

### 2.2 Praktischer Teil

Basierend auf der Spezifikation werden die Komponenten implementiert. Zusätzlich sollen Tests erstellt und eine Dokumentation des Quellcodes erstellt werden. Die so erstellte Lösung soll in Hinblick auf wichtige Metriken wie IOPs, Bandbreite, Latenz etc. evaluiert werden.

## **Literatur**

- [1] The Apache Software Foundation. What is arrow? Abgerufen: 26.11.2020. [Online].  
Available: <https://arrow.apache.org/>

## Abbildungsverzeichnis

1	Skizze der angedachten Architektur. . . . .	2
---	---	---