1. **Was ist Machine learning**

Ist ein Konstrukt aus Algorithmen die aus einem Datenbestand lernen und daraus intelligente Aktionen generieren können. Machine learning ist adaptiv da sich die Eingabedaten und der Datenbestand jederzeit ändern können. Es gibt zwei Arten von Machine learning Algorithmen:

***Unsupervised:*** Algorithmus lernt aus den Daten ohne im Voraus bekannte Zielwerte zu kennen. Bsp.: Lernt selbstständig aus einer Reihe von Fotos Gesichter zu erkennen indem er selbständig Daten in Gruppen clustert. ( 🡪 Landschaftsbilder sind signifikant unterschiedlich zu Gesichtern, der Algorithmus weiß zwar nicht was ein Gesicht ist aber er lernt das Gesichter anders aussehen als Landschaftsbilder und kann diese somit von anderen Bildern unterscheiden)

***Supervised:*** Algorithmus lernt durch Trainingsdatensatz und muss zuvor antrainiert werden. Bsp.: Lernt aus Beispielen welche durch einen Benutzer vorgelegt werden. Es wird eine Reihe von Bildern mit Gesichtern und ohne Gesichter eingespielt. Ein guter Algorithmus lernt mit der Zeit aus diesen Daten und kann später selbständig differenzieren was ein Gesicht ist und was nicht.

1. **Was ist ein MAPE loop und was passiert in den einzelnen Schritten**

?

1. **Was ist ein Peer-To-Peer Netzwerk?**

Ist eine spezielle Art eines Rechnernetzwerkes bei dem alle Rechner im Netz gleich berechtigt zusammenarbeiten. Einfache Peer-To-Peer Netze organisieren sich selbst das heißt es gibt keine zentralen Server der kontrolliert. Nodes stellen sich gegenseitig Ressourcen und Betriebsmittel zur Verfügung. Peer-To-Peer Netzwerke können in 2 Arten eingeteilt werden:

***Strukturiertes Netzwerk:*** Peers verwalten auch Informationen über Ressourcen von

anderen Peers wodurch eine Zielorientierte Suche ermöglicht wird. Nachteil es ist

zusätzlicher Aufwand für die Verwaltung dieser Information notwendig.

🡪 Bsp. Routing Tabellen

***Unstrukturiertes Netzwerk:*** Peers verwalten keine Informationen über die

Ressourcen von anderen Peers wodurch eine gezielte Weiterleitung einer

Suchanfrage nicht möglich ist 🡪 Bsp. Gnutela

1. **Nennen Sie je einen Vorteil, einen Nachteil und einen Anwendungsfall von P2P Systemen.**

***Vorteile:*** Relativ Robust, kein single point of failure. Fällt ein Rechner aus bricht das Netzwerk nicht auseinander, es fallen lediglich die von dem Rechner zur Verfügung gestellten Ressourcen aus.

***Nachteile:*** Verfügbarkeit von Services ist nicht garantiert. Bei unstrukturierten P2P Netzten entsteht ein hoher Grad an flooding (Bei Suchanfragen wird Netztwerk von Anfragen überschwemmt).

***Anwendungsfälle:*** File Sharing, Distributed Computing, Multiplayer Games, Vehicle/ Road comminucation

1. **Was ist das Ziel einer Distributed HashTable?**

Ein Schlüssel wird durch eine Hashfunktion auf einen Wert gemappt. Ziel ist es eine Struktur zur Kommunikation zwischen einzelne Nodes einzuführen. Generell soll ein gutes Verteilen bzw. Auffinden von Informationen auf mehreren Nodes im Netzwerk gewährleistet werden.

1. **Wie funktioniert DHT (Chord)**

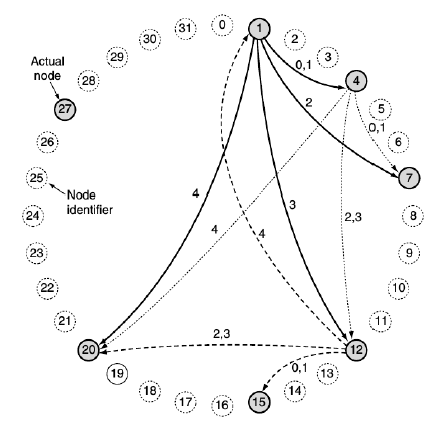
Jede Chord (DHT) besitzt n Knoten

Jeder Knoten besitzt eine IP Adresse

Der Gesamtindex ist über die Knoten verteilt

Die Navigation erfolgt über einen Index in einem virtuellen Netzwerk. Nicht über die IP und nicht über den Ressourcen Namen.

IDs sind m-BIT Zahlen die aufsteigend um den Ring gehen.



Eine Hashfunktion wandelt eine IP Adresse eines Knoten in einen Identifierer (ID) um. IP Adresse wird in eine 160 Bit Knoten ID umgewandelt.

~~Alle Netzwerkknoten werden in einer Ringstruktur angeordnet. Jeder Knoten hat eine Verbindung zu seinen Vorgänger und Nachfolger. Die Ringstruktur ermöglicht es, durch den Einsatz einer verteilten HashTable, eine binäre Suche durchzuführen. .. Beispiel + Erkärung~~

1. **Welche Funktionen kennt DHT**

***Successor:*** Funktion lautet *find\_successor(k)* und gibt den nächsten direkten Nachfolger vom

Knoten *k* an.

***Hash Funktion:*** Um aus einen Key einen identifier zu erzeugen.

***Insert:*** Einfügen eines Key-Value-pair auf einem Knoten.

***~~Remove:~~*** ~~Knoten wird aus Netzwerk entfernt.~~ NICHT IN DEN FOLIEN wenn dann **leave**

***Lookup:*** Anfragender Knoten sendet seine IP und seine Key an seinen Nachfolger. Diese Nachricht wird solange weitergereicht bis die Ressource gefunden ist. Der Knoten welche die Resource schickt diese dann direkt zum Anfragenden Knoten.

***Join the swarm:*** Ein neuer Knoten wird ins Netzwerk eingefügt.

1. ***Wie funktioniert insert in DHT***

Insert(torrent, myIP)

Key erzeugen: key = hash(torrent)

Finde den zuständigen Knoten der für den key der zuständig ist für das Speichern: successor(key)

So wird der Index zufällig im Netzwerk verteilt. Der key wird immer auf dem Knote gespeichert der dem key am nächsten liegt.

1. ***Wie funktioniert Join in DHT***

* Der neue Knoten **r** fragt existierende Knoten: wer ist mein Nachfolge Knoten (***successor(r)***)
* Dann fragt Knoten **r** seinen Nachfolger Knoten successor(r): wer ist mein Vorgänger Knoten (predecessor).
* Knoten **r** setzt sich dann zwischen diesen beiden Knoten in das Netzwerk indem er sich successor IP und predecessor IP speichert.
* Natürlich stimmt die Fingertable auf den successor nodes nicht mehr. Diese werden regulär bei den successor anfragen upgedated.

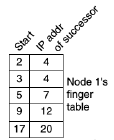
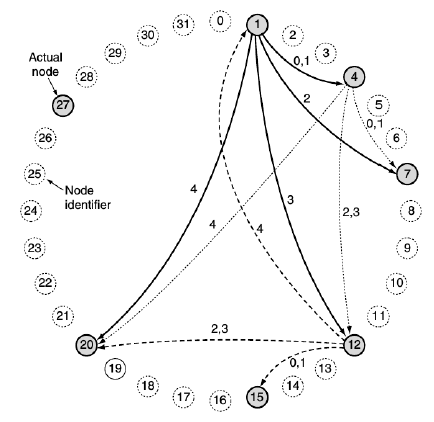
1. ***Wie kann man lookup in DHT beschleunigen***

Der lineare lookup O(n) ist bei einem Ring mit vielen Knoten sehr ineffizient.

Um bei der Suche nicht von Knoten zu Knoten zu hüpfen, gibt es ein Fingertable.

Die Fingetable biete ShortCuts zu bestimmten Knoten im Netzwerk.

Beispiel einer Fingertabelle von Knoten 1 in einem 5 BIT Network (=32 Knoten):

Zb.: Von Knoten 1 ausgehend:

Key 9 wird gesucht: Finger zeigt auf Knoten 12 -> Knoten 12 hat den Key und weis wo die Ressource zu finden ist. Daher direkter sprung zu Knoten 12 und nicht über alle Knoten (4 -> 7 -> 12) durchzuwandern.

Von average zu