TU München, Fakultät für Informatik
Lehrstuhl III: Datenbanksysteme
Prof. Alfons Kemper, Ph.D.

Übung zur Vorlesung *Einsatz und Realisierung von Datenbanken* im SoSe20

Maximilian {Bandle, Schüle}, Josef Schmeißer (i3erdb@in.tum.de)
http://db.in.tum.de/teaching/ss20/impldb/

Blatt Nr. 12

Hausaufgabe 1

Berechnen Sie für folgende drei Dokumente die TF-IDF-Werte:

1. „Beim Fußball dauert ein Spiel neunzig Minuten – und am Ende gewinnen die Deutschen“
2. „Beim Fußball muss das Runde (der Ball) in das Eckige (das Tor)“
3. „Nie war ein Tor so wertvoll wie jetzt“

Welches Ranking ergibt sich gemäß der Relevanzwerte für die Anfrage: „Fußball“ ∧ „Tor“. Zur Ermittlung des TF Wertes gehen sie davon aus, dass alle Wörter eines Dokuments *interessant* sind?
Zur Berechnung des Rankings reicht es nur die TF-IDF-Werte von *Fußball* und *Tor* zu berechnen.

Fußball IDF: 0.176

	Dokument 1	Dokument 2	Dokument 3
TF	0.077	0.083	0
TF-IDF	0.014	0.015	0

Tor IDF: 0.176

	Dokument 1	Dokument 2	Dokument 3
TF	0	0.083	0.125
TF-IDF	0	0.015	0.022

Ranking Dokument 2: 0.029
Dokument 3: 0.022
Dokument 1: 0.014

Hausaufgabe 2

In dem in Abbildung 1 gezeigten Netzwerk von Web-Seiten wird ein kleines Beispiel für einen Webgraphen gezeigt. Lösen Sie folgende Aufgaben.

1. Berechnen Sie, für das in Abbildung gezeigte Netzwerk, den PageRank, sowie die HITS-Werte nach 2 Iterationen. Nutzen Sie $1/|V|$ als Anfangswert für den PageRank und 1 für HITS. $a = 0.1$
2. Formulieren sie eine Iteration des Pagerank Algorithmus in SQL. Der Graph ist dabei in der Tabelle *edges(VFrom,To)* gespeichert, die aktuelle PageRank Gewichtung in der Tabelle *pagerank(Vertex,Weight)*. Sie können die Anzahl der Knoten als Konstante annehmen, z.B. 1000.

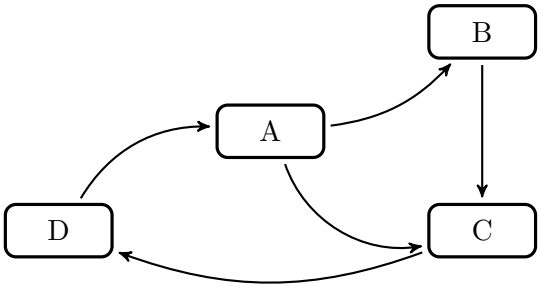

Abbildung 1: Ein kleiner Webgraph.

HITS: Iteration 1

	A	B	C	D
Hub	2	1	1	1
Auth (vorläufig)	1	2	3	1
Auth (normalisiert)	$\frac{1}{3}$	$\frac{2}{3}$	$\frac{3}{3}$	$\frac{1}{3}$

HITS: Iteration 2

	A	B	C	D
Hub	$\frac{5}{3}$	1	$\frac{1}{3}$	$\frac{1}{3}$
Auth (vorläufig)	$\frac{1}{3}$	$\frac{5}{3}$	$\frac{8}{3}$	$\frac{1}{3}$
Auth (normalisiert)	$\frac{1}{8}$	$\frac{5}{8}$	$\frac{8}{8}$	$\frac{1}{8}$

PageRank

1.

	A	B	C	D
PR Iter 1	$\frac{1}{4}$	$\frac{11}{80}$	$\frac{29}{80}$	$\frac{1}{4}$
PR Iter 2	$\frac{1}{4}$	$\frac{11}{80}$	0.2613	0.3513

2.
```
select VTo, 0.1/(CAST((select count(*) from pagerank)AS FLOAT))
    +0.9*sum(Beitrag)
from(
  select e.VTo, p.Weight/
     (select count(*) from edges x where x.VFrom=e.VFrom) as Beitrag
  from edges e , pagerank p
  where e.VFrom=p.Vertex
) i
group by VTo
```

Hausaufgabe 3

In Abbildung 2 sind drei Graphen gegeben, ein sternförmiger, eine Clique und ein linear angeordneter.

1. Berechnen Sie den Grad der Knoten für jeden der Graphen.
   Stern: $C_D(A) = 4, C_D(B) = 1, C_D(C) = 1, C_D(D) = 1, C_D(E) = 1$
   Clique: $C_D(A) = 4, C_D(B) = 4, C_D(C) = 4, C_D(D) = 4, C_D(E) = 4$
   linear: $C_D(A) = 1, C_D(B) = 2, C_D(C) = 2, C_D(D) = 2, C_D(E) = 1$
2. Berechnen Sie die Verbindungszentralität $C_D(G)$ der drei Graphen, sowie deren normierte Verbindungszentralität $C'_D(G)$.

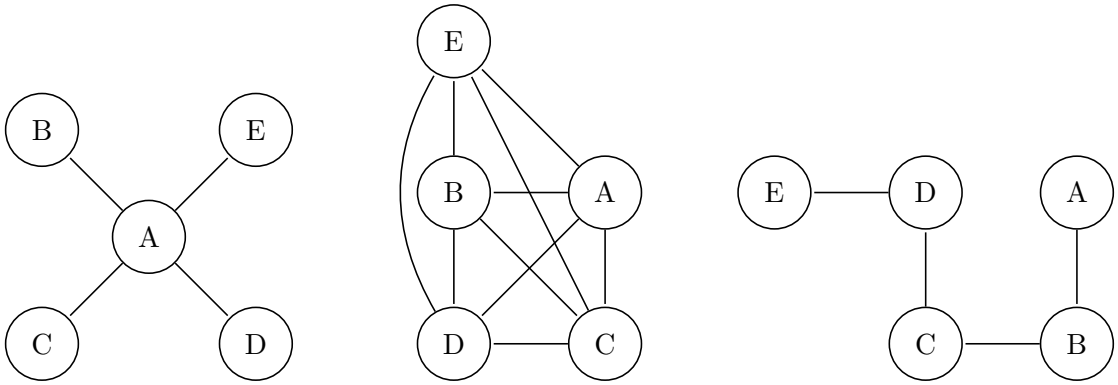

Abbildung 2: Star, Clique und Linie.

$$
\begin{aligned}
C_D(G*) &= \sum_{v \in V} [C_D(v*) - C_D(v)] = \sum_{v \in V} [C_D(A) - C_D(v)] = (|V| - 2)(|V| - 1) \\
&= (4-4) + (4-1) + (4-1) + (4-1) + (4-1) = 12 = (5-2)(5-1) \\
C'_D(G*) &= \frac{C_D(G*)}{C_D(G*)} = 1 \\
C_D(G_{Clique}) &= \sum_{v \in V} [C_D(v*) - C_D(v)] = 0 \\
C'_D(G_{Clique}) &= \frac{C_D(G_{Clique})}{C_D(G*)} = 0/12 = 0 \\
C_D(G_{lin}) &= \sum_{v \in V} [C_D(v*) - C_D(v)] = \sum_{v \in V} [C_D(B) - C_D(v)] = \\
&= (2-1) + (2-2) + (2-2) + (2-2) + (2-1) = 2 \\
C'_D(G_{lin}) &= \frac{C_D(G_{lin})}{C_D(G*)} = 2/12
\end{aligned}
$$

3. Berechnen Sie die Nähe-Zentralität $H_G(v)$ für einen Knoten der drei Graphen.

Für $G*$:

$$\begin{aligned}
H_{G*}(A) &= \sum_{y \neq A \in V} [\frac{1}{d(y,A)}] = \frac{1}{d(B,A)} + \frac{1}{d(C,A)} + \frac{1}{d(D,A)} + \frac{1}{d(E,A)} \\
&= \frac{1}{1} + \frac{1}{1} + \frac{1}{1} + \frac{1}{1} = 4 \\
H_{G*}(B) &= \sum_{y \neq B \in V} [\frac{1}{d(y,B}] = \frac{1}{d(A,B)} + \frac{1}{d(C,B)} + \frac{1}{d(D,B)} + \frac{1}{d(E,B)} \\
&= \frac{1}{1} + \frac{1}{2} + \frac{1}{2} + \frac{1}{2} = 2.5 \\
H_{G*}(C) &= H_{G*}(D) = H_{G*}(E) = 2.5 \text{ analog.}
\end{aligned}$$

Für G_{Clique}:

$$\begin{aligned}
H_{G_{Clique}}(A) &= \sum_{y \neq A \in V} [\frac{1}{d(y,A)}] = \frac{1}{d(B,A)} + \frac{1}{d(C,A)} + \frac{1}{d(D,A)} + \frac{1}{d(E,A)} \\
&= \frac{1}{1} + \frac{1}{1} + \frac{1}{1} + \frac{1}{1} = 4 \\
H_{G_{Clique}}(B) &= H_{G_{Clique}}(C) = H_{G_{Clique}}(D) = H_{G_{Clique}}(E) = 4 \text{ analog.}
\end{aligned}$$

Für G_{linear}:

$$\begin{aligned}
H_{G_{linear}}(A) &= \sum_{y \neq A \in V} [\frac{1}{d(y,A)}] = \frac{1}{d(B,A)} + \frac{1}{d(C,A)} + \frac{1}{d(D,A)} + \frac{1}{d(E,A)} \\
&= \frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} = \frac{25}{12} \\
H_{G_{linear}}(B) &= \sum_{y \neq B \in V} [\frac{1}{d(y,B)}] = \frac{1}{d(A,B)} + \frac{1}{d(C,B)} + \frac{1}{d(D,B)} + \frac{1}{d(E,B)} \\
&= \frac{1}{1} + \frac{1}{1} + \frac{1}{2} + \frac{1}{3} = \frac{17}{6} \\
H_{G_{linear}}(C) &= \sum_{y \neq C \in V} [\frac{1}{d(y,C)}] = \frac{1}{d(A,C)} + \frac{1}{d(B,C)} + \frac{1}{d(D,C)} + \frac{1}{d(E,C)} \\
&= \frac{1}{2} + \frac{1}{1} + \frac{1}{1} + \frac{1}{2} = 3 \\
H_{G_{linear}}(D) &= \sum_{y \neq D \in V} [\frac{1}{d(y,D)}] = \frac{1}{d(A,D)} + \frac{1}{d(B,D)} + \frac{1}{d(C,D)} + \frac{1}{d(E,D)} \\
&= \frac{1}{3} + \frac{1}{2} + \frac{1}{1} + \frac{1}{1} = \frac{17}{6} \\
H_{G_{linear}}(E) &= \sum_{y \neq E \in V} [\frac{1}{d(y,E)}] = \frac{1}{d(A,E)} + \frac{1}{d(B,E)} + \frac{1}{d(C,E)} + \frac{1}{d(D,A)} \\
&= \frac{1}{4} + \frac{1}{3} + \frac{1}{2} + \frac{1}{1} = \frac{25}{12}
\end{aligned}$$

4. Berechnen Sie die Pfad-Zentralität $H_G(v)$ für einen Knoten der drei Graphen.

Für $G*$:

$$\begin{aligned}
C_{G*}(A) &= \sum_{s\neq A\neq t\in V} [\frac{\sigma_{st}(v)}{\sigma_{st}}] \\
&= \frac{\sigma_{BC}(v)}{\sigma_{BC}} + \frac{\sigma_{BD}(v)}{\sigma_{BD}} + \frac{\sigma_{BE}(v)}{\sigma_{BE}} + \frac{\sigma_{CD}(v)}{\sigma_{CD}} + \frac{\sigma_{CE}(v)}{\sigma_{CE}} + \frac{\sigma_{DE}(v)}{\sigma_{DE}} \\
&= \frac{1}{1} + \frac{1}{1} + \frac{1}{1} + \frac{1}{1} + \frac{1}{1} + \frac{1}{1} = 6 \\
C_{G*}(B) &= \sum_{s\neq B\neq t\in V} [\frac{\sigma_{st}(v)}{\sigma_{st}}] \\
&= \frac{\sigma_{AC}(v)}{\sigma_{AC}} + \frac{\sigma_{AD}(v)}{\sigma_{AD}} + \frac{\sigma_{AE}(v)}{\sigma_{AE}} + \frac{\sigma_{CD}(v)}{\sigma_{CD}} + \frac{\sigma_{CE}(v)}{\sigma_{CE}} + \frac{\sigma_{DE}(v)}{\sigma_{DE}} \\
&= \frac{0}{1} + \frac{0}{1} + \frac{0}{1} + \frac{0}{1} + \frac{0}{1} + \frac{0}{1} = 0 \\
C_{G*}(C) &= C_{G*}(D) = C_{G*}(E) = 0 \text{ analog.}
\end{aligned}$$

Für G_{Clique}:

$$\begin{aligned}
C_{G*}(A) &= \sum_{s\neq A\neq t\in V} [\frac{\sigma_{st}(v)}{\sigma_{st}}] \\
&= \frac{\sigma_{BC}(v)}{\sigma_{BC}} + \frac{\sigma_{BD}(v)}{\sigma_{BD}} + \frac{\sigma_{BE}(v)}{\sigma_{BE}} + \frac{\sigma_{CD}(v)}{\sigma_{CD}} + \frac{\sigma_{CE}(v)}{\sigma_{CE}} + \frac{\sigma_{DE}(v)}{\sigma_{DE}} \\
&= \frac{0}{1} + \frac{0}{1} + \frac{0}{1} + \frac{0}{1} + \frac{0}{1} + \frac{0}{1} = 0 \\
C_{G*}(B) &= C_{G*}(C) = C_{G*}(D) = C_{G*}(E) = 0 \text{ analog.}
\end{aligned}$$

Für G_{linear}:

$$\begin{aligned}
C_{G*}(A) &= \sum_{s\neq A\neq t\in V} [\frac{\sigma_{st}(v)}{\sigma_{st}}] \\
&= \frac{\sigma_{BC}(v)}{\sigma_{BC}} + \frac{\sigma_{BD}(v)}{\sigma_{BD}} + \frac{\sigma_{BE}(v)}{\sigma_{BE}} + \frac{\sigma_{CD}(v)}{\sigma_{CD}} + \frac{\sigma_{CE}(v)}{\sigma_{CE}} + \frac{\sigma_{DE}(v)}{\sigma_{DE}} \\
&= \frac{0}{1} + \frac{0}{1} + \frac{0}{1} + \frac{0}{1} + \frac{0}{1} + \frac{0}{1} = 0 \\
C_{G*}(B) &= \sum_{s\neq B\neq t\in V} [\frac{\sigma_{st}(v)}{\sigma_{st}}] \\
&= \frac{\sigma_{AC}(v)}{\sigma_{AC}} + \frac{\sigma_{AD}(v)}{\sigma_{AD}} + \frac{\sigma_{AE}(v)}{\sigma_{AE}} + \frac{\sigma_{CD}(v)}{\sigma_{CD}} + \frac{\sigma_{CE}(v)}{\sigma_{CE}} + \frac{\sigma_{DE}(v)}{\sigma_{DE}} \\
&= \frac{1}{1} + \frac{1}{1} + \frac{1}{1} + \frac{0}{1} + \frac{0}{1} + \frac{0}{1} = 3 \\
C_{G*}(C) &= \sum_{s\neq C\neq t\in V} [\frac{\sigma_{st}(v)}{\sigma_{st}}] \\
&= \frac{\sigma_{AB}(v)}{\sigma_{AB}} + \frac{\sigma_{AD}(v)}{\sigma_{AD}} + \frac{\sigma_{AE}(v)}{\sigma_{AE}} + \frac{\sigma_{BD}(v)}{\sigma_{BD}} + \frac{\sigma_{BE}(v)}{\sigma_{BE}} + \frac{\sigma_{DE}(v)}{\sigma_{DE}} \\
&= \frac{0}{1} + \frac{1}{1} + \frac{1}{1} + \frac{1}{1} + \frac{1}{1} + \frac{0}{1} = 4 \\
C_{G*}(D) &= \sum_{s\neq D\neq t\in V} [\frac{\sigma_{st}(v)}{\sigma_{st}}] \\
&= \frac{\sigma_{AB}(v)}{\sigma_{AB}} + \frac{\sigma_{AC}(v)}{\sigma_{AC}} + \frac{\sigma_{AE}(v)}{\sigma_{AE}} + \frac{\sigma_{BC}(v)}{\sigma_{BC}} + \frac{\sigma_{BE}(v)}{\sigma_{BE}} + \frac{\sigma_{CE}(v)}{\sigma_{CE}} \\
&= \frac{0}{1} + \frac{0}{1} + \frac{1}{1} + \frac{0}{1} + \frac{1}{1} + \frac{1}{1} = 3
\end{aligned}$$

Hausaufgabe 4

Zeigen Sie, dass die Suche in einem Chord-Overlaynetzwerk durch die Nutzung der FingerTabellen in maximal logarithmisch vielen Schritten zur Größe des Zahlenrings (bzw. der Anzahl der Stationen) durchgeführt werden kann. Verwenden Sie die Suche nach K57 beginnend an Station P11 (siehe Abbildung 3) zur Illustration.

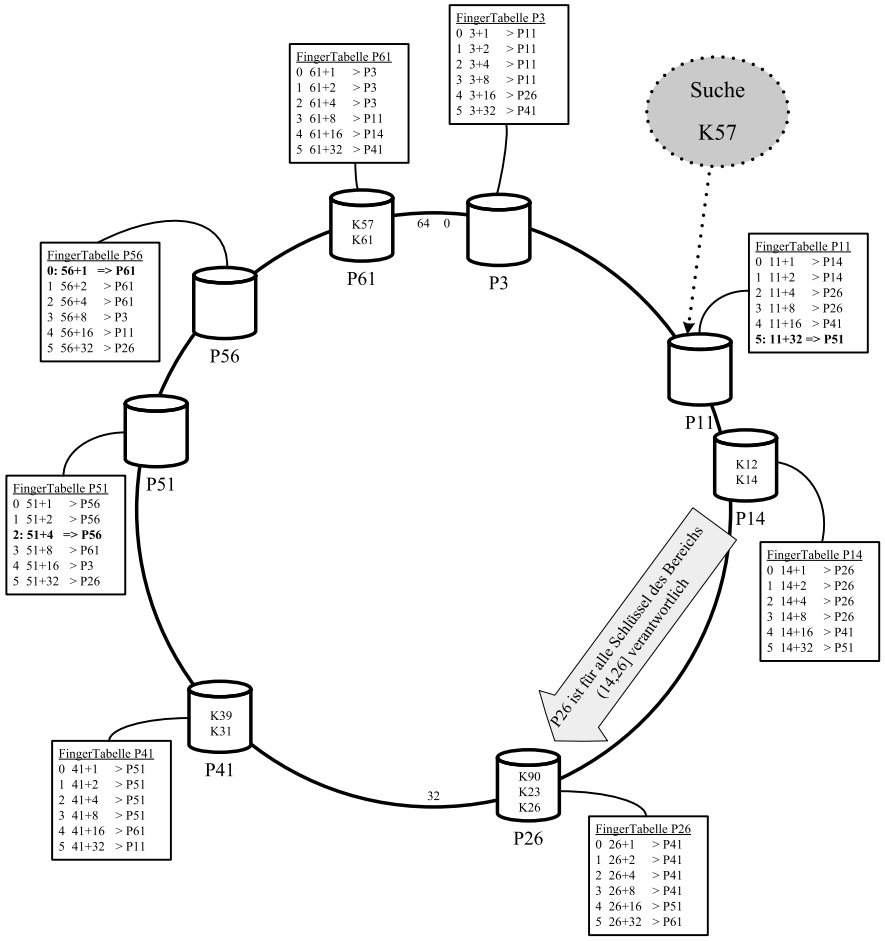

Abbildung 3: Beispiel eines Chord-Overlaynetzwerks.

Vgl. Übungsbuch.
z.z.: Mit jedem Sprung halbiert sich die Anzahl der Stationen.
Beweis durch vollständige Induktion über die Anzahl der Sprünge i.
Induktionsanfang: Sind wir direkt vor dem gesuchten Peer, also $i = 0$, müssen wir nicht mehr springen, es verbleiben $ln(1) = ln(2^0) = 0$ Sprünge.
Induktionsschritt: Vom i-ten zum $i+1$-Sprung halbiert sich die Anzahl verbleibender Stationen von 2^{n-i} nach 2^{n-i-1}
Induktionsvoraussetzung: Nach i Sprüngen reduziert sich die Anzahl verbleibender Stationen auf 2^{n-i}.
Wir wissen, dass sich innerhalb eines Intervalls $[p + 2^i, p + 2^{i+1})$ genau 2^i Stationen befinden, da wir sonst bereits zum nächsten Peer gesprungen wären. Damit haben wir bereits den Abstand auf 2^{n-i} Stationen halbiert (Induktionsvoraussetzung). Jetzt springen wir in das Intervall $[(p+2^i)+2^{i-1}, (p+2^i)+2^i)$ mit 2^{i-1} möglichen Stationen. Für den maximalen Abstand zum Ziel-Peer verbleiben nach dem $i + 1$-ten Sprung maximal $2^{n-(i+1)}$ Stationen, wenn wir $n - (i + 1)$ mal springen. Damit halbiert sich die Zahl verbleibender Sprünge auf $2^{n-(i+1)} = 2^{n-i-1}$

Hausaufgabe 5

Skizzieren Sie die Vorgehensweise beim Hinzufügen eines neuen Peers im Chord Netzwerk. Als Beispiel nehmen Sie die Hinzunahme eines Peers P33 in dem Beispiel-Netzwerk aus Abbildung 3.

Vgl Übungsbuch: Zuerst kontaktiert der neue Peer einen bekannten Peer im Chord Netzwerk. Dieser leitet ihn mittels der Hash-Funktion an den direkten Vorgänger des neuen Peers weiter. Von diesem übernimmt er die Fingertabelle und sein direkter Nachfolger wird ihm bekannt. Von diesem wiederum übernimmt er alle relevanten Daten, für die er nun zuständig ist.
Nach dem Einfügen eines neuen Peers Pk in ein Chord-Netzwerk müssen die Fingertabellen anderer Peers, „die bislang Pj referenziert haben, jetzt möglicherweise auf Pk geändert werden“ [a]. Dies passiert allerdings nicht beim Einfügen, sondern beim Stabilisieren [b]. Der Stabilisierungsalgorithmus wird in 30s Abständen aufgerufen [c], bzw. beim Einfügen eines neuen Peers [d]. Beim Einfügen wird nur der direkte Vorgänger über seinen neuen Nachfolger informiert.

[a] Kemper, Eickler: Datenbanksysteme, 10. Auflage
[b] `https://en.wikipedia.org/wiki/Chord_(peer-to-peer)#Pseudocode`
[c] `http://graal.ens-lyon.fr/~abenoit/reso06/papier/chord1.pdf`
[d] `http://sarwiki.informatik.hu-berlin.de/Chord#Eintritt_neuer_Knoten`

Hausaufgabe 6

Zum CAP-Theorem hieß es in der Vorlesung, dass in verteilten Systemen nur zwei der drei “Wünsche” (Konsistenz, Verfügbarkeit und Partitionstoleranz) gleichzeitig erfüllbar sind.

Welche der drei Kombinationen CA, CP, und AP sind jedoch sehr ähnlich?

Lösung:

Das CAP-Theorem beschreibt das Problem, dass bei einem verteilten System nur zwei der drei wichtigen Eigenschaften Konsistenz (**C**onsistency), Verfügbarkeit (**A**vailability), und Partitionstoleranz (**P**artition tolerance) eingehalten werden können. Dementsprechend sollte es drei verschiedene Arten von Systemen geben:

CA: Konsistent und verfügbar, aber nicht partitionstolerant.

CP: Konsistent und partitionstolerant, aber nicht verfügbar.

AP: Verfügbar und partitionstolerant, aber nicht konsistent.

Wie Daniel Abadi in einem Blogartikel[1] beschreibt, gibt es jedoch keinen praktischen Unterschied zwischen CP- und CA-Systemen. Ein CP-System gibt die Verfügbarkeit auf, wenn das Netzwerk partitioniert ist (der Definition nach könnte es auch nie verfügbar sein; ein solches System macht jedoch wenig Sinn). CA-Systeme tolerieren per Definition Netzwerkpartitionen nicht. Aber was bedeutet das? In der Praxis ist es so, dass auch bei einem CA-System die Verfügbarkeit verloren geht, sobald das Netzwerk partitioniert ist. Auf diese Weise ist sichergestellt, dass bei Wiederherstellung des Netzwerkkonnektivität wieder CA erfüllt werden kann.

Es gibt daher nur zwei Arten von Systemen: CP/CA und AP. Es stellt sich nur die folgende Frage: Wie reagiert das System, wenn das Netzwerk partitioniert wird? Entweder das System gibt die Verfügbarkeit auf (CP/CA) oder die Konsistenz (AP).

Als Schlussbemerkung, sei noch darauf hingewiesen, dass CAP nichts über die Performanz des Systems aussagt. Es kann Sinn machen, sowohl die Verfügbarkeit als auch die Konsistenz einzuschränken, um eine höhere Leistungsfähigkeit zu erreichen (z.B. eine geringere Latenz). Mehr zu diesem Thema findet sich in dem oben erwähnten Blogeintrag von Daniel Abadi.

[1] `http://dbmsmusings.blogspot.de/2010/04/problems-with-cap-and-yahoos-little.html`