

Tutorübung zur Vorlesung Grundlagen Rechnernetze und Verteilte Systeme Übungsblatt 12 (8. Juli – 12. Juli 2013)

Hinweis: Die mit * gekennzeichneten Teilaufgaben sind ohne Kenntnis der Ergebnisse vorhergehender Teilaufgaben lösbar.

Aufgabe 1 Domain Name System (DNS)

Eine zentrale Aufgabe des Domain Name Systems (DNS) ist es, menschenlesbare Namen auf IP-Adressen abzubilden, die dann für die Wegewahl auf der Netzwerkschicht verwendet werden können. Bei dem aus der 3. Programmieraufgabe bekannten Namen `farina.net.in.tum.de` handelt es sich um einen sog. *Fully Qualified Domain Name (FQDN)*.

a)* Benennen Sie *alle* Bestandteile dieses FQDNs.

`farina.net.in.tum.de.` ← ①
⑥ ⑤ ④ ③ ②

1. Root (Beginn des Namensraums)
2. Top Level Domain (TLD)
3. Second Level Domain
4. Third Level Domain
5. Fourth Level Domain
6. Hostname

In Abbildung 1 sind ein PC sowie eine Reihe von DNS-Servern dargestellt. Wir nehmen an, PC1 nutze einen DNS-Server von Google unter der IP-Adresse 8.8.8.8 zur Namensauflösung. Ferner nehmen wir an, dass der Google-Server gerade neu gestartet wurde (also insbesondere keine DNS-Einträge gecached hat) und rekursive Namensauflösung anbietet. Der Server D.ROOT-SERVERS.NET sei ein DNS-Rootserver während F.NIC.de einer der autoritativen Namensserver für de-TLDs ist.

b)* Welche Funktion erfüllen D.ROOT-SERVERS.NET und F.NIC.de?

Der Rootserver ist autoritativ für die Rootzone, d. h. er kennt der DNS-Server, welche für die einzelnen TLDs verantwortlich sind, so z. B. F.NIC.de als einen der autoritativen Namensserver für de-Domains. F.NIC.de kennt wiederum die zuständigen Namensserver für alle Second Level Domains unterhalb der de-TLD.

c)* Für welche Zonen sind die Server ns.tum.de, ns.in.tum.de und ns.net.in.tum.de (vermutlich) autoritativ?

Die FQDNs der Server lassen vermuten, dass sie jeweils für tum.de, in.tum.de und net.in.tum.de verantwortlich sind. Allerdings sollte man nicht aus dem FQDN eines DNS-Servers voreilige Schlüsse über ziehen – der FQDN eines Servers ist i. A. unabhängig von den Domänen, für die ein Server zuständig ist.

d) Zeichnen Sie in Abbildung 1 alle DNS-Nachrichten (Request / Response) ein, die ausgetauscht werden, sobald PC1 auf farina.net.in.tum.de zugreift. Nummerieren Sie die Nachrichten gemäß der Reihenfolge, in der sie zwischen den einzelnen Computern ausgetauscht werden.

s. Abbildung 1.

e) Erklären Sie den Unterschied zwischen iterativer und rekursiver Namensauflösung.

Rekursive Namensauflösung bedeutet, dass eine DNS-Anfrage an einen DNS-Server (oder allgemeiner Resolver) gestellt wird. Dieser wird das endgültige Ergebnis zurücksenden.

Bei iterativer Auflösung hingegen werden schrittweise die autoritativen Namensserver der einzelnen Zonen angefragt.

f)* Wie wird im DNS sichergestellt, dass kein bösartiger DNS-Server Anfragen für andere Domänen beantwortet? (Wir gehen davon aus, dass keine Man-in-the-Middle-Angriffe möglich sind.)

Dies wird lediglich indirekt dadurch sichergestellt, dass während der iterativen Namensauflösung stets nur die jeweils autoritativen Namensserver kontaktiert werden. Sofern die

- Antwort des Rootservers zuverlässig war und
- die Antwort auf dem Weg vom Rootserver zum anfragenden Namensserver nicht modifiziert wurde

kann ein bösartiger Namensserver keine falschen Antworten liefern – eben da er nie gefragt wird. Selbstverständlich wird auf diese Weise nicht verhindert, dass DNS-Antworten mittels Man-in-the-Middle-Attacken abgefangen und modifiziert werden können.

Dagegen helfen lediglich kryptographische Verfahren, wie sie in der DNSSEC-Erweiterung zu finden sind. Dies ist allerdings bis dato kaum verbreitet und auch nicht weiter Bestandteil der Vorlesung.

Der FQDN eines DNS-Servers muss nicht notwendiger Weise mit der Domain korrelieren, für die dieser Server autoritativ ist. Beispiele hierfür sind bereits die Rootserver sowie TLD-Server.

g)* Überprüfen Sie mittels des Kommandozeilen-Tools dig (Linux / OS X) bzw. nslookup (Windows), welche DNS-Server tatsächlich für die Domain tum.de sowie deren Subdomains zuständig sind. **Hinweis:** Bedingen Sie Ihre Abfragen auf den Resource Record Type NS (Name Servers).

dig -t NS in.tum.de liefert beispielsweise folgende Liste von DNS-Servern:

```
;; ANSWER SECTION:
in.tum.de. 84123 IN NS deneb.dfn.de.
in.tum.de. 84123 IN NS tuminfo1.informatik.tu-muenchen.de.
in.tum.de. 84123 IN NS dns2.lrz.de.
in.tum.de. 84123 IN NS tuminfo2.informatik.tu-muenchen.de.
in.tum.de. 84123 IN NS dns1.lrz.de.
in.tum.de. 84123 IN NS dns3.lrz.de.
```

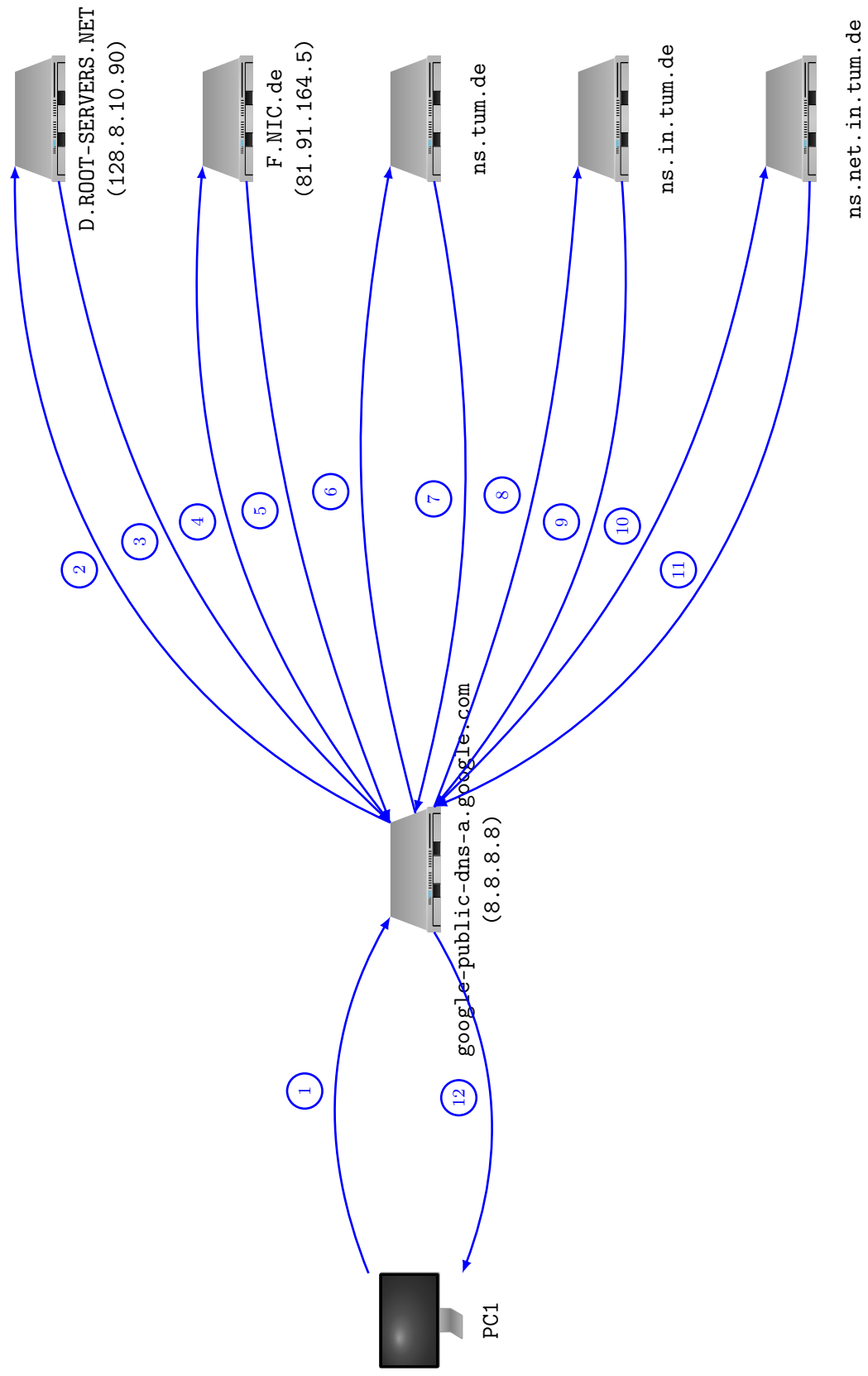


Abbildung 1: Loesungsblatt zu Aufgabe 1.

Aufgabe 2 MapReduce

Wir haben von verschiedenen Verkäufern Preislisten und möchten für jeden Artikel den günstigsten Anbieter und Preis bestimmen. Eine Preisliste ist eine Tabelle, in der in jeder Zeile ein Produktname und ein Preis steht. Am Ende hätten wir gerne eine Tabelle, in der für jeden Produktnamen der günstigste Anbieter und der günstigste¹ Preis steht.

Die gesamte Transformation lässt sich wie folgt formulieren:

$$(\text{Anbieter} \times (\text{Produkt} \times \text{Preis}) \text{ list}) \text{ list} \rightarrow (\text{Produkt} \times (\text{Anbieter} \times \text{Preis}) \text{ list}) \text{ list}$$

Schreiben Sie geeignete Map- und Reduce-Funktionen (z. B. in Pseudo-Code oder mathematischer Notation), die das gewünschte Ergebnis im MapReduce-Framework berechnen. Die Eingabe für die Map-Funktion besteht dabei aus dem Anbieternamen und der Preisliste.

```
Map(Anbieter, Preisliste):  
  foreach (Produkt, Preis) in Preisliste:  
    emitIM(Produkt, (Anbieter, Preis))
```

Kernpunkte:

- über Paare der Liste iterieren
- ein Paar ausgeben, das aus dem Produktnamen und einem Paar aus Anbieter und Preis besteht

```
Reduce(Produkt, AnbieterPreisListe):  
  BesterAnbieter ← empty  
  BesterPreis    ← infinity  
  
  foreach (Anbieter, Preis) in AnbieterPreisListe:  
    if (Preis < BesterPreis):  
      BesterAnbieter ← Anbieter  
      BesterPreis    ← Preis  
  
  emit( (BesterAnbieter, BesterPreis) )
```

Kernpunkte:

- Funktion erwartet Produktname und eine Liste aus Anbieter/Preis-Paaren
- Anbieter/Preis-Paar mit dem Preis-Minimum in der Eingabeliste finden
- am Ende Anbieter/Preis-Paar mit dem Preis-Minimum ausgeben

¹Wir ignorieren den Fall, dass mehrere Verkäufer zum günstigsten Preis anbieten.

Aufgabe 3 Amdahl's Gesetz

Ein Programm soll in einem Cluster mit n Knoten verteilt ausgeführt werden. Das Programm lässt sich in einen Anteil zerlegen, der optimal parallelisierbar ist (p) und einen sequentiellen Anteil $(1 - p)$, der nur auf einem Knoten ausgeführt werden kann.

Mit Hilfe von Amdahl's Gesetz kann man den maximalen Speedup s bestimmen, den dieses Programm auf diesem Cluster erreichen kann:

$$s = \frac{1}{(1 - p) + \frac{p}{n}}$$

Um die maximal erreichbare Effizienz mit anderen Programmen und Clustern erreichen zu können, ist der Speedup relativ zur Gesamtzahl an Knoten $s^* = s/n$ interessanter als der absolute Speedup.

a)* Was ist der Unterschied zwischen Speedup s und relativem Speedup s^* ?

- Der absolute Speedup s gibt an, um welchen Faktor die Ausführung des Programms beschleunigt wird. Für die Dauer des parallel ausgeführten Programms T_p gilt in Abhängigkeit zur Ausführungszeit auf einem einzelnen Rechner T_s : $T_p = T_s/s$
- Der relative Speedup s^* liegt zwischen 0 und 1 und gibt den Anteil der verfügbaren Gesamtleistung an, der maximal genutzt werden kann.

b)* Bestimmen Sie mit Hilfe von Amdahl's Gesetz den Anteil des Programms, der parallelisierbar sein muss (p), um einen bestimmten, relativen Speedup (s^*) zu erreichen!

$$s^* = s/n \quad \text{relativer Speedup}$$
$$\frac{1}{1 - p + \frac{p}{n}} = ns^* \quad \text{Amdahl's Gesetz mit relativem Speedup}$$

... Auflösen nach p ...

$$p = \frac{1}{1 - n} \left(\frac{1}{s^*} - n \right)$$

c) Bestimmen Sie den parallelisierbaren Anteil des Programms für verschiedene relative Speedups und Anzahlen an Knoten:

p	$s^* = 50 \%$	$s^* = 80 \%$
$n = 2$	0%	75%
$n = 10$	89%	97%
$n = 100$	99%	100%

(Werte in Formel aus vorheriger Teilaufgabe einsetzen.)