

Praktikum

Rechnernetze und Verteilte Systeme

Block 5

Zeitsynchronisation

18. – 29.01.2021

Präsenzaufgaben

Die folgenden Aufgaben werden im Termin besprochen.

Aufgabe 1:

Beantworten Sie die folgenden Fragen rund um Zeitsynchronisation:

- (a) Wofür ist die Synchronisation von Uhren erforderlich?
- (b) Wie ist es möglich, Uhren in einem verteilten System exakt zu synchronisieren?
- (c) Welche Alternativen zu der Synchronisation von Uhren haben Sie in der Vorlesung kennengelernt?

Aufgabe 2:

Für die Synchronisation von 4 verteilten Systemen A, B, C und D soll der **Berkeley Algorithmus** verwendet werden. System A hat einen Time Daemon. Zu Beginn des Resynchronisationsintervalls haben die Uhren die folgenden Werte:

A	B	C	D
11550	11570	11515	11525

- (a) Was schickt der Time-Daemon an die Systeme?
- (b) Wie lauten die Antworten der Systeme?
- (c) Der Time-Daemon verwendet die Mittelwertbildung zur Ermittlung der Uhrzeit.

Welchen Wert ermittelt er?

- (d) Welche Werte werden zurück an die Systeme gesendet?

Aufgabe 3:

In einem verteilten System kommt **Cristian's Algorithmus** zum Einsatz, um die Uhren zu synchronisieren. Zu seiner Uhrzeit 10:27:54,0 (Stunden, Minuten, Sekunden) fragt System B bei einem Zeit-Server A nach der Zeit. Um 10:28:01,0 Uhr seiner Zeit empfängt B die Antwort von A mit dem Zeitstempel 10:27:37,5.

- (a) Was ist die Round-trip time (RTT) zwischen B und A?
- (b) Kann man davon ausgehen, dass diese RTT symmetrisch ist?
- (c) Was ist B's Schätzung der Zeit von A?
- (d) Was ist B's Offset in Bezug auf die Zeit von A?
- (e) Geht die Uhr von B zu schnell oder zu langsam?
- (f) Angenommen die Zeit von B läuft zu schnell, was muss bei der Anpassung an die Zeit von A beachtet werden?

Aufgabe 4:

In einem verteilten System kommt **NTP** zum Einsatz, um die Uhren zu synchronisieren.

- (a) Wie viele Zeitstempel verwendet NTP? Was ist der Unterschied zum Cristian's Algorithmus?
- (b) Der Server ist ein Stratum 1 Server. Was bedeutet das?
- (c) Leiten Sie anhand der Zeitstempel t_1 , t_2 , t_3 und t_4 eine allgemeine Formel für die Round-trip time (RTT) (Vorlesung: Delay δ) her. Warum fällt der Offset zwischen den Uhren von Client und Server nicht ins Gewicht?

Praktische Aufgaben

Die praktischen Aufgaben sind in Kleingruppen von i.d.R. 3 Personen zu lösen. Reichen Sie bitte den Quelltext bzw. Lösungen dieser Aufgaben bis zum Mittwoch der zweiten Woche (27.01) 20:00 Uhr per ISIS ein. Aufgaben werden sowohl auf Plagiate als auch auf Richtigkeit hin automatisch getestet, halten Sie sich deshalb genau an das vorgegebene Abgabeformat.

Aufgabe 5:

In dieser Aufgabe sollen Sie einen Datagram-Socket Client schreiben, welcher mit NTP-Servern kommunizieren kann. Wir wollen hier NTP in der gebräuchlichen Version 4 benutzen². Server und Client verwenden bei NTP das gleiche Paketformat, welches wie folgt aussieht:

0	1	2	4	5	7	8	15	16	23	24	31		
LI	VN	Mode	Stratum					Poll					Precision
Root Delay													
Root Dispersion													
Reference ID													
Reference Timestamp (64bit)													
Origin Timestamp (64bit)													
Receive Timestamp (64bit)													
Transmit Timestamp (64bit)													

Dabei können bei einer Anfrage alle Felder mit 0 gefüllt sein, außer der NTP Version (oben VN, hier 4) und dem Mode (Client = 3) Feld. NTP Server sind unter Port 123 mittels UDP zu erreichen. Von vielen Institutionen werden NTP-Server zur Synchronisation angeboten. Das NTP-Pool Projekt sammelt außerdem NTP-Server und stellt

²<https://tools.ietf.org/html/rfc5905%7D>

sie unter einfach zu merkenden DNS-Namen bereit. So nutzt man in Deutschland vorwiegend die Server `0.de.pool.ntp.org`, `1.de.pool.ntp.org`, `2.de.pool.ntp.org` und `3.de.pool.ntp.org` zur Synchronisation. Die Antwort der Server enthält die beiden Zeitstempel, die auf Serverseite genommen werden (*Receive Timestamp* und *Transmit Timestamp*).

Eine weitere Hürde bei der Implementierung auf Unix-artigen Betriebssystemen wie Linux ist der verwendete Zeitstempel. Dieser ist bei NTP die Anzahl von Sekunden seit dem 1.1.1900 und bei Unix die Anzahl der Sekunden seit dem 1.1.1970. Der Offset zwischen den beiden Zeitstempeln ist 2208988800 Sekunden, die zur Umrechnung jeweils addiert bzw. subtrahiert werden müssen. Außerdem ist das Format des NTP-Zeitstempels anders. Hier stellen die 64bit eine Fixpunktzahl dar und sind aufgeteilt in 32bits für die Sekunden und 32bit für die Fraction, also den Nachkommateil.

Sie sollen nun das NTP Protokoll benutzen, um bei verschiedenen Zeitservern die Zeit abzurufen. Der Client soll dabei die folgenden Eigenschaften erfüllen:

- (a) Als Übergabeparameter soll ihr Client die Anzahl der Requests n und eine Liste von NTP Servern akzeptieren: `./ntpclient n server1 server2 server3 ...`
- (b) An jeden Server sollen dann die angegebene Zahl n (zum Beispiel 8) aufeinander folgende Anfragen mit einem Abstand von 8 Sekunden zwischen jeder Anfrage an den gleichen Host gesendet werden.
- (c) Benutzen Sie für die Zeitmessung unbedingt die Funktion `clock_gettime`³. Die Funktion arbeitet mit Zeitstempeln der Form:

```
struct timespec {  
    time_t    tv_sec;    /* seconds */  
    long      tv_nsec;   /* nanoseconds */  
};
```

- (d) Geben Sie den Anfragen pro Server eine fortlaufende Nummer, beginnend mit 0.
- (e) Für jede Anfrage sollte der Delay (RTT/2) und der Offset berechnet werden.

³siehe man `clock_gettime`

- (f) Berechnen Sie außerdem die Dispersion für jeden Server. [^][Für die Anfragen 0-6 genügt es die Dispersion über die bisherigen Anfragen zu berechnen)
- (g) Geben Sie für jede Anfrage eine Zeile mit den folgende Informationen aus:

```
{host};{n};{root_disp};{disp};{delay};{offset}
```

host: Hostname oder IP-Adresse

n: Fortlaufende Nummer der Anfrage

root_disp: Root Dispersion des Servers in Sekunden

disp: Von euch berechnete Dispersion der Verbindung

(lt. VL) in Sekunden

delay: Berechneter Delay in Sekunden

offset: Berechneter Offset in Sekunden

Alle Zeitangaben sind mit min. 6 Nachkommastellen auszugeben

- (h) Die Uhrzeit des Betriebssystems muss **nicht** angepasst werden.
- (i) Da sich ihr Client an den NTP Standard hält, können Sie ihn mit beliebigen öffentlichen Servern testen, z.B. mit dem NTP-Server der TU times.tubit.tu-berlin.de oder mit einem der Server des öffentlichen NTP-Pools: 0.de.pool.ntp.org, 1.de.pool.ntp.org, 2.de.pool.ntp.org und 3.de.pool.ntp.org. Achten Sie darauf, dass Sie nicht zu viele Anfragen in zu kurzer Zeit schicken, da Ihre Anfragen sonst vom Server eine gewisse Zeit ignoriert werden.
- (j) Hinter den Domains der Pool-Server verbergen sich eine Reihe unterschiedlicher NTP-Server. Stellen Sie deshalb sicher, dass sie für wiederholte Anfragen an den gleichen Server die Domain **nicht** erneut auflösen mit getaddrinfo(), da die Funktion dann ggf. eine andere IP zurück liefert.

Aufgabe 6:

In dieser Aufgabe sollen Sie sich nochmals mit den Metriken von NTP anhand ihrer Im-

plementierung vertraut machen. Stellen Sie mit Ihren Client an die vier oben genannten Server aus dem NTP-Pool 100 Anfragen. Plotten sie nun die Ergebnisse in geeigneter Weise. Erzeugen Sie dafür insgesamt **3 Grafiken**:

- eine für den Offset
- eine für den Delay
- und eine für die Root Dispersion.

Stellen Sie die Metriken in Abhängigkeit von der Anfragenummer dar. Interpretieren Sie die Ergebnisse kurz in einem PDF-Dokument (inklusive der Grafiken). Achten Sie in Ihrer Interpretation auch auf mögliche Korrelationen der Werte untereinander.

- (a) Laden Sie **beide Lösungen zusammen** bei ISIS bei der entsprechenden Abgabe hoch. Wie immer sollen Sie in Ihrem Programmier-Projekt aus Aufgabe 5 CMake benutzen. Der folgende Aufruf soll im Unterverzeichnis `build` eine ausführbare Datei mit dem Namen `ntpclient` erstellen:

```
$ mkdir build; cd build
$ cmake .. && make
```

Außerdem sollte eine Datei mit dem Namen `aufgabe6.pdf` enthalten sein. Erstellen Sie aus Ihrer Lösung ein `tar.gz` Archiv, z.B. so:

```
tar -czvf Block5.TXXGYI.tar.gz Block5
```

Vertiefungsaufgaben

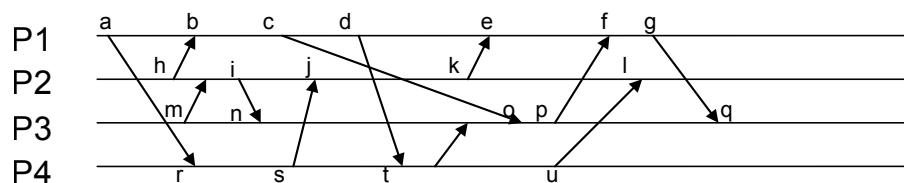
Diese Aufgaben sind zu Ihrer eigenen Vertiefung in Hinblick auf die Klausurvorbereitung gedacht:

Aufgabe 7:

Erläutern Sie (unterstützt durch eine Skizze) die Berechnung des Offsets zwischen zwei das Network Time Protocol (NTP) verwendenden Servern. Was ist jeweils der Hintergrund (z.B. die Motivation, der Ansatz oder die Annahme) Delay und Offset so zu berechnen?

Aufgabe 8:

Gegeben sei der unten stehende Datenaustausch zwischen den Prozessen P1, P2, P3 und P4. Die Prozesse benutzen jeweils logische Uhren. Diese seien jeweils initial mit Null initialisiert; die gestrichelten Pfeile repräsentieren den Nachrichtenaustausch zwischen Komponenten.



Nehmen Sie an, dass die Prozesse Lamport-Uhren benutzen, um sich zu synchronisieren. Geben Sie den Lamport-Zeitstempel für jedes Event im Beispiel. Nehmen Sie an, dass jeder Prozess einen Integer als Lamport-Uhr benutzt.