

## Verteilte Systeme im Sommersemester 2022

Patrick Felschen, Matr. Nr. 932056  
Julian Voß, Matr. Nr. 934505

Osnabrück, 25.03.2022

### Aufgabenblatt 1

#### Aufgabe 1 – Beispiele für verteilte Anwendungen

Beispiel: GIT

| Indikator  | Bewertung   |
|--|---|
| Gemeinsame Nutzung von Ressourcen                      | Jeder Benutzer besitzt eine lokale Kopie des gesamten Repositorys, inklusive der Versionsgeschichte. Über bestimmte Befehle kann das Repository rechnerübergreifend verteilt werden (clone, pull, merge, push). |
| Anbieten von offenen Schnittstellen                    | GIT bietet verschiedene Zugriffsmöglichkeiten (Integrierung in IDE, Terminal, Browser)  |
| Parallele / nebenläufige Ausführung von Aktivitäten    | Über verschiedene Entwicklungszweige (branches) lässt sich ein Projekt parallel bearbeiten und anschließend zusammenführen (merge).   |
| Dynamische Skalierung des Rechenbedarfs                | Je nachdem wie groß Projekte sind, ist mehr oder weniger Rechenleistung nötig   |
| Erhöhte Fehlertoleranz                                 | Im Falle eines Fehlers kann immer auf eine funktionierende Version zurückgesetzt werden (rollback).   |
| Erhöhte Entwicklungskosten                             | Durch Aufgabenteilung in Projekten verringern sich die Entwicklungskosten.  |
| Gefordertes Sicherheitsniveau ist schwerer zu erfüllen | Zugriffsrechte können verwaltet werden, sodass Projekte von Fremdzugriffen geschützt sind.  |
| Höhere Wartungsaufwände                                | Code-Zusammenführung beansprucht Zeit   |
| Verhalten nicht vorhersehbar                           | Dadurch, dass viele Entwickler gleichzeitig auf ein Projekt zugreifen, ist das Verhalten durch Zusammenführung vorhersehbar   |

Beispiele für ein verteiltes System:

- Struktur des Internets
- Telekommunikationsnetze

Beispiele für ein zentralisiertes System:

- Zeitserver

## Aufgabe 2 - Plattform-Unterstützung für Verteilte Systeme

### 1. Unix/Linux-Befehle

a. Erreichbarkeit: `ping 131.173.110.26`

```
pfelsche@id.hsos.de@si0024-025-lin:~$ ping 131.173.110.26
PING 131.173.110.26 (131.173.110.26) 56(84) Bytes Daten.
64 Bytes von 131.173.110.26: icmp_seq=1 ttl=64 Zeit=0.205 ms
64 Bytes von 131.173.110.26: icmp_seq=2 ttl=64 Zeit=0.239 ms
64 Bytes von 131.173.110.26: icmp_seq=3 ttl=64 Zeit=0.229 ms
64 Bytes von 131.173.110.26: icmp_seq=4 ttl=64 Zeit=0.267 ms
64 Bytes von 131.173.110.26: icmp_seq=5 ttl=64 Zeit=0.232 ms
64 Bytes von 131.173.110.26: icmp_seq=6 ttl=64 Zeit=0.233 ms
```

b. Sockets: `ss -tu`

```
pfelsche@id.hsos.de@si0024-025-lin:~$ ss -tu
Netid  State      Recv-Q    Send-Q      Local Address:Port      Peer Address:Port      Process
udp    ESTAB      0          0          127.0.0.1:34003         127.0.0.1:34003
tcp    ESTAB      0          0          131.173.110.25:53340    131.173.251.210:3268
tcp    FIN-WAIT-2 0          0          131.173.110.25:51598    131.173.251.210:kerberos
tcp    ESTAB      0          0          131.173.110.25:ssh      131.173.88.37:1082
tcp    ESTAB      0          0          131.173.110.25:ssh      131.173.62.169:52213
tcp    ESTAB      0          0          131.173.110.25:48410    131.173.251.202:ldap
tcp    ESTAB      0          0          131.173.110.25:ssh      131.173.88.37:51774
tcp    ESTAB      0          0          131.173.110.25:ssh      131.173.62.169:55493
tcp    ESTAB      0          0          131.173.110.25:54042    131.173.251.210:ldap
tcp    ESTAB      0          0          131.173.110.25:1002     131.173.241.14:nfs
tcp    ESTAB      0          0          131.173.110.25:46910    131.173.241.13:microsoft-ds
```

c. Netzwerk-Adapter: `arp -a`

d. Hostname zu IP: `nslookup si0024-025-lin`

```
pfelsche@id.hsos.de@si0024-025-lin:~$ nslookup si0024-025-lin
Server:      127.0.0.53
Address: 127.0.0.53#53

Non-authoritative answer:
Name:   si0024-025-lin.res.hsos.de
Address: 131.173.110.25
Name:   si0024-025-lin.res.hsos.de
```

```
pfelsche@id.hsos.de@si0024-025-lin:~$ nslookup 131.173.110.25
25.110.173.131.in-addr.arpa      name = si0024-025-win.res.hsos.de.
```

### 2. IP-Ports zu Diensten

| Dienst   | Port        | Port-IANA | Funktion                                     |
|----------|-------------|-----------|--|
| sftp     | 2431 / 2433 | 115       | Dateiübertragung über ssh                    |
| ftps     | 990         | 990       | Dateiübertragung über TLS                    |
| ntp      | 123         | 123       | Zeitsynchronisierung                         |
| https    | 443         | 443       | Sicheres Internetkommunikationsprotokoll     |
| echo     | 7           | 7         | Empfangene Daten zum Client zurücksenden     |
| ssh      | 22          | 22        | Sicherer Verbindungsaufbau zwischen Rechnern |
| kerberos | 88          | 88        | Authentifizierungsdienst                     |
| rsync    | 873         | 873       | Synchronisation von Dateien                  |

```
pfelsche@id.hsos.de@si0024-025-lin:~$ grep sftp /etc/services
venus-se 2431/udp          # udp sftp side effect
codasrv-se 2433/udp        # udp sftp side effect
```

### Aufgabe 3 - Entwicklungsumgebung für Verteilte Systeme

1.,2.,3.

| Aufruf  | Funktion                         |
|---|----------------------------------|
| <code>g++ -o daytime daytime.c</code>   | Kompilieren des Skripts          |
| <code>./daytime</code>  | Ausführung des Skripts           |
| <code>CTRL + Z</code><br><code>bg</code>  | Prozess im Hintergrund ausführen |
| <code>telnet 127.0.0.1 9013</code>  | Aufbau einer Telnet-Verbindung   |
| Trying 127.0.0.1...<br>Connected to 127.0.0.1.<br>Escape character is '^]'.<br>Tuesday Mar 29 19:02:57 2022<br>Connection closed by foreign host. | Server Antwort                   |

4.

#### Zeit Server 1:

Aufruf: `telnet time-a-g.nist.gov 13`

Ausgabe: 59667 22-03-29 17:05:07 50 0 0 403.3 UTC(NIST) \*

#### Zeit Server 2:

Aufruf: `telnet time-b-g.nist.gov 13`

Ausgabe: 59667 22-03-29 17:09:39 50 0 0 514.0 UTC(NIST) \*

5.

Bei Aufgabe 3. sind die Latenzzeiten kaum bemerkbar und somit nicht relevant, da sich Client und Server im Gleichen Netzwerk befinden.

Bei Aufgabe 4. müssen Zeiten der Anfrage, Antwort und der Bearbeitung des Servers in die korrekte Zeitberechnung mit einbezogen werden.