

Testování síťových aplikací

Pavel Šimerda
pavlix@pavlix.net

LinuxDays 2016, Praha

<http://data.pavlix.net/linuxdays/2016/>

Na úvod

Problémy

Nástroje

Klíent-server

Diskuze

Na úvod

Na úvod

Problémy

Nástroje

Klíent-server

Diskuze

- Testování síťových aplikací
- Chování klientů a serverů
- Dualstack

Proč o tom mluvím

Na úvod

Problémy

Nástroje

Klíent-server

Diskuze

- Spravuju síťové RPM balíčky
- Analyzuju různé druhy chyb
- Řeším podporu síťových protokolů
- Zabývám se konfigurací sítě

Chyby a nedostatky síťových aplikací

Integrace s operačním systémem

- Řazení služeb při bootování
- Nuance síťového API
- Souběh instancí

Server na dualstacku

- Schopnost přijímat IPv4 i IPv6 spojení
- Zpracování informací o IPv4/IPv6 klientech

Klíent na dualstacku

- Preference protokolu IPv6
- Čekání na fallback na IPv4
- Řešení neúspěšných spojení

Na úvod

Problémy

Nástroje

Klíent-server

Diskuze

Nástroje

Ladící výpisy

- Lehce dostupné v Pythonu
- Podobně v C spustitelném z pracovního adresáře
- Ale mnohdy nepohodlné

Sledování provozu na síti

- Pomocí tcpdump
- Možnost zápisu do souboru
- Mnohdy neobjasní interní fungování programu

Core dump analysis (gdb)

Na úvod

Problémy

Nástroje

Klíent-server

Diskuze

- Vyžaduje přesné verze binárek
- Řeší především pády nebo cykly
- Event loop kazí výstup
- Často toho moc neprozradí

Interaktivní debugging

- GNU debugger
- Python debugger
- Dobře funguje s reproducerem
- Lze skriptovat

LD_PRELOAD

- Umožňuje nahradit konkrétní symboly
- Dobře se automatizuje
- Výhodná hotová řešení

ptrace, strace, python-pttrace

- Jednoduchý výpis syscallů
- Sofistikované zpracování syscallů
- Sledování socketů

Na úvod

Problémy

Nástroje

Klient-server

Diskuze

Klient-server aplikace

Systemová volání - klient

- `socket()`
- `connect()`
- `recv()`, `recvfrom()`
- `send()`, `sendto()`

Systemová volání - server

- `socket()`
- `bind()`, `listen()`
- `recv()`, `recvfrom()`
- `send()`, `sendto()`

Systemová volání - špeky

- IP_FREEBIND
- IPV6_V6ONLY
- SO_REUSEADDR
- SO_REUSEPORT

Definované chování – sekvenční connect s preferencí

Na úvod

Problémy

Nástroje

Klient-server

Diskuze

- Blokující socket pro první IPv6
- Krátký timeout (15s)
- Blokující socket pro první IPv4
- Dále obdobně

Definované chování – paralelní connect s preferencí

- Neblokující socket na IPv4 a IPv6
- Krátký timeout pro IPv6 (100ms)
- Fallback na IPv4

Definované chování – paralelní connect s preferencí

- Neblokující socket na IPv4 a IPv6
- Krátký timeout pro IPv6 (100ms)
- Fallback na IPv4

Testovací framework a sada testů

- <https://github.com/pavlix/network-testing>
- Sada klientských a serverových skriptů
- Testovací framework v Pythonu
- Debugging pomocí python-pttrace
- Separace pomocí network namespaces

Na úvod

Problémy

Nástroje

Klient-server

Diskuze

- <https://github.com/pavlix/netresolve>
- Neblokující režim (fd)
- Flexibilní request/response
- Ukázková implementace high-level socket API
- Drop-in replacement using LD_PRELOAD

Diskuze...

<http://data.pavlix.net/linuxdays/2016/>

pavlix@pavlix.net

<https://github.com/pavlix/network-testing>