

VSY

Contents

1	Interaktionssemantik	2
1.1	Semantika	2
2	Server - Klassifikation	3
2.1	Servertypen	3
2.2	Server Zustand	3
3	Caching	4
3.1	Client-Caching	4
3.1.1	zentrale Kontrollinstanz	4
3.2	Broker, Trader, Proxy, Balancer, Agent	4
4	Peer-to-Peer Netzwerke	5
4.1	unstrukturiert: zentralisiert, pur, hybrid	5
4.2	strukturiert: DHT-basiert	5

1 Interaktionssemantik

blockierend / synchron: Der Client wartet nach Absenden der Anforderung auf die Antwort des Servers.

nicht blockierend / asynchron: Der Client verschickt die Anforderung, arbeitet sofort weiter und nimmt irgendwann später die Antwort entgegen.

iterativer Server bearbeitet zunächst die Anfrage, sendet die Rückantwort und wartet erst dann auf die nächste Anfrage.

paralleler Server startet nach dem Eintreffen einer Anfrage einen neuen Prozess/Thread zur Bearbeitung der Anfrage. Der Thread sendet die Rückantwort, der Hauptprozess steht sofort zur Annahme weiterer Anfragen zur Verfügung.

1.1 Semantika

at-least-once Semantik:

- Der Sendeprozess wartet, bis die Rückantwort innerhalb einer gewissen Zeit eintrifft. Ein Überschreiten der Zeitschranke führt zu einem erneuten Verschicken der Nachricht und Setzen der Zeitschranke.
- Schlagen Versuche mehrmals fehl, ist im Moment kein Senden möglich. Vielleicht ist die Leitung gestört, der Adressat nicht empfangsbereit, ...
- Erhält der Empfänger durch mehrfaches Senden dieselbe Nachricht mehrmals, kann durch erneute Bearbeitung sichergestellt werden, dass die Anforderung mindestens einmal bearbeitet wird.

at-most-once Semantik:

- Der Empfänger verwaltet eine Anforderungsliste, die die bisher gesendeten Anfragen enthält.
- Anfragen werden mit Hilfe der Liste überprüft:
 - Steht die Anfrage bereits in der Liste, so ging die Rückantwort verloren und die Rückantwort wird nochmal gesendet.
 - Andernfalls wird die Anfrage in der Liste vermerkt, die Anfrage bearbeitet und die Rückantwort gesendet.
- Wurde die Rückantwort bestätigt, wird die zugehörige Anfrage aus der Liste gelöscht.
- Bei vielen Clients und Anfragen muss die Historie nach einiger Zeit gelöscht werden, um den Speicherbedarf zu beschränken.

exactly-once Semantik: Um Systemfehler wie einen Plattenausfall auszuschalten, wird die Anforderungsliste in einem stabilen Speicher gehalten. So wird aus der at-most-once-Semantik die exactly-once-Semantik.

2 Server - Klassifikation

2.1 Servertypen

Unterscheidung nach Anzahl der Dienste pro Server:

- **shared Server** stellt mehrere Dienste bereit, für jeden Service/Dienst steht ein eigener Prozess zur Verfügung. Bei einer Anfrage wird zunächst getestet, ob der Prozess aktiv ist und ggf. wird ein neuer Prozess gestartet. Ein solcher Prozess kann zur Erledigung der Anforderung einen Thread starten.
- **unshared Server** stellt nur einen Dienst bereit, verschiedene Dienste liegen auf unterschiedlichen Servern, nicht notwendigerweise auf verschiedenen Rechnern.

Unterscheidung anhand der Aktivierung:

- **per request Server** startet jedesmal neu, wenn eine Anfrage eintrifft. Mehrere Prozesse für den gleichen Dienst können konkurrenzaktiv sein.
- **persistent Server** wird beim Starten (Boot-Vorgang) des Systems oder beim Start der Server-Applikation aktiv. Client-Anfragen werden direkt an den Server-Prozess weitergeleitet oder bei nicht gestartetem Server vom BS mit einer Fehlermeldung beantwortet.

2.2 Server Zustand

zustandsinvariant Server liefern Informationen, die sich zwar ändern können, die aber unabhängig von Client-Anfragen sind. Beispiele: Web-, Name- und Time-Server. Die Reihenfolge der Anfragen spielt keine Rolle

zustandsändernd Server wechseln zwischen 2 Client-Anfragen ihren Zustand, wodurch evtl. Anforderungen von Clients nicht mehr erfüllt werden können (z.B. File-Server: löschen einer Datei). Die Reihenfolge der Anfragen ist zur Erledigung der Aufgaben entscheidend.

- **zustandsspeichernd** (stateful) Server speichern Zustände in internen Zustandstabellen (Gedächtnis).
 - Vorteile
 - * Clients müssen den Zustand nicht in jeder Anfrage mitteilen, die Netzlast wird also reduziert.
 - * Der Server kann auf zukünftige Anfragen schließen und entsprechende Operationen im Voraus durchführen.
 - * Informationen sind gegen Ausspähen schützbar.
 - Nachteil: Hoher Speicherbedarf beim Server.
- **zustandslos** (stateless) Server müssen vom Client bei jeder Anfrage die komplette Zustandsinformation erhalten, um die Anforderung korrekt erfüllen zu können. stateless server -> statefull client
 - Vorteile: Nach einem Absturz des Servers muss der alte Zustand nicht wiederhergestellt werden, siehe at-most-once-Semantik. Der Client schickt seine Anfrage einfach noch einmal.
 - Nachteil: Es müssen evtl. sensitive Daten übertragen werden.

3 Caching

3.1 Client-Caching

write-through Ändert ein Client einen Cache-Eintrag, so wird der neue Wert auch an den Server geschickt, damit der Server die Änderung an der Originaldatei nachvollziehen kann.

delayed write Zur Reduktion des Netzverkehrs werden mehrere Änderungen gesammelt und erst nach Ablauf eines Zeitintervalls (bspw. 30s bei SUN NFS) an den Server geschickt.

write on close Weitere Reduktion, wenn Änderungen erst beim Schließen der Datei an Server geschickt werden.

Probleme:

- Nicht fehlertolerant bei Client-Crash: Alle noch nicht geschriebenen Daten gehen verloren.
- Bei delayed write und write on close bekommt ein dritter Client die Änderungen von Client 1 und 2 ggf. nicht mit.
- ie Konsistenz zwischen Client- und Server-Daten ist nicht sichergestellt. Beispiel: Client 1 und 2 löschen aus derselben Datei jeden zweiten Datensatz. Ergebnis?

Lösung:

3.1.1 zentrale Kontrollinstanz

- Der Server tabelliert, welcher Client welche Datei zum Lesen oder Schreiben geöffnet hat:
 - Eine zum Lesen geöffnete Datei kann anderen Clients zum Lesen bereitgestellt werden.
 - Der Zugriff auf eine zum Schreiben geöffnete Datei wird anderen Clients untersagt. Die Anfrage wird entweder verweigert oder in eine Warteschlange gestellt.
- Sobald ein Client eine Datei zum Schreiben öffnet, schickt der Server allen lesenden Clients eine Nachricht zum Löschen der Cache-Einträge: Schreib-/Lesezugriffe gehen ab dann direkt zum Server, der Cache wird quasi ausgeschaltet. ⇒ Leser und Schreiber können parallel weiterarbeiten, aber die Netzlast steigt aufgrund des ausgeschalteten Caches.

3.2 Broker, Trader, Proxy, Balancer, Agent

Proxy Stellvertreter für mehrere Server. Oft wird Clients der Zugriff auf externe Server untersagt, nur über den Proxy kann mit der Außenwelt kommuniziert werden.

Broker Vermittelt zwischen Client und Server: Welcher Server stellt den gesuchten Dienst zur Verfügung?

- **intermediate oder forwarding** Die Client-Anfrage wird vom Broker angenommen und an den betroffenen Server übergeben. Der Broker nimmt die Rückantwort des Servers an und leitet die Antwort an den Client weiter.
- **seperate oder handle-driven** Der Broker gibt die ermittelten Server-Informationen an den Client zurück. Anschließend kommuniziert der Client direkt mit dem entsprechenden Server.
- **hybrider** unterstützt beide genannten Versionen. Der Client gibt an, welches Modell er wünscht.

Trader Sucht den am besten geeigneten Server heraus: Welcher Zeit-Server hat die geforderte Genauigkeit?

Balancer Spezieller Trader, der die Arbeitslast auf verschiedene Server verteilt.

Agent Koordiniert mehrere Server-Anfragen für einen Client: Durch eine Suche über verschiedene Web-Seiten findet der Agent z.B. den günstigsten Anbieter eines gesuchten Produkts.

4 Peer-to-Peer Netzwerke

4.1 unstrukturiert: zentralisiert, pur, hybrid

4.2 strukturiert: DHT-basiert