

Übungsblatt 5

zur Vorlesung

Verteilte Systeme/Ubiquitous Computing

Wintersemester 2007/2008

Achtung: Bitte beachten Sie auch die Web-Site der Vorlesung:
<http://www.mobile.ifi.lmu.de/Vorlesungen/ws0708/vs/>

Aufgabe 15: (H) Dienstvermittlungssysteme in der Praxis

- Stellen Sie die Dienstvermittlung innerhalb des “Jini” Systems der Firma Sun Microsystems vor.
- Beschreiben Sie das Grundkonzept von “Universal Plug and Play” (UPnP).
- Erklären Sie die Funktionalität des “Service Location Protocol” (SLP), das von der IETF in den RFCs 2165, 2608, 2609 und 3224 spezifiziert wurde.

Antwort:
siehe Folien.

Aufgabe 16: (T) Kontextverteilung

- Ein kontextsensitives System besteht aus Kontextquellen (KQ), kontextsensitiven Diensten (KSD) und Nutzern. Diese können jeweils dicht bei einander sein oder weit aus einander. Nehmen Sie an, es gäbe jeweils eine Kontextquelle, einen kontextsensitiven Dienst und einen Nutzer. Welche Kombinationen von Nähe und Entfernung zwischen den drei Entitäten gibt es?
- Bei der Kontextverteilung ist wichtig, dass ein kontextsensitiver Dienst alle relevanten Kontextinformationen erhält und nur diese. Welche Arten von Relevanz kennen Sie? Erklären Sie diese.
- Was ist “localized scalability”? Wie müssen Kontextquellen, der kontextsensitive Dienst und die Nutzer zu einander angeordnet sein, damit “localized scalability” vorliegt?
- Welche Formen der Kontextbereitstellung kennen Sie?
- Beschreiben Sie, wie skalierbare Kontextverteilungsplattformen aufgebaut sind.

Antwort:

- a.
 - (i) KQ, KSD und Nutzer nahe bei einander.
 - (ii) KQ und Nutzer nahe bei einander. KSD entfernt.
 - (iii) KSD und Nutzer nahe bei einander. KQ entfernt.
 - (iv) KQ und KSD nahe bei einander. Nutzer entfernt.
 - (v) KQ, KSD und Nutzer sind weit aus einander.
- b. Es gibt zeitliche, räumliche und inhaltliche Relevanz. Um inhaltliche Relevanz zu erhalten, muss eine Antwort in semantischer Nähe zu einer Anfrage sein. Außerdem müssen Redundanzen bei der Antwort vermieden werden.
- c. "Localized scalability" wird erreicht, indem Interaktionen zwischen weit entfernten Entitäten weitest gehend vermieden werden (Definition von Satyanarayanan) oder aber indem die Verteilung von Kontextinformationen lokal beschränkt bleibt (Definition von Schmidt). Damit "localized scalability" vorliegen kann, müssen KQ, KSD und Nutzer nahe bei einander sein.
- d. Es gibt infrastrukturlose und infrastrukturgebundene Kontextbereitstellung. Bei den infrastrukturlosen Verfahren greifen die KSD entweder direkt auf KQ über drahtlose Verbindungen zu, oder aber es wird ein Multihop-Verfahren benutzt. Bei den Multihop-Ansätzen kooperieren entweder die KQ (Sensornetze), oder aber die KSD (Datendiffusion), um Kontextinformationen zu verteilen. Bei den infrastrukturgebundenen Systemen kann man zwischen eng und lose gekoppelten Ansätzen unterscheiden. Die lose gekoppelten Ansätze können in dienstzentrische und datenzentrische Infrastrukturen unterschieden werden.
- e. Skalierbare Kontextverteilungsplattformen besitzen jeweils eine physikalische Schicht, eine Datenschicht, eine Routingschicht und eine Anfragschicht.
 In der physikalischen Schicht befinden sich die Sensoren und andere Entitäten wie Nutzer. Die Sensoren und Nutzer werden jeweils lokalen Servern in der Datenschicht zugeordnet. Die lokalen Server fungieren entweder als eine Art Datenbehälter (Tuple Space oder XML Datenbank) oder als eine Ausführungsplattform für anwendungsspezifische Code-Teile. Die Zuordnung von Entitäten zu lokalen Servern erfolgt entweder aufgrund von geographischer Nähe oder aber die lokalen Server werden Repräsentant einer einzigen Entität. Z.B. würde so ein Server alle Informationen zu einer Person oder einem Objekt beinhalten.
 Skalierbare Kontextverteilungsplattformen müssen nun fähig sein, Anfragen zu beantworten, die Daten von mehreren lokalen Servern benötigen. Hierfür muss eine Anfrage in Unteranfragen zerlegt werden und diese müssen an die entsprechenden lokalen Server gesendet werden. Diese Funktion übernimmt die Routingschicht. Verwendete Technologien hierbei sind Peer-to-Peer-basiertes Routing, das Domain Name System oder aber auch XML Hyperlinks.
 Die Anfragschicht ist die Schnittstelle zwischen den kontextsensitiven Diensten und den Kontextverteilungsplattformen. Die kontextsensitiven Dienste stellen die Anfragen entweder in XPath, sie kodieren in XML mehrere Attribute-Werte-Paare oder aber sie spezifizieren einen Operatorgraphen in XML.

Aufgabe 17: (H) Uhren

$C_r(t)$ sei die Zeit zur UTC-Zeit t auf einem Rechner r . Der Hersteller der Uhr gibt die maximale Abweichung p der Uhr mit 10^{-12} an. Am Anfang sei $C_r(t) = t$. Die Rechneruhr laufe maximal, d.h. mit p , zu schnell. Wie viel geht sie nach einem Jahr vor?

Antwort:

Ein Jahr hat rund $60 * 60 * 24 * 365 = 31536000$ Sekunden.

Abweichung nach einem Jahr = $31536000 \text{ Sekunden} * 10^{-12} = 31,536 \text{ Mikrosekunden}$.

Aufgabe 18: (H) Uhrensynchronisation

Gegeben seien drei Rechner A, B und C. Die Zeit wird in dem Format Stunden:Minuten:Sekunden, Millisekunden angegeben.

- a. Rechner A verfügt über die physikalische Uhrzeit (UTC-Zeit). Zum Zeitpunkt 10:05:02,081 sende Rechner B eine Anfrage an A nach der UTC-Zeit. Zum Zeitpunkt 10:05:02,093 auf B kommt die Antwort von A mit dem Wert 10:05:01,120 an. Es werde angenommen, dass Rechner A 2 Millisekunden für die Bearbeitung benötigt und Rechner B in vernachlässigbarer Zeit nach dem Erhalt des Ergebnisses seine interne Uhr korrigieren kann. Auf welche Zeit muss Rechner B nach dem Algorithmus von Christian gestellt werden, und wie wird die Uhrenaktualisierung realisiert?
- b. Rechner A fungiert als Zeitdämon nach dem Berkeley-Verfahren. Rechner A habe die Zeit 10:05:02,000, Rechner B 10:05:02,500 und Rechner C 10:05:01,800. Stellen sie die Prozesse dar, die nach dem Berkeley-Verfahren ablaufen.
- c. Nehmen Sie an, Rechner A will von Rechner B die Uhrzeit nach dem Network Time Protocol erfragen. Er sendet um 10:05:03,000 eine Nachricht los und empfängt eine Antwort um 10:05:04,000. Diese Antwort gibt an, dass Rechner B die Nachricht um 10:05:04,100 empfangen und um 10:05:04,700 gesendet hat. Wie groß ist der Zeitversatz σ zwischen Rechner A und B? Nehmen Sie an, dass die Zeiten für den Hin- und Rückweg gleich sind.
- d. Was ist eine logische Uhr?
- e. Wie funktioniert der Algorithmus von Lamport?

Antwort:

$$a. \quad t = C_{UTC} + \frac{T_1 - T_0 - I}{2}$$

t = neue Zeit

C_{UTC} = erfragte UTC-Zeit

T_1 = Ankunftszeit der Anfrage

T_0 = Absendezeit der Anfrage

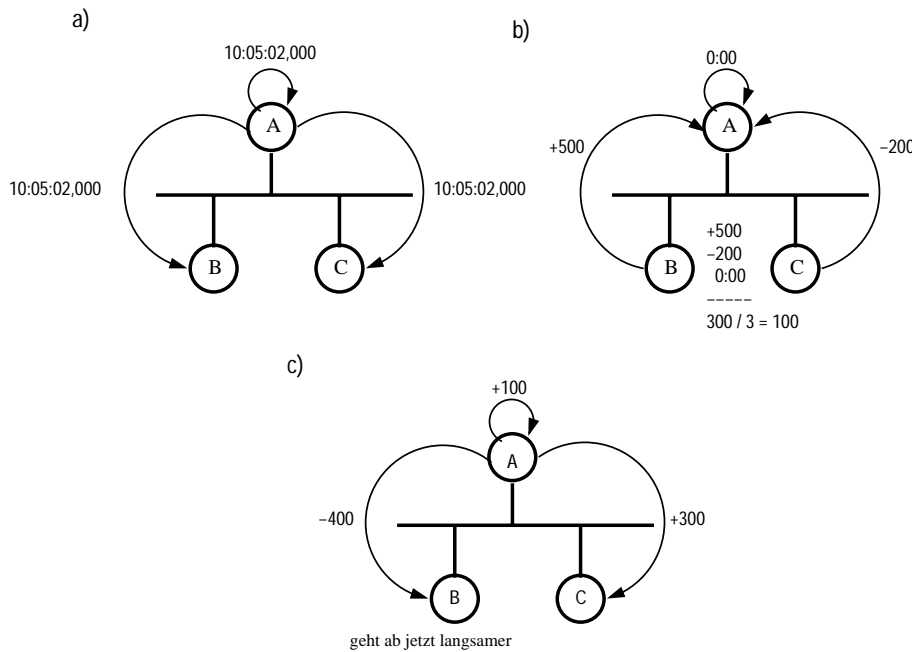
I = Bearbeitungszeit am Zeitserver

Einsetzen ergibt:

$$10 : 05 : 01, 120 + \frac{10:05:02,093 - 10:05:02,081 - 0,002}{2} = 10 : 05 : 01, 125$$

Da zwei aufeinander folgende Ereignisse auch gemäß der Uhr auf einer Maschine hintereinander zu registrieren sind, kann eine Uhr nicht einfach zurückgestellt werden. Deswegen wird die errechnete Zeit genommen und weiter gezählt. Die Zeitählung auf der Maschine wird verlangsamt, bis die fortgeschriebene neue Zeit erreicht ist.

b.



c.

$$T_1 = 10 : 05 : 03,000$$

$$T_2 = 10 : 05 : 04,100$$

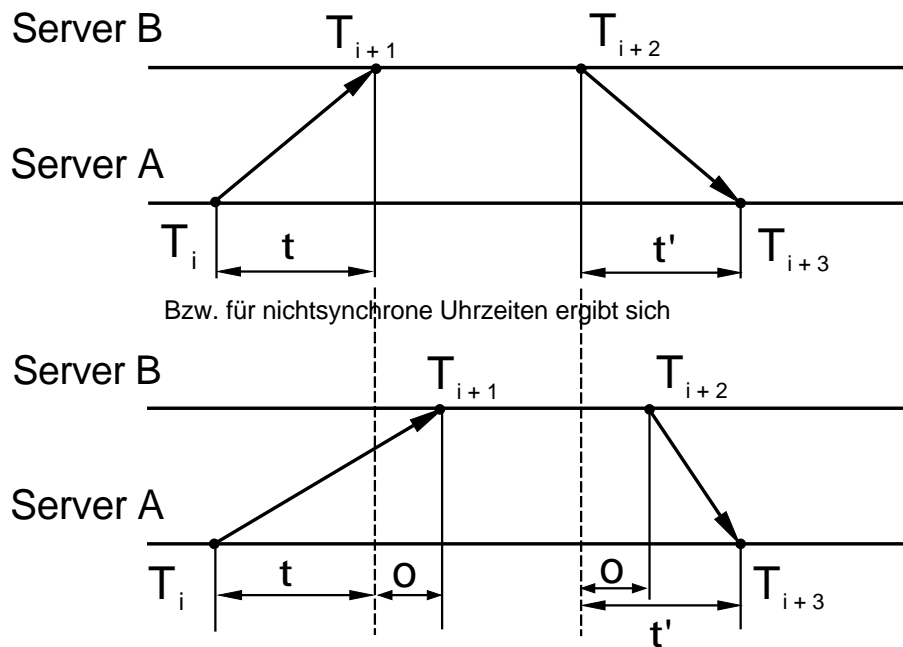
$$T_3 = 10 : 05 : 04,700$$

$$T_4 = 10 : 05 : 04,000$$

$$T_2 = T_1 + t + \sigma$$

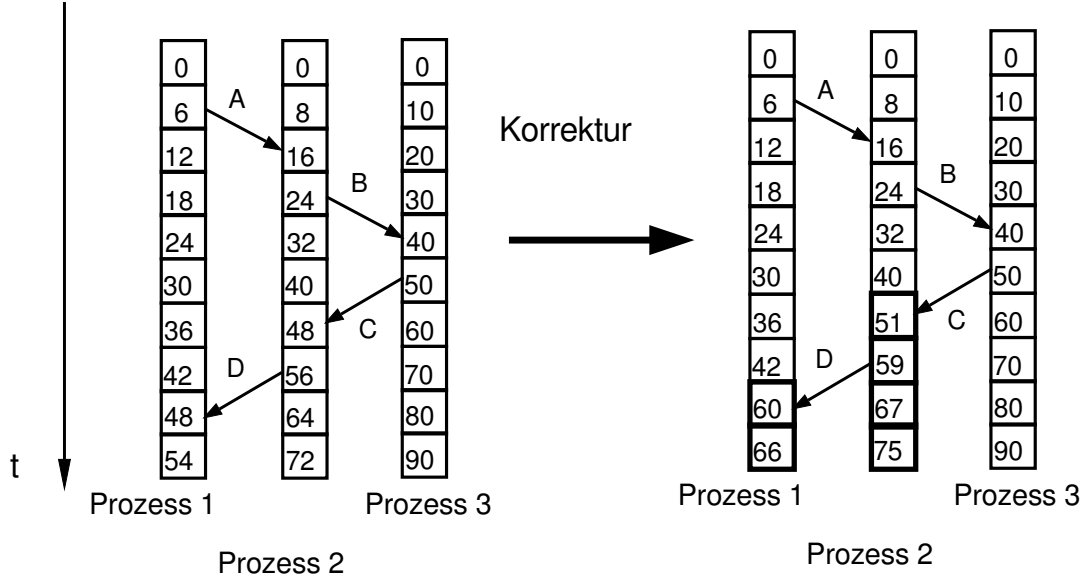
$$T_4 = T_3 + t - \sigma$$

$$\sigma = \frac{T_2 - T_1 + T_3 - T_4}{2} = 900 \text{ Millisekunden}$$



- d. Eine logische Uhr ermöglicht es, die zeitliche Reihenfolge von Ereignissen korrekt zu bestimmen. Dafür müssen sich die Rechner eines Verteilten Systems untereinander auf eine einheitliche Zeit verständigen. Die Zeit muss nicht mit der physischen Zeit identisch sein.
- e. Jeder Nachricht wird beim Versenden die lokale Zeit mitgegeben. Erhält ein Empfänger eine Nachricht, die einen Zeitstempel trägt, der vor der Zeit der lokalen Uhr des Empfängers liegt, so wird die lokale Uhr auf eine Zeit gestellt, die dem Zeitstempel der Nachricht + 1 entspricht.

Beispiel:



Aufgabe 19: (H) Wechselseitiger Ausschluss

- a. Welche Vor- und Nachteile hat ein zentraler Ansatz zur Realisierung eines wechselseitigen Ausschlusses?
- b. Gegeben seien wieder drei Rechner A, B und C. Alle drei Rechner haben ihre Uhrzeiten nun um eine totale Ordnung der Ereignisse erweitert und wollen auf ein gemeinsam genutztes Betriebsmittel zugreifen. Dabei sei der wechselseitige Ausschluss nach dem verteilten Algorithmus von Ricart und Agrawala organisiert. Gleichzeitig senden die Rechner folgende Anforderungen:

A: 10:06:03, B: 10:06:00, C: 10:06:07

Beschreiben Sie die zeitliche Reihenfolge der auftretenden Kommunikationsabläufe und rechnerinternen Entscheidungen, bis alle drei Rechner Zugriff auf das Betriebsmittel hatten. Kennzeichnen Sie auch die in den Warteschlangen gespeicherten Informationen, und markieren Sie aktive Rechner durch z.B. fette Kreisumrandungen.

- c. Welche Vor- und Nachteile hat der Algorithmus von Ricart und Agrawala?
- d. Gegeben sei folgende Situation: Rechner A wartet auf Rechner B. Rechner B wartet auf Rechner C und Rechner C wartet auf Rechner A. Wie würde ein solcher Deadlock mit dem Algorithmus von Chandy-Misra-Haas erkannt werden?

Antwort:

a. Vorteile:

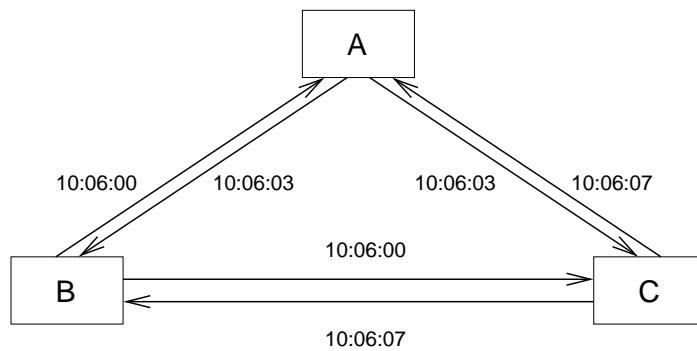
- (i) Fairness
- (ii) einfache Implementierung

Nachteile:

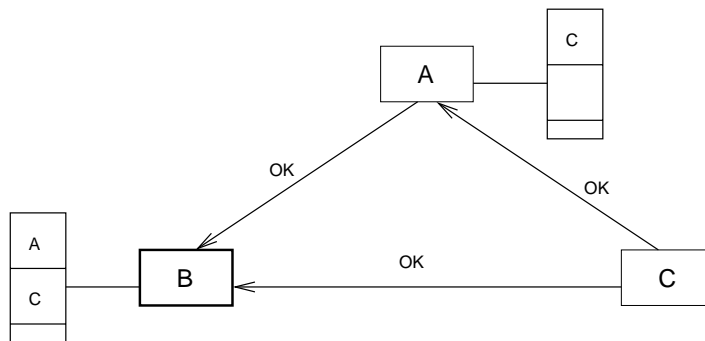
- (i) Zentraler Koordinator ist ein Engpass ...
- (ii) ... und eine zentrale Fehlerstelle.

b.

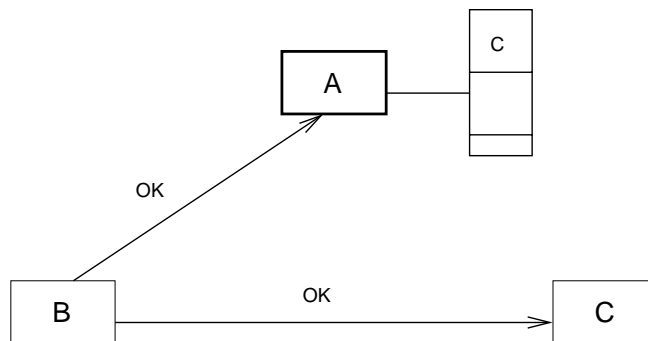
(i)



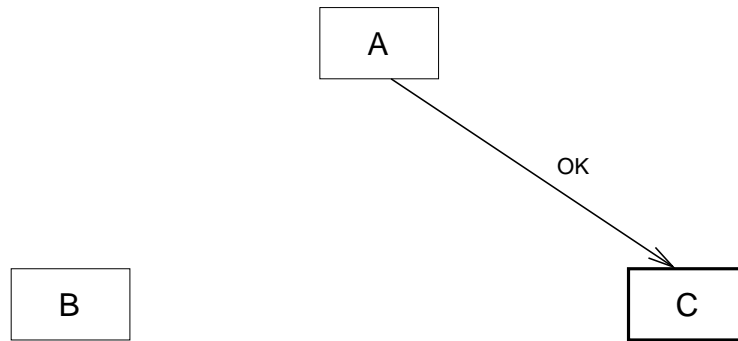
(ii)



(iii)



(iv)



c. Vorteile:

- (i) Fairness
- (ii) Deadlockfreiheit.

Nachteile:

- (i) Netzwerkbelastung sehr hoch. $2(n-1)$ Nachrichten sind nötig bei n Komponenten.
- (ii) Ausfall einer Komponente kann zu langem Warten führen.

d. Eine Untersuchungsnachricht wird an alle Prozesse gesendet, die das Betriebsmittel belegen, auf das der Prozess wartet. Die Untersuchungsnachricht enthält:

- (i) einen Bezeichner für den wartenden Prozess
- (ii) einen Bezeichner für den sendenden Prozess
- (iii) den Bezeichner des Prozesses, an den die Nachricht gesendet wird.

Diese Nachricht wird vom Empfänger an alle Prozesse weiter geleitet, die eine Ressource belegen, auf die wiederum der Empfänger wartet.

Erhält ein Prozess eine Untersuchungsnachricht, in der er als wartender Prozess aufgeführt wird, so existiert ein Deadlock.

Konkreter Ablauf für das Beispiel:

