

Resümee (1)

- Institut für Pervasive Computing
 - Prof. Gustavo Alonso
 - Prof. Friedemann Mattern
 - Prof. Roger Wattenhofer
 - Master in CS, Major in Distributed Systems
 - Organisatorisches
 - "Labor" (= internes Praktikum)
-

- Einordnung der Vorlesung
- Verteilte Systeme: Begriff, Sichtweisen, Eigenschaften...
- Motivation; Gründe für verteilte Systeme
 - Kooperation von geographisch verteilten Einheiten
 - qualitatives Wachstum des Internet
 - Beispiel-Anwendungsszenario einer virtuell verteilten Bibliothek: Anforderungen an eine Lösung
- Middleware für das Internet

Resümee (2)

- Transparenzeigenschaften (Verbergen von Verteiltheit etc.)
 - Charakteristika und praktische Problembereiche verteilter Systeme
 - Historische Entwicklung von Systemen und Konzepten
 - Phänomene und konzeptionelle Probleme
 - Schnappschussproblem (inkonsistente globale Sicht)
 - Phantom-Deadlocks
 - Uhrensynchronisation
 - kausal inkonsistente Beobachtungen
 - Geheimnisaustausch über unsicheren Kanal
-

- Multiprozessoren (gemeinsamer Speicher)
 - Buskoppelung
 - Schaltnetz-koppelung (Crossbar, Permutationsnetze)
- Multicomputer (verteilter Speicher)
 - Transputer
 - Bewertungskriterien für Verbindungstopologien
 - Hypercube (rekursives Konstruktionsprinzip)
- Mehrrechner-Verbindungstopologien
 - Torus
 - Cube Connected Cycle
 - Zufallstopologien

Resümee (3)

- Nachrichtenkommunikation
 - Message-passing-Systeme / -Bibliotheken
 - Prioritäten von Nachrichten
 - Zuverlässigkeitsgrade
- Fehlermodelle
 - fehlerhaftes Senden / Empfangen
 - Verlust von Nachrichten
 - crash, fail-stop
 - byzantinische Fehler
- Kommunikationsmuster
 - Mitteilung <--> Auftrag
 - synchron <--> asynchron
- Synchroner Kommunikation
 - Definition
 - Realisierung
 - virtuelle Gleichzeitigkeit; Gummibandtransformation
 - Blockaden und Deadlocks
- Asynchrone Kommunikation
 - Vor- / Nachteile gegenüber synchroner Kommunikation

Resümee (4)

- Synchroner Kommunikation mit asynchroner simulieren
 - Warten auf ein explizites Acknowledgement
- Asynchrone Kommunikation mit synchroner simulieren
 - Puffer(prozess!) zur Entkoppelung dazwischenschalten
- Implementierung von Pufferprozessen
 - durch Inversion der Kommunikationsbeziehung
- Puffer beschränkter Kapazität
 - Implementierungsaspekte
- Alternatives Empfangen von Nachrichten
 - "select"-Anweisung: elegantes und mächtiges Konstrukt
 - aber: Semantik genau festlegen
- Verschiedene Kommunikationsmuster
 - no-wait-send; RPC; remote service invocation; rendezvous
- Datagramm
- Rendezvous-Protokoll
- RPC
 - Implementierung
 - Parameter-Marshalling
 - Stubs
 - Transparenzproblematik
- RPC-Fehlerproblematik

Resümee (5)

- RPC-Fehlerproblematik
 - Fehlerursachen (verlorene Nachrichten, Crash von Server / Client)
 - Fehlersemantik (maybe, at-least-once, at-most-once, exactly once)
 - Orphans
- RPC
 - Binding
 - Protokolle
 - Effizienz
 - asynchroner RPC (“remote service invocation”)
- Socket-Programmierschnittstelle
 - Client-Server-Beispiel in C
 - Sockets in Java (in den Übungen)
- Java als “Internet-Programmiersprache”
- Adressierungsarten
 - 1:1, direct naming
 - m:n, mailbox
 - n:1, port
 - Kanäle
- Empfangen von Nachrichten
 - non-blocking (--> aktives Warten)
 - Zeitüberwachung

Resümee (6)

- Empfangen von Nachrichten
 - selektives Empfangen
 - implizites Empfangen
- Sprachaspekte beim verteilten Programmieren
 - kommunizierbare Datentypen?
- Gruppenkommunikation (Broadcast / Multicast)
 - Anwendungen
 - idealisierte Sicht
 - Fehlerproblematik
 - Zuverlässigkeitsgrad (“best effort”, k-zuverlässig)
 - “reliable Broadcast” mit ACK, NACK
- Algorithmus für “reliable Broadcast”
- FIFO-Broadcasts
 - zwei nacheinander ausgeführte Broadcasts ein und desselben Senders erreichen alle Empfänger in dieser Reihenfolge
 - nicht stark genug, um akasale Beobachtungen zu verhindern
- Kausale Broadcasts
 - kausale Abhängigkeit zweier Nachrichten
 - “Causal Order”: Nachrichtenempfang “respektiert” kausale Abhängigkeit von Nachrichten (“kausaltruu”)

Resümee (7)

- Atomare Broadcasts
 - logisch gleichzeitiger Empfang der Einzelnachrichten eines Broadcasts
 - Realisierung über zentralen Sequenzer bzw. Token auf einem logischen Ring
- Kausal atomare Broadcasts
 - virtuelle Synchronität
- Multicast
 - Zweck
 - Adressierung von Multicast-Gruppen
- Gruppenüberlappung
 - lokale / globale Gültigkeit von Reihenfolgebedingungen etc.
- Multicast: Membership-Problem
 - atomare Änderung der Gruppenzugehörigkeit
 - Tolerieren von Prozessausfällen
- Push-Prinzip und Publish & Subscribe
- Ereigniskanäle als “Softwarebus”
- Tupelräume
 - Linda-Modell
 - JavaSpaces
- Logische Zeit
 - Raum-Zeitdiagramme, Ereignisse
 - Zeitstempel von Ereignissen
 - Uhrenbedingung. kausale Unabhängigkeit

Resümee (8)

- Logische Uhren von Lamport
 - Definition
 - Realisierung
 - injektive Abbildung, eindeutige Zeitpunkte
- Wechselseitiger Ausschluss (mit logischer Zeit)
 - replizierte Warteschlangen von Lamport (request, reply, ack)
 - Lösung von Ricart / Agrawala 1981
 - safety, liveness, fairness?
- Zufall

Resümee (9)

- Namensverwaltung
 - Zweck von Namen
 - Namen und Adressen
 - Binden
 - Namenskontexte, hierarchische Namensräume
 - Aufgaben einer Namensverwaltung
 - Namensverwaltung in verteilten Systemen
- Nameserver
 - Replikation und Caching
- Internet Domain Name Service (DNS)
 - Namensauflösung im Internet
 - resource records
 - nslookup (bzw. "dig" oder "host")
- Client-Server-Modell (\Leftrightarrow Peer-to-Peer-Strukturen)
 - Prinzip
 - Client/Server-Maschinen
 - Client/Server-Rollen
- Zustandsändernde / -invariante Dienste und Server
 - idempotente und wiederholbare Aufträge
 - stateless / statefull
- Zustandsändernde / -invariante Dienste und Server
 - Beispiel WWW-Server
 - cookies

Resümee (10)

- Konkurrente Server
 - dynamische / statische Handler-Prozesse ("slaves")
 - quasi-konkurrente Server (internes Multiplexen)
- X-Window als "klassisches" Client-Server-System
 - aber: events zur asynchronen Rückmeldung Server --> Client
- Servergruppen / verteilte Server
 - Strukturen kooperierender Server
 - Server-Auswahl bei einem Lastverbund
 - Replikation von Servern ("Überlebensverbund")
- Middleware: Der Weg zum "Netzwerk als Computer"

Resümee (11)

- Sun-RPC

- Identifikation entfernter Prozeduren (host, Programm-, Version-, Prozedur-Nummer)
- Registrieren eines Dienstes auf Serverseite

- Portmapper

- Zuordnung Port / Programmnummer eines Dienstes

- Sun-RPC: rpcgen

- Generieren von Filtern und Prozedurstubs aus Schnittstellenspezifikation

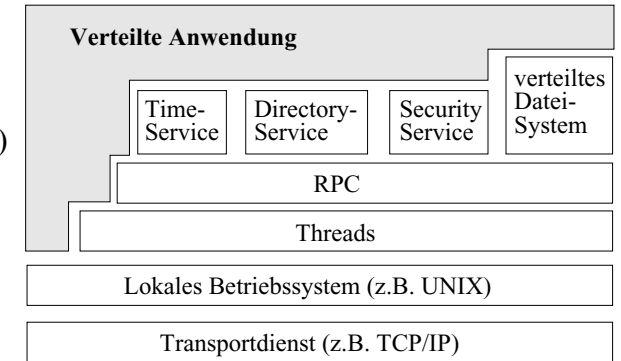
- Schutzaspekte bei Sun-RPC

- “UNIX flavor”: Automatisches Mitsenden von Benutzererkennung etc.
- “Secure RPC”: Authentifizierung und Verschlüsselung (DES bzw. Kerberos)

Resümee (12)

- DCE

- Hauptkomponenten
- Zellenarchitektur
- Threads (Problematik)
- RPC (Unterschied zu Sun-RPC; Fehlersemantik; Bindevorgang)
- Sicherheitsaspekte



- CORBA

- CORBA-Architektur
- Object Services und Common Facilities
- neuere Erweiterungen bei CORBA

Resümee (13)

- Jini

- Motivation: Dienstparadigma, Netzzentrierung,...
- Java-Bezug
- Lookup-Service
- Discovery
- Join
- Proxies und smart Proxies
- RMI
- Codemobilität
- Leases
- verteilte Ereignisse
- Vorteile und Probleme von Jini

Resümee (14)

- Sicherheit in verteilten Systemen: Anforderungen
- Authentifizierung
- Einmalpasswörter mit Einwegfunktionen
- One-time-Pads mit XOR
- Symmetrische und asymmetrische Kryptosysteme
- DES
- Authentizität
- Authentifizierung mit sym. und asym. Schlüsseln
- Techniken zur Verhinderung von Replays
- “Geheime” Schlüsselvergabe
 - Schlüsselaustausch mit Diffie-Hellman-Prinzip
- Authentifizierung mit geheimen Zertifikaten
- Zero-Knowledge-Proofs
 - Beispiel: Isomorphie von Graphen
- Kerberos
 - Protokoll für Ticket-Granting-Ticket- und Service-Ticket-Erwerb
 - Anwendungsbeispiel: Einrichtung sicherer Kanäle
 - Sicherheitsaspekte