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-Anwendungszenario 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
  - Schaltnetzkoppelung (Crossbar, Permutationsnetze)
- Multicomputer (verteilter Speicher)
  - Transputer
  - Bewertungskriterien für Verbindungstopologien
  - Hypercube (rekursives Konstruktionsprinzip)
- Mehrrechner-Verbindungstopologien
  - Torus
  - Cube Connected Cycle
  - Zufallstopologien

Resümee (3)

- Mehrrechner-Verbindungstopologien
  - Zufallstopologien
- 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
- Synchrone Kommunikation
  - Definition
  - Realisierung

Resümee (4)

- Synchrone Kommunikation
  - virtuelle Gleichzeitigkeit; Gummibandtransformation
  - Blockaden und Deadlocks
- Asynchrone Kommunikation
  - Vor- / Nachteile gegenüber synchroner Kommunikation
- Synchrone 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

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

Resümee (6)

- Socket-Programmierschnittstelle
  - Sockets in Java
- 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 (7a)

- Empfangen von Nachrichten
  - selektives Empfangen
  - implizites Empfangen
- Sprachaspekte beim verteilten Programmieren
  - Einheiten der Modularisierung = Verteilung = Parallelität?
  - kommunizierbare Datentypen?

Resümee (7b)

- 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 akausale Beobachtungen zu verhindern
- Kausale Broadcasts
  - kausale Abhängigkeit zweier Nachrichten
  - “Causal Order”: Nachrichtenempfang “respektiert” kausale Abhängigkeit von Nachrichten (“kausaltreu”)
- Atomare Broadcasts
  - logisch gleichzeitiger Empfang der Einzelnachrichten eines Broadcasts
  - Realisierung über zentralen Sequenzer bzw. Token auf einem logischen Ring
- Kausal atomare Broadcasts
  - virtuelle Synchronität

Resümee (8)

- Multicast
  - Zweck
  - Adressierung von Multicast-Gruppen
- Gruppenüberlappung
  - lokale / globale Gültigkeit von Reihefolgebedingungen etc.
- Multicast: Membership-Problem
  - atomare Änderung der Gruppenzugehörigkeit
  - Tolerieren von Prozessausfällen
- Mbone
- Push-Prinzip und Publish & Subscribe
- Ereigniskanäle als “Softwarebus”
- Tupelräume
  - Linda-Modell
  - JavaSpaces

Resümee (9)

- Logische Zeit
  - Raum-Zeitdiagramme, Ereignisse
  - Zeitstempel von Ereignissen
  - Uhrenbedingung. kausale Unabhängigkeit
  - 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?
- Namensverwaltung
  - Zweck von Namen
  - Namen und Adressen
  - Binden
  - Namenskontexte, Hierarchische Namensräume
  - Aufgaben einer Namensverwaltung
  - Namensverwaltung in verteilten Systemen
- Echter Zufall?

Resümee (10)

- Nameserver
  - Replikation und Caching
- Internet Domain Name Service (DNS)
  - Namensauflösung im Internet
  - resource records
  - nslookup
- Client-Server-Modell (<==> 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
- 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

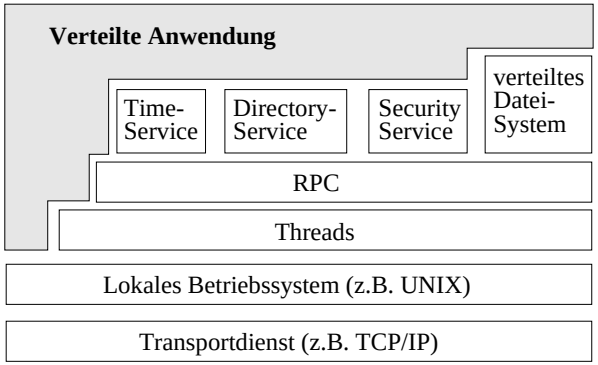
Resümee (11a)

- Servergruppen / verteilte Server
  - Strukturen kooperierender Server
  - Server-Auswahl bei einem Lastverbund
  - Replikation von Servern (“Überlebensverbund”)

- Middleware: Der Weg zum “Netzwerk als Computer”
- 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 Benutzerkennung etc.
  - “Secure RPC”: Authentifizierung und Verschlüsselung (DES bzw. Kerberos)

Resümee (11b)

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



Resümee (12)

- DCE
  - Sicherheitsaspekte
- CORBA
  - CORBA-Architektur
  - Object Services und Common Facilities
  - neuere Erweiterungen bei CORBA
- Jini
  - Motivation: Dienstgparadigma, Netzzentrierung,...
  - Java-Bezug
  - Lookup-Service
  - Discovery
  - Join
  - Proxies und smart Proxies
  - RMI
  - Codemobilität
  - Leases
  - verteilte Ereignisse
  - Vorteile und Probleme von Jini

Resümee (13)

- 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
  - Schatztruhe mit zwei Vorhängeschlössern
  - Schlüsselaustausch mit Diffie-Hellman-Prinzip

Resümee (14)

- Aktive Angreifer
  - Erkennen von Eindringlingen schwierig
- 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