Plan 9 from Bell Labs

Seminar aus Informatik

Pree, Wolfgang, Univ.-Prof. Dipl.-Ing. Dr.techn.

Mutlu Ertas

Kenan Önal

Was ist Plan9

- ein Distributed Operating System
- ein Research System

Wer steckt dahinter?

- Entwickelt wurde Plan9 erstmals 1980 in den AT&T Bell Labs
- Die Gruppe Entwickelte auch **Unix, C** und **C++**.

Team

- *Rob Pike*
- *Dave Presotto*,
- *Sean Dorward*,
- *Bob Flandrena*,
- *Ken Thompson*,
- *Howard Trickey*
- *Phil Winterbottom*.

Woher kommt der Name Plan9?

- Der name plan9 bezieht sich auf den Film *Plan9 from Outer Space* von Ed wood

- In diesem „ schlechtesten US Film aller Zeiten“ geht es u.a. darum, dass Tote mittels Strahlen wiederbelebt werden.

Geschichte

- Paln9 wurde wie bereits 1980 erwähnt
- Mitte der 80er Jahre wurde Plan 9 erfolgreich in den Bell Labs eingesetzt.
- Das erste Major-Release wurde jedoch erst 1992 veröffentlicht
- *First Edition* ist vorerst nur für Universitäten verfügbar
- *Second Edition* im Jahre 1995 für alle veröfentlicht
- Im Jahre 2000 für Jedermann frei über das Internet zugänglich gemacht

Hintergrund

- Bis Mitte der 1980er war der Trend im Computerbereich weg von großen zentralisierten time-shared Computern hin zu Netzwerken kleinerer, persönlicher Maschinen, typischerweise UNIX Workstations. Trotz dem Verlust von Rechenleistung ist dieser Stil bis heute populär.

Schwächen von Unix

- Unix ist für großen Maschinen entwickelt worden.
- Arbeiten mit Grafik, im Netz oder mit mehreren Prozessoren ist nur unzureichend möglich.
- Ein solches Netzwerk mit vielen kleinen privaten Systemen zu administrieren ist äußerst aufwendig, da jeder Computer immer ein Eigenleben entwickelt, sobald er eine Festplatte besitzt.
- Schwierigkeiten bei der nahtlosen Zusammenarbeit von Netzwerken.

Motivation

- Ziel der Entwickler war es nun, ein Unix aus vielen kleinen Systemen zu bauen, anstatt ein System aus vielen kleinen Unixen.

- verschiedene Computer für verschiedene Aufgaben

Einsatzgebiete

- Plan9 wird häufig für große Fileserver eingesetzt, wo viele und wichtige Daten gespeichert und abgerufen werden müssen.
- Ein weiteres Gebiet liegt bei Verteilten Berechnungen und paralleler Programmierung.
- Plan9 wird auch aus Sicherheitsgründen verwendet.

File-Server

- Speichert permanent Dateien und exportiert sie als Dateisystem mittels 9P
- Server ist stand-alone System, Zugang nur über das Netzwerk, es laufen keine Benutzer-Prozesse
- das exportierte Dateisystem ist ein einzelner Baum, der mehrere Datenträger repräsentiert
- 3 Ebenen der Speicherung

RAM	128 MB
magnetic disks	27 GB
WORM	350 GB

WORM: write-once-read-many

Namensbereiche

- Konstruktion des Namensbereich mittels 3 Systemaufrufen:

  mount*:* erzeugt neue Verzeichnisstrukturen durch Hinzufügen von Verzeichnisbäumen anderer Server zum eigenen

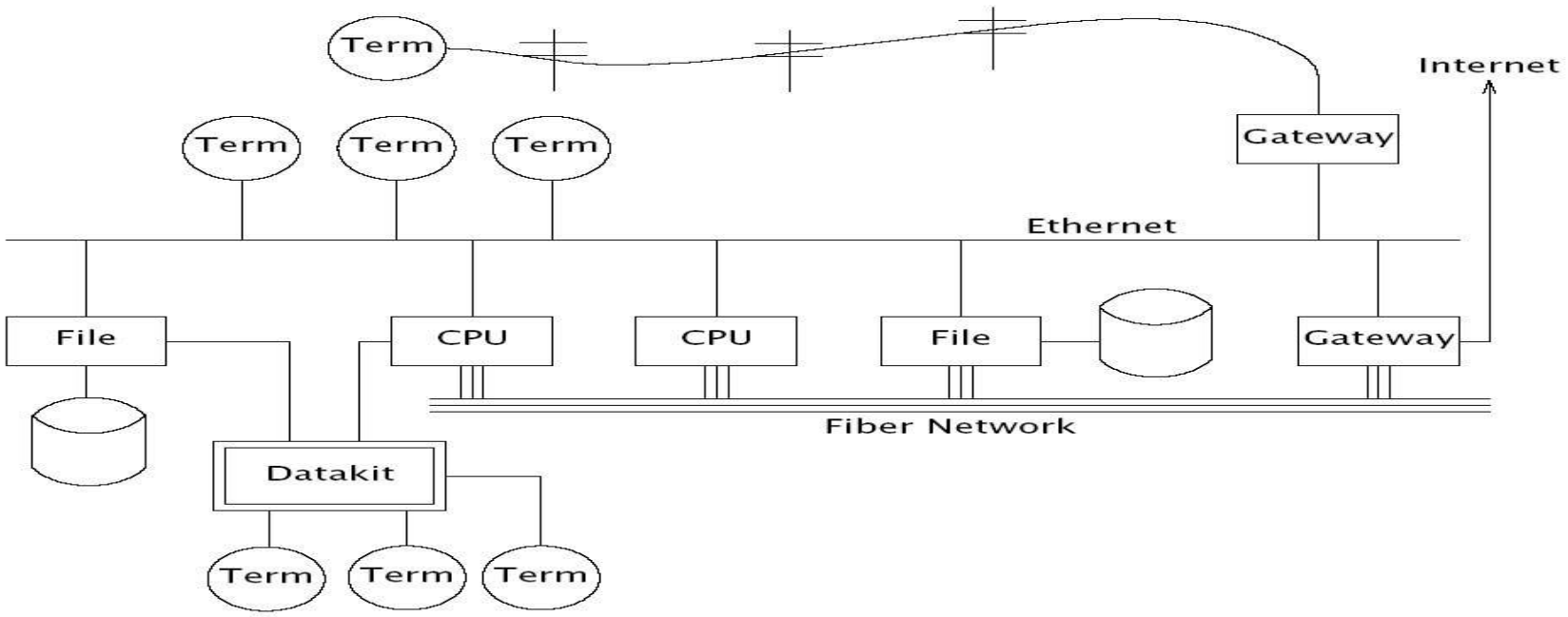
  bind: dupliziert Teil eines bestehen Namensbereiches an eine andere Stelle im Namensbereich

  unmount**:** entfernt Komponenten aus den Namensbereich

- durch *mount/bind* können sich Verzeichnisse überlagern -> union directory

Design - Struktur

- große Plan 9 Installationen bestehen aus mehren vernetzten Rechnern
- jeder Rechner stellt einen bestimmten Dienst bereit (CPU Server, File Server, Terminal)



Design Prinzipien

- Zugriff auf die Ressourcen erfolgt mittel standard Protokoll 9P
- alle Terminals besitzen dieselbe Funktionalität
- Benutzer erstellen sich eigene persönliche Sicht auf das System mit lokalen Namen für globale Ressourcen.
- Dateien/Verzeichnisse dienen als Schnittstelle zu Dienste

9P – Protokoll

- einheitliches Protokoll für alle Services
- Anforderungen an darunterliegende Transportschicht:
    - Nachrichten kommen zuverlässig und in richtiger Reihenfolge an.
    - Begrenzungen zwischen Nachrichten bleiben erhalten

    (trifft eine Bedingung nicht zu, gibt es spez. Marshalling-Mechanismen)

- Dateien als Sequenz von Bytes (anstatt von Blöcken)
- 9P ist zustandsorientiert (RPCs erzeugen Zeiger auf Objekte in Remote File-Servern, sog. fids)
- alle Operationen auf Dateien verwenden fids um Objekte zu identifizieren

- alle Operationen auf lokale Dateien benötigen einen channel (jeder fid wird im Kern durch einen channel repräsentiert)

9P Protokoll - Methoden

- ***session:*** startet Authentifzierung zwischen Server und Client
- ***attach:*** verbindet den vom Clienten bereitgestellten fid mit der Wurzel des Server-Verzeichnisses (channel wird erzeugt)
- ***walk:*** bewegt channel durch eine Ebene im Datei-Verzeichnis
- ***clone:*** erzeugt eine Kopie eine bestehenden channels
- ***open:*** lockt einen channel auf eine bestimmte Datei, prüft Zugriffsrechte und bereitet den channel für I/O vor
- ***read/write:*** I/O an beliebigen Offsets in der Datei
- ***clunk:*** verwirft channel ohne die Datei zuverändern
- ***stat/wstat:*** lesen und schreiben der Datei-Attribute
- ***remove*:** verwirft channel und löscht die Datei

Netzwerk – Benutzerschnittstelle

- ***clone*:** öffnen der Datei findet eine freie Verbindung und öffnet dessen ctl Datei
- ***ctl:*** lesen liefert ASCII Verbindungsnummer, schreiben konfiguriert Verbindung
- ***type:*** lesen liefert Pakettyp, schreiben legt Pakettyp fest (-1 = alle Pakete)
- ***data:*** lesen liefert nächstes empfangenes Paket des gewählten Typs, schreiben reiht das Paket nach hinzufügen des Headers (Quelladresse, Packettyp) in die Warteschlange zum Senden
- ***stats****:* liefert als ASCII Text die Schnittstellen-Adresse, Zähler über Ein- und Ausgänge von Paketen, Fehlerstatistiken, Zustand der Schnittstelle
- ***listen:*** Öffnen blockiert Prozess bis Call eingeht; wenn Call eingeht, wird Öffnen fortgesetzt und liefert File Descriptor zur ctl-Datei der neuen Verbindung

Grafik mit Rio

- grafische Oberfläche direkt integriert
- einfaches Programmiermodell
- Implementation von Rio ist trickreich
- verabschiedet das klassische zeilenorientierte Terminalmodell

Ausblick

- Ändert sich noch immer täglich
- Von anderen Betriebssystemen adoptiert werden.

Danke für Ihre Aufmerksamkeit!