

Weltweit verteilte Datenbanksysteme

Wer bin ich?

- Persönliches:
 - Name: Daniel Steudel
 - Studium: Informatik im 8. Semester
 - Vertiefungsfach (geplant):
Informationssysteme
- Für diesen Vortrag gilt:
 - Fragen was man will!
 - Fragen wann man will!
 - Fragen wie man will!

Routenplan

- Weltweit oder lokal
- Weltweit eindeutig
- Weltweit wiederfinden
- Weltweit sicher
- Weltweit gleich

Weltweit oder lokal

Einfache Welt

- Singuläre einfache Datenbanken (DBMS)
- Einzelner Server mit „einer“ Platte
- Anwender greifen durch überschaubares Netzwerk zu

Datenbankserver

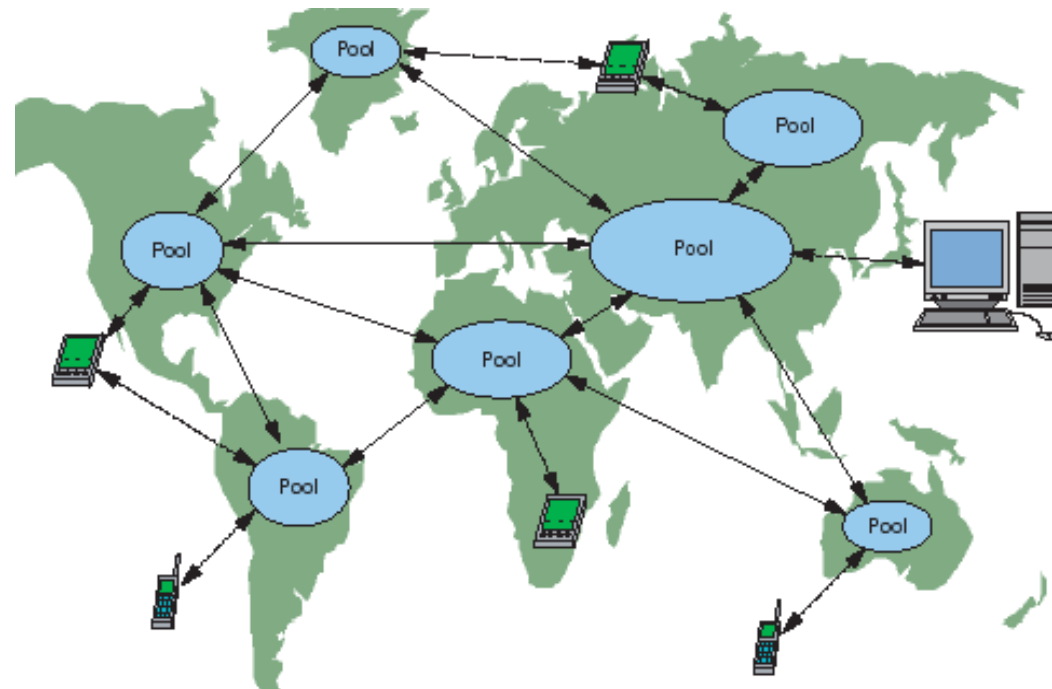


Was aber, wenn ...?

- Internationales Unternehmen
 - Sitz in Möhringen (Stuttgart)
 - Aussenstelle in Auburn Hills (Detroit – USA)
 - Buchhaltungssystem für einheitliche Steuererklärung
- ⇒ entweder zentralverwaltet oder verteilt

Wichtige Eigenschaften

- Lokalität
 - Ausfall- und Datensicherheit
- ⇒ Replikate der Objekte



Richtig austeilen

- Verteilung über Backup-Recovery
=> Aktualität der Daten
- Verteilung über Clustering
=> Performance und Ausfallsicherheit
- Verteilung über P2P
=> Authentisierung und Autorisierung

Weltweit eindeutig

Generelle Überlegungen

- Eindeutigkeit über Globally Unique Identifier (GUID)
- (Pseudo-)Zufällig und mit fester Länge
- Rechenbeispiel:
 - 10^{10} Benutzer mit 1.000 Informationen
=> 10^{14} Objekte
 - Wahrscheinlichkeit bei 160-Bit Schlüssel 2 von 10^{14} (unterschiedlichen) Objekten gleiche GUID ist 10^{-18}

Implementierung OceanStore

- Schlüsselgenerierung über SHA-1
- Verschlüsselung mit Benutzerschlüssel
- Für GUID werden Objektdaten benutzt

Das bedeutet auch ...

- Objekte sind selbstverifizierend
- Objekte sind schreibgeschützt
- Archivierung von Objekten

Weltweit wiederfinden

Generelle Überlegungen

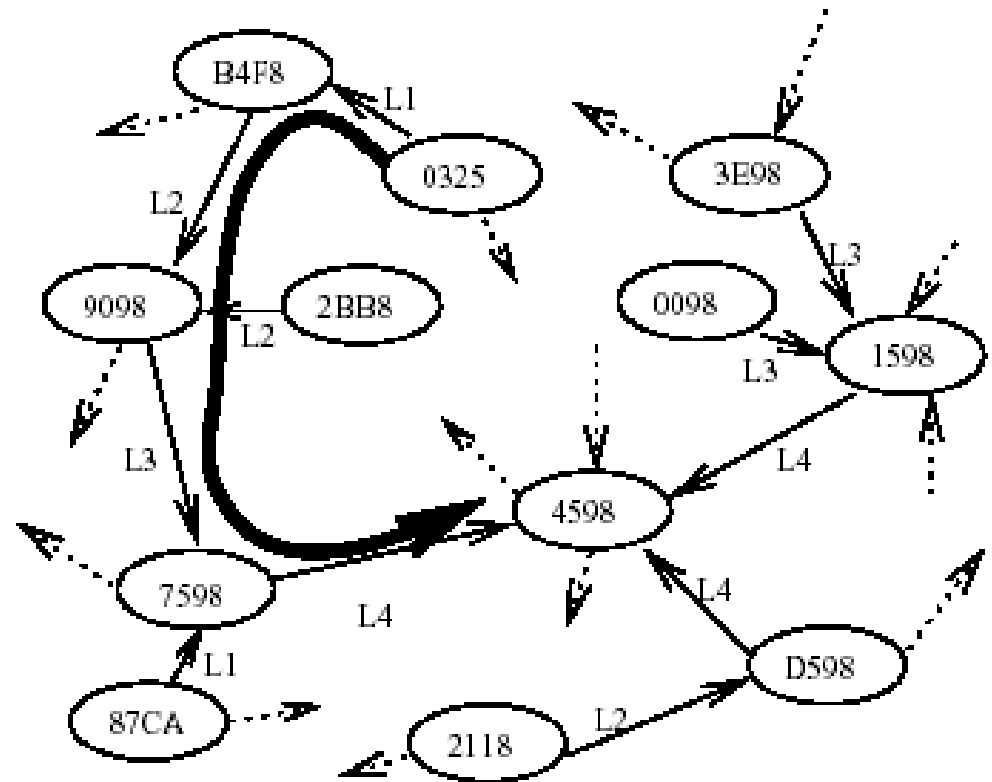
- Deterministischer Aufbewahrungsort
- Routinglokalität
- Minimalität und Ausgeglichenheit
- Dynamische Infrastruktur

Plaxton, Rajaraman und Richa (PRR)

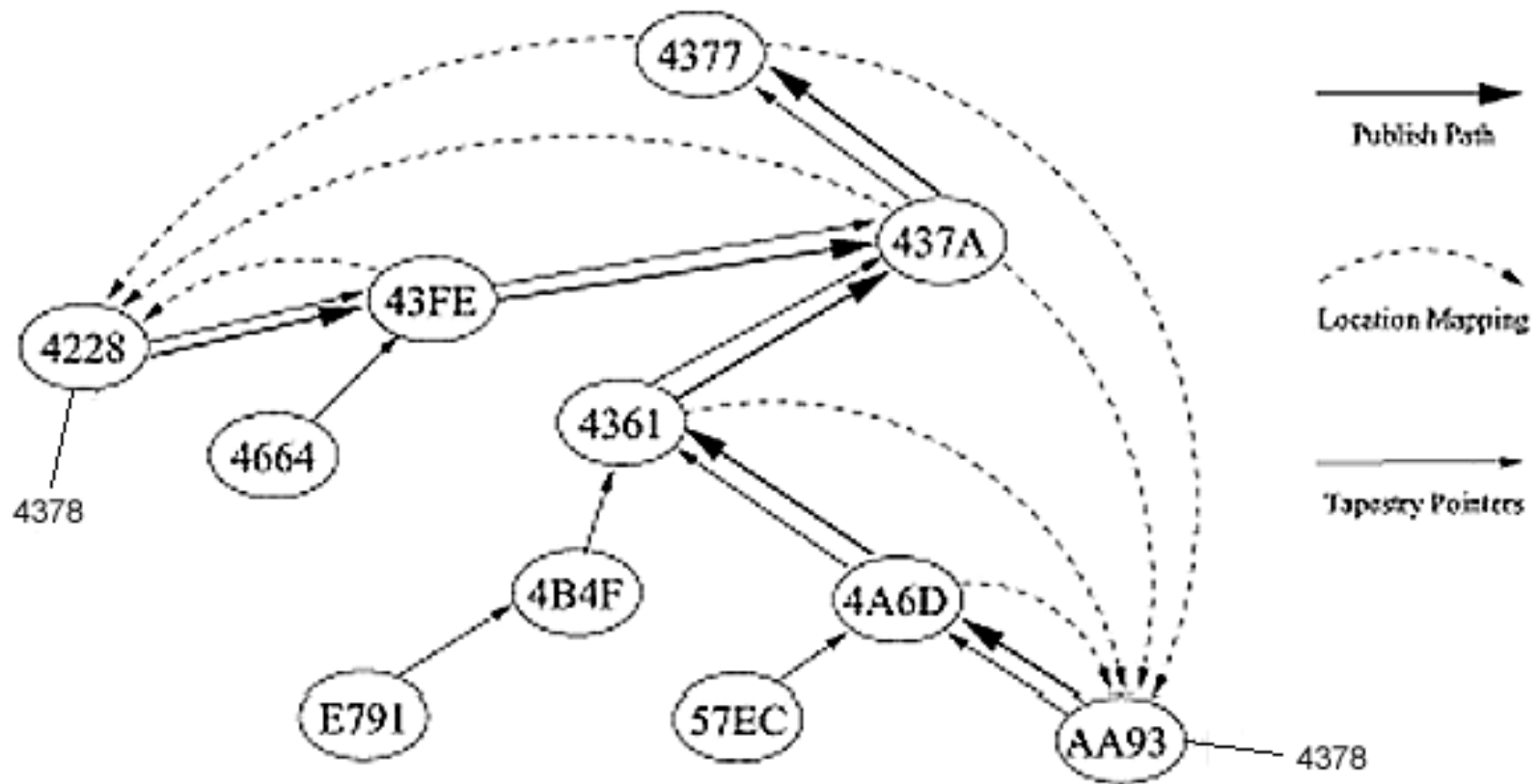
- Grundlegender Algorithmus
- Erfüllt nicht alle Anforderungen
- Weiterentwickelt in Tapestry
(OceanStore Routing Dienst)

Die Idee

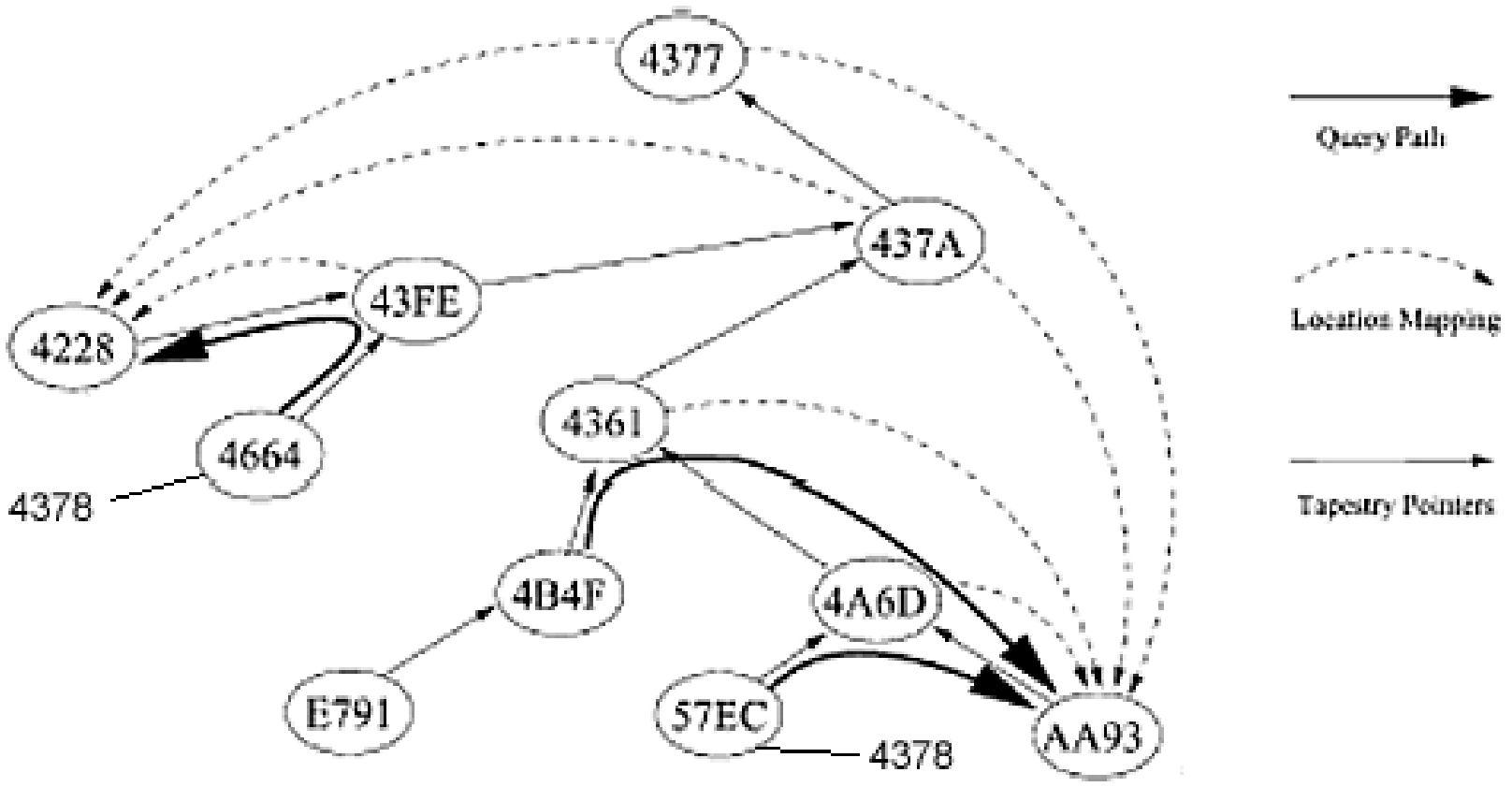
- Eindeutige ID's
=> GUID
 - Suche nach 4598
 - Begin bei 0325
 - Auswerten der Stufen
- ⇒ Deterministisch



Replikate einfügen



Replikat suchen



Weltweit sicher

Generelle Überlegungen

- Welchem Benutzer gehören welche Daten
- Wer darf auf die Daten zugreifen
- Wie werden die Rechte verwaltet

Datenbesitz

- Daten werden auf dem Client verschlüsselt
 - GUID sind mit Benutzerschlüssel verschlüsselt
 - Aus verschlüsselter GUID lässt sich der Besitzer erkennen
- ⇒ Prüfung des Datenbesitzes ist systemimmanent

Datenzugriff

- Nicht vollständig öffentliche Daten werden Verschlüsselt
- Access Control Lists (ACL) sind eigenständige Objekte
- ACL's werden nicht verschlüsselt

Rechteverwaltung

- Zugelassene Benutzer bekommen öffentlichen Schlüssel des Objekts
 - Löschung des Schlüssels
 - Neuverschlüsselung
- ⇒ Alte Daten können immer noch sichtbar sein.

Weltweit gleich

Generelle Überlegungen

- Änderung des Objektinhalts
=> Änderung der GUID
- Auf verschlüsselten Daten
- Nebenläufigkeit
- ACID-Bedingung

Updates und GUID's

- Objekte besitzen einen „inneren Ring“
- Innerer Ring für serialisierbarkeit zuständig
- Innerer Ring arbeitet mit Byzantinisches Agreement Protokoll (m-von-n Kodierung)
- Über Selbstbeobachtung (introspection) werden u.a. Replikate verwaltet

Nebenläufigkeit und ACID

- ACID ist erfüllbar aber normalerweise nicht gefordert
- Innerer Ring ist für einhalten der Beschränkungen zuständig
- Aktionen werden Atomar ausgeführt

Fazit

- OVID benötigt Datenbank die schnelle Updates bietet
- Datensicherheit ist nicht zwingend notwendig
- Verkehrsdaten relativ lokal
- Wie sehen „select“-Anfragen in OceanStore aus?

Ring frei zur Diskussion !