Google Chubby

Ein verteilter Lockservice

Fachseminar Datenbank-Optimierung

Hochschule RheinMain
Wintersemester 2015/16

Patrick Canterino
20.01.2016

Inhalt

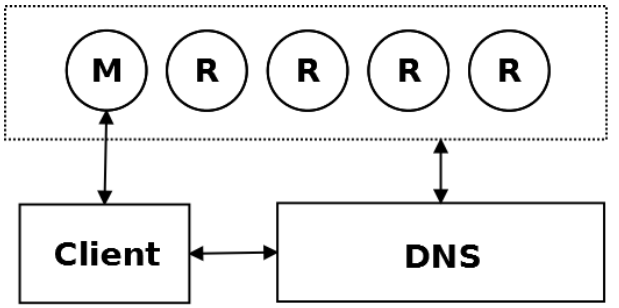
- Was ist Google Chubby?
- Chubby-Zelle
- Dateisystem
- Client-Bibliothek
- Caching und Sessions
- Master-Failover
- Verwendung bei Google
- Erfahrungen
- Quellen

Was ist Google Chubby?

- Interner Dienst von Google
- Verteilter Lockservice und Dateisystem
- Grobkörnige, empfehlende (*advisory*) Sperren
- Primärziele bei Entwicklung:
  - Verteilte Systeme synchronisieren
  - Zentraler Dienst statt Paxos-Bibliothek
  - Hohe Verfügbarkeit

Chubby-Zelle

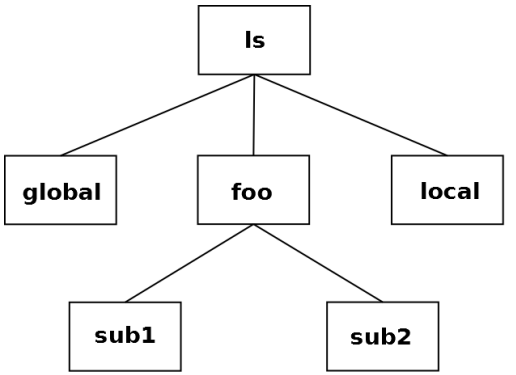
- Verbund von Servern
- *Master* wird mit Hilfe von Paxos gewählt
- Master hat eine *Lease*-Zeit
- Zelle wird über DNS gefunden
- Master kommuniziert mit den Clients und bearbeitet Datenbank
- Replikation auf restliche Server (*Replicas*) durch Paxos



Quelle: Eigene Darstellung nach *Burrows*

Dateisystem 1

- Schnittstelle für Client-Entwicklung
- Ähnlich Unix-Dateisystem, jedoch einfachere Struktur
- Besteht aus Baum von *Knoten* (Dateien und Verzeichnisse)
- Dateipfad: */ls/foo/sub1*
- Vorteile:
  - Leicht verständlich für Entwicklung
  - Leicht mit Google File System zu verbinden



Quelle: Eigene Darstellung

Dateisystem 2

- Knoten können permanent oder flüchtig sein
- Jeder Knoten verfügt über Meta-Daten (ACLs, Zähler und Prüfsummen)
- Jeder Knoten kann Lese- oder Schreibsperre erhalten
- *Sequencer* zur Validierung von Sperren
- Event-Mechanismus
- Partitionierung möglich, jedoch nicht genutzt (2006)
- Spiegelung möglich

Client-Bibliothek

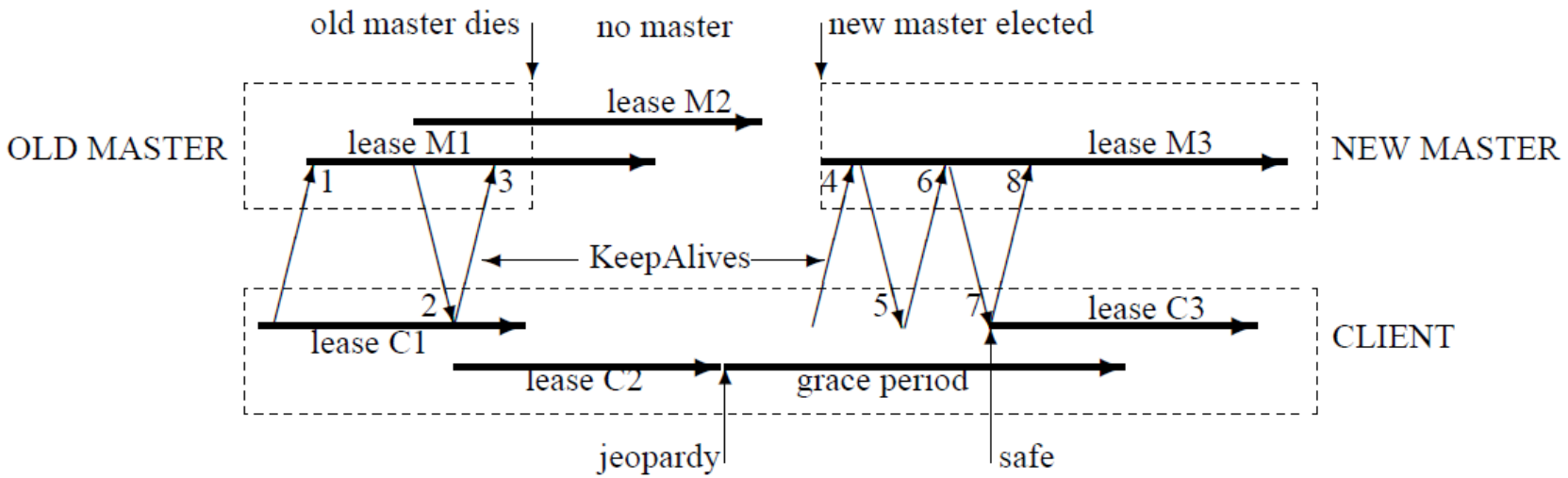
- C++-Bibliothek und Java-Proxy
- Typische Datei-Operationen
- Beispiel-Ablauf: Bestimmen eines Masters
  1. Server im Cluster verbinden sich mit Chubby-Zelle, öffnen eine Chubby-Datei und setzen Schreibsperre
  2. „Gewinner“ wird Master und hinterlegt Referenz in Datei
  3. Master wird über Datei gefunden
  4. Master beantragt Sequencer

Caching und Sessions

- Clients verfügen über lokalen Cache
- Explizite Cache-Invalidierung durch Zelle
- Session wird durch Client aufgebaut und besteht für eine *Lease*-Zeit
- Client verlängert Lease-Zeit mit *KeepAlive*-Paketen
- Keine Antwort vom Server auf KeepAlive:
  - Cache leeren (*Jeopardy*-Modus)
  - Letzte Versuche, KeepAlives zu senden (*Grace Period*)
- Nutzung von Proxys möglich

Master-Failover

- Neuer Master wird gewählt
- Clients finden neuen Master
- Neuer Master stellt Sessions und Handles anhand der Clients wieder her



Quelle: *Burrows*

Verwendung bei Google (2006)

- Master-Ermittlung und Wurzel für Google File System (GFS) und Bigtable
- Serverliste für Bigtable
- Konfigurationsablage
- Monitoring
- Hauptnutzen jedoch: DNS-Ersatz (wegen effizienterem Caching)
- Nutzung von Konvertierungsservern (z. B. für DNS)

Erfahrungen 1 (2006)

- Viele kleine Dateien (90 % unter 1 KB)
- Hauptsächliche Nutzung als Namensdienst (60 %)
- KeepAlive-Pakete größter Teil des Datenverkehrs (93 %)
- Überlastung bei vielen Sessions (über 90.000)
- Verbesserung der Performance durch höhere Lease-Zeit, sowie Proxys und Partitionierung

Erfahrungen 2 (2006)

- Ausfälle haben nur kleine Auswirkungen
- Nur seltene Fälle von Datenverlust
- Entwickler nutzen Chubby manchmal falsch oder prüfen Verfügbarkeit nicht

Quellen

- *Burrows* (2006): http://research.google.com/archive/chubby-osdi06.pdf
- *Du* (2012): https://www.cs.columbia.edu/~du/ds/assets/lectures/lecture19.pdf
- *Jacotin* (2014): http://de.slideshare.net/romain_jacotin/the-google-chubby-lock-service-for-looselycoupled-distributed-systems

Danke fürs Zuhören!