Datenbankadministration

6. Hochverfügbarkeit

AG DBIS University of Kaiserslautern, Germany

Karsten Schmidt kschmidt@informatik.uni-kl.de
(Vorlage TU-Dresden)

Wintersemester 2008/2009

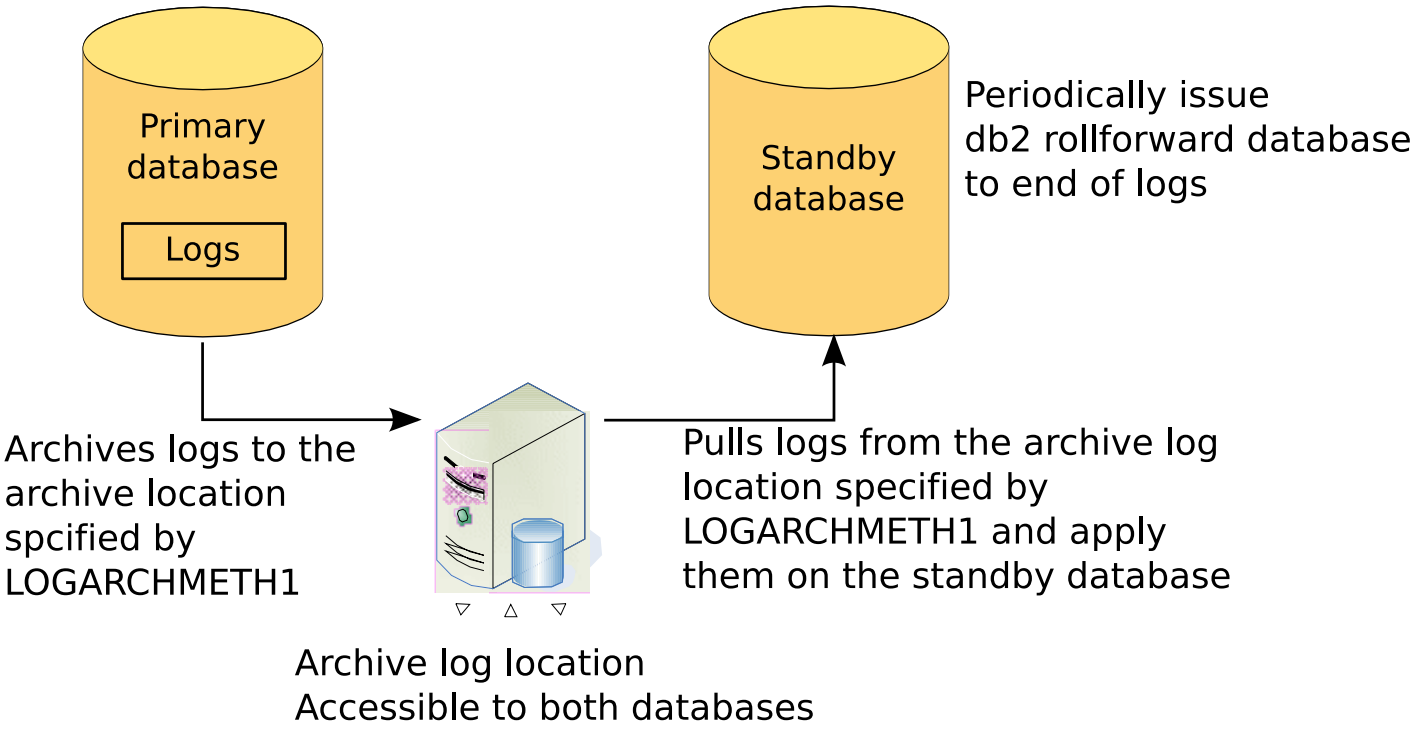
Hochverfügbarkeit

- **Hochverfügbarkeitssystem**
  - Eigenschaften
    - permanent verfügbar
    - Funktionsübernahme im Fehlerfall durch Bereitschaftssystem
    - Schutz vor Datenverlust
  - Architektur
    - Primärdatenbank
      - aktuell laufende Datenbank
    - Bereitschaftsdatenbank (*standby database*)
      - Spiegeldatenbank der Primärdatenbank
      - Zugriff auf Protokolldateien der Primärdatenbank
      - an anderem Standort möglich (TCP/IP-Kommunikation)
  - Archivprotokollierung erforderlich

Protokollübertragung

Protokollübertragung

- **Protokollübertragung** (*log shipping*)
  - Bereitstellung von Transaktionsprotokollen der Primärdatenbank an Bereitschaftsdatenbank
    - Primärdatenbank: Speicherung der Archivprotokolle auf Netzwerk- oder gemeinsamen Laufwerk (`LOGARCHMETH1`)
    - Bereitschaftsdatenbank: `LOGARCHMETH1` wie in Primärdatenbank
    - alternativ: gemeinsame Nutzung eines `USEREXIT`-Programms



Onlineteilung einer Spiegeldatenbank

Onlineteilung einer Spiegeldatenbank

- **Onlineteilung einer Spiegeldatenbank** (*online split mirroring*)
  - Spiegelung = Plattenkopie des Datenbankverzeichnisses
  - Teilung = Trennen der Kopien in primäre und sekundäre DB
  - Vorteil
    - kein Backup-Overhead
    - schnelle Wiederherstellung
  - Voraussetzung: keine Schreiboperationen während Spiegelung
- **Ausgesetzte E/A** (*suspended I/O*)
  - Verhindern von Schreiboperationen durch Versetzen der Tabellenbereiche in `SUSPEND_WRITE`-Zustand
  - normale Funktionalität der Datenbank
  - Ablauf
    - `CONNECT TO <dbname>`
    - `SET WRITE SUSPEND FOR DATABASE`
    - <Kopie des Datenbankverzeichnisses anlegen>
    - `SET WRITE RESUME FOR DATABASE`

Onlineteilung einer Spiegeldatenbank

- **Wiederherstellung**
  - Erzeugen einer Datenbank aus Plattenkopie (`DB2INIDB`)
  - Möglichkeiten
    - Klon der Datenbank (`SNAPSHOT`)
    - Bereitschaftsdatenbank (`STANDBY`)
    - Wiederherstellung der Originaldatenbank (`MIRROR`)
- **Erzeugen einer Bereitschaftsdatenbank**
  - `SET WRITE SUSPEND FOR DATABASE`
  - gleichzeitig Kopie anlegen von
    - vollständigem Inhalt des Datenbankverzeichnisses
    - allen Containern der Tabellenbereiche
    - lokalem Datenbankverzeichnis
    - aktiven Protokollen, falls nicht im Datenbankverzeichnis
      $\longrightarrow$ `SYSIBMADM.DBPATHS`
  - `SET WRITE RESUME FOR DATABASE`

Onlineteilung einer Spiegeldatenbank

- **Erzeugen einer Bereitschaftsdatenbank (Fortsetzung)**
  - auf Zielsystem:
    - Instanz mit gleichem Namen anlegen
    - Kopie in exakt gleichem Pfad wie auf Quellsystem speichern
    - Datenbank katalogisieren
  - `DB2INIDB <dbname> AS STANDBY`
  - `ROLLFORWARD DATABASE <dbname> TO END OF LOGS`
  - Funktionsübernahme durch Bereitschaftsdatenbank (nur im Fehlerfall)
    - Bereitstellen der aktiven Protokolle der Primärdatenbank an Bereitschaftsdatenbank (auch manuell möglich)
    - `ROLLFORWARD DATABASE <dbname>`
      `TO END OF LOGS AND COMPLETE`

High Availability Disaster Recovery

High Availability Disaster Recovery

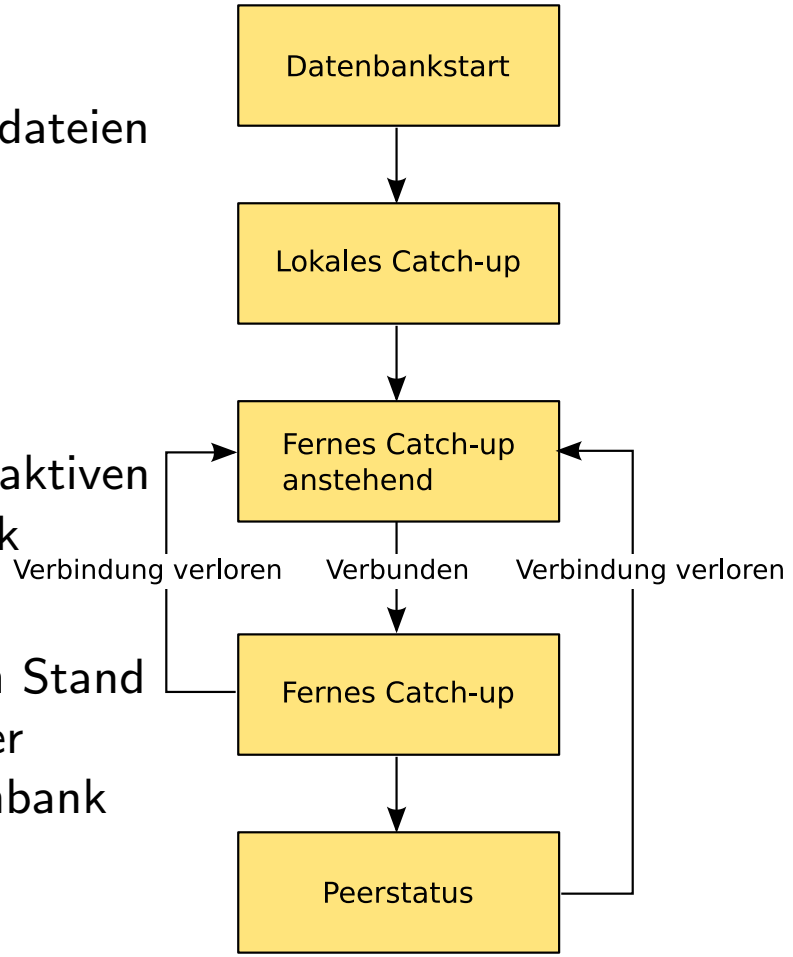
- **High Availability Disaster Recovery (HADR)**
  - Datenbankreplikationsfunktion für Hochverfügbarkeit
  - Erstellung einer Bereitschaftsdatenbank durch Backup & Restore oder Onlineteilung einer Spiegeldatenbank
  - Austausch von Überwachungsnachrichten
  - Verwendung von Protokollübertragung
  - HADR-spezifische Datenbankkonfigurationsparameter (`HADR_*`)
- **Voraussetzungen für Primär- und Bereitschaftsdatenbank**
  - Server mit gleicher Architektur und vom gleichen Hersteller
  - gleiche Hard- und Software (Betriebssystem, DB2-Version)
  - gleiche Datenbank- und Instanzkonfigurationsparameter
  - identische Tabellenbereiche und Container
  - gleiche Namen der Datenbanken
  - Kommunikation über TCP/IP
  - keine partitionierten Datenbanken

High Availability Disaster Recovery

- **Replizierte Daten und Operationen**
  - DDL- & DML-Operationen
  - Operationen an Tabellenbereichen und Pufferpools
  - Reorganisationen
  - Metadaten für Stored Procedures und UDFs (aber nicht deren Daten)
- **Nicht replizierte Daten und Operationen**
  - Datenbank- und Instanzkonfigurationsparameter und deren Änderungen
  - LOBs $>$ 1GB
  - STMM (Self Tuning Memory Manager) Ausführung

HADR-Zustände der Bereitschaftsdatenbank

- **HADR-Zustände der Bereitschaftsdatenbank**
  - lokales Catch-up
    - beim ersten Start
    - Anwenden von lokalen Protokolldateien
  - fernes Catch-up anstehend
    - Warten auf Verbindung zu Primärdatenbank
  - fernes Catch-up
    - Anwenden von archivierten und aktiven Protokollen der Primärdatenbank
  - Peerstatus
    - beide Datenbanken auf gleichem Stand
    - Übertragung und Anwendung der Protokolle an Bereitschaftsdatenbank



HADR-Synchronisationsmodi

- **HADR-Synchronisationsmodi**
  - Datenbankkonfigurationsparameter `HADR_SYNCMODE`
  - Schreiben der Protokolle erfolgreich, wenn:
    1. Protokoll in Protokolldatei der Primärdatenbank geschrieben **UND**
  - synchron (`SYNC`)
    1. Primärdatenbank hat Bestätigung der Bereitschaftsdatenbank über erfolgreiches Schreiben des Protokolls in Protokolldatei der Bereitschaftsdatenbank
  - fast synchron (`NEARSYNC`)
    1. Primärdatenbank hat Bestätigung der Bereitschaftsdatenbank über erfolgreiches Schreiben des Protokolls in Hauptspeicher der Bereitschaftsdatenbank
  - asynchron (`ASYNC`)
    1. Protokolle an Bereitschaftsdatenbank übertragen (ohne Bestätigung)

Erstellung einer HADR-Umgebung

- **Erstellung einer HADR-Umgebung**
  1. Bereitschaftsdatenbank erstellen durch
     - Wiederherstellung eines Backup-Images der Primärdatenbank oder
     - Initialisierung einer geteilten Spiegeldatenbank
  2. HADR-Konfigurationsparameter beider Datenbanken setzen
     - lokaler Hostname: `HADR_LOCAL_HOST`
     - lokaler HADR-Service (Port): `HADR_LOCAL_SVC`
     - entfernter Hostname: `HADR_REMOTE_HOST`
     - entfernter HADR-Service (Port): `HADR_REMOTE_SVC`
     - weitere: `HADR_REMOTE_INST`, `HADR_TIMEOUT`, `HADR_SYNCMODE`
  3. Bereitschaftsdatenbank starten
     - `START HADR ON DB <dbname> AS STANDBY`
  4. Primärdatenbank starten
     - `START HADR ON DB <dbname> AS PRIMARY`

TAKEOVER-Kommando

- **Umschalten der Datenbankrollen** (*takeover*)
  - nur an Bereitschaftsdatenbank möglich
  - Anforderungen
    - Datenbank im Peerstatus
    - Verbindung zu Primärdatenbank vorhanden
  - Trennung aller Verbindungen zu Primärdatenbank
  - `TAKEOVER HADR ON DB <dbname>`
- **HADR-Funktionsübernahme** (*failover*)
  - bei Unerreichbarkeit der Primärdatenbank → kein Rollentausch!
  - nur an Bereitschaftsdatenbank möglich
  - Anforderungen
    - Datenbankstatus: Peerstatus oder fernes Catch-up anstehend
    - Primärdatenbank muss vollständig deaktiviert sein
  - `TAKEOVER HADR ON DB <dbname> BY FORCE`

Stoppen von HADR

- **Stoppen von HADR**
  - Überführung einer HADR-Datenbank in Standarddatenbank (`HADR_DB_ROLE`: `STANDARD`)
  - Beibehaltung der HADR-Konfigurationsparameter
  - Unterbrechung/Deaktivierung: `DEACTIVATE DATABASE <dbname>`
  - Stoppen einer aktiven/inaktiven Primärdatenbank
    - aktiv: Protokollübertragung an Bereitschaftsdatenbank gestoppt, wird Standarddatenbank, bleibt online
    - inaktiv: wird Standarddatenbank, bleibt offline
  - Stoppen einer aktiven/inaktiven Bereitschaftsdatenbank
    - aktiv: Fehlermeldung
    - inaktiv: Rollforward Pending-Zustand, wird Standarddatenbank, bleibt offline
  - `STOP HADR ON DATABASE <dbname>`

Automatische Clientweiterleitung

- **Automatische Clientweiterleitung** (*automatic client reroute feature*)
  - Angabe eines alternativen Servers für den Fall der Unerreichbarkeit des Datenbankservers
  - Speicherung am Client → einmalige erfolgreiche Verbindung zum Datenbankserver erforderlich
  - Ablauf der Weiterleitung bei HADR-Funktionsübernahme:
    - aktive Anwendungen: Trennung und Neuverbindung zu neuer Primärdatenbank
    - neue Anwendungen: automatische Weiterleitung an neue Primärdatenbank
  - `UPDATE ALTERNATE SERVER FOR DATABASE <dbname> USING HOSTNAME <hostname> PORT <port>`
  - Achtung: `port` ist `SVCNAME` (nicht HADR-Port!)
    Achtung 2: `PORT` is korrekt ;o)

Zusammenfassung

- **Protokollübertragung**
  - Bereitstellung von Protokollen für Bereitschaftsdatenbank
- **Onlineteilung einer Spiegeldatenbank**
  - schnelles Sichern und Wiederherstellen von Datenbanken
  - Verhinderung von Schreiboperationen durch ausgesetzte E/A
  - Erstellung eines Klons oder einer Bereitschaftsdatenbank, Wiederherstellung der Originaldatenbank
- **HADR**
  - Hochverfügbarkeitsmechanismus in DB2
  - Funktionsübernahme durch Bereitschaftssystem
  - automatische Clientweiterleitung