

SCALING AND CONCURRENT DATA PROCESSING

Inhalt

- ⊙ Motivation
- ⊙ Einsatzgebiete
- ⊙ Vorarbeiten
- ⊙ Frameworks
- ⊙ Masterarbeit
 - Zielsetzung
 - Chancen & Risiken
 - Vorgehensweise
- ⊙ Ausblick

Motivation

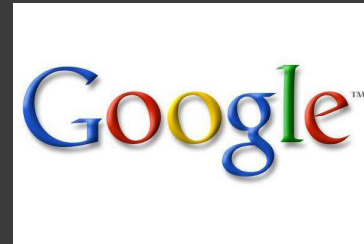
- ◎ Nebenläufigkeit
 - Moore'sches Gesetz
 - SIMD & MIMD
 - Persistenz: RDBMS ungeeignet
- ◎ Trend
 - Große Datenmengen (> TB)
 - GPGPUs (FASTR A II)
 - Cloud Computing

Einsatzgebiete

- Facebook



- Google



- FlightCaster



-

Vorarbeiten

⊙ AW1

- Multicore CPUs
- Traditionelles Locking
- Clojure

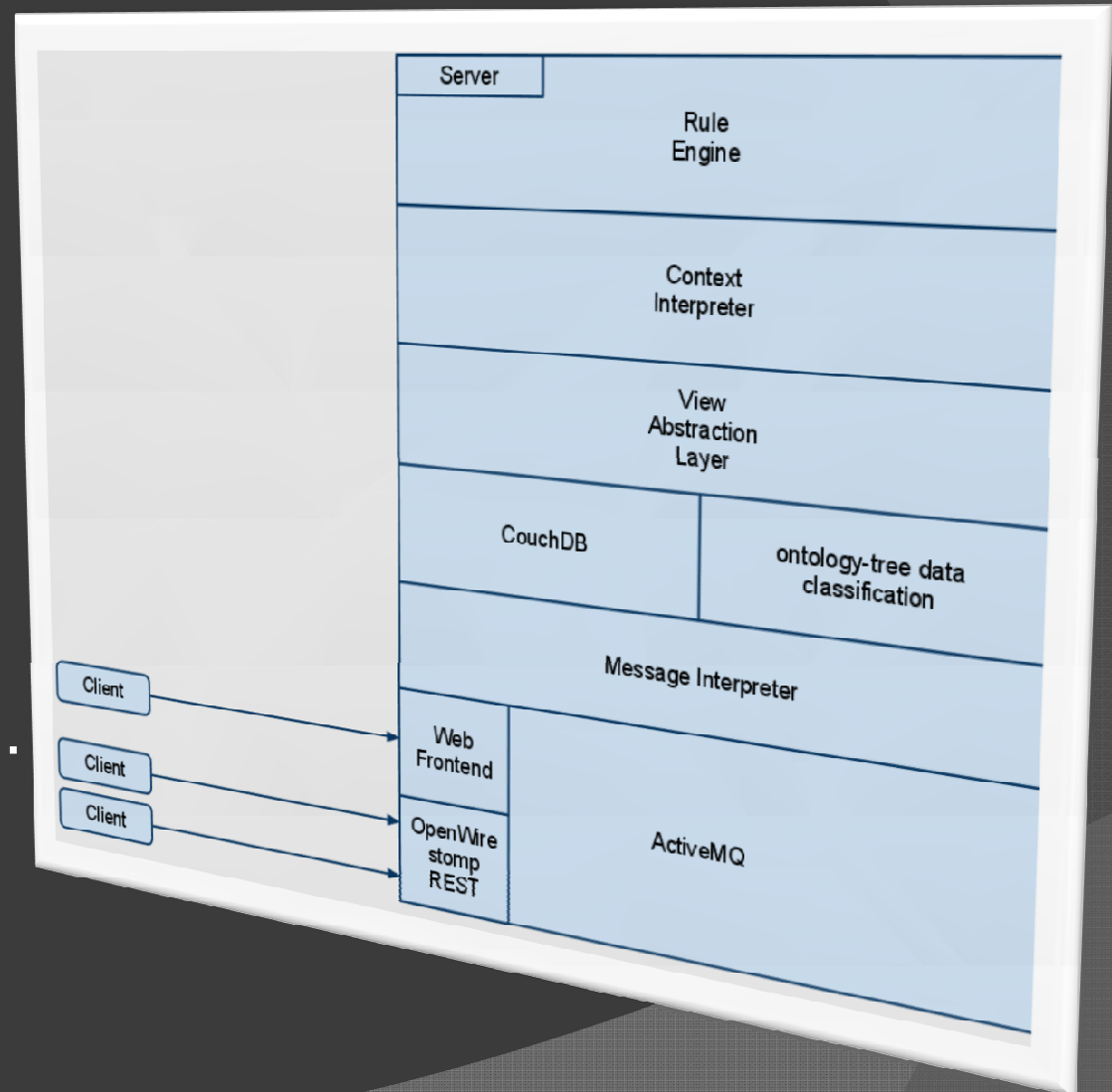
⊙ AW2

- Microsoft STM .Net
- Scala Actors
- Googles-MapReduce

Vorarbeiten

⦿ Projekt 1

- Living Place
- Middleware-softwarestack
- Persistenz
- Interpretation steht noch aus...



Frameworks

- ⊙ Node-Scale

- STM
- Aktoren
- ...



- ⊙ Cluster-Scale

- OpenMP
- Distributed JVM
- ...



- ⊙ Web-Scale

- Google-MapReduce
- Apache Hadoop
 - Pig Latin



Masterarbeit

- ⦿ Bezug zu Vorarbeiten
- ⦿ Living Place Hamburg
- ⦿ Concurrency
 - Trends
 - Frameworks
- ⦿ Zusammenarbeit mit anderen Projekten



Zielsetzung

- ⊙ Nebenläufigkeit im Living Place
- ⊙ Aspekte der Ausfallsicherheit
- ⊙ Skalierbarkeit für Jahre
- ⊙ Überwachung der Teilnehmer

Chancen & Risiken

⊙ Chancen

- Generelle Erkenntnisse
- Einsatz als Teil der Living Place Infrastruktur

⊙ Risiken

- Rahmen zu groß
- Testumgebung eventuell nur „leihbar“
- Zu viele Details

Vorgehensweise

- ⊙ Living Place Entwicklung weiter begleiten
- ⊙ Thema finden
- ⊙ Konkreteren Rahmen abstecken
- ⊙ Wissenschaftlichen Kontext konkretisieren

Ausblick



Fragen

