

Begleitdokumentation

Datenbankprojekt Wahlinformationssystem

WS 2012/2013

Marcus Vetter, Sebastian Wöhrle

28. Januar 2013

Dieses Dokument fasst alle nicht in den anderen abgegebenen Dokumenten zu findenden Informationen zu unserer Implementierung des Wahlinformationssystems(WIS) zusammen.

1 Architektur

Das WIS wurde als Serversystem mit browserbasierter Weboberfläche entwickelt um einem möglichst breiten Nutzerkreis den einfachen Zugriff zu ermöglichen.

Auf Serverseite kommt das Webframework Play¹ mit einer klassischen Model-View-Controller-Architektur zum Einsatz. Play wurde in Java und Scala entwickelt. Die Controller wurden in Java implementiert, für die Views kommt Scala mit einer in Play integrierten Template-Engine zum Einsatz. Dies ermöglicht, kompakte, einfach verständliche und trotzdem mächtige Funktionen zu entwickeln.

Als Backend kommt eine PostgreSQL-Datenbank (Version 9.2) zum Einsatz. Dabei wurde sämtliche Logik, die zur Berechnung von Ergebnissen und Analysen benötigt wird, in SQL geschrieben, um somit den größtmöglichen Effizienzgewinn zu erzielen.

2 Datenpflege

Bei den in der Datenbank gespeicherten Daten muss nach statischen (Bundesländer, Wahlkreise, Parteien, Kandidaten) sowie dynamischen (Stimmen) Daten unterschieden werden.

Eine Umladung sowohl mit statischen als auch dynamischen Daten fand mit Hilfe verschiedener Skripte statt, die auf Basis von tabellarischen Daten des Bundeswahlleiters CSV-Import-Dateien generieren, die anschließend in die Datenbank geladen wurden.

Zusätzlich können über das Webinterface ebenfalls Stimmen für die Bundestagswahl 2009 abgegeben werden (Simulation eines Wahlautomaten).

Für die Bundestagswahl 2009 werden alle Stimmen einzeln vorgehalten (d.h. pro abgegebener Stimme gibt es in der Datenbank einen eigenen Datensatz). Da die Ausführung von Analysen auf diesen Einzelstimmen jedoch ineffizient und langsam ist, werden diese nach Wahlkreis

¹<http://www.playframework.org>, Version 2.0

und Partei/Kandidat aggregierten Stimmen in speziellen Tabellen vorgehalten. Wird eine neue Stimme über das Webinterface abgegeben, werden die zugehörigen Datensätze der Aggregationstabellen auf ungültig (dirty) gesetzt, sodass sie durch ein regelmäßig zu startendes Wartungsskript neu berechnet werden können. Dieses Verfahren ermöglicht eine schnelle Delta-basierte Neuberechnung der Aggregationen mit minimalem Aufwand.

3 Performance

Die Leistungsfähigkeit unseres Systems haben wir mit einem selbst entwickelten Benchmark-tool geprüft. Die Ausgabe mehrerer Läufe des Benchmarks mit variierenden Parametern Anzahl Clients und Abstand zwischen Zugriffen findet sich mit in der Abgabe.

Solange das Verhältnis zwischen gleichzeitig zu bearbeitenden Anfragen und vorhandenen CPU-Kernen klein bleibt, bleiben auch die Antwortzeiten nahe an den minimal erreichbaren Werten. Wird die Anzahl gleichzeitiger Anfragen stark erhöht, zeigt sich eine deutliche Verlängerung der Antwortzeiten. Dies liegt darin begründet, dass das Datenbanksystem im Wesentlichen nur eine Abfrage pro CPU-Kern gleichzeitig bearbeiten kann. Der Grund dafür ist, dass - aufgrund eines großen Hauptspeichers und entsprechend großer Puffer und Caches für die Datenbank - praktisch alle Daten im Speicher gehalten werden können. Dadurch entfallen langwierige Zugriffe auf die Festplatte, die es dem Datenbank-Scheduler ermöglichen würden, andere Abfragen auszuführen.

4 Datenschutz und Sicherheit

Bei der elektronischen Stimmabgabe werden die abgegebenen Stimmen und alle zur ordnungsgemäßen Durchführung der Wahl notwendigen Daten grundsätzlich getrennt und ohne Verknüpfungsmöglichkeit gespeichert, um die Anonymität der Wahl zu sichern. Gleichzeitig werden durch die Ausnutzung der Transaktionseigenschaften des Datenbanksystems doppelte oder falsche Stimmabgaben verhindert.

Genauere Erläuterungen finden sich im beiliegenden Dokument zur Erläuterung der Stimmabgabe.