



Vorstellung Open Source Projekte

Steven Berg, Jan Christiansen, Max Winde, Manuel Kiessling

LargeXMLSitemap

Jan Christiansen

Was kann LargeXMLSitemap?

Kann Sitemaps mit sehr vielen URLs erstellen (für MyHammer ca. 75 Mio.)

Generiert auch die Sitemapindizes

Beachtet die Beschränkungen der Sitemaps (50000 URLs, 10MB pro Sitemap)

Legt beim Erreichen eines Limits automatisch eine neue Indexdatei an

Was kann LargeXMLSitemap nicht?

Webseiten crawlen und daraus die Sitemaps erstellen

Validieren der XML Dateien nach dem Generieren

Bestehende Sitemaps schnell bearbeiten

Wie funktioniert LargeXMLSitemap?

Baut das XML als String zusammen und schreibt es in eine Datei

Zum Löschen / Bearbeiten von bestehenden Sitemaps benutzt es die DOM Methoden

Warum nicht mit den DOM Methoden von PHP?

Geschwindigkeit: Die DOM Methoden sind um ein vielfaches langsamer, und das auch noch exponentiell mit der Anzahl der URLs

Benchmarks

Hinzufügen von URLs

Anzahl URLs	LargeXMLSitemap	PHP DOM
1000	0.10507	0.15021
2000	0.20032	0.46726
5000	0.43281	6.51824
10000	0.74687	36.41465
50000	6.30431	946.72994

<http://github.com/MyHammerOpenSource/LargeXMLSitemap>

IndexedMemcache

Steven Berg

Was ist IndexedMemcache?

Durchsuchbarer Memcache

Jedes Zeichen wird indexiert

Suchoperation: OR, AND, LIKE, IS NULL, ...

Index: keine Baumstruktur, da begrenzter Speicher pro Value

Objekt wird als Json gespeichert, da plattformunabhängig

Kein Datenbanksatz

Technischer Ablauf

TestObject: Objekt mit Username (Inhalt "max")

1. Objekt bekommt eindeutige Id zugewiesen (Guld)
 - a) Guld => Object (Json)
2. Guld wird mit fortlaufender Nummer verknüpft
 - a) fortlaufendeNummer => Guld
3. Indexerstellung:
 - a) Jedes Zeichen wird Indexiert
 - b) Aufbau: [Klassenname][Spaltenname][Version][Zeichen][Position] => Identifier
 - c) Beispiel:
 - I. cModelUser_username_1_m(0)
 - II. cModelUser_username_1_a(1)
 - III. cModelUser_username_1_x(2)
4. Suche:
 - a) Suchwort: m%
 - I. Key für Suche: cModelUser_username_1_m(0)
 - II. Alle Usernamen mit „m“ an Anfang stehen in diesem Index
 - b) Findet Identifier und lädt die dazugehörigen Objekte

Benchmarks

Testeinträge:

Id	Zahl
Username	10-20 Zeichen (Random)
Email	10-20 Zeichen (Random)

Insert:

Einträge	Datenbank	Memcache
100	000.022594 s	002.989397 s
1000	000.297856 s	017.590395 s
10.000	001.155497 s	122.207023 s
25.000	004.401448 s	275.205302 s

Search:

Einträge	Datenbank (a%)	Memcache (a%)	Datenbank (%a)	Memcache (%a)
35.000	000.002612 s	000.001780 s	000.048337 s	000.002929 s
50.000	000.004670 s	000.002145 s	000.078337 s	000.003314 s
75.000	000.015669 s	000.002871 s	000.098710 s	000.003903 s

<http://code.google.com/p/indexedmemcache/>

sqlHammer

Max Winde

Was ist sqlHammer?

PHP-App zum einfachen Ausgeben von SQL Queries

Einbau eines Queries in eine Seite:

```
<table sql="SELECT * FROM myTable"></table>
```

(+ ein <script>-include)

Abfragen können in statische HTML-Dateien eingebaut werden

Daten werden per Ajax/JSON(P) geladen

Simple, einfache Grafikfunktion zur Darstellung von Balkendiagrammen

Optionale Verschlüsselung der SQL-Queries um Manipulation der Queries zu verhindern

Leichte Anpassbarkeit der Ausgabe via CSS & jQuery / JavaScript

Integration von weiteren Bibliotheken einfach möglich

<http://github.com/MyHammerOpenSource/sqlHammer>

MyHammer