

Memcached

Lightning Talk
Software Freedom Day 2010

Daniel Hiller, dbap GmbH

Über mich

- Java seit 1996 (mit Unterbrechungen)
- Seit 2002 bei dbap
 - Backend Administration Software
 - Webservice als Zwischenschicht zur Datenermittlung für Frontends und externe Dienstleister

Caching - Wozu?



Problem:

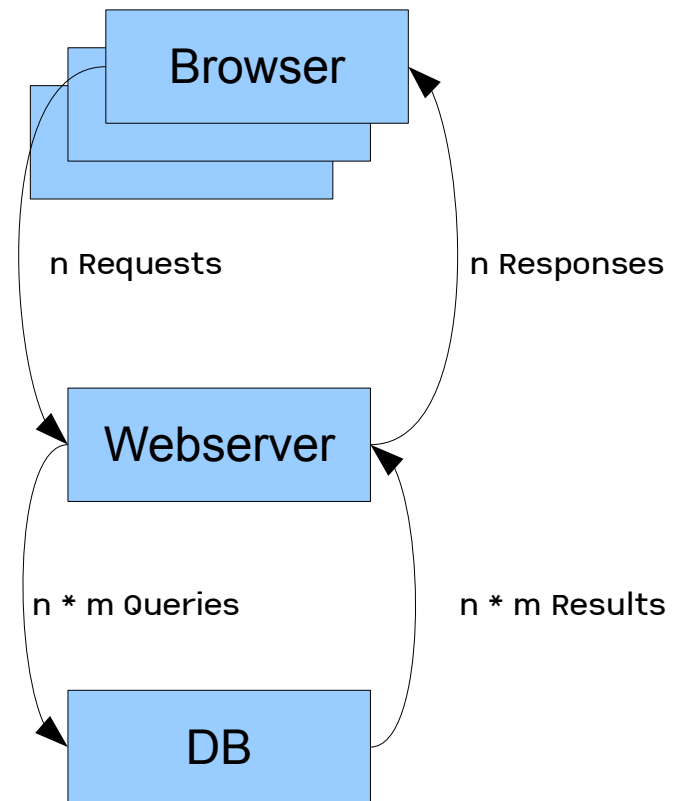
Schlechte
Antwortzeiten durch
lange dauernden
Vorgang

Beispiel

- n Benutzer fordern die selbe URL an
- Zum Erstellen der Response sind m umfangreiche Datenermittlungen nötig

dadurch:

- $n * m$ identische Datenermittlungen auf der Datenbank
- $n * m$ identische Ergebnismengen



Probleme bei hoher Konkurrenz

- Hohe Last auf Datenbankserver durch viele Zugriffe
- schlechte Antwortzeiten des Datenbankservers
- dadurch längeres Bestehen von Datenbank-Verbindungen
- dadurch längeres Bestehen von Webserver-Verbindungen
- Bei Überschreiten der maximalen Anzahl Verbindungen „weiße Seite“ oder Fehlerseiten

Lösung

- Caching des **Resultats** lang andauernder Vorgänge im Speicher, z.B.
 - Ergebnismengen
 - „select * from ...“
 - „exec stored_proc“
 - ...
 - Dokumente (HTML, PDF, XML, ...)

Codebeispiel ohne Caching

...

```
Result r = database.query(  
    „SELECT * FROM Questions WHERE Answer='42'“);  
final List<Question> questions = transform(r);
```

...

Codebeispiel mit Caching

...

```
final String key = „QuestionsForAnswer42“;  
List<Question> questions = memcache.get(key);  
if (questions==null) {  
    Result r = database.query(  
        „SELECT * FROM Questions WHERE Answer='42'“);  
    questions = transform(r);  
    memcache.set(key, expiryTimeInMsecs, questions);  
}
```

...

Kompromisse

- „Stale Data“
 - Daten haben ein gewisses maximales Alter (Cache-Expiration)
 - Kumulatives Alter durch mehrere Caching-Layer
- Einfügen eines weiteren Layers nötig
- Cache Warming
 - Initiales Füllen des Caches durch z.B. Script

Was ist Memcached?

- In memory caching system
- Key/Value Paare
 - (vgl. `Map<String, Object>`)
- Ablegen serialisierbarer Objekte
- Clustering
- telnet

Features

- Getmulti – Mehrere Werte gleichzeitig holen
- Asynchrones Lesen/Speichern
- Check-And-Set
- Transparente Kompression
- APIs für C/C++, PHP, Java, Python, Ruby, Perl, ...

Einschränkungen

- pro Schlüssel maximal 1MB
- Keine Möglichkeit, das Alter eines Datums zu ermitteln
- Keine Unterstützung von Kompression bei „append“
- Kein Einlesen von Serialisierten Cache-Daten möglich, um das Cache-Warming zu umgehen

Alternativen

- **Ehcache (Java-basierte Cache Server-Impl.)**
RESTful web service (GET/PUT/DELETE), Persistenz, Cache Warming, Clustering, Monitoring per JMX, Cache-Entry-Expiry
- **OSCache (J2EE caching framework)**
Object Caching, Persistenz, Clustering, Cache-Entry-Expiry
- **Cache4j**
Object Caching, Einfache API, schnelle Implementierung
- ...

Fragen?

Links

- <http://memcached.org/>
- <http://code.google.com/p/spymemcached/>
- <http://ehcache.org>
- <http://java-source.net/open-source/cache-solutions>
- <http://www.dbap.de>

dbap.

one-stop e-business