



Beispielkonfiguration bellybutton.de

Bellybutton (www.bellybutton.de) wird derzeit von den Autoren dieses Beitrags weiterentwickelt und basiert auf einem typischen Rails-Setup: Eingehende Requests werden von einem einzigen Apache-Webserver entgegengenommen und per Loadbalancer auf zwei Applikationsserver verteilt. Auf den Applikationsservern laufen jeweils fünf Mongrel-Prozesse, die die eigentliche Rails-Applikation ausführen. Im Hintergrund arbeiten zwei MySQL-Datenbankserver, wobei der Primärserver sämtliche Anfragen bearbeitet und der Sekundärserver nur im Fehlerfall zum Einsatz kommt. Damit dies funktioniert, treten die beiden MySQL-Server nach aussen als Cluster auf und werden während des Betriebs kontinuierlich repliziert. Fällt der Primärserver aus, schaltet MySQL automatisch um. Die Rails-Anwendung muss dafür weder speziell vorbereitet noch gesondert konfiguriert werden.

durch die Entwicklung eines eigenen Adapters einfach möglich. Was die Anzahl der Datenbankserver angeht, ist den Autoren dieses Artikels keine Rails-Anwendung bekannt, die aus Performance-Gründen mehr als einen Datenbankserver benötigt. Anders hingegen sieht es in bezug auf Ausfallsicherheit aus. Hier empfiehlt sich der Einsatz eines weiteren Datenbankservers, der im Falle eines Ausfalls die Arbeit übernimmt.

Abfahrt

Sind alle Plattformfragen geklärt und die benötigte Hardware beschafft und installiert, muss die Anwendung deployed werden. Deployment besteht aus einer Vielzahl von Einzelschritten: Auschecken der Software aus dem Versionskontrollsystem, Kopieren der Software auf den beziehungsweise die Produktionsserver, Aktualisierung und Migration der Datenbankschemata und einigen weiteren Aufgaben.

Mit Capistrano steht Rails-Entwicklern ein ausgereiftes Werkzeug für das automatisierte Deploy-

ment von Rails-Anwendungen zur Verfügung. Einmal konfiguriert, lässt sich die Anwendung von jedem Entwicklungsrechner aus auf Knopfdruck und ohne weiteres menschliches Zutun deployen. Deployment als fehleranfällige und zeitaufwändige Disziplin gehört damit der Vergangenheit an.

Last abfedern

Die Zahl potentieller Benutzer einer öffentlich zugänglichen Webanwendung ist nur schwer vorhersehbar. Steigende Benutzerzahlen erzeugen zunehmende Last, auf die das System vorbereitet sein sollte.

Zentraler Mechanismus für gute Performance ist Caching, bei dem dynamische Seiten nach ihrer Erzeugung zwischengespeichert und bei Folge-Requests direkt aus dem Cache geliefert werden. Rails-Caching ist für Produktionsumgebungen standardmässig aktiviert und steht in drei Varianten zur Verfügung: Seiten-Caching, Action-Caching und Fragment-Caching. Jede dieser Varianten hat ihre Vor- und Nachteile und muss entsprechend der jeweiligen Bedürfnisse konfiguriert werden.

ENVIRONMENTS FOR ELECTRONICS

www.knuerr.ch



DataCenter
InfraStructure®

Racks Security Cooling
Monitoring Power Services

White Paper für DataCenter
Klimakonzepte kostenlos
bestellen unter: info@knuerr.ch



Knürr AG - Bruggacherstrasse 16 - 8117 Fällanden
Tel. 044 / 806 54 54 - Fax 044 / 806 54 64
info@knuerr.ch - www.knuerr.ch



