



Nick Weisser

DER AUTOR



Nick Weisser Geschäftsführer bei der Schweizer E-Commerce Agentur Openstream und dort für Projektleitung und Qualitätskontrolle zuständig. Er verwendet gern Tools wie JIRA und Git, um Aufgaben und Code-Änderungen zu jonglieren. Als Early Adopter ist er stets von neuen Web-Technologien fasziniert.

HOSTING-PERFORMANCE IM GOOGLE-MASSSTAB

Über vier Millionen Apps basieren mittlerweile auf der bewährten Technologie der Google-Cloud-Plattform. Seit dem man erkannt hat, dass Geschwindigkeit für Nutzer wie für Suchmaschinen ein wichtiges Kriterium darstellt, tüfteln Experten daran, wie man Informationen vom Server schneller in die Browserfenster der Suchenden bekommt. Am Beispiel von WordPress schildert Nick Weisser beispielhaft die Möglichkeiten, eine Webseite oder einen Webshop mit Google App Engine performant in der Google Cloud zu betreiben.

Foto: ikerCollection / StockPhotoStockphotos.de

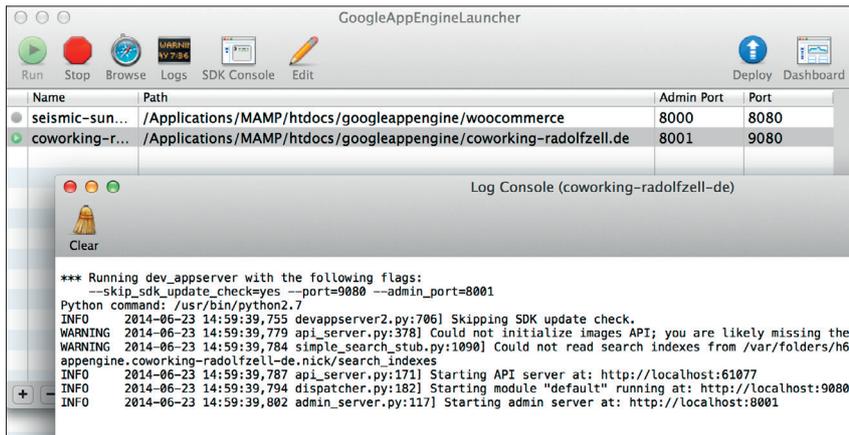


Abb. 1: Google App Engine Launcher mit Log-Konsole unter Mac OS X, nachdem lokaler Server gestartet wurde

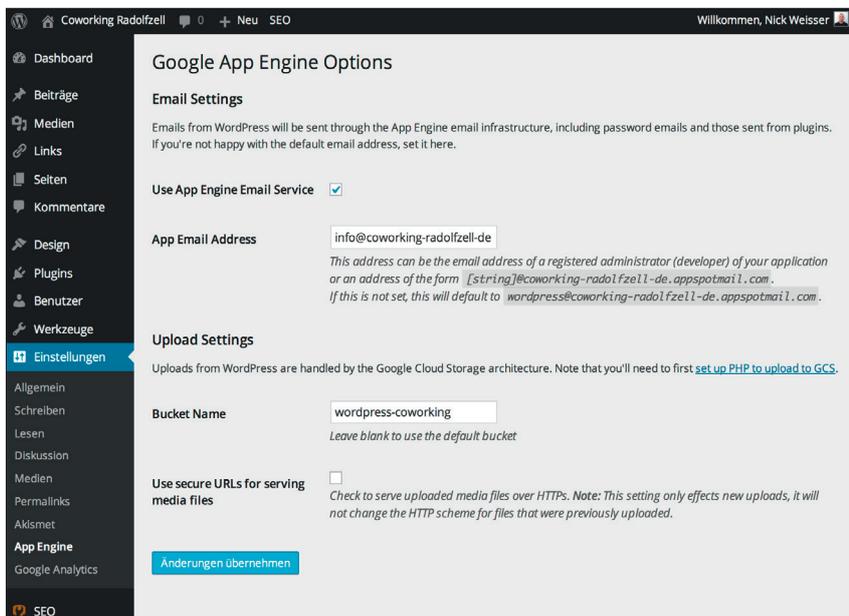


Abb. 2: Google App Engine WordPress – Konfigurationseinstellungen

Anfangs unterstützte [Google App Engine](#)* nur die Programmiersprachen Python, Java und Go. Bei der Developer-Konferenz Google I/O im Mai 2013 wurde die Limited Preview für [PHP](#)* bekannt gegeben und im Oktober 2013 folgte die allgemeine Verfügbarkeit. Dies ebnete den Weg für WordPress-Entwickler und -Anwender, das offene Blog-System, welches auch immer häufiger als Basis für Webseiten und Online-Shops verwendet wird, in der Google Cloud zu skalieren.

Komplexere Webshop-Systeme wie Magento oder Shopware benötigen oft ein speziell ausgerichtetes Shared

Hosting (mehrere Shops teilen sich einen Server) oder gar einen komplett eigenen Server (ein Shop pro Server), um größeren Besucherzahlen standzuhalten. Doch selbst ein kompletter Server, der für viele kleinere Shop-Betreiber bereits relativ kostspielig ist, kann zu Stoßzeiten, wenn z. B. ein Newsletter versendet wird oder bei einer Social-Media-Kampagne, schnell überlastet sein und Seitenaufrufe von 5-15 oder mehr Sekunden verursachen. Im schlimmsten Fall kommt es sogar zu Timeouts, sodass die Besucher eine Fehlermeldung statt der gewünschten Webseite zu Gesicht bekommen.

Dies ist weder für Besucher noch für Suchmaschinen akzeptabel und wird logischerweise von beiden als negativ eingestuft.

WordPress und WooCommerce

Für einfach konzipierte Shops ist WordPress mit einem Plug-in wie WooCommerce eine sehr interessante Alternative zu Magento oder Shopware. Die im Februar erschienene Version 2.1 von WooCommerce enthält sogar eine [REST API](#)*, welche Warenwirtschafts- und ERP-Anbietern die Anbindung an WooCommerce nicht nur erleichtern, sondern sehr schmackhaft machen wird.

Die Code-Basis von WordPress und WooCommerce ist sehr viel schlanker und eignet sich auch deshalb besser für Google App Engine, weil der Code in einem speziell für PHP-Applikationen konzipierten Container ausgeführt wird. Es gab bereits Versuche, Magento auf Google App Engine zu betreiben, das Projekt <https://github.com/kirmorozov/magento-gae> wurde aber wieder eingestellt. Es gibt allerdings vergleichbare Technologien für Magento, die ihre Wurzeln sogar im deutschen Rosenheim haben, jedoch unabhängig von der Google Cloud (siehe www.appserver.io). Das ist indes ein Thema für einen eigenen spannenden Artikel.

Bereitstellung im Google-Maßstab

Mit dem [PaaS](#)*-Dienst Google App Engine erhält man eine Bereitstellung im Google-Maßstab. Bei durchschnittlichen Besucherzahlen bezahlen Sie hierfür lediglich so viel wie für einen gewöhnlichen virtuellen Server. Wenn es darauf ankommt, können Sie allerdings auf bis zu sieben Milliarden Anfragen pro Tag hochskalieren und bei nachlassendem Traffic automatisch herunterskalieren. Abgerechnet wird exakt gemäß Nutzung der Ressourcen. Sieben Milliarden Anfragen werden

*siehe Online-Glossar unter www.websiteboosting.com

wohl die wenigsten in der Realität benötigen, aber selbst ein geteilter Link zu Ihrem Shop auf einem Facebook-Posting mit 10'000 Likes kann schnell mal 500 aktive Benutzer bedeuten und ein normales Shared Hosting in die Knie zwingen.

WordPress ist für seine Fünf-Minuten-Installation bekannt, mit Google App Engine ist das bei der ersten Einrichtung aber eher utopisch. Ohne Konsolenerfahrung wird es den meisten Shop-Betreibern, die auch selbst gern an Ihrem Shop „schrauben“, wohl eher schwerfallen, WordPress unter Google App Engine zum Laufen zu bekommen. Dies soll aber nicht abschrecken, sich selbst mit der spannenden Technologie auseinanderzusetzen.

In vier Schritten skalierbar

Schritte 1, 2 und 4 sind relativ simpel, bei Schritt 3 sind weniger erfahrene Entwickler und Anwender aber möglicherweise überfordert.

1. Erstellen eines Google-Developer-Accounts auf <https://console.developers.google.com/start/appengine>. Um dort später eine Datenbank anlegen zu können, wird eine Kreditkarte benötigt, obgleich der Service unter gewissen Mengenbeschränkungen der genutzten Ressourcen kostenlos ist.
2. Installation des Google App Engine Launchers unter Linux, OS X oder Windows unter <https://developers.google.com/appengine/downloads>.
3. Installation von WordPress + Google App Engine Plug-in <http://wordpress.org/plugins/google-app-engine/> in einer Testumgebung auf dem eigenen Rechner. Alternativ gibt es auch eine vorkonfigurierte WordPress-Version auf Github <http://googlecloudplatform.github.io/appengine-php-wordpress-starter-project/>.
4. Deployment („Upload“) der lokalen Installation auf Google App Engine.

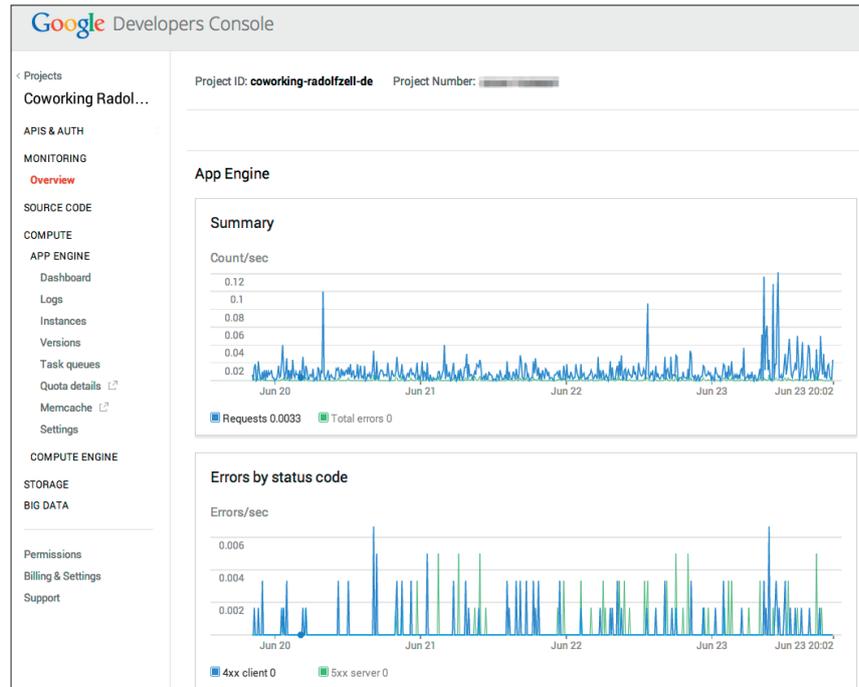


Abb. 3: Google-Developers-Konsole – Monitoring-Übersicht

Cloud Storage und Caching

Nachdem WordPress dann auch in der Cloud läuft, muss man in der App-Engine-Konfiguration von WordPress noch einen Cloud Storage Bucket hinterlegen, welcher zuvor in der Developers-Konsole eingerichtet wird. In einem solchen Bucket (dt.: „Eimer“) werden die Daten durch eine redundante Speicherung an mehreren physischen Standorten geschützt. Bilder und Dokumente, die über die Medienverwaltung hochgeladen werden, landen dann automatisch in einem persistenten Google-Cloud-Speicher mit eigener Sub-Domain, z. B. <http://wordpress-coworking.storage.googleapis.com/image.jpg>.

Wenn man es bis hierher geschafft hat, ist die automatische Skalierung bereits gewährleistet. Um die Seitenladezeiten noch weiter zu optimieren, kann man mithilfe des WordPress-Plugins Batcache zusätzlich noch Google Memcached nutzen, einem Cache-Server zum allgemeinen Hinterlegen und Abholen von Daten aus dem Arbeitsspeicher. Tests haben gezeigt, dass dadurch die Seitenladezeit ca. 40-mal reduziert werden kann, von 200 ms auf 5 ms.

Fazit

Google App Engine ist eine sehr mächtige Technologie zum Entwickeln und Hosten von Webanwendungen auf den Servern von Google. Die Einrichtung ist nicht so einfach wie der Upload von Dateien mit einem FTP-Programm bei einem gewöhnlichen Hosting-Anbieter. Die vollautomatische Skalierung erfordert hier einiges an Kenntnissen auf dem Niveau eines eher fortgeschrittenen Entwicklers.

Mit der Bekanntgabe erheblicher Preisermäßigungen und der Einführung neuer Funktionen und Entwickler-Tools während des Google-Cloud-Plattform-Live-Events im März 2014 ist die von Google bekannte Performance der Websuche nun auch für ambitionierte Start-ups und experimentierfreudige Entwickler verfügbar.¶