

[unimr.external_upload] Optimized File Uploads for ZODB Blobs via Nginx

Dr. Andreas Gabriel
Hochschulrechenzentrum



Motivation

Zope's ZODB implementiert mittels Blob Storage ein effizientes Datenbank-Konzept zur Verwaltung großer Dateien.

Jedoch in Bezug auf Datei-Uploads

- ist das Zope-seitige Decoding ineffizient (CPU, Speicher).
- wird während eines Uploads ein Zope Thread blockiert, d.h. es können zugleich keine anderen bzw. weniger Requests in Zope verarbeitet werden. Zope Threads sind per se „teuer“.

Fazit

Zope skaliert nicht für viele gleichzeitig stattfindende und/oder länger andauernde Datei-Uploads.

Lösungsansatz

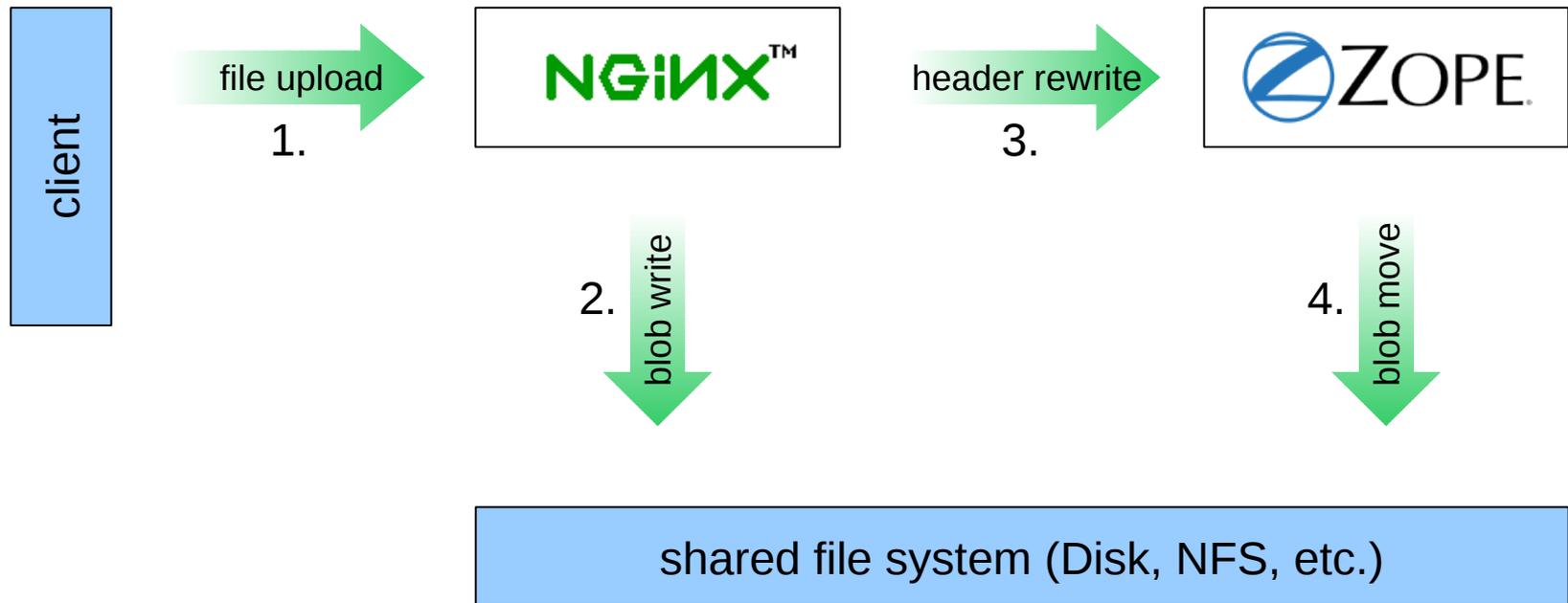
1. Frontend Server fängt Datei-Upload ab und legt die Datei direkt im Dateisystem ab.
2. Frontend Server teilt per Header-Anweisung Zope den Ort der Dateiablage mit.
3. Zope verschiebt die hochgeladene Datei von der Dateiablage in den ZODB Blob Storage.

Vorteile

- Threads des Frontend Servers sind „billig“ (Skalierbarkeit).
- Zope muss keine Datei-Uploads decodieren, sondern nur „Metadaten“ verarbeiten (Performanz).

Implementation [unimr.external_upload]

Optimized file Uploads for ZODB Blobs via Nginx



unimr.external_upload

Seiteneffekte

- Nginx und Zope müssen wegen gemeinsamer Zugriffsrechte auf Dateiebene unter gleicher User ID laufen

Voraussetzungen

- Zope2 (2.13 tested)
- Nginx (1.2.1) with third party „upload module“

Quelle

- http://github.com/hrz-unimr/unimr.external_upload

