

Einrichten eines Clusters zum Einspielen von Wikipedia-Traces

Praktikum Paralleles Rechnen (SS11)

Sebastian Menski

Institute für Informatik
Universität Potsdam

19. Juli 2011

Motivation

Erster Versuch

Neue Aufgabenstellung

Zweiter Versuch

Zusammenfassung

Hinweise zum Einspielen

Motivation

- ▶ Doktorarbeit von Jörg Zinke
- ▶ Server Load Balancing Benchmark
- ▶ Realistisches Szenario bzw. Anwendung
- ▶ Einspielen von originalen Wikipedia-Traces
- ▶ Vorhanden¹:
 - ▶ Wikipedia-Traces (18. September 2007 - 3. Januar 2008)
 - ▶ englischer Wikipedia-Dump (3. Januar 2008)

¹<http://www.wikibench.eu> [2]

Motivation

Leistungsanalyse: Messen, Modellieren und Simulation (WS10/11)

- ▶ Fabian Hahn [1]
- ▶ Perlbal Load Balancer mit httpperf und den Wikipedia-Traces testen
- ▶ Erkenntnisse:
 - ▶ Wikipedia-Dump enthält keine Bilder
 - ▶ Einspielen des Dumps ist zeitaufwendig und instabil
 - ▶ Es existiert ein Crawler um die Bilder nachzuladen
 - ▶ Die Thumbnails werden dynamisch beim Aufrufen des Wikipedia-Artikels erzeugt
 - ▶ Traces rufen Thumbnails direkt auf
 - ▶ Seriell aufrufen aller Wikipedia-Artikel sehr zeitaufwendig

Motivation

Aufgabestellung

- ▶ Aufsetzen mehrerer Wikipedia-Instanzen auf dem IB-Cluster
- ▶ Erstellen einer parallelen Anwendung zum Aufrufen aller Wikipedia-Artikel vom Leibniz-Cluster aus

Erster Versuch

Ausgangssituation

- ▶ Wikipedia-Dump: 14GB (6.226.028 Wikipedia-Artikel; nur englische Wikipedia)
- ▶ Bilder (Stand 24.05.2011):
 - ▶ 347GB
 - ▶ 984.641 Dateien
 - JPEG: 72,34%
 - PNG: 18,06%
 - SVG: 5,97%
 - GIF: 3,03%
- ▶ Mediawiki 1.16.5
- ▶ IB-Cluster: 8 Nodes
- ▶ Leibniz-Cluster: 15 Nodes

Erster Versuch

IB-Cluster (Stand: 24.05.2011)

Node	ib1-4	ib5/ib6	ib7/ib8
CPU	Intel Xenon 2,66GHz	Intel Pentium 4 2,8GHz	2x Intel Xenon 1,86GHz
Cores	1	1	2
RAM	1GB	4GB	4GB

Tabelle: IB-Cluster Nodes

- ▶ CentOS 5.6
- ▶ Alle Nodes 33GB /
- ▶ ib6/ib7 1TB /data

Erster Versuch

Leibniz-Cluster

Node	node001-010	node011-015
CPU	2x AMD Opteron 244 1,8GHz	Intel Xeon E5520 2,27GHz
Cores	2	4
RAM	4GB	6GB

Tabelle: Leibniz Nodes

Erster Versuch

Umsetzung (Stand: 24.05.2011)

ib1: LVS Load Balancer mit least connections ohne Gewichte

ib2-4: MySQL und Apache lokal (NFS ib6:/data)

ib5/ib8: MySQL und Apache lokal (NFS ib7:/data)

ib6: MySQL, Apache und /data lokal

ib7: MySQL, Apache und /data lokal

leibniz: Liste aller Wikipedia Artikel; Python Skript (Threading, Async, HTTP1.0/1.1)

Erster Versuch

Probleme

- ▶ Speicherplatz
 - ▶ MySQL Datenbank mit eingespielten Wiki-Dump: ca. 25GB
 - ▶ Speicher auf / verfügbar: ca. 29GB
 - ▶ Bilder müssen in Datenbank eingelesen werden (Speicherung von Metadaten)
 - ▶ Folge: MySQL Datenbank passt nicht mehr auf /
 - ▶ Lösung: MySQL über Netzwerk aber bereits /data über NFS
- ▶ Thumbnail Generierung:
 - ▶ Nicht genau feststellbar ob Thumbnails erfolgreich generiert wurden
 - ▶ ulimit Probleme (Lösung: Anpassung der Mediawiki Einstellungen)
 - ▶ Imagemagick segmentation fault beim Umwandeln von SVG (Redhat Bug im Paket gnome-vfs2)
- ▶ ib1-4 sehr alt und instabil
- ▶ Mediawiki alles andere als stabil und schnell

Erster Versuch

Fazit

- ▶ riesige Datenmengen
- ▶ Speicherplatz Mangel
- ▶ MySQL über Netzwerk und NFS verfälschen Messergebnisse
- ▶ Brauchen wir wirklich alle Bilder und Thumbnails?

Neue Aufgabenstellung

- ▶ Gespräch: Frau Schnor, Jörg Zinke und Sebastian Menski (09.06.2011)
- ▶ Resultat:
 - ▶ Jörg wählt 30 Minuten aus dem Trace aus
 - ▶ Sebastian erstellt eine Skript, das die Wikipedia Instanz für genau diese 30 Minuten einrichtetA
 - ▶ Filtern:
 - ▶ <http://en.wikipedia.org>
 - ▶ <http://upload.wikimedia.org/wikipedia/commons/>
 - ▶ <http://upload.wikimedia.org/wikipedia/en/>

Neue Aufgabenstellung

Trace Ausschnitt

tracefile: wiki.1194899823.gz

start time: Mon, 12 Nov 2007 18:31:30 +0000

end time: Mon, 12 Nov 2007 19:31:35 +0000

duration: 3604.986 sec

lines: 17.518.902

requests: 17.518.817

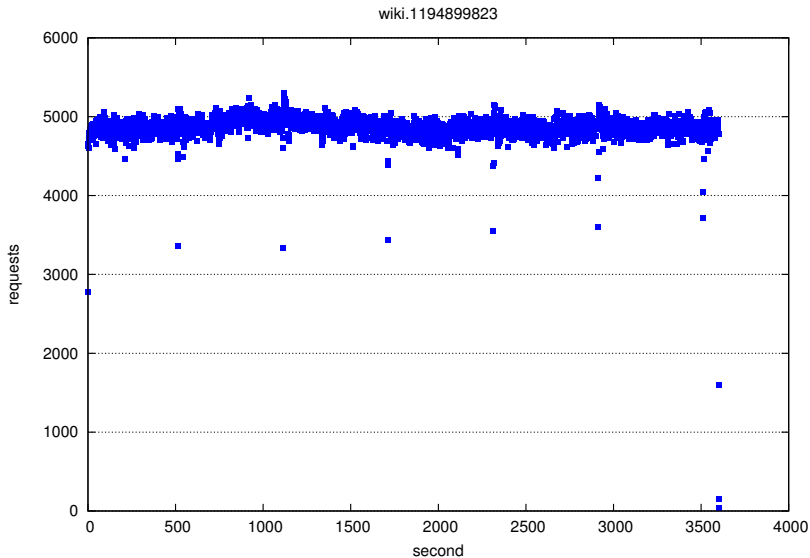
errors: 85

Trace Zeile:

5396292369 1194892290.546 http://en.wikipedia.org/wiki/Travis_Barker -

Neue Aufgabenstellung

Trace Ausschnitt - Request pro Sekunde



Neue Aufgabenstellung

Trace Ausschnitt (gefiltert)

tracefile: wiki.1194899823.1194892290-1194894090.gz

start time: Mon, 12 Nov 2007 18:31:30 +0000

end time: Mon, 12 Nov 2007 19:01:30 +0000

duration: 1800.788 sec

lines: 4.795.845

requests: 4.795.845

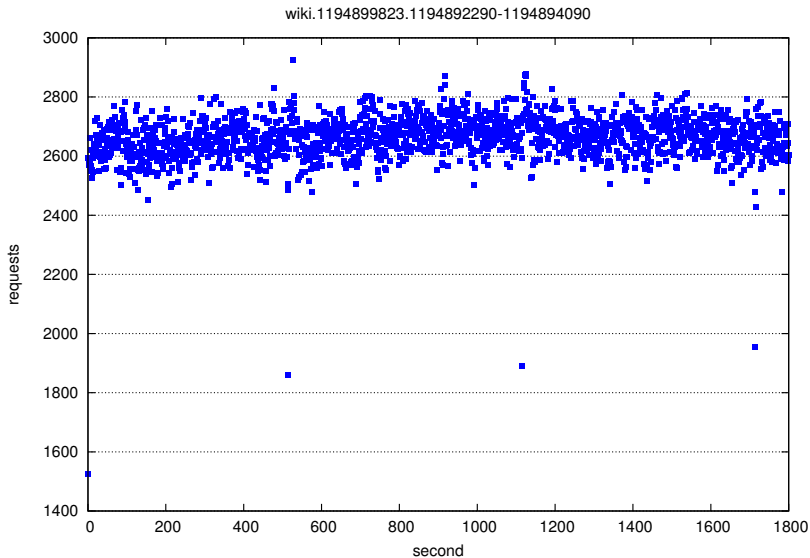
errors: 0

upload.wikimedia.org: 3.196.418 (66,6%)

en.wikipedia.org: 1.599.427 (33,3%)

Neue Aufgabenstellung

Trace Ausschnitt (gefiltert) - Requests pro Sekunde



Zweiter Versuch

Idee

1. Filtern des Traces
2. Download aller enthaltenen Bilder im gefilterten Trace
3. Download aller enthaltenen Thumbnails im gefilterten Trace
4. Importieren der Bilder in die Datenbank
5. Bilder und Datenbank packen
6. Bilder und Datenbank auf anderen Server installieren

Zweiter Versuch

Grundlagen

- ▶ Python2.6 (für den Download)
- ▶ Python2.4 (für die Installation auf den Servern)
- ▶ ib5-8: besitzen 1TB /data für die Datenbank
- ▶ ib2-4: lokal Bilder; Datenbank über Netzwerk
- ▶ ib1: LVS Load Balancer
- ▶ Verluste:
 - ▶ ib1/2/3: irreparabel defekt
 - ▶ ib1-4 durch anderen Maschinen ersetzt

Zweiter Versuch

IB-Cluster (Stand: 19.07.2011)

Node	ib1-4	ib5/ib6	ib7/ib8
CPU	2x AMD Opteron 244 1,8GHz	Intel Pentium 4 2,8GHz	2x Intel Xenon 1,86GHz
Cores	2	1	2
RAM	4GB	4GB	4GB

Tabelle: IB-Cluster Nodes

- ▶ CentOS 5.6
- ▶ ib1: 143GB /
- ▶ ib2: 69GB /
- ▶ ib3: 31GB /
- ▶ ib4: 69GB /
- ▶ ib5-8 33GB /; 1TB /data

Zweiter Versuch

Python Skript I

Code: <https://github.com/menski/ppr-s11>

- Files:
- ▶ `setup_env.py` (Hauptskript; benötigt Konfigurationsdatei)
 - ▶ `example.cfg` (Beispiel Konfiguration)
 - ▶ `ppr/`
 - ▶ `basic.py` (Basis Klassen)
 - ▶ `http.py` (Klassen für HTTP Requests und Download)
 - ▶ `server.py` (Funktionen und Klassen zum ausführen von Befehlen)
 - ▶ `trace.py` (Klassen zum Analysieren und Filtern von Traces)

Zweiter Versuch

Python Skript II

- ▶ Fast jede Klasse ist ein Prozess
- ▶ Ablauf genau durch Konfigurationsdatei steuerbar

Zweiter Versuch

Konfiguration

example.cfg

Zusammenfassung

- ▶ Bilder die in der aktuellen Wikipedia existieren werden heruntergeladen
- ▶ Das sind nicht alle Bilder
- ▶ Das Skript ermöglicht die volle Automatisierung der Einrichtung der Testumgebung
- ▶ Sofern Public-Key Authentifizierung möglich, komplett ohne Nutzerinteraktion
- ▶ Hohe Laufzeit, da große Datenmengen verarbeitet und transportiert werden müssen
- ▶ Zeitintensiv: Packen und übermitteln der Bilder und der Datenbank

Hinweise zum Einspielen

- ▶ Der Response auf ein nicht existierendes Thumbnail ist '500 Internal Server Error' und nicht '404 Not Found'

References I



Fabian Hahn.

Server load balancing: Wikipedia.

Leistungsanalyse: Messen, Modellieren und Simulation.



Guido Urdaneta, Guillaume Pierre, and Maarten van Steen.

Wikipedia workload analysis for decentralized hosting.

Elsevier Computer Networks, 53(11):1830–1845, July 2009.

http://www.globule.org/publi/WWADH_comnet2009.html.