

# Skalierbarkeit von Serversystemen

Was tun, damit das technische Korsett mein Business nicht einschnürt?



Dipl.-Inform. Thomas Witzenrath ■ 21.01.2014

### *Dipl.-Inform. Thomas Witzenrath*

- Studium der Informatik in Karlsruhe
- Seit 15 Jahren Erfahrung im Bereich Servermanagement
- Seit 2010: Geschäftsführer der Tralios IT GmbH: Betrieb von Linuxbasierten Web-/Mailservern
- Betreuung von über 200 Kunden in ganz Deutschland

## Typischer Webserver

- Webserver mit Datenbank
- Auslieferung von dynamischem und statischem Content



# Wissen wo der Engpass ist

Welcher Teil meiner Webseite belastet welche Teile der Hardware?

- **Statische Inhalte:** lesen von Festplatte (IO)
- **Datenbankzugriffe:** lesen/schreiben von/auf Festplatte (IO), CPU
- **Dynamische Inhalte:** lesen von Festplatte (IO), CPU



### Typischer Skalierungsablauf eines Projekts

- virtueller Server
- eigene Hardware
- Ausbau der Hardware - mehr RAM, mehr/schnellere Festplatten, etc.

### Typischer Skalierungsablauf eines Projekts

- virtueller Server
- eigene Hardware
- Ausbau der Hardware - mehr RAM, mehr/schnellere Festplatten, etc.



Engpass ist meist beim Festplattenzugriff

## Das Skalierungsproblem

- Hardware skaliert nicht unbegrenzt
- Daher Verteilung auf verschiedene Systeme

## Das Skalierungsproblem

- Hardware skaliert nicht unbegrenzt
- Daher Verteilung auf verschiedene Systeme

aber

- Richtiges System muss angesprochen werden
- Systeme müssen miteinander kommunizieren

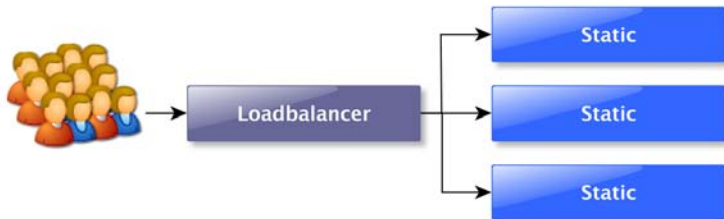


## Wie lässt sich weiter skalieren?

- Logische Einheiten voneinander trennen
- Auslagern verschiedener Subsysteme auf eigene Hardware
- Vervielfältigen der Systeme

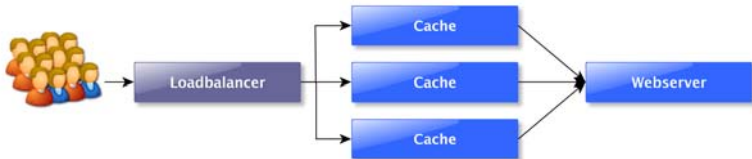


- Trivial verteilbar: unabhängige Webserver
- Lastverteilung über:
  - Subdomains
  - Round Robin DNS
  - Loadbalancer



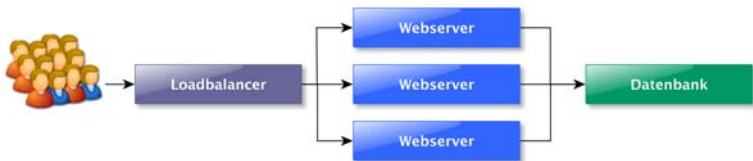
## Quasi-Statische Daten

- Geringe Änderungsrate
- Stark unterschiedliche Zugriffszahlen
- Berechnen der Daten aufwändig
- Einsatz von Caches



## Dynamischer Content, Applikation

- Bei jedem Aufruf Neuberechnung notwendig
- Verteilung auf mehrere Webserver
- Probleme:
  - Session
  - Datenkonsistenz



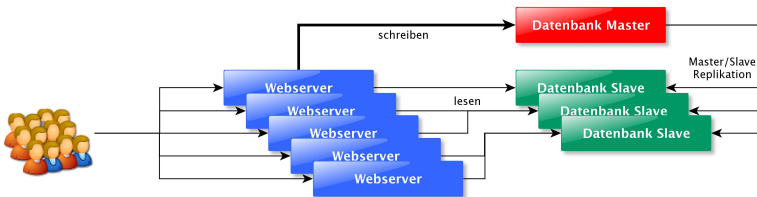
## Skalieren von Datenbanken

- Aufteilen der Tabellen auf mehrere Datenbanken  
aber: keine Tabellenverknüpfungen, Konsistenzproblem

## Skalieren von Datenbanken

- Aufteilen der Tabellen auf mehrere Datenbanken  
aber: keine Tabellenverknüpfungen, Konsistenzproblem
- Typisch:
  - viel lesen
  - wenig schreiben

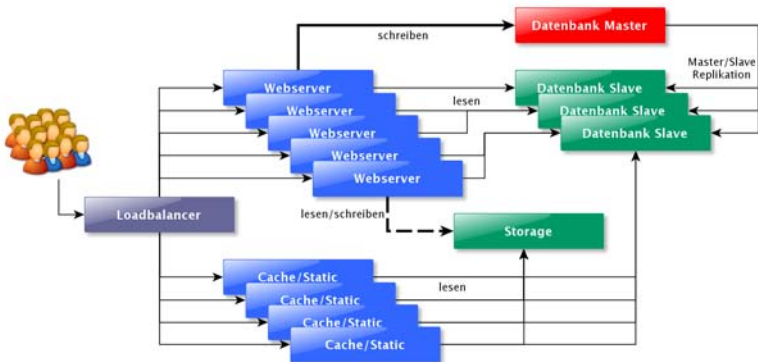
### Master/Slave-Replikation



## Konkretes Beispiel

- Online-Community
- 500.000 angemeldete Benutzer
- Regelmäßig 15.000 Benutzer gleichzeitig aktiv
- 25 Server, ca. 300 GB Arbeitsspeicher, mehrere Gbit/s Traffic

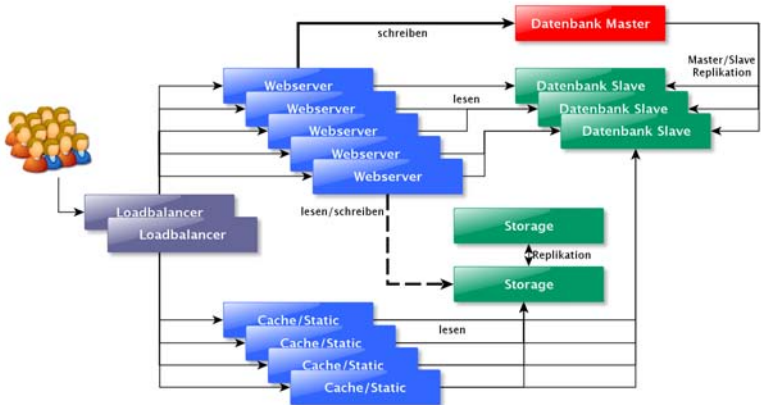
## Konkretes Beispiel





- Ausfallsicherheit inklusive? **Fast!**
- Nicht redundante Systeme:
  - Loadbalancer
  - Storage
- durch Zweit-Systeme absichern:
  - Loadbalancer im HA-Cluster
  - Storage per DRBD spiegeln

# Ausfallsicherheit inklusive



## Wo hilft mir Cloud Computing?

- Fertige Lösungen erhältlich (Storage, DB-Replikation)
- Keine Hardware-Beschaffung
- Flexibilität



ideal für StartUps!

## Wo hilft mir Cloud Computing?

- Fertige Lösungen erhältlich (Storage, DB-Replikation)
- Keine Hardware-Beschaffung
- Flexibilität



ideal für StartUps!

Aber:

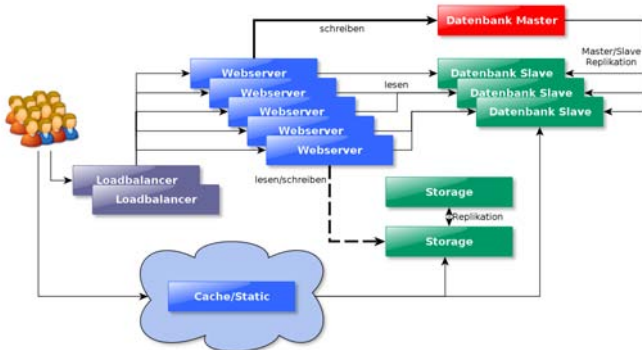
- Kosten bei großen Instanzen vergleichsweise hoch
- Datenschutzproblematik
- Gedacht für dynamische Zugriffszahlen



Kurzzeitige Erweiterung der Basis-Infrastruktur

# Clouddienste einbinden - Ein Beispiel

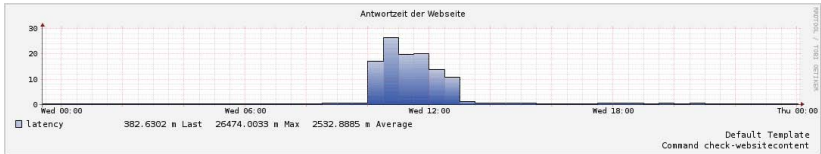
Verteilung statischer Inhalte über Cloudserver



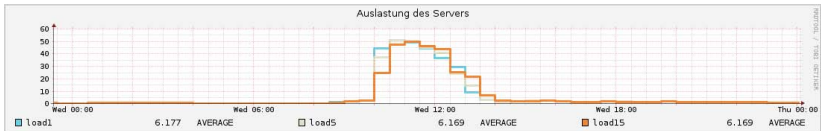
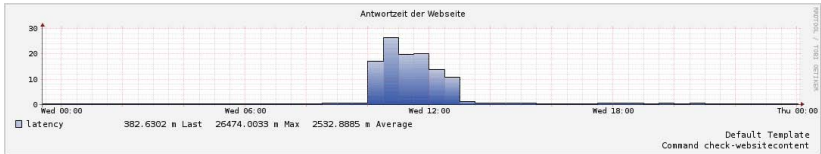
## Wann skalieren?

- System dauerhaft überwachen:
  - Veränderungen erkennen
  - Bedarf ableiten
- Performance bricht typischerweise schlagartig ein
- Daher vorher Messung über einfache Seitenzugriffe, komplexe eigene Tests

# Wann skalieren?

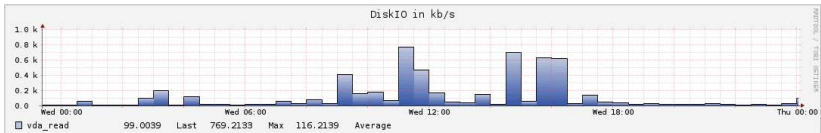
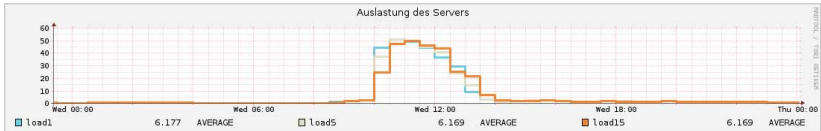


# Wann skalieren?

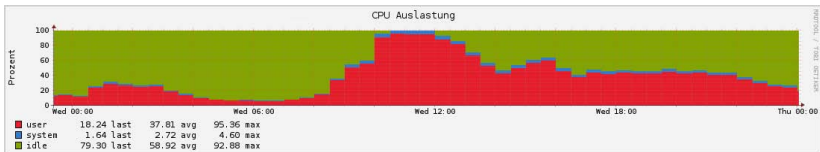
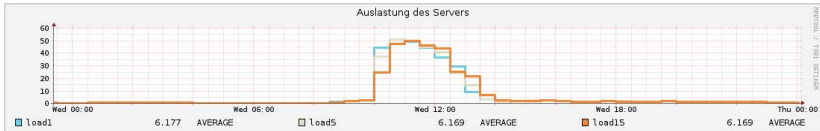




# Wann skalieren?



# Wann skalieren?



## Wann skalieren?

- Nicht alles lässt sich durch mehr Hardware lösen



Webseiten- und Datenbank-Optimierung nicht vernachlässigen

## Wann skalieren?

- Nicht alles lässt sich durch mehr Hardware lösen



Webseiten- und Datenbank-Optimierung nicht vernachlässigen

### Beispiel MySQL

```
mysql>select id from users where status='confirmed' order by id desc limit 1;
```

vs.

```
mysql>select max(id) as id from users where status='confirmed';
```



■ *Dipl.-Inform. Thomas Witzentrath*

Tralios IT GmbH

Bannwaldallee 46

76185 Karlsruhe

Telefon: 0721 - 94269660

E-Mail: [witzentrath@tralios.de](mailto:witzentrath@tralios.de)