

# Cloud-Plattform: Appscale Hochschule Mannheim

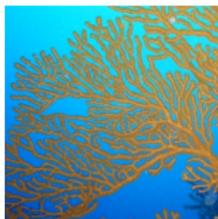
Florian Weispfenning

Fakultät für Informatik  
Hochschule Mannheim

`florian.weispfenning@stud.hs-mannheim.de`

18.12.2009

# Agenda



# AppScale



- Appscale <sup>1</sup>
  - Einführung
  - Voraussetzung
  - Komponenten
- Aufbauten
- Werkzeuge
- Demo
- Visionen
- Fazit

---

<sup>1</sup><http://appscale.cs.ucsb.edu/>

# Appscale - Open-source platform-as-a-service (PaaS)

- Projekt der UCSB (University of California, Santa Barbara)
  - 7.3.2009 Veröffentlichung von 1.0 <sup>2</sup>
- “emuliert” Google App Engine (GAE)
  - Java & Python Programme
- Version 1.3 Beta Anfang Dezember
- Version 1.3 seit Montag 14.12.09

---

<sup>2</sup><http://en.wikipedia.org/w/index.php?title=AppScale>

# Appscale - Auslieferung

- Server (als VM-Dateisystem)
  - 6 GB Dateisystem (komplettes Appscale-System)
  - 2.6.27 Kernel
  - Xen-Konfiguration
- Administrations-Konsole
  - ein Archiv
  - Kommandozeilen-Werkzeug

# Warum Appscale

- Abhängigkeit zu PaaS-Anbietern aufweichen
- kompatibel zu GAE
- eingebaute Authentifizierung (ohne Google-Accounts)
- Datenbanken<sup>3</sup>
  - HBase
  - Hypertable
  - MySQL
  - Cassandra
  - Voldemort
  - MongoDB
  - MemcacheDB
- komplettes transparentes System
- Forscher
  - Verhalten einer Cloud
  - Verhalten von Anwendungen in einer Cloud

---

<sup>3</sup><http://code.google.com/p/appscale/>

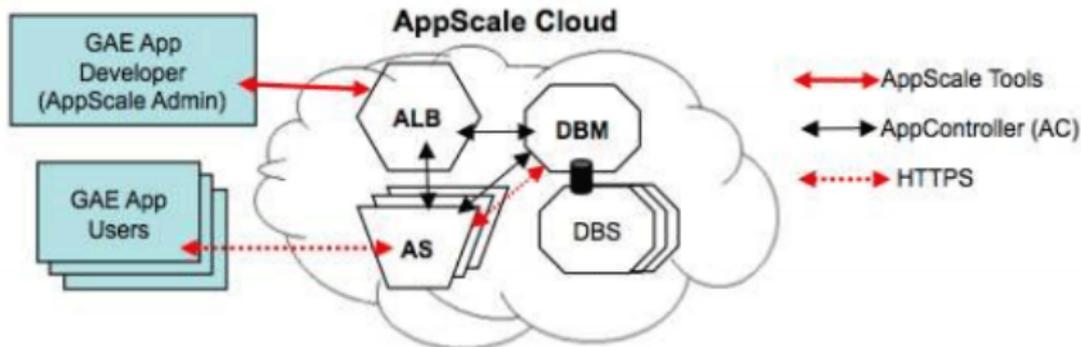
# Voraussetzungen



- Virtualisierungsschicht (Xen oder KVM)
  - eigene Hardware
    - pro VM 0,5GB - 4GB RAM
    - 1 (besser 2) Kerne
  - EC2
  - Eukalyptus
- Ohne Virtualisierung (Ubuntu Jaunty)

# Komponenten

- AppController (AC)
- AppLoadBalancer (ALB)
- Database Master/Peer (DBM)
- AppServer (AS)
- Database Slave/Peer (DBS)



# AppController (AC)

- in Ruby geschrieben
- Verwaltete alle Appscale-Instanzen
- Darstellung aller Anwendungen
- Startet zum Start den “Database Master” (DBM)
- Iniziiert das “deployment”
- Startet konfigurierte Appscale-VMs
- “heartbeat” von 10 Sekunden zu allen VMs

# AppLoadBalancer (ALB)

- Ruby on Rails - Programm
- nginx <sup>4</sup>
- Benutzer-Login
- Verteilt die Anfragen der Anwender zufällig
- leitet direkt zum AppServer weiter

Situationen um ALB anzufragen

- Erster Aufruf des Programms
- Logout
- aktueller AppServers offline

---

<sup>4</sup><http://nginx.org/>

# AppServer (AS)

- in Python
- basiert auf “dev\_appserver.py” (GAE)
  - Mini-Entwicklungsserver von Google
  - Datei als Datenbank (Google)
  - Generische Datenbankschnittstelle (AppScale)
- Speichert HTTP-Cookie lokal

# Datenbankschnittstelle

## Funktionalität

- “put” - neues Element in Tabelle
- “get” - Element über ID aufrufen
- Abfragen (Sprache ist SQL-ähnlich)
- “delete” - Löschen eines Elements anhand einer ID

# Project Voldemort - Verteilte Datenbank

Key-Value Datenbank

Alternativen <sup>5</sup>

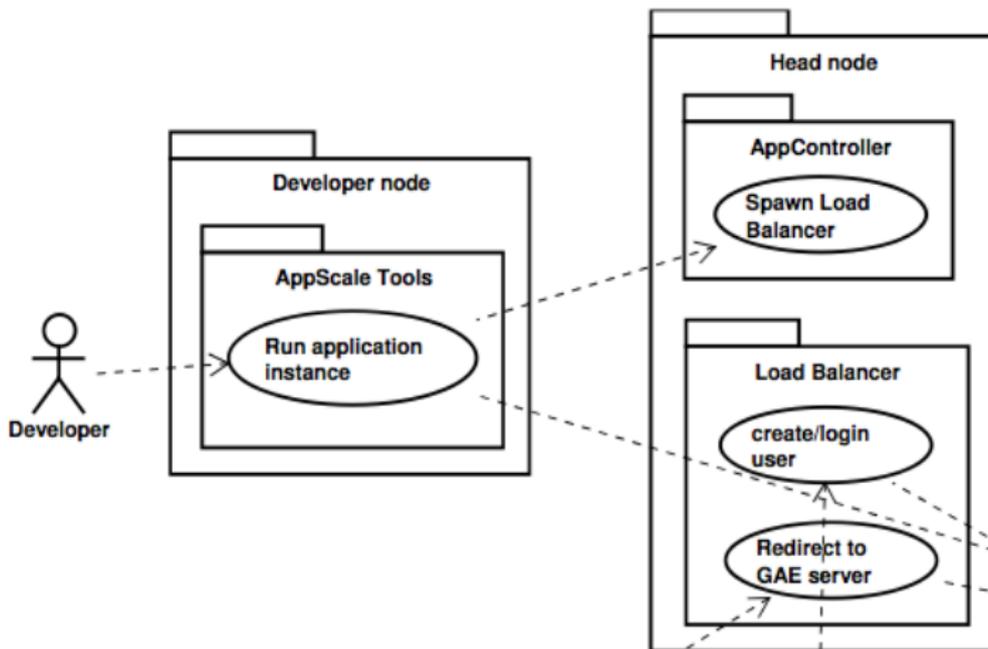
- Amazon's Dynamo
- Google's BigTable

---

<sup>5</sup>[http://en.wikipedia.org/wiki/Project\\_Voldemort](http://en.wikipedia.org/wiki/Project_Voldemort)

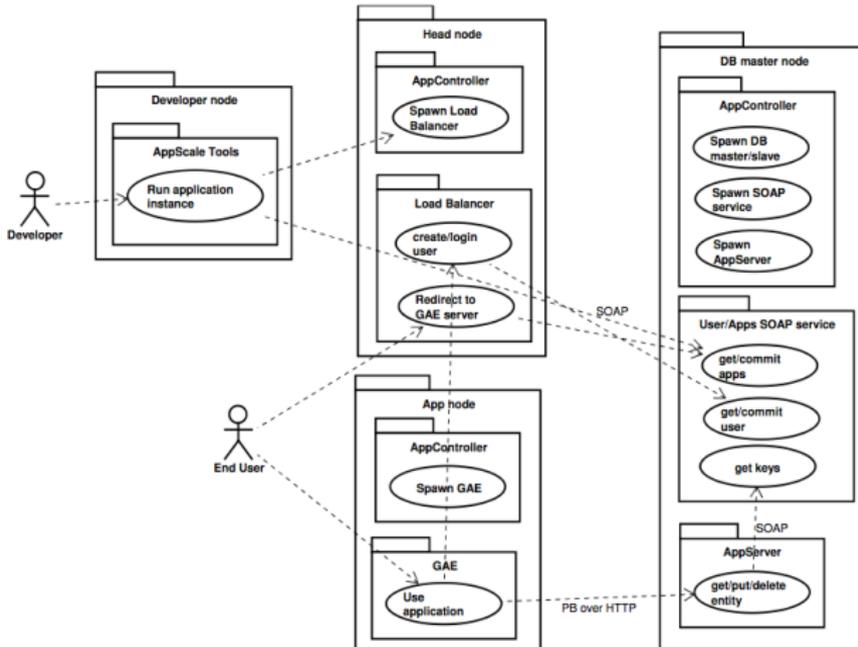
# Entwicklungsaufbau

- eine AppScale-Instanz



# Produktivlandschaft

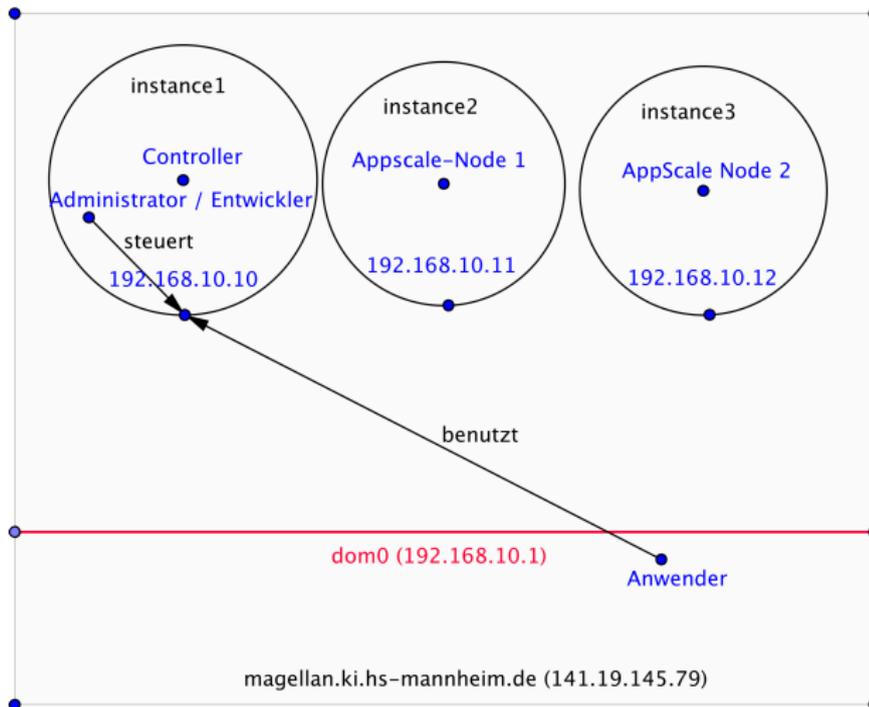
- mehrere AppScale-Instanzen



# Aufbau in der Hochschule Mannheim

- ein Desktop-Rechner
  - 2.4 GHz
  - 2 Kerne
  - 3 GB RAM
  - 250 GB Festplattenspeicher
- Platz für 3 AppScale Instancen

# Aufbau in der Hochschule Mannheim



# Werkzeuge

- Für Entwickler Administrator
- ruby-Skripte
- Nutzt SSH-Schlüssel zur Authentifizierung

	<b>Befehl</b>	<b>Beschreibung</b>
1	appscale-add-keypair	Erstellt SSH-Schlüssel für jede Server Instanz
2	appscale-describe-instances	Beschreibt Status aller Server Instanzen
3	appscale-remove-app	Löscht ein Programm von allen Servern
4	appscale-reset-pwd	Administrator-Passwort setzen
5	appscale-run-instances	Alle Server Instanzen starten
6	appscale-terminate-instances	Alle Server Instanzen stoppen
7	appscale-upload-app	Programm auf allen Servern installieren

Keyname-Konzept (mehrere Clouds von einem Konfigurationstool verwaltbar)

# Server-Konfiguration

Nur bei appscale-add-keypair

- 1x Hauptserver (Load-Balancer + Datenbank)
- 2x Server

ips.yaml

```
—  
:controller: 192.168.10.10  
:servers:  
- 192.168.10.11  
- 192.168.10.12
```

# Schritte zu einer funktionierenden Anwendung

Ziel: Beispielprogramm *guestbook.tar.gz* (*Administrator-Werkzeugen*)

- *Eine ips.yaml erzeugen*
- *appscale-add-keypair -ips ips.yaml*
- *appscale-run-instances -file guestbook.tar.gz -table voldemort*

Über "Controller" kann nun auf die Cloud zugegriffen werden.

# Demo

## Demo

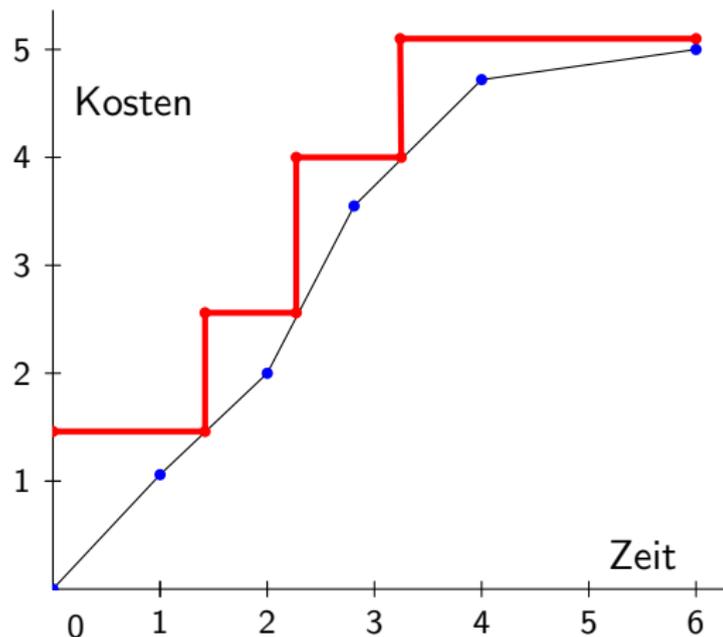
- AppScale Maschinen starten
- “Cloud” starten
- Beispielprogramm hochladen

# Demo

## Demo

- Verteilte Datenbank demonstrieren.

# wachsende Firma



## Wachsende Serverkosten in einem Unternehmen

# Alternativen

- Google App Engine <sup>6</sup>
- AppDrop (speziell für Amazon EC2) <sup>7</sup>
  - proof-of-concept
  - in 4 Tagen implementiert
  - Datenspeicherung in Datei
  - Alles auf einem Server
  - keine Flexibilität
  - keine Datenredundanz

---

<sup>6</sup><http://code.google.com/intl/de-DE/appengine/>

<sup>7</sup><http://jchris.mfdz.com>

# Fazit



- komplexes System
- keine "One-Click" Installation :-(
- 32 Sekunden lange Berechnungen ;-)
- häufig neue Versionen
- benötigt Hardware



- kompatibel zu GAE
- dynamisch wachsendes System
- komplette Kontrolle über die Daten
- keine Internetanbindung nötig
- aktive Entwicklung sichtbar

- Forschung
- Offline-Installationen

Hier könnte sich etwas entwickeln. . .

# Danke

- AppScale - Support
  - Appscale Community <sup>8</sup>
- Xen - Support
  - Maximilian Hoecker
  - Benjamin Simroth
  - Tobias Mache
- Amazon EC2 - Support
  - Peter Gehrt
- AppEngine - Support
  - Peter Sutter

---

<sup>8</sup>[http://groups.google.com/group/Appscale\\_Community](http://groups.google.com/group/Appscale_Community)

# Fragen

Gibt es noch Fragen?

Ende

Vielen Dank für Ihre Aufmerksamkeit

# Quellen

- “AppScale Design and Implementation” (AppScale Tech Report <sup>9</sup>)
- AppScale Präsentation von Juli 2009

---

<sup>9</sup>[http://groups.google.com/group/appscale\\_community/files](http://groups.google.com/group/appscale_community/files)