

Магистерская диссертация на тему:

Балансировка нагрузки приложений в «облачной» среде

Выполнил: студент группы ДА-32м

Бабенко А.А.

Руководитель: Гиоргизова-Гай В.Ш.

Задачи

- Провести систематизацию распространенных методов и алгоритмов балансировки нагрузки приложений в «облачных» системах.
- Сделать сравнительную характеристику распространенных систем балансировки нагрузки.
- Предоставить практические рекомендации по настройке системы балансировки нагрузки платформы CloudStack на примере приложения дистанционного видеообучения.
- Экспериментально исследовать эффективность созданных правил балансировки нагрузки для выбранного приложения.

Балансировка нагрузки

Метод распределения задач между несколькими сетевыми устройствами с такими целями:

- оптимизация использования ресурсов;
- сокращение времени обслуживания запросов;
- горизонтальное масштабирование;
- обеспечение отказоустойчивости

Способы балансировки



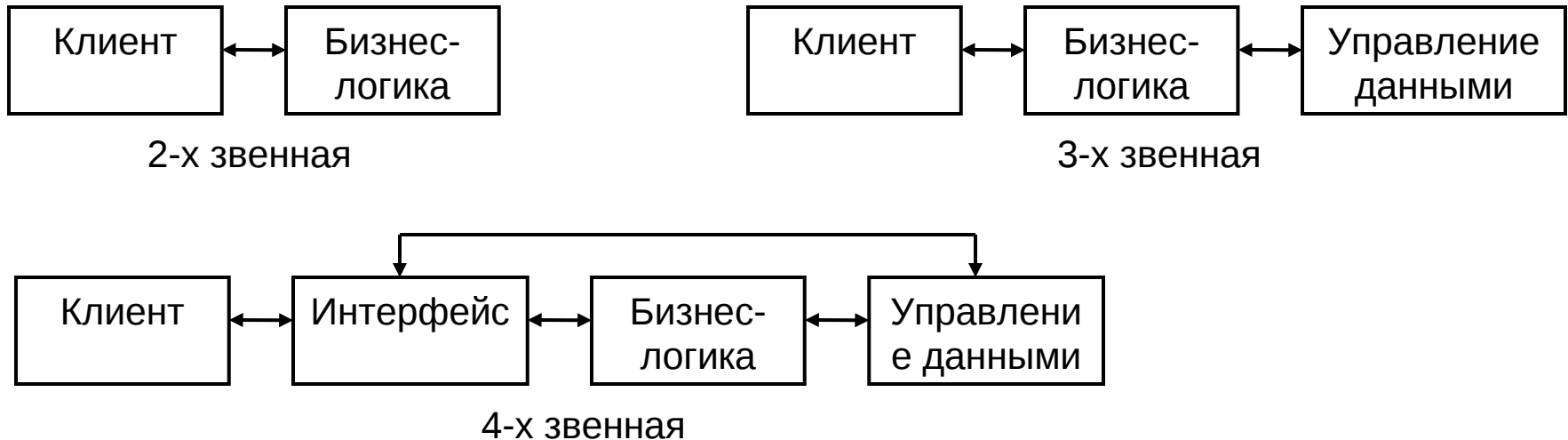
Уровни балансировки

- **канальный** (LVS);
- **сетевой** (Google Compute Engine, Elastic Load Balancer);
- **транспортный** (Nginx, HAProxy, LVS, Google Compute Engine, Azure LoadMaster, Elastic Load Balancer, CloudStack);
- **прикладной** (Nginx, HAProxy, Google Compute Engine, Azure LoadMaster, Elastic Load Balancer).

Критерии сравнения методов балансировки

- Эффективность распределения нагрузки
- Показатели загрузки узлов
- Учет гео-положения узлов
- Возможности масштабирования
- Контентно зависимый анализ, кэширование
- Обеспечение долгосрочных сессий
- Возможность неоднородности мощностей и платформ ВУ
- Независимость от типа сети
- Проверка работоспособности (здоровья) ВУ
- Независимость от типа прикладного протокола
- Затраты ресурсов на балансировку

Типовые архитектуры веб-приложений



В зависимости от требований бизнес-логики и физической архитектуры обслуживающих узлов балансировка нагрузки производится при помощи:

- На уровне точки входа – распределенные методы, NLB кластер, методы с единым устройством 3 – 4 уровней OSI
- На уровне бизнес-логики – методы с единым устройством 7 уровня OSI (для 4-х звенной), с единым устройством 3 – 4 уровней OSI
- На уровне управления данными – средства ОС, БД, методы с единым устройством 3 – 4 уровней OSI

Средства балансировки нагрузки в «облачных» средах

	Google Compute Engine	Microsoft Azure LoadMaster	Amazon EC2 Elastic Load Balancing	HAProxy	LVS	Nginx	CloudStack
Эффективность распределения	+	+	+	+	+	+	+
Учет гео-положения ВУ	+	-	+	-	-	-	-
Возможности автомасштабирования	+	+	+	-	-	-	+
Контентно зависимый анализ, кэширование	+	+	+	+	+	+	+
Обеспечение длительных сессий	+	+	+	+	+	+	+
Независимость от типа сети	+	+	+	+	+	-	-
Проверка здоровья ВУ	+	+	+	+	-	+	+
Незалежність від прикладного протоколу	+	+	+	+	+	-	+

Создание облачного сервиса в CloudStack

- Создать шаблон из Ubuntu 14.04 64 бит
- Создать VM из шаблона и настроить ее
- Установить BigBlueButton на VM
- Создать шаблон с готовой машины
- Зделать вторую копию VM

Веб-интерфейс BigBlueButton

Opera BigBlueButton - Demo Meeting 192.168.1.103/client/BigBlueButton.html

Demo Meeting Russian Клавиши быстрого доступа

Пользователи Презентация: default Чат

Статус	Имя	Медиа
	Вабенко (вы)	

Презентация: default

Welcome!

To get started quickly, click an image below to watch a tutorial video.

Moderator/Presenter

Viewer

You control the audio and can make anyone (including yourself) the presenter.

You can view, chat, and listen. If enabled, you can also unmute and share your webcam.

A green checkmark appears when you have joined the audio bridge. Click to join/leave the audio.

Click to make this user the presenter.

Click to mute/unmute this user's microphone.

Click to lock this user's microphone (user can't mute/unmute).

(launch [test applet](#) for testing desktop sharing)

Чат

Все Настройки

00:00

Welcome to **Demo Meeting!**

For help on using BigBlueButton see these (short) [tutorial videos](#).

To join the audio bridge click the headset icon (upper-left hand corner). Use a headset to avoid causing background noise for others.

This server is running [BigBlueButton 0.81](#).

Отправить

100% 400%

(c) 2013 BigBlueButton Inc. [build 4359-2014-07-30] - Для получения дополнительной информации посетите <http://www.bigbluebutton.org> Default

Правила балансировки

- Алгоритм: из трех доступных в CloudStack в данном случае целесообразнее использовать алгоритм Least Connection, так как при работе с данным приложением предусматриваются длительные сессии, что может привести к неравномерной нагрузке при использовании других алгоритмов.
- Поддержка длительной сессии: нужна, чтобы один пользователь имел возможность присоединиться к одной и той же машине. Не требуется сессия дольше суток, потому что достаточно политики липкости по IP адресу пользователя.
- Проверка работоспособности: нужна. Зададим опрос каждые 10 секунд.
- Автомасштабирование: требуется. Масштабирование вверх будем осуществлять при использовании CPU или RAM больше чем на 80%, а вниз при использовании CPU или RAM меньше чем на 20%.

Настройка правил балансировки

Load Balancing

Name	Public Port	Private Port	Algorithm	Stickiness	Health Check	Add VMs	Actions
web	80	80	Least conn	Configure	Configure	Add	

Stickiness method

SourceBased

Sticky Name

Test

Table size

200k

Expires

50h

Save

Cancel

Ping Path

Response Timeout (in sec)

10

Health Check Interval (in sec)

10

Healthy Threshold

1

Unhealthy Threshold

1

Save

Cancel

Template: Compute offering: * Min Instances: * Max Instances:

Scale Up Policy

* Duration(in sec):

Hide

Counter	Operator	Threshold	Add
<input type="text" value="Linux User CPU - percentage"/>	<input type="text" value="greater-than"/>	<input type="text"/>	<input type="button" value="Add"/>
Linux User CPU - percentage	greater-than	80	<input type="button" value="✕"/>
Linux User RAM - percentage	greater-than	80	<input type="button" value="✕"/>

Scale Down Policy

* Duration(in sec):

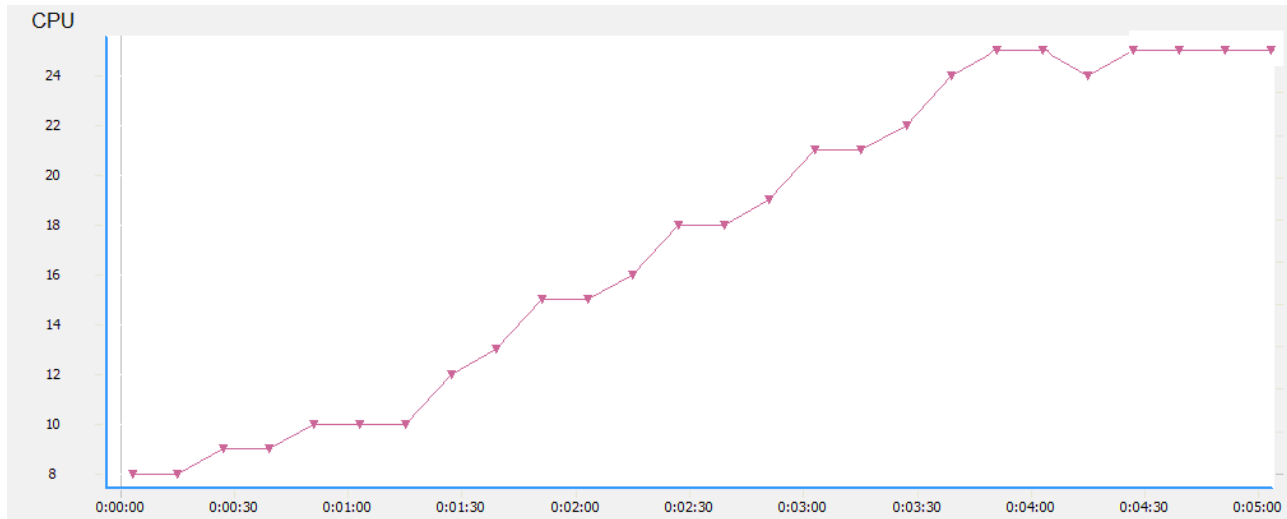
Hide

Counter	Operator	Threshold	Add
<input type="text" value="Linux User CPU - percentage"/>	<input type="text" value="greater-than"/>	<input type="text"/>	<input type="button" value="Add"/>
Linux User CPU - percentage	less-than	20	<input type="button" value="✕"/>
Linux User RAM - percentage	less-than	20	<input type="button" value="✕"/>

Нагрузочное тестирование

- В первом тесте проверим эффективность (равномерность распределения нагрузки) балансировки. Для него зададим в настройках тестирования количество пользователей, увеличивается от 0 до 40 с добавлением по 2 пользователя каждые 10 секунд.
- Во втором - тестирование работы средства автомасштабирования CloudStack. Для этого сделаем два сценария теста, которые будут выполняться последовательно. В первом сценарии количество пользователей будет увеличиваться от 0 до 160 с добавлением 2 пользователей каждые 6 секунд. Во втором - те же пользователи, созданные в первом сценарии будут завершать сессию и их количество пользователей будет уменьшаться от 160 до 0 по 2 пользователя каждые 6 секунд.

Проверка эффективности правил

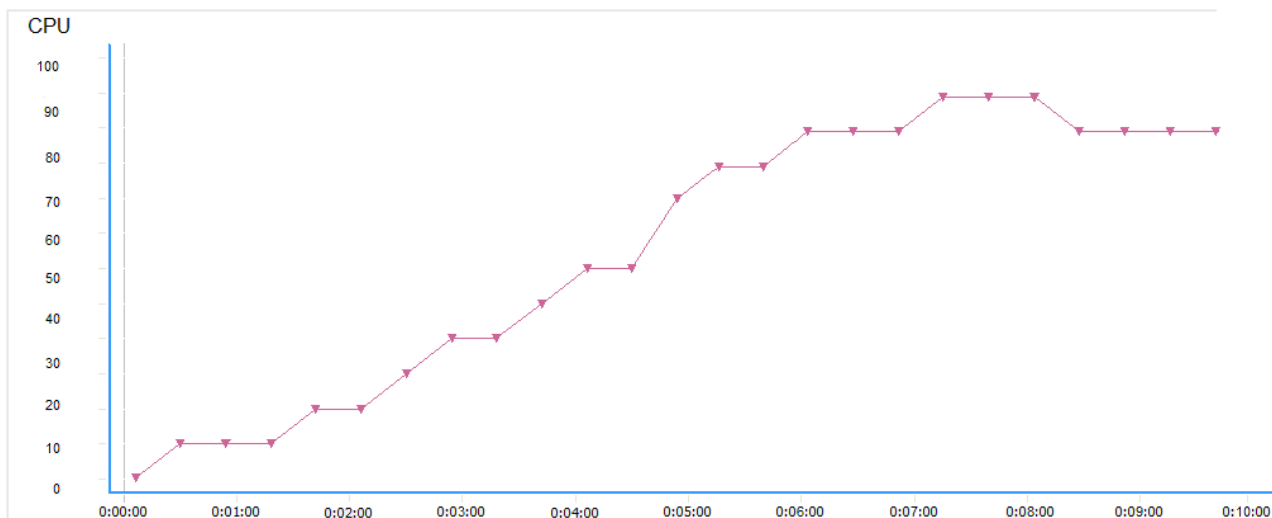


Нагрузка на ЦП первой VM

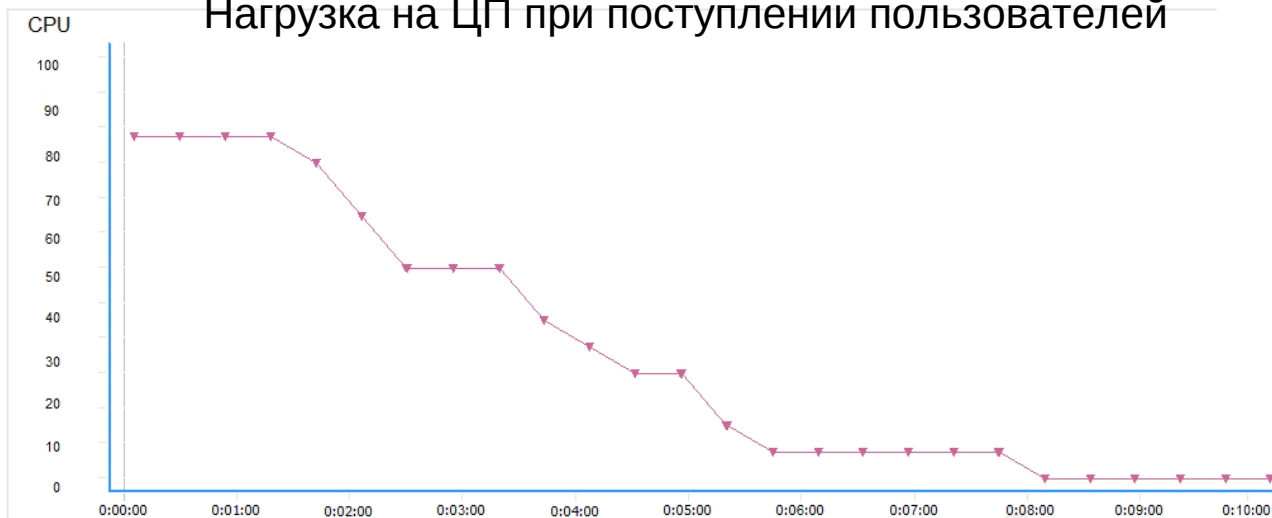


Нагрузка на ЦП второй VM

Проверка автомасштабирования



Нагрузка на ЦП при поступлении пользователей



Нагрузка на ЦП при завершении сессий пользователей

Результаты данных исследований могут быть использованы:

- для выбора методов, алгоритмов и систем балансировки нагрузки при создании "облачных" сервисов,
- в учебных дисциплинах по "облачным вычислениям",
- в дальнейшей работе кафедры СП по внедрению частного «облачного» сервиса дистанционного обучения.

Спасибо за внимание!