

Влияние развития облачных вычислений на разработку программного обеспечения



Александр Климов, @alex_klimov
Игорь Сухарев, @igorsukharev

Интернет и облачные вычисления

“Because the Internet is primarily a software artifact, there seems to be no end to its possibilities”

Vint Cerf

Шум вокруг облачных вычислений



Технологические тренды, формирующие экосистему облачных платформ

- Широкополосный интернет
- Распределенные вычисления
- Виртуализация
- Прогресс hardware-технологий и центров обработки данных
- Мобильные технологии

Почему облачные вычисления?

- Скорость разработки
- Снижение затрат на проектирование и поддержку собственной технологической инфраструктуры для разработки ПО
- Географически распределенные системы

Городские легенды

- Облачные вычисления способны решить все проблемы
- Облачные вычисления снижают ТСО
- Облачные вычисления создают нехватку рабочих мест
- Облачные вычисления и виртуализация - это одно и то же
- Облачные вычисления делают данные “прозрачно глобализированными”

Эволюция разработки ПО в облаке

- Инструментарий (development environment)
- Платформа
- Бизнес/Доставка

Облако: инструментарий/спектральный анализ

- Системы управления версиями (github, Bitbucket, bazaar, SourceForge, Google code, etc.)
- Управление проектами (BaseCamp, Trello, asana, etc.)
- IDE (Cloud9, Exo Cloud, etc.)
- Интеграция различных провайдеров облачных инструментов разработки в гетерогенные гибридные облачные системы/системы для организации жизненного цикла ПО (например, Cloud9+github)
- Мультиплатформенность и мобильность рабочего места (миграция тяжелых систем и клиентов для разработки в облако и мобильный доступ к ним)

Облако: инструментарий/субкультура

The screenshot shows the GitHub interface for the repository 'norinori2222 / boyfriend_require'. The repository has 372 stars and 147 forks. The 'Code' tab is selected, showing the 'master' branch. The file 'README-en.md' is open, showing a commit by 'modocache' from 17 days ago. The README content includes a 'Background' section and a 'Preferences' section with five numbered requirements.

github [Signup and Pricing](#) [Explore GitHub](#) [Features](#) [Blog](#) [Sign in](#)

[norinori2222 / boyfriend_require](#) [★ Star 372](#) [Fork 147](#)

[Code](#) [Network](#) [Pull Requests 17](#) [Issues 44](#) [Wiki](#) [Graphs](#)

[branch: master](#) [Files](#) [Commits](#) [Branches 1](#) [Tags](#) [Downloads](#)

[boyfriend_require / README-en.md](#)

modocache 17 days ago Add English README.

1 contributor

file | 135 lines (122 sloc) | 5.756 kb [Edit](#) [Raw](#) [Blame](#) [History](#)

Background

After having spent one too many years not interacting with the opposite sex, I began to feel like I would never find a partner, and so it is with a sense of urgency that I've decided to invite people to contact me here.

Preferences

1. No smokers. (Required)
2. No one whose personality changes when they drink. (Required)
3. Someone who has at least 5 friends from elementary school, high school, or university that they still keep in contact with.
4. Someone who has professional relationships, which have lasted at least one year, with over 20 people (this includes relationships on Facebook).
5. Someone who can speak with his neighbors (regardless of their age or gender) for at least 30 minutes.

Облако: платформа для разработчика/ основные характеристики

- Вычисления (compute)
- Сеть (networking)
- Системы хранения данных (storage)

Облако: платформа для разработчика/ качественные характеристики

- Отказоустойчивость (designed for failure)
- Масштабируемость (scalability)
- Эластичность

Облако: платформа для разработчика/ архитектура

- Миграция legacy систем в облако
- Профилирование архитектуры приложения под облако
- Характерные, доминирующие дизайн паттерны: компонентизация, separation of concerns, loosely-connected компоненты
- Декомпозиция облачного приложения в терминах сервисов провайдера облачных услугах (compute, service, load-balancing, dns, db и тд)
- Облачные API - “клей” между различными компонентами архитектуры
- Кластеризация архитектуры по зонам доступности провайдера облачных услуг

Облако: платформа для разработчика/ революция или эволюция?

- Растущее доминирование нового вида разработчиков - “Born To Be Wild On The Web”
- Фокус на agile методологиях разработки ПО
- Тенденции упрощения среды (runtime, настройка окружения разработки и тд) компенсируются возрастающей сложностью распределенной архитектуры облачного приложения
- Фундаментальные архитектурные принципы - SOA, REST, MVC, Multi-tenancy

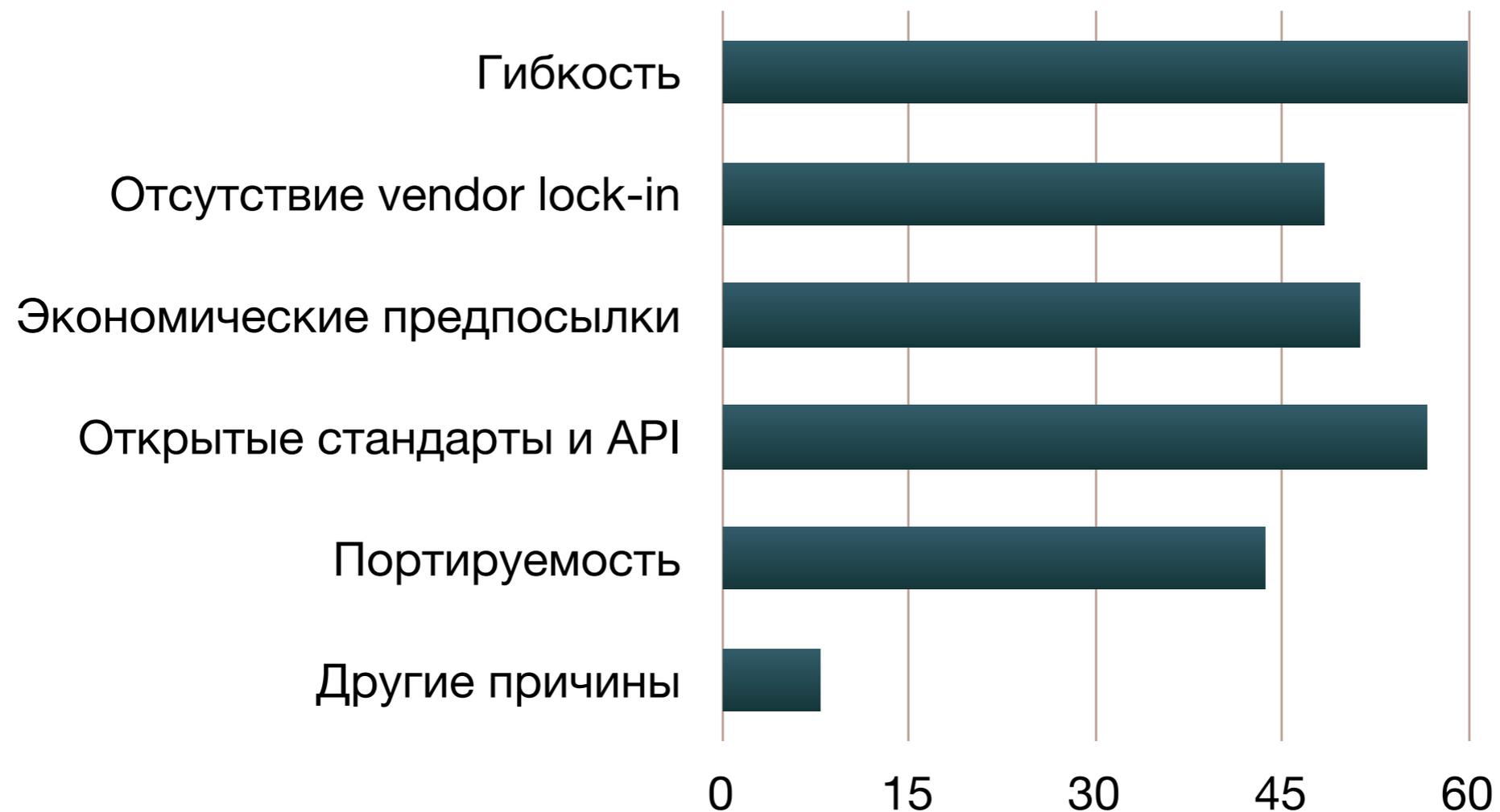
Облачные архитектуры - эволюционная модель развития распределенных систем в Интернете/Интранете, базирующаяся на известных принципах проектирования и дизайна ИТ систем

Облако: платформа для разработчика/ архитектура/паттерны

- Охуморон - design for failure для отказоустойчивости
- Кластеризация с использованием зон доступности (availability zones)
- Автоматизация цикла разработки и деплоя
- Кэширование везде, где это возможно
- Безопасность - интегральная часть проектирования каждой части архитектуры
- Глобальная распределенность
- Перманентная оптимизация финансовых затрат на облако

Облако: платформа для разработчика/ открытые облака/мотивация

■ Причины для миграции на открытые облака



(данные Zenoss, "The State Of Open Source Cloud" , 2012)

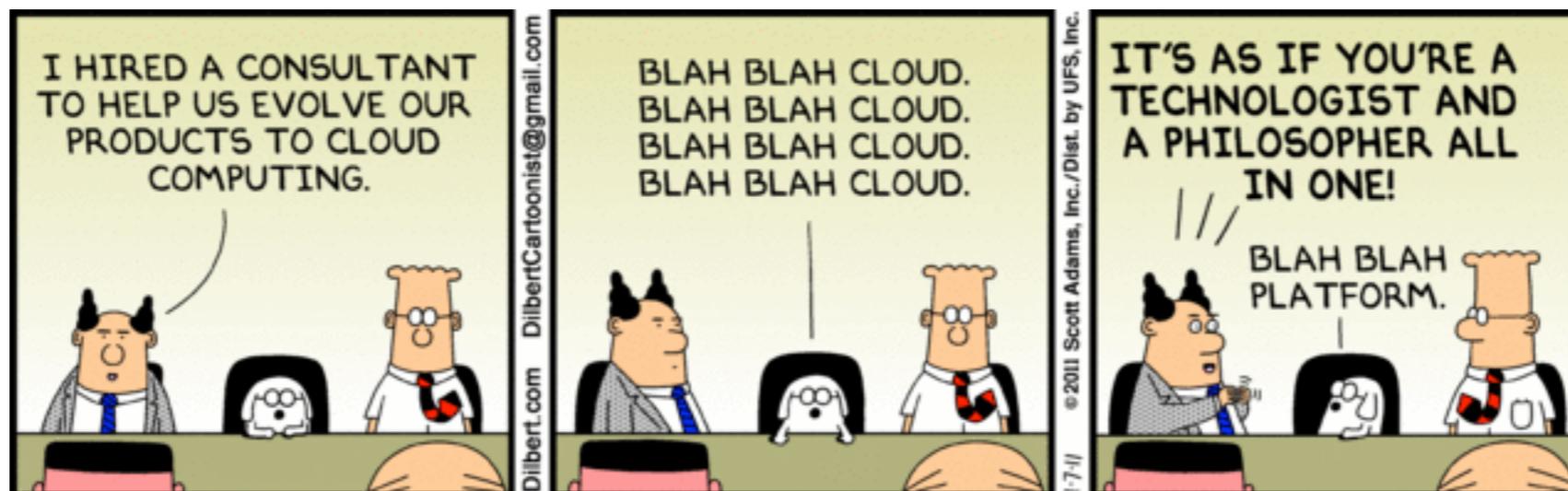
Облако: бизнес-платформа

- Магазины приложений (Apple App Store, Google Play, Windows Store, Amazon AppStore etc.)
- SaaS (facebook, force.com, Google Apps Marketplace, etc.)
- CDN (AWS CloudFront, Akamai SOLA, etc.)

Варианты использования облачных вычислений

“No size fits all”

Unknown



Облако: IaaS/атттрибуты

- Вычисления (different capacity, load balancing)
- Хранилище
- БД
- Сеть (VPN, dedicated con with premise)

Облако: IaaS/анализ

Преимущества:

- Полный контроль
- Близка к традиционным разработке и деплою
- Минимальный vendor lock-in*

Недостатки:

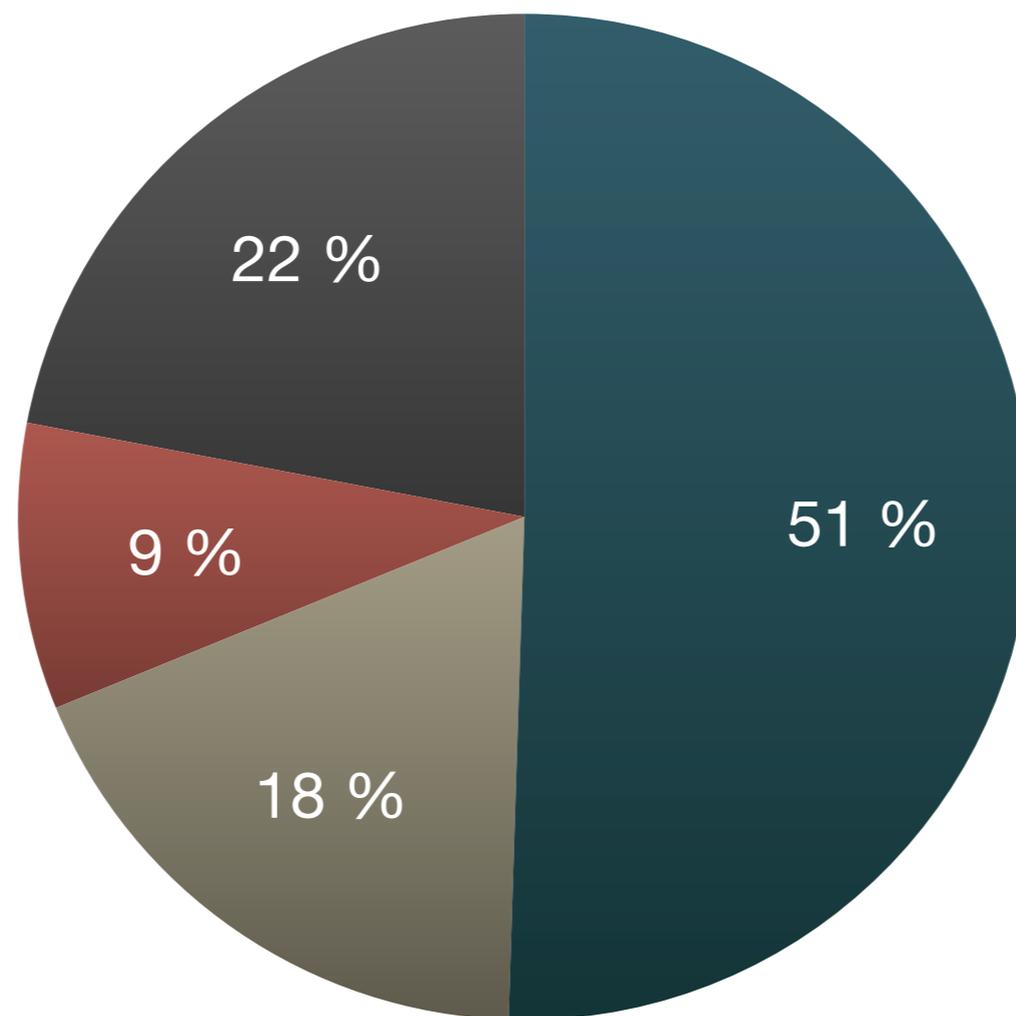
- Минимум функциональности “из коробки”
- Временные затраты и администрирование

Облако: IaaS/платформы

- Публичные (AWS/EC2, Rackspace Cloud, Google Compute Engine, Windows Azure Virtual Machines, Joyent, SoftLayer, IBM SCP etc.)
- Частные (openstack, nebula, Eucalyptus)

Облако: платформа для разработчика/ открытые облака

● OpenStack ● CloudStack ● Eucalyptus ● Other



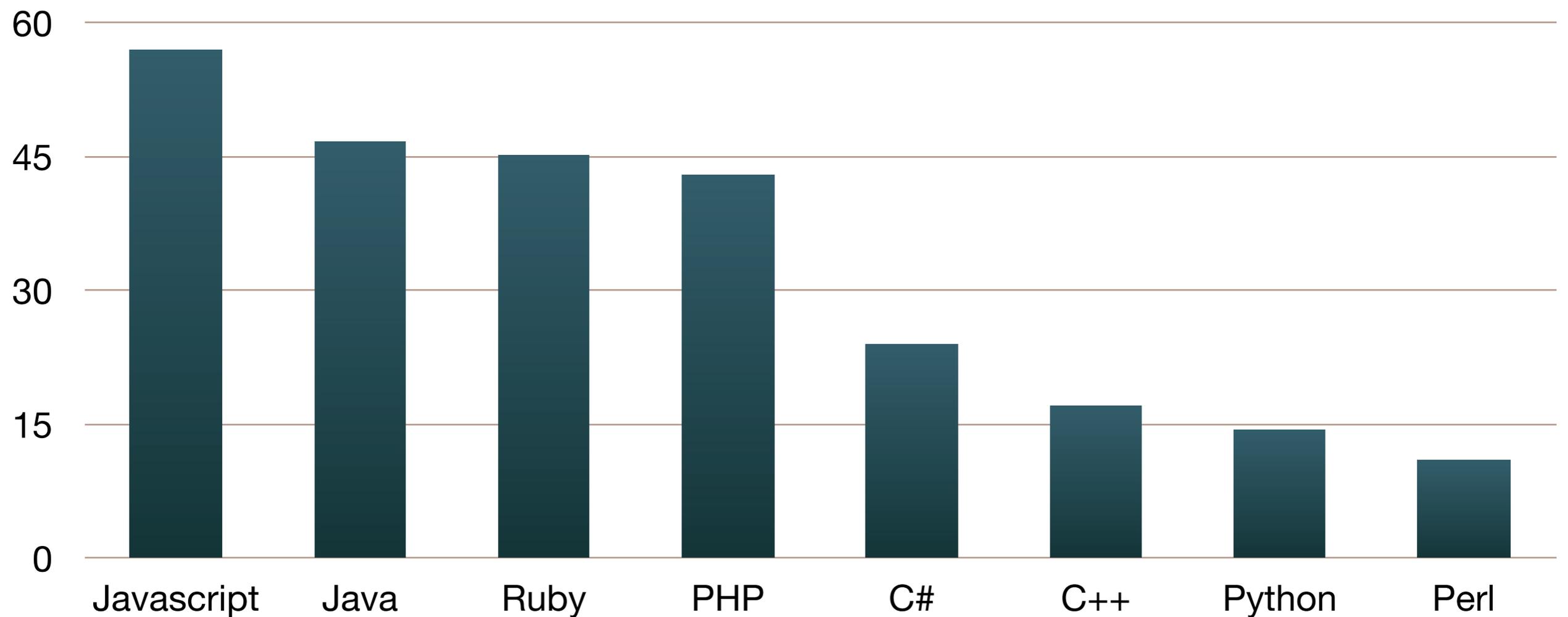
(данные Zenoss, "The State Of Open Source Cloud" , 2012)

Облако: PaaS/атттрибуты

- Языки программирования
- БД
- Веб сервер/сервер приложений

Облако: PaaS/языки программирования

■ Языки программирования, % (данные Engine Yard, Май 2012)



Облако: PaaS/анализ

Преимущества:

- Горизонтальная масштабируемость почти из коробки
- Отказоустойчивость для экземпляров приложений
- Экономия времени разработки

Недостатки:

- Меньшая степень контроля
- Наличие vendor lock-in

Облако: PaaS/платформы

- Публичные (AWS/Elastic Beanstalk, heroku, Google App Engine, Windows Azure Compute, Engine Yard, AppFrog, CloudBees, Jelastic, etc.)
- Частные (CloudFoundry, RedHat OpenShift, etc.)

Облако: SaaS/атттрибуты

- Максимальная абстракция от программного стека
- Multi-tenancy важный параметр архитектуры
- Высокая скорость выпуска новых версий ПО, которая происходит прозрачно для пользователя
- Композиция сервисов - легкость и удобство интеграции с другими сервисами (например, для разработки так называемых mashups)

Облако: SaaS/сценарии

- Распространение ПО
- Обработка платежей
- HR системы
- Социально-ориентированные системы
- Collaboration системы и инструменты project management
- ERP
- CRM
- Игры

Облако: SaaS/анализ

Преимущества:

- Скорость появления новых функциональных возможностей и прозрачность (для пользователя) их введения
- Приём платежей/Защита от мошеннических операций
- Защита от пиковых нагрузок (“the slashdot effect” или “black Friday effect”)

Недостатки:

- Меньшая степень контроля
- Зависимость от политики и правил поставщика SaaS
- Дата выпуска ПО в магазине может варьироваться

Облако: SaaS/платформы

- Force.com
- Facebook
- Google Apps

Облако: выводы

“There is no silver bullet”

Frederick Brooks, "The Mythical Man-Month"



Облачные вычисления, нужно помнить

- SLA/Поддержка
- Географическое размещение данных (e.g. USA Patriot Act)
- Vendor lock-in
- Downtime (unexpected outages)
- Безопасность

Paradigm shift?

