

Облачные вычисления

Горских А.Г.

ММ-426

Оглавление

- ▶ Определение облачных вычислений
- ▶ Отличия облака от GRID и кластеров
- ▶ Достоинства
- ▶ Недостатки
- ▶ Типы платформ облачных вычислений
- ▶ Обзор основных платформ
- ▶ Пример разработки под Cloud
- ▶ Будущее облачных платформ

Что такое облачные вычисления?

- ▶ Облачные вычисления — вычисления проводимые в готовой инфраструктуре к которой есть доступ через сеть.



Отличия облаков от GRID и кластеров

Отличия облаков от GRID и кластеров (1)

Системы/ Характеристики	Кластеры	Grids	Облака
Состав	Обычные компьютеры	Сервера, кластеры	Обычные компьютеры, серверы и NAS
Размер	сотни	тысячи	сотни тысяч
ОС узла	Стандартные ОС	Стандартные ОС	Стандартные ОС, работающие поверх гипервизора
Количество владельцев	один	много	один
Тип связи узлов	Высокоскоростная с малой задержкой и высокой пропускной способностью связь	Преимущественно через интернет с высокой задержкой и низкой пропускной способностью	Высокоскоростная с малой задержкой и высокой пропускной способностью связь
Безопасность	Традиционный Логин/Пароль Средний уровень приватности(зависит от привилегий пользователя)	Аутентификация с помощью открытого/закрытого ключей и с привязкой к аккаунту пользователя	Каждый пользователь работает в своей виртуальной машине. Высокий уровень приватности/безопасности.

Отличия облаков от GRID и кластеров (2)

Системы/ Характеристики	Кластеры	Grids	Облака
Управление ресурсами	Централизованное	Децентрализованное	Централизованное/ Децентрализованное
Стандарты/ Взаимодействие с другими системами	Основан на VIA(Virtual Interface Architecture)	Какой-то из стандартов Open Grid Forum	Web-сервисы (SOAP и REST)
Единый образ системы для каждого узла	Да	Нет	Опционально
Объём памяти	Стабильная и гарантированная	Варьируется, но высокая	Предоставляется по запросу
Способность к самовосстановлению	Ограничена (Перезапуск неудавшихся задач)	Ограничена (Перезапуск неудавшихся задач)	Сильная поддержка реплицирования данных VMs могут легко мигрировать между узлами

Достоинства

- ▶ Распределённость
- ▶ Удалённость
- ▶ Масштабируемость
- ▶ Отказоустойчивость
- ▶ Надёжность хранения данных
- ▶ Стандартизированный интерфейс для работы в “облаке”
- ▶ Оплата только потребляемых ресурсов
- ▶ Высокая степень утилизации простаивающих ресурсов



Недостатки

- ▶ Сложность переноса существующих приложений
- ▶ Необходимость в постоянном подключении к Сети
- ▶ Конфиденциальность хранимой информации
- ▶ Надёжность хранения данных



Типы платформ облачных вычислений

Типы платформ облачных вычислений



IaaS

- ▶ IaaS (Infrastructure as a Service) – инфраструктура как сервис. Вы арендуете сервера/кластеры и оплачиваете использованные ресурсы. Сами выбираете ОС, необходимые сервисы и пишете конечное приложение. Целевая аудитория: разработчики ПО. Типичные представители: Amazon EC2, GoGrid, ElasticHosts и др.

PaaS

- ▶ PaaS (Platform as a Service) – платформа как сервис (IaaS + специальная программная платформа). За вас уже поставили ОС, поставили определённый набор ПО и всё настроили, предоставили удобный набор API для работы. Оплачиваются только израсходованные ресурсы. Целевая аудитория: разработчики ПО. Типичные примеры: Google App Engine, Windows Azure, AptanaCloud.

SaaS

- ▶ SaaS (software as a Service) – приложение как сервис. Готовое приложение для конечных пользователей. Вариантов оплаты море – за ресурсы, просмотр рекламы, абонплата и так далее. Типичные представители: современные почтовые службы, файлообменники, многие CRM/ERP–системы. Отличительной чертой является доступ к приложению посредством веб–браузера, но не всегда.

Другие виды *aaS

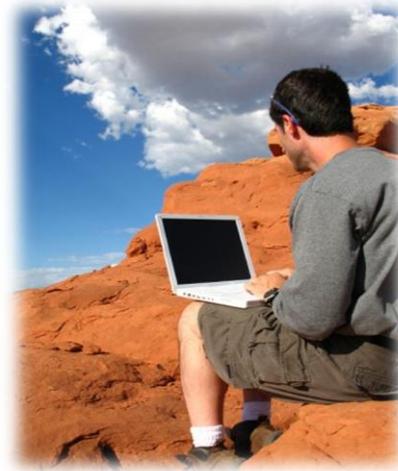
DaaS

- ▶ DaaS (Desktop as a Service) предлагает каждому пользователю стандартизированное виртуальное рабочее место, с возможностью настройки и установки других программ. Доступ осуществляется по сети посредством тонкого клиента.



СaaS

- ▶ СaaS (Communications as a Service) – сочетания программно–аппаратных средств для организации всех видов общения (голос, почта, IM ...) между сотрудниками одного предприятия за счёт сторонних решений.



Обзор основных платформ

Amazon EC2



- ▶ IaaS
- ▶ Позволяет выбирать место размещения инстанса
- ▶ Amazon Elastic Block Store – шардинг, репликация, независимость от инстанса
- ▶ Elastic IP Addresses – привязываются не к инстансу а к аккаунту
- ▶ Amazon Virtual Private Cloud – мост между инфраструктурой предприятия и АЕС2, посредством VPN
- ▶ High Performance Computing (HPC) Clusters – даже с GPU

Amazon EC2



- ▶ Доступные операционные системы
 - Red Hat Enterprise Linux
 - Windows Server 2003/2008
 - Oracle Enterprise Linux
 - OpenSolaris
 - Amazon Linux AMI
 - Ubuntu Linux
 - Fedora
 - Gentoo Linux
 - Debian
 - SUSE Linux Enterprise

Amazon EC2



▶ Доступное ПО

- Веб-хостинг
 - Apache HTTP
 - IIS/Asp.Net
 - IBM Lotus Web Content Management
 - IBM WebSphere Portal Server
- Серверы приложений
 - IBM WebSphere Application Server
 - Java Application Server
 - Oracle WebLogic Server
- Базы данных
 - IBM DB2
 - IBM Informix Dynamic Server
 - Microsoft SQL Server Standard 2005/2008
 - MySQL Enterprise
 - Oracle Database 11g

Google App Engine



- ▶ PaaS
- ▶ Разработка на Java и Python
- ▶ Постоянное хранилище с запросами, сортировкой и транзакциями
- ▶ Автоматическое масштабирование и регулировку нагрузки
- ▶ API для аутентификации пользователей и отправку электронной почты с помощью аккаунтов Google
- ▶ Полнофункциональная локальная среда разработки, имитирующая Google App Engine на вашем компьютере

Microsoft Azure



- ▶ PaaS
- ▶ Windows Azure: предоставляет основанную на Windows среду для выполнения приложений и хранения данных на серверах в дата центрах Microsoft;
- ▶ Windows .NET Services: предоставляют сервисы распределенной инфраструктуры для облачных и локальных приложений.
- ▶ Microsoft SQL Services: предоставляют сервисы для работы с данными, основанные на SQL Server.
- ▶ Live Services: Через Live Framework предоставляет доступ к данным из приложений на Microsoft Live. Live Framework также позволяет синхронизировать эти данные между десктопами и устройствами, искать и загружать приложения и другое.

Сравнение облачных платформ

Платформы/ Характеристики	Amazon Elastic Compute Cloud (EC2)	Google App Engine	Microsoft Azure
Тип	IaaS	PaaS	PaaS
Тип разрабатываемых сервисов	Вычислительные сервисы, сервисы хранения	Web-приложения	Как Web-приложения, так и не Web-приложения
Виртуализация	Уровня ОС , с запущенным гипервизором Xen	Контейнер приложений	Уровня ОС
Интерфейс доступа пользователя	Утилиты командной строки Amazon EC2	Web-консоль администрирования	Портал Microsoft Windows Azure
Web APIs	Да	Да	Да
Среда разработки	Конфигурируемый, основанный на linux Amazon Machine Image (AMI)	Python Java	Microsoft .NET

Пример разработки с использованием Google App Engine

Пишем гостевую книгу

- ▶ Используется среда выполнения Python
- ▶ Формы в webapp
- ▶ Хранилище данных
- ▶ Служба пользователей Google.



app.yaml

application: helloworld

version: 1

runtime: python

api_version: 1

handlers:

- url: /*

script: helloworld.py

helloworld.py (1)

```
import cgi
```

```
from google.appengine.api import users
```

```
from google.appengine.ext import webapp
```

```
from google.appengine.ext.webapp.util import run_wsgi_app
```

```
from google.appengine.ext import db
```

```
class Greeting(db.Model):
```

```
    author = db.UserProperty()
```

```
    content = db.StringProperty(multiline=True)
```

```
    date = db.DateTimeProperty(auto_now_add=True)
```

helloworld.py (2)

```
class MainPage(webapp.RequestHandler):
    def get(self):
        self.response.out.write('<html><body>')

        greetings = db.GqlQuery("SELECT * FROM Greeting ORDER BY date DESC LIMIT 10")

        for greeting in greetings:
            if greeting.author:
                self.response.out.write('<b>%s</b> wrote:' % greeting.author.nickname())
            else:
                self.response.out.write('An anonymous person wrote:')
            self.response.out.write('<blockquote>%s</blockquote>' %
                                    cgi.escape(greeting.content))

        # Write the submission form and the footer of the page
        self.response.out.write("""
            <form action="/sign" method="post">
                <div><textarea name="content" rows="3" cols="60"></textarea></div>
                <div><input type="submit" value="Sign Guestbook"></div>
            </form>
            </body>
            </html>""")
```

helloworld.py (3)

```
class Guestbook(webapp.RequestHandler):
    def post(self):
        greeting = Greeting()

        if users.get_current_user():
            greeting.author = users.get_current_user()

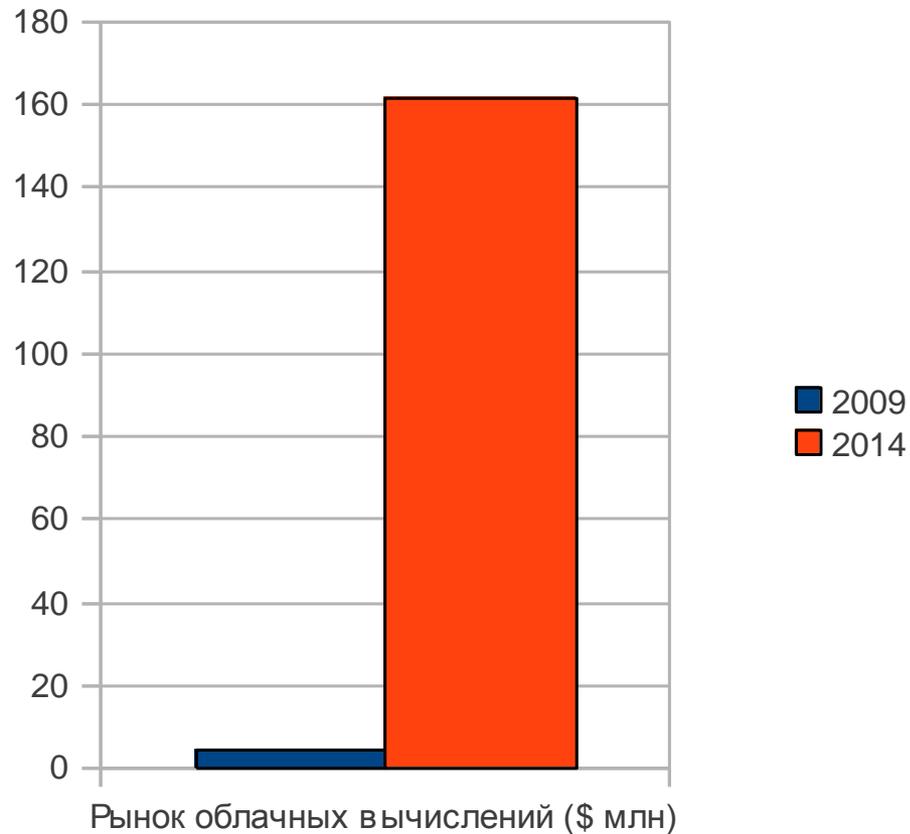
        greeting.content = self.request.get('content')
        greeting.put()
        self.redirect('/')

application = webapp.WSGIApplication(
    [('/', MainPage),
     ('/sign', Guestbook)],
    debug=True)

def main():
    run_wsgi_app(application)

if __name__ == "__main__":
    main()
```

Прогноз динамики рынка облачных вычислений в России



Будущее облачных вычислений

- ▶ Облачные вычисления станут неотъемлемой частью нашей жизни, как и вода, электричество, телефония, газ.
- ▶ Облака объединятся в одно облако
- ▶ Вычисления будут доступны по первому требованию