

Облачные системы

И прочие проекты

Борисенко Олег, ИСП РАН

СПИСОК ОСНОВНЫХ ПРОЕКТОВ

Native XML Sedna

REST API Gateway & throttling control

Предсказание сайтов связывания транскрипционных факторов

Приватное облако с виртуальными машинами по запросу под управлением Xen

Система документации исторических и исчезающих естественных языков

Openstack Sahara

Openstack Spark

Основные возможности облачных сред

- Выделение виртуальных машин с заданными характеристиками по запросу
- Построение сложных сетевых инфраструктур
- Виртуализация блочного хранения
- Объектные хранилища
- Встроенные системы оркестрации ресурсов
- Все это можно использовать при помощи простого REST API

Основные особенности

В чем преимущества для пользователей:

- Не нужно иметь собственный парк железа; плата только за использованные ресурсы
- Возможность адаптироваться под нагрузку
- Виртуализация как вычислительной части, так сетей и хранения
- Низкий порог необходимых знаний для использования
- Можно выбирать ресурсы под задачу, а не использовать “что есть”

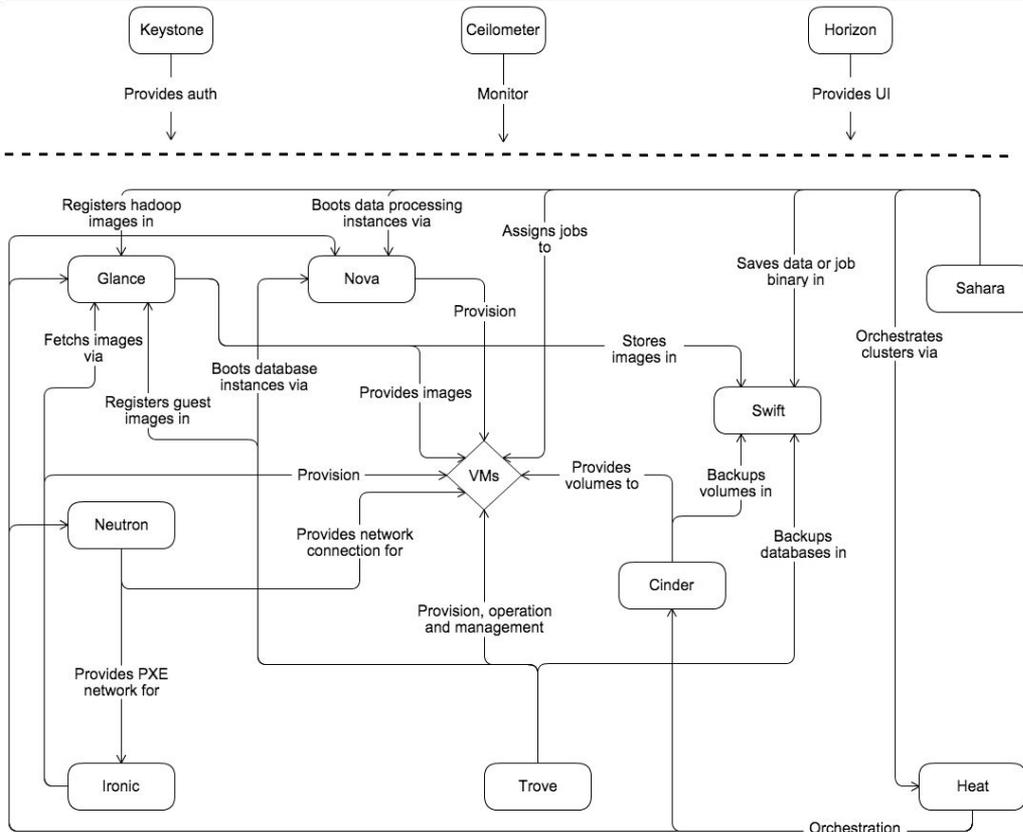
В чем преимущества для провайдеров облачных услуг:

- Дешевле в плане электроэнергии
- Можно продавать более сложные вещи (PaaS) за больше денег, чем “просто хостинг”

Самые известные из коммерческих — Amazon EC2 и Microsoft Azure

Самые известных из открытых — Openstack и Eucalyptus

Openstack, КОМПОНЕНТЫ



Openstack состоит из т.н. “проектов”.
В реальности их больше, чем на этой
схеме, основные из них:

- Keystone
- Nova
- Horizon
- Neutron
- Cinder
- Glance
- Swift
- Heat
- Sahara
- Ceilometer

Минимальный необходимый набор —
Keystone, Glance и Nova.

Keystone

Центральный сервис аутентификации и авторизации.

- Обслуживает и пользователей, и сами сервисы
- Работает на основе токенов
- В качестве бэкенда может использовать другие сервисы аутентификации
- Это обычное WSGI-приложение

Nova

- Сервис предоставляет виртуальные машины.
- Умеет работать с разными гипервизорами.
- Умеет обслуживать Plain-сети (без Neutron).
- Отвечает за выделение физических ресурсов.

Horizon

Веб-интерфейс для всего Openstack.

- Обычное веб-приложение на Django.
- Позиционируется как “фреймворк для построения веб-интерфейсов к Openstack”
- Модульный.
- Использует нативные библиотеки для Openstack на Python для коммуникации с Openstack.

Neutron

Сервис виртуальных сетей.

- Предоставляет изоляцию служебных и публичных сетей внутри Openstack (при помощи Vlan, VXLAN, GRE)
- Следит за Security группами (по сути - фаерволл)
- Позволяет программно задавать сетевую топологию виртуальных кластеров.
- Предоставляет публичные IP-адреса виртуальным машинам (реализовано как floating IP)

Cinder

Сервис предоставления блочных устройств виртуальным машинам.

- Умеет делать снимки образов.
- Отдает все по iSCSI (или при помощи драйвера, но тоже по iSCSI).
- Поддерживает множественные бэкенды для хранения.

Glance

Сервис предоставления базовых образов ОС.

- Хранит и отдает образы для гипервизоров в Nova.
- В случае использования Docker в Nova по сути ведет себя как локальный Docker-registry.

Swift

Объектное хранилище Openstack.

- Позволяет хранить метаданные для файлов (хранятся как xattr).
- Предназначен для использования на обычных машинах.
- Прозрачно масштабируется.
- Поддерживает хранение реплик.
- Может использоваться как замена HDFS в Sahara

Heat

Сервис оркестрации в Openstack.

- Принимает в себя шаблоны кластеров виртуальных машин (AWS и HOT).
- Предоставляет слой совместимости с метаданными Amazon (неполный).

Sahara

Сервис обработки Big Data в PaaS и IaaS ключе.

- Позволяет создавать кластеры с нужными инструментами обработки данных (Hadoop, Spark, Hive, etc)
- Позволяет запускать на исполнение уже написанные задания - не нужно думать про инфраструктуру (закинул-посчитал).

*мы реализовали поддержку Spark-заданий и использование Swift вместо HDFS для Spark

Проблемы масштабируемости Openstack в контексте Keystone

- Обработывает слишком мало запросов в секунду.
- Вне зависимости от используемых слоев (расскажу на словах), производительность деградирует со временем.
- Не позволяет строить кластеры Openstack более чем из 120 серверов при любой скорости канала из-за постоянных сообщений между сервисами и Keystone.

Тестовая среда

Был разработан инструментарий тестирования, позволяющий:

- Разворачивать с нуля Keystone во всех возможных конфигурациях:
 - HTTP/WSGI слой - Apache2+mod_wsgi, Nginx+uwsgi, eventlet
 - RDBMS слой - MariaDB, PostgreSQL
 - Слой хранения - HDD, рейд из SSD, tmpfs (RAM)
- Запускать набор синтетических тестов для нагрузки Keystone
- Находить число N при заданном временном окне, при котором система noticeably деградирует в производительности.

Подход к решению проблемы

- Построение описанного тестового инструментария.
- Анализ результатов профилирования по трем срезам: при малой нагрузке, при средней нагрузке и при N-1 RPS.
- Время исполнения функций должно увеличиваться примерно линейно при линейном увеличении нагрузки. Так где это не так, нужно внимательно копать.

Борисенко Олег, ИСП РАН
al@somestuff.ru

Спасибо за внимание.