

**Я**ндекс

# Python в инфраструктуре поиска

**Л**еоид Васильев

Разработчик

Я.Субботник, Екатеринбург, 2 июля 2011 года

# Инфраструктура поиска

## Проблемы:

- Тысячи серверов в нескольких датацентрах
- Постоянный поток новых серверов
- Сервера ломаются и демонтируются
- Много активного сетевого оборудования
- Отключения питания и ошибки сети
- Разные ОС: Linux, FreeBSD

# Как с этим бороться?

До:



После:



# Как управляем кластером?

## Наши инструменты:

- **Skynet**: выполнение Python кода на кластере
- **Sandbox**: сборка, тестирование и выкладка программ на кластер
- И т. д. Инструментов у нас много :)

# Skynet: выполнение Python кода на кластере

# Skynet: пример проблемы

## Дедушкин метод:

```
#!/bin/sh

SCRIPT_NAME='/home/my_name/check_something.sh'

for host in serv{1..50}; do
    echo "Checking $host"
    scp "$SCRIPT_NAME" "$host:/$SCRIPT_NAME"

    ssh $host "sh $SCRIPT_NAME"
done
```

# Skynet: пример проблемы

## Современный подход:

```
#!/usr/bin/env python

from skynet import client

class CheckSomething:
    def __init__(self, config): self.config = config
    def run(self): return self.check_something()
    def check_something(self): pass # check using config

if __name__ == "__main__":
    c = client.Client()
    c.run("serv{1..50}", CheckSomething({'debug': True}))
    c.wait()
    print c.result()
```

# Skynet: пример проблемы

## Сравнение

Дедушкин метод:	Современный подход:
системные требования	системные требования
функционал	функционал
поддержка	поддержка
кроссплатформенность	кроссплатформенность

# Skynet: как это работает?

- `code = pickle.dumps(Task())`
- `modules = Zipfile.PyZipFile(sys.modules)`
- Передаем `(code, modules)` на клиента
- `import modules.zip`
- `result = pickle.loads(code.pckl).run()`
- Передаем `result` на сервер

# Skynet: копирование данных

## Используем torrent:

```
#!/usr/bin/env python

from skynet import copier

def share(files):
    c = copier.Copier()
    return c.create(files).torrent

def download(torrent, dest_dir):
    c = copier.Copier()
    c.get(torrent, dest_dir).wait()
```

# Skynet: зачем torrent?

- Хорошо ложится на распределённую архитектуру
- Высокая отказоустойчивость

**Sandbox:** сборка,  
тестирование и выкладка  
программ на кластер

# Sandbox: что это?

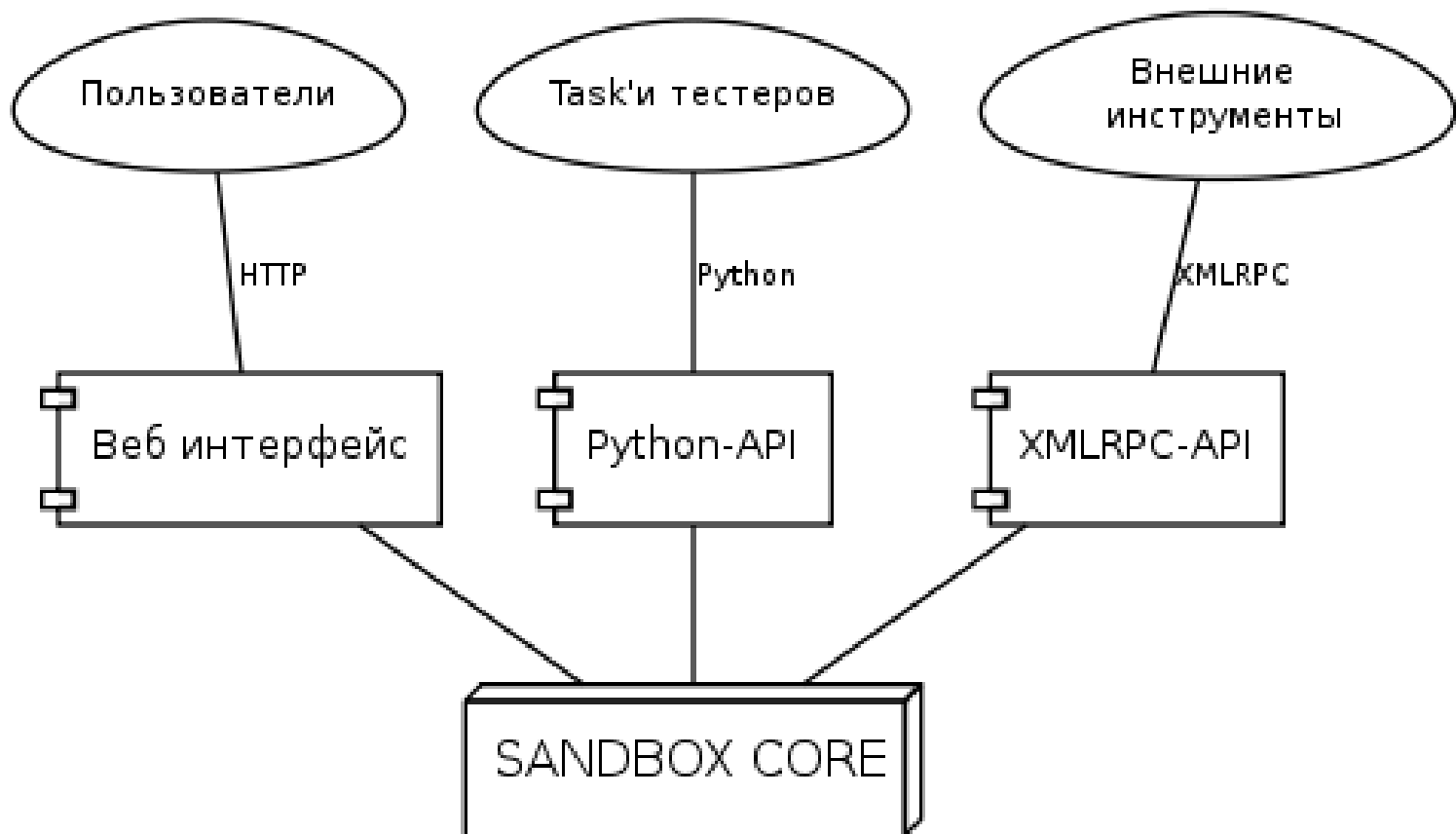
Небольшой кластер (десятки серверов) и среда выполнения, которые нужны

**Разработчикам:** для сборки и отладки программ на разных архитектурах (linux, freebsd) и моделях CPU.

**Тестировщикам:** для проведения ресурсоёмких тестов

**Администраторам:** для внедрения программ в production систему

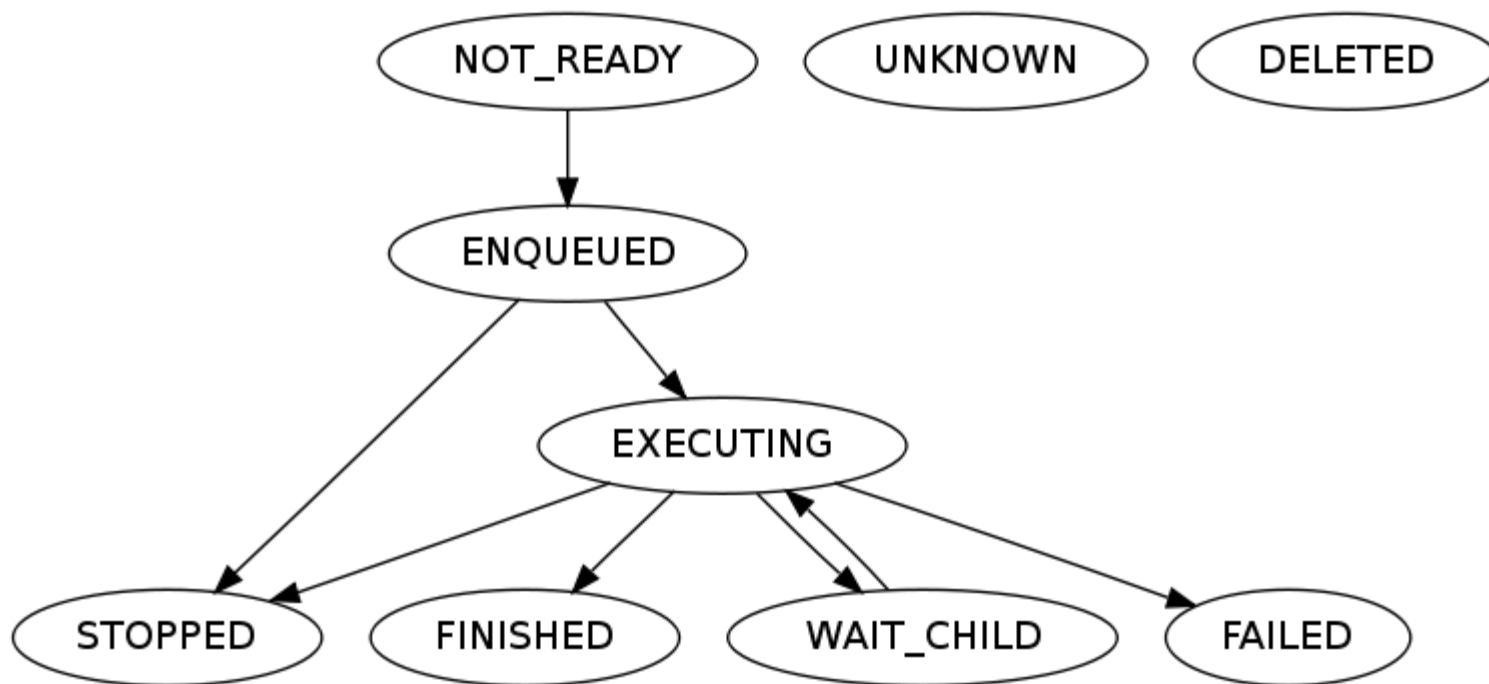
# Sandbox: Архитектура



# Sandbox:главные сущности

- **Task**: контейнер для кода и состояния, который будет выполняться на клиенте
- **Ресурс**: описывает данные, которые Task'и используют/создают в процессе работы
- **Контекст выполнения**: параметры Task'a, которые были заданы при его создании (через веб-интерфейс или xmlrpc)

# Sandbox: Состояния Task'a



# Sandbox: типовой сценарий

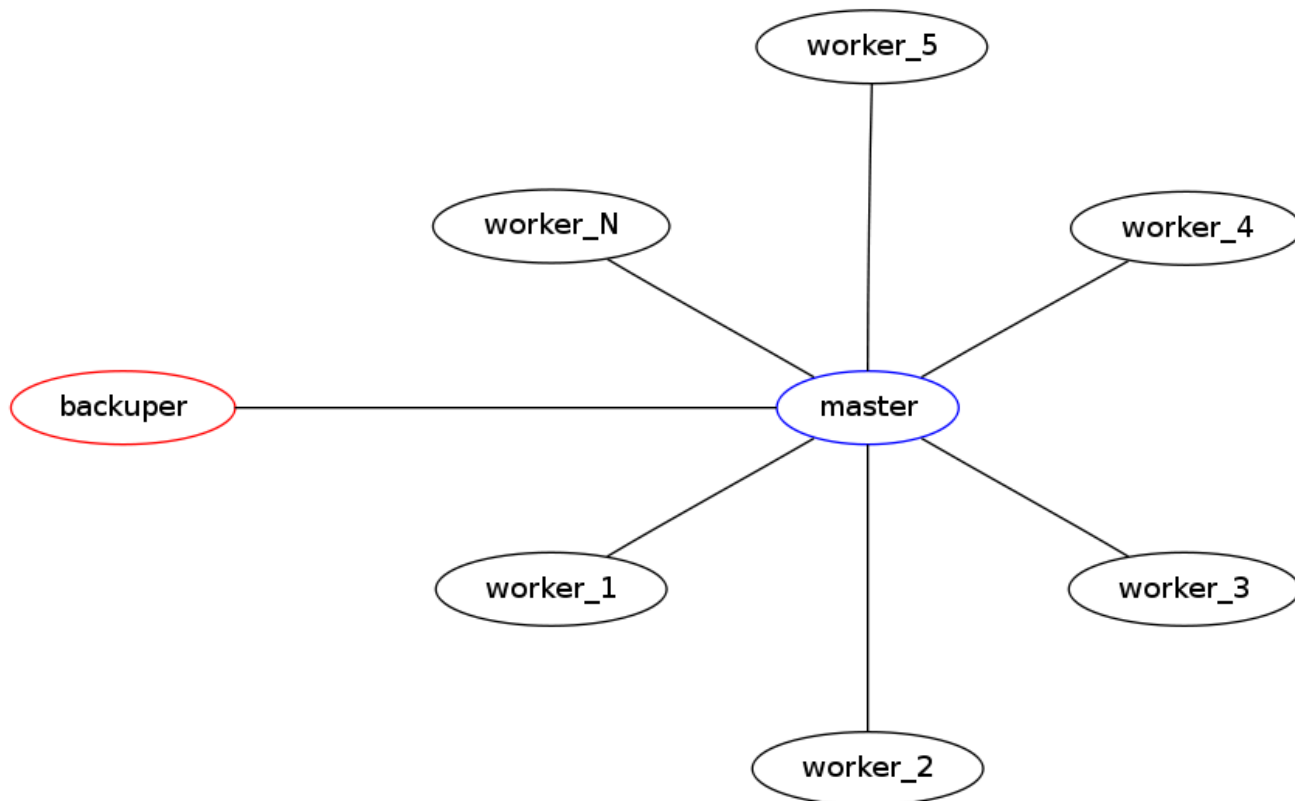
- Грузим файлы необходимые для сборки
- Собираем проект#1
- Собираем проект#2
- Тестируем проект#1 и проект#2
- Получаем результаты от проекта#1 и проекта#2
- Сравниваем результаты проекта#1 и проекта#2

# Sandbox: использование API

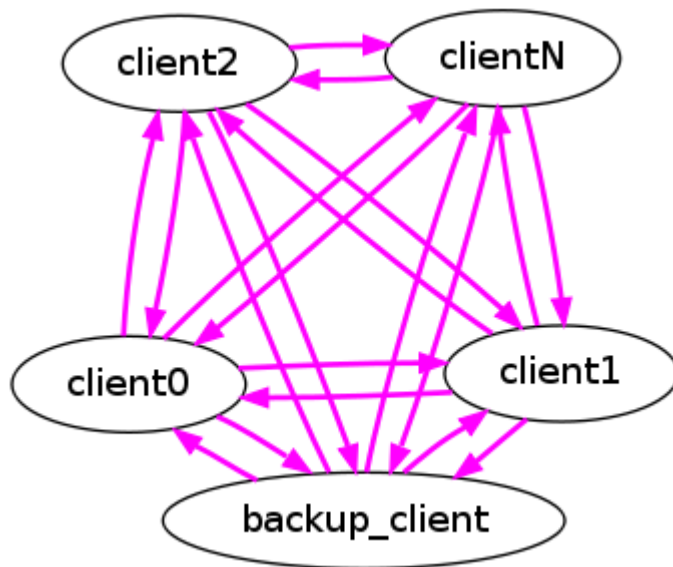
## Сборка проекта:

```
def build(branch, owner):
    con = xmlrpclib.ServerProxy('http://sandbox/xmlrpc')
    task_id = con.createTask({...})
    while True:
        status = con.getTaskStatus(task_id)
        if not is_in_progress(status):
            break
        time.sleep(1)
    if is_failed(status):
        raise Exception(...)
    return con.listTaskResources(task_id)
def main():
    print build("branches/my_project/rel-1.0.0", "chin")
```

# Sandbox: Инфраструктура



# Sandbox: клиенты Обмен данными (ресурсами)



Клиенты скачивают ресурсы через p2p сеть. Сервер используется как «треккер» ресурсов. Поддерживаемые протоколы: **http**, **scp**, **rsync**, **bittorrent**.

# Полезные практики

# Полезные практики

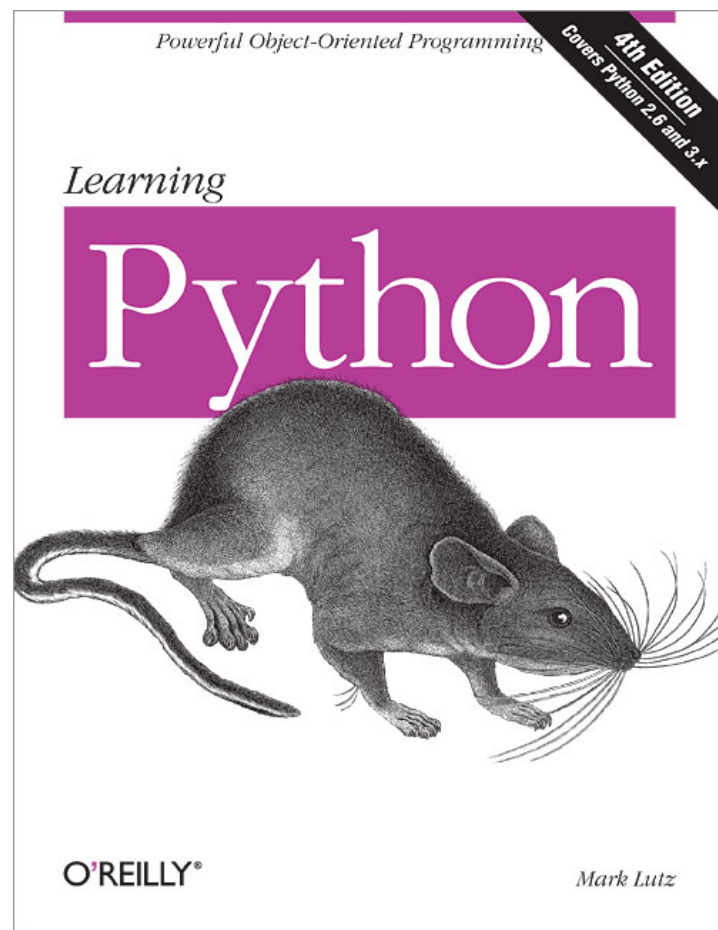
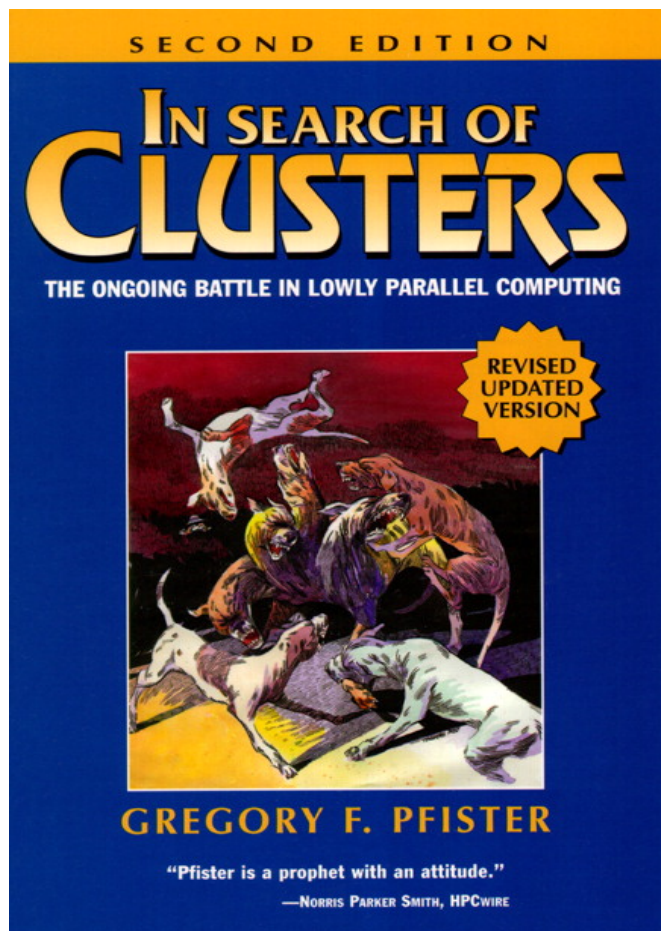
Крайне полезно иметь в проекте:

- **Хранилище файлов** необходимых для сборки, отладки, профилирования и тестирования.
- **Список релизов** с веб-интерфейсом в котором есть информация о том, что, когда и куда выкладывалось.
- Утилиту или framework для **выполнения скриптов на всем кластере.**

# Open-source аналоги

- *Execnet*: <http://codespeak.net/execnet/>
- *FileMap*: <http://mfisk.github.com/filemap/>
- *BuildBot*: <http://trac.buildbot.net/>

# Литература





**Леонид Васильев**

Разработчик

[chin@yandex-team.ru](mailto:chin@yandex-team.ru)