Создание масштабируемых приложений

на примере сервиса сканирования веб сайтов в Лаборатории Касперского

Обо мне

Лаборатория Касперского

Лаборатория контентной фильтрации

Группа развития инфраструктуры

Наша работа

- Сервис получения спама
- Генерация автоправил (Даниил Зайцев)
- Стенды сравнения продуктов
- Сервис сканирования сайтов

Общий инструментарий

- XP (extreme programming)
- DevOps

Системы контроля версий

1. Git

git

Системы контроля версий

1. Git
2. Mercurial

git

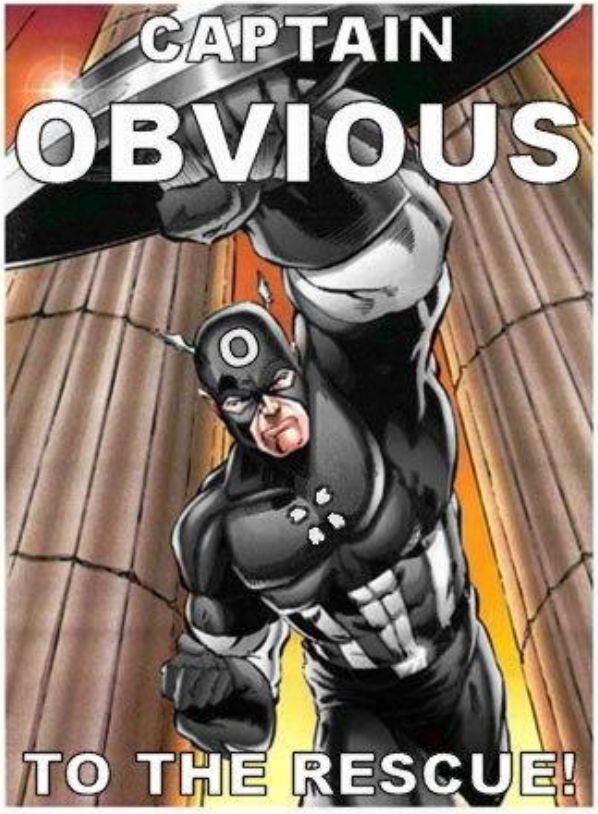
Системы контроля версий

1. Git
2. Mercurial
3. Fossil

git

Что хранить?

1. Код



Что хранить?

1. Код
2. Вспомогательные скрипты

Что хранить?

1. Код
2. Вспомогательные скрипты
3. Файлы конфигурации

Что хранить?

1. Код
2. Вспомогательные скрипты
3. Файлы конфигурации
4. Тестовые данные

Что хранить?

1. Код
2. Вспомогательные скрипты
3. Файлы конфигурации
4. Тестовые данные
5. Входные данные необходимые для работы вашего ПО

Что хранить?

1. Код
2. Вспомогательные скрипты
3. Файлы конфигурации
4. Тестовые данные
5. Входные данные необходимые для работы вашего ПО
6. Специфические зависимости (libVasya.so)

Какая польза?

```
git clone <project> <directory>
```

Поздравляю! Вы великолепны.

Модульное тестироване

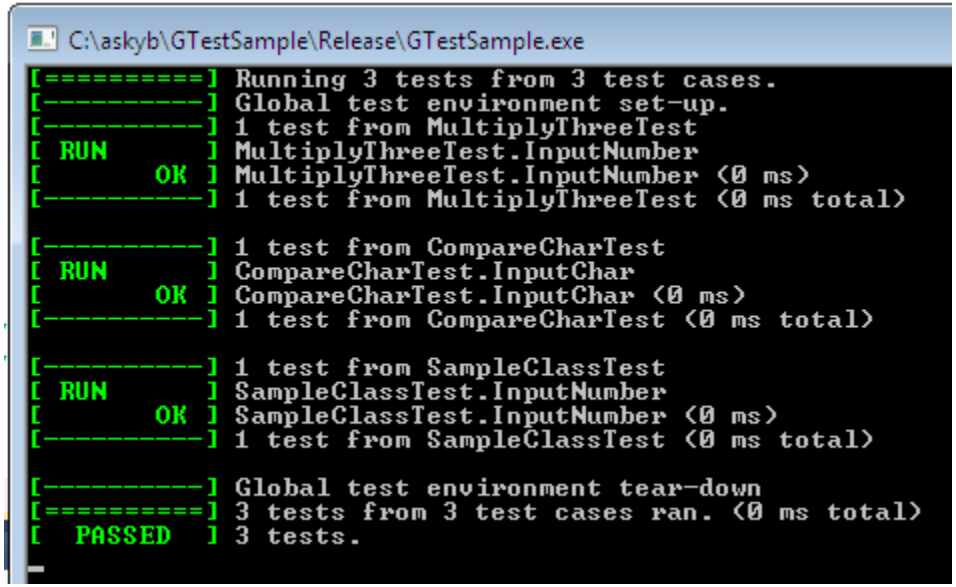
- Регрессии не миф!

Модульное тестироване

- Регрессии не миф!
- Если разработчиков, более одного - суровая реальность

Модульное тестироване

- Регрессии не миф!
- Если разработчиков, более одного - суровая реальность
- Замечательное ощущение, когда видишь такое



Статический анализ кода

“Потерянный Бозон Хиггса”

```
if ( A != 2 || B != 3 || A != 3 || B != 2 )
```

Статический анализ кода

“Потерянный Бозон Хиггса”

```
void G4QEnvironment::DecayBaryon(G4QHadron* qH)
{
  ....
  else if(qM<mSzPi) // Only Lambda+PiM is possible
  {
    fQPDG=lQPDG;    // Baryon is Lambda
    fMass=mLamb;
    sQPDG=pimQPDG;  // Meson is Pi-
    sMass=mPi;
  }
  else if(qM<mSzPi) // Both Lambda+PiM & Sigma0+PiM are possible
  {
    if(G4UniformRand()<.6)
    {
  ....
}
```

Статический анализ кода

“Потерянный Бозон Хиггса”

```
void G4QEnvironment::DecayBaryon(G4QHadron* qH)
{
  ....
  else if(qM<mSzPi) // Only Lambda+PiM is possible
  {
    fQPDG=lQPDG;    // Baryon is Lambda
    fMass=mLamb;
    sQPDG=pimQPDG;  // Meson is Pi-
    sMass=mPi;
  }
  else if(qM<mSzPi) // Both Lambda+PiM & Sigma0+PiM are possible
  {
    if(G4UniformRand()<.6)
    {
  ....
}
```

Динамический анализ кода

- Самый страшный враг сервисов 24/7...

Динамический анализ кода

- Самый страшный враг сервисов 24/7...

Динамический анализ кода

- Самый страшный враг сервисов 24/7...

Динамический анализ кода

- Профилировка
- Анализ покрытия кода тестами
- Отладка памяти

Динамический анализ кода

- Профилировка
- Анализ покрытия кода тестами
- Отладка памяти

```
ArrayType receiving( 3, 5 );

SomeAnotherArrayType init = { 1.0, 1.0, 1.0,
                              1.0, 1.0, 1.0,
                              1.0, 1.0, 1.0,
                              1.0, 1.0, 1.0,
                              1.0, 1.0, 1,0,
                            };

std::copy( init.cbegin(), init.cend(), receiving.begin() );
```

Динамический анализ кода

- Профилировка
- Анализ покрытия кода тестами
- Отладка памяти

```
ArrayType receiving( 3, 5 );

SomeAnotherArrayType init = { 1.0, 1.0, 1.0,
                              1.0, 1.0, 1.0,
                              1.0, 1.0, 1.0,
                              1.0, 1.0, 1.0,
                              1.0, 1.0, 1,0,
                            };

std::copy( init.cbegin(), init.cend(), receiving.begin() );
```

Интеграционное тестирование

- Вынесение длительных тестов

Интеграционное тестирование

- Вынесение длительных тестов
- Изоляция от окружения разработки

Интеграционное тестирование

- Вынесение длительных тестов
- Изоляция от окружения разработки
- Тестирование совместной работы компонентов

Continuous integration

~~LEEROY~~

Jenkins

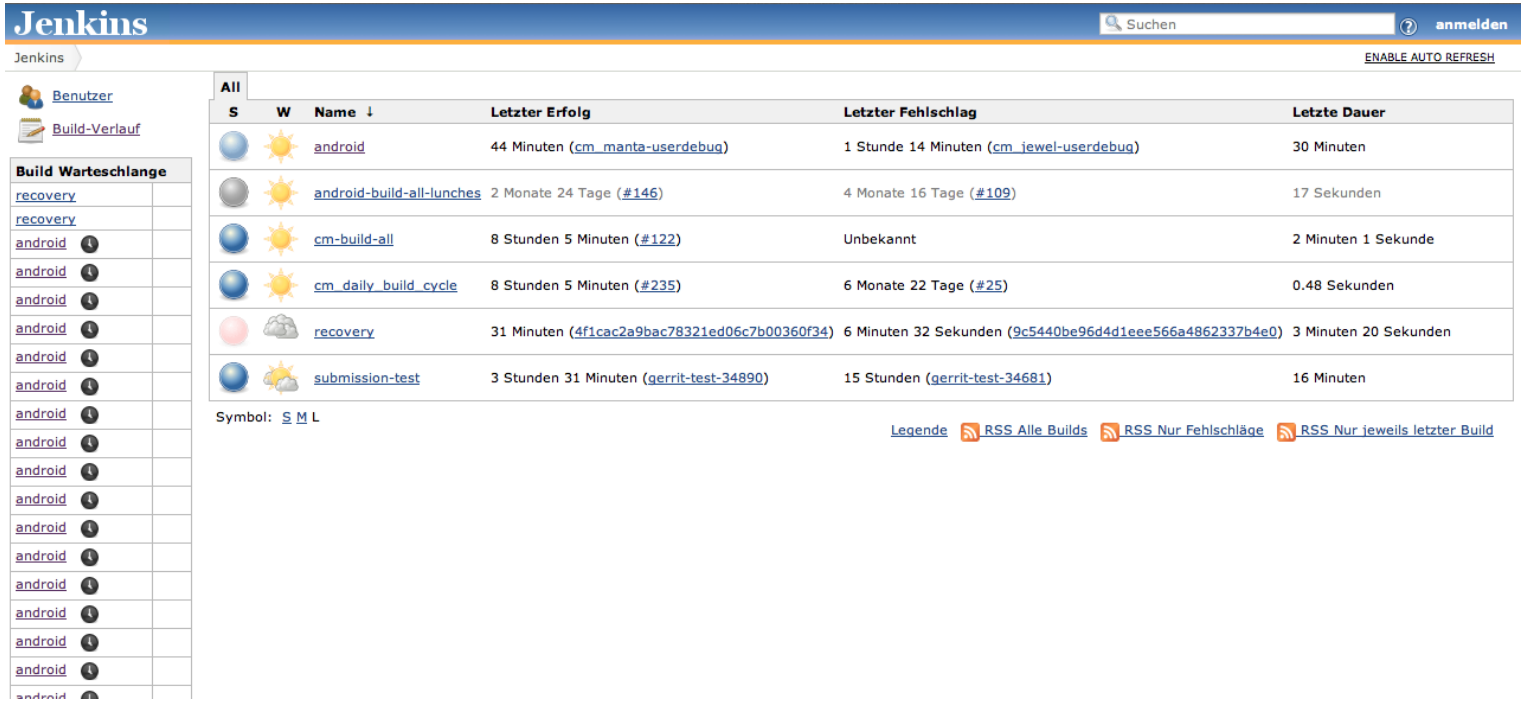
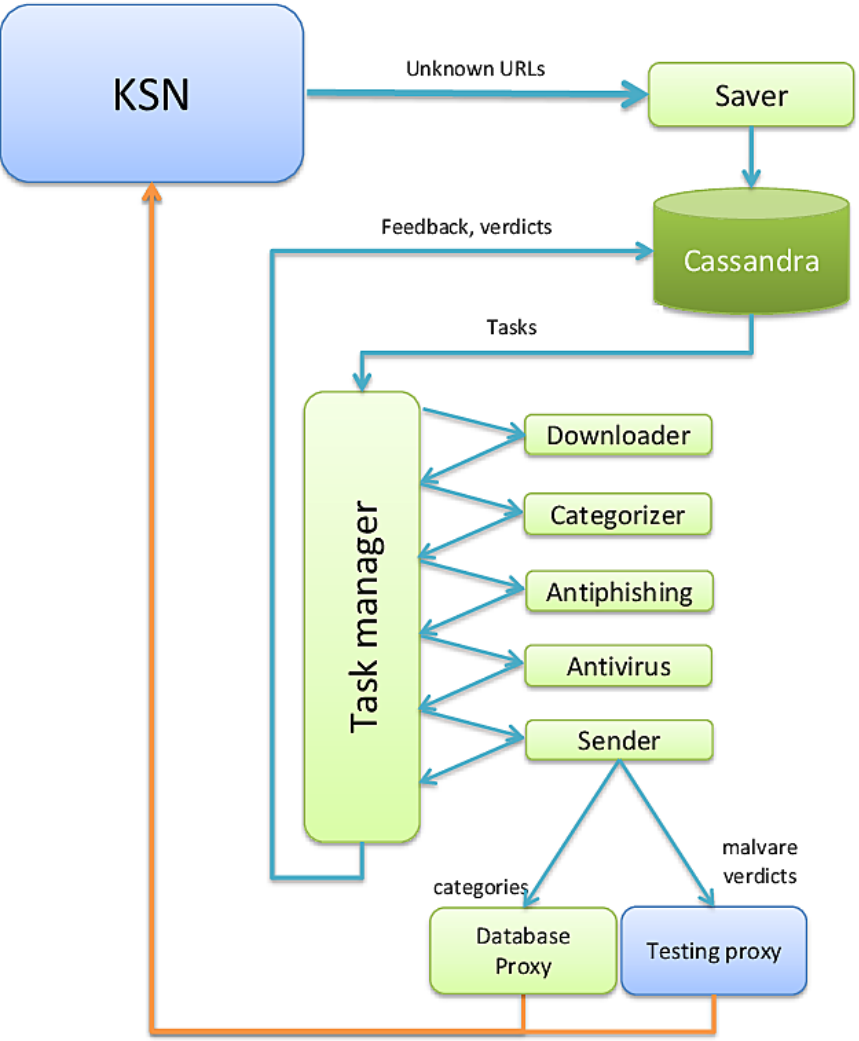
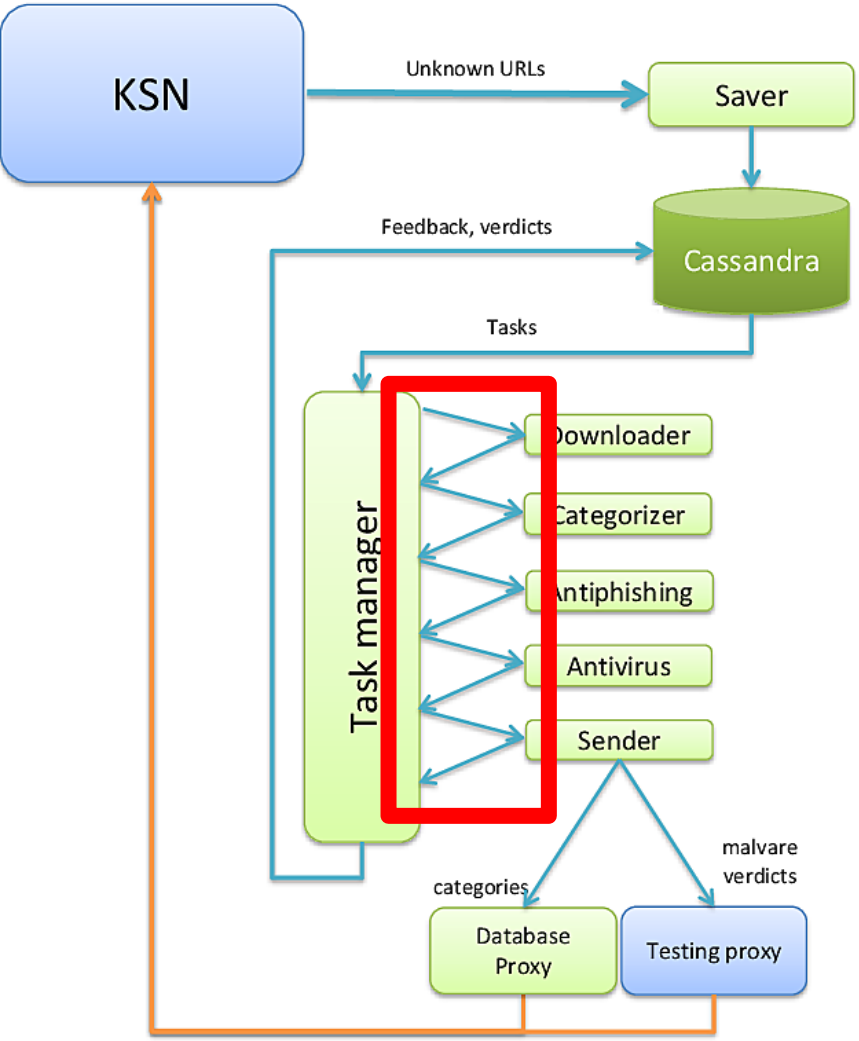


Схема сервиса



1. Получение запроса (из “облака”).
2. Сохранение в БД
3. Помещение в очередь задач
4. Процессинг на конвейере сканеров
5. Ответ в “облако”
6. Сохранение новых данных в БД

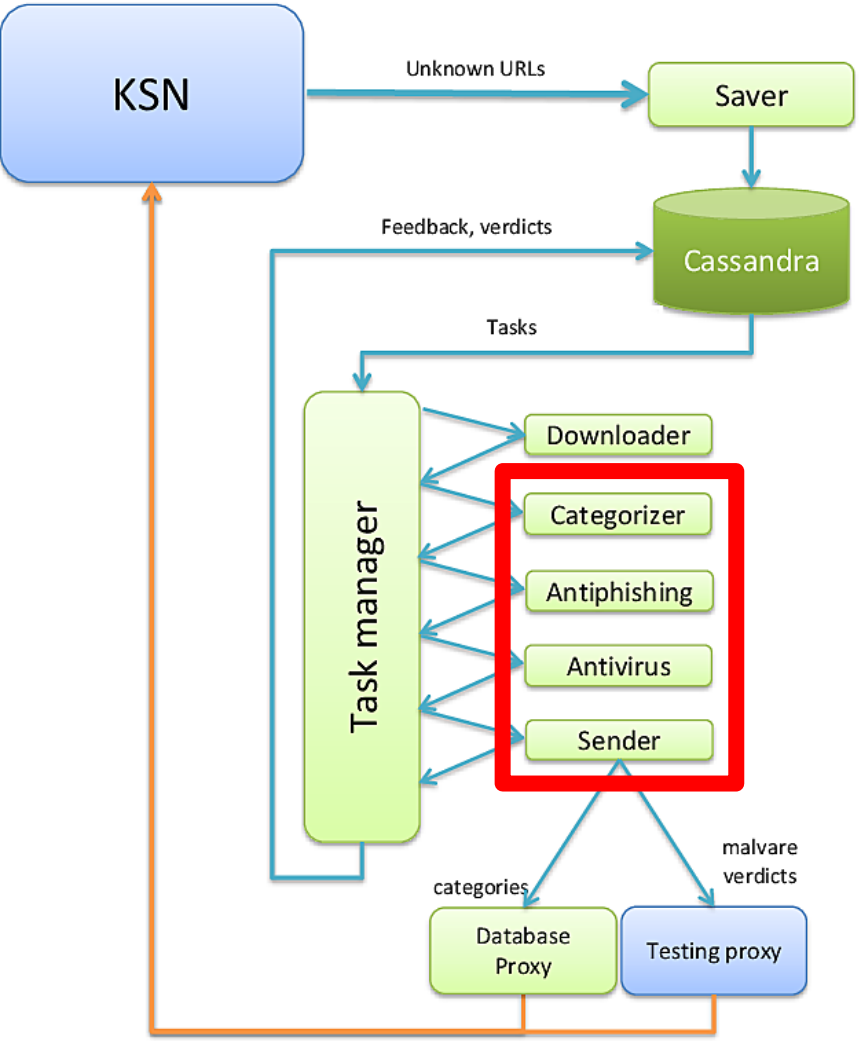
Горизонтальное масштабирование



На уровне машин

- распределенные очереди

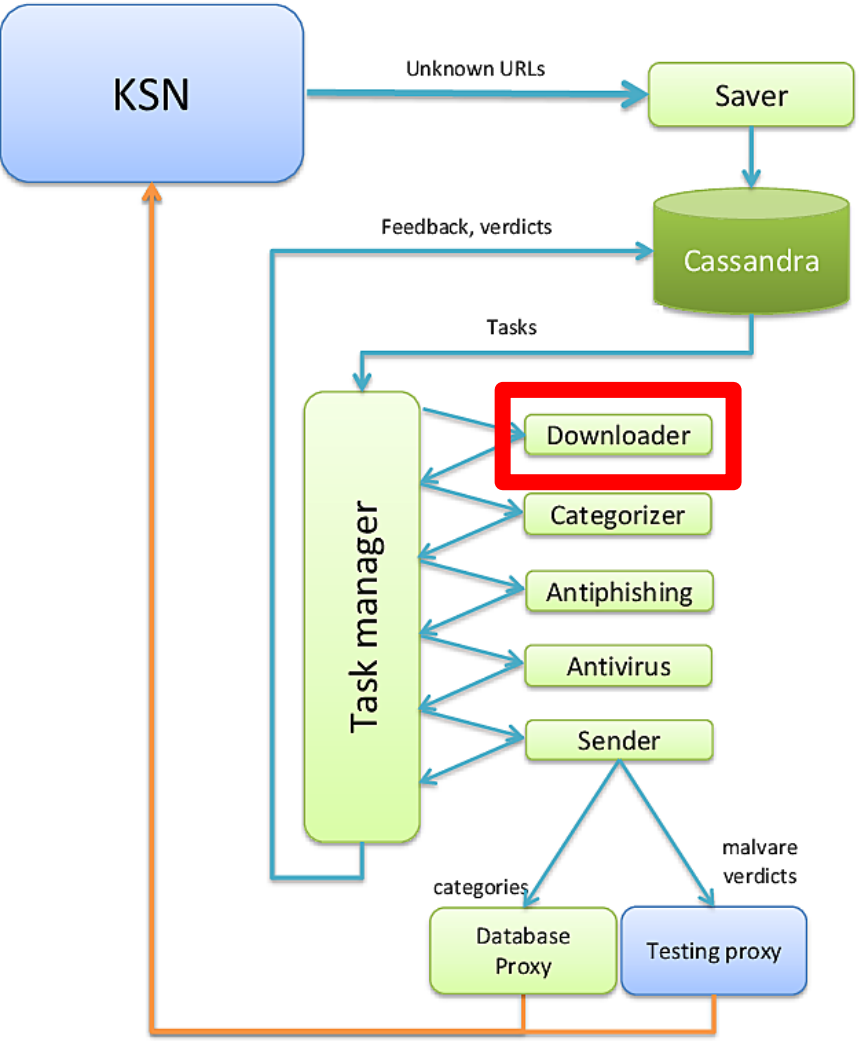
Горизонтальное масштабирование



На уровне процессоров/ядер

- многопоточность

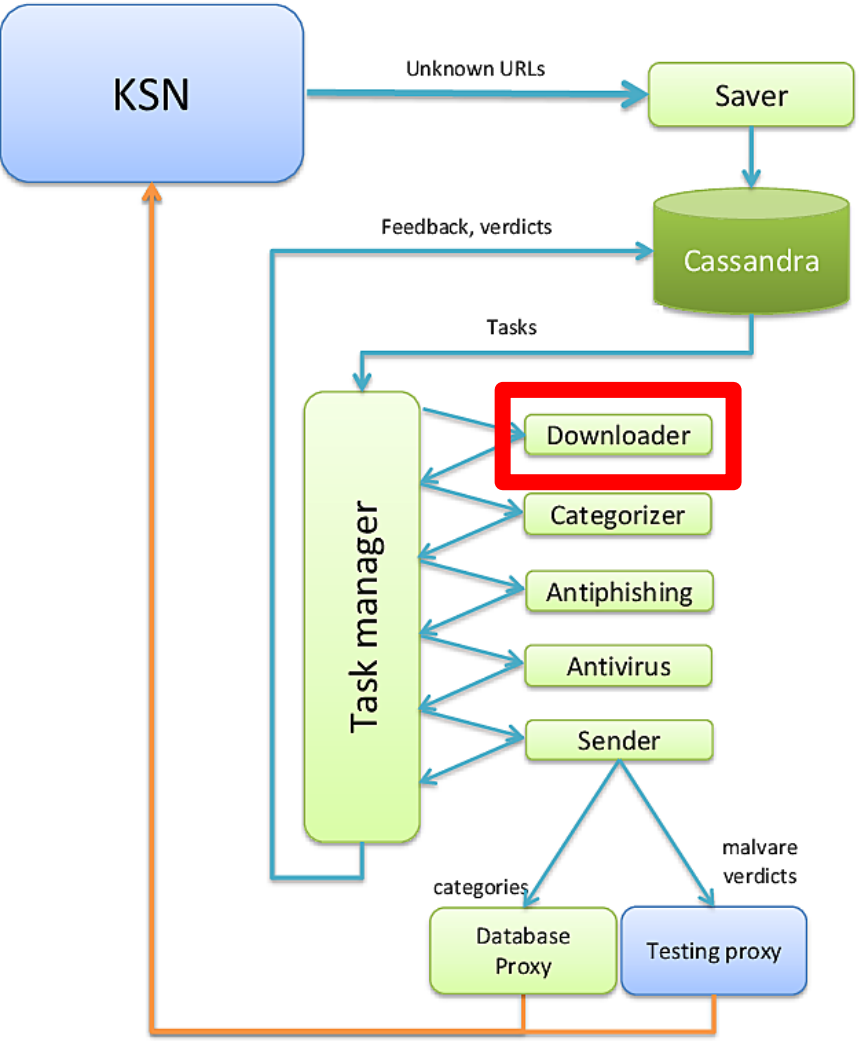
Горизонтальное масштабирование



На уровне процессоров/ядер

- многопоточность
- многопроцессность

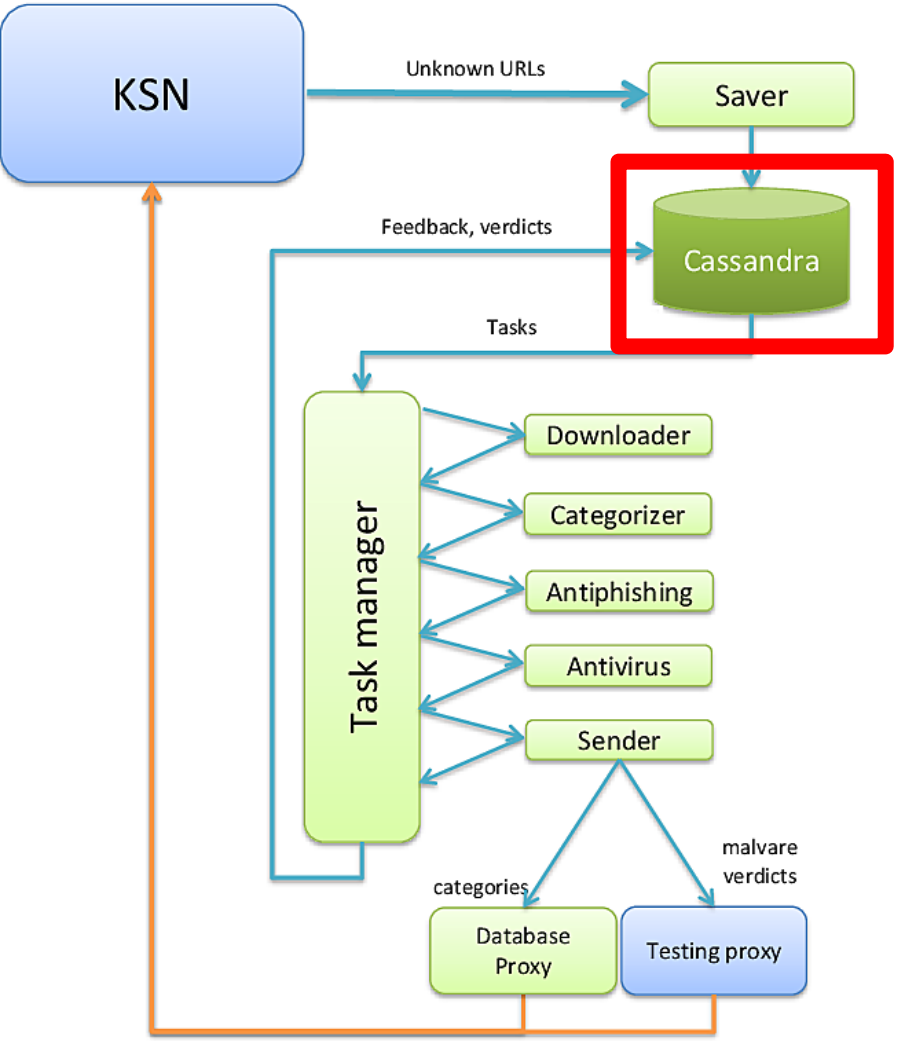
Горизонтальное масштабирование



На уровне процессоров/ядер

- многопоточность
- многопроцессность
  - утечки памяти в третьесторонних компонентах

Горизонтальное масштабирование



На уровне подсистемы хранения данных

NoSQL

- Cassandra
- Apache HBase
- MongoDB
- etc

Расширение “узких мест”

Использование жесткого диска:

3 M сайтов; из них **0.1M** с вердиктами

Расширение “узких мест”

Использование жесткого диска:

3 M сайтов; из них **0.1M** с вердиктами

1. Сохраняем всё; удаляем без вердиктов

Расширение “узких мест”

Использование жесткого диска:

3 M сайтов; из них **0.1M** с вердиктами

1. ~~Сохраняем всё; удаляем без вердиктов~~

Расширение “узких мест”

Использование жесткого диска:

3 M сайтов; из них **0.1M** с вердиктами

1. ~~Сохраняем всё; удаляем без вердиктов~~
2. Сохраняем всё; переносим с вердиктами; удаляем раздел

Расширение “узких мест”

Использование жесткого диска:

3 M сайтов; из них **0.1M** с вердиктами

1. ~~Сохраняем всё; удаляем без вердиктов~~
2. ~~Сохраняем всё; переносим с вердиктами; удаляем раздел~~

Расширение “узких мест”

Использование жесткого диска:

3 M сайтов; из них **0.1M** с вердиктами

1. ~~Сохраняем всё; удаляем без вердиктов~~
2. ~~Сохраняем всё; переносим с вердиктами; удаляем раздел~~
3. Всё держим в памяти; сохраняем только с вердиктами