

Skoltech

Skolkovo Institute of Science and Technology



Lomonosov Moscow
State University

RUNOS 2.0 OpenFlow Controller

Лекция 4

Василий Пашков

`pashkov@lvk.cs.msu.su`

План лекции



- 1. RUNOS 2.0**
- 2. Архитектура**
- 3. Особенности реализации**
- 4. Исследование производительности**



RUNOS контроллер



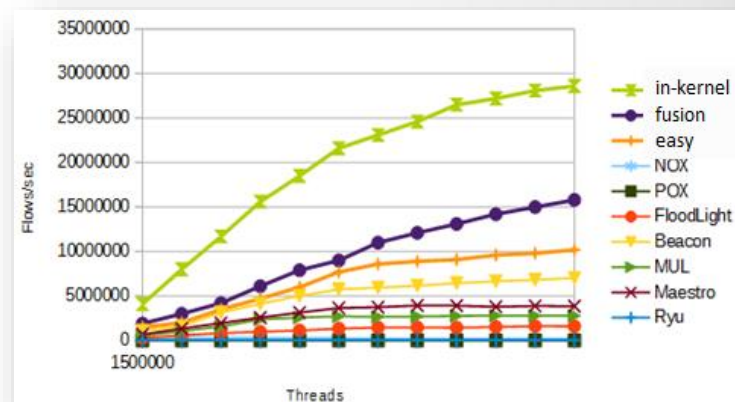
Система управления сетью первый российский SDN-контроллер RUNOS

RUssian Network Operation System

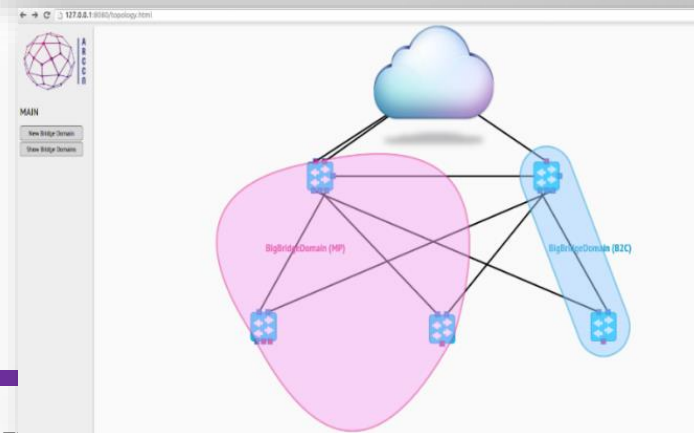
Есть разные варианты контроллера с единой базой и различным набором сервисов и приложений



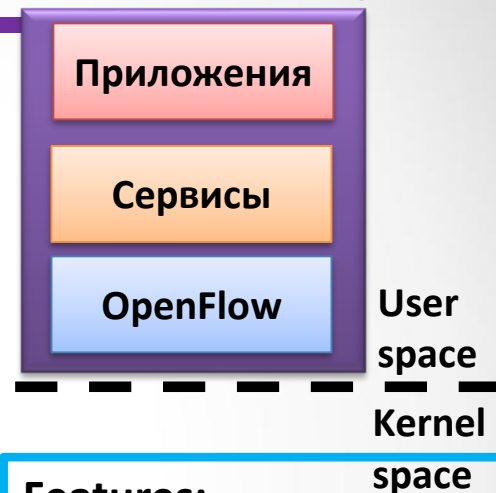
- **Открытая версия на Github** <http://arccn.github.io/runos/>
 - Своя база на C++11/14, а не Java
 - цель: упростить разработку сетевых приложений и не забывать о производительности
 - приложения: топология, маршрут, перестроение в случае обрыва, REST, WebUI, проактивная загрузка правил, резервирование Active-Passive



- **Внутренняя ядерная версия**
 - Супер-производительность 30 млн событий в секунду
 - Разработка приложений под заказчика
- **Внутренняя версия с приложениями под оператора связи**
 - База такая же, как и на Github. Заказчики сами могут разрабатывать приложения. Учиться по доступным материалам
 - Сервисы B2C, B2B (p2p, mp2mp, multicast, и т.п.)
 - Active-Standby режим



RUNOS: особенности



Features:

- Algorithmic policies (rule generation)
- Client-friendly API using EDSL grammar (low level details are hidden inside the runtime – overloading, templates)
- Modules composition (parallel and sequential composition)

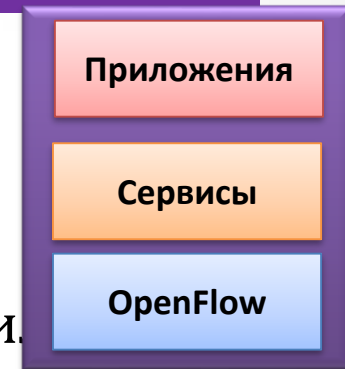
- Проблема запуска нескольких приложений, интеграция с приложениями других разработчиков
 - требуется статическая подстройка приложений под себя, порядок и способ передачи информации между ними.
 - нет механизма контроля и разрешения конфликтов между приложениями (генерация пересекающихся правил).
- В RUNOS стоит задача решить указанные выше проблемы:
 - часть настройки происходит автоматически по мета информации, связывание происходит динамически
 - разработана система разрешения конфликтов
 - Широкий набор сервисов для упрощения разработки новых приложений



Описания релизов

База:

- ядро контроллера
- построение топологии
- построение маршрута через всю сеть
- первая версия системы генерации правил
 - Распределение приоритетов, комбинация правил
 - LOAD, MATCH, READ абстракции
 - На основе MAPLE
- Rest API (совместимый с Floodlight)
- WebUI (мониторинг загрузки, просмотр таблиц, удаление и добавление правил)
- Проактивная загрузка правил
- Холодное резервирование
- ARP кеширование



User
space

Kernel
space



Описания релизов

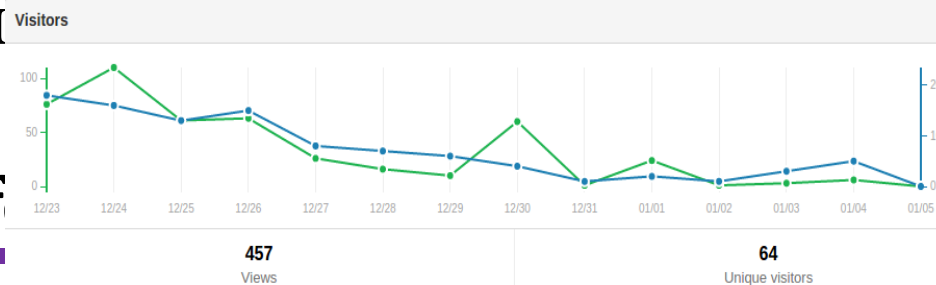
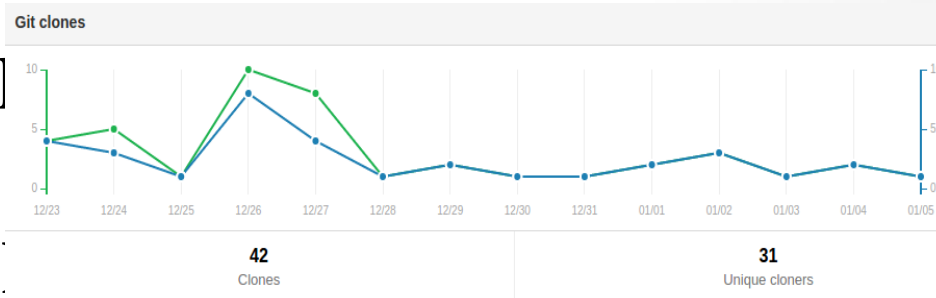
Версия **0.6** – один из последних больших релизов

- **Полное обновление структуры ядра контроллера.** Нет привязка к конкретной версии протокола OpenFlow. Своя модель, расширяемая под любые новые поля, в том числе и специфические для оборудования.
- **Пакетная грамматика для сетевых приложений.** Упрощает разработку новых приложений.
 - “pkt[eth src] == eth addr”
 - “if (ethsrc == A || ethdst == B) doA else doB”
 - “test((eth_src & “ff0.....0”) == “....”)”
 - “modify(ip_dst >> “10.0.0.1”)”
 - decision are “unicast()”, “broadcast()”, “drop()”
- **Обновление системы генерации правил** — повышена скорость работы и улучшена генерация правил (по количеству правил и числу приоритетов).
- **Система тестов.**
- **Runos-book** подробная документация и инструкция по разработке новых приложений.
- **Приложения:** stp, arp, flow-manager

Проект RUNOS



- Исходный код <http://arccn.github.io/runos/>
 - Apache, version 2.0
- Tutorial (Readme.md + Runos-book)
 - Как установить, запустить, написать свое первое п
- Виртуальная машина
 - Уже собранный конт
 - Средства для работы
- Список рассылки
 - Google group runos-of

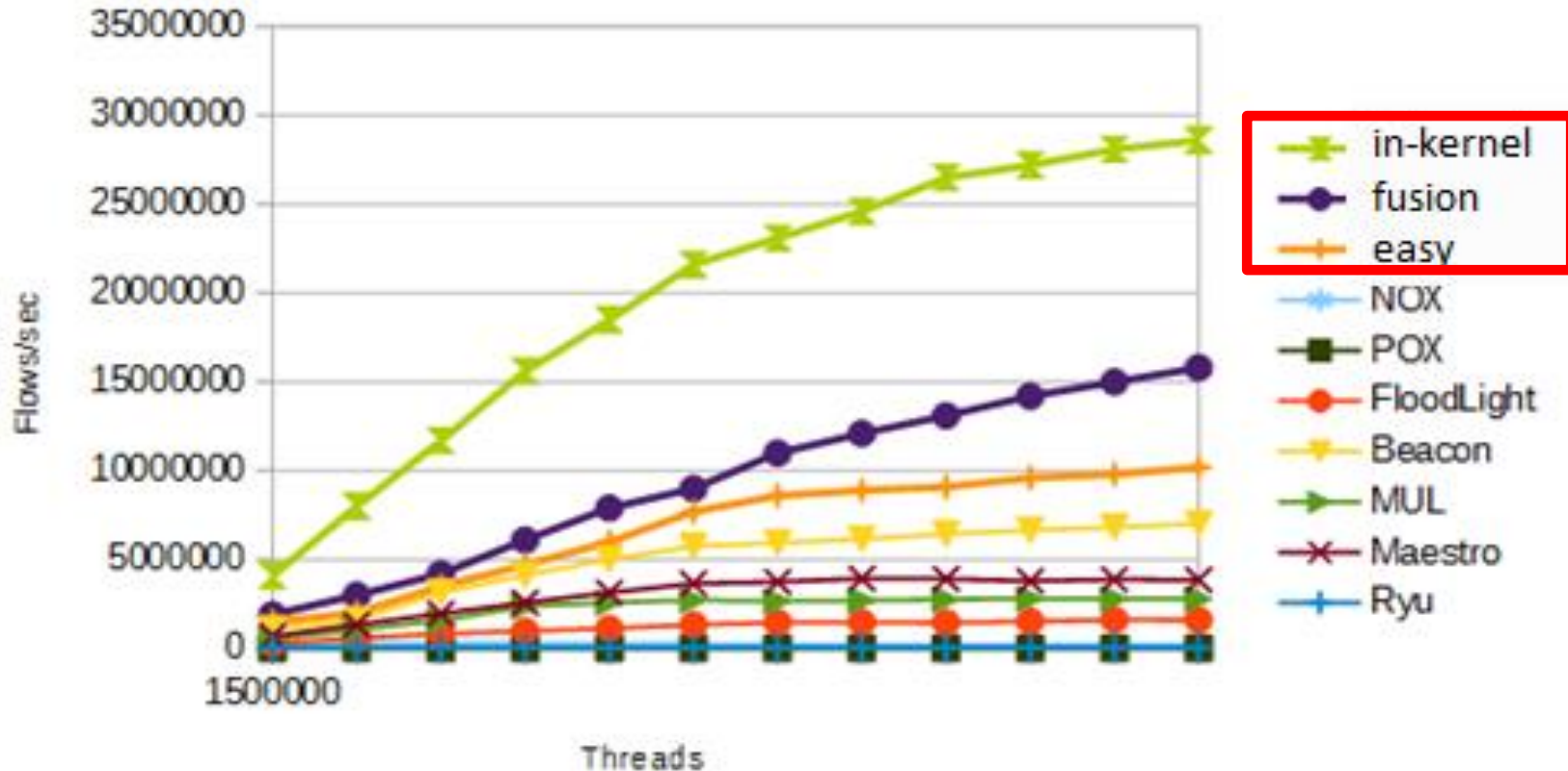


Новые релизы Runos (открытая версия)



- v0.6.1
 - Чистый OpenFlow интерфейс для программирования коммутаторов (возможность самим формировать правила)
 - Обновленный REST: совместимость с Ryu, библиотека для Postman
- v0.7
 - Оптимизации системы генерации правил:
 - Глобальное видение сети
 - Оптимизация работы – по кол-ву FlowMod
 - Новые приложения: корпоративная сеть
 - Улучшение Web интерфейса (перенос часть функционала из коммерческой версии)

Производительность



- Производительность : **10 млн. потоков в секунду**
- Задержка: **55 мкс**



Реализация

Ключевые слова: C++11/14/17, QT, Boost (asio, proto, graph)

Основные сторонние компоненты:

- **libfluid project** (_base, _msg)
 - для взаимодействия со свитчами и разбор OpenFlow 1.3 сообщений
- **libtins**
 - разбор пакетов внутри OpenFlow сообщений
- **glog** (google log)
 - логирование, многопоточное
- **tcmalloc** (google performance tools)
 - альтернативная более быстрая реализация malloc/free
- **json11**
 - разбор конфигурационного файла
- **boost graph**
 - Хранение топологии, поиск маршрута



Параметры запуска

Config (json):

```
“controller”: {  
  “threads”: 4  
},  
“loader”: {  
  “threads”: 3  
},  
“link discovery”: {  
  “poll-interval” : 10,  
  “pin-to-thread” : 2  
},  
“learning switch”: {  
}  
...
```

- Задается количество нитей контроллера
 - для взаимодействия со свитчами
 - для работы приложений
- Список приложений
 - их параметры (poll-interval)
 - зафиксировать нить выполнения или выделить в монопольное пользование (pin-to-thread, own-thread)

Архитектура



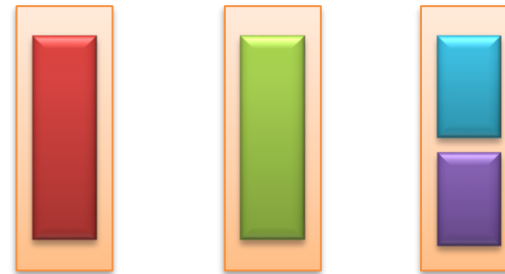
Инициализация контроллера:

1. Запуск нужного количества нитей
2. Запуск служебных компонент
3. Запуск приложений и распределение их по нитям
4. Определение порядка обработки событий приложениями

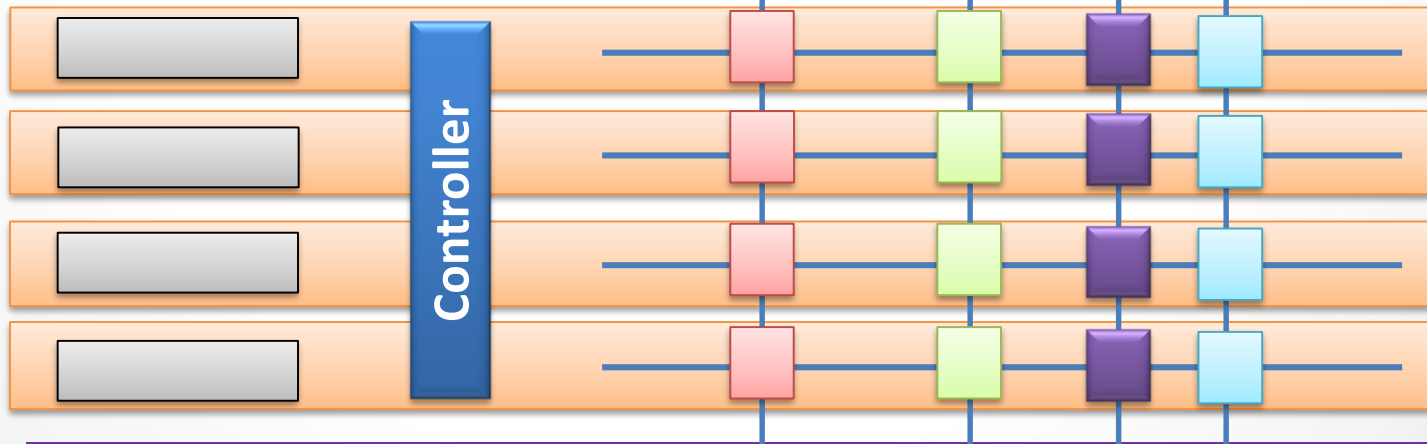
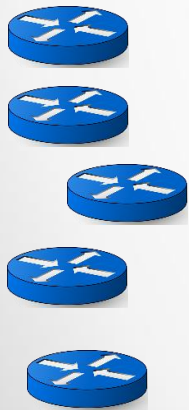
Apps



App pool



Workers



Logical pipelines

Open Source



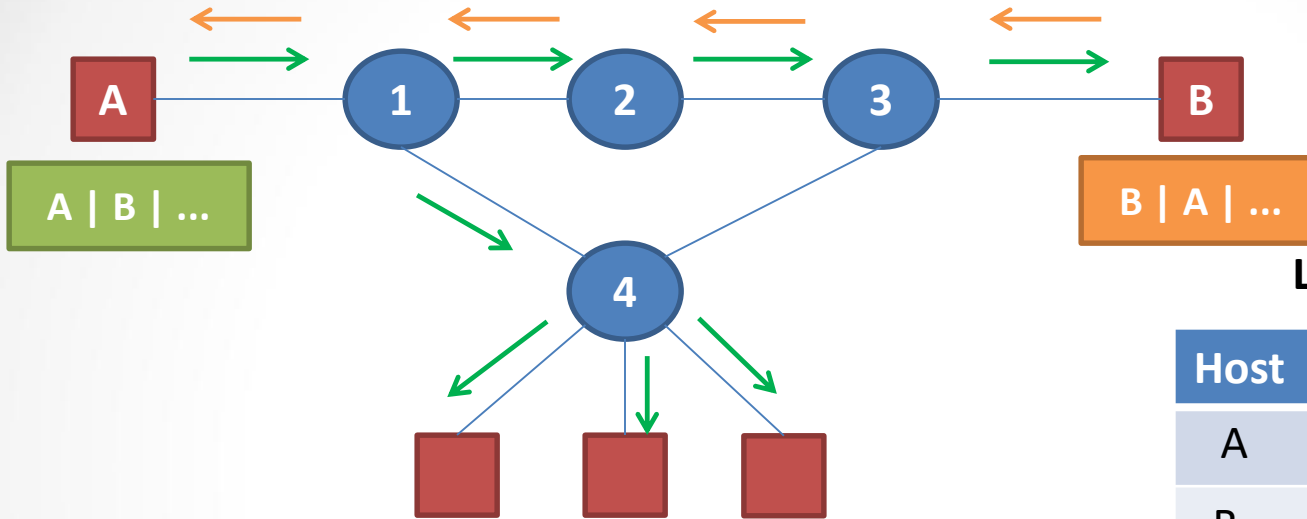
- Два типа OpenSource проектов:
 - ради Идеи: “Free as in Freedom”
 - продаем свои компетенции, а не продукт
 - доработка под нужды заказчика,
 - продавать advanced версии и плагины (eg, приложения для Runos)
 - community вокруг (семинары, обучение)
- Важна лицензия (*): Apache, BSD или GPL, Eclipse, проприетарные лицензии, перелицензирование за деньги
- Угрозы:
 - Скопируют и будут продавать под другим названием (eg, runos-ng)
 - Будут дорабатывать своими силами (для компаний с большим R&D)



Часть VI: Разработка приложений для RUNOS контроллера



First application – L2 learning



L2 learning table

Host	Switch:port
A	1:1
B	3:2

- What is L2 learning?
 - L2 table – where particularly host resides (host <->sw:port)
- A->B. What should we do on sw1?
 - Learn and broadcast
- B->A. What should we do on sw3?
 - Learn and unicast
- **Advanced question: will it work for ping utilities? Ping 10.0.0.2 (assuming B has this IP)**
 - Yes, arp (broadcast), ip (icmp)



Host Databases

```
class HostsDatabase {
    boost::shared_mutex mutex;
    std::unordered_map<ethaddr, switch_and_port> db;

public:
    void learn(uint64_t dpid, uint32_t in_port, ethaddr mac)
    {
        LOG(INFO) << mac << " seen at " << dpid << ':' << in_port;
        {
            boost::unique_lock< boost::shared_mutex > lock(mutex);
            db[mac] = switch_and_port{dpid, in_port};
        }
    }

    boost::optional<switch_and_port> query(ethaddr mac)
    {
        boost::shared_lock< boost::shared_mutex > lock(mutex);

        auto it = db.find(mac);
        if (it != db.end())
            return it->second;
        else
            return boost::none;
    }
};
```

.insert(...)

L2 forwarding application



```
// Get required fields
ethaddr dst_mac = pkt.load(ofb_eth_dst);

db->learn(connection->dpid(),
          pkt.load(ofb_in_port),
          packet_cast<TraceablePacket>(pkt).watch(ofb_eth_src));

auto target = db->query(dst_mac);
// Forward
if (target) {
    flow->idle_timeout(60.0);
    flow->hard_timeout(30 * 60.0);
} else {
    flow->broadcast();
    return PacketMissAction::Continue;
}

auto route = topology->computeRoute(connection->dpid(),
                                     target->dpid);

if (route.size() > 0) {
    flow->unicast(route[0].port);
} else {
    flow->idle_timeout(0.0);
    LOG(WARNING) << "Path from " << connection->dpid()
                 << " to " << target->dpid << " not found";
}

return PacketMissAction::Continue;
```



Спасибо за внимание!

Василий Пашков
pashkov@lvk.cs.msu.su
