

Обеспечение качества сервиса в центрах обработки данных

Чемерицкий Евгений Викторович

к.ф.-м.н.



Эффективность современных ЦОД

страшная тайна – утилизация в 30% это хорошо

- Неравномерное потребности приложений
 - Большинство приложений требовательны к одному из ресурсов машины – остальные простаивают
- Сложность планирования вычислений
 - Требования теннантов и их приложений изменяются во времени, их сложно предсказывать
- Поддержание высокого уровня сервиса
 - Нужно быть готовыми к необходимости подключить дополнительное оборудование при возникновении отказов или обновлении

Повышение гибкости

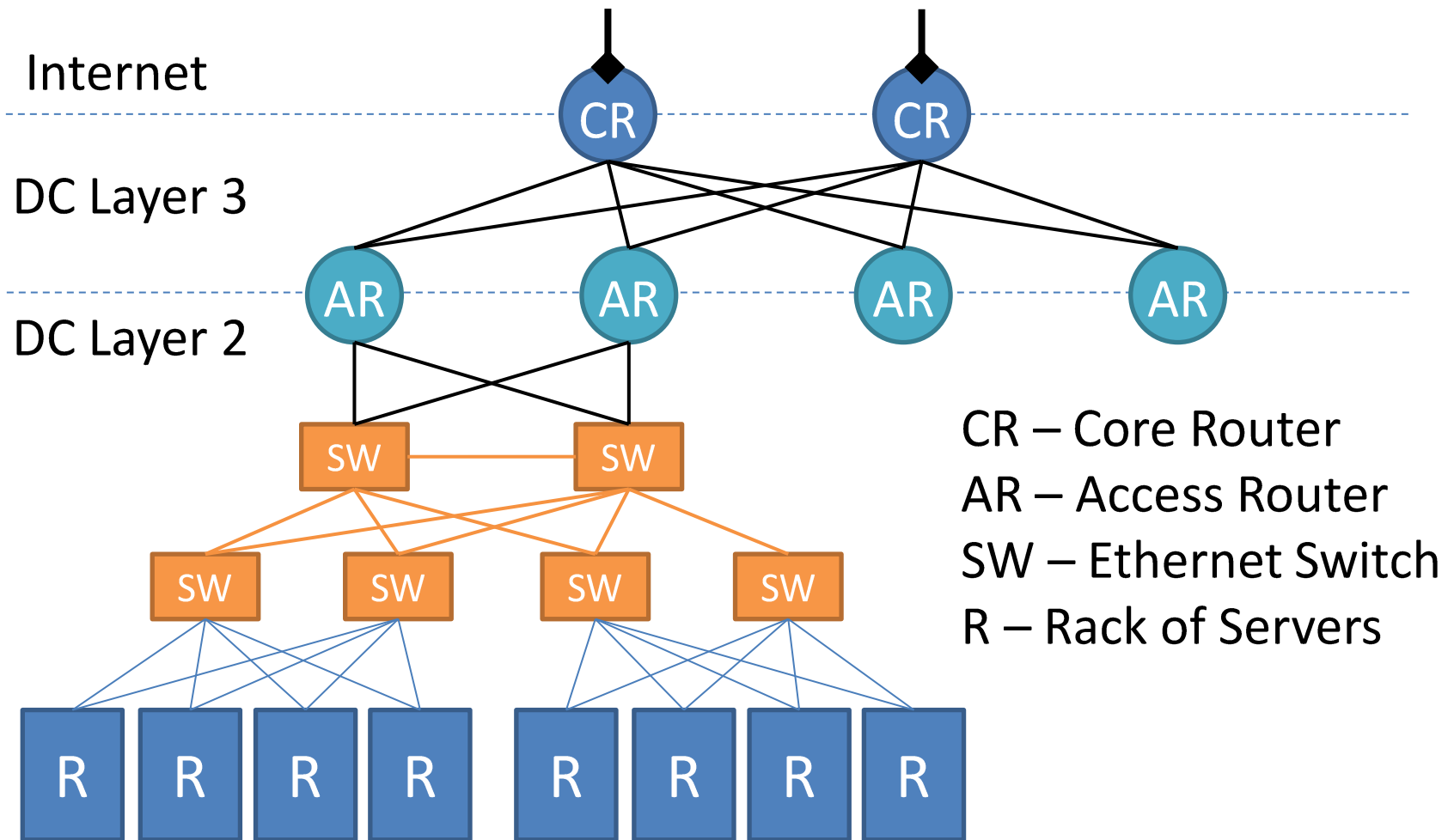
любой сервис на любом сервере

Принцип ***resource pooling*** –
превращение множества серверов ЦОД в
единый пул вычислительных ресурсов,
которые можно распределять по задачам
произвольным образом

- Гибкое управление вычислителями
Распределённые файловые системы
- Быстрый обмен данными между
приложениями, вне зависимости от их
физического расположения

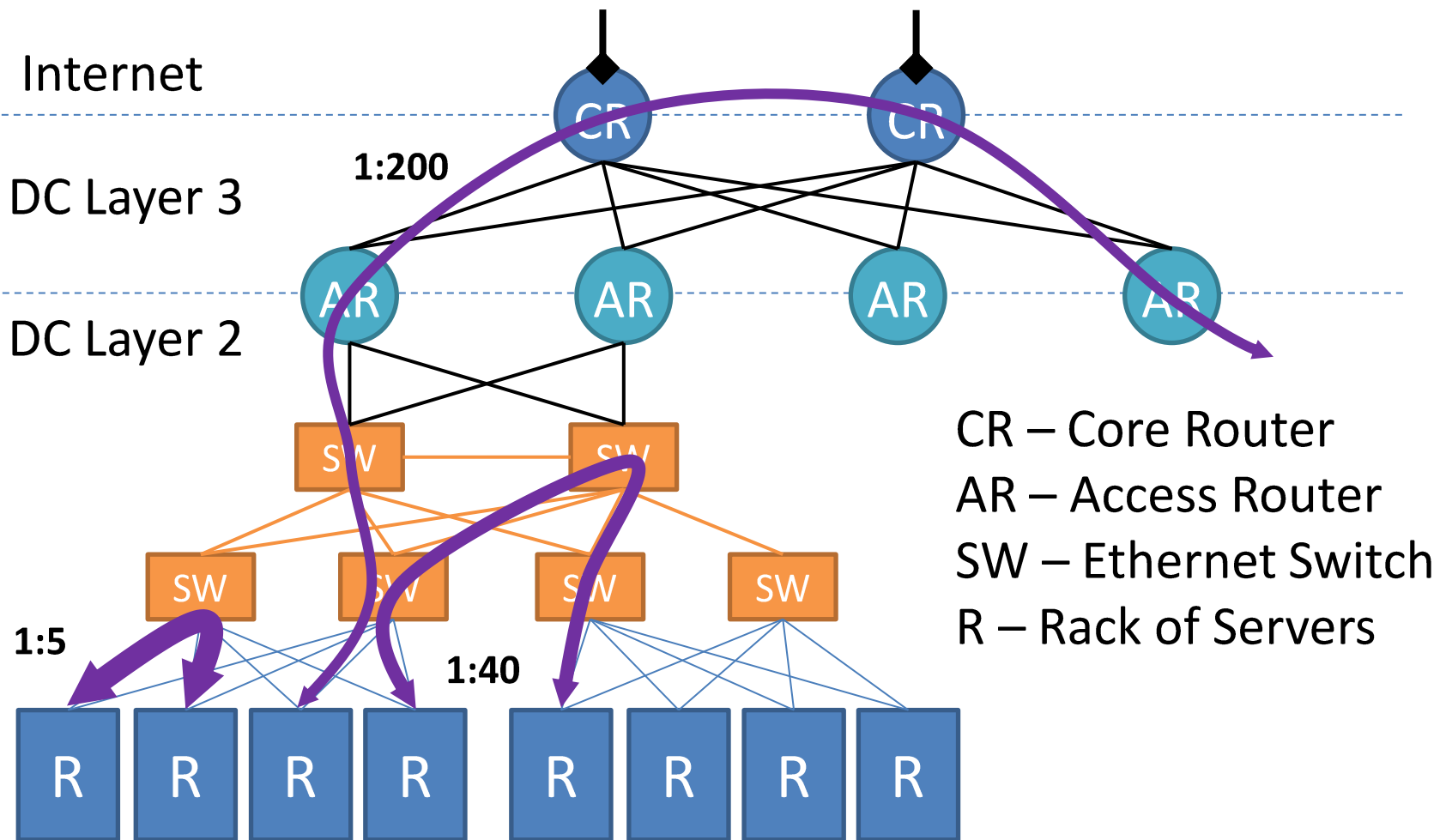
Классическая топология ЦОД

Data Center: Load balancing Data Center Services, Cisco 2004



Классическая топология ЦОД

Data Center: Load balancing Data Center Services, Cisco 2004

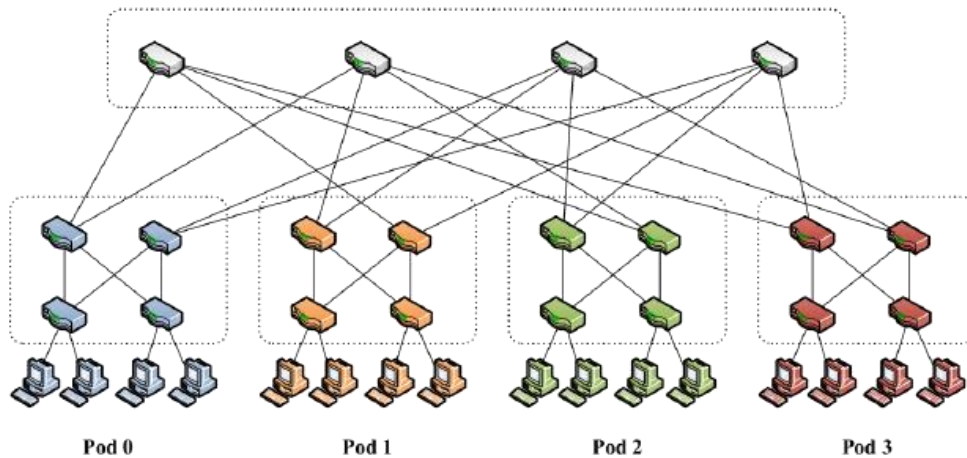


Проблемы традиционной топологии

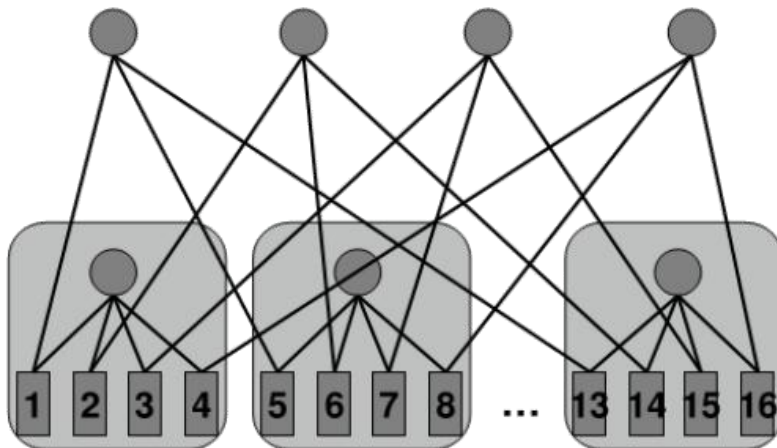
- Слишком большая нагрузка на центральные коммутационные устройства
 - Плохая масштабируемость
 - Высокая стоимость оборудования
 - Низкая отказоустойчивость
- Неравномерная связность топологии
 - Фрагментированность ресурсов
 - Низкая *Bisection-Bandwidth* – совокупная пропускная способность сети ЦОД при наихудшем распределении нагрузки по узлам

Альтернативные топологии

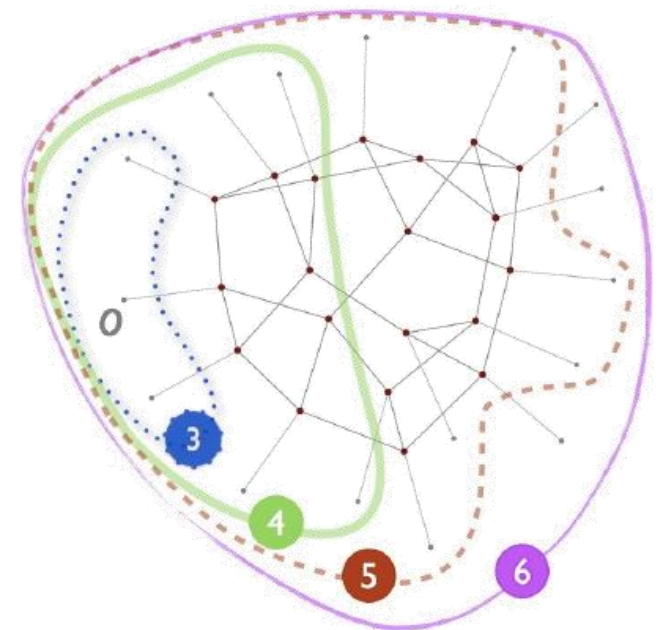
Fat-tree [SIGCOMM'08]



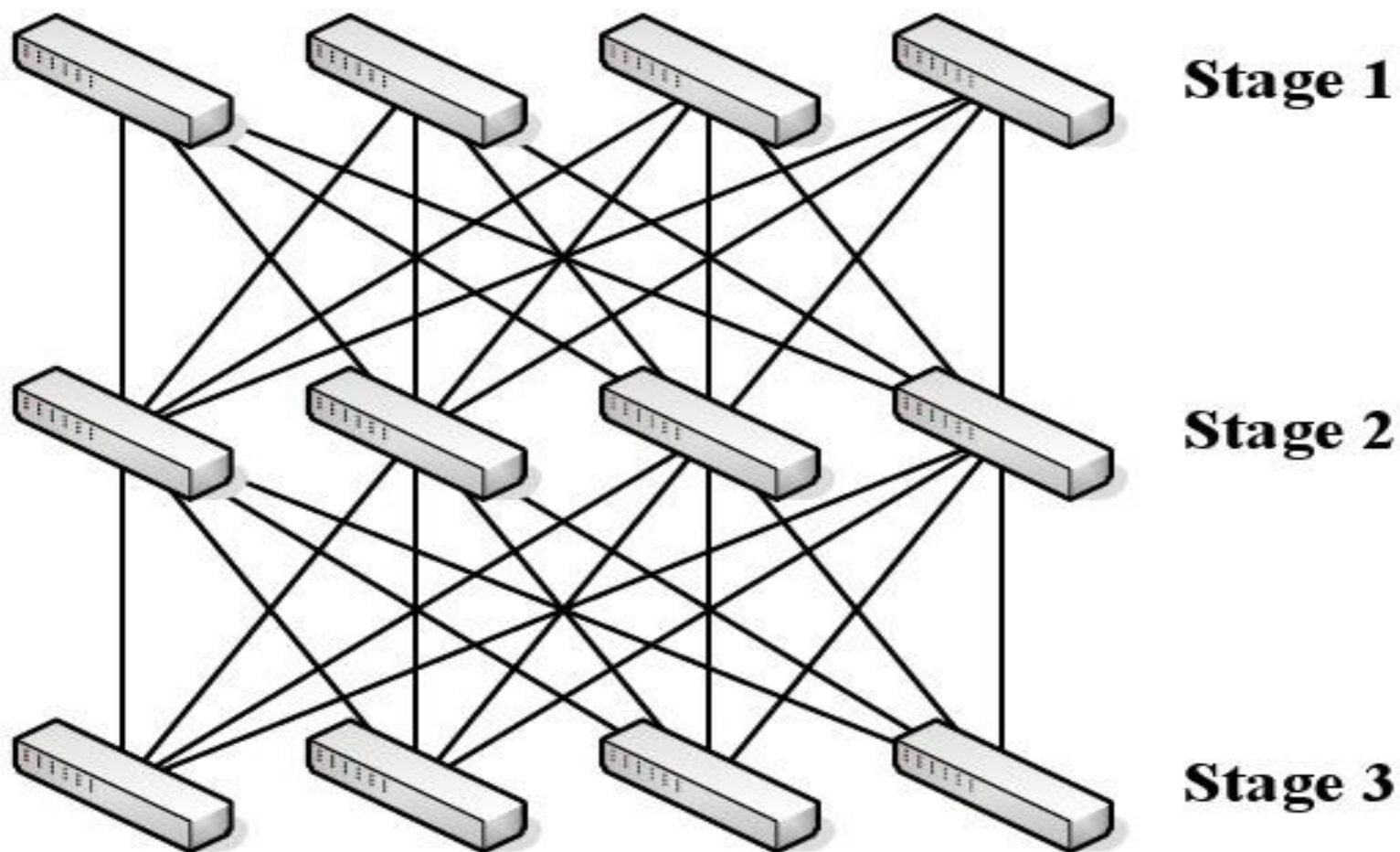
BCube [SIGCOMM'10]



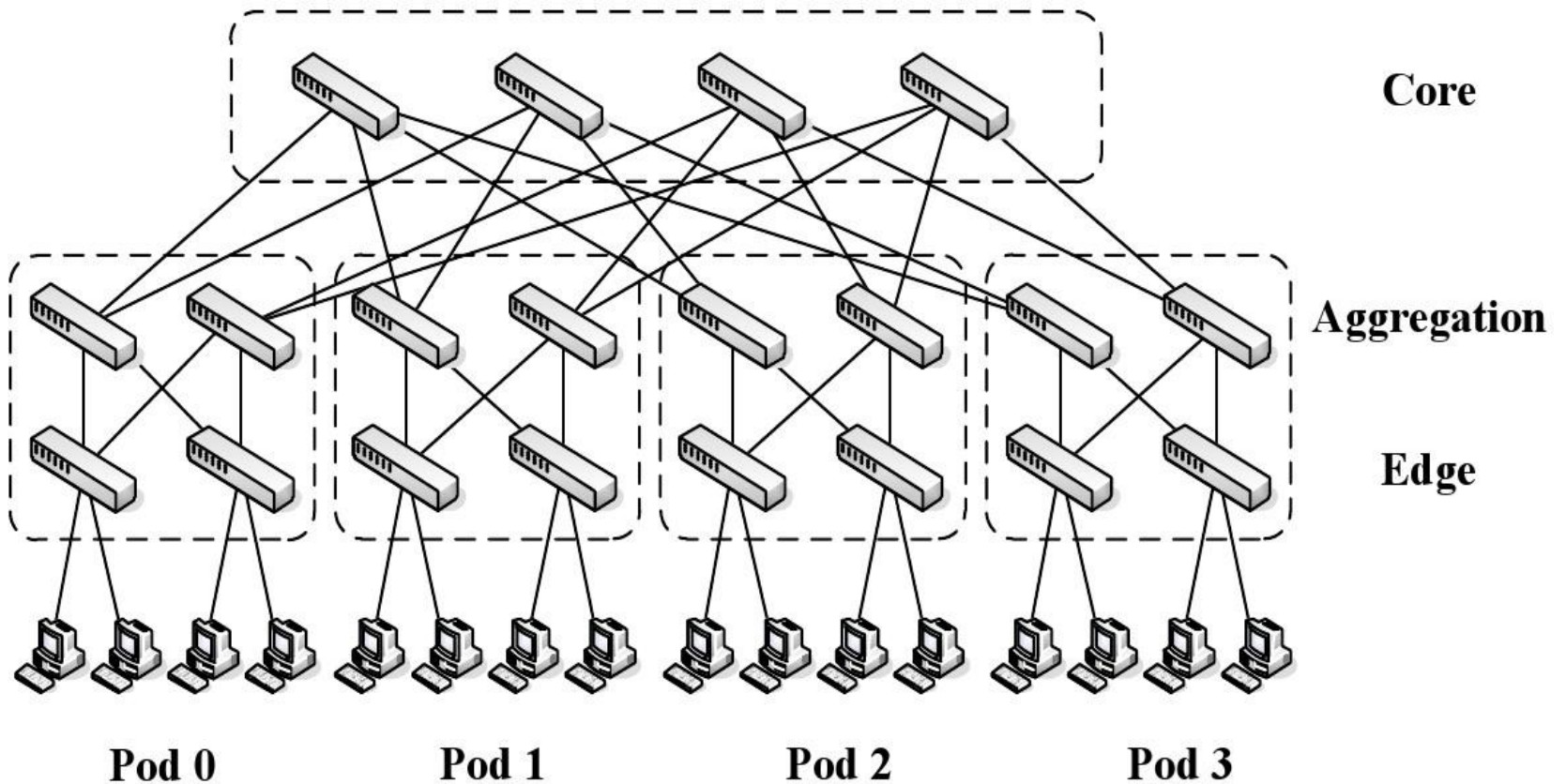
Jellyfish (random) [NSDI'12]



Топологии на базе сетей Клозе

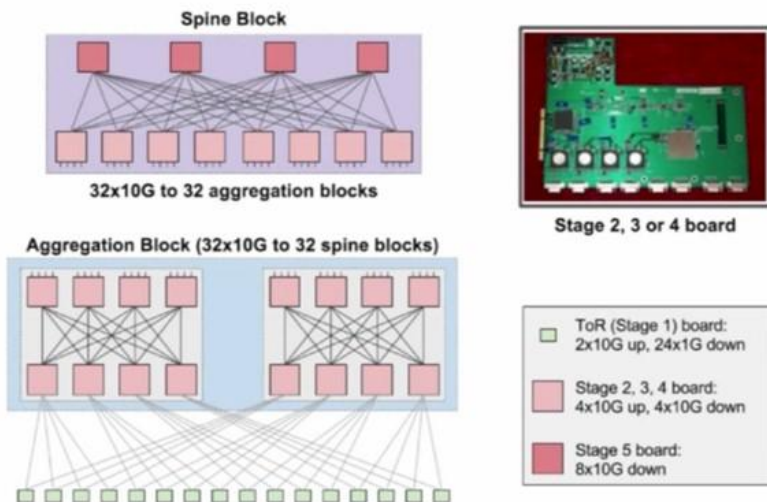


FAT Tree

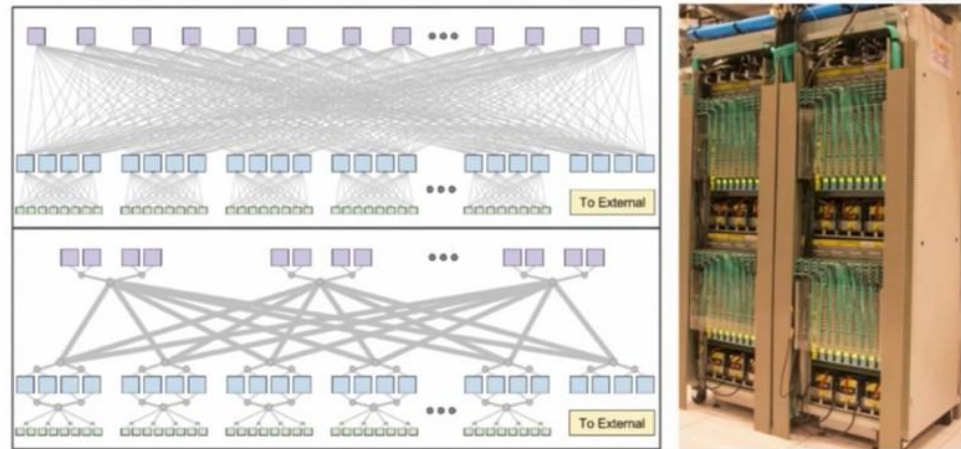


Поколения ЦОД Google

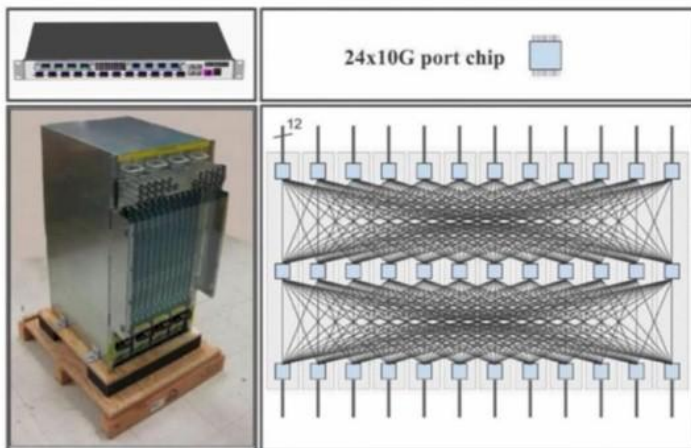
Firehose 1.0



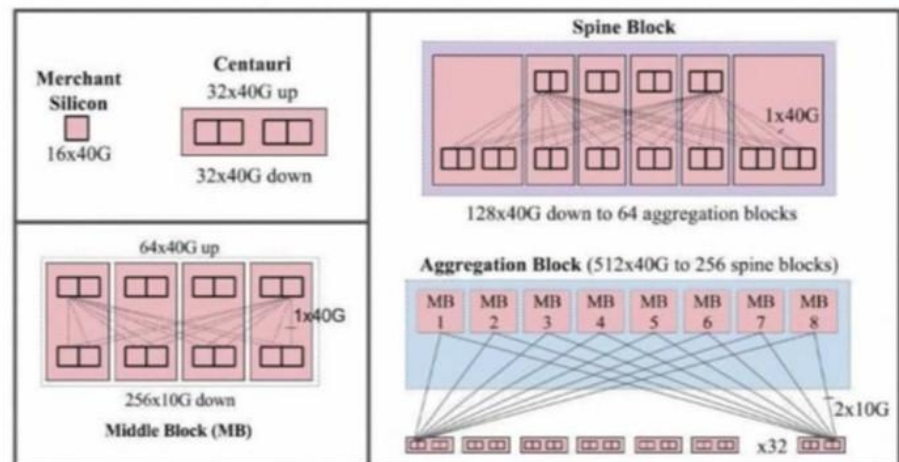
Watchtower



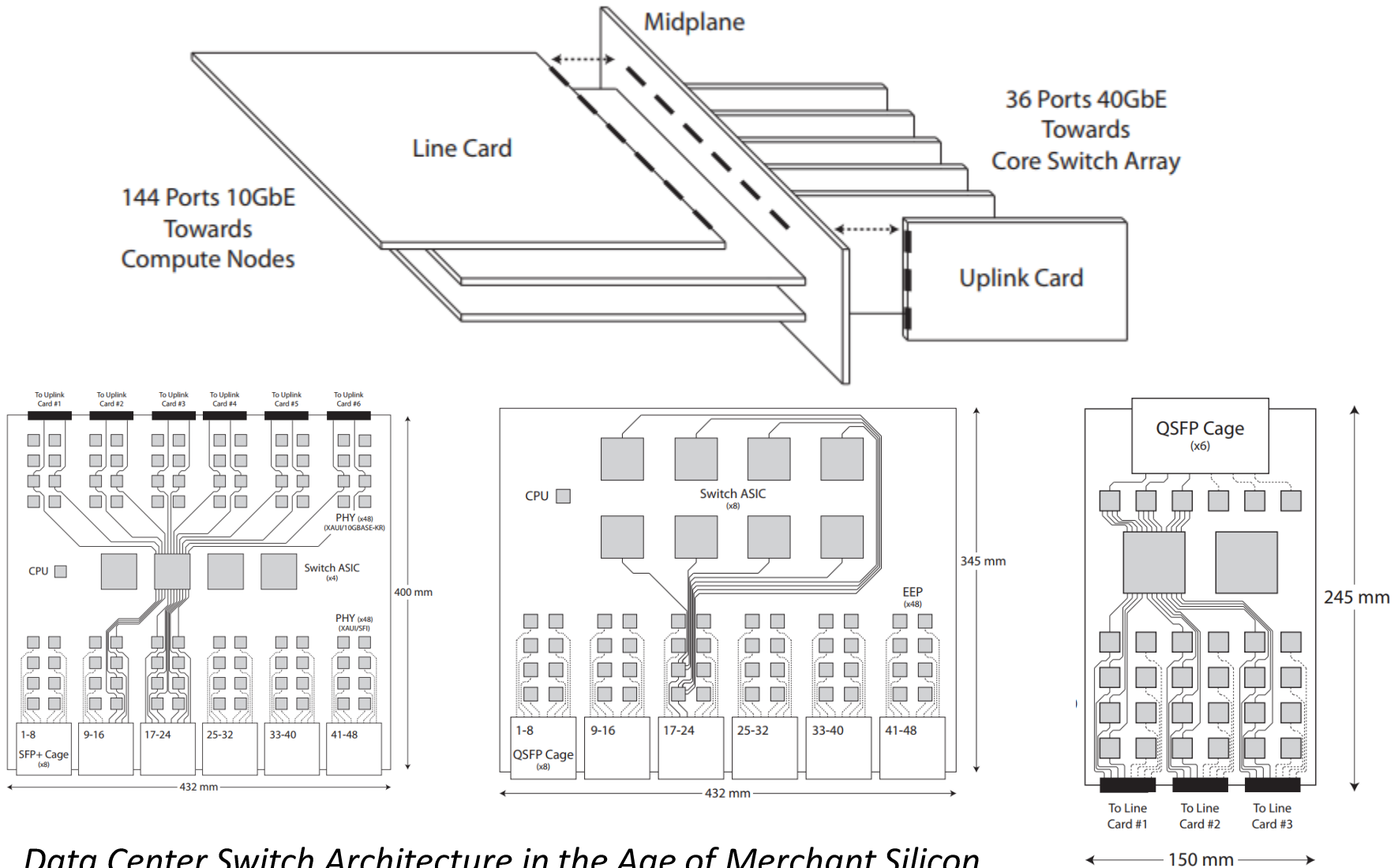
Saturn



Jupiter Building Blocks



Как строить коммутаторы для ЦОД?

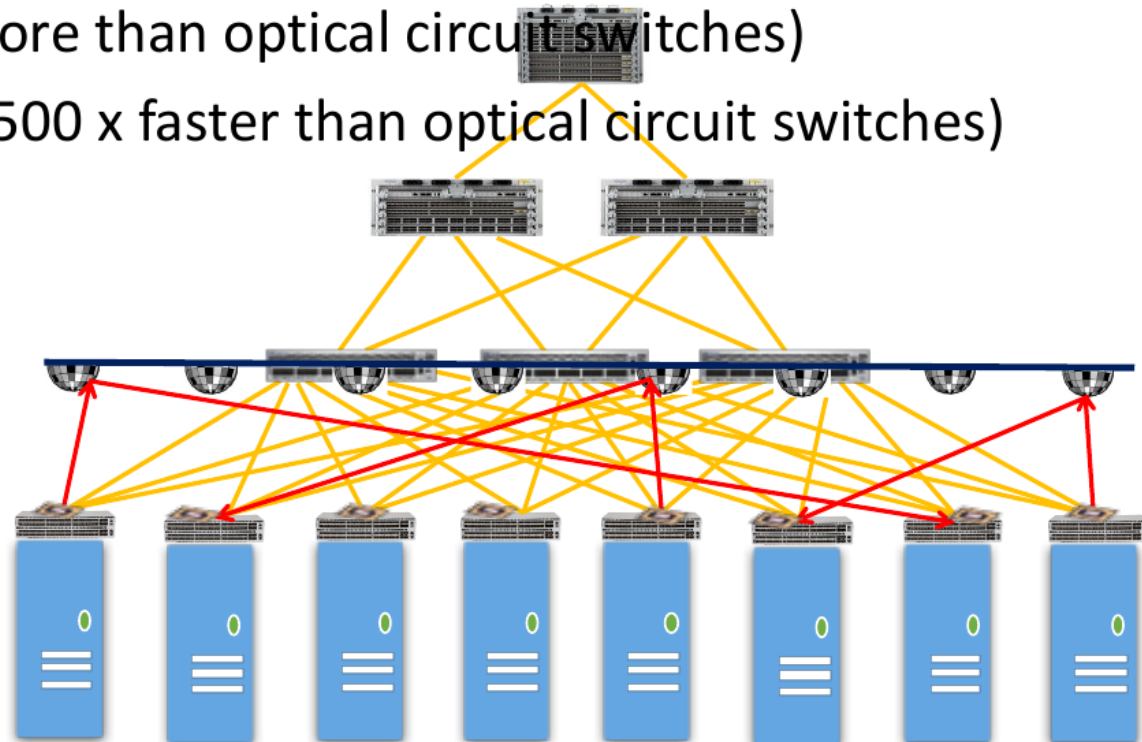
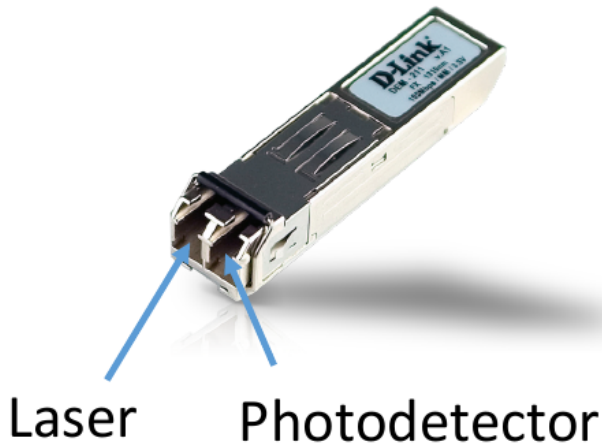


Data Center Switch Architecture in the Age of Merchant Silicon

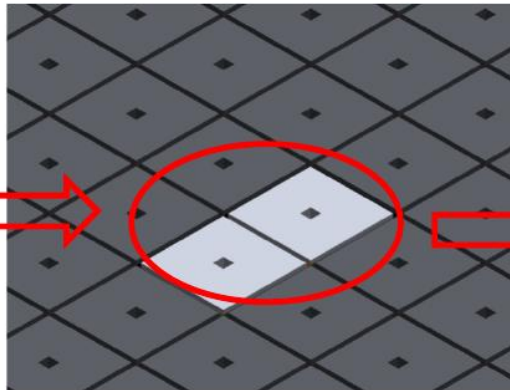
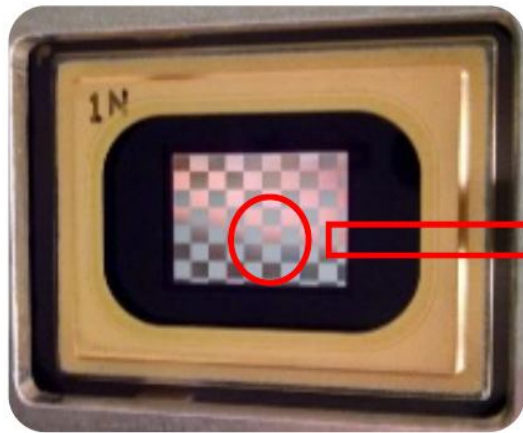
Можно ли обеспечивать высокую скорость обмена по запросу?

ProjecToR interconnect

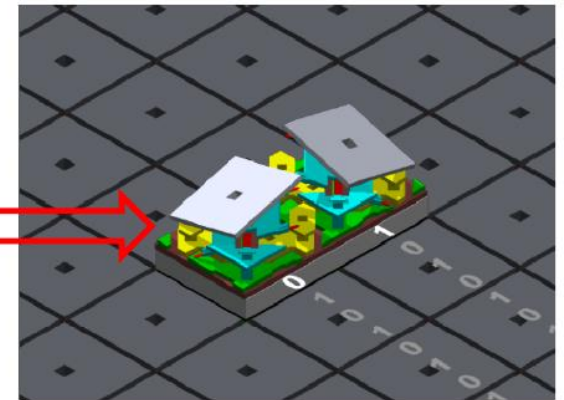
- Free-space topology (seamless)
- 18,000 fan-out (60 x more than optical circuit switches)
- 12 us switching time (2500 x faster than optical circuit switches)



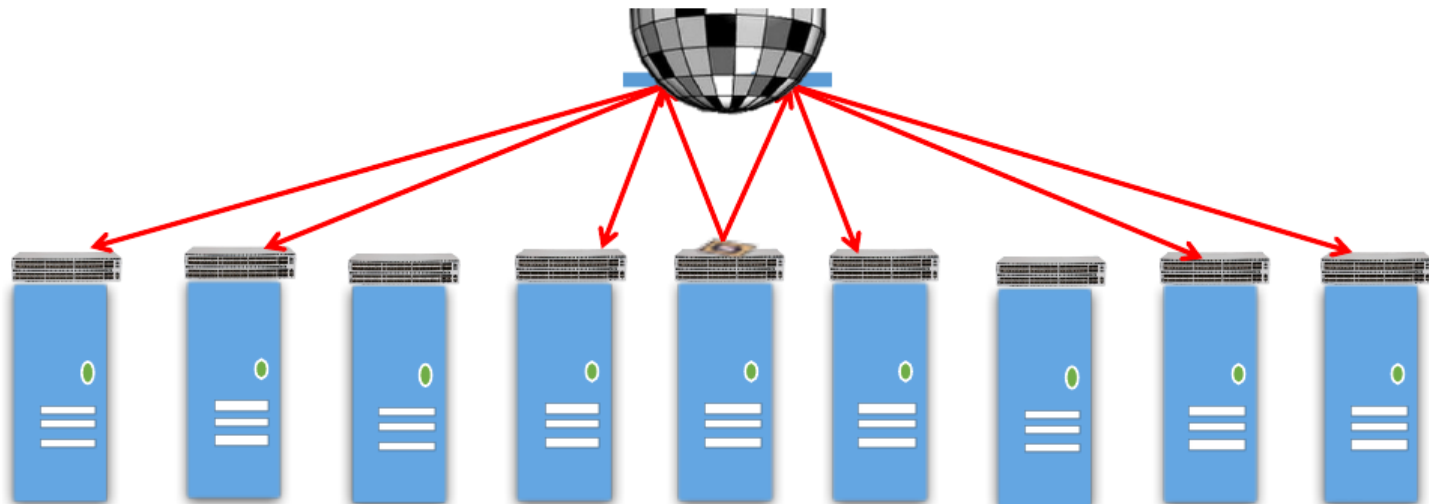
Static topology



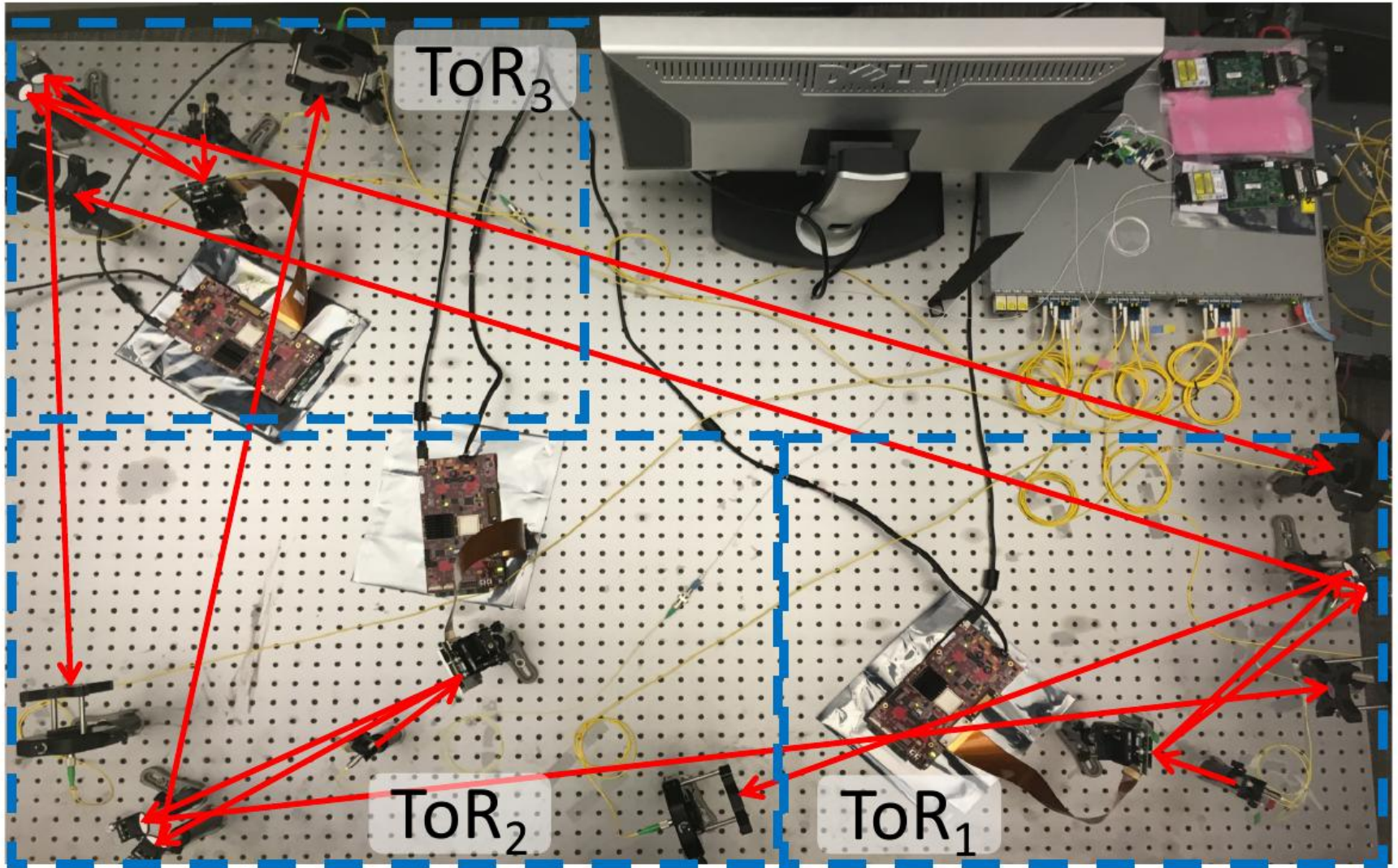
Array of micromirrors (10 μm)



Memory cell



Prototype: A 3-ToR ProjecToR interconnect

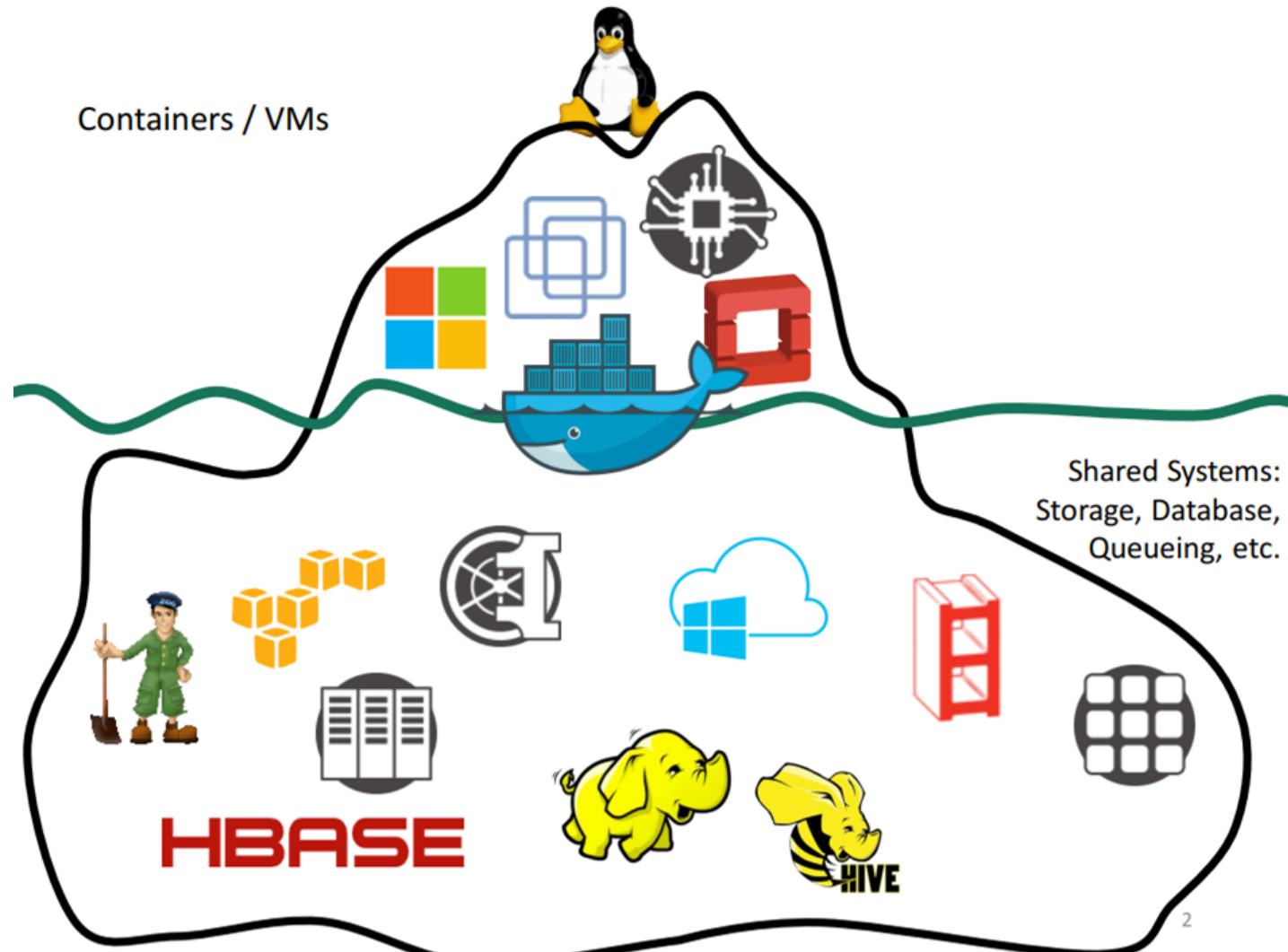


ProjecToR: Agile Reconfigurable Data Center Interconnect. Sigcomm 2016.

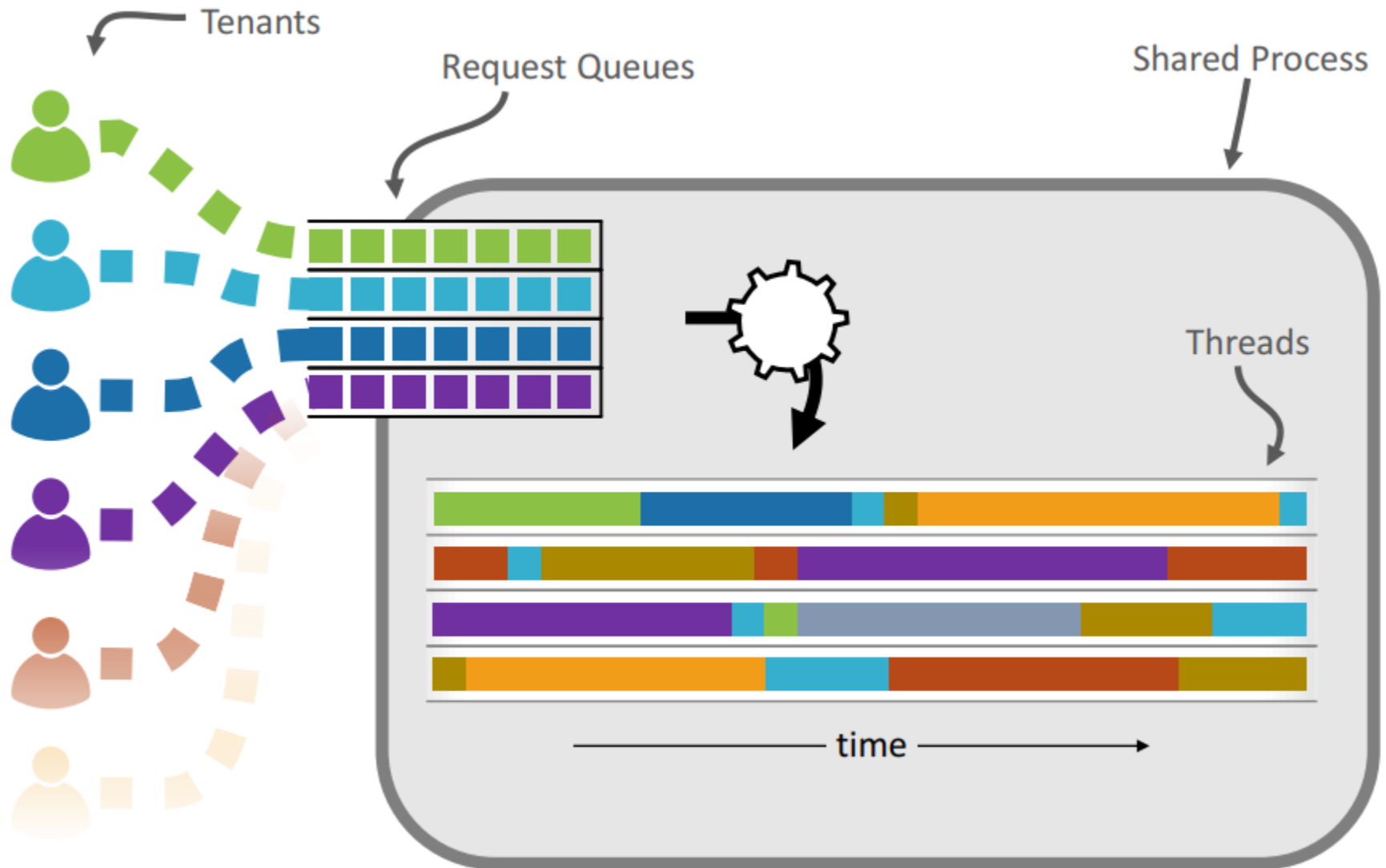
Планирование в ЦОД

- Распределение приложений по серверам
 - Выбор группы серверов внутри ЦОД
 - Разделение процессорного времени
 - Квотирование жёсткого диска
- Управление передачей пакетов
 - Распределение пропускной способности
 - Обработка пакетов на серверах приложений

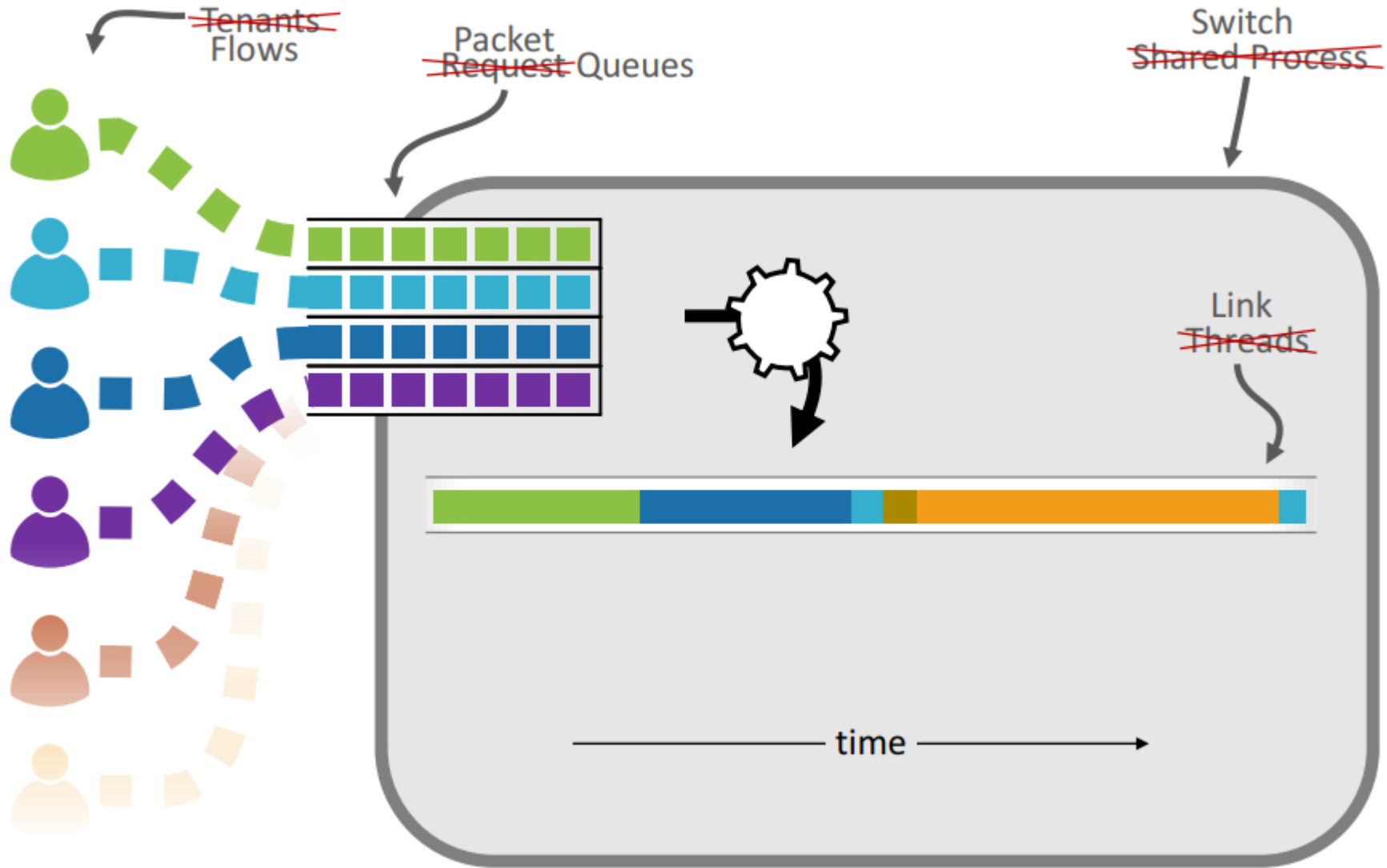
Иерархия облачных приложений



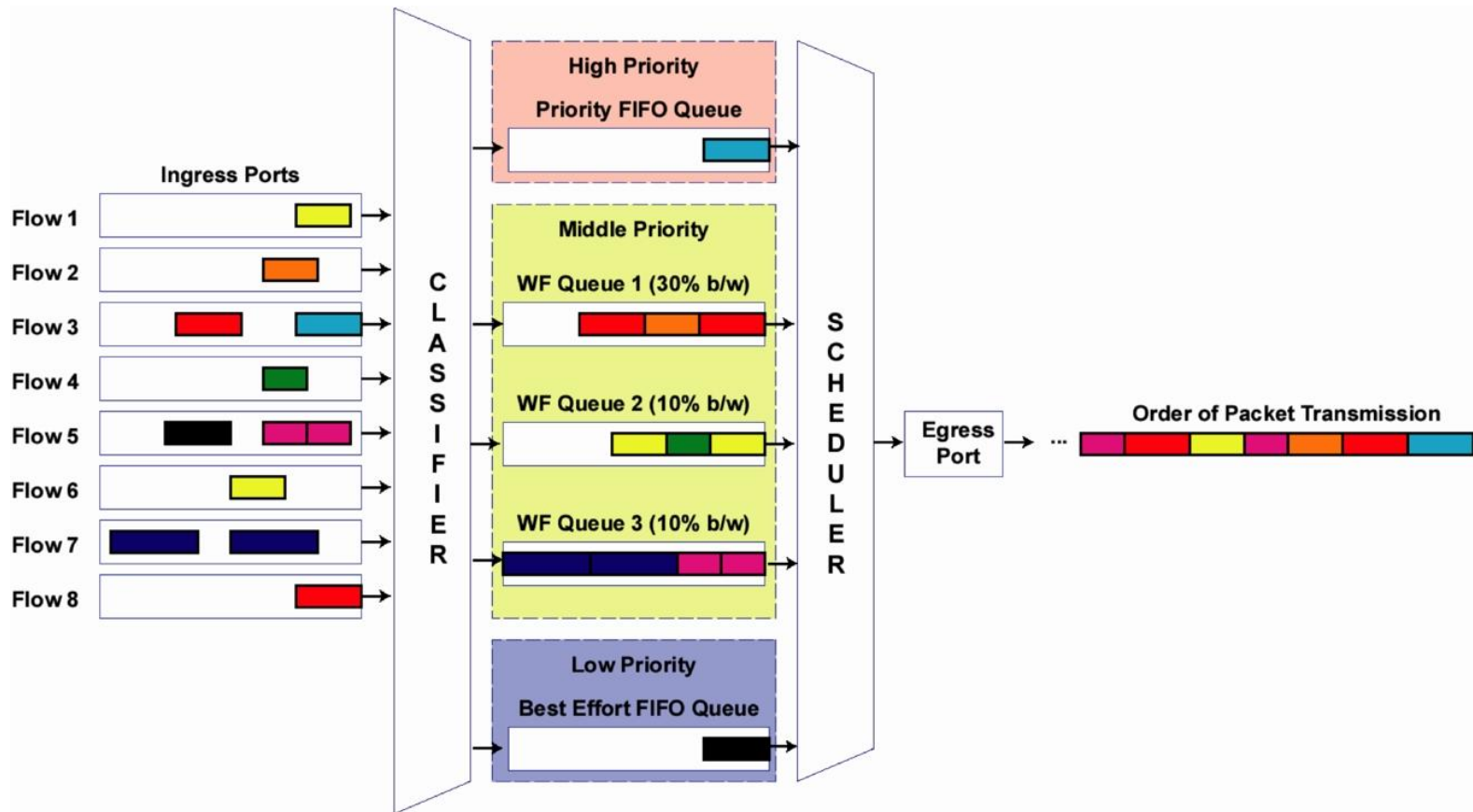
Инфраструктурные приложения обслуживают запросы от тенантов



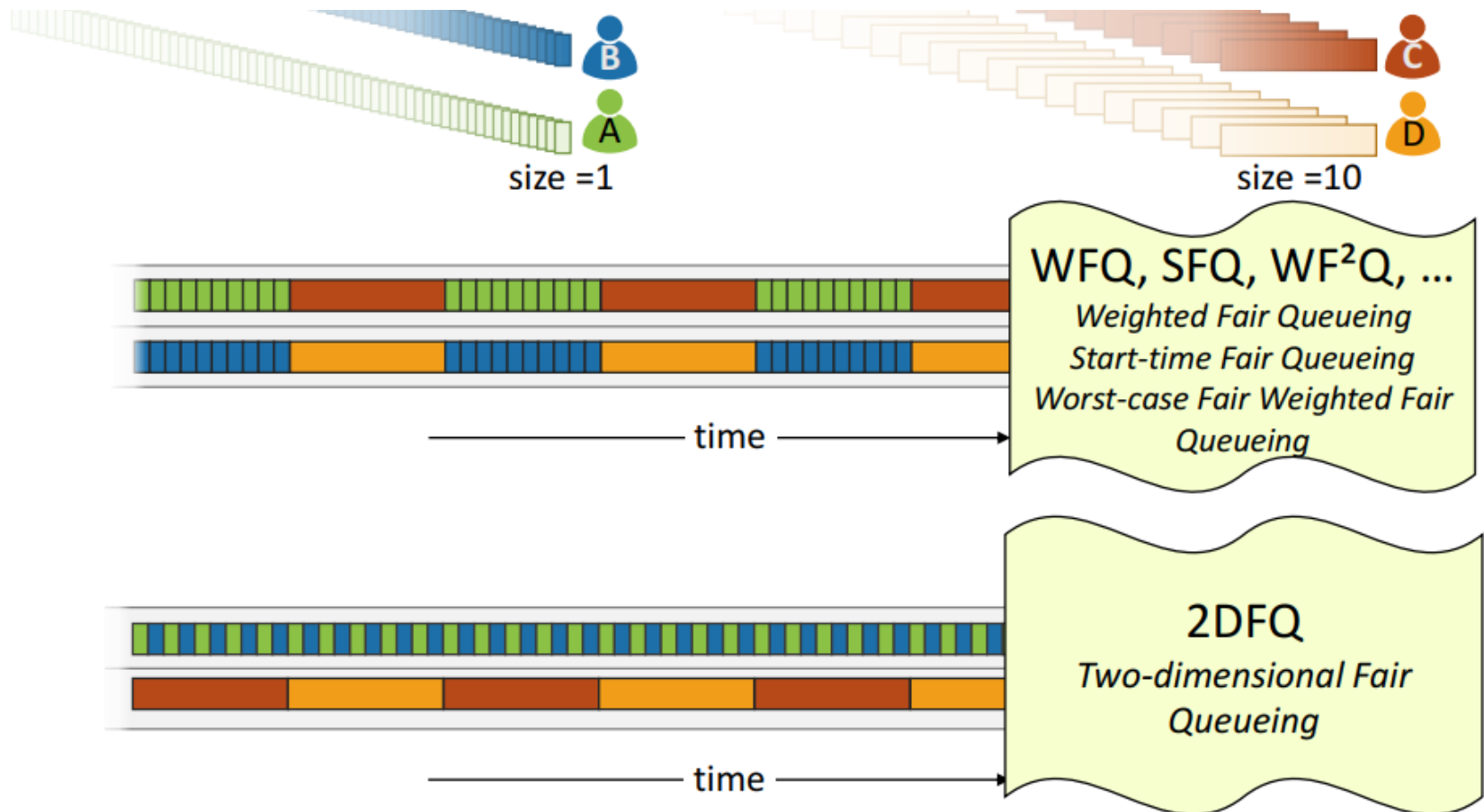
Проблема распределения ресурсов похожа на планирование пакетов



Дисциплины обслуживания и их устройство на коммутаторах

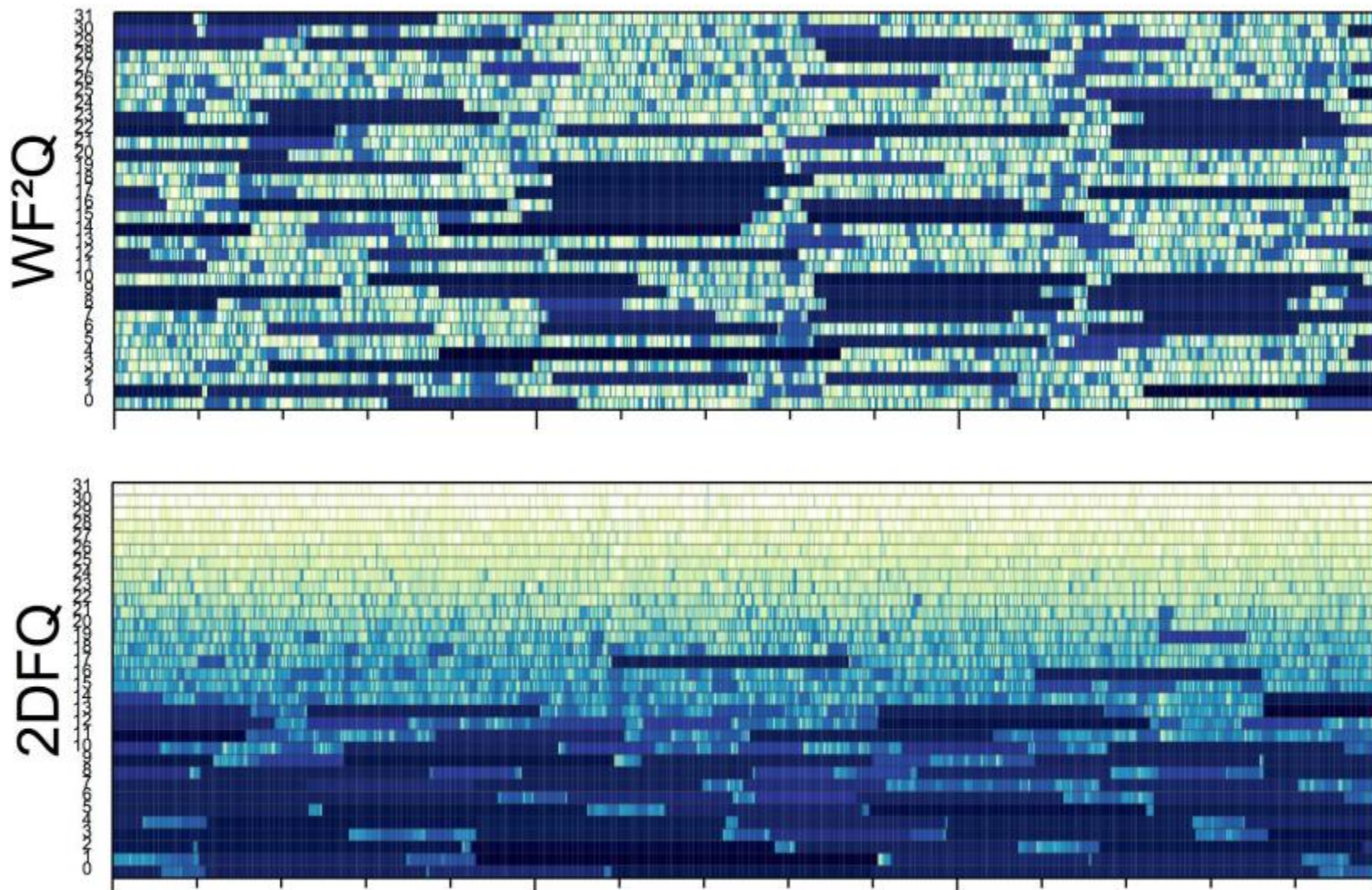


Алгоритмы планирования потоков для приложений ЦОД

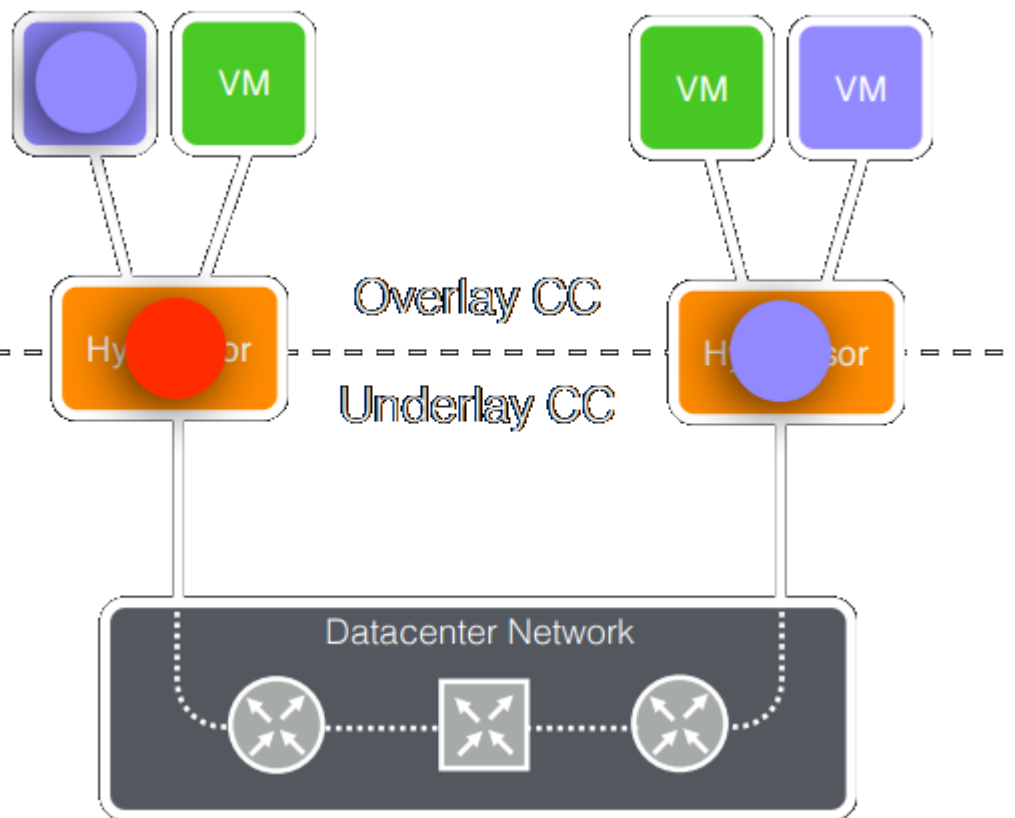


Построение таких расписаний, которые не только бы распределяли нагрузку, но и позволяли бы достичь наибольшей равномерности распределения

Распределение нагрузки с 2DFQ



Оптимизация приложений под ЦОД



В ЦОД используются более эффективные алгоритмы управления перегрузкой TCP (DCTCP, MCP, etc)

Как подменить протокол перегрузки, который использует приложение на тот протокол, который используется в ЦОД?

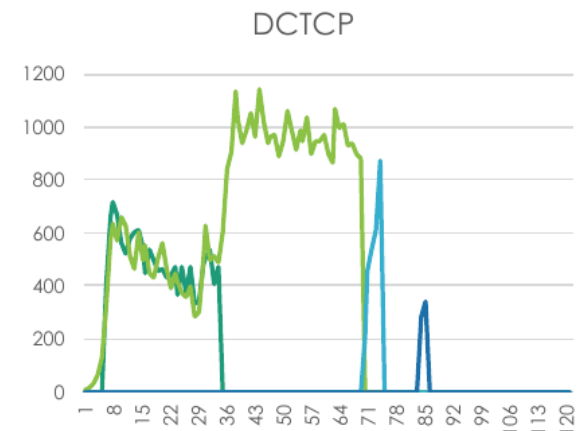
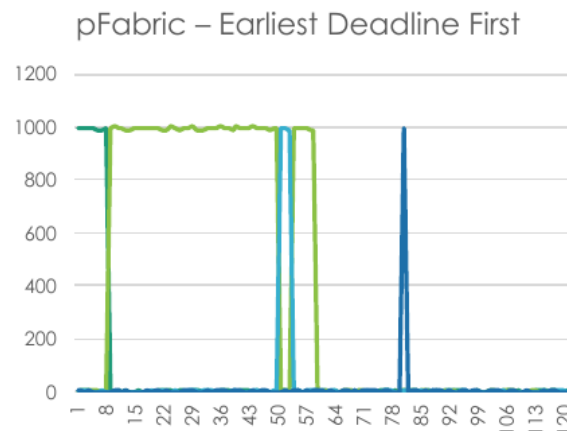
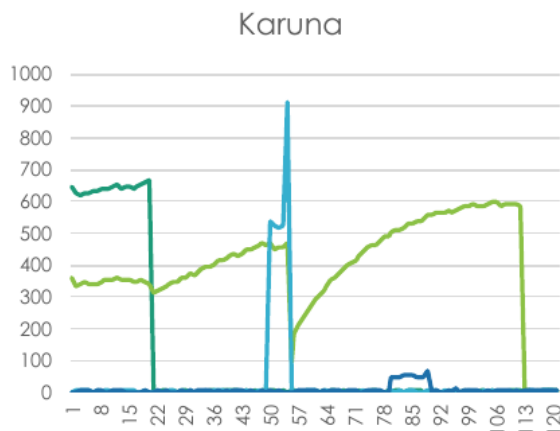
Планирование потоков данных

- Потоки без жёстких директивных интервалов
 - Резервное копирование
- Потоки реального времени
 - Map-reduce
- Как распределять пропускные способности каналов между потоками?
 - Использовать для разных типов потоков разные алгоритмы управления перегрузкой TCP
 - Вычислять функцию изменения congestion window в зависимости от директивного интервала
 - Какие эвристики использовать для вычисления функции управления перегрузкой?

Karuna

ЦОД – место для прокрастинации

Flow	Size	Deadline	Start Time
1	14.4MB	20ms	0ms
2	48MB	120ms	0ms
3	3MB	5ms	50ms
4	0.5MB	10ms	80ms

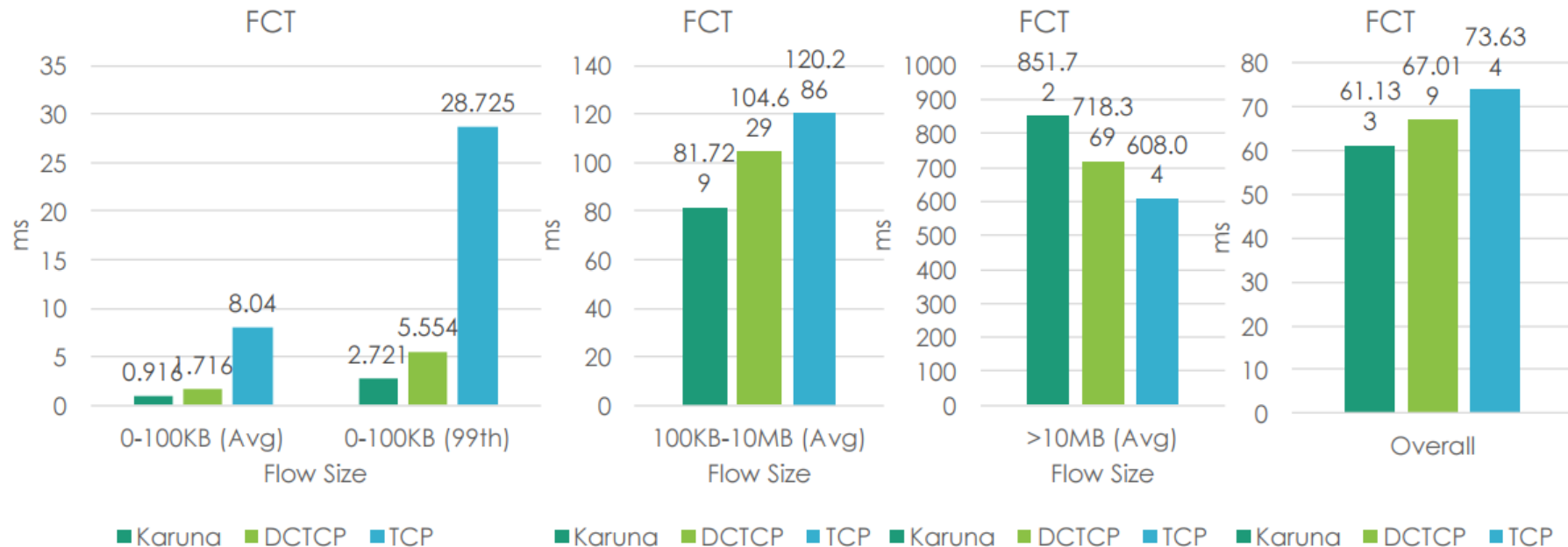


Каруна стремится передать данные прямо перед наступлением дедлайна, оставляя больше ресурсов для потоков без жёстких ограничений

Scheduling Mix-flows in Commodity Datacenters with Karuna. Sigcomm 2016.

Karuna

ЦОД – место для прокрастинации



Karuna стремится передать данные прямо перед наступлением дедлайна, оставляя больше ресурсов для потоков без жёстких ограничений

Scheduling Mix-flows in Commodity Datacenters with Karuna. Sigcomm 2016.

Связанные потоки (coflows)

- Выполнение задач часто требует завершения сразу нескольких потоков
 - Map-Reduce
 - Бессмысленно ускорять отдельные потоки, если приложение всё равно не завершится
- Планирование связанных потоков может существенно ускорить работу ЦОД
 - Как идентифицировать связанные потоки?

Coflow
HotNets'12

Baraat
SIGCOMM'14

Varys
SIGCOMM'14

Rapier
Infocom'14

Aalo
SIGCOMM'15



Assumption: all distributed data-parallel applications
have to be modified to correctly use the same coflow API

- **Difficulty 1:** Enable current Coflow API requires intrusive refactoring
- **Difficulty 2:** Hard to modify all applications and keep them up to date

*Can we automatically identify and schedule coflows without
manually modifying any data-parallel applications?*

Резюме

- Несмотря на огромную мощность современных ЦОД существует большой простор для увеличения их эффективности
- Основная проблема ЦОД – внутренние инфраструктурные ограничения
- Концепция resource pooling стремиться преодолеть эти ограничения, представляя вычислительные ресурсы ЦОД в виде однородного множества
- Поддержание такой абстракции требует преодоления множества трудностей