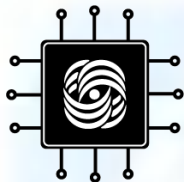


Планирование вычислений в сетевых вычислительных системах с ограничениями на время выполнения задач

Кафедра АСВК

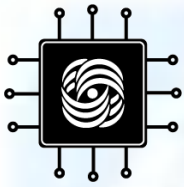
Балашов Василий Викторович (hbd@cs.msu.ru)

Костенко Валерий Алексеевич (kostmsu@gmail.com)

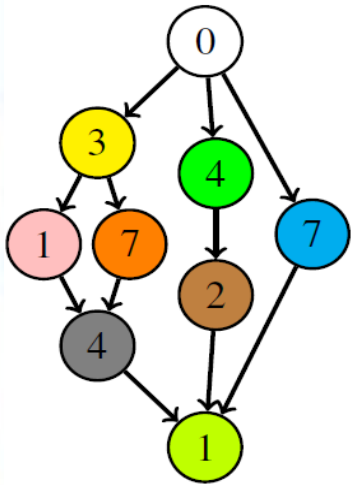


Аннотация

- Планирование вычислений – что это такое, где и зачем нужно
- Примеры задач планирования вычислений – в т.ч. при чем здесь сети
- Что предлагает кафедра АСВК по данной тематике
 - как устроена курсовая работа
 - направления исследований
 - поддержка в учебных курсах
- Профессиональные перспективы

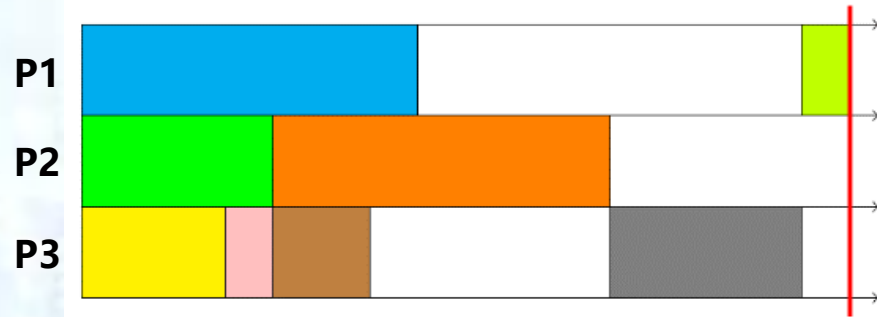


Что такое планирование вычислений?



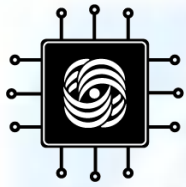
Граф зависимостей работ по данным

Планирование



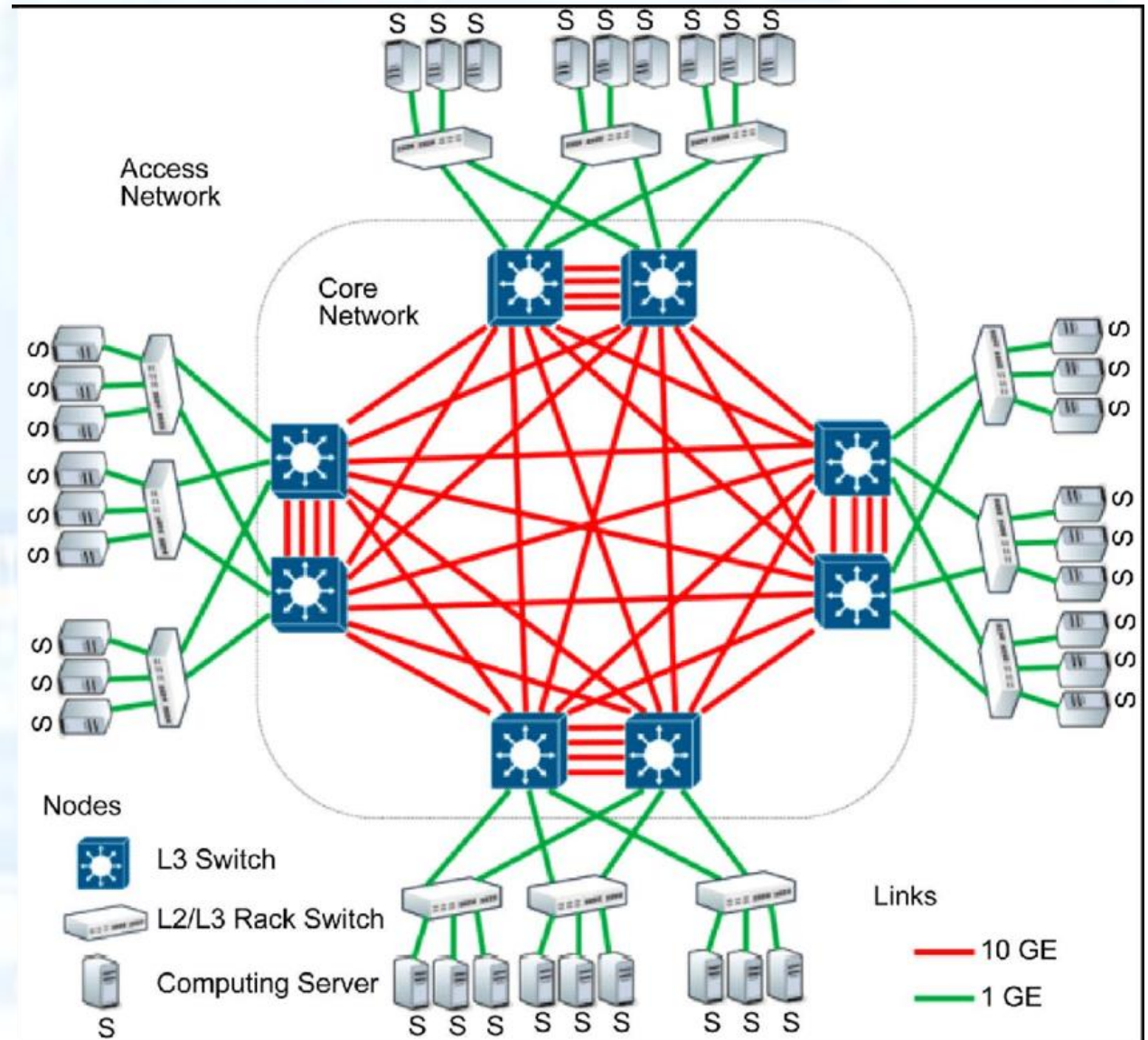
Расписание выполнения работ на трех процессорах

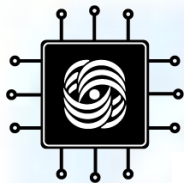
- Распределить работы по процессорам
- Определить последовательность выполнения работ на каждом процессоре
- Определить времена старта работ
 - учет зависимостей: работа стартует, когда все входные данные готовы (или позже)
- «Хорошее» расписание:
 - Самое короткое
 - Соблюдены все директивные сроки
 - ... (зависит от области применения)



Где нужно планирование вычислений?

Центр
обработки
данных
(динамическая
нагрузка)





Где нужно планирование вычислений?

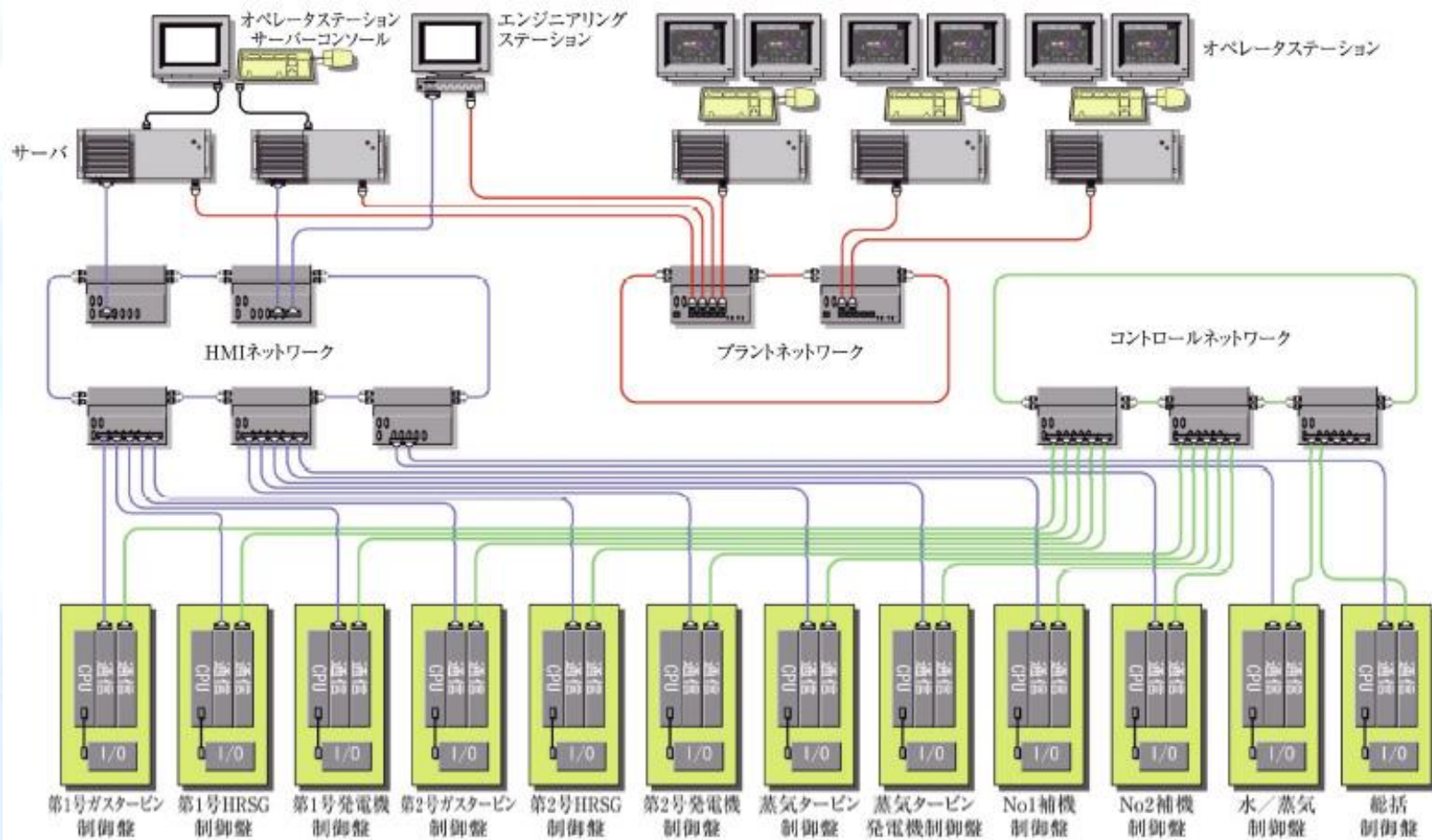
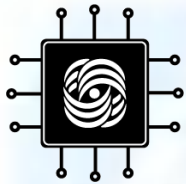


図2 中型発電制御装置のシステム構成例

Fig. 2 System configuration of medium class gas turbine power plant

Управление технологическими процессами
(нагрузка заранее известна)

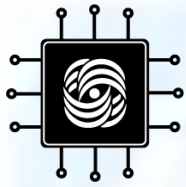


Где нужно планирование вычислений?

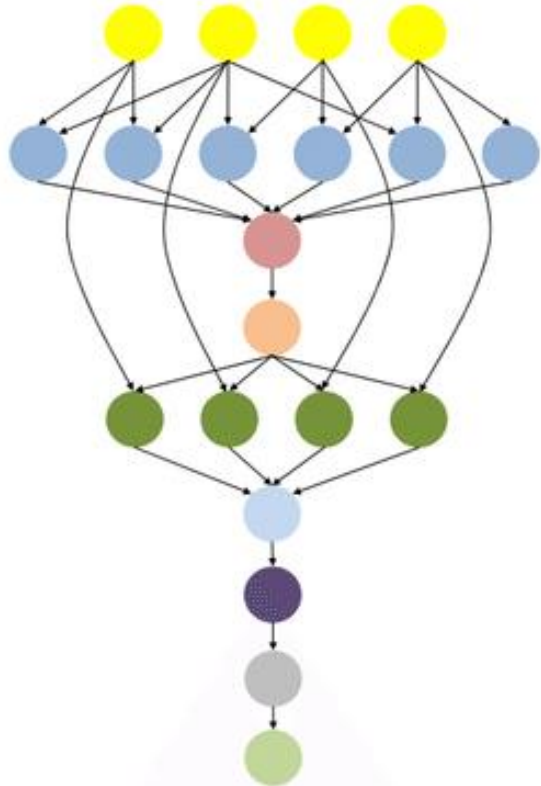


Базовая станция 5G:

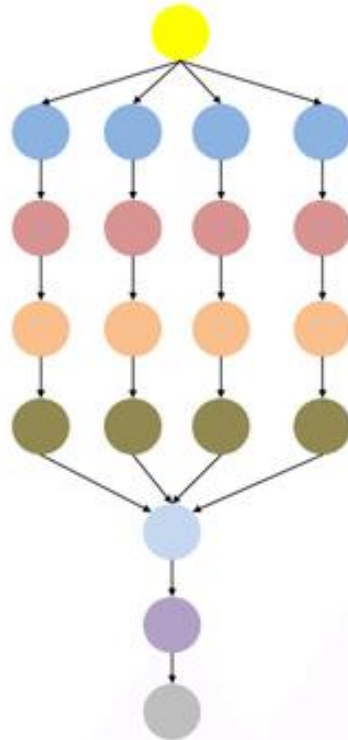
- Параллельная обработка трафика на гигабитных скоростях
- разные категории трафика



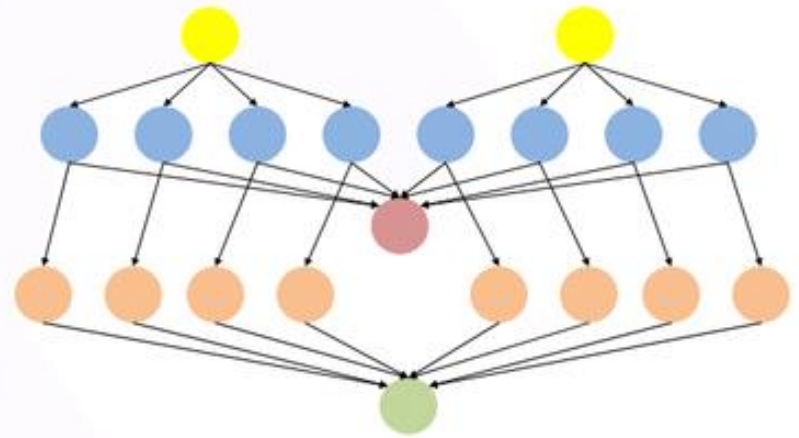
Примеры графов работ



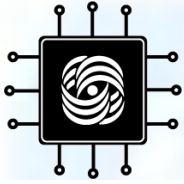
Montage.



Epigenomics.



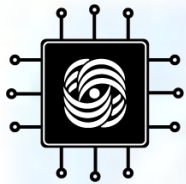
CyberShake.



Примеры графов работ

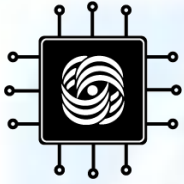


Цифровая обработка сигналов
(QR-факторизация)



Пример задачи

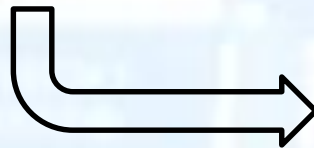
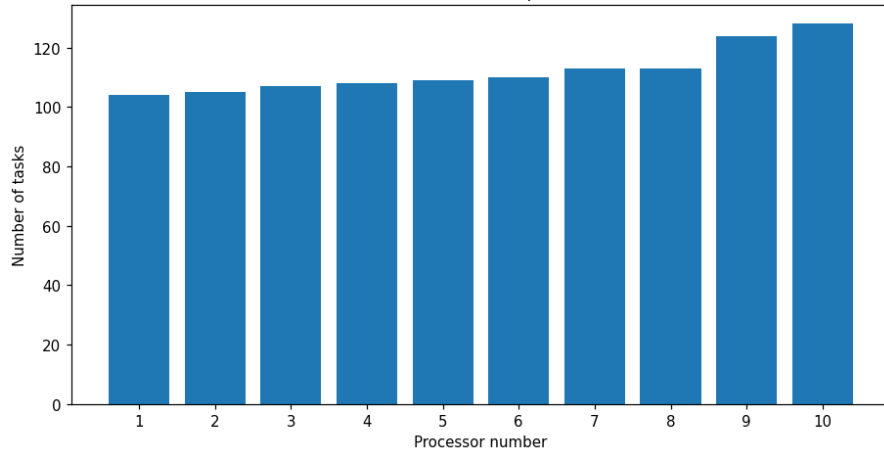
- Дано:
 - граф зависимостей работ
 - длительности работ
 - число процессоров
(процессоры одинаковые, каждый связан с каждым, длительности передачи данных одинаковые для каждой пары процессоров)
- Требуется: построить расписание минимальной длительности



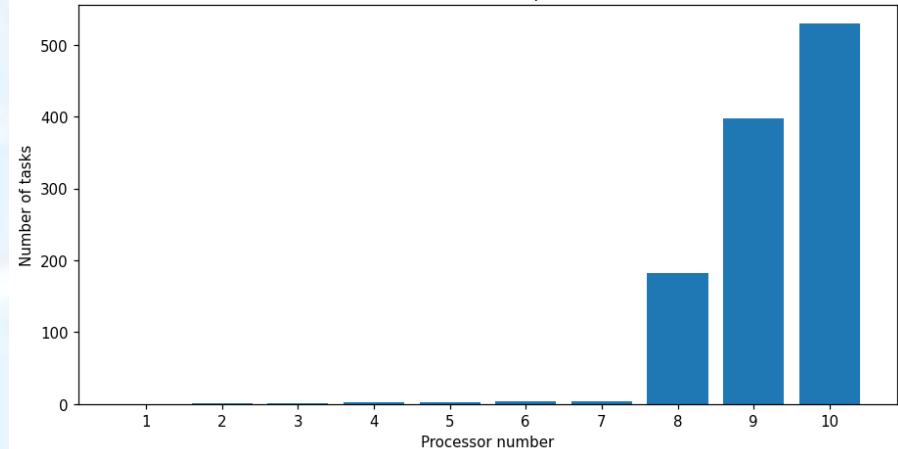
Усложняем задачу...

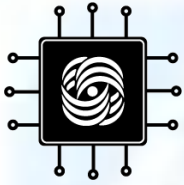
- Доля межпроцессорных передач данных не более 40%

(DAG_B, bad CR)
Number of tasks on processor



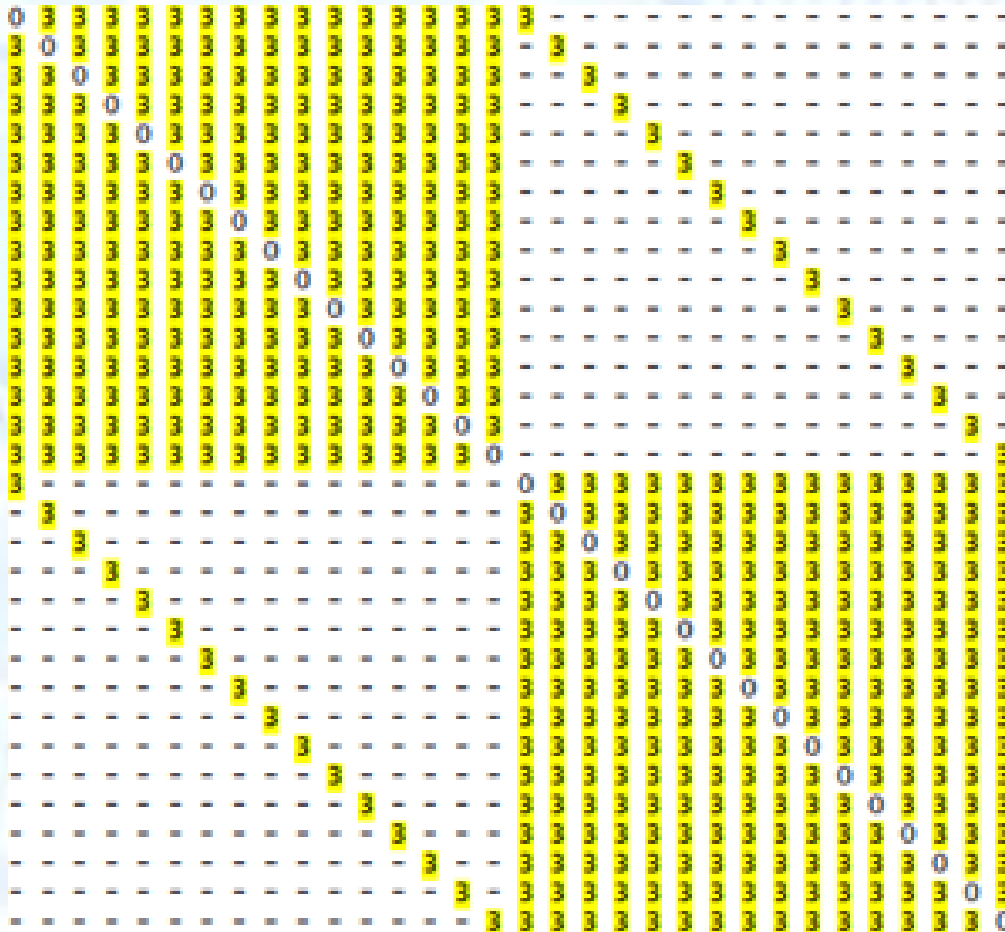
(DAG_B, good CR)
Number of tasks on processor

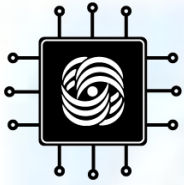




Усложняем задачу...

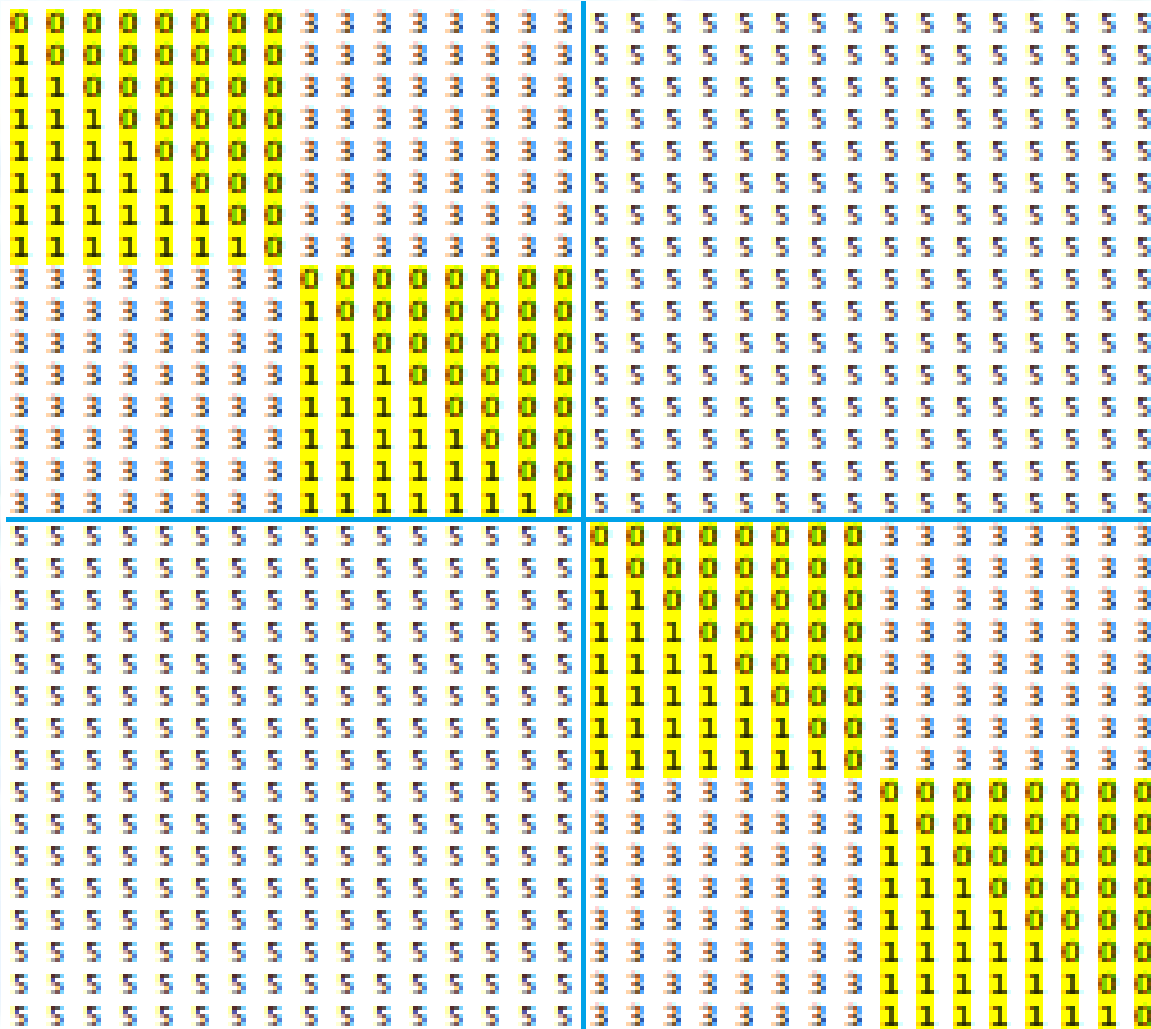
- Неполносвязные процессоры
- Доля «двухзвенных» передач данных не выше 5%

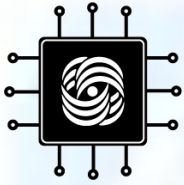




Усложняем задачу...

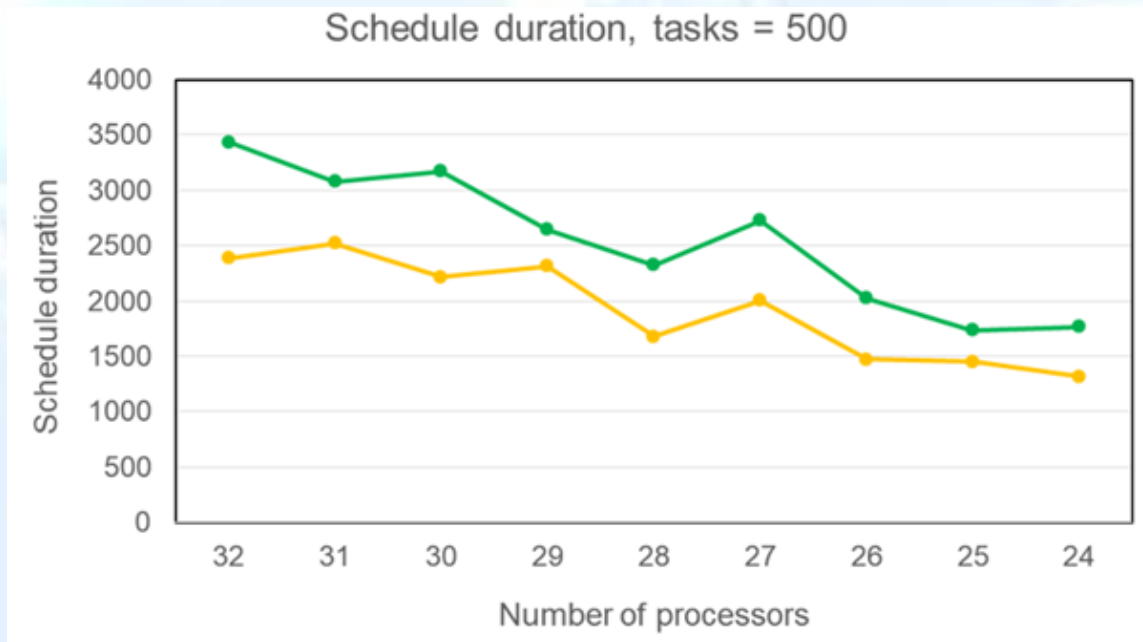
- Разные длительности передачи данных между процессорами



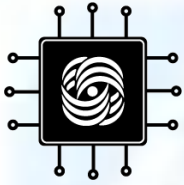


Усложняем задачу...

- Разные типы процессоров => длительность выполнения работы зависит от процессора
- Пример:
 - 4 типа процессоров
 - длительность работы: x_1 , $x_{1.5}$, x_2 , $x_{2.5}$
 - в системе 32 процессора, по 8 каждого типа
- Все ли процессоры нужно использовать?

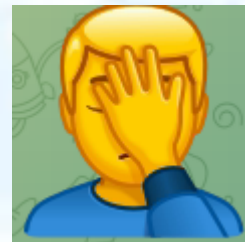


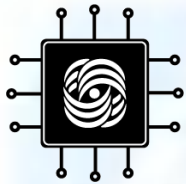
- По-одному убираем самые медленные процессоры



Усложняем задачу...

- Энергоэффективное планирование:
 - Процессор может динамически менять частоту и напряжение питания
 - Частоту и напряжение уменьшили вдвое =>
 - Такт процессора стал вдвое *дольше*, но требует вдвое *меньше* энергии
 - Длительность работы увеличилась вдвое
 - Затраты энергии на выполнение работы уменьшились вдвое
 - То есть даже на одном и том же процессоре работа может выполняться разное время...
- Новые задачи:
 - Построить расписание, потребляющее минимум энергии, при условии что оно не превышает заданную длительность
 - Построить расписание наименьшей длительности, при условии что для его выполнения требуется не более чем заданное количество энергии
- А теперь представим себе, что частоту процессора можно менять *по ходу* выполнения работы...



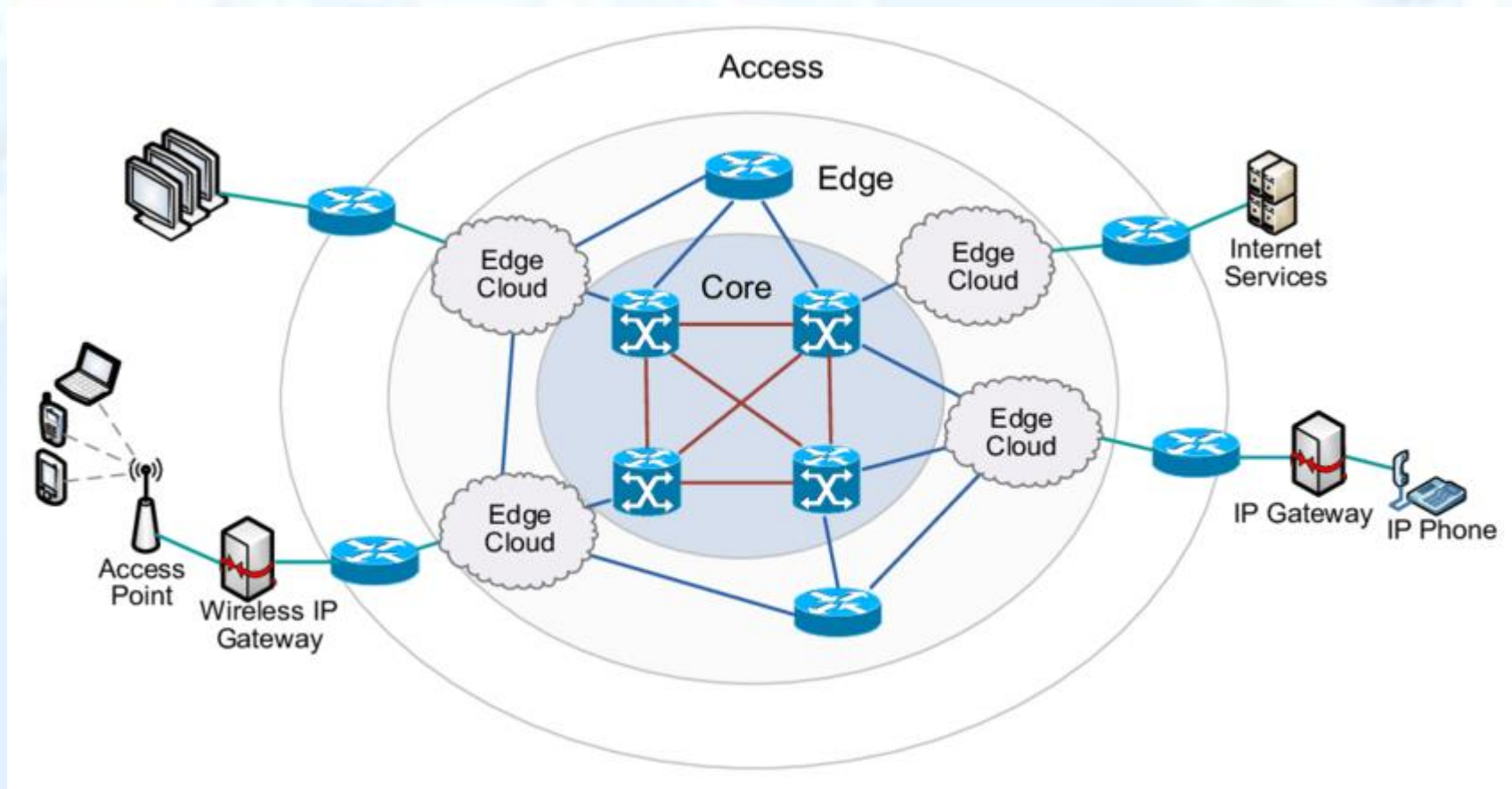


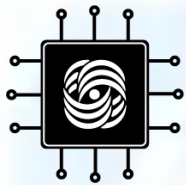
Вспоминаем про сеть

- Процессоры сгруппированы в серверы
 - даже внутри одного сервера, длительность передачи данных между процессорами зависит от объема данных
- Серверы объединены сетью, состоящей из коммутаторов и каналов
- Необходимо определить *маршруты* передачи данных
- При передаче данных могут быть *конфликты* за доступ к каналу
- Сеть работает в соответствии с набором протоколов, позволяющих ограничить конфликты за счет:
 - Расписания передачи данных
 - Использования виртуальных каналов
 - ...
- => задача *совместного планирования* вычислений и передачи данных



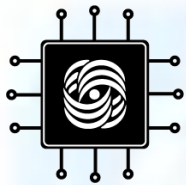
Многоагентное планирование для Edge Cloud





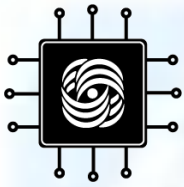
Многоагентное планирование для Edge Cloud

- Работы поступают на Е.С. от устройств Интернета вещей (IoT)
 - работа = запрос на выполнение сетевого сервиса
 - в запрос входит пакет данных
 - программный код сервиса есть на всех Е.С.
- Работа поступает на Е.С., ближайший к устройству IoT
- Все Е.С. связаны между собой сетью
- У разных Е.С.:
 - разная производительность и энергопотребление процессоров
 - поддержка разных функций энергосбережения
 - разная доступность «зеленой» энергии (солнечные батареи => уровень освещенности, и т.п.)



Многоагентное планирование для Edge Cloud

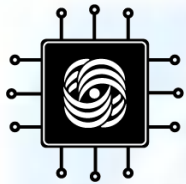
- Требуется: выполнить все работы в рамках временных ограничений (заданных SLA для сервисов), минимизируя потребление «не-зеленой» энергии (углеродный след)
- Е.С. должны договориться о распределении поступающих работ между собой, с учетом:
 - работ, уже принятых для выполнения
 - задержек передачи данных между Е.С., с учетом текущей нагрузки на сеть
 - доступности «зеленой» энергии
- => Многоагентный подход.
- На каждом узле необходимо построить энергоэффективное расписание
- Вычислительная нагрузка и нагрузка на сеть динамически меняется => планируем использовать машинное обучение (подробности в докладе Е.П. Степанова через неделю)



Сочетание математики и программирования

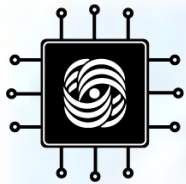
Курсовая работа:

- Практическая задача, возникающая при разработке реальных систем
 - задачи от индустриальных партнеров
- Математическая формулировка задачи
- Обзор известных методов решения похожих задач
- Построение алгоритма решения задачи
- Исследование алгоритма
- Реализация программной системы



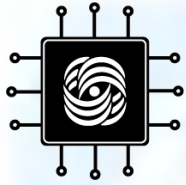
Направления исследований

- Совместное планирование вычислений и конфигурирование сети передачи данных
- Многоагентное планирование с использованием машинного обучения
- Планирование в вычислительных системах большого масштаба
 - сотни и тысячи процессорных ядер
 - сотни тысяч задач
- Энергоэффективное планирование вычислений
- Планирование вычислительных ресурсов в сетевых коммутаторах



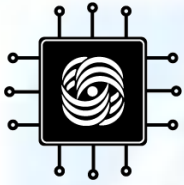
Поддержка в учебных курсах

- Архитектура сетевых устройств
- Планирование вычислений в распределённых системах
- Основы анализа и построения управляющих систем реального времени
- Имитационное моделирование в исследовании и разработке информационных систем
- Групповая разработка приложений на Python
 - управление версиями исходного кода (git)
 - организация групповой разработки вокруг git
 - документирование кода, подготовка дистрибутивов, ...



Профессиональные перспективы

- Разработка управляющих систем для
 - средств телекоммуникации (5G, Интернет вещей, ...)
 - транспорта (наземного, водного, воздушного...), в т.ч. беспилотного
 - технологических процессов (энергетика, производственные линии, добыча полезных ископаемых, «умный» дом, ...)
- Разработка систем планирования ресурсов для ЦОД, облачных платформ, телекоммуникационных систем
- Применение алгоритмов оптимизации во многих других областях (задачи оптимизации *на удивление* похожи)
- Профессиональные роли:
 - разработчик
 - алгоритмист
 - аналитик
 - архитектор



Спасибо за внимание!

Через неделю: Е.П. Степанов с рассказом
о методах управления качеством сервиса
в сетях, в т.ч. с применением машинного
обучения