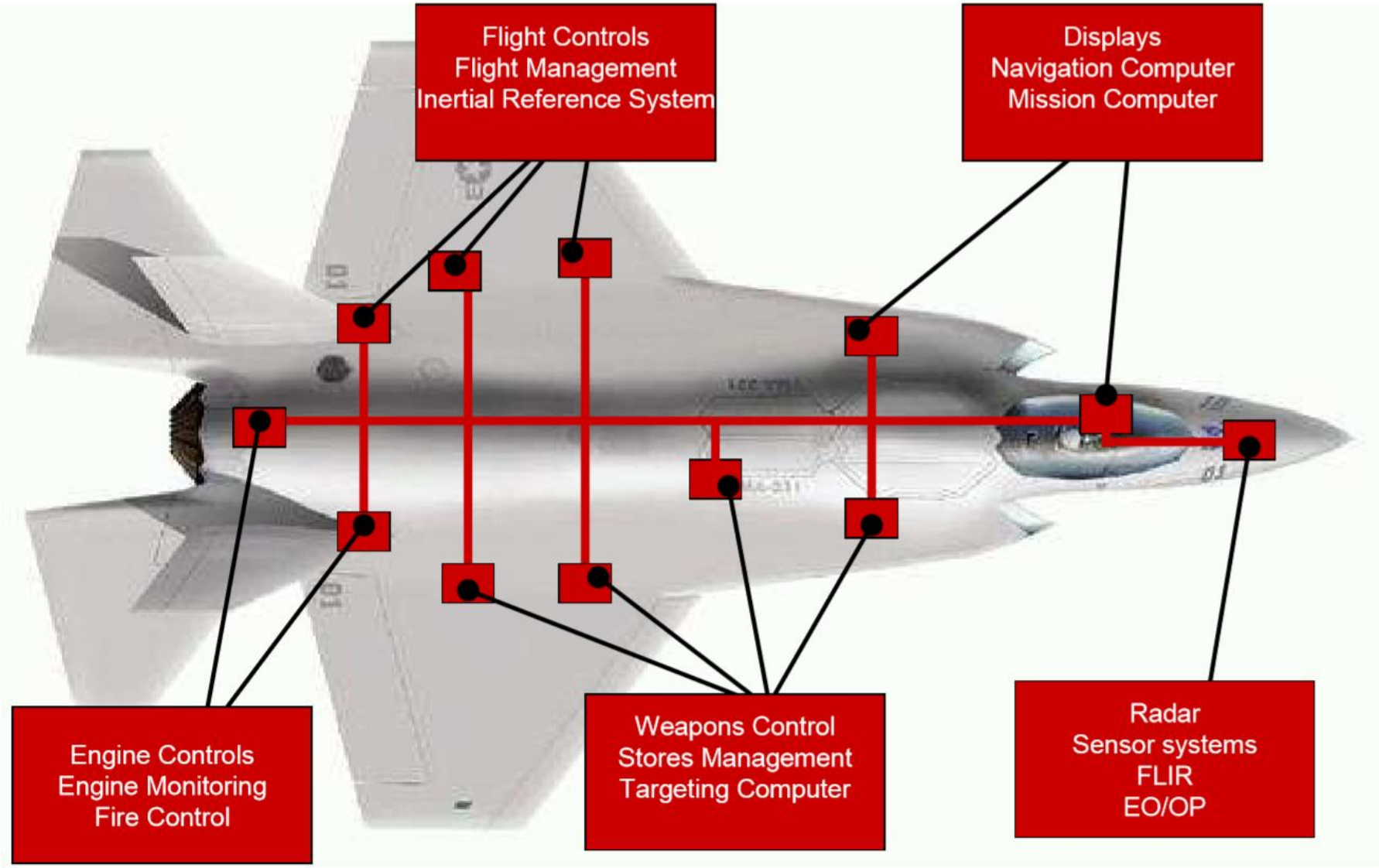
ИНФОРМАЦИОННО-УПРАВЛЯЮЩИЕ СИСТЕМЫ РЕАЛЬНОГО ВРЕМЕНИ

Лекция 10:

Сквозное планирование вычислений и информационного обмена на примере системы интегрированной модульной авионики

Кафедра АСВК,
Лаборатория Вычислительных Комплексов
Балашов В.В.

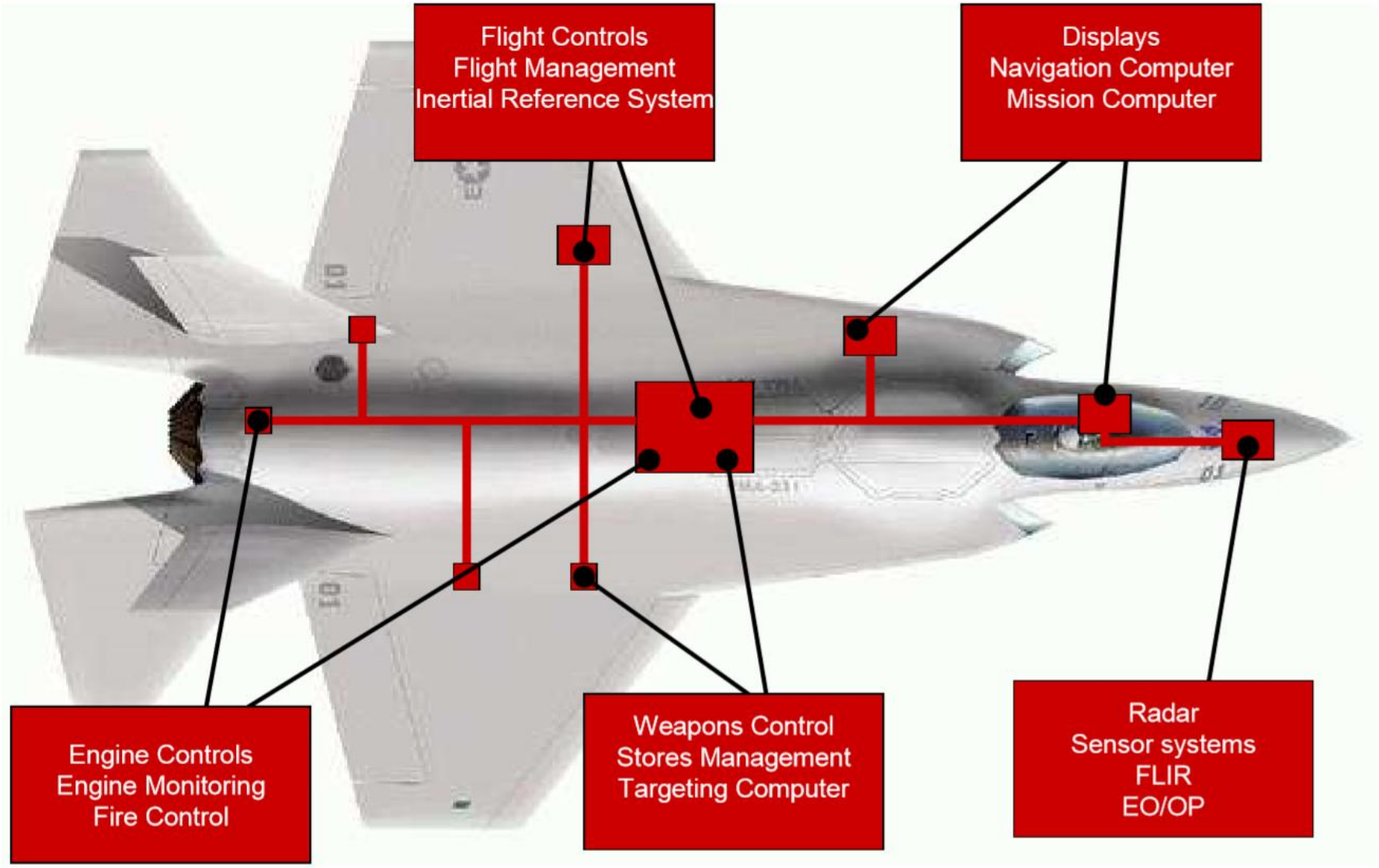
Федеративная ИУС РВ



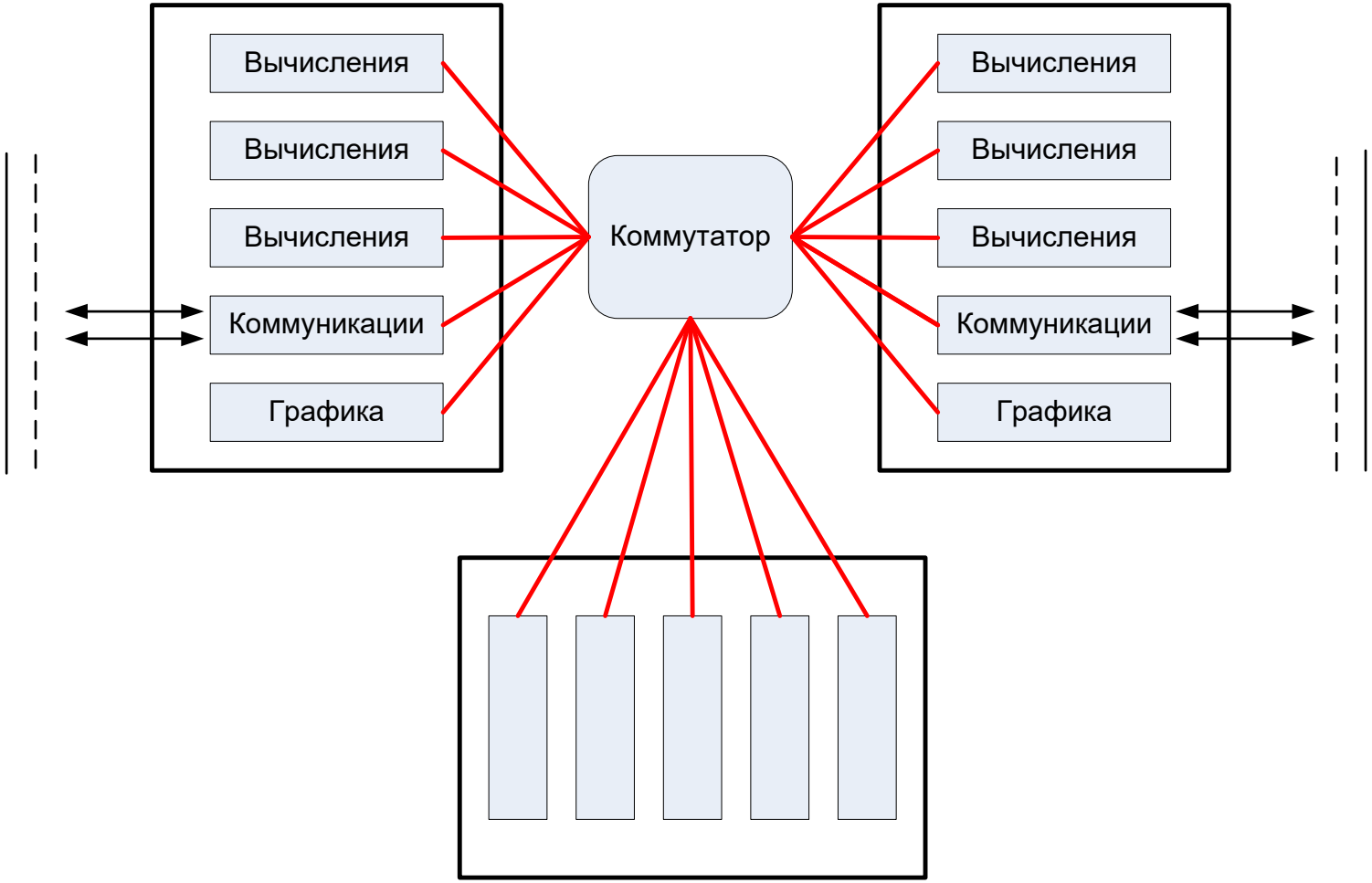
Федеративная ИУС РВ

- Специализированные блоки
  - по назначению
  - по архитектуре
- ПО различных подсистем – на различных блоках
  - изоляция по памяти
  - нет конкуренции подсистем за процессорное время
- Недостатки:
  - низкая переносимость и повторная используемость ПО
  - «зоопарк» процессорных архитектур и программных интерфейсов

ИУС РВ с архитектурой ИМА



ИУС РВ с архитектурой ИМА с сетью на базе коммутатора



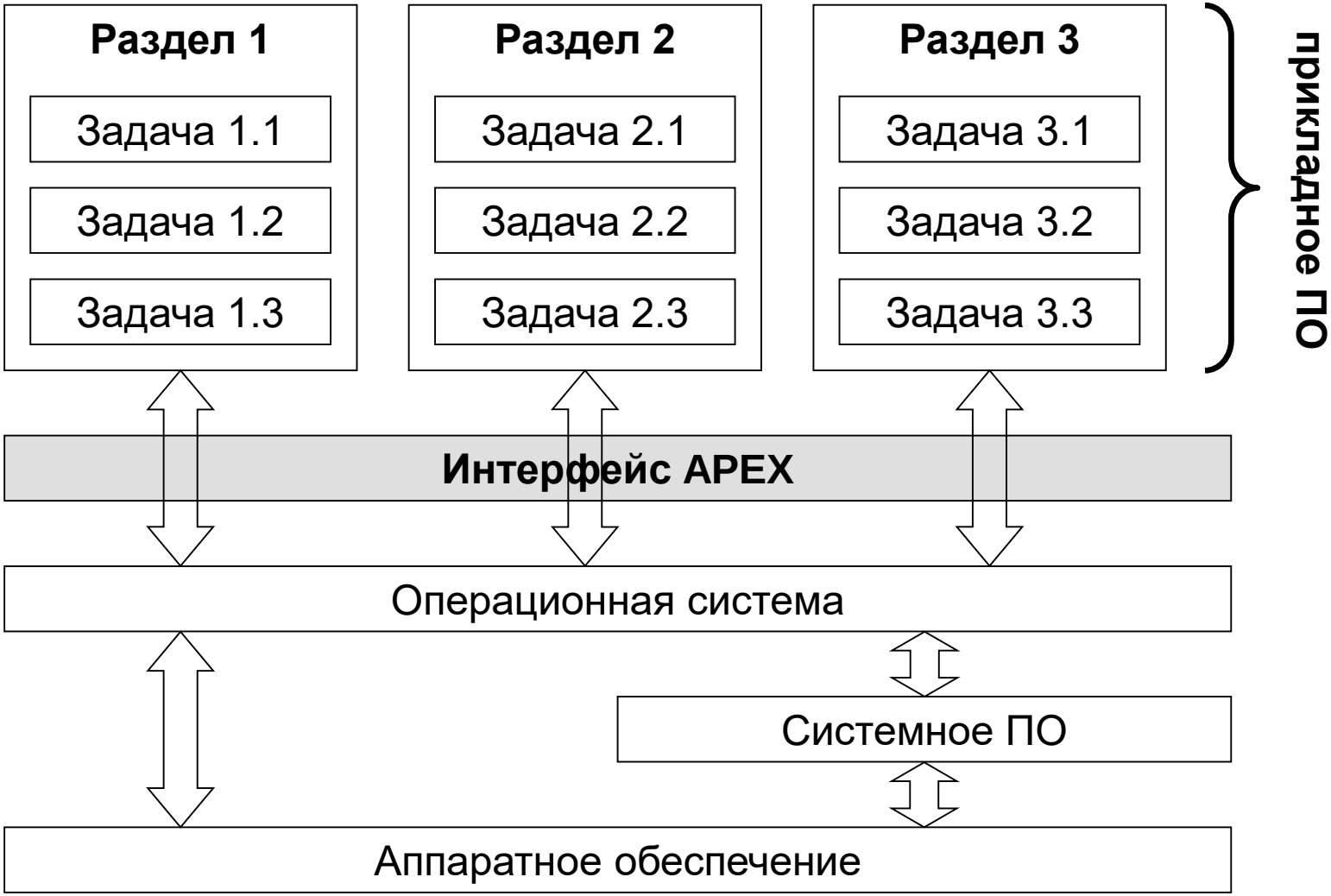
ИУС РВ с архитектурой ИМА

- Логически единый распределенный вычислитель
  - унифицированные модули нескольких типов
  - единая архитектура процессоров
  - унифицированный программный интерфейс
- Разделение вычислительных ресурсов между ПО различных подсистем
- Проблемы:
  - изоляция по памяти
  - разделение процессорного времени
- Решение: каждой подсистеме → раздел

ИМА и разработка ПО

- Особенности:
  - Стандартное API со стороны ОС
  - Статическое разделение времени, памяти и ресурсов
- Преимущества:
  - Надежность
  - Переносимость
  - Возможность повторного использования
  - Модульность
  - Упрощение верификации и сертификации

Структура ПО. Интерфейс APEX



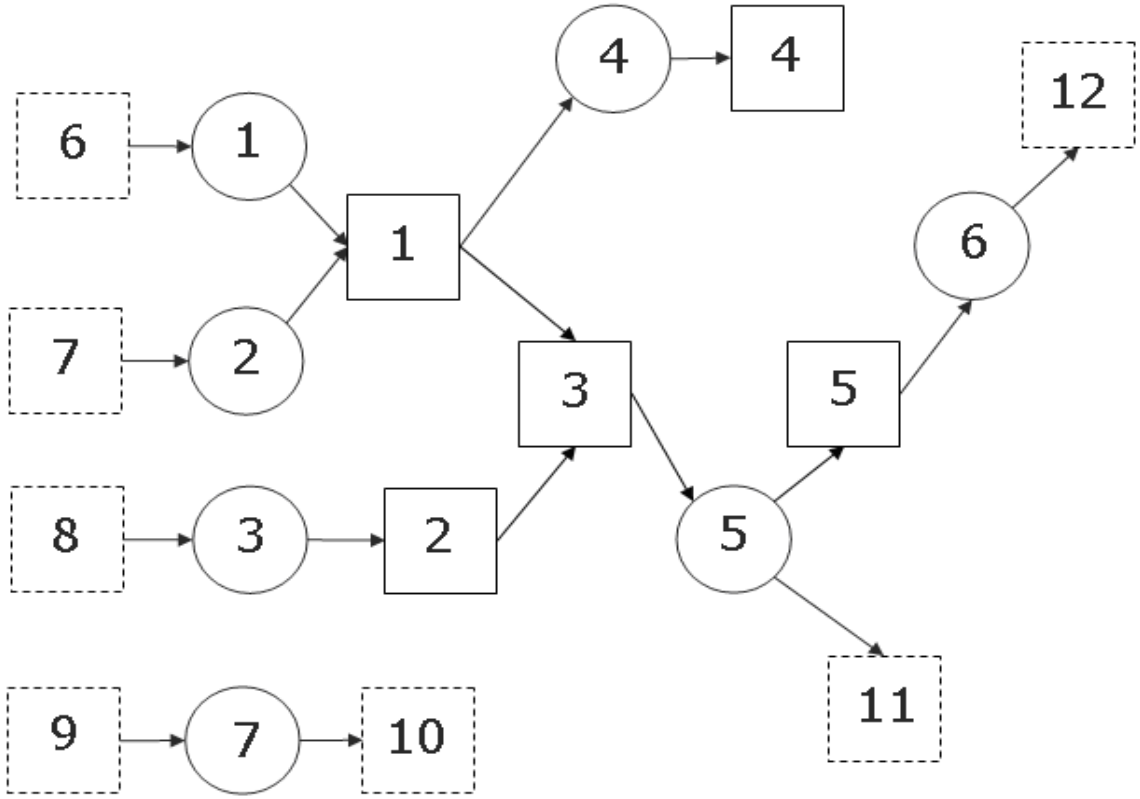
Взаимодействие между разделами

- Порты с очередью сообщений
- Порты с перезаписью сообщений
  - Буфер фиксированной длины
  - Сообщение в буфере перезаписывается
  - Отправка с заданным периодом или по факту обновления

Взаимодействие внутри разделов

- Обмен данными
  - Передача сообщений
  - Общая память
- Механизмы синхронизации
  - Семафоры
  - События

Рабочая нагрузка



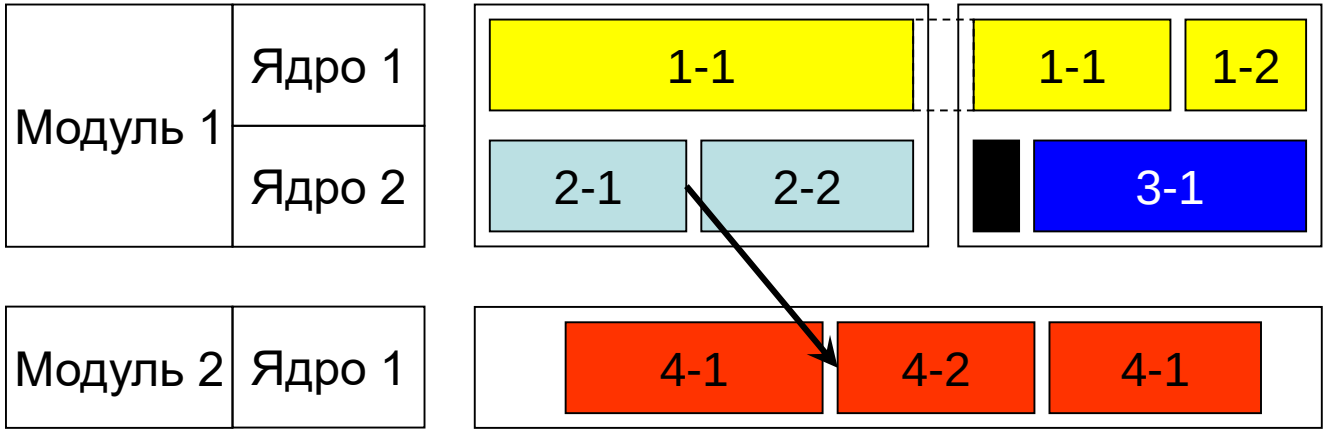
- периодические задачи
- сообщения
- зависимости по данным

Синхронизация на практике

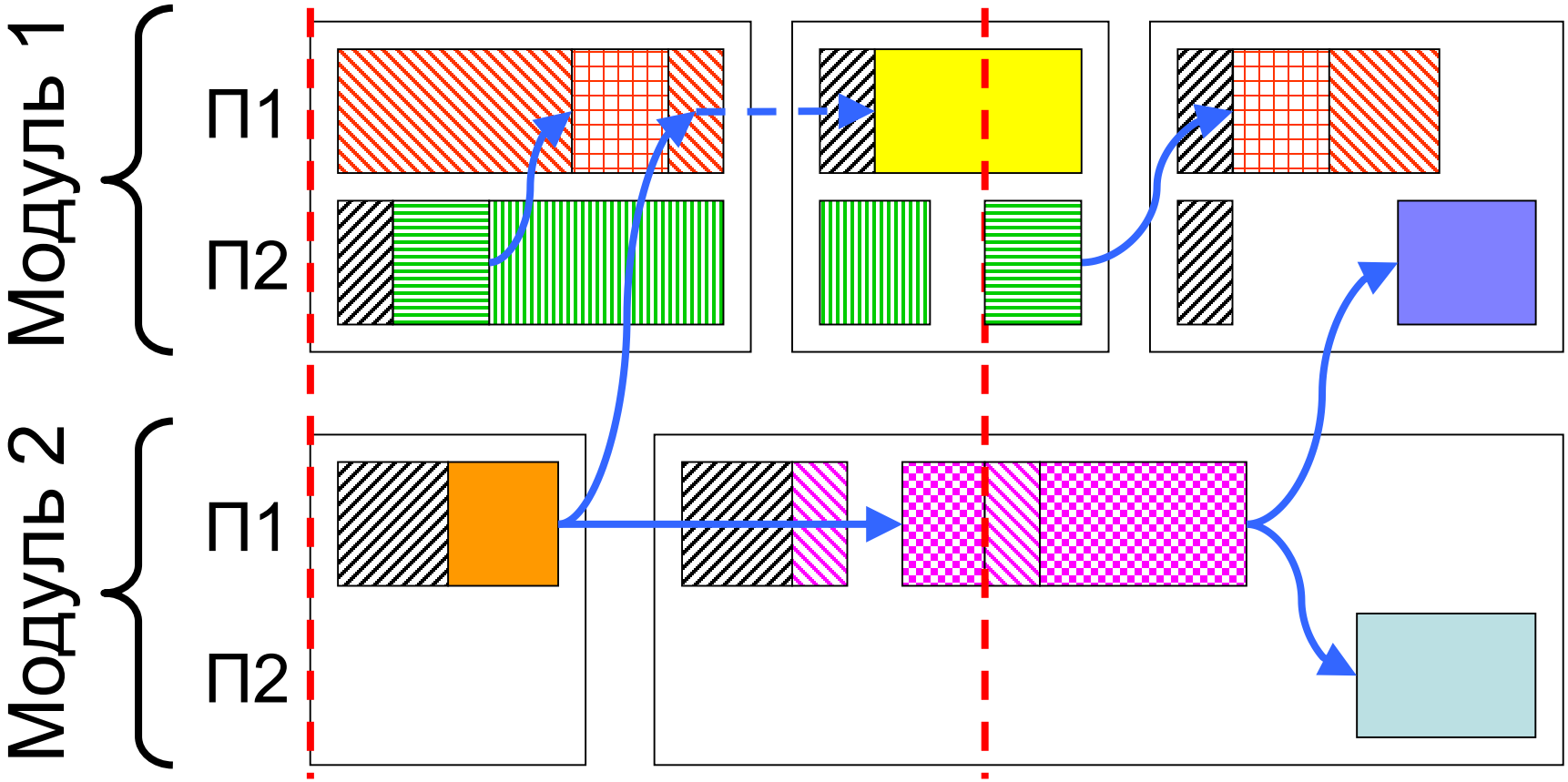
- Ожидание данных только от задач с той же частотой
  - «синхронная» зависимость по данным
- Отправитель выполняется чаще получателя:
  - на входе получателя почти всегда актуальные данные
  - ожидать нет смысла
- Отправитель выполняется реже получателя
  - ожидание нарушает требования по частоте для задачи-получателя

Выполнение задач в системе

- Задачи раздела выполняются в рамках окон
  - статическое расписание окон
  - границы окон одинаковы для всех ядер одного модуля
- Между разделами происходит переключение контекста
- Работы в окне: динамическое планирование
  - очередь выполнения
  - приоритеты
  - вытеснение
  - ожидание входных данных
- Незавершенная работа может быть возобновлена в следующем окне



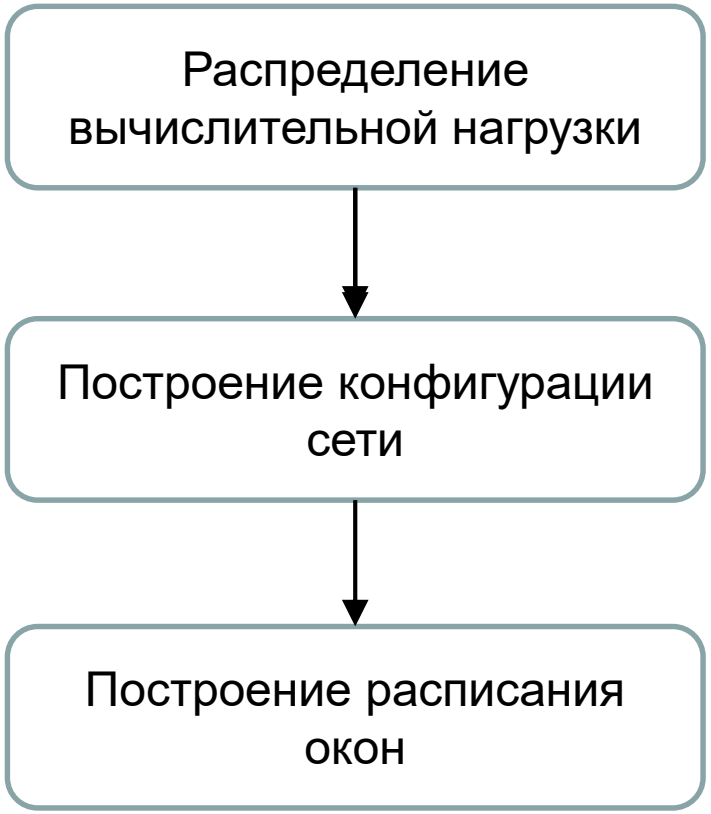
Пример расписания



Сквозное планирование вычислений и информационного обмена

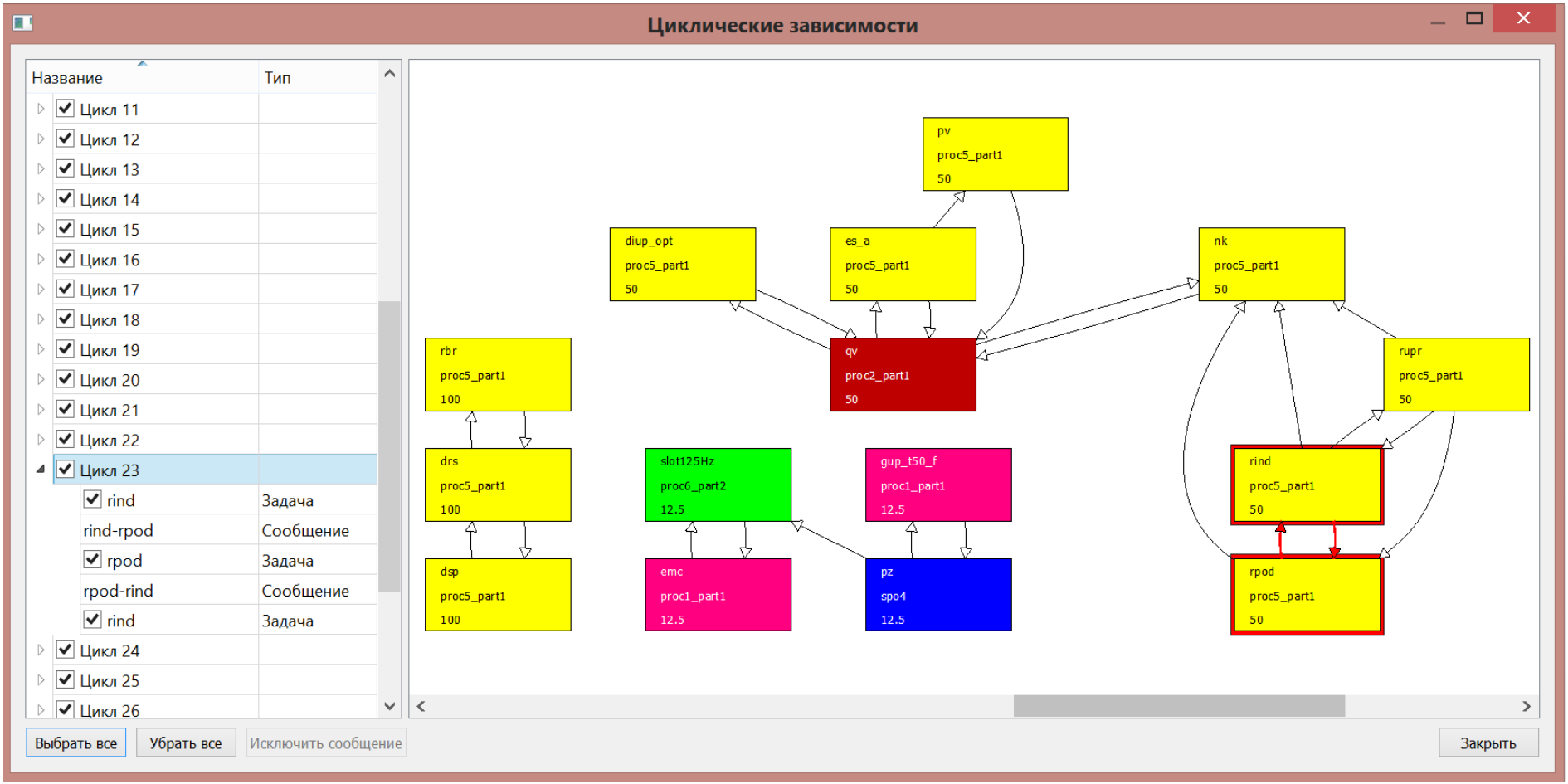
- Дано
  - Описание вычислительной системы с архитектурой ИМА
    - вычислительные средства (модули, процессоры в их составе)
    - сеть передачи данных (структура сети, транспортный протокол)
  - Описание рабочей нагрузки
    - на вычислители: задачи (сгруппированы в разделы)
    - на сеть: сообщения, передаваемые между задачами
- Требуется:
  - Распределить вычислительную нагрузку (привязать разделы к процессорным ядрам)
  - Построить конфигурацию сети (виртуальные каналы, их маршруты и характеристики)
  - Построить расписание окон для каждого ядра

Поэтапная схема сквозного планирования



Вычислительные ресурсы и рабочая нагрузка на них

- Вычислительные ресурсы
  - набор модулей
  - модуль → набор и типы процессорных ядер
  - ядро → верхняя граница загрузки
- Рабочая нагрузка на вычислительные ресурсы (вычислительная нагрузка)
  - набор задач
  - набор разделов
  - задача → период, приоритет, WCET (для типа ядра), принадлежность к разделу
  - раздел → допустимые ядра
  - сообщение → отправитель, получатель, размер, длительность передачи (через память)


Циклические зависимости
Название
Тип
Цикл 11
Цикл 12
Цикл 13
Цикл 14
Цикл 15
Цикл 16
Цикл 17
Цикл 18
Цикл 19
Цикл 20
Цикл 21
Цикл 22
Цикл 23
rind
Задача
rind-rpod
Сообщение
rpod
Задача
rpod-rind
Сообщение
rind
Задача
Цикл 24
Цикл 25
Цикл 26
pv
proc5_part1
50
diup_opt
proc5_part1
50
es_a
proc5_part1
50
nk
proc5_part1
50
rbr
proc5_part1
100
qv
proc2_part1
50
rupr
proc5_part1
50
drs
proc5_part1
100
slot125Hz
proc6_part2
12.5
gup_t50_f
proc1_part1
12.5
rind
proc5_part1
50
dsp
proc5_part1
100
emc
proc1_part1
12.5
pz
spo4
12.5
rpod
proc5_part1
50
Выбрать все
Убрать все
Исключить сообщение
Закрыть

Сеть передачи данных и рабочая нагрузка на нее

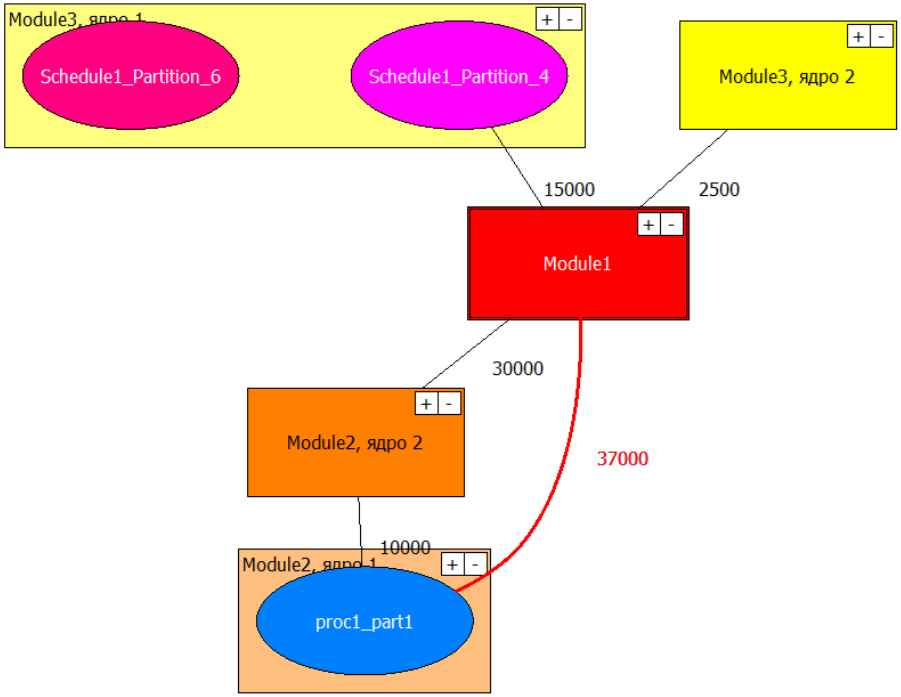
- Сеть передачи данных
  - набор коммутаторов
  - структура каналов (коммутатор-модуль, коммутатор-коммутатор)
  - пропускные способности каналов
  - транспортный протокол + соответствующая ему схема оценки задержек и джиттера передачи сообщений через сеть
- Рабочая нагрузка на сеть
  - набор сообщений
  - сообщение → задача-отправитель, задача-получатель, размер
  - период передачи сообщения определяется периодом задачи-отправителя
- Длительность передачи сообщения через сеть зависит от конфигурации сети (=> заранее не известна, может быть оценена снизу)

Распределение вычислительной нагрузки

- Дано:
  - Описание вычислительной системы
  - Описание рабочей нагрузки
- Требуется:
  - Построить привязку разделов к процессорным ядрам
- Оптимизируемый критерий:
  - Трафик между модулями → min
- Ограничения:
  - Каждый раздел привязан к одному ядру, допустимому для него
  - Загрузка каждого ядра не превышает максимально допустимой
- Инкрементальная постановка
  - Задано существующее распределение нагрузки
  - Распределить дополнительные разделы без изменения существующего распределения

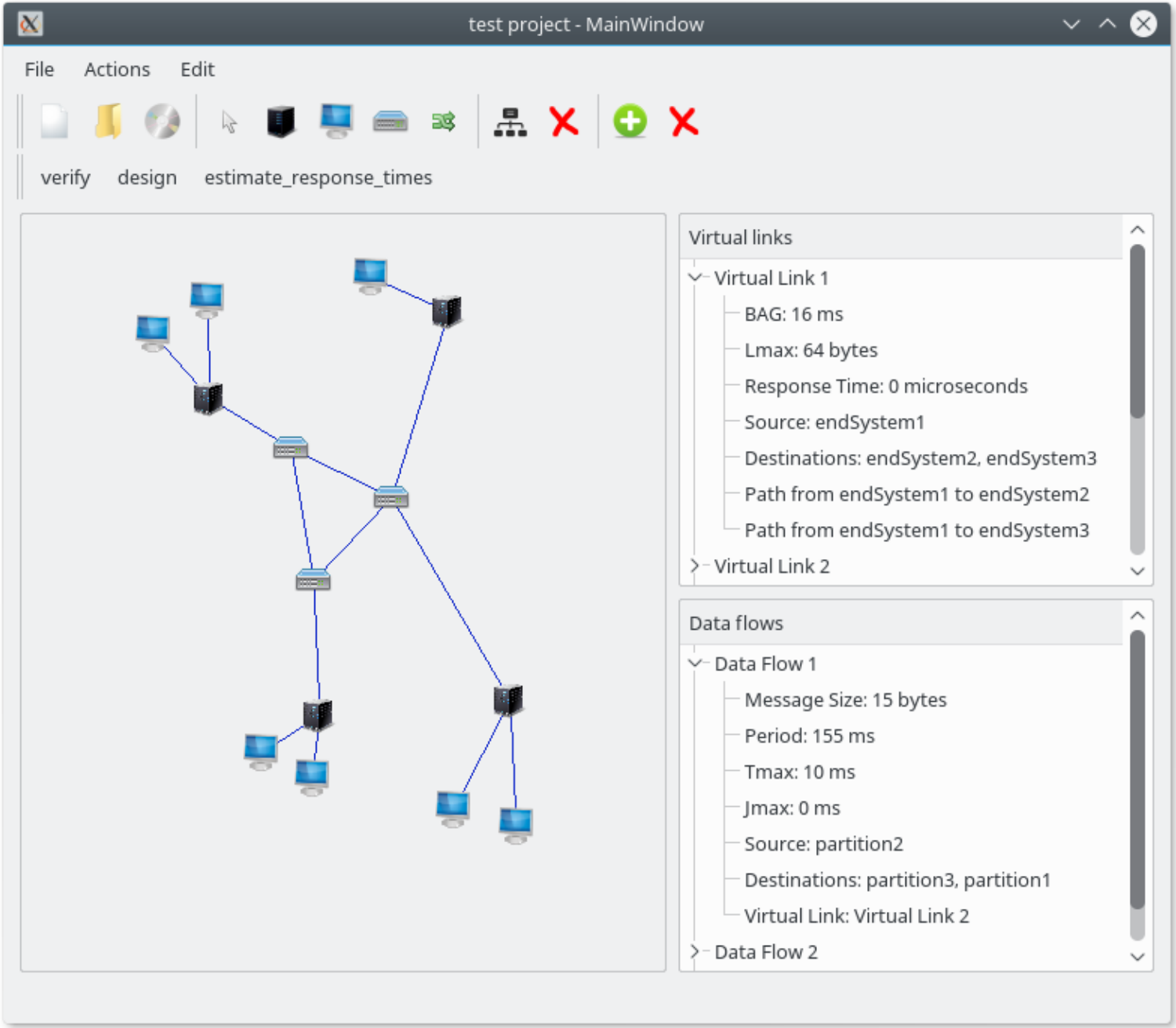
Привязка разделов

Название	Цвет
Module1	
Ядро 1	
proc2_part1	
Schedule1_Partition_2	
Schedule1_Partition_3	
Schedule1_Partition_7	
Ядро 2	
proc3_part1	
proc4_part1	
Schedule1_Partition_1	
Module2	
Ядро 1	
proc1_part1	
Ядро 2	
Schedule1_Partition_5	
Schedule1_Partition_8	
Module3	
Ядро 1	
Schedule1_Partition_4	
Schedule1_Partition_6	
Ядро 2	
Schedule1_Partition_9	



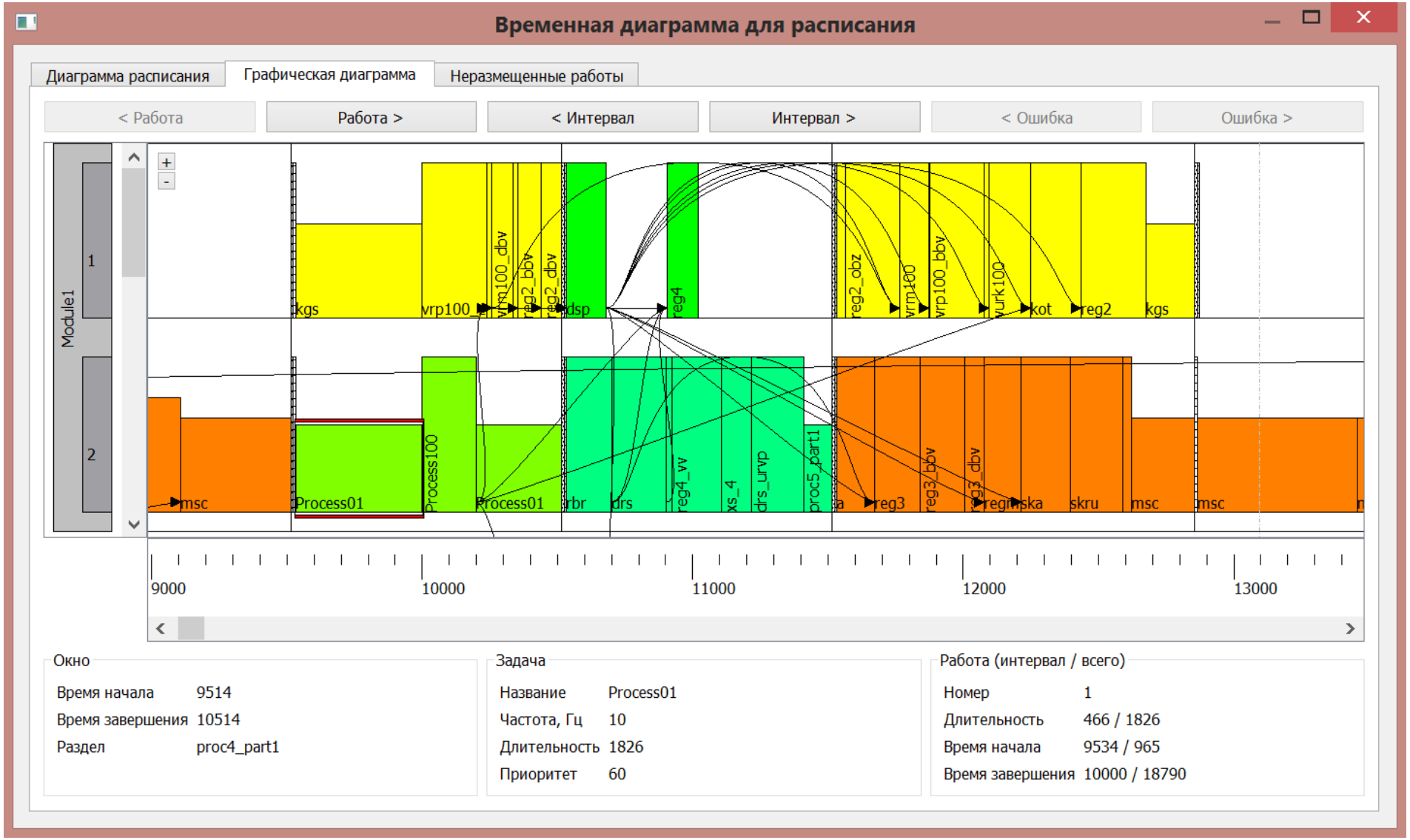
Построение конфигурации сети

- Дано:
  - Описание вычислительной системы
  - Описание рабочей нагрузки
  - Распределение вычислительной нагрузки (=> модуль-отправитель и модуль-получатель для каждого сообщения)
  - Опционально: максимально допустимые длительности и джиттеры передачи сообщений через сеть (индивидуально по сообщениям)
- Требуется:
  - Построить систему виртуальных каналов
  - Для каждого виртуального канала:
    - Модуль-отправитель
    - Набор модулей-получателей
    - Маршрут по сети от отправителя к получателям через коммутаторы и каналы
    - Параметры (на примере AFDX): максимальный размер кадра, BAG
    - Верхние оценки джиттера передачи и длительности передачи сообщения
- Оптимизируемый критерий: нет (задача удовлетворения ограничений)
- Ограничения:
  - Непревышение пропускной способности каналов
  - Непревышение максимально допустимых длительностей и джиттеров
- Инкрементальная постановка
  - Достраивание существующей системы виртуальных каналов


test project - MainWindow
File
Actions
Edit
verify
design
estimate_response_times
Virtual links
Virtual Link 1
BAG: 16 ms
Lmax: 64 bytes
Response Time: 0 microseconds
Source: endSystem1
Destinations: endSystem2, endSystem3
Path from endSystem1 to endSystem2
Path from endSystem1 to endSystem3
Virtual Link 2
Data flows
Data Flow 1
Message Size: 15 bytes
Period: 155 ms
Tmax: 10 ms
Jmax: 0 ms
Source: partition2
Destinations: partition3, partition1
Virtual Link: Virtual Link 2
Data Flow 2

Построение расписания окон

- Дано:
  - Описание вычислительной системы
  - Описание рабочей нагрузки
  - Распределение вычислительной нагрузки
  - Длительности передачи сообщений через сеть
  - Минимально и максимально допустимые длительности окон
- Требуется:
  - Построить расписание окон для каждого ядра
    - Ядро → набор окон
    - Окно → время начала, время завершения, раздел
- Оптимизируемый критерий: нет (задача удовлетворения ограничений)
- Ограничения:
  - окна для каждого ядра не перекрываются
  - к каждому окну привязан единственный раздел
  - длительность каждого окна не меньше минимально допустимой и не больше максимально допустимой
  - границы окон для всех ядер одного модуля должны совпадать
  - все работы выполняются в рамках директивных сроков, задаваемых периодами (при длительностях, равных WCET)
- Соблюдение директивных сроков проверяется **имитационным моделированием** работы динамического планировщика


Временная диаграмма для расписания
Диаграмма расписания
Графическая диаграмма
Неразмещенные работы
< Работа
Работа >
< Интервал
Интервал >
< Ошибка
Ошибка >
Module1
1
2
kgs
vrp100_
vrm100_dbv
reg2_bbv
reg2_dbv
dsp
reg4
reg2_obz
vrm100
vrp100_bbv
vurk100
kot
reg2
kgs
msc
Process01
Process100
Process01
rbr
drs
reg4_vv
xs_4
drs_urvp
proc5_part1
reg3
reg3_bbv
reg3_dbv
ska
skru
msc
msc
9000
10000
11000
12000
13000
Окно
Время начала 9514
Время завершения 10514
Раздел proc4_part1
Задача
Название Process01
Частота, Гц 10
Длительность 1826
Приоритет 60
Работа (интервал / всего)
Номер 1
Длительность 466 / 1826
Время начала 9534 / 965
Время завершения 10000 / 18790

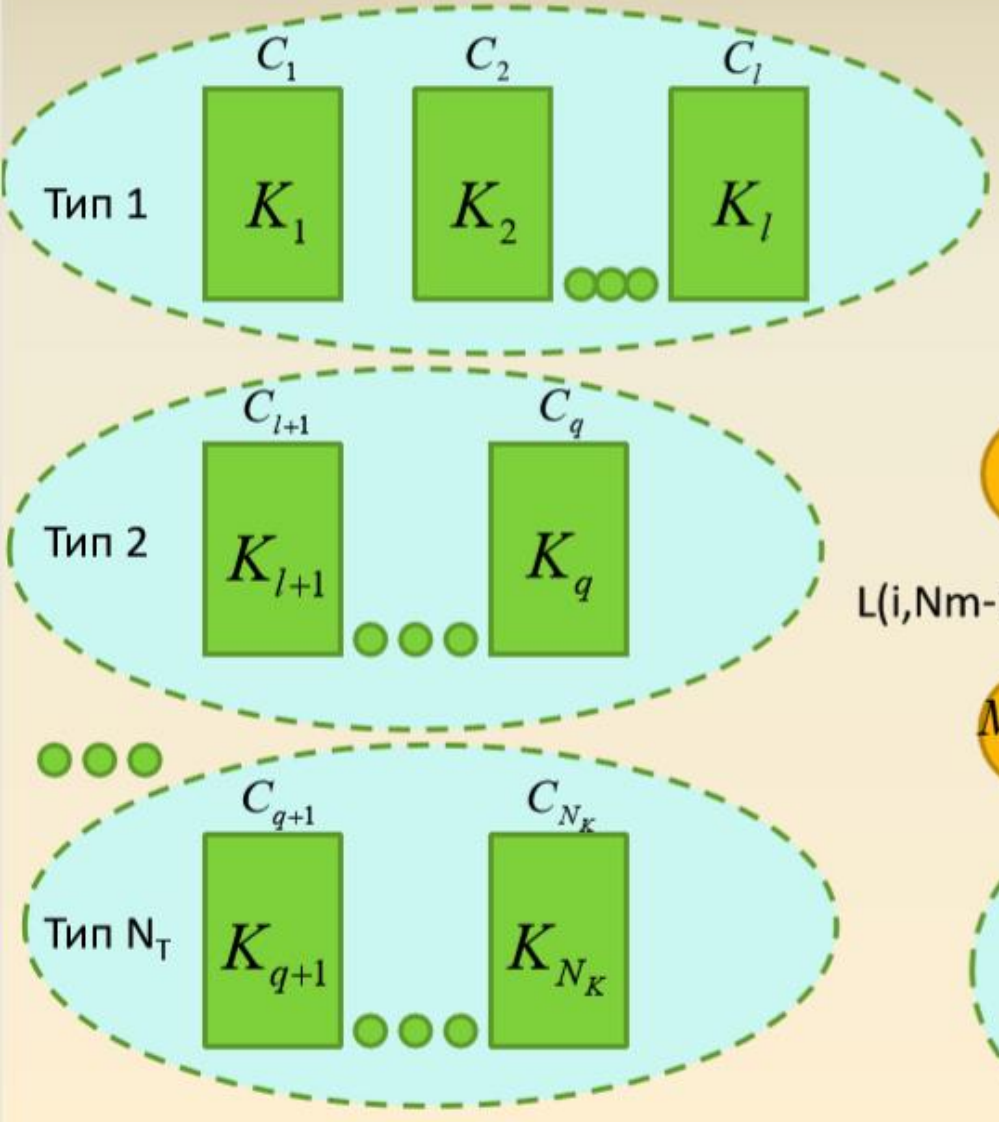
Жадный алгоритм распределения вычислительной наргрузки

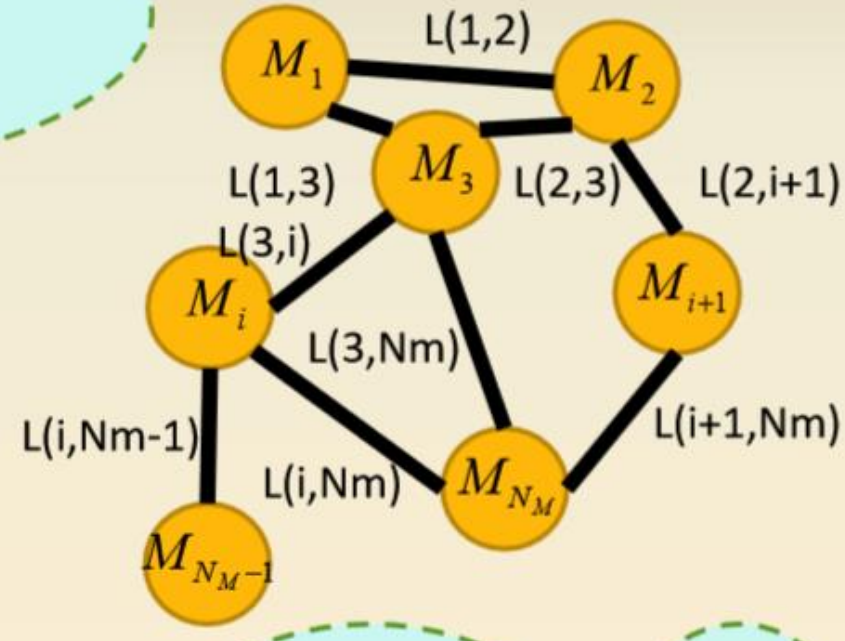
1. Выбрать непривязанный раздел *P* с максимальным трафиком между *P* и привязанными разделами.
2. Для каждого модуля рассчитать трафик между *P* и разделами, привязанными к ядрам других модулей.
3. Упорядочить модули по убыванию этого трафика.
4. В цикле по модулям:
   - если раздел *P* может быть привязан к какому-либо ядру данного модуля без нарушения ограничений на загрузку ядра, то привязать *P* к этому ядру и выйти из цикла;
   - иначе, если разделы на данном модуле могут быть перераспределены между ядрами этого модуля так, чтобы в достаточной мере «разгрузить» одно из ядер, то выполнить перераспределение, привязать *P* к этому ядру и выйти из цикла;
5. Если на шаге 4 не выбрано никакое ядро, то **стоп** (неуспех).
6. Если остались непривязанные разделы, то перейти к шагу 1, иначе **стоп** (успех).

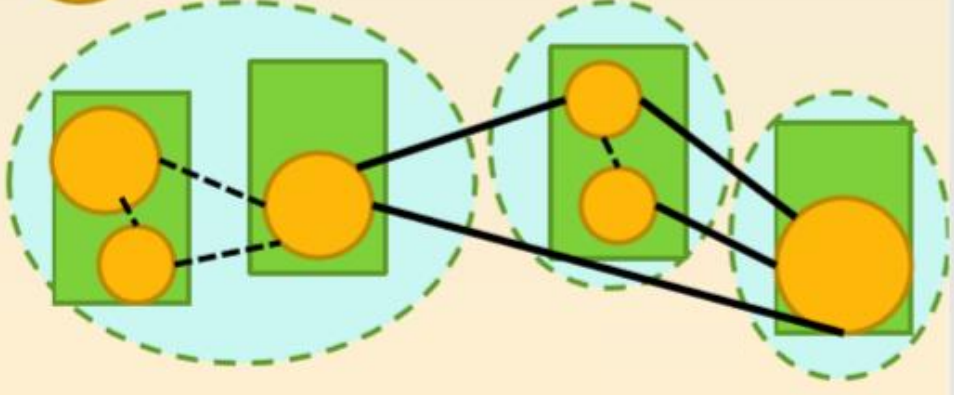
Модифицированная обобщенная задача о мультипликативном рюкзаке

$$\forall i \in \overline{1, N_M} \; \forall j \in \overline{1, N_K}$$

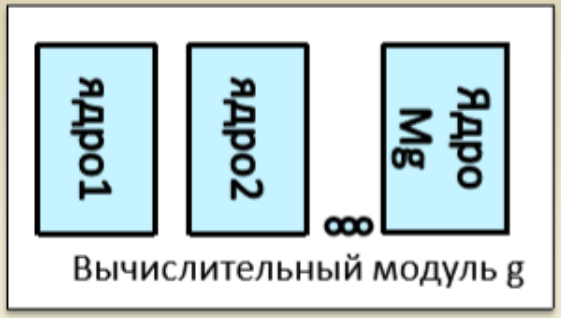
задан объем $w_{i,j}$

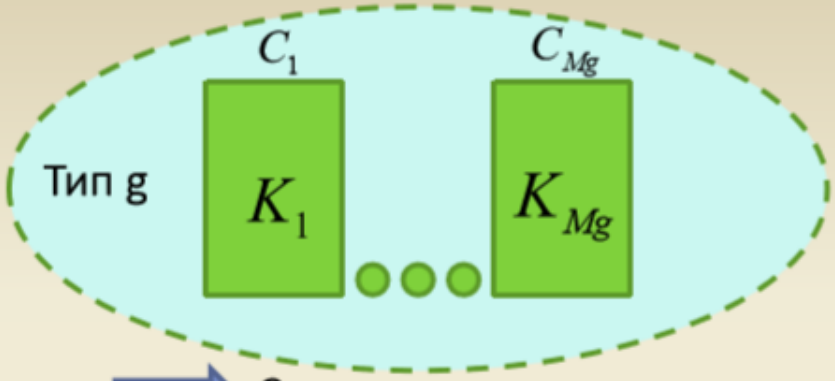






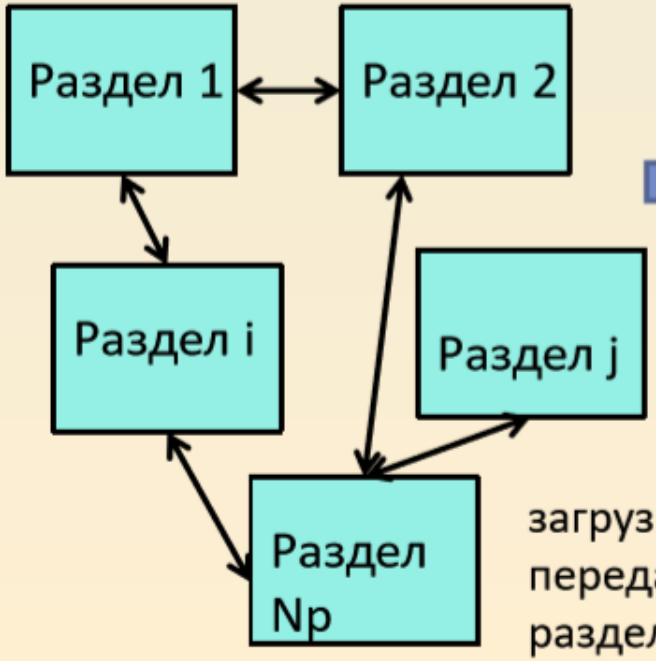
Представление задачи РВН в виде модифицированной обобщенной задачи о мультипликативном рюкзаке

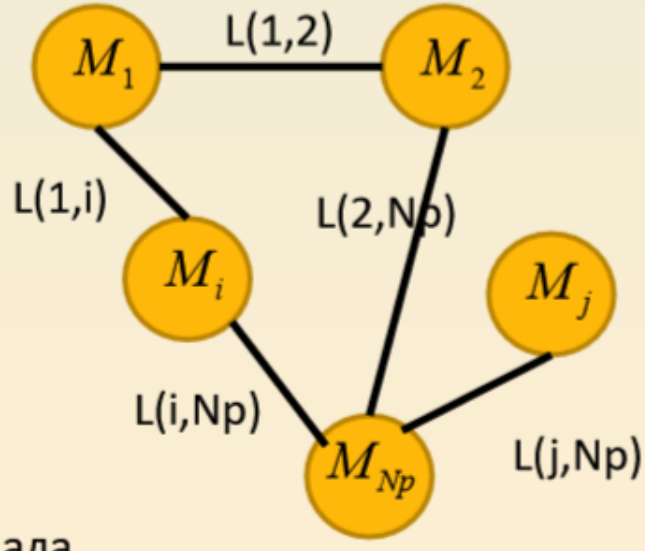




максимальная загрузка каждого ядра → C_n

вклад каждого раздела в загрузку каждого процессорного ядра → w_{ij}



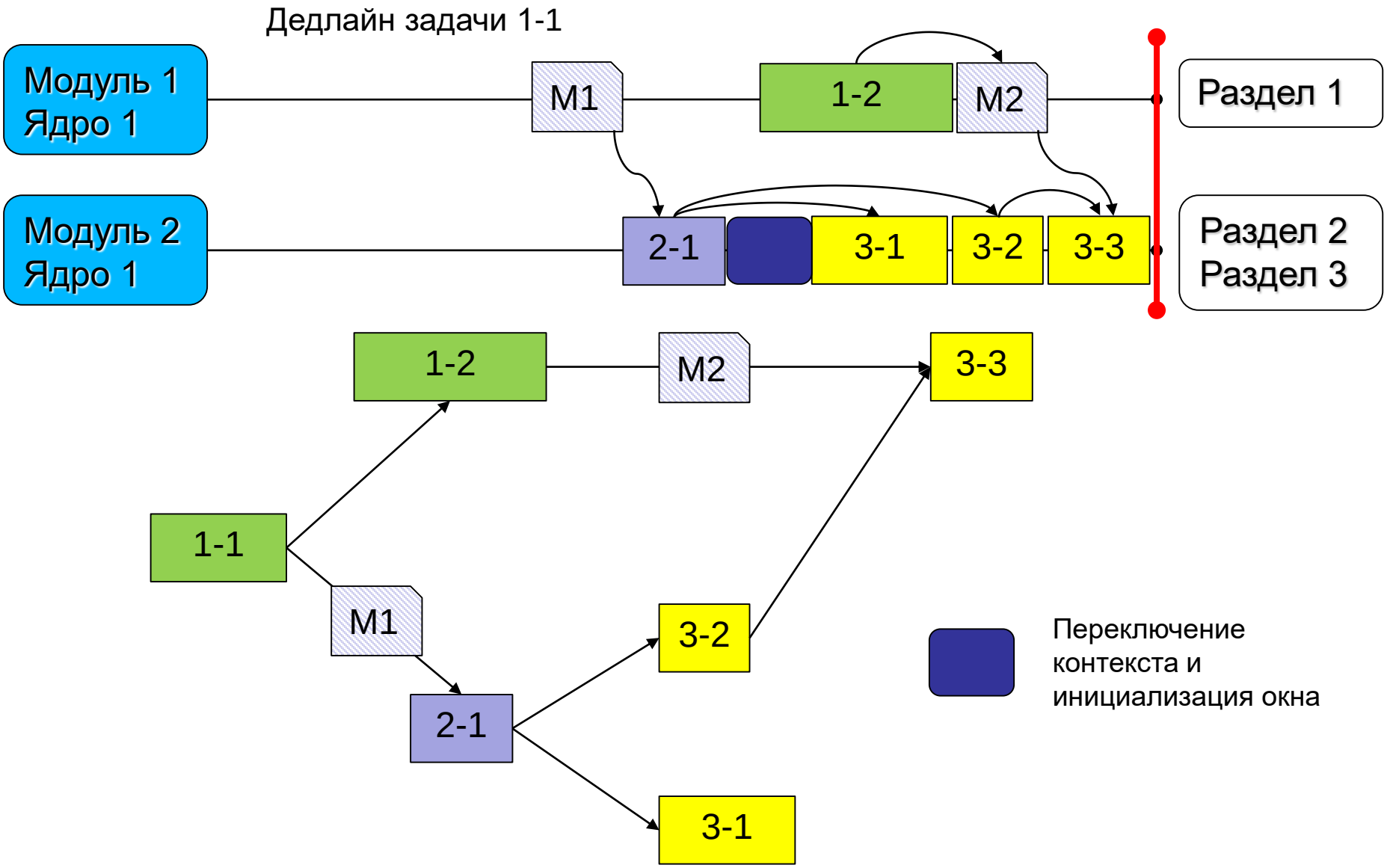


загрузка межмодульного канала передаваемыми между каждой парой разделов сообщениями → L(x,y)

Построение расписания окон: схема алгоритма

1. Построение графа зависимостей работ
2. Уточнение директивных интервалов работ на основе зависимостей
3. Построение последовательностей выполнения работ «слева направо», параллельно для всех ядер
4. Построение расписания окон на основе построенной последовательности работ
5. Назначение приоритетов задач на основе построенной последовательности работ

Уточнение директивного срока



Планирование «слева направо»

1. Работа помещается в список готовых для выполнения, если:
   - Начался ее директивный интервал
   - Получены все синхронные входящие сообщения для данной работы
2. Очередная работа для размещения выбирается из списка:
   - Если отсутствуют готовые работы с большим приоритетом
   - По критерию EDF (если приоритет еще не определен)

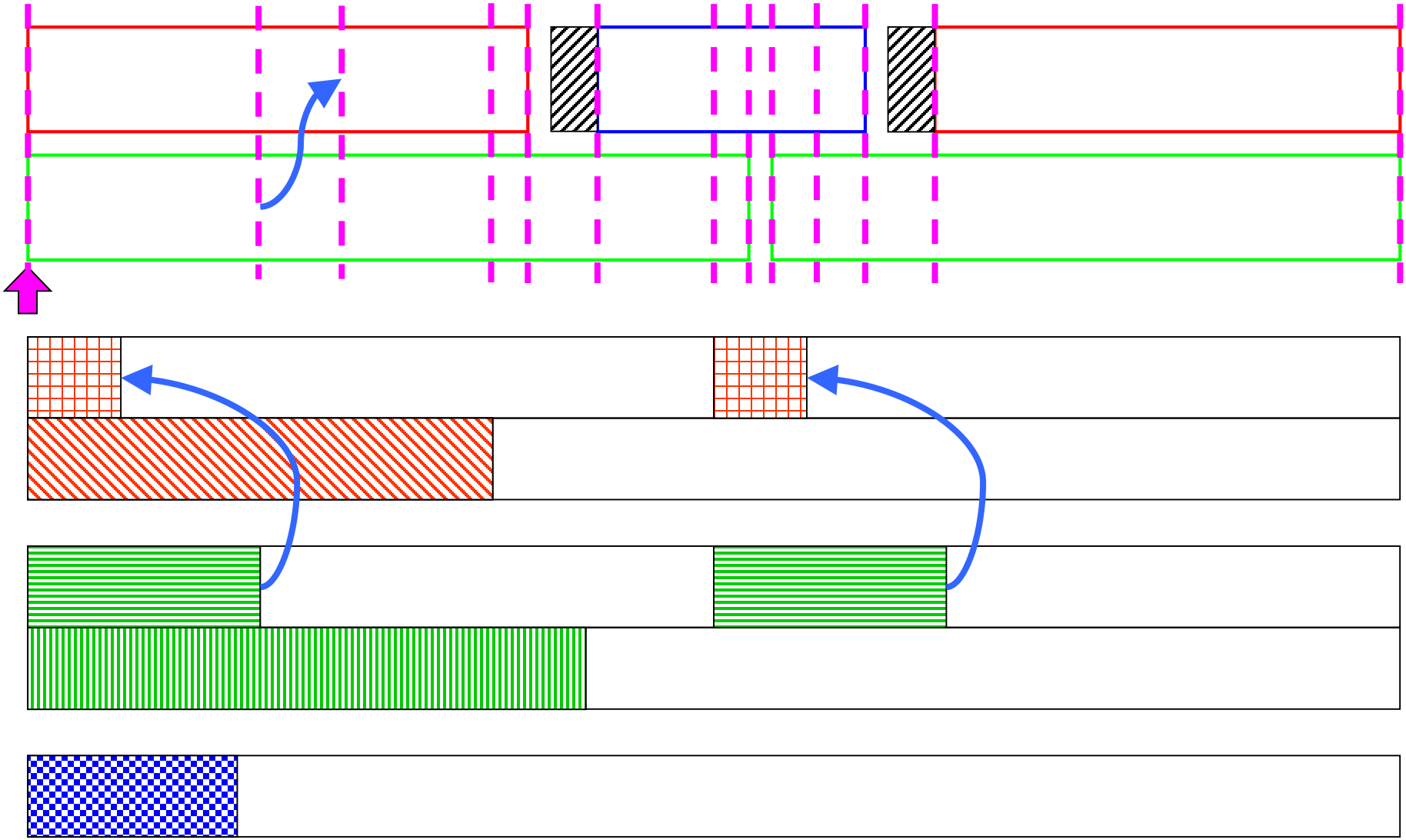
Модель вычислительного процесса

- Служит для проверки корректности расписания окон
- Модель основана на событиях
- Схема работы:
  1. Создание начальных событий
  2. Выбор событий с минимальным временем
  3. Обновление списка готовых к выполнению работ
  4. Обработка выбранных событий
  5. Если список событий не пуст, переход к п.2

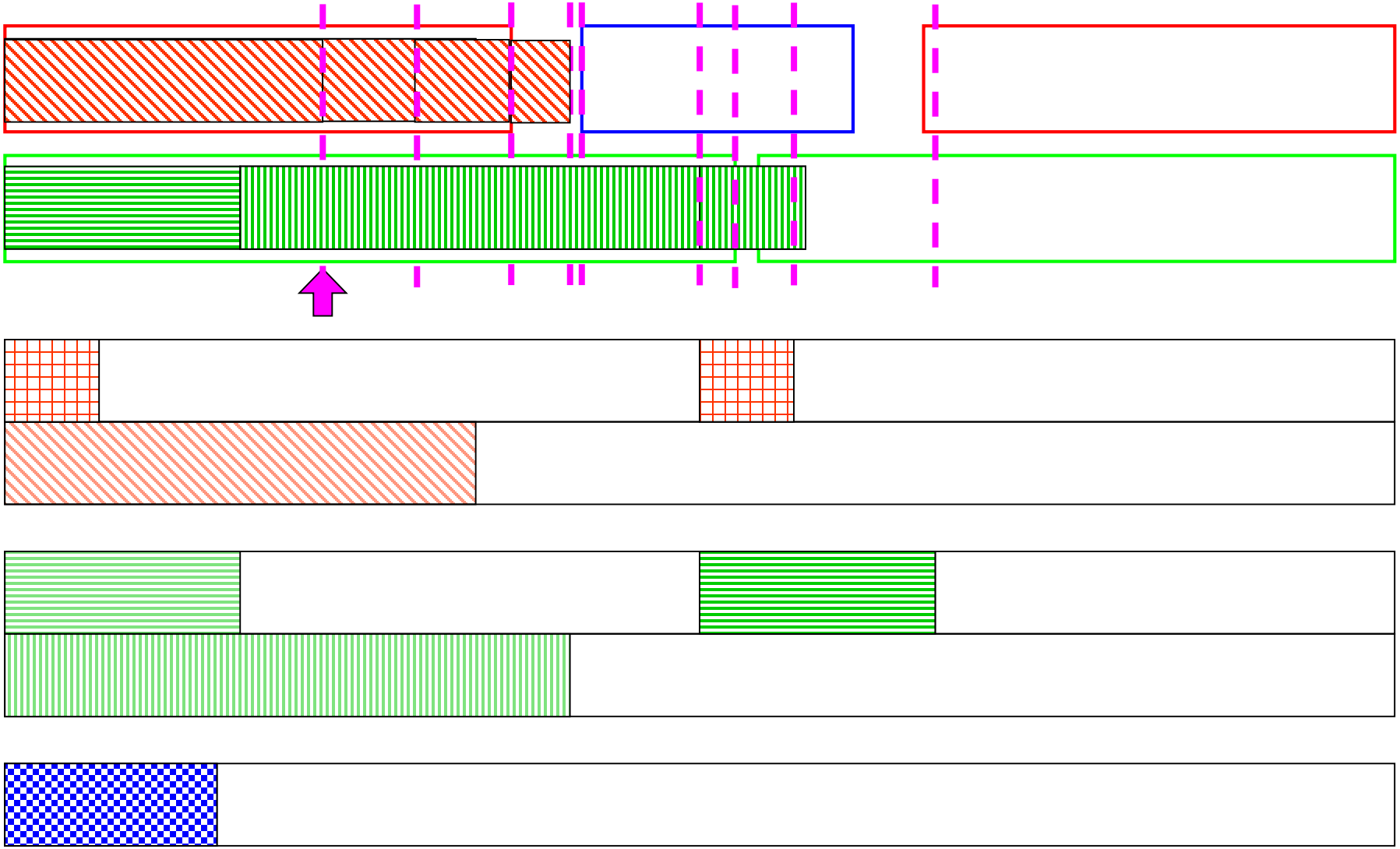
Типы событий

- Открытие окна
- Закрытие окна
- Начало директивного интервала
- Завершение работы
- Доставка сообщения

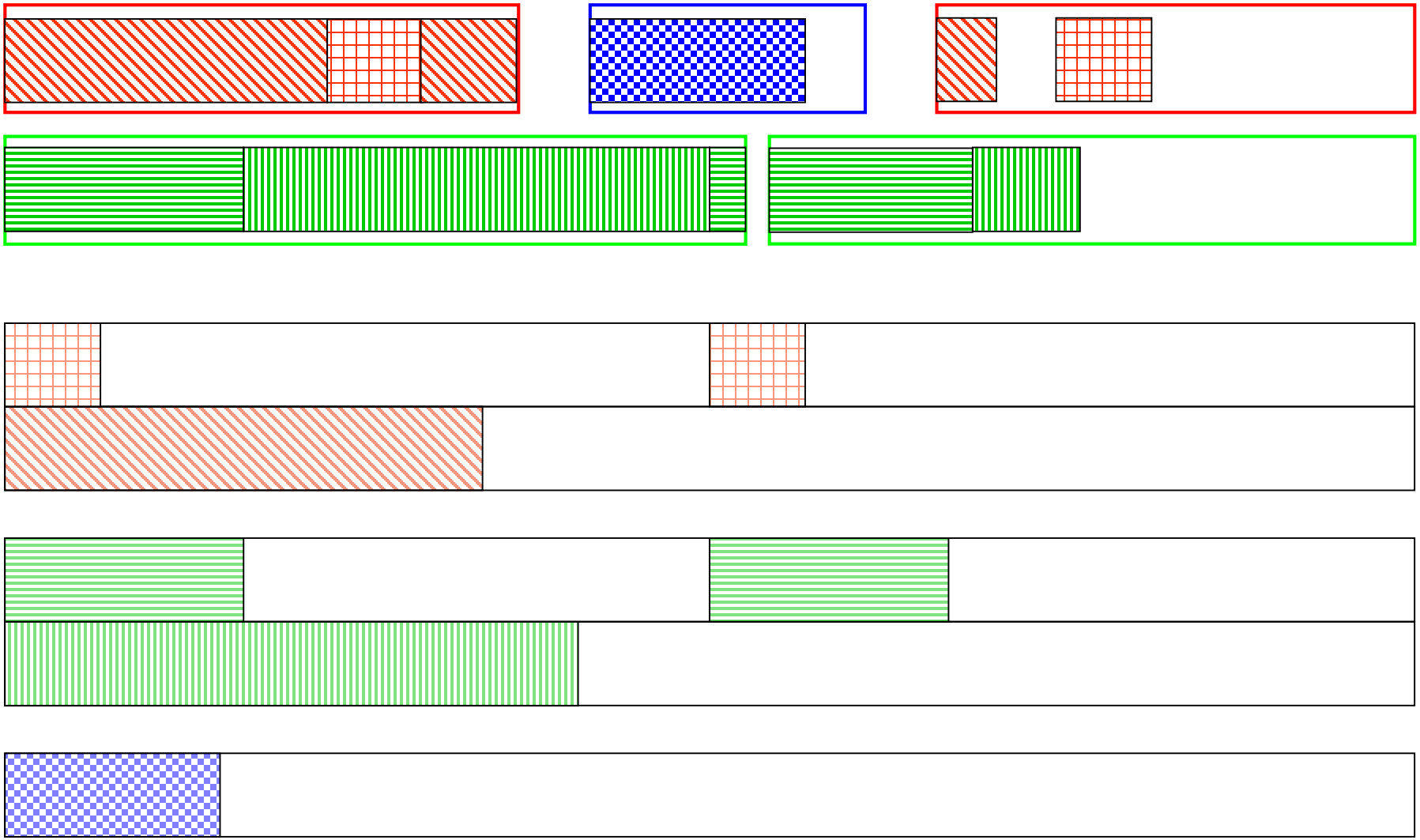
Работа модели



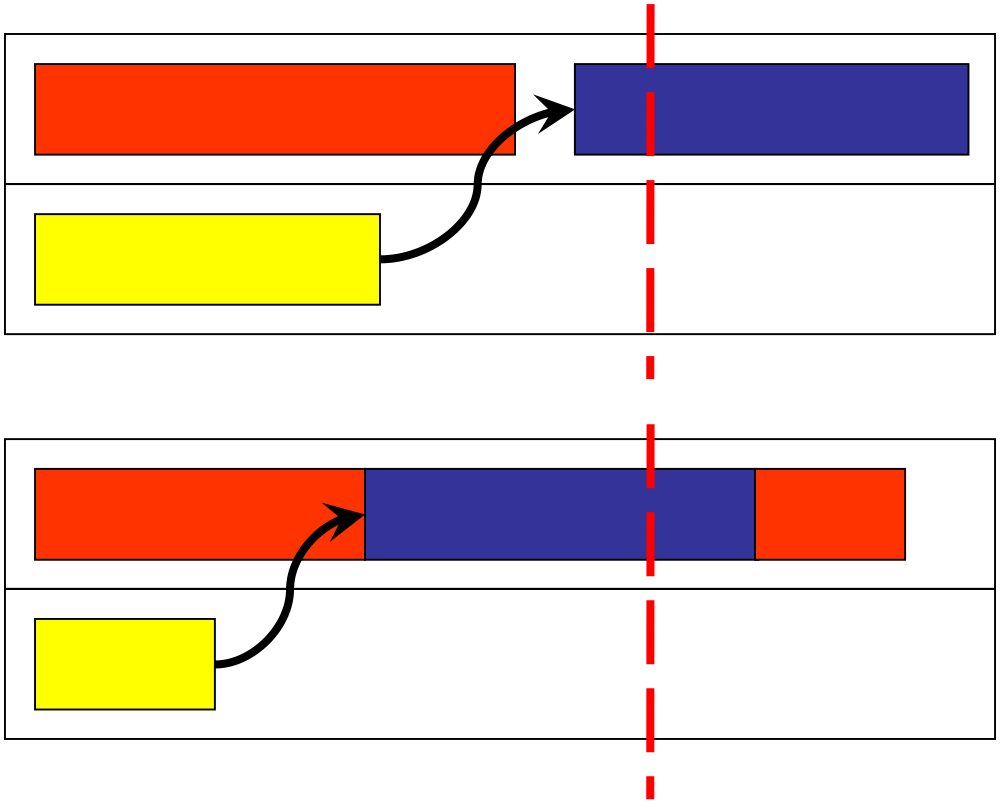
Работа модели



Работа модели



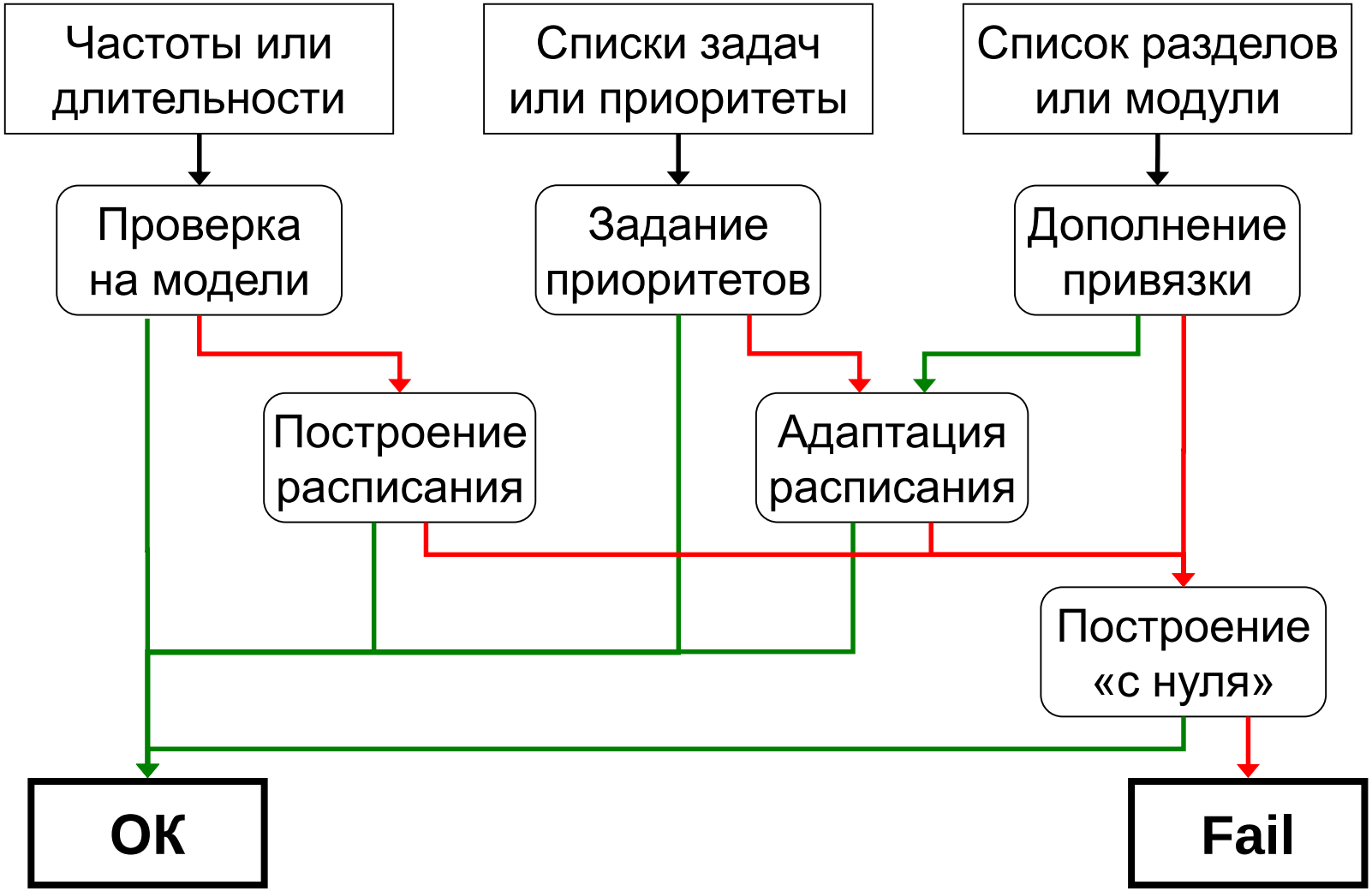
Недостатки модели



Решение:

не выполнять низкоприоритетные задачи, пока есть высокоприоритетные, ожидающие сообщений

Адаптация к изменениям



Перспективные задачи

- Построение расписания с минимизацией затрат на переключение контекста
- Корректировка распределения разделов по ядрам при неуспешном построении расписания
- Построение расписания, гарантированно корректного при временах выполнения задач, меньших чем WCET
- Минимизация сетевой загрузки с учётом синхронности зависимостей по данным
  - сообщения, которые ожидает задача-получатель, следует передавать через память (=> привязка отправителя и получателя к одному модулю)
- Совместное планирование задач и конфигурирование сети

Спасибо за внимание