Планирование вычислений в распределенных системах

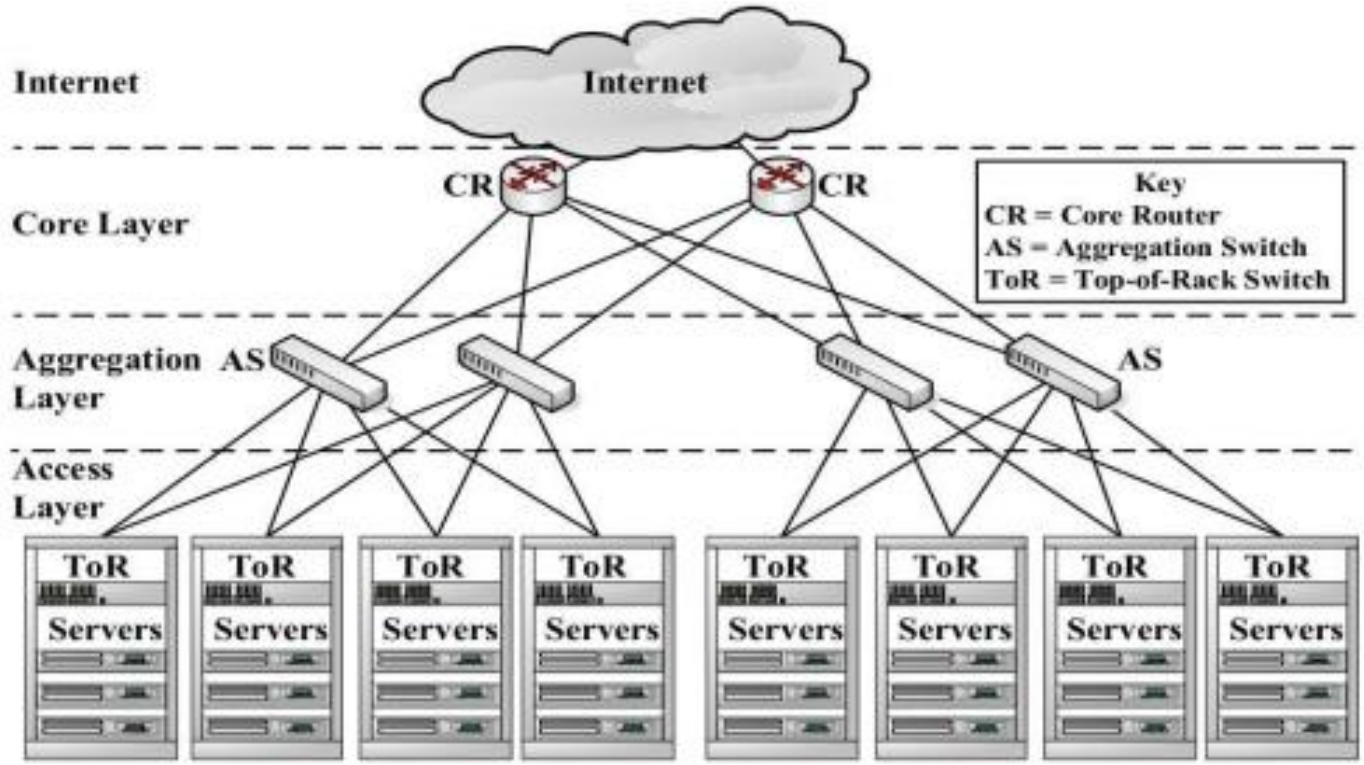
Костенко Валерий Алексеевич
МГУ им. М.В. Ломоносова,
факультет ВМК
kost@cs.msu.su

2021 г.

Центры обработки данных (ЦОД)

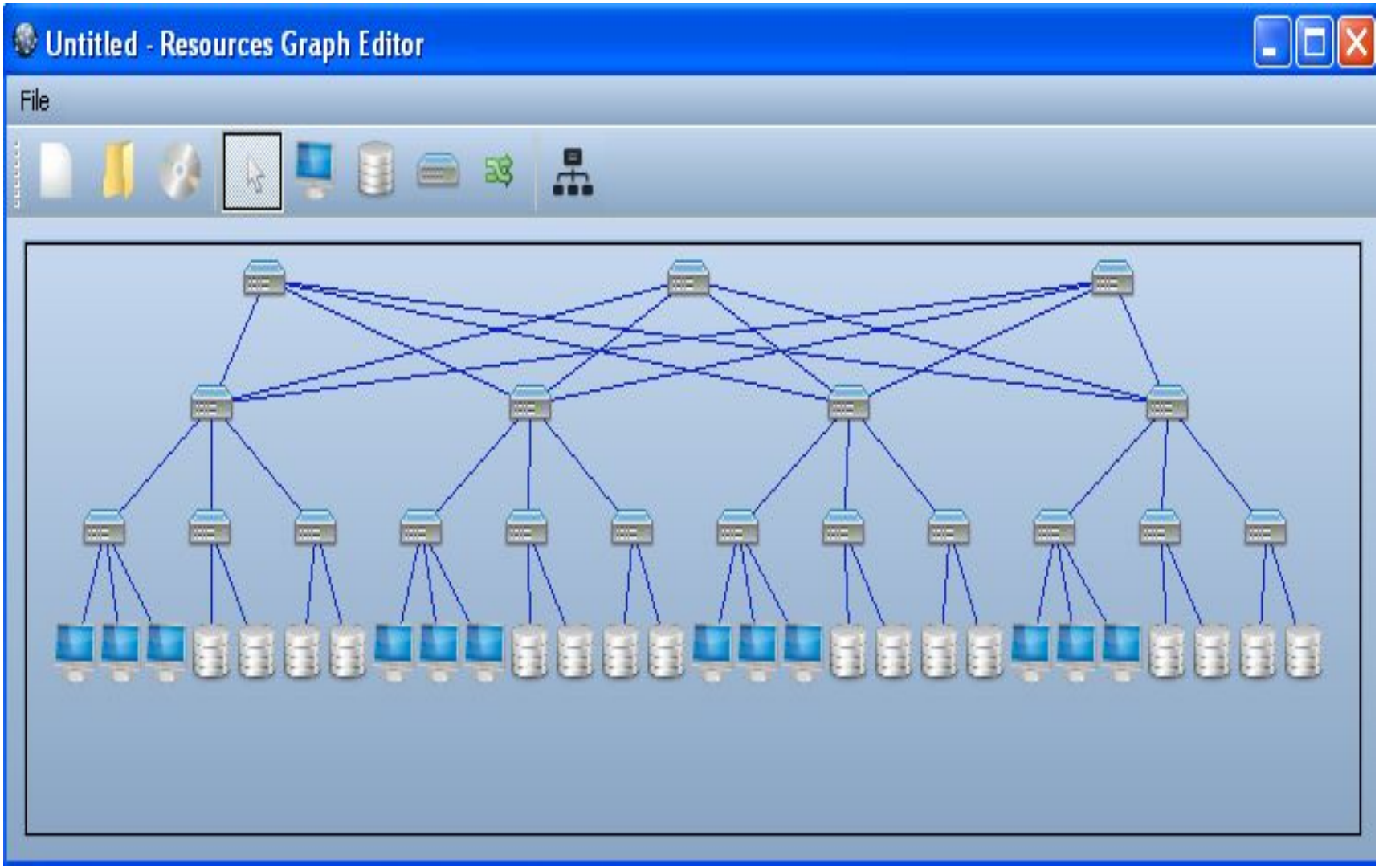
Виртуализация — предоставление набора вычислительных ресурсов или их логического объединения, абстрагированное от аппаратной реализации, и обеспечивающее при этом логическую изоляцию вычислительных процессов, выполняемых на одном физическом ресурсе.

Bhanu P. Tholeti. Hypervisors, virtualization, and the cloud: Learn about hypervisors, system virtualization, and how it works in a cloud environment. c Copyright IBM Corporation 2011 — 8c. [Электронный ресурс]. URL: https://www.ibm.com/developerworks/cloud/library/clhypervisorcompare/ (дата обращения: 10.04.2016)



- *Core Layer - корневой уровень состоящий из маршрутизаторов, имеющих доступ в интернет.*
- *Agregation Layer - уровень агрегации (коммутаторы)*
- *Access Layer - уровень доступа (прикладной уровень), состоящий из серверов*

Топология сети обмена ЦОД fattree



- *Вычислительные ресурсы*
- *Ресурсы хранения данных*
- *Сетевые ресурсы*

Характеристики физических ресурсов

- *Вычислительные узлы:*
  - *<число ядер>,*
  - *<частота>,*
  - *<объем оперативной памяти>,*
  - *<объем дисковой памяти>;*
- *Хранилища данных:*
  - *<объем памяти>,*
  - *<тип хранилища данных>;*
- *Коммутационные элементы и каналы передачи данных:*
  - *<пропускная способность>,*
  - *<задержка>.*

Типы виртуальных ресурсов

1. *Виртуальная машина (сервер).*
2. *Виртуальная система хранения данных.*
3. *Виртуальная не администрируемая сеть.*
4. *Виртуальная администрируемая сеть.*

Модели обслуживания

- *SaaS (Software-as-a-Service) – программное обеспечение как услуга.*

- *PaaS (Platform-as-a-Service) – платформа как услуга.*

- *IaaS (Infrastructure-as-a-Service) – инфраструктура как услуга.*

Модель SaaS

Преимущества:

- *с точки зрения потребителя отсутствует необходимости установки ПО на рабочих местах пользователей — доступ к ПО осуществляется через обычный браузер;*
- *с точки зрения поставщика сравнительно низкие затраты ресурсов на обслуживание конкретного клиента.*

***Основной недостаток** - сравнительно малый класс решаемых задач.*

Модель PaaS

Премущества:

- *У поставщика нет необходимости приобретать оборудование и программное обеспечение для разработки.*
- *Весь перечень операций по разработке, тестированию и разворачиванию приложений можно выполнить в одной интегрированной среде.*
- *Возможность создавать исходный код и предоставлять его в общий доступ внутри команды разработки.*

Недотатки:

- *Пользователь может пользоваться только библиотеками и инструментами, предоставляемыми системой.*
- *При разработке облачной системы данного типа необходимо тщательно продумывать политику безопасности.*

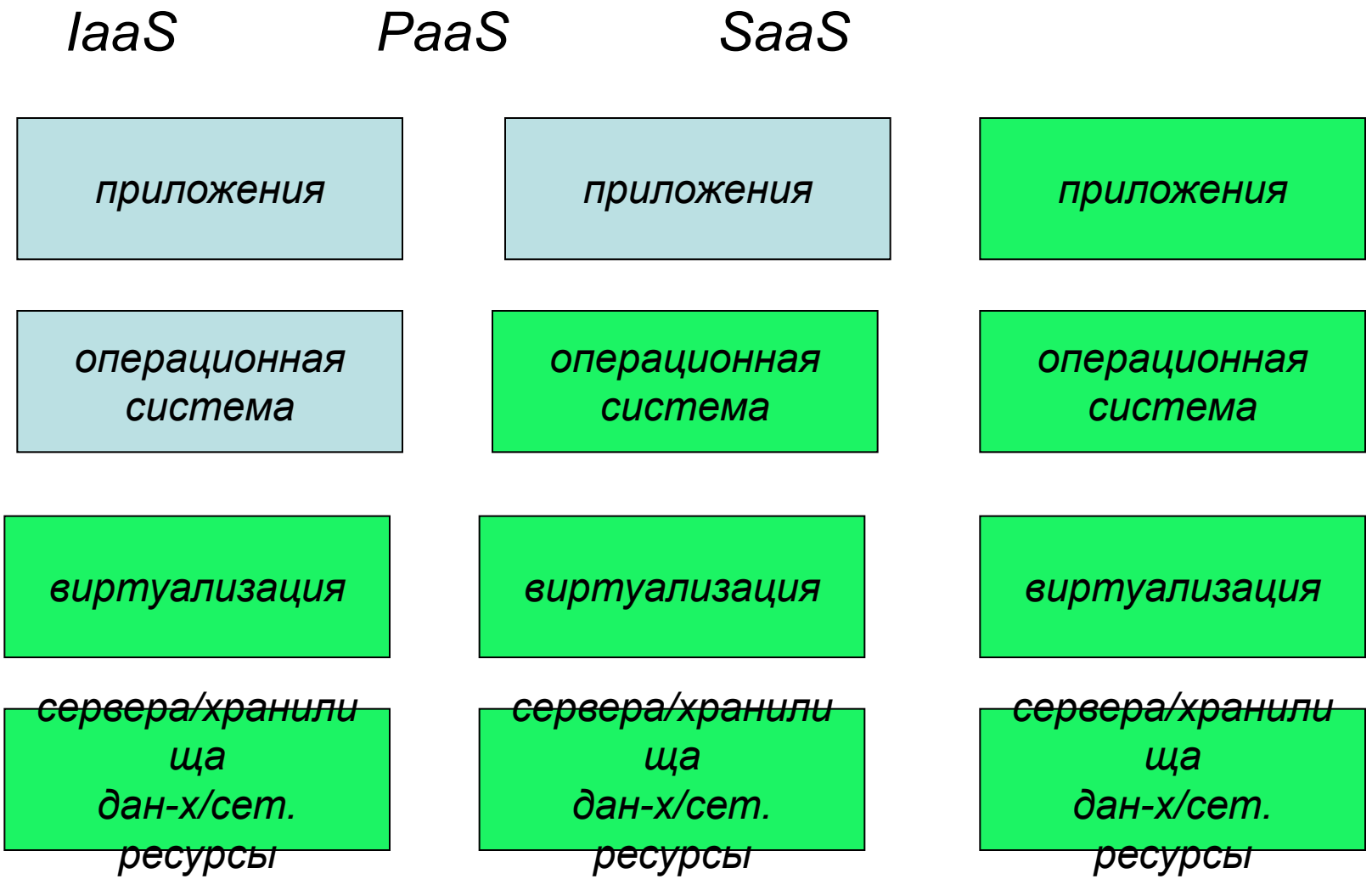
Модель IaaS

Данная модель обладает наибольшей гибкостью.

Потребителю предоставляется свобода в пользовании системы, поставщиком определяется только выделение физических ресурсов ЦОД.

Возможность создания администрируемых виртуальных корпоративных сетей?

Модели обслуживания



Модели обслуживания

- *SaaS (Software-as-a-Service) – программное обеспечение как услуга.*

- *PaaS (Platform-as-a-Service) – платформа как услуга.*

- *IaaS (Infrastructure-as-a-Service) – инфраструктура как услуга.*

Функционирование ЦОД

Жизненный цикл виртуального ресурса

1. *Создание запроса на выделение ресурсов ЦОД (пользователь).*
2. *Построение отображения запросов на физические ресурсы (планировщик).*
3. *Реализация отображения (средства облачной платформы).*
4. *Использование ресурса (управление работой и мониторинг виртуального ресурса - средства облачной платформы).*
5. *Снятие виртуального ресурса, добавление/удаление виртуальных элементов (пользователь, планировщик, средства облачной платформы).*

Соглашения об уровне обслуживания (Service Level Agreement (SLA))

Гарантированные SLA – выполнение соглашений о качестве сервиса в любой момент времени.

Негарантированные SLA – максимальное значение критерия качества сервиса, которое может быть предоставлено.

Методы повышения эффективности использования ресурсов ЦОД .

Аппаратные решения.

Программные решения:

- *Повышение качества работы планировщиков ресурсов.*

Алгоритмы отображения запросов на физические ресурсы ЦОД

Алгоритмы используются в планировщиках облачных платформ.

Планировщик ресурсов для облачных платформ

Планировщик определяет, какие элементы запроса на получение ресурсов на каких физических ресурсах ЦОД должны быть размещены.

Модель физических ресурсов ЦОД

Модель физических ресурсов: $H = (P \cup M \cup K, L)$

- *P – множество вычислительных узлов: vh(p), qh(p)*
- *M – множество систем хранения данных: uh(m), type(m)*
- *K – множество коммутационных элементов сети обмена: ch(k)*
- *L – множество физических каналов передачи данных: rh(l)*

Модель запроса

$$G = (W \cup S, E)$$

- *W – множество виртуальных машин реализующих приложения: v(w), q(w)*
- *S – множество storage-элементов: u(s), type(s)*
- *E – множество виртуальных каналов: r(e)*

Размещение элементов запроса на физические ресурсы ЦОД

Размещением запроса будем называть отображение

$$A: G \to H = \{W \to P, S \to M, E \to \{k, l\}\}$$

которое удовлетворяет ограничениям:

$$\sum_{w \in W_p} v(w) \le vh(p), \quad \sum_{w \in W_p} q(w) \le qh(p)$$

$$\sum_{e \in E_l} r(e) \le rh(l)$$

$$\sum_{e \in E_k} r(e) \le \tau h(k)$$

$$\sum_{s \in S_m} u(s) \le uh(m), \qquad \forall s \in S_m : type(s) = type(m)$$

Политики размещения

- ***Две виртуальных машины должны быть размещены на разных серверах*** .

  Это может потребоваться для обеспечения надежности работы приложения.

- ***Две виртуальных машины должны быть размещены на одном сервере.***

  Это может быть важно для обеспечения требуемой производительности параллельного приложения.

Граф остаточных ресурсов H_{res}

- *Для графа H_{res} переопределены значения функций:* $vh(p), qh(p), rh(l), \tau h(k), uh(m)$

$$vh_{res}(p) = vh(p) - \sum_{w \in W_p} v(w) \qquad qh_{res}(p) = qh(p) - \sum_{w \in W_p} q(w)$$

$$rh_{res}(l) = rh(l) - \sum_{e \in E_l} r(e)$$

$$\tau h_{res}(k) = \tau h(k) - \sum_{e \in E_k} r(e)$$

$$uh_{res}(m) = uh(m) - \sum_{s \in S_m} u(s)$$

Задача распределения ресурсов

Дано:

- *Множество поступивших запросов $Z = \{G_i\}$.*
- *Множество выполняемых запросов для которых допустима миграция $M = \{G_i\}$ и их отображение $A_M: M \rightarrow H$.*
- *Граф остаточных ресурсов ЦОД: H_{res}.*

Требуется:

- *определить максимальное число запросов из Z, которые можно разместить не нарушая SLA (множество $L \in Z$), и построить отображения:*
  - $A_L: L \in Z \rightarrow H_{res}\ U\ H_M$
  - $A^*_M: M \rightarrow H_{res}\ U\ H_M$

Эффективность эксплуатации ЦОД

- *Загрузка физических ресурсов.*
- *Процент размещенных запросов на создание виртуальных сетей из исходно поступивших.*
- *Производительность виртуальных машин.*

Производительность виртуальных машин.

Коэффициенты вариации:

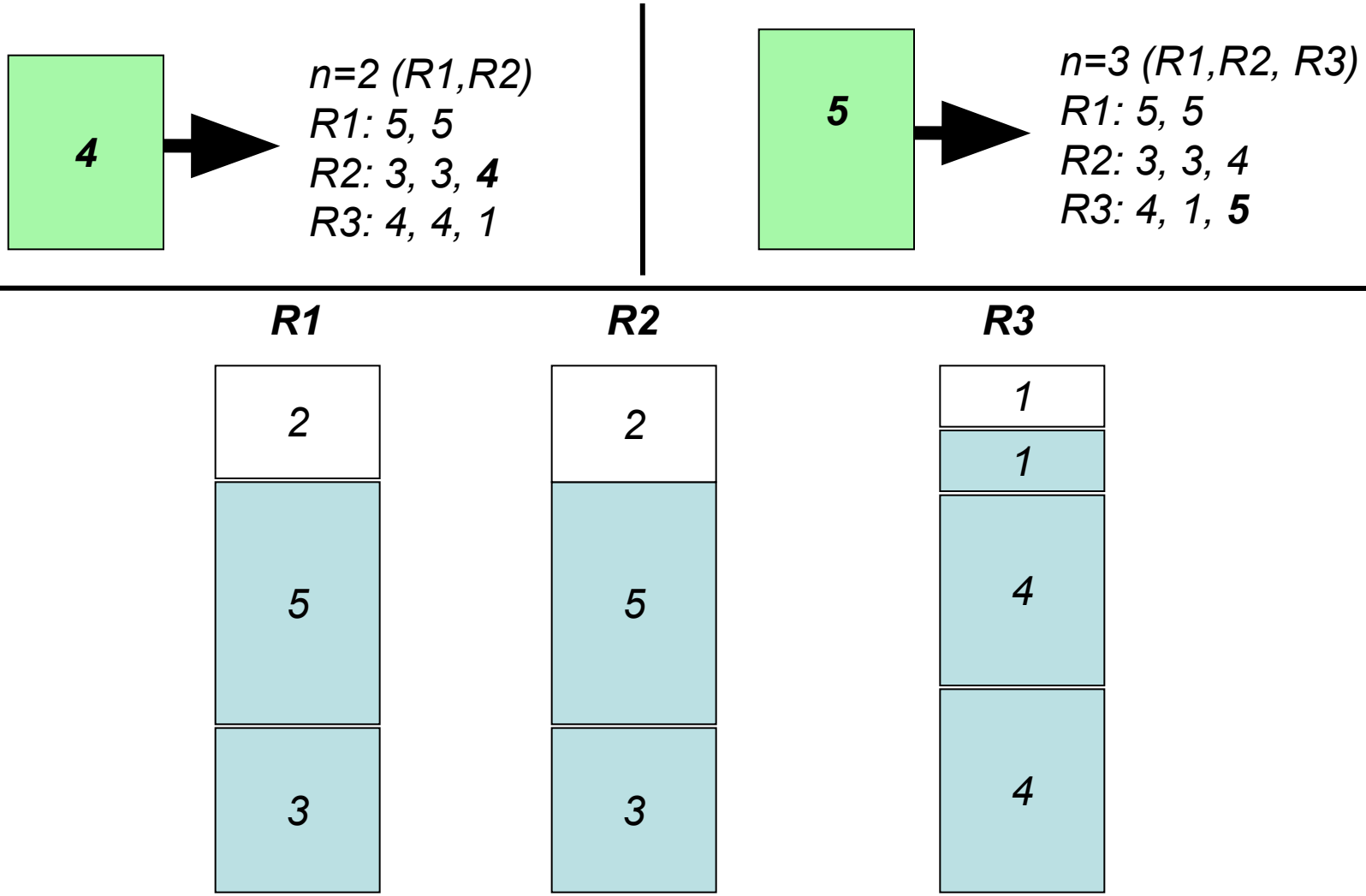
- *Производительность CPU (утилита Ubench) — 24 %.*
- *Скорость оперативной памяти (утилита Ubench) — 10 %.*
- *Последовательное чтение с диска (Кбайт/с) — 17%.*
- *Последовательная запись на диск (Кбайт/с) — 19 %.*
- *Случайные чт./зап. на диск (с) — от 9 % до 20 %.*

Производительность виртуальных машин.

Причины:

- ***Гетерогенная инфраструктура физических ресурсов ЦОД.*** *Виртуальные машины могут размещаться на серверах с разным типом процессора.*
- ***Переиспользование ресурсов.*** *Суммарное количество ресурсов, необходимых для работы виртуальных машин, может быть больше, чем доступно на сервере. Ресурсы сервера в этом случае становятся разделяемыми между виртуальными машинами. Они не могут постоянно использовать некоторый набор ресурсов, так как их переключает гипервизор.*
- ***NUMA архитектура.*** *Ядра выделенные виртуальной машине могут принадлежать различным или одному NUMA блоку.*

Фрагментация ресурсов ЦОД



Миграция ВМ

Основные критерии, по которым сравнивают различные схемы миграции:

1. *Промежуток времени, после которого сервер-источник освободится (eviction time).*
2. *Общее время миграции (total migration time).*
3. *Промежуток времени, в течение которого в процессе миграции виртуальная машина недоступна (downtime).*
4. *Общее количество переданных данных в процессе миграции виртуальной машины.*
5. *Уменьшение производительности мигрирующей ВМ.*

Миграция ВМ

Приостановленная виртуальная машина (suspended):

- *состояние всех работающих приложений сохраняется, ВМ переходит в «приостановленное» состояние (аналогично спящему режиму персонального компьютера),*
- *далее происходит перемещение ВМ с сервера-источника на сервер-приемник,*
- *ВМ вновь запускается с восстановлением состояний всех ранее работающих приложений.*

Миграция ВМ

Этапы «живой» миграции:

1. *Передача состояния процессора.*
2. *Передача состояния оперативной памяти.*
3. *Передача содержимого внешнего диска (опциональная часть миграции).*

Миграция ВМ

Схема организации «живой» миграциии Pre-Copy.

- *Оперативная память копируется с сервера-источника на сервер-получатель итерациями. ВМ в процессе миграции остается на сервере-источнике.*
- *На первой итерации копируется вся память, на последующих итерациях копируются только те страницы, над которыми были произведены изменения.*
- *ВМ приостанавливается и полностью переносится на сервер-получатель вместе с состоянием процессора и оставшимися модифицированными страницами памяти если:*
  - *количество итерации превысило заданный порог или,*
  - *количество модифицированных страниц на очередной итерации стало меньше заданного порога.*

Миграция ВМ

Схема организации «живой» миграции Post-Copy.

- *Копируется состояние процессора на сервер-получатель. Далее ВМ в процессе миграции уже выполняется на сервере-получателе.*
- *Во время миграции сервер-источник посылает страницы памяти на сервер-получатель — в надежде, что большинство страниц будет передано на сервер-получатель до того, как ВМ начнет к ним обращаться.*

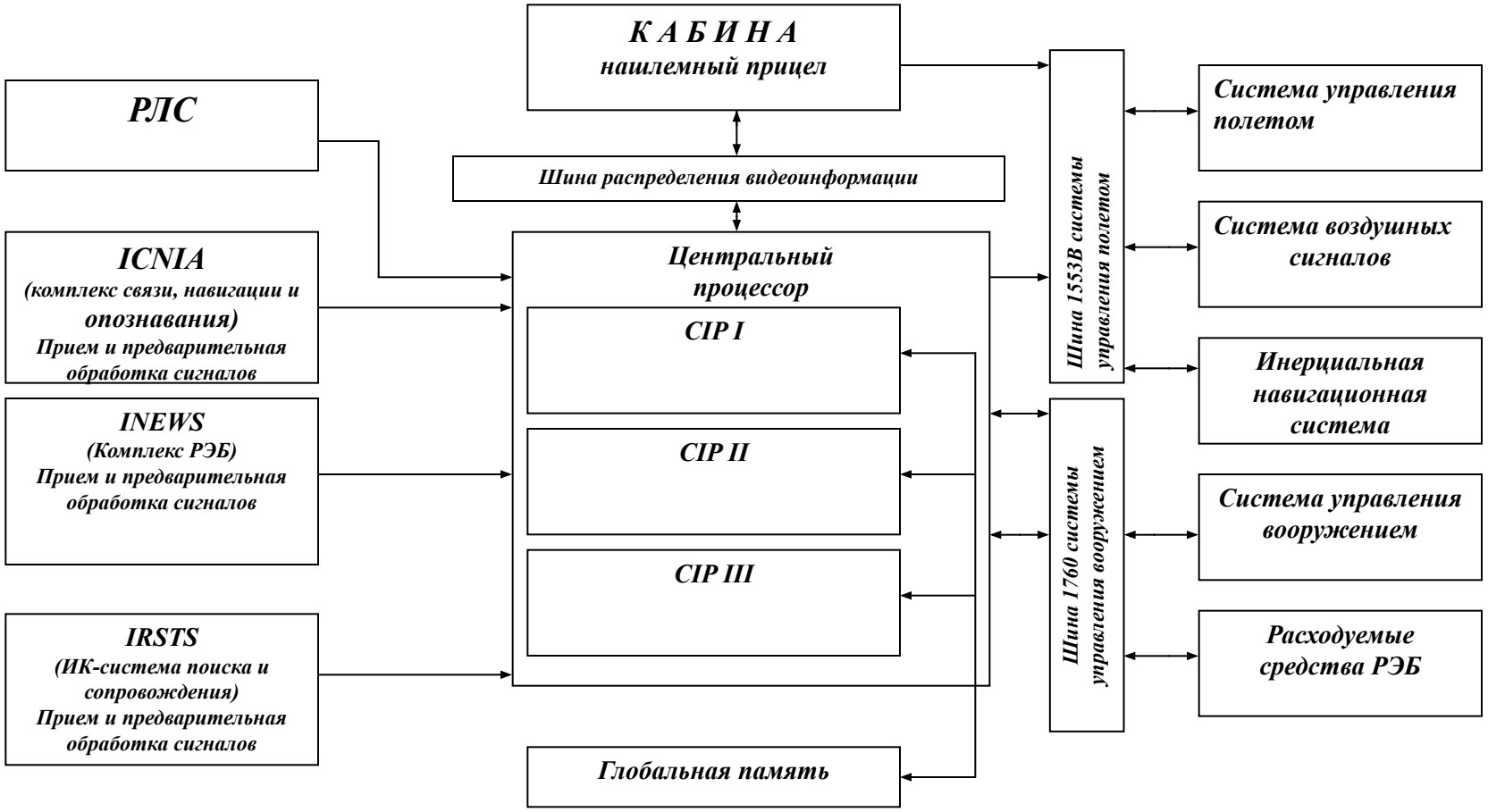
Миграция ВМ

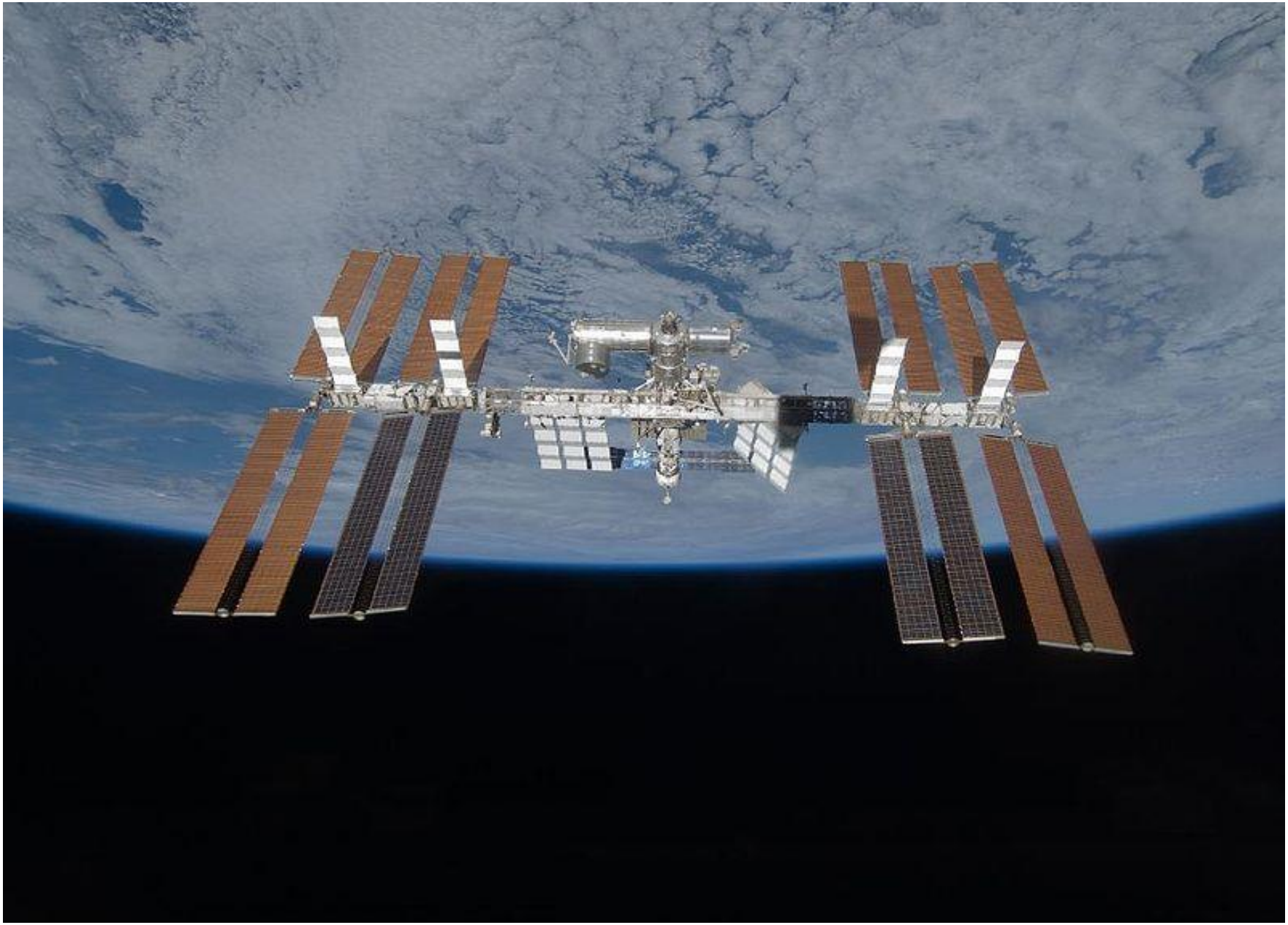
Scatter-Gather «живая» миграция.

- *Основана на Post-Copy схеме.*
- *В схеме Scatter-Gather используются посредники. Сначала ВМ переносится с сервера-источника на эти посредники, потом сервер-получатель скачивает с посредников данную ВМ.*

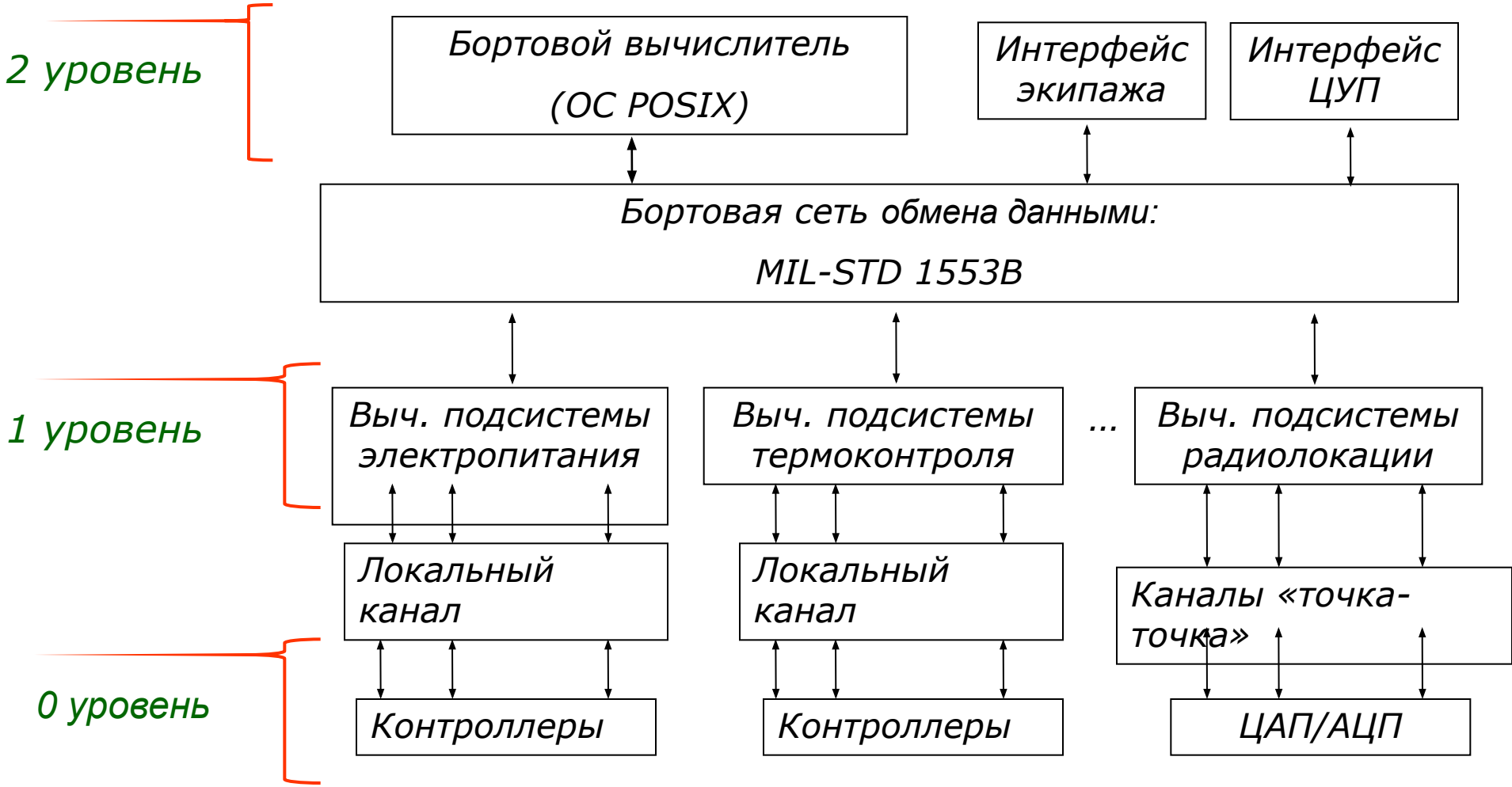
АРХИТЕКТУРЫ ВС РВ

ИУС самолета F-22





Архитектура КБО МКС

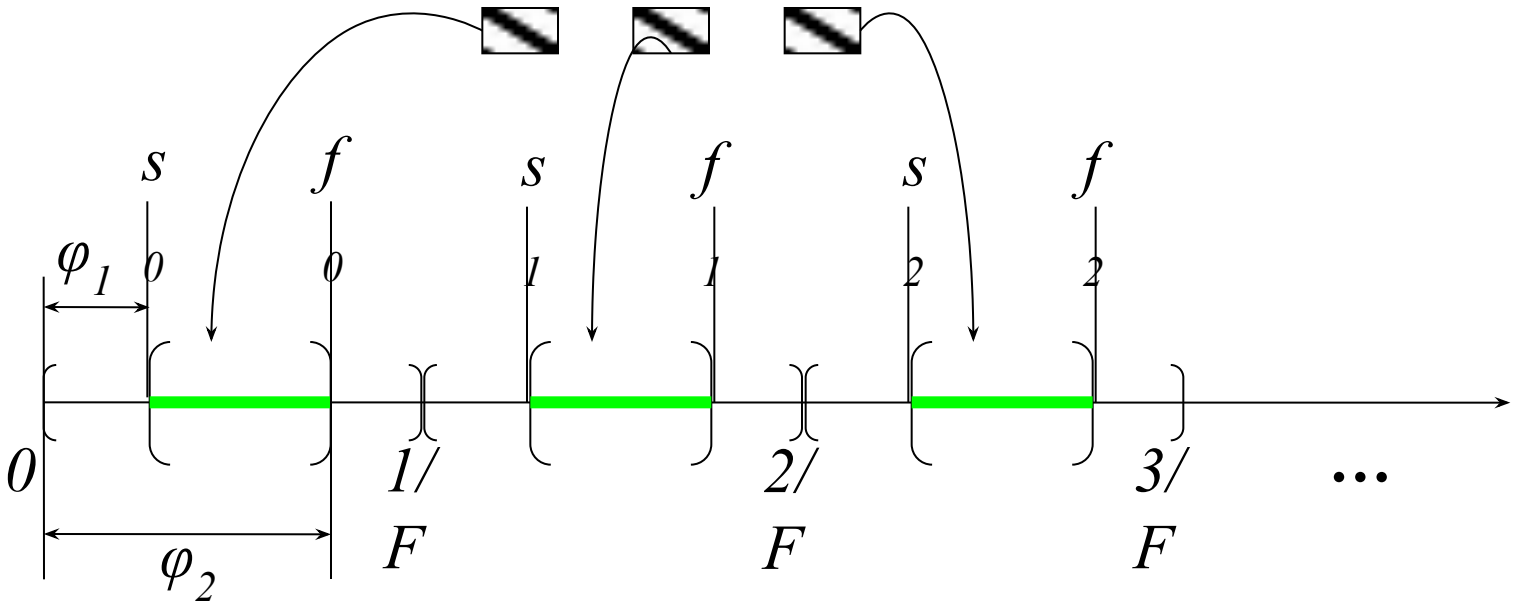


Штатные режимы КБО МКС

- *стандартный режим;*
- *режим микрогравитации для выполнения научных экспериментов;*
- *режим сближения и стыковки с транспортными кораблями;*
- *режим для выхода экипажа в открытый космос;*
- *режим выживания с отключением наименее важных экспериментов и систем;*
- *режим аварийного покидания экипажем МКС.*

Режим работы КБО

- *Набор функциональных задач (программ), которые должны выполняться в режиме.*
- *Для каждой программы: (F, φ_1, φ_2, t).*



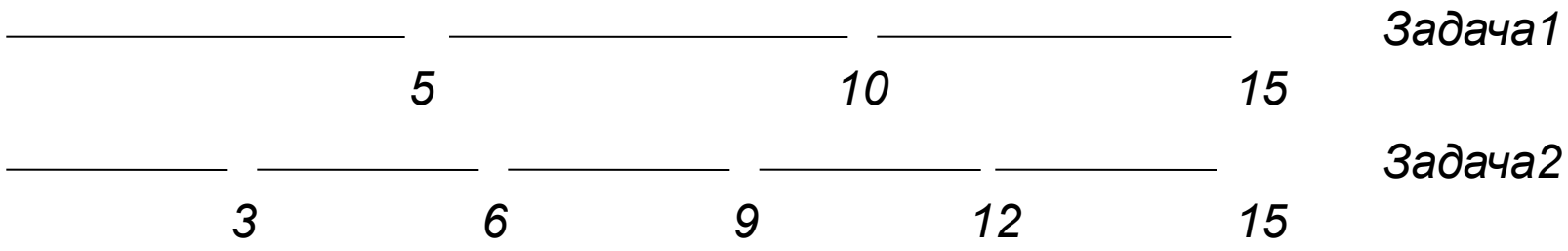
Большой цикл

Большой цикл режима работы – наименьшее общее кратное периодов задач режима.

Задача1 – период 5 тактов.

Задача2 – период 3 такта.

Большой цикл – 15 тактов.



Цель разработки новых архитектур и технологий построения КБО

- *Сокращения сроков и стоимости проектирования.*
- *Уменьшение аппаратных затрат на вычислительные и сетевые ресурсы.*
- *Повышение надежности.*

IMA (подходы к достижению целей)

- *Унификация аппаратных и программных компонентов комплекса.*
- *Перенос программ первичной обработки информации из вычислителей систем на единый бортовой вычислитель.*

Стандарты ИМА

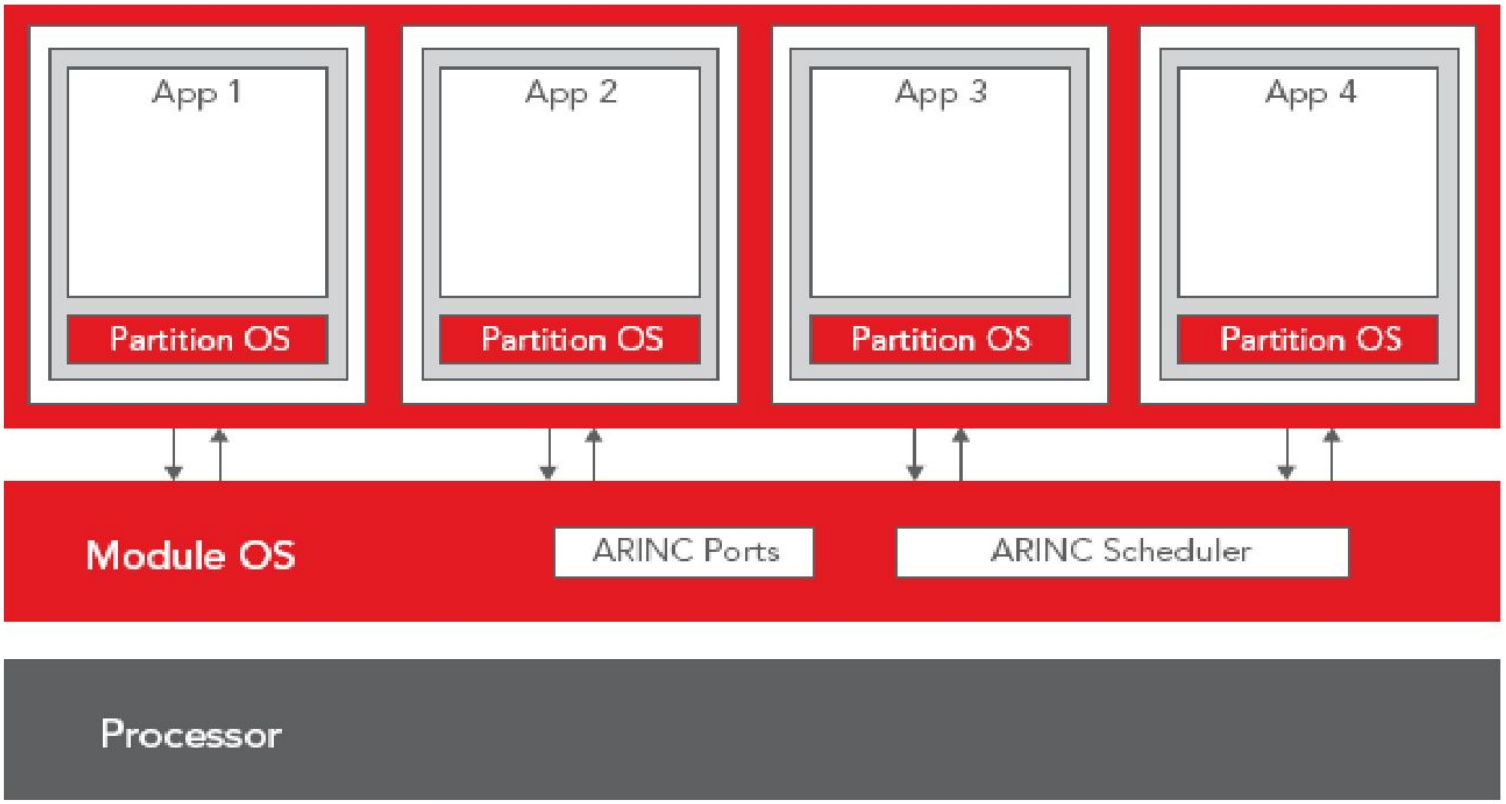
ARINC 651 *– основные принципы построения систем на основе ИМА.*

ARINC 653 *– спецификация операционных систем*

FC-AE-ASM-RT *– спецификация сети информационного обмена на основе базовых топологий FC: точка-точка, коммутируемая сеть, кольцо с арбитражем.*

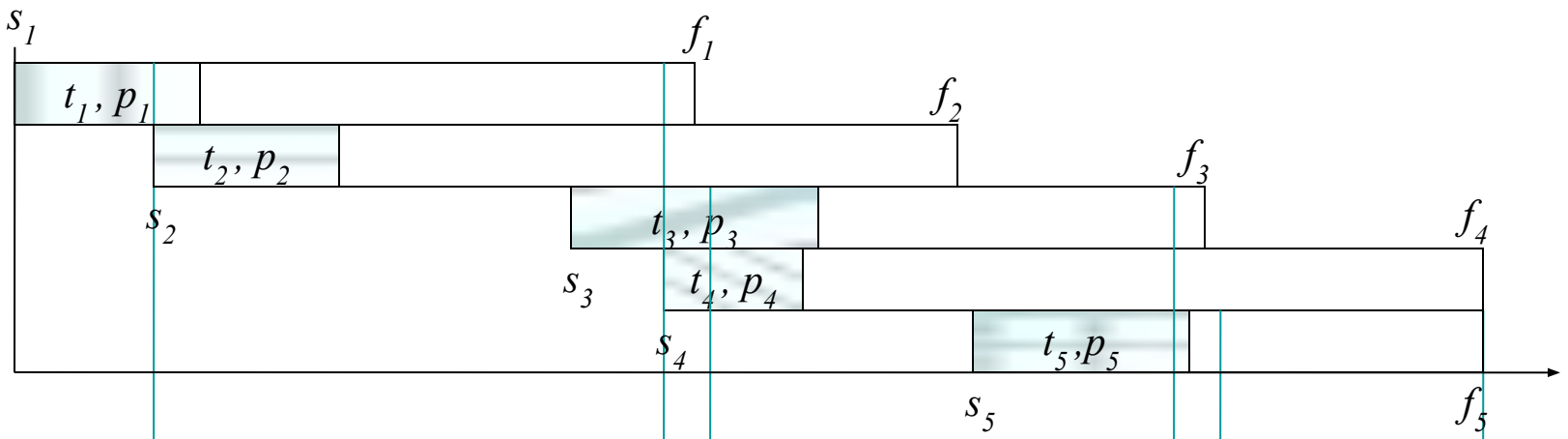
ARINC 664 (AFDX) *– спецификация сети информационного обмена на основе Ethernet.*

Операционная система VxWorks 653

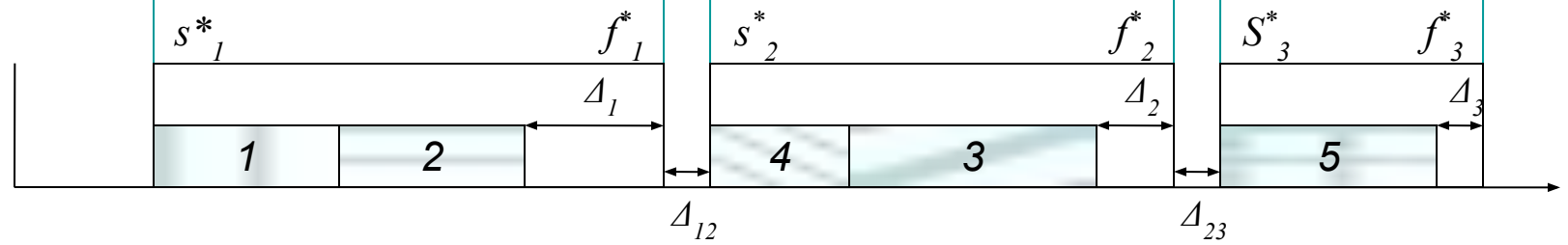


Статико-динамическое расписание (ARINC-653)

Множество процессов: $p_1=p_2=1$ $p_3=p_4=2$ $p_5=3$



Расписание:



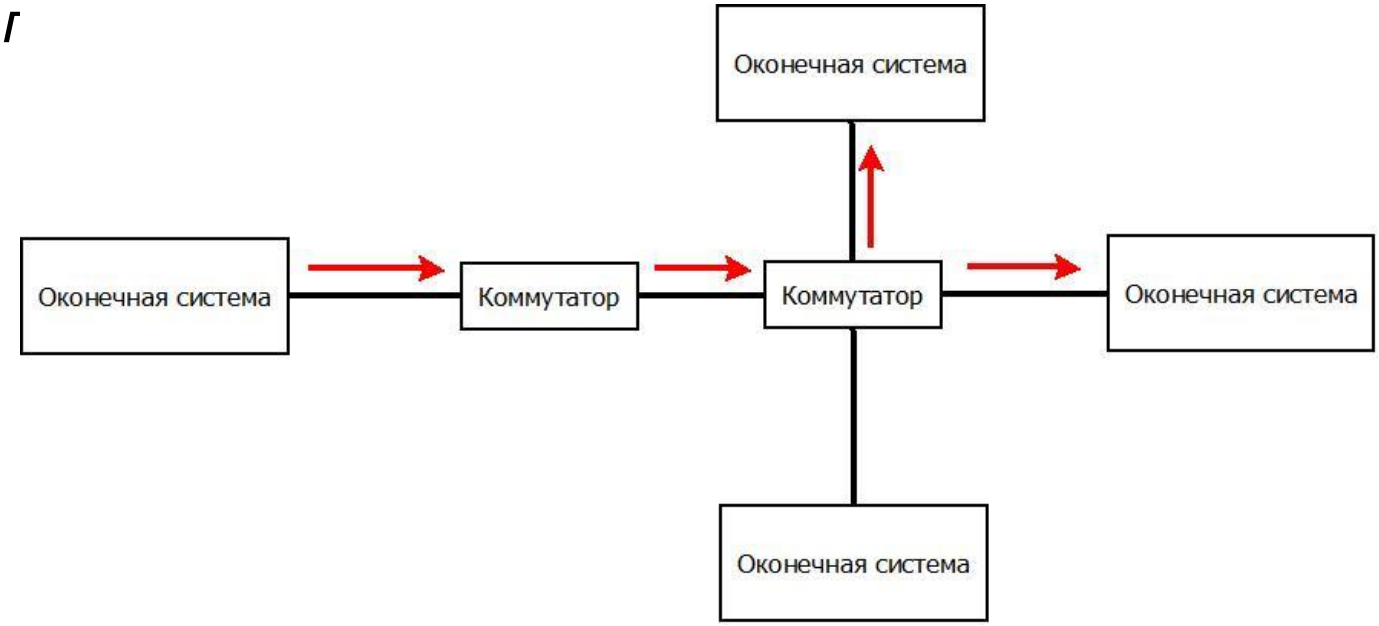
$\Delta_{ij} \le \Delta 1, \ \Delta_i \le \Delta 2$

Сети AFDX

- *Avionics Full-Duplex Ethernet (AFDX) – стандарт построения бортовых сетей на основе протокола Ethernet*
- *Компоненты:*
  - *Абоненты (бортовые подсистемы, отправители и получатели данных)*
  - *Оконечные системы – интерфейс между абонентами и сетью*

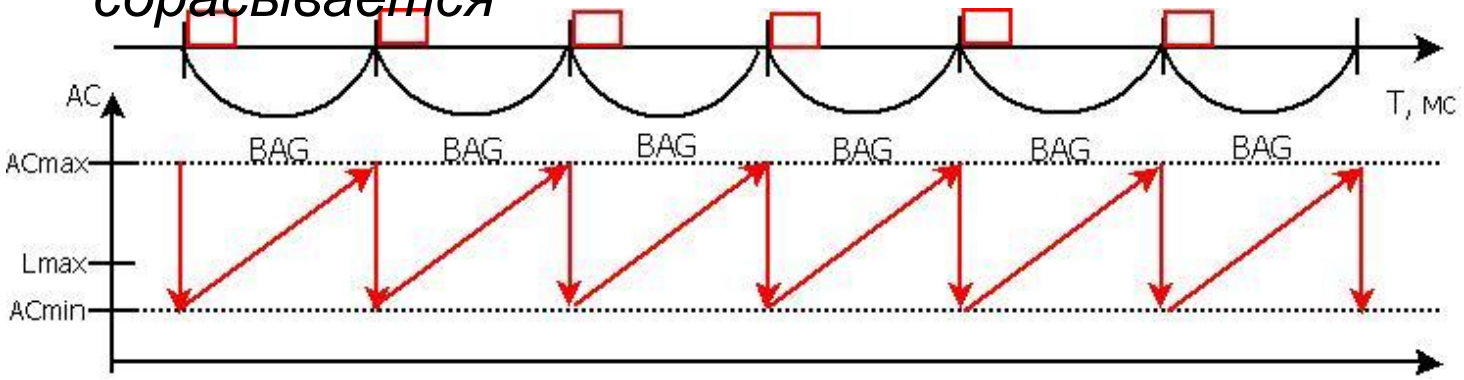
Виртуальные каналы

- *Резервирование сети – на основе виртуальных каналов*
  - *Одна оконечная система – отправитель и одна или более оконечная система – получатель*
  - *Маршрут следования кадров виртуального канала г*



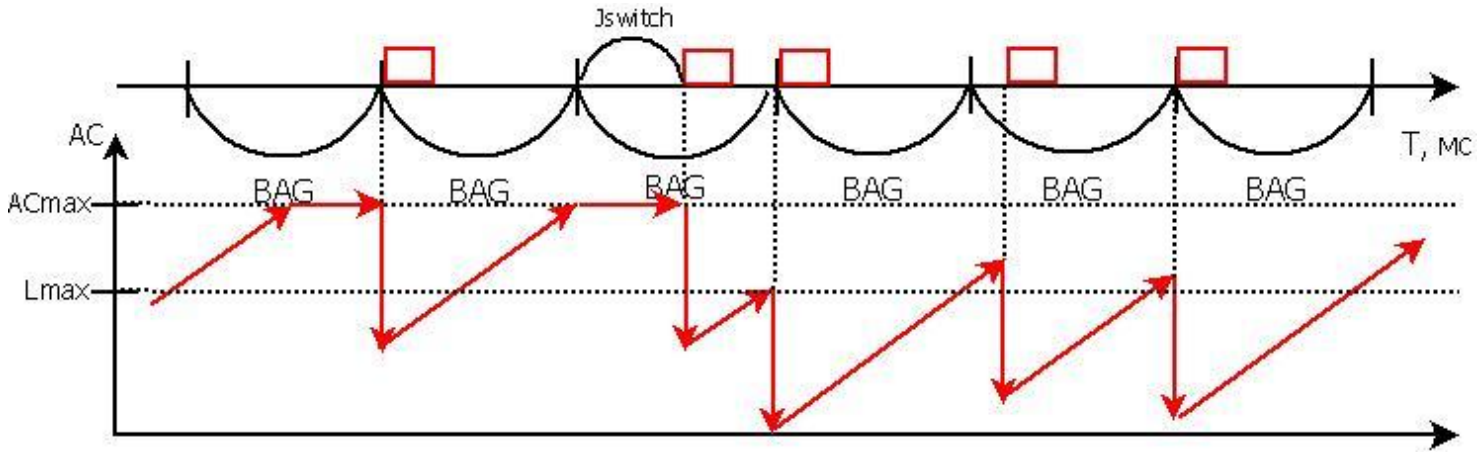
Контроль трафика на коммутаторе

- *Контроль прихода кадров на соответствие BAG и Jmax:*
  - *Производится на входном порту коммутатора*
  - *Используется алгоритм, основанный на вычислении кредита*
  - *AC – кредит, растет с течением времени до значения ACmax*
  - *При приходе кадра AC уменьшается на размер кадра, если кредита не хватает – кадр сбрасывается*

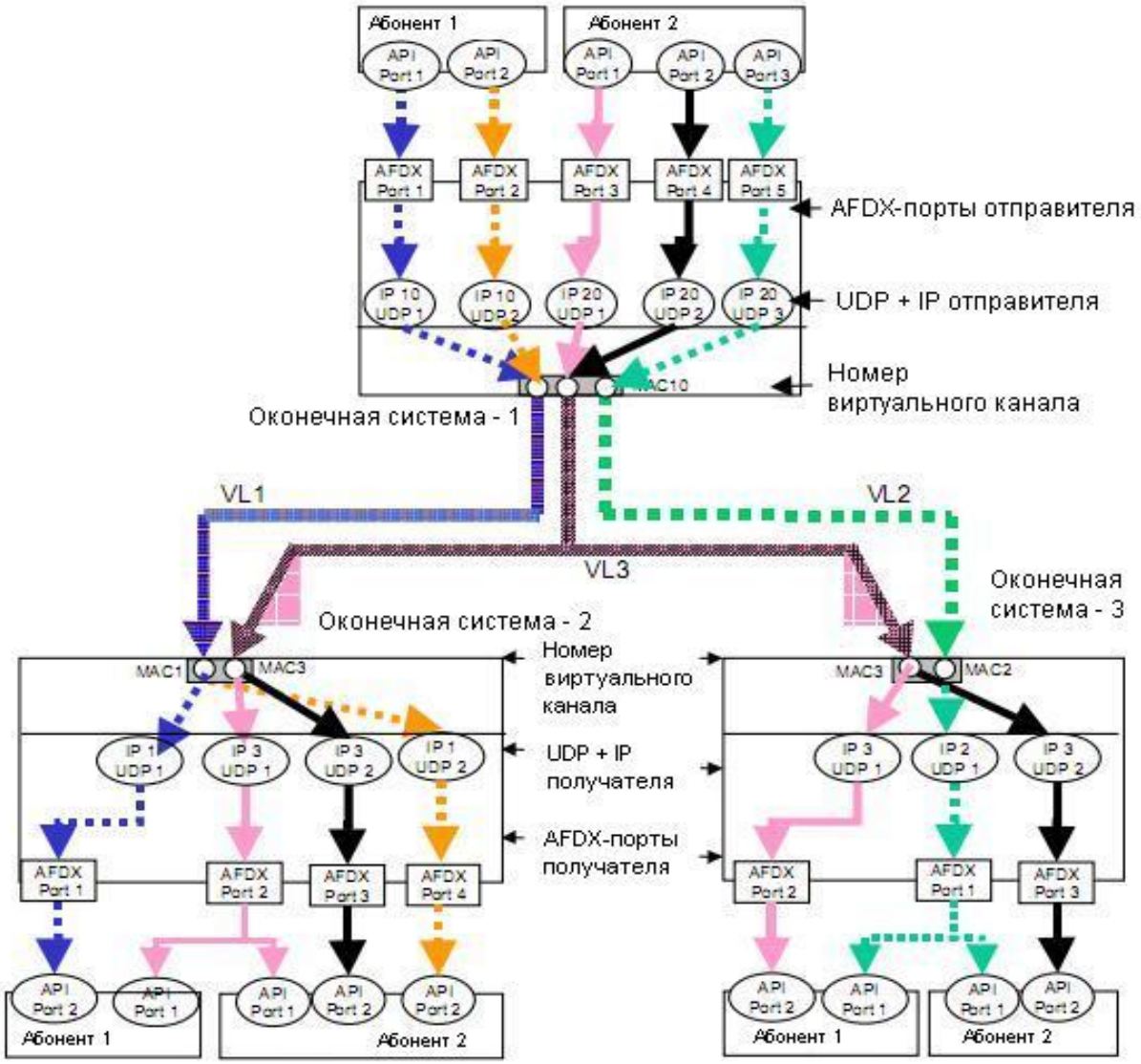


Контроль трафика на коммутаторе

- *Кредит соответствует количеству байт, которые пропускает канал*
  - *За время BAG кредит увеличивается на Lmax*
  - *ACmax – соответствует количеству байт, которое позволяет пропустить 2 кадра за (BAG – Jmax)*
  - *Случай с неравномерной передачей кадров:*



Стек протоколов



Задачи проектирования ИМА

1. Структурный синтез бортового вычислителя:

- *число вычислительных модулей,*
- *число ядер,*
- *объем памяти вычислительного модуля.*

2. Структурный синтез бортовой сети обмена:

- *число коммутаторов,*
- *топология сети,*
- *характеристики коммутаторов.*

3. Построение согласованных расписаний прикладных задач и обменов данными:

- *распределение разделов по ядрам,*
- *расписания окон на ядрах,*
- *построение виртуальных каналов и маршрутов для них, определение значений характеристик каналов.*

Подсистема радиолокации

(федеративная архитектура)

1. Оцифровка и нормализация данных снимаемых с ФАР.

Слов данных: K . Период опроса: 1/a·B .

2. БПФ.

K опер. БПФ на L точек. Период выполнения: L/a·B.

3.Получение оценки взаимно-спектральной матрицы.

L ·n опер. внеш. прозв. век. размера K. Период выполнения: n·L/a·B.

4. Нахождение собственных значений и векторов или обращение взаимно-спектральных матриц.

L матриц размера $K^x K$. Период выполнения: n·L/a·B.

5. Нахождение угловых координат.

Результат: 3·M переменных . Период выдачи: n·L/a·B.

Интегрированная модульная архитектура

Подсистема радиолокации:
1. Оцифровка и нормализация данных снимаемых с ФАР.

K слов в период = 1/a·B

Бортовой вычислитель:
2. БПФ.
3.Получение оценки взаимно-спектральной матрицы.
4. Нахождение собственных значений и векторов или обращение взаимно-спектральных матриц.
5. Нахождение угловых координат.

Проблема увеличение потока данных в бортовой сети обмена

Тип архитектуры	Кол-во СД	Период выдачи
федеративная	$3 \cdot M$	$n \cdot L/a \cdot B$
ИМА	K	$1/a \cdot B$

$32 \le K \le 2048 \quad 1 \le M \le 100 \quad n > 100 \quad 32 \le L \le 2048$

Поток данных в бортовой сети обмена увеличивается :

в $10^3 – 10^5$ раз

Проблема переходного процесса в бортовой сети обмена при переключении режима работы КБО

Проблема использования универсальных процессоров

Спецпроцессоры в разы эффективнее по критерию производительность/аппаратные затраты».

Применение спецпроцессоров в бортовом вычислителе:

проблема настройки на размер обрабатываемы массивов данных

Компромисс между спецпроцессорами и универсальными процессорами:

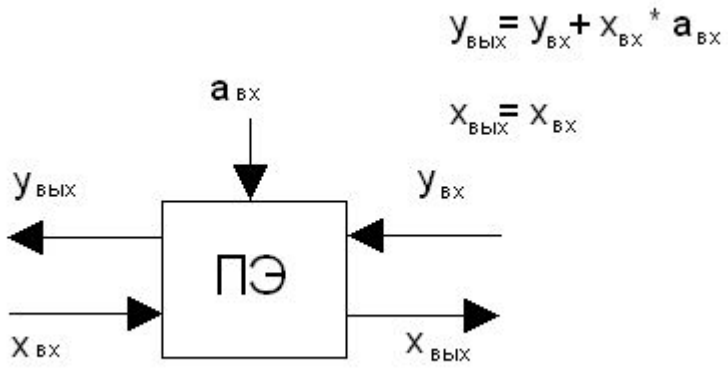
процессоры цифровой обработки сигналов (DSP)

Пример спецпроцессоров
Систолические процессоры

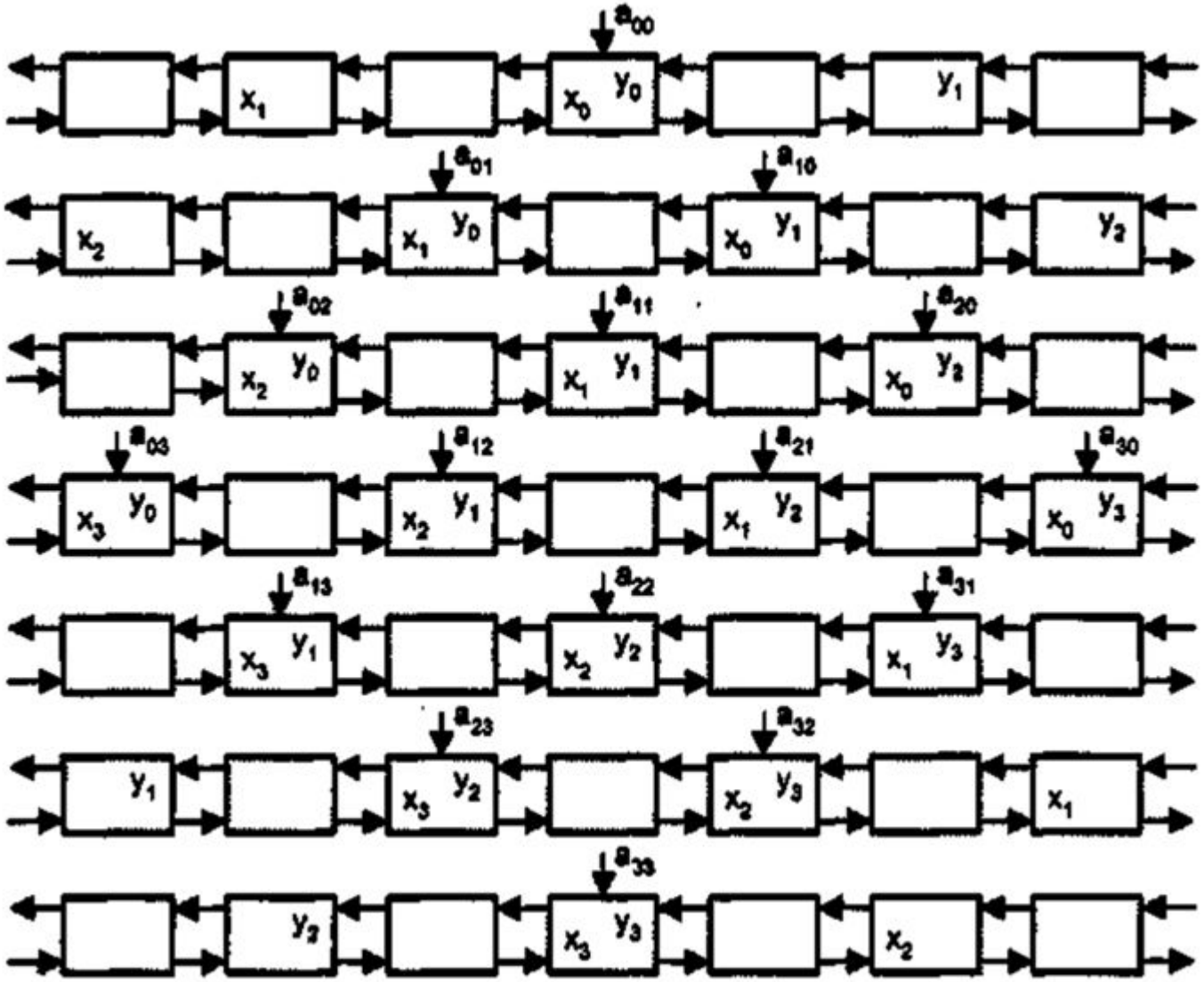
Систолическая структура *— это однородная вычислительная среда из процессорных элементов (ПЭ), совмещающая в себе свойства конвейерной и матричной обработки и обладающая следующими особенностями:*

- *вычислительный процесс в систолических структурах представляет собой непрерывную и регулярную передачу данных от одного ПЭ к другому без запоминания промежуточных результатов вычисления;*
- *каждый элемент входных данных выбирается из памяти однократно и используется столько раз, сколько необходимо по алгоритму, ввод данных осуществляется в крайние ПЭ матрицы;*
- ***образующие систолическую структуру ПЭ однотипны и каждый из них может быть менее универсальным, чем процессоры обычных многопроцессорных систем****;*
- *потоки данных и управляющих сигналов обладают регулярностью, что позволяет объединять ПЭ локальными связями минимальной длины;*
- *алгоритмы функционирования позволяют совместить параллелизм с конвейерной обработкой данных;*
- *производительность матрицы можно улучшить за счет добавления в нее определенного числа ПЭ, причем коэффициент повышения производительности при этом линеен.*

Пример спецпроцессоров
Систолические процессоры



Пример спецпроцессоров
Систолические процессоры



Проблема использования языков высокого уровня

Операция	N=16	N=32	N=64	N=128	$N \to \infty$
Скалярное произведение векторов	1402 164 8.549	2682 292 9.185	5242 548 9.566	10362 1060 9.775	**10.0**
Умножение вектора на скаляр	1322 164 8.061	2570 292 8.801	5066 548 9.245	10058 1060 9.489	**9.5**
Поэлементное умножение векторов	1268 166 7.639	2452 294 8.340	4820 550 8.764	9556 1062 8.998	**9.25**
Внешнее произведение векторов	5996 676 8.870	23180 2340 9.906	91340 8740 10.451	362828 33828 10.726	**11.0**
Умножение матриц	116502 11554 10.083	891318 82214 10.841	6972150 590374 11.810	-	**12.5**
Транспонирование матриц	10622 1242 8.552	41662 4506 9.246	165182 17178 9.616	657982 67098 9.806	**10.0**

Какие программы не надо переносить в бортовой вычислитель?

1. *Разница между временем выполнения на бортовом вычислителе и периодом выполнения программы мала.*
2. *Подсистема используется во всех режимах работы КБО.*

Для выполнения программы в реальном времени ресурсы бортового вычислителя выделяются ей монопольно.

Увеличение аппаратных ресурсов, используемых программой, в 8-30 раз.

---

3. *Перенос программы на бортовой вычислитель приводит к увеличению потока данных в бортовой сети обмена.*

Аппаратные затраты на увеличение пропускной способности бортовой сети могут быть больше экономии затрат на вычислительные ресурсы.

Развитие концепции ИМА

- *Разработка аппаратно и программно масштабируемых спецпроцессоров работающих под ОС удовлетворяющей ARINC 653.*
- *Повышение качества компиляторов DSP.*
- *Разработка бортовых сетей обмена, которые позволяют удалять и добавлять виртуальные каналы без изменения характеристик работающих каналов.*

- *Классы P и NP (формальные определения, примеры).*
- *Массовая задача, индивидуальная задача, частная задача (подзадача).*
- *Определение полиномиальной сводимости.*

- *Новикова Н.М. Основы оптимизации. Курс лекций. М.: МГУ, 1998. – 65с.*
- *Гэри М., Джонсон В. Вычислительные машины и труднорешаемые задачи. – М.: Мир, 1982. – 416с.*