ПАРАЛЛЕЛЬНЫЕ ФАЙЛОВЫЕ СИСТЕМЫ

к. т. н. Аладышев Олег Сергеевич

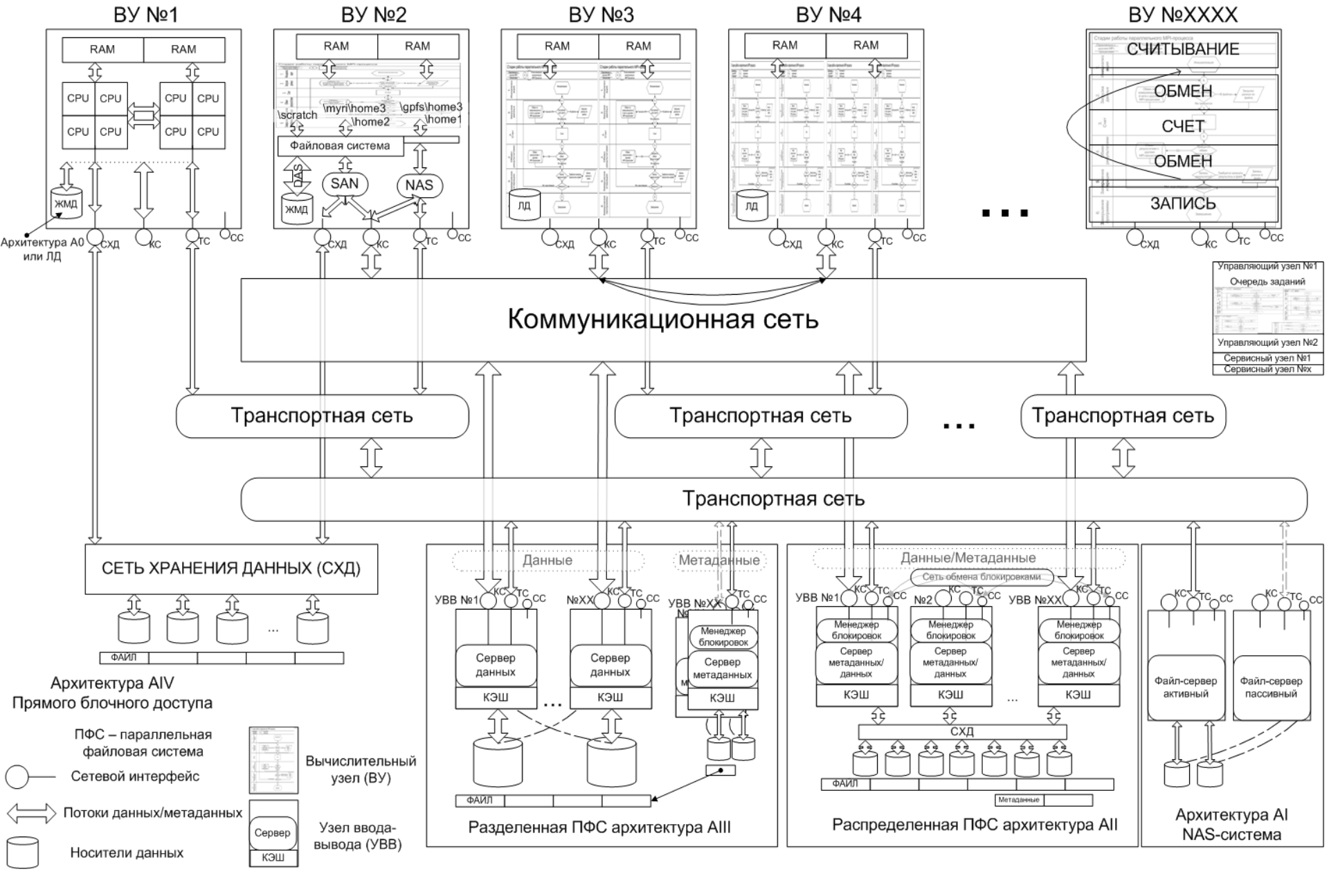
30 июня 2012

Лекция
Летняя суперкомпьютерная академия МГУ

Структура лекции

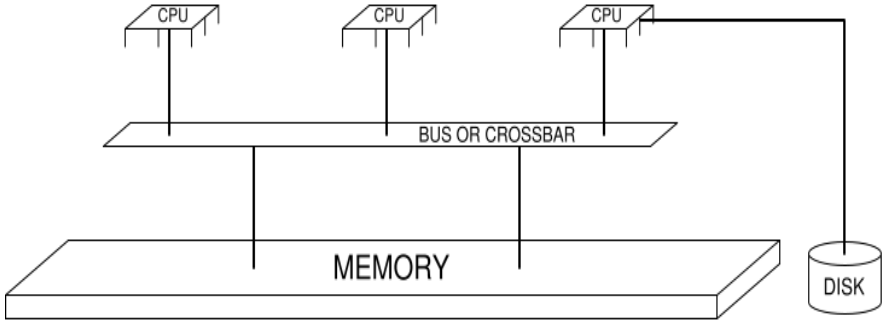
- Иерархия памяти суперкомпьютера (СК)
- Внешняя память, системы хранения данных (СХ), внешние СХ (ВСХД)
- Технические аспекты подключения внешний памяти
- Иерархия памяти ВСХД
- Для чего нужны ВСХД
- Общие свойства ввода-вывода (IO) в высокопроизводительных задачах
- Проблема контрольной точки (КТ)
- Файловые системы (ФС) СК. Цели, задачи, требования к ФС
- Параллельные ФС (ПФС). Принципы работы.
- Механизмы разделения доступа к данным в ПФС.
- Классификация систем хранения СК
- Сравнение ПФС
- Заключение

Кластерный вычислительный комплекс

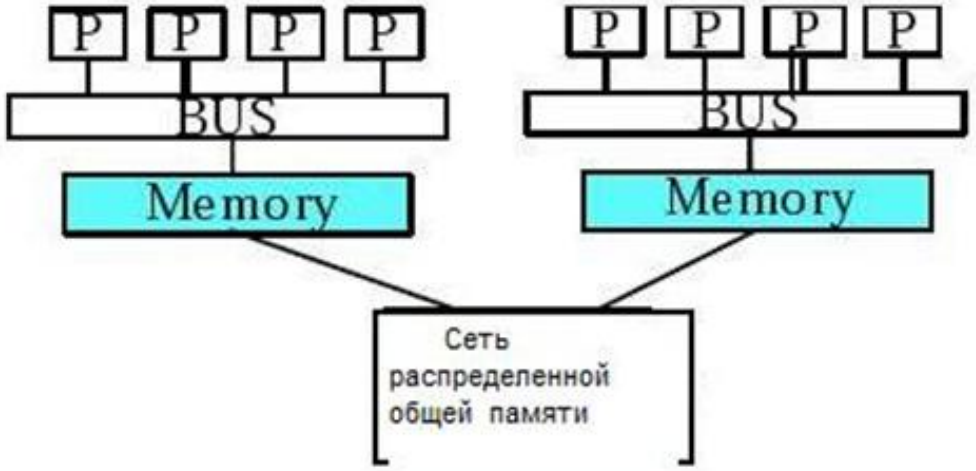


Иерархия памяти (1)

SMP - Симметричный доступ

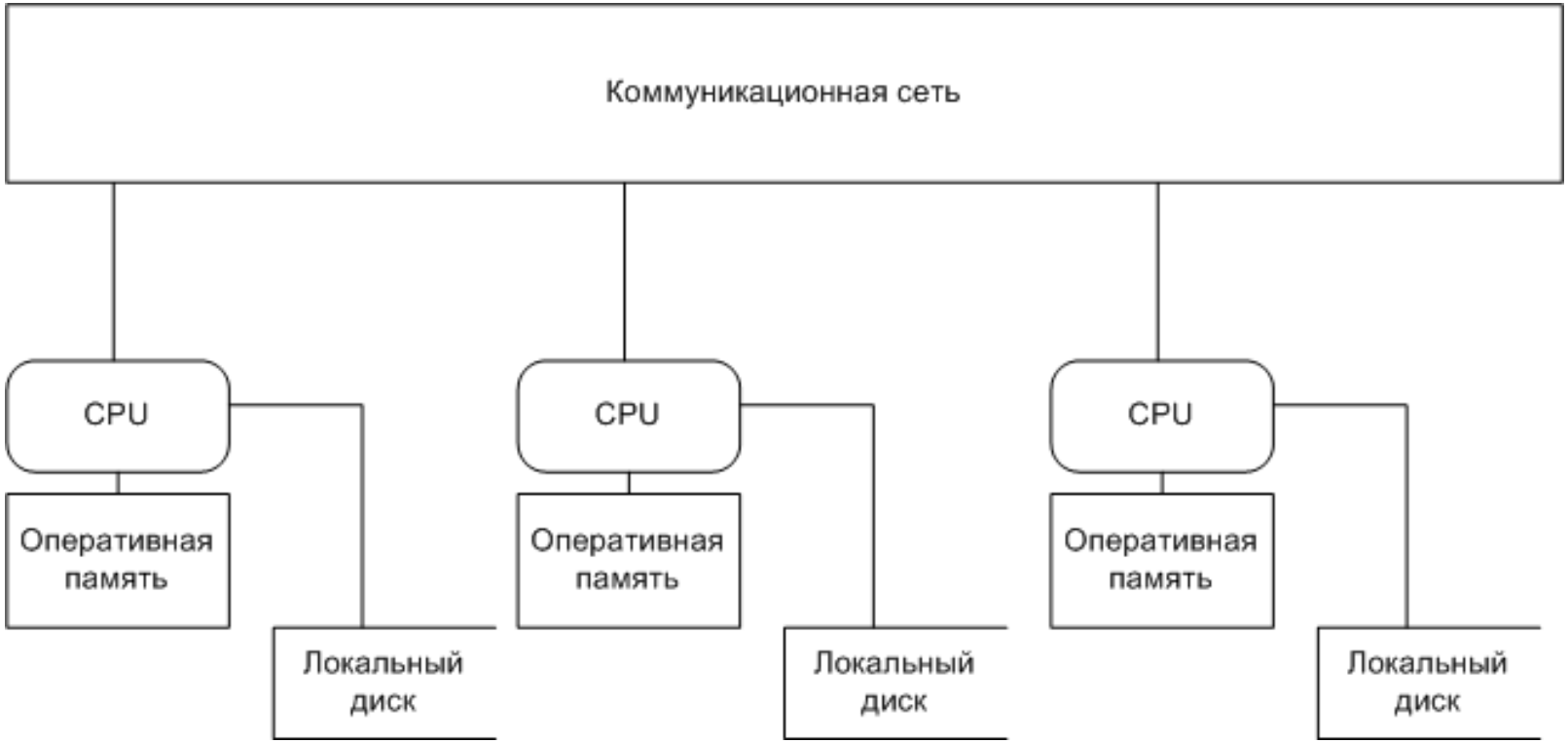


NUMA - Неравномерный доступ



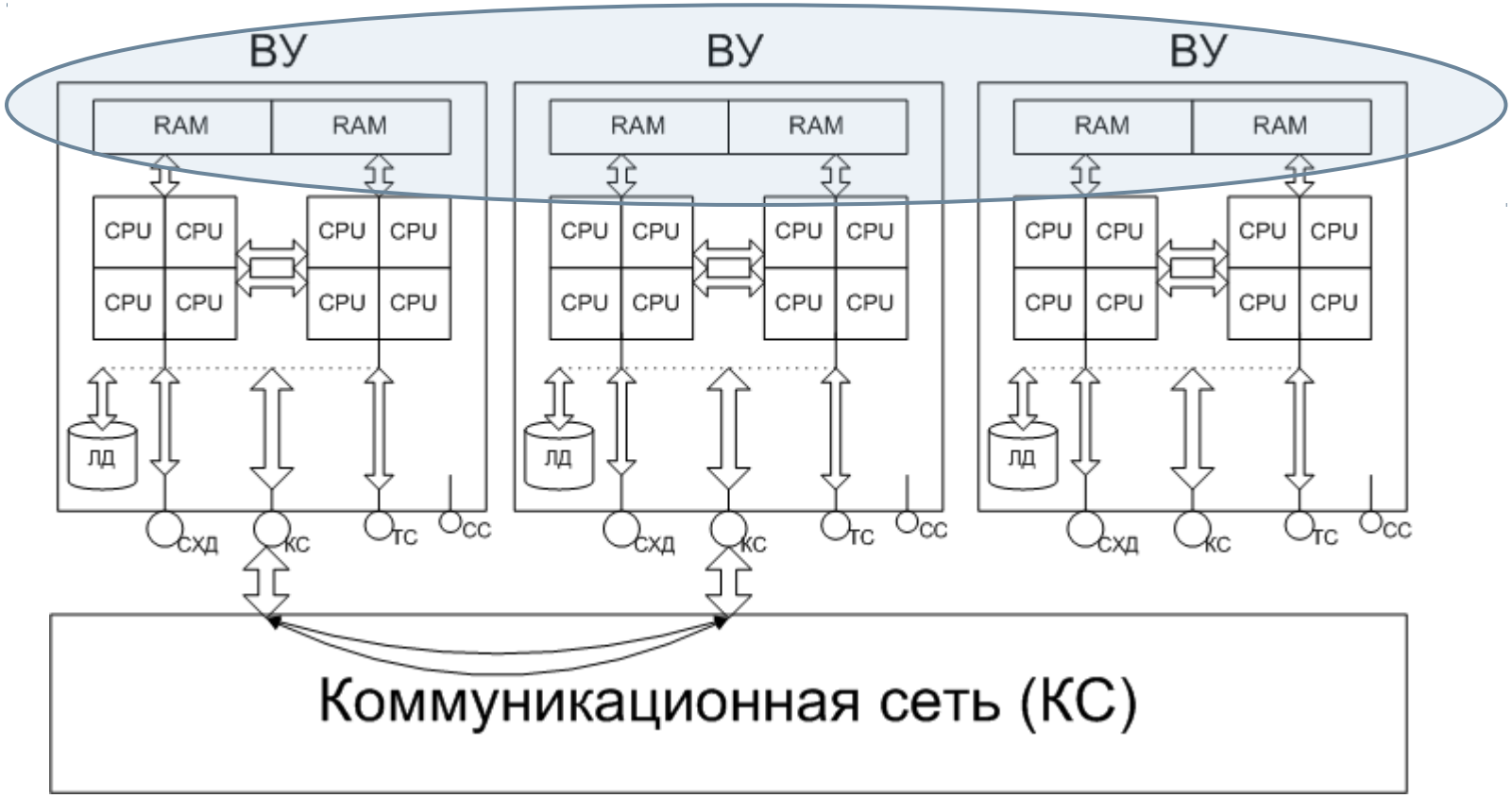
Иерархия памяти (2)

Cluster – Распределённая память



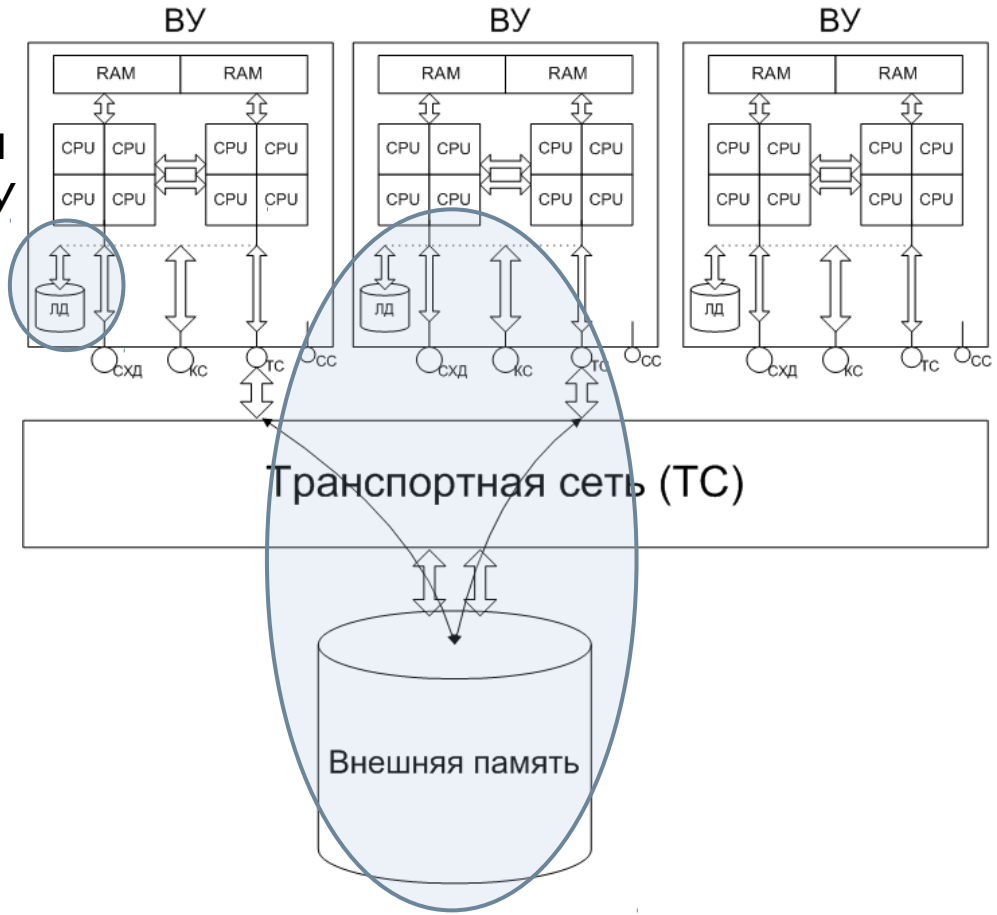
Иерархия памяти (3)

Решающее поле СК (Кластер) – память всех вычислительных узлов (ВУ)

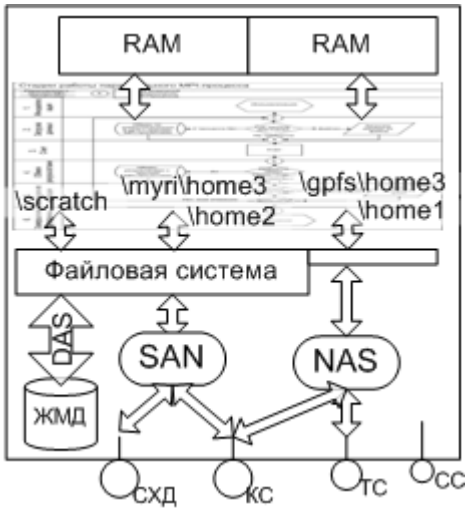


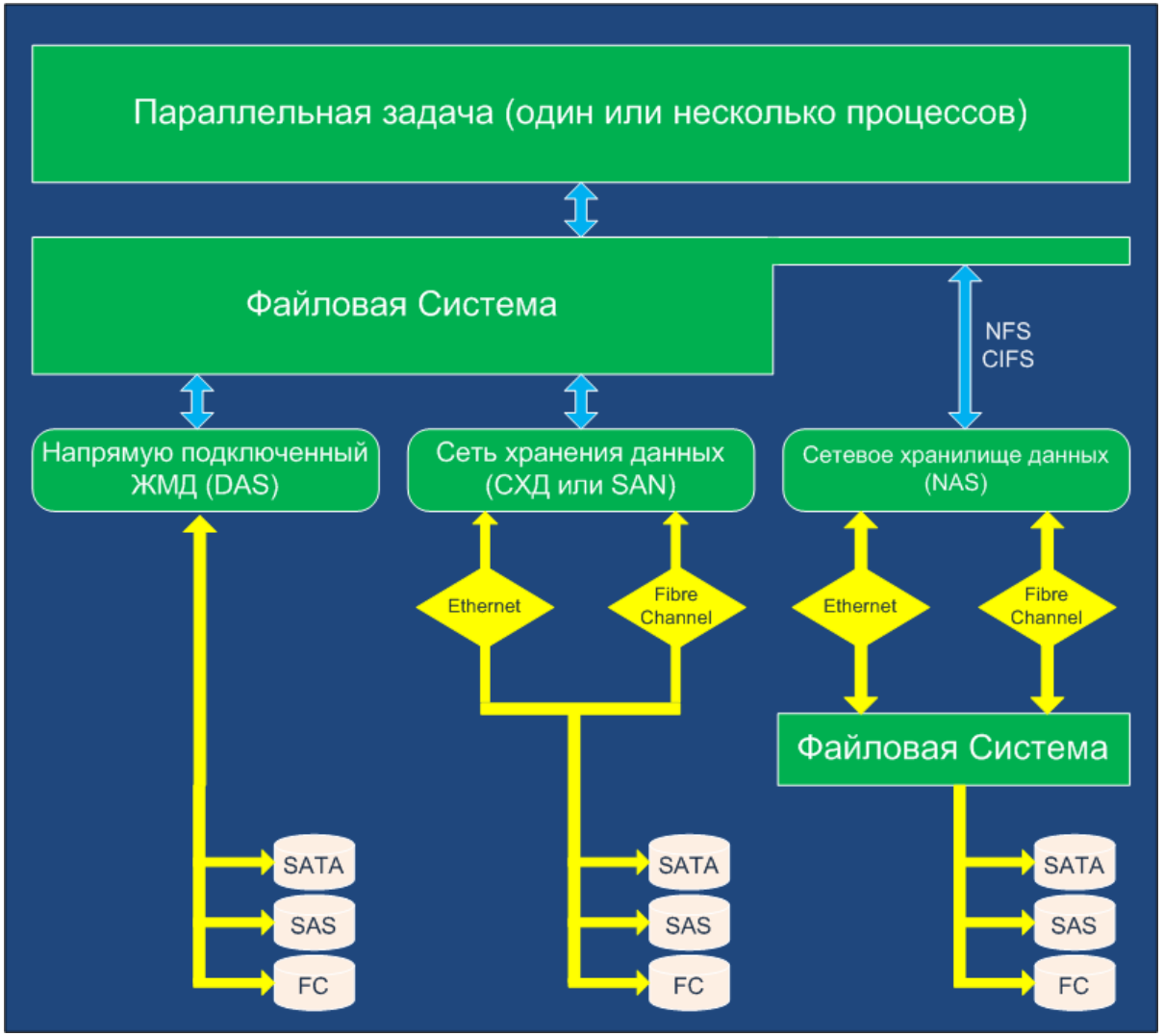
Внешняя память

Старое определение – ЖМД в компьютере
IO – ввод-вывод на внешнюю память
Система хранения данных (СХ) – аппаратно-программный комплекс, обеспечивающий доступ к единому адресному пространству данных.
ВСХД – внешняя система хранения данных.
Носители данных логически вынесены за пределы ВУ

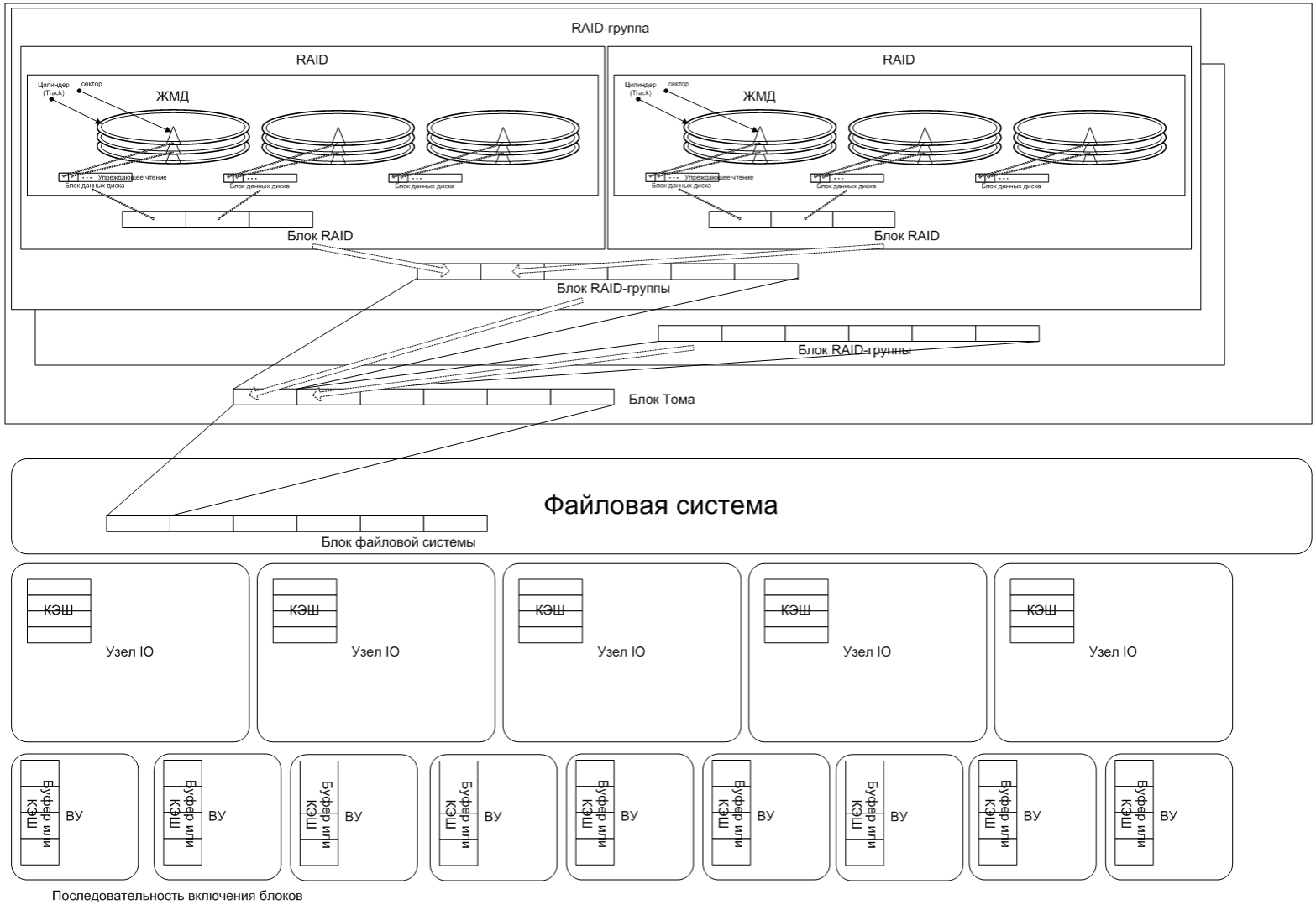


Технические аспекты организации доступа к данным ВСХД





Дисковая подсистема и иерархия блоков данных



Последовательность включения блоков нижнего уровня (диск) в верхний показана условно.

Правила POSIX

Где нужна внешняя память и ВСХД

1. IO нужен самим параллельным приложениям
   a) Декомпозиция данных. Ввод, распределение по решающему полю
   b) Счёт и обмен данными между параллельными процессами.
   c) Вывод временных и/или конечных результатов.

2. Обеспечение работы СК
   a) Профайлы пользователей, системный и пользовательское ПО
   b) Обеспечение надёжности СК, гарантированности результата, контрольная точка (КТ)

Декомпозиция данных в параллельной задаче

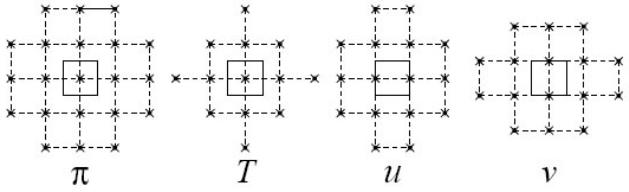


Figure 1. Dependence stencils for the Helmholtz equation right hand side generation.

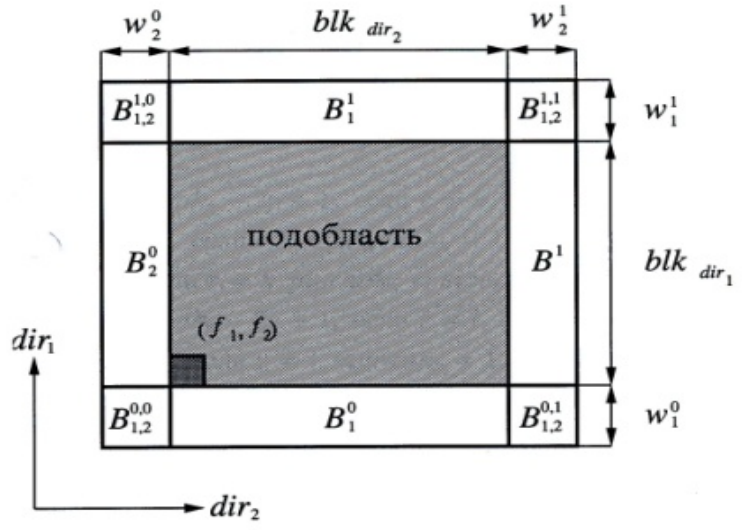
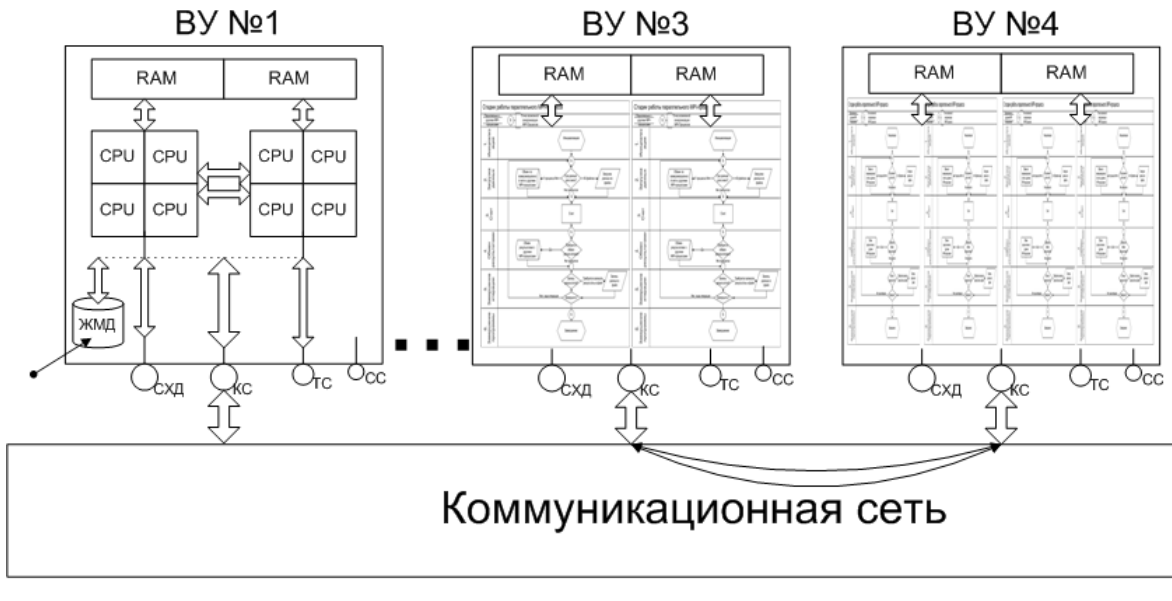


Рис. 2. Подобласть и ее внешние границы (случай $m = 2$).

ИВМ РАН
д.ф.-м.н., професор Михаил Толстых
к.ф.-м.н. Валерий Глухов

Обмен данными между параллельными процессами



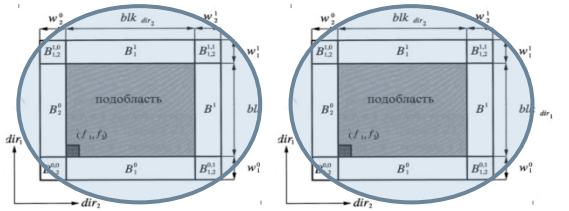


Рис. 3. Подобласть и ее внешние границы (случай m

Рис. 3. Подобласть и ее внешние границы (случай $m = 2$).

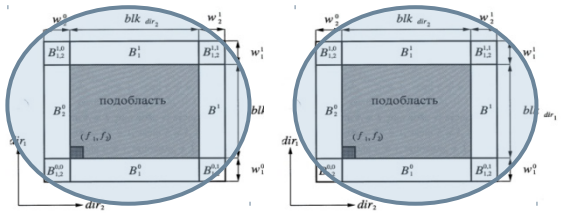
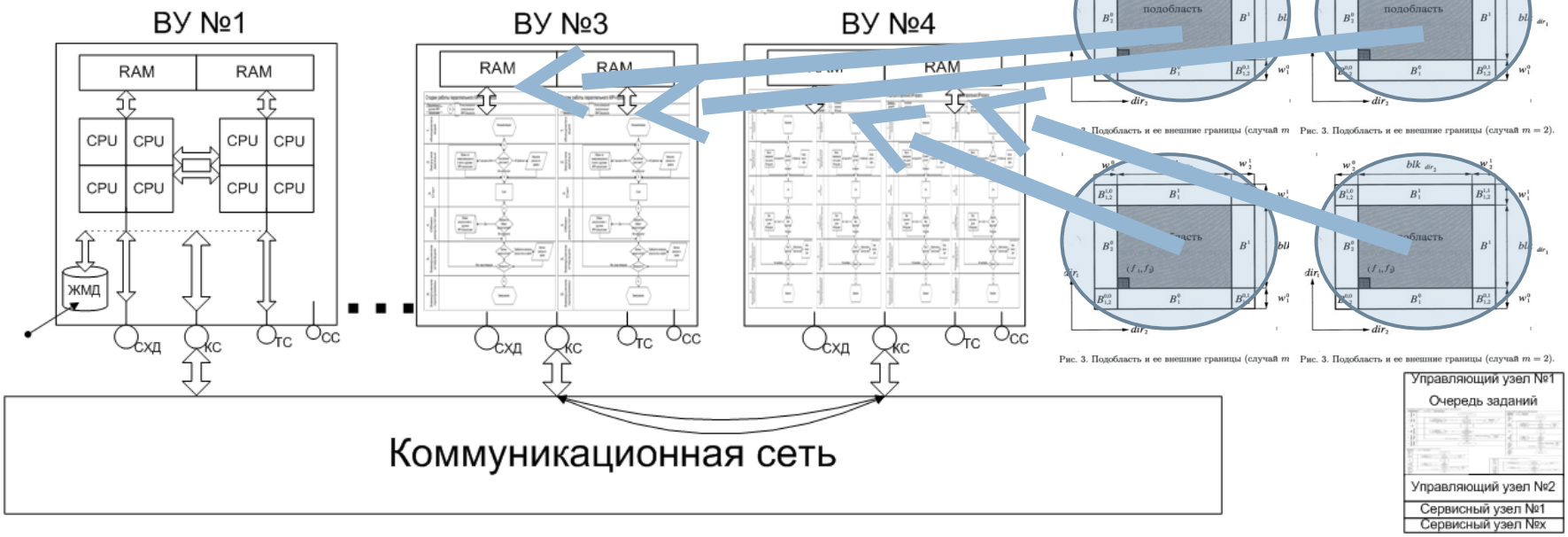


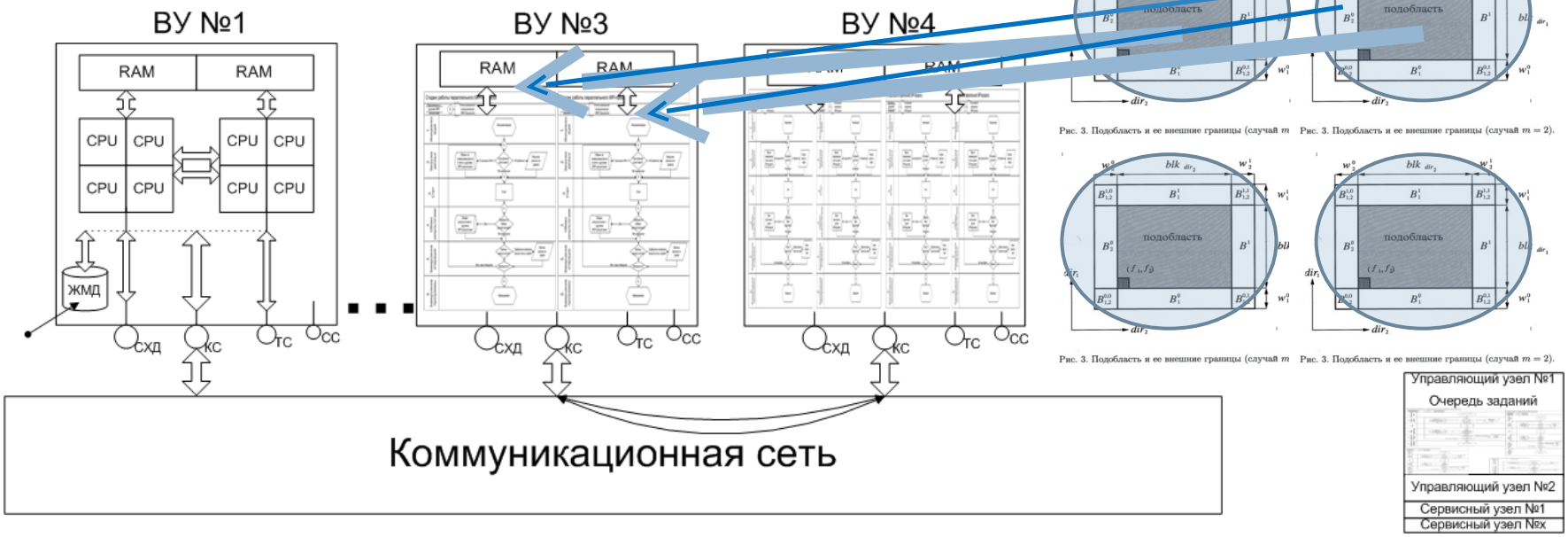
Рис. 3. Подобласть и ее внешние границы (случай m

Рис. 3. Подобласть и ее внешние границы (случай $m = 2$).

Обмен данными между параллельными процессами (1)



Обмен данными между параллельными процессами (2)



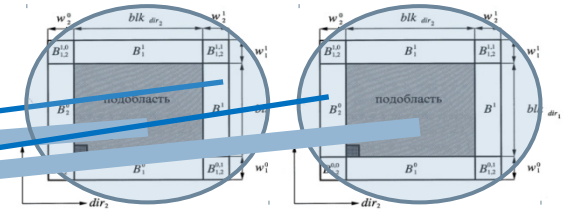
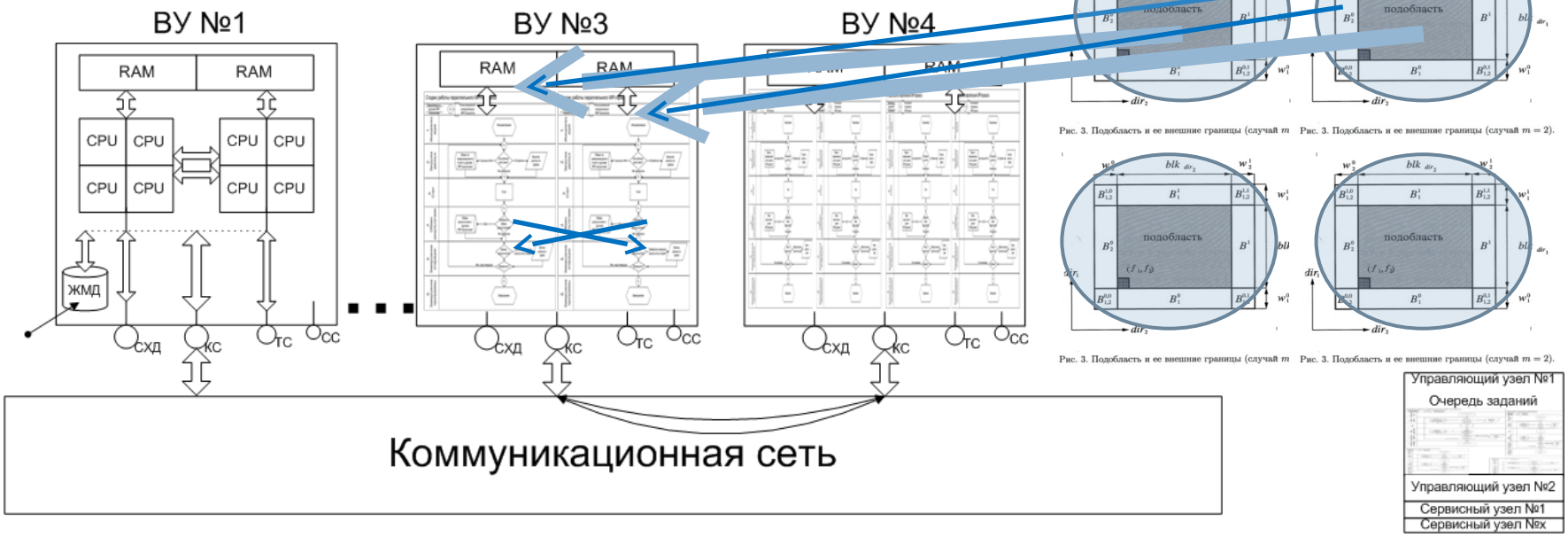


Рис. 3. Подобласть и ее внешние границы (случай m = 2).

Рис. 3. Подобласть и ее внешние границы (случай m = 2).

Обмен данными между параллельными процессами (3)



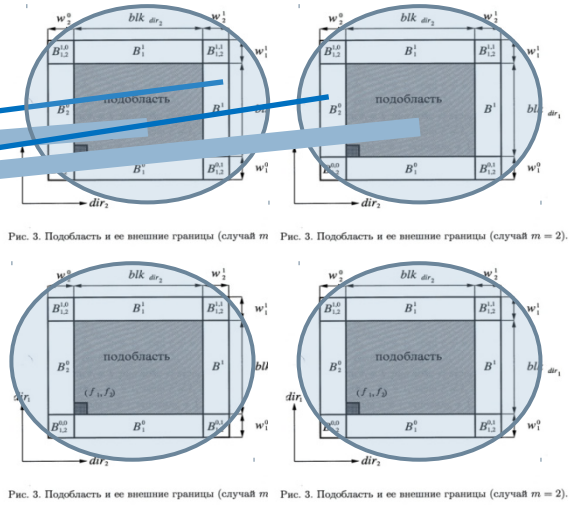
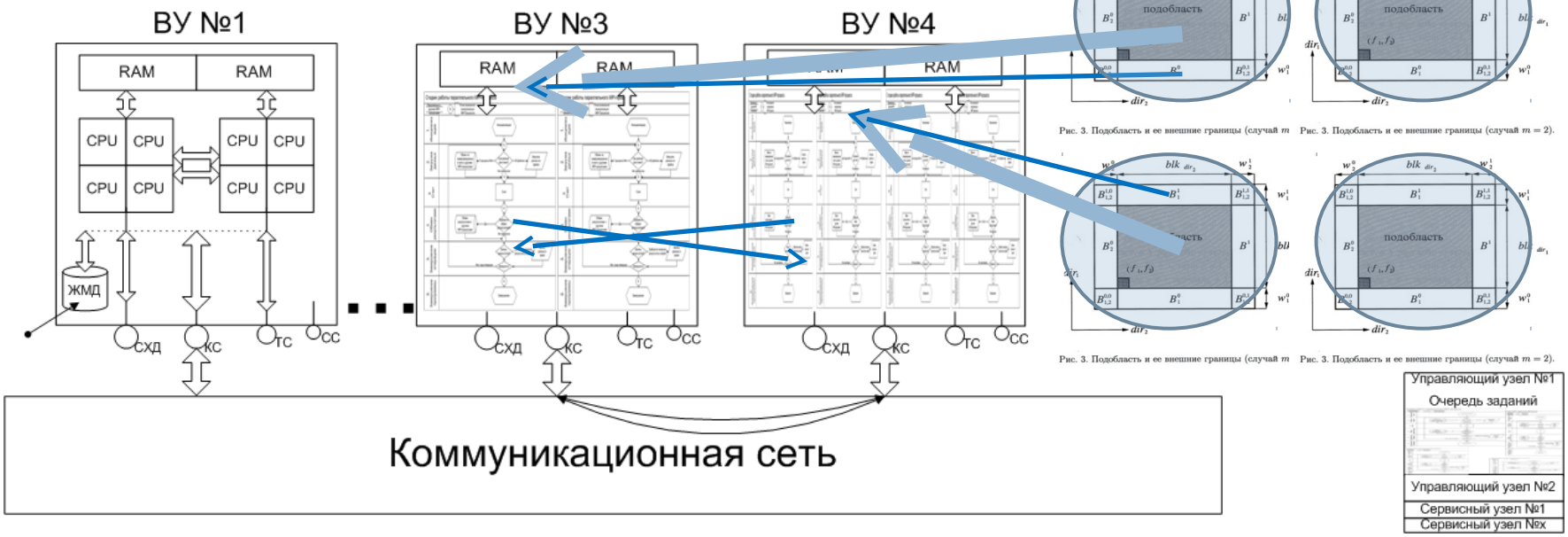


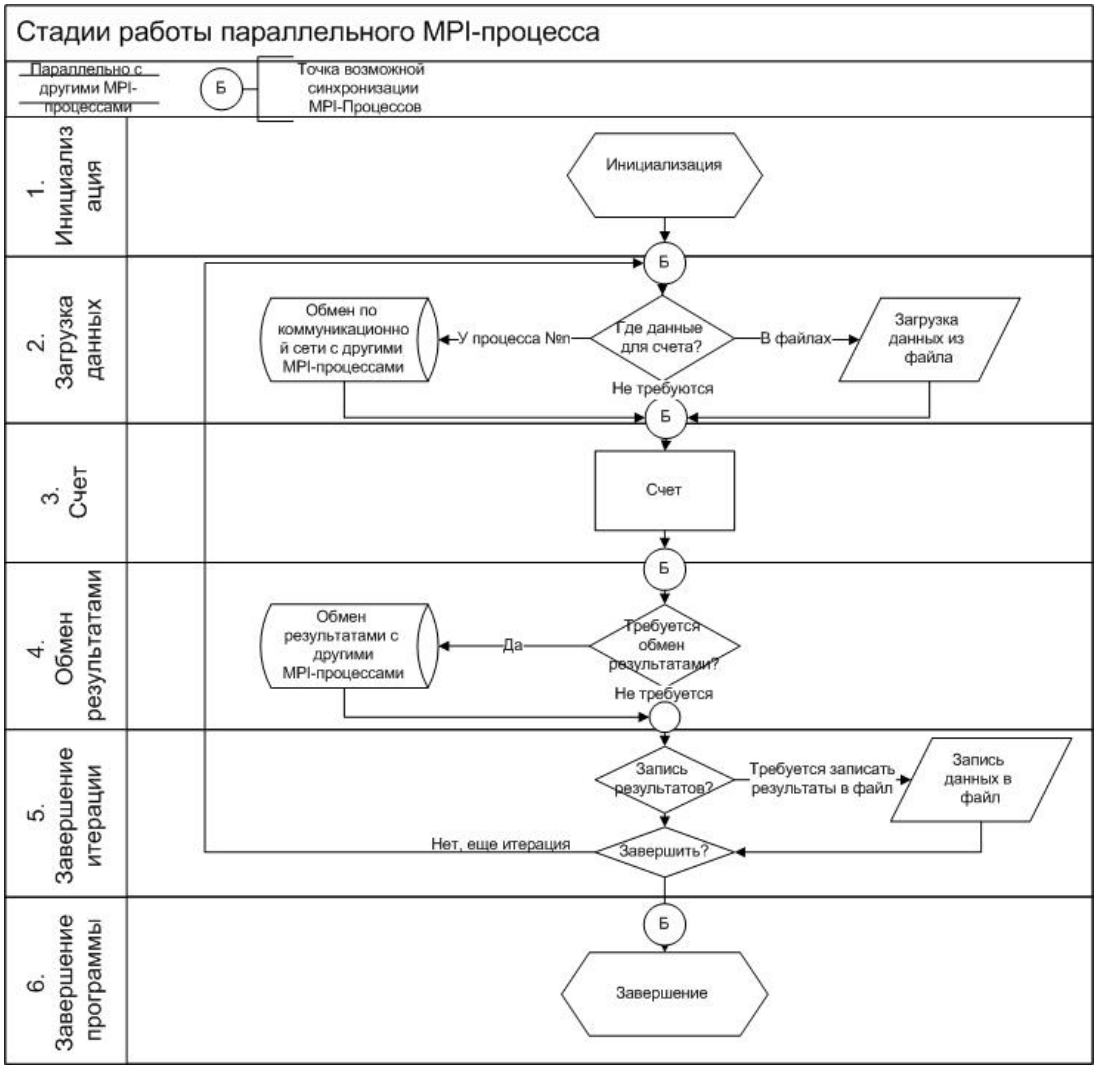
Рис. 3. Подобласть и ее внешние границы (случай m = 2).

Рис. 3. Подобласть и ее внешние границы (случай m = 2).

Обмен данными между параллельными процессами (4)



Общее в структуре параллельных задач



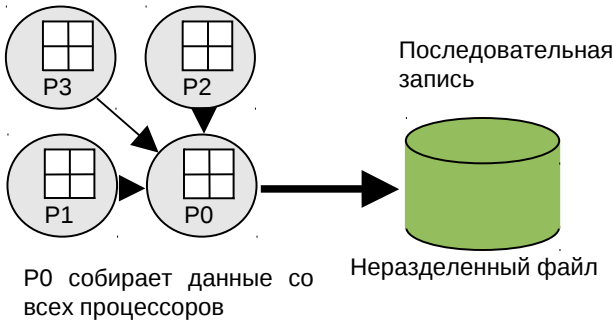
Отличия в структуре параллельных задач

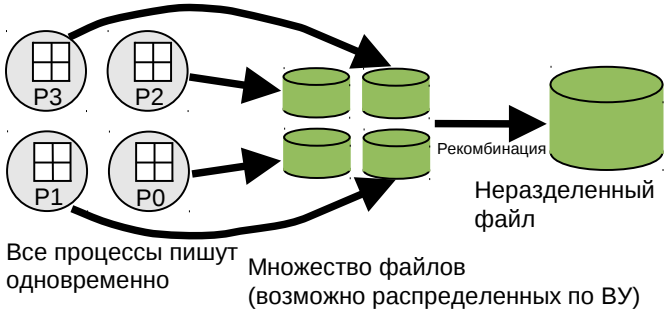
1. Задачи, которые почти не используют ввод-вывод;
2. **Счетные задачи, которые используют ввод-вывод и проводят операции существенных объемов.**
3. Задачи обработки больших массивов данных, для которых сам счет является малой составляющей.

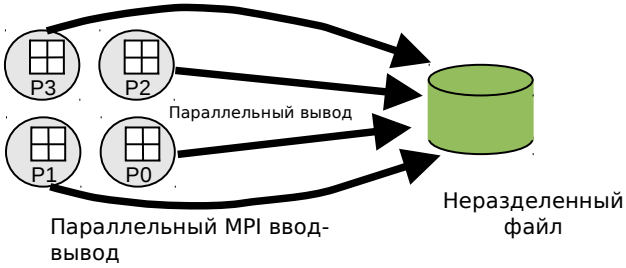
$$O = \begin{cases} SW - \text{последовательная запись} \\ RW - \text{случайная запись} \\ SR - \text{последовательное чтение} \\ RR - \text{случайное чтение} \end{cases}$$

Базовые операции ввода-вывода

Запись в единый файл







Проблема «контрольной точки»

КТ – «контрольная точка»

1. Системная КТ
2. Пользовательская КТ

Общие свойства КТ:

- запись на ВСХД, операция последовательной записи;
- в массовом порядке;
- записываются только те данные, которые могут быть востребованы для последующего счета;
- размер блока операции выбирается достаточно большой.

Файловые системы СК

Цели:
Программная организация доступа к внешним данным

Задачи:
Разделение доступа к данным между клиентами ФС

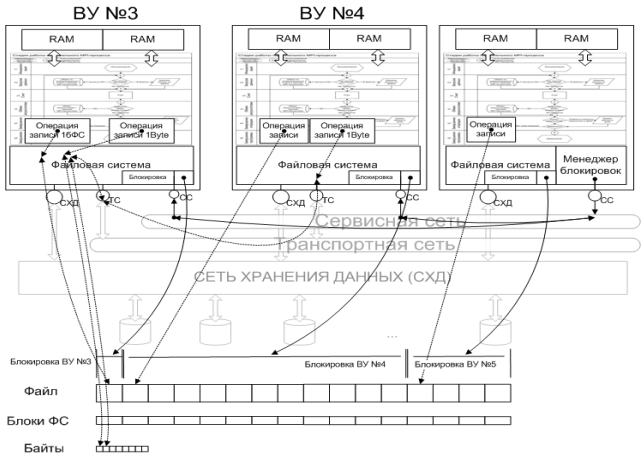
Требования к ФС:
•POSIX (полное удовлетворение только на одном узле)
•Масштабируемость (наращивание объёмов и производительности)
•Обеспечение независимости процессов разных ВУ по данным (механизм блокировок). Параллельный доступ к данным и метаданным.
•Репликация, целостность данных, восстанавливаемость после сбоев

Структура:
Данные (последовательность байт)
Метаданные (информация о файле, права доступа, карта (layout) расположения данных, директории)

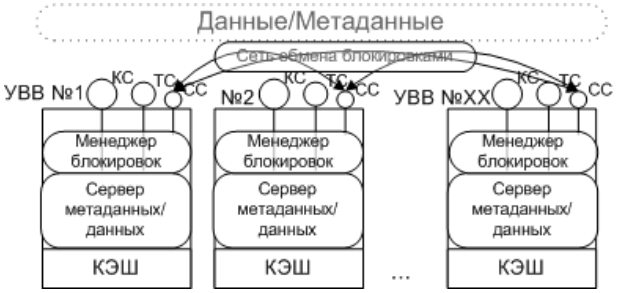
ПФС (1)

Разделение данных по дискам, упреждающее чтение, кэширование записи.

Поддержка больших директорий
Регистрация изменений и восстановление метаданных
Механизм блокировок данных и метаданных

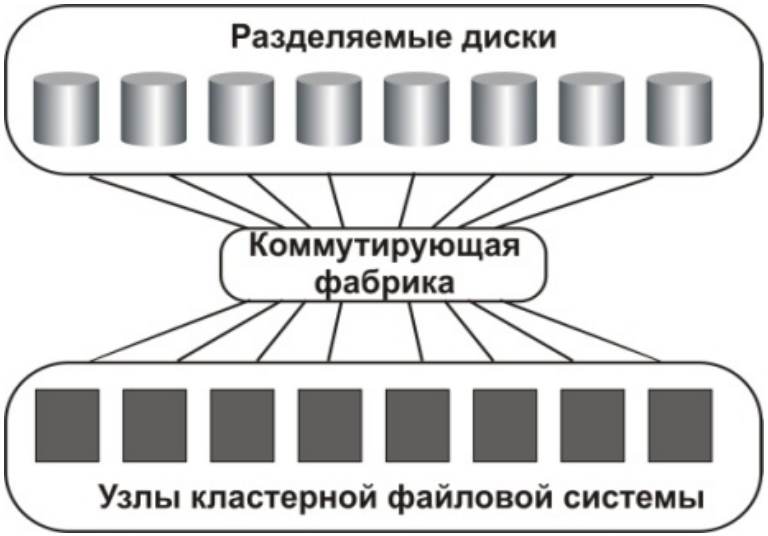


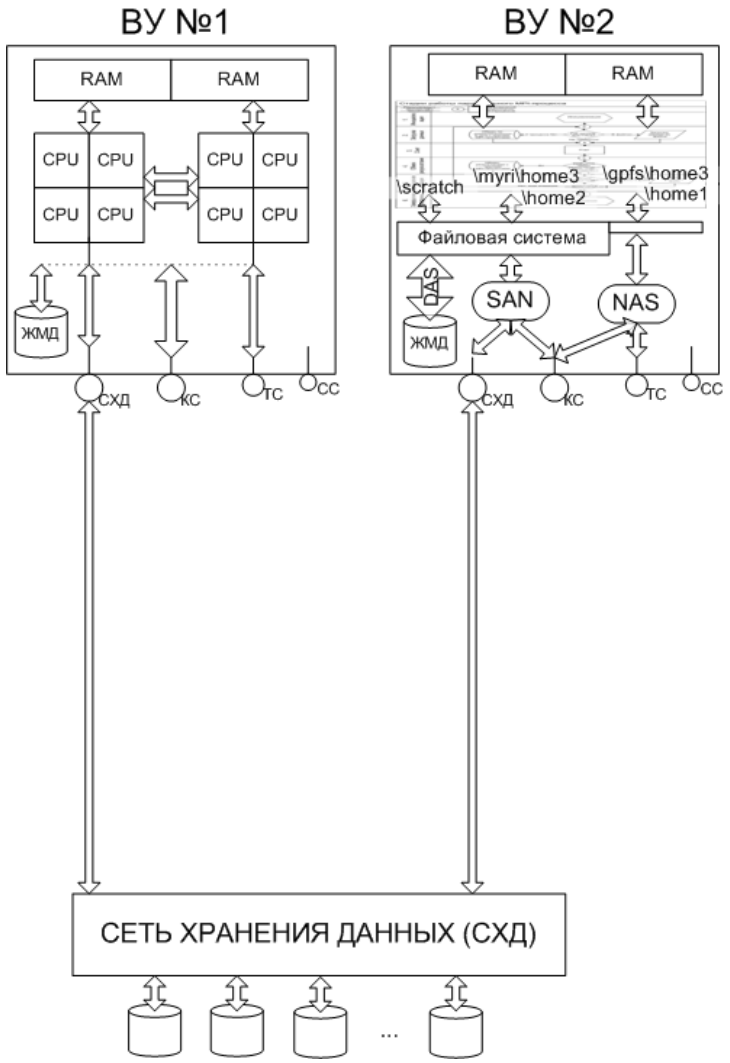
Маленькие файлы, хвосты больших файлов
Управление блокировками



Таблицы размещения, разделение прав на добавление
Глобальные метаданные ФС
Отказоустойчивость, определение момента сбоя (fail IO узла, сбой канала или карты, сбой дисков

Особенности файловых систем кластеров





Механизм распределённой блокировки

Принцип работы на примере записи в существующий файл.
Каждый ВУ пишет в свою, неперекрывающуюся область файла.
Каждому узлу IO доступны все носители данных (не метаданных).

1. Клиент открывает файл
2. Назначается узел IO, который получает жетон (token) на блокировку (далее «блокировка») на метаданные файла (inode)
3. Клиент работает с данными атомарными операциями (позиция в файле, длина блока операции). Записывает с 1-го байта по n байт.
4. Соответствующий узел IO получает блокировку на весь файл
5. Другой клиент открывает этот же файл общаясь с назначенным узлом метаданных и проводит атомарную операцию с m (m>n) по M байт с другим узлом IO.
6. У первого узла IO отзывается блокировка, данные узлом сбрасываются на диск. И первому узлу IO даётся блокировка на файл с 1-го по m-1 байт, а второму с m по конец файла.

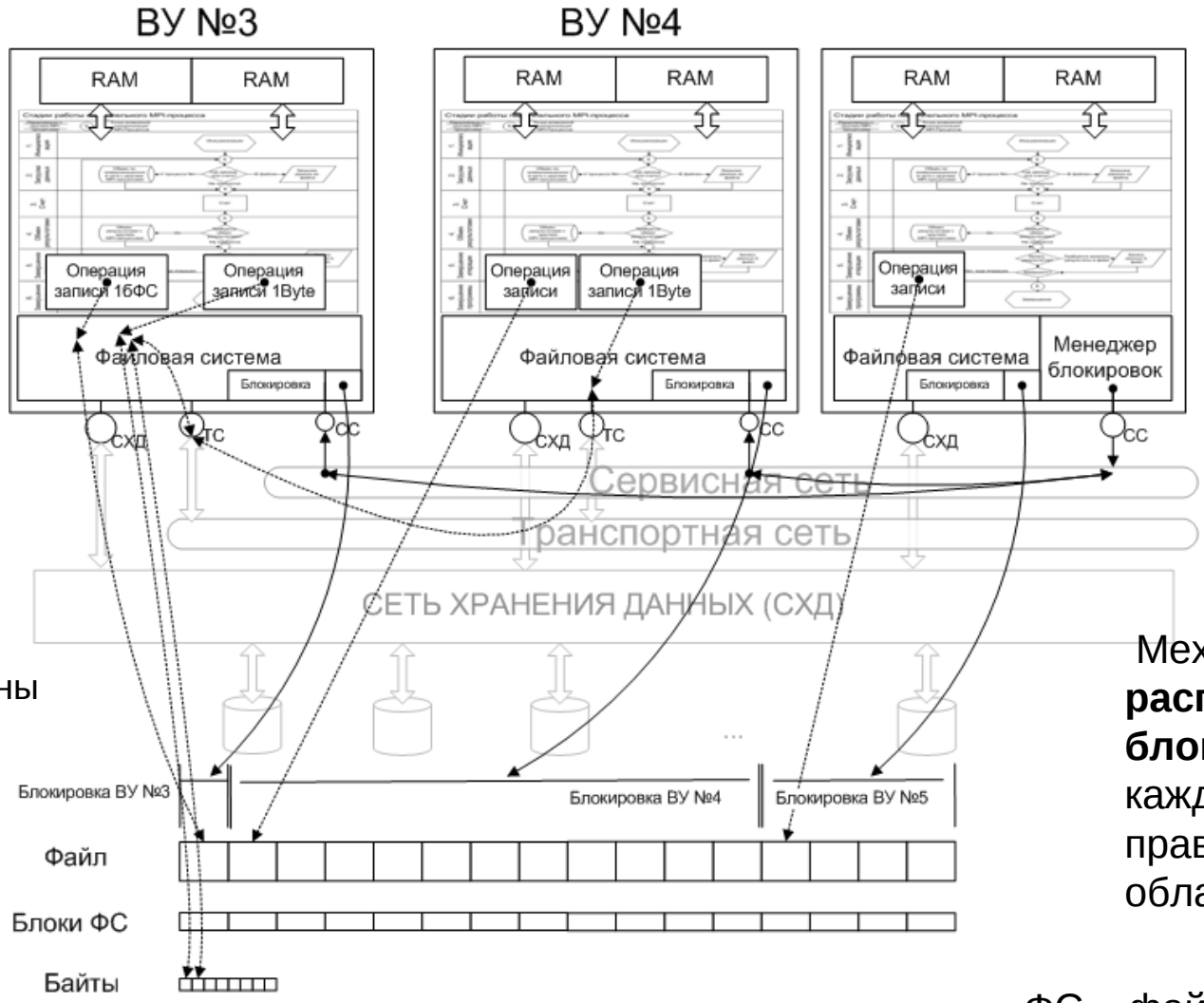
И т.д.

Механизм доставки данных

Принцип работы на примере считывания данных из файла
Каждый ВУ читает одну и туже область файла.
Каждому узлу IO доступны все носители данных (не метаданных).

1. Клиент открывает файл
2. Назначается узел IO, который получает жетон (token) на блокировку (далее «блокировка») на метаданные файла (inode)
3. Клиент читает область файла.
4. Соответствующий узел IO получает блокировку на весь файл
5. Другой клиент открывает этот же файл общаясь с назначенным узлом метаданных и проводит операцию тем же блоком ФС с другим узлом IO.
6. Второй узел IO запрашивает данные у первого узла IO (предварительно узнав его адрес у метаузла) и отдаёт клиенту.

Сочетание механизмов блокировки

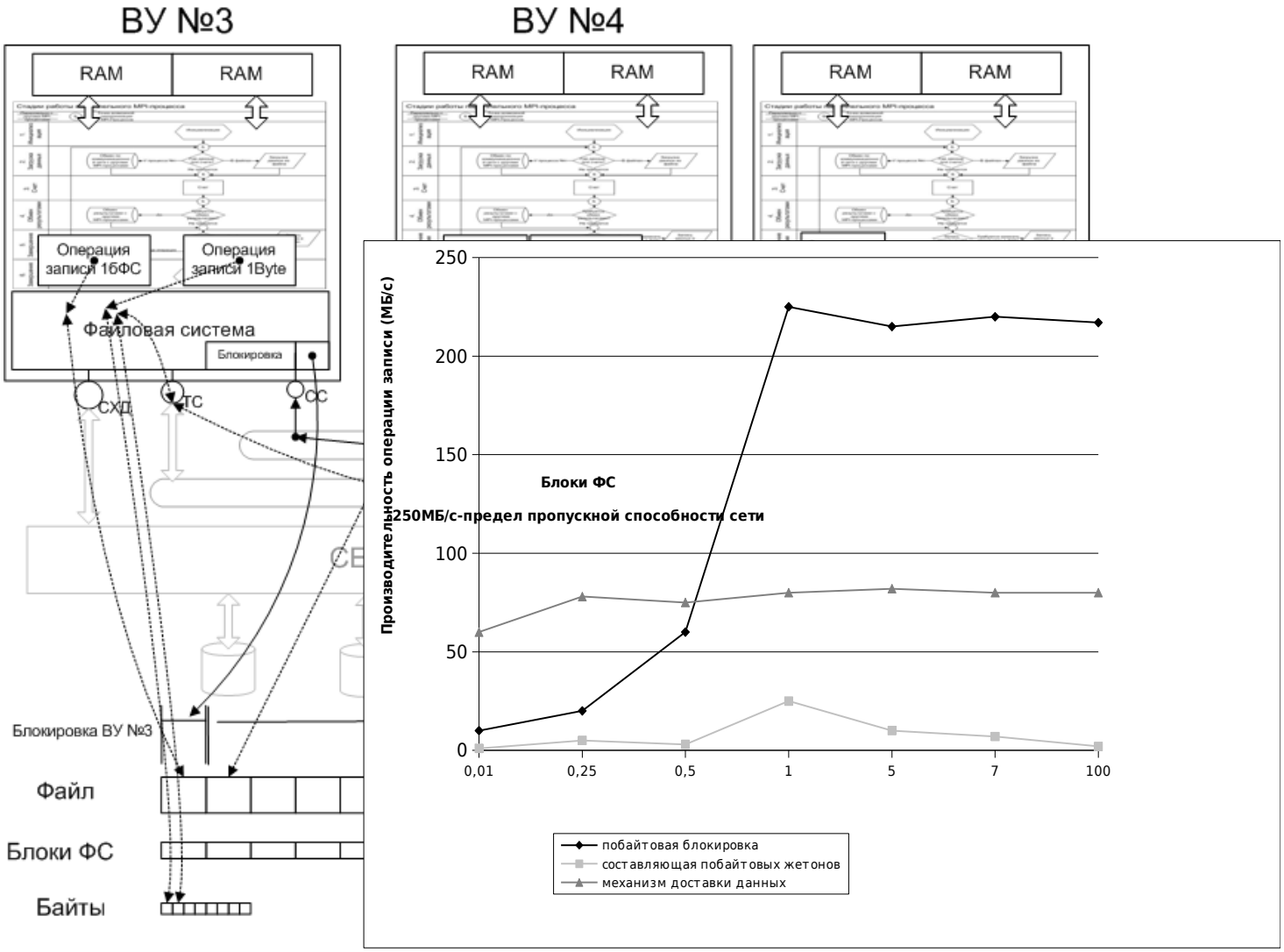


Механизм **Центральной синхронизации**: права на область данных закреплены за одним узлом

Механизм **распределенной блокировки**: каждый узел имеет права на свою область данных

ФС – файловая система

Сравнение механизмов блокировок



ПФС (2)

Масштабируемость системного инструментария

системный менеджер
распараллеливание работы
фоновый режим, приоритет операций
(исключение - полная проверка целостности файловой системы)

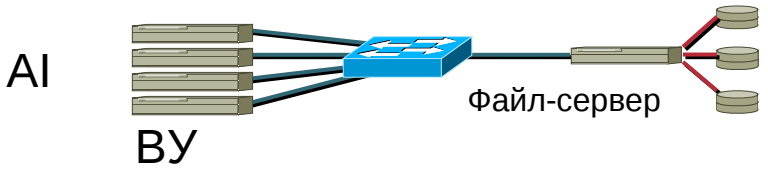
Классификация систем хранения

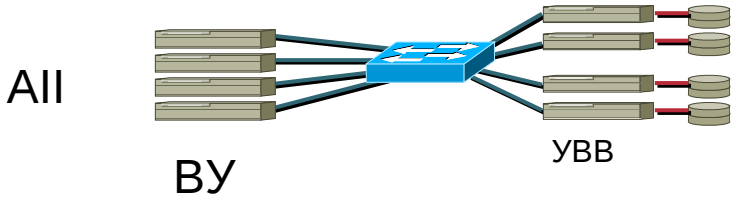
- AI - Централизованная (Сетевые ФС, NAS-системы)
- AII - Распределенная
- AIII - Разделенная

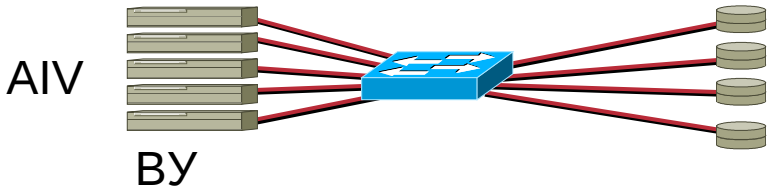
(Параллельные файловые системы - ПФС)

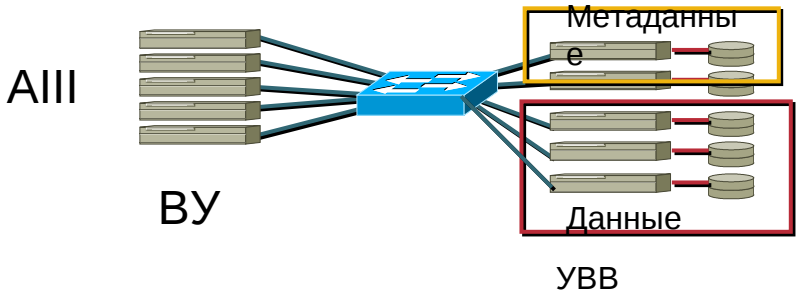
- AIV - Прямого блочного доступа
- A0 – на базе локальных дисков

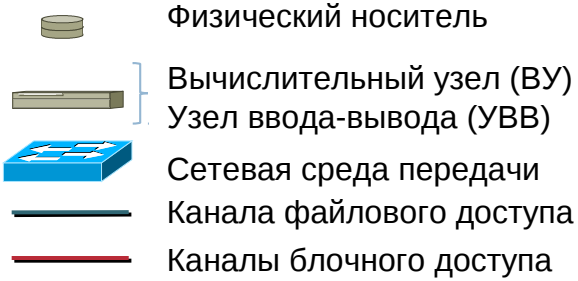
(ПФС прямого блочного доступа, SAN-системы)

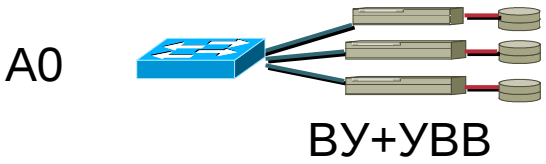




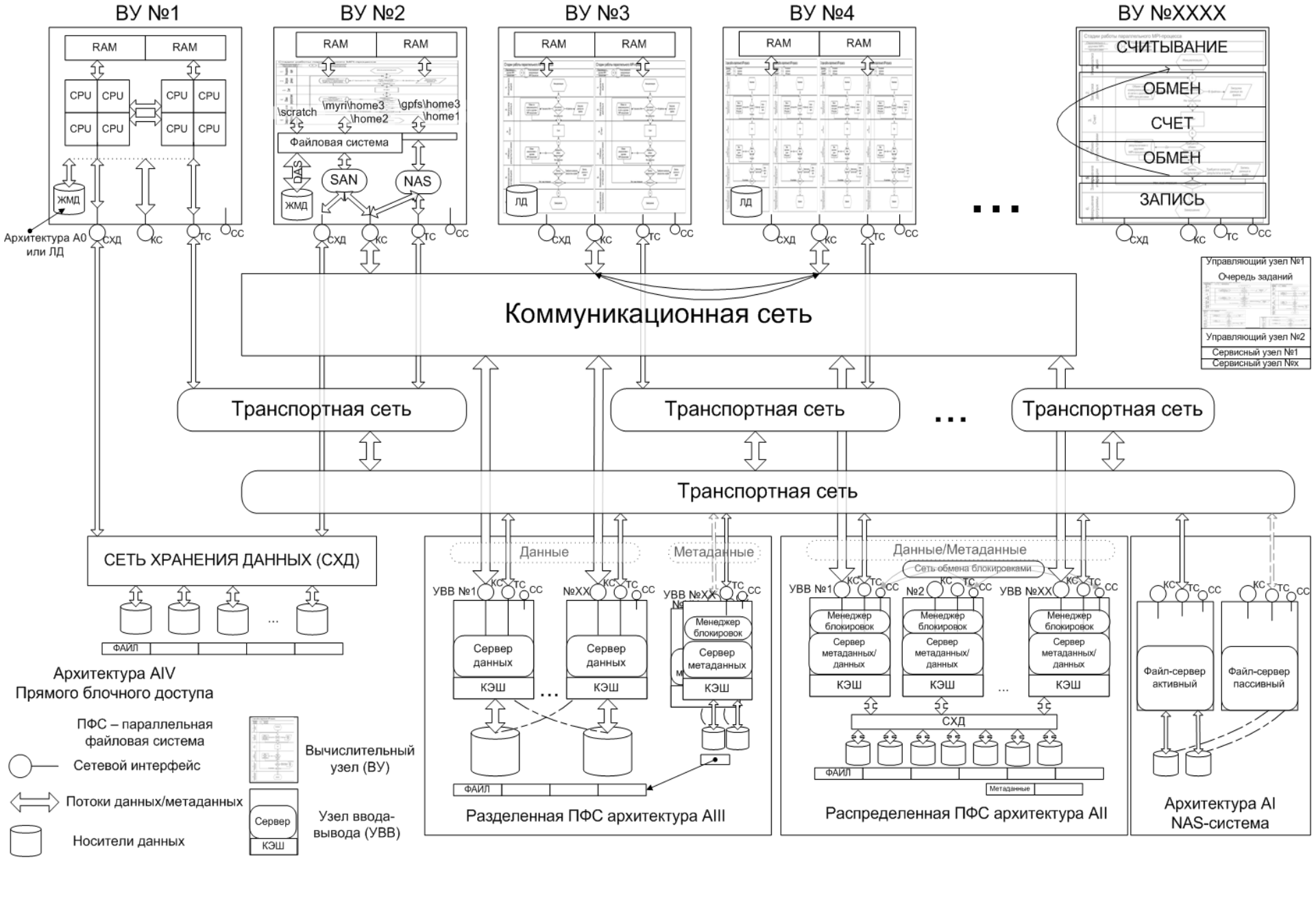








Кластерный вычислительный комплекс



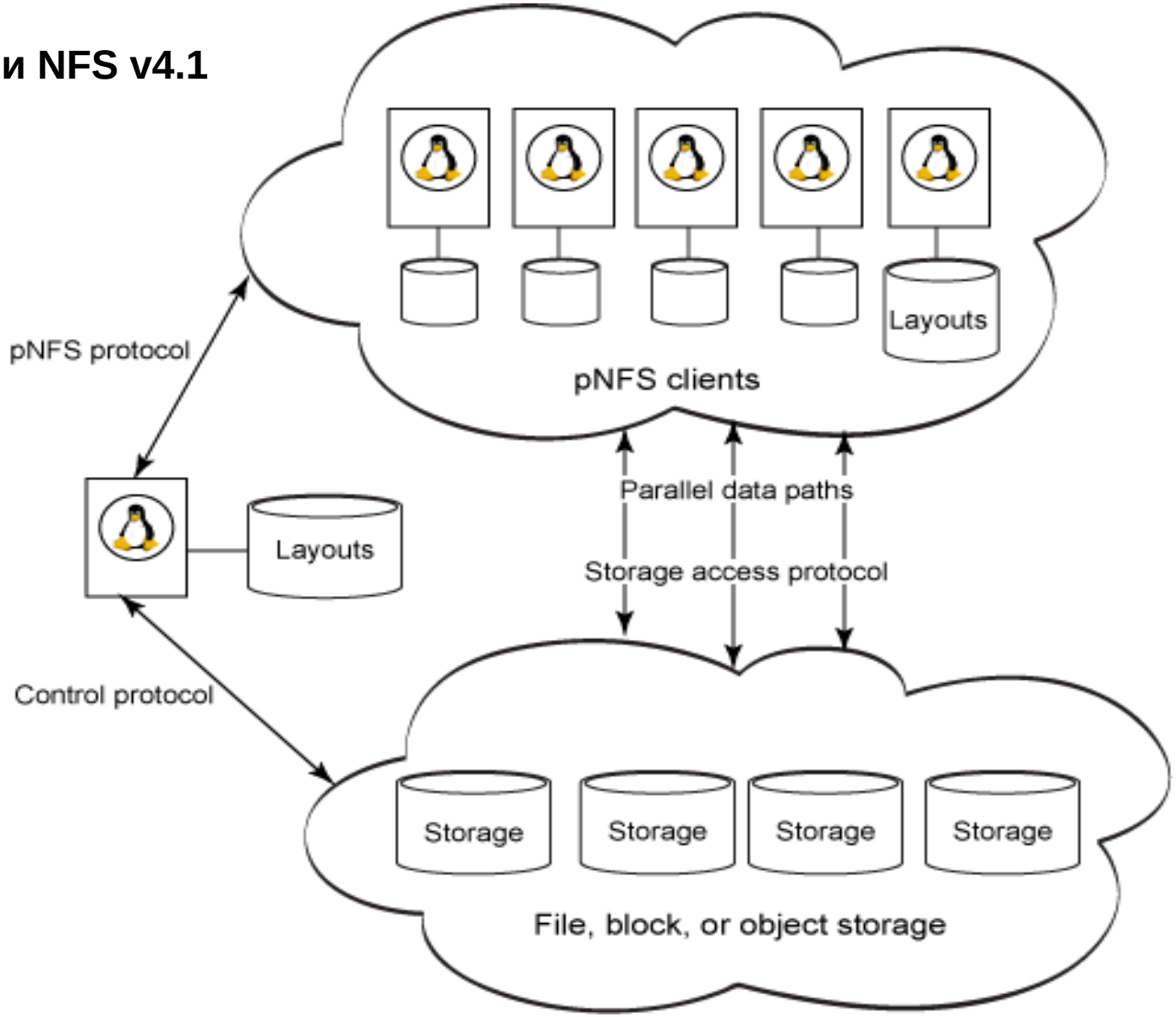
Сравнение ПФС

IBM GPFS (AIX, вариант с linux) - параллельная ФС разделения дисков (shared-disk architecture) для больших ВС. Архитектура **AIV.** Она позволяет совместно использовать диски всеми ВУ. Основная идея - механизм распределенной блокировки (distributed locking).
(Журналирование метаданных, доставка данных при побайтовой блокировки, разделение данных по сетевым и общим дискам).
IBM GPFS (linux) - ПФС с распределённой архитектурой **AII**, предназначенная для кластерных вычислительных систем.
GFS (Global file system) **AII** – open-source для linux. Механизм блокировки подобен GPFS и завязан с физической реализацией хранилища информации[1].
SGI XFS файловая система спроектирована для хорошо масштабируемых высокопроизводительных приложений. **SGI CXFS AIII**– кластерный вариант. Центральный сервер метаданных.
LUSTRE – open-source проект для Linux. **AIII.** ПФС работает с объектами. Один сервер метаданных (второй для пассивной избыточности), который управляет распределением и доступом к данным, находящихся на серверах данных.

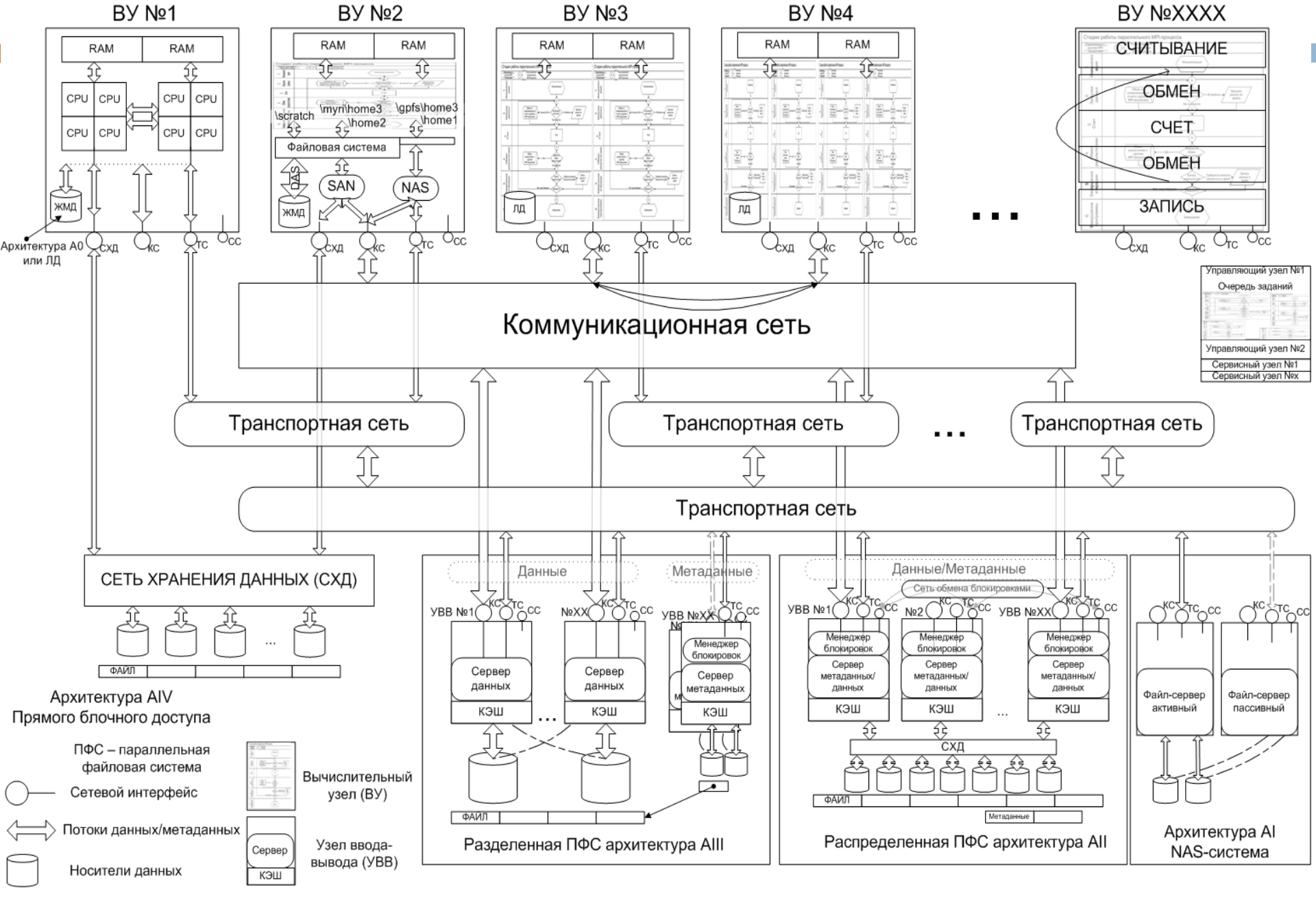
1. В ранних версиях требовалось, чтобы механизм блокировки поддерживался самим протоколом SCSI и физическим дисковым устройством. Новые версии используют механизм распределенной блокировки, но в отличии от GPFS могут использовать только блоки размером 4 или 8 KB.

Parallel NFS

pNFS или NFS v4.1



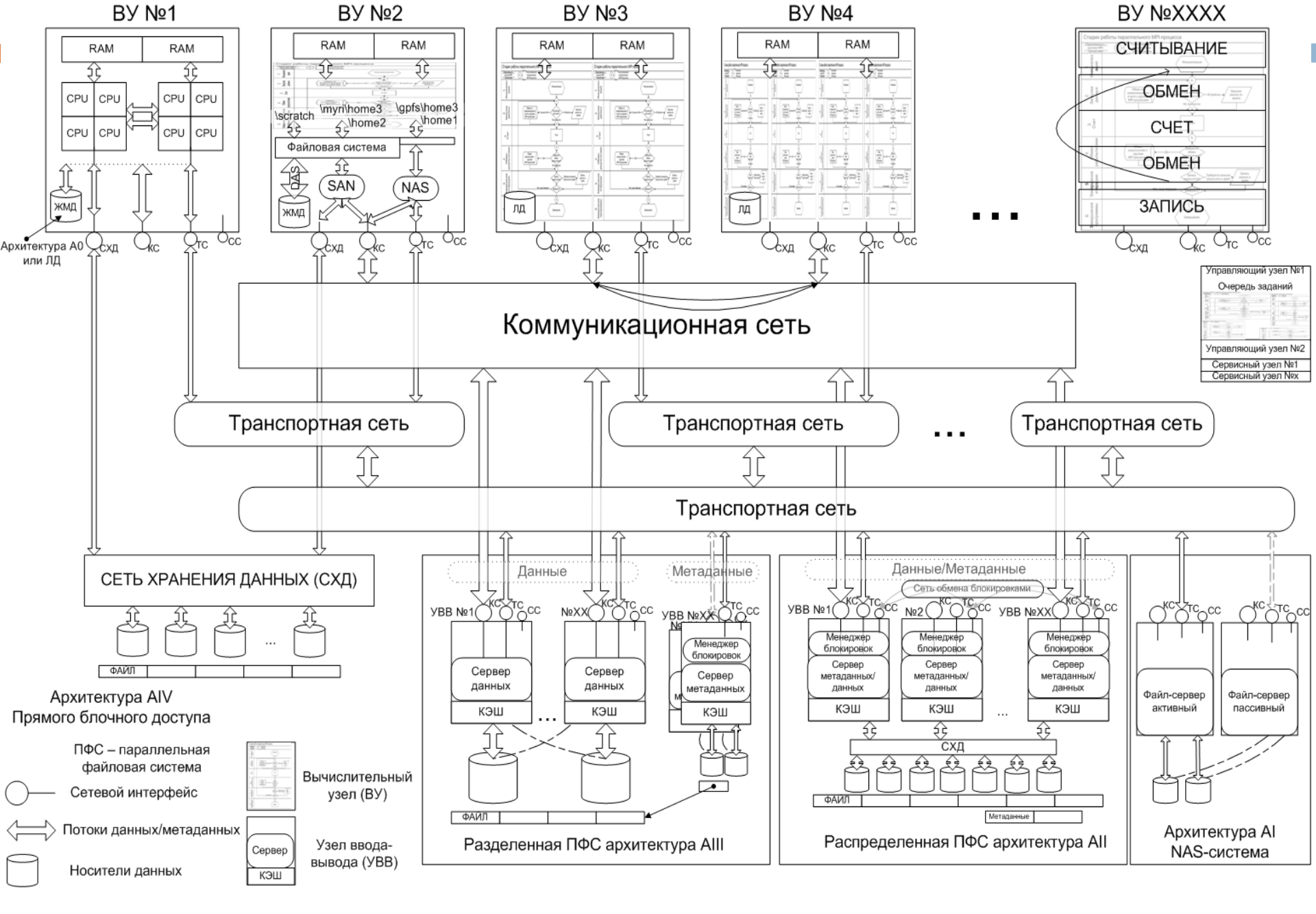
Кластерный вычислительный комплекс



Заключение

- Главный принцип ПФС – **распределение данных и метаданных** по носителям для обеспечения **раздельного/независимого доступа**. (проблема обеспечения независимости операций по данным – основной вопрос современных IT)
- Распараллеливание функциональности ФС (механизмы блокировок)
- Архитектуры ВСХД и ФС должны соответствовать друг другу или даже повторять.
- Основополагающий момент для свойств ПФС – какие механизмы блокировок используются. Этот же вопрос определяет архитектуру соответствующей СХ.
- IO в приложениях HPC имеет схожие свойства. Основная проблема всех ФС для HPC – сброс КТ (запись в разные файлы одной директории большими блоками данных). Вторая важная проблема – одновременный массовый ввод данных (Из одного файла или каждый из своего).
- IO в HPC и Cloud – две принципиально противоположные темы. Хотя некоторые приёмы работы с данными совпадают.

Кластерный вычислительный комплекс



Российская академия наук
Межведомственный суперкомпьютерный ценртр РАН
О.С. Аладышев
30 июня 2012 года