**Основные определения и характеристики персональных компьютеров и различных вычислительных систем**

1. **Быстродействие(производительность)**-возможность компьютера обрабатывать большие объемы информации. Основные определяющие структуры- процессор и оперативная память, а также скорость доступа между центральным процессором и оперативной памятью.
2. **Быстродействие(производительность)** **процессора-**количество элементарных операций за 1 секунду
3. **Тактовая частота процессора(частота синхронизации)**-промежуток времени за который выполняются простейшие микрооперации. **Тактовая частота** – число вырабатываемых за секунду импульсов, которые синхронизируют работу всех узлов компьютера. Кварцевый резонатор является основным генератором тактовой частоты(обозначается ZQ)
4. **Разрядность процессора**-количество разрядов (максимальная длина), который процессор может обработать одновременно (целиком)
5. **Время доступа**-определяется быстродействием оперативной памяти(ОЗУ). Период времени, необходимый для считывания минимальной порции информации из ячеек памяти и записи в неё. А также это процесс записи минимальной информации.
6. **Объем памяти(емкость)-** максимальны объем информации, который может храниться в памяти
7. **Плотность записи** – объем информации, записанный на единицы дорожки. Размерность бит/мм
8. **Скорость обмена информации**- скорость записи/считывания на носитель. Она определяется скоростью вращения либо перемещения этого носителя в устройстве

**Классификация вычислительных систем (классификация потоков данных, команд и их обработки)**

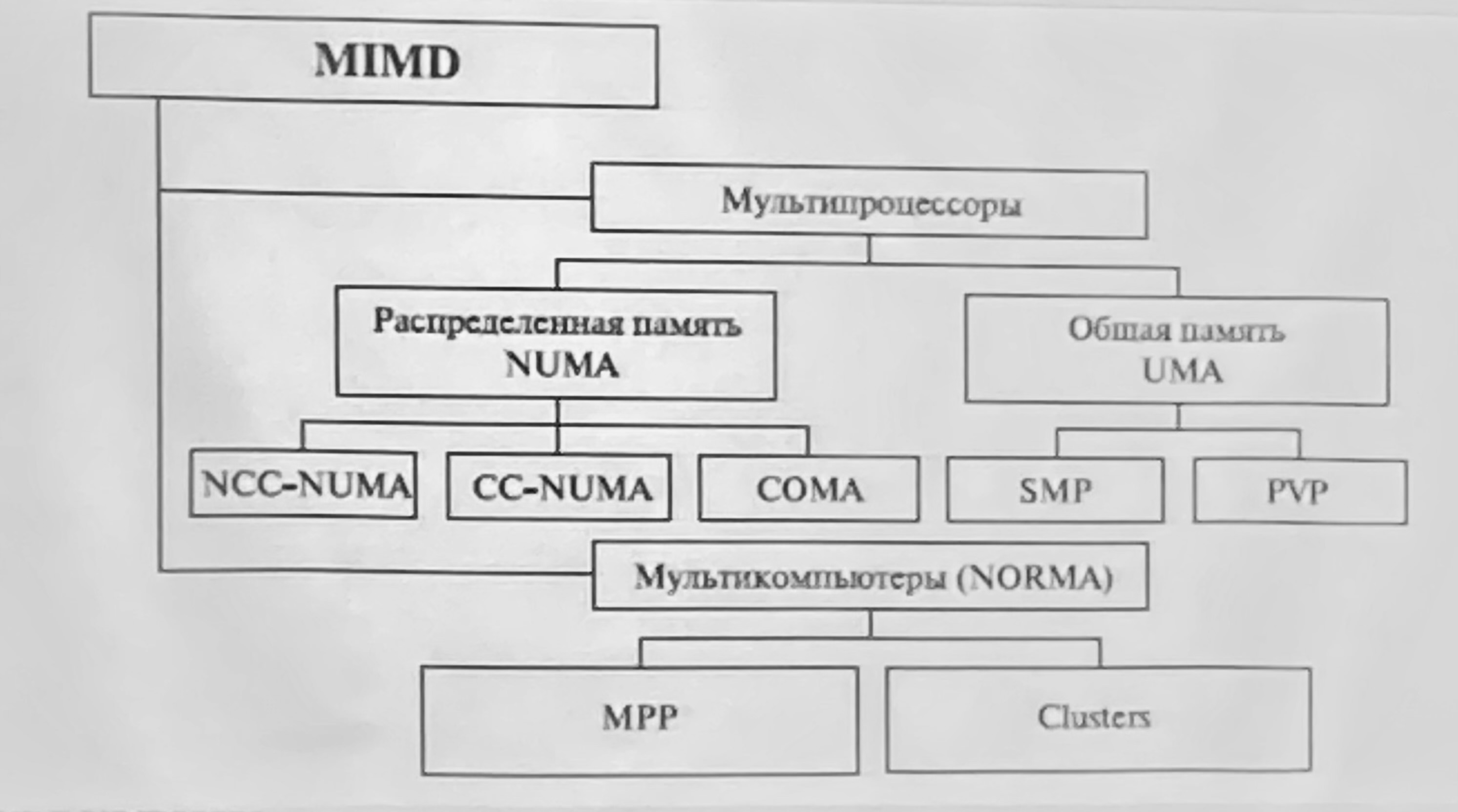
По способам взаимодействия потоков выполняемых команд и обрабатываемых данных ЭВМ делится на:

SISD – система в которых существует одиночный поток команд и одиночный поток данных

SIMD(single instructions multiple date) – одна команда и много данных

MISD – множественный поток инструкций и одиночный поток данных (не утверждённый класс, многие считают в ВМ соответствующих данному классу не существует. Система с конвейерной обработкой данных может относиться)

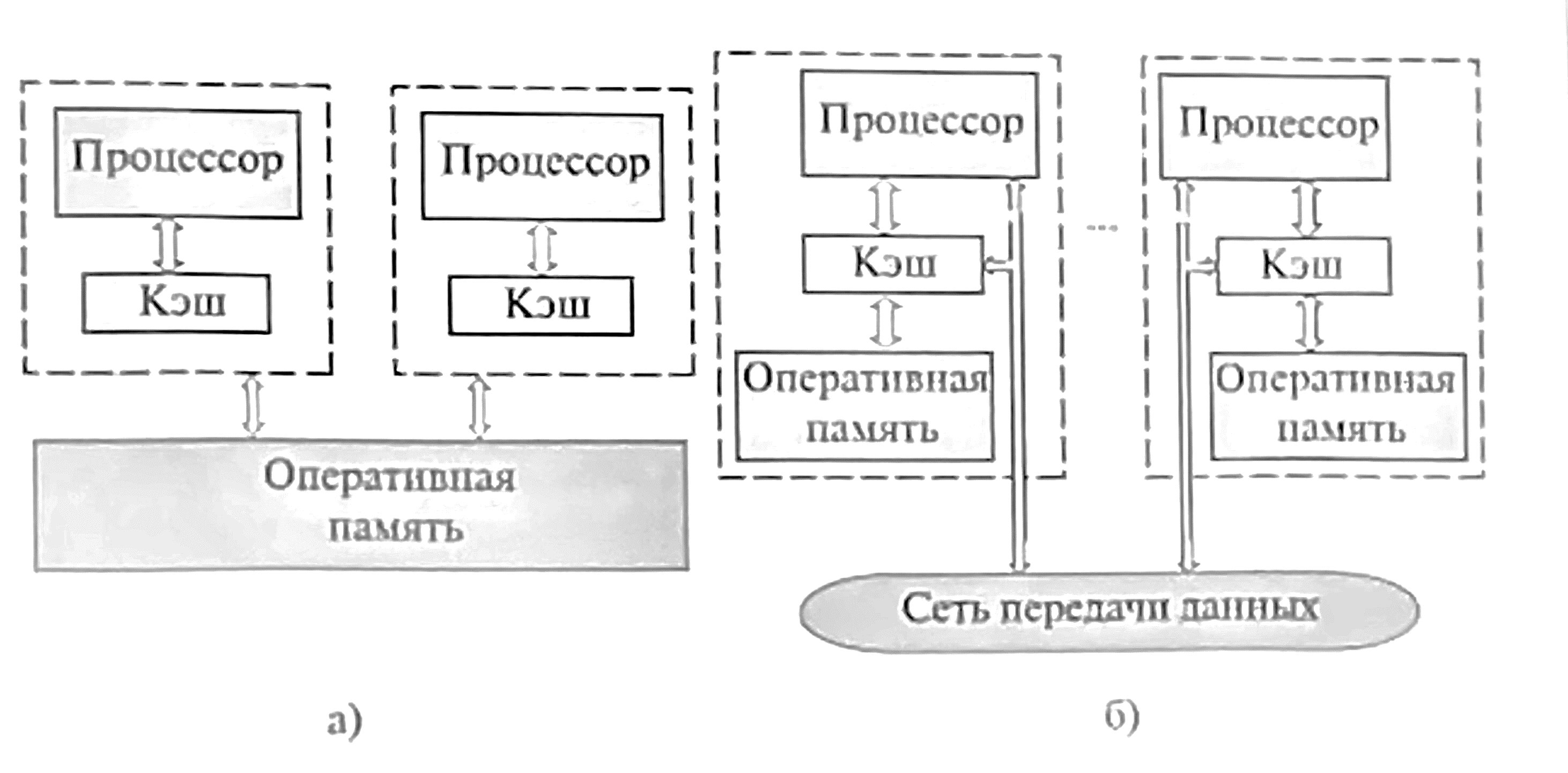
MIMD – множественный поток с множеством данных



Мультипроцессоры – системы с общей разделяемой памятью

Мультикомпьютеры- системы с общей распределенной памятью

**Мультипроцессоры**

****

А)использование единой т.е. централизованной общей памяти. Подход называется UMA(Uniform Memory Access). Основной проблемой данного подхода является организация параллельных вычислений(одновременный доступ центральных процессоров к общей памяти, а соответственно и к общем данным. При этом возникает проблема когерентности-Cash Coherent Problem-проблема когерентности кэша)

Если в такой ситуации(т.е. при наличии копий общих данных) один из процессоров выполнит изменение значения разделяемой переменной, то значение копий в кэше становится разным (не соответствует действительности). Их использование приведет к некорректности вычислений. Проблема решается на аппаратном уровне.

Если общая переменная изменилась, то все ее копии в остальных кэшах считаются устаревшими(недействительными). Следовательно, когда необходимо получить доступ к этой переменной повторной необходимо обращаться к оперативной памяти.

Общий доступ к данным осуществляется при физически распределённой памяти. Длительность доступа для всех элементов памяти не одинакова, т.е. называется это неоднородный доступ к памяти. (NUMA)  
COMA – используется только локальная кэш-память, используемая процессором.

* **UMA (**Uniform Memory Access**)** – память общего доступа (единая, центральная память)

NUMA (Non-Uniform Memory Access) – общий доступ реализован при физически распределенной памяти.

В таком случае, длительность доступа будет не одинаковой для всех элементов памяти

Использование распределенной общей памяти называется **DSM**. Она упрощает создание мультипроцессоров.

Возникают проблемы эффективного использования распределенной памяти. Скорости доступа к локально и глобально распределенной памяти отличаются на несколько порядков. Из этого следует повышение сложности параллельного программирования

**Мультикомпьютеры**

****

**Мультикомпьютеры** – мультипроцессоры с распределенной памятью, не обеспечивают общего доступа ко всей памяти

NoRMA (no-remote memory access) – многопроцессорная системы с распределенной памятью

Мультикомпьютеры имеют важное **отличие** от систем с распределенной памятью (тип Б). В данном случае каждый процессор может использовать только свою локальную память. В то время как для доступа к данным, располагаемым на других процессорах, необходимо выполнить **явные операции передачи** сообщений

message passing operation

MPP

Применяется для массивно-параллельных систем

Также данная технология применяется в кластерах

Под **кластером** понимается множество отдельных компьютеров, которые объединены в сеть, для которых при помощи специальных программно-аппаратных средств обеспечивается возможность унифицированного управления, надежности функционирования, а также эффективности использования

**Кластеры** могут быть образованы на базе уже имеющихся у пользователей компьютеров или сконструированы из отдельных типовых компьютерных элементов. А также на базе некоторых миниатюрных компьютерных платформ

Самое большое достоинство кластеров-не требуются полностью одинаковые машины.

Для параллельного выполнения в алгоритмах решения вычислительных задач достаточно выделять только крупные, независимые части расчетов, это снижает сложность построения параллельных методов вычисления, как в случае с предыдущими видами подключения. Тем самым уменьшаются потоки передаваемых данных между компьютерами в кластере

**Отказоустойчивость компьютерных систем**

Выделяют два основных метода повышения КС (компьютерных систем)

* Аппаратурный
* Программный

**Аппаратурный метод**

Компьютерная система будет содержать аппаратурные излишки (аппаратурная избыточность)

Основные моменты

1. Все операции производятся параллельно на одинаковых компонентах. При этом результаты логически сравниваются между собой
2. В случае выходе из строя одного из узлов, его партнёр будет продолжать работу без прерываний. При этом в течении оговоренного срока должна произойти замена вышедшей из строя части

Троированная мажоритарная система используется в системах, в которых необходима высокая отказоустойчивость. Главный недостаток-очень высокая стоимость и аппаратурная избыточность

**Программный метод**

Предусматривает одновременную работу нескольких машин

* Дублирование данных и процессов
* Процедура автоматического восстановления операционных систем, данных и приложений

Зачастую аппаратный метод становится дешевле программного

Надежность системы рассматривается с точки зрения двух основных критериев

1. Доступность системы
2. Целостность данных

**Управляемость системы**

Под управляемость системы понимается следующий набор функций

1. Замена и ремонт повреждённых модулей должны производится незаметно для работы всей системы
2. Непрерывность функционирования системы в случае любого изменения конфигурации. Либо сбой какого модуля (downgrade) или расширение системы (upgrade)
3. Эффективное управление массивами данных в случае очень больших объёмов
4. Быстрая обработка очередей
5. Эффективное распределение нагрузки между ресурсами
6. Сохранение и архивация данных в онлайн режиме
7. Быстрое обнаружение и ликвидация ошибок
8. Мониторинг самой системы

**Способы повышения надёжности**

RAID-массивы (Redundant Array of Inexpensive Disks)

Отказы других элементов можно корректировать различными способами

* Дублирование (избыточность)
* Сбои питающей сети контролируется установкой батарей, источников бесперебойного питания ИБП (UPS)
* Кластеризация

**Кластер** – несколько компьютеров соединенных коммуникационных каналом и разделяющих между собой общие ресурсы

Кластер в основном имеет общую файловую систему. При этом выглядит для пользователя и управляется как единая система

Надежность кластеров достигается за счет специального ПО, которое регулирует скоординированное использование общих кластерных ресурсов. А также производит взаимный контроль работоспособности и осуществляет обмен кластерной информации

Отдельные компьютеры называются узлами кластеров

Другие способы увеличения надежности также используются в кластерах, RAID-массивы и ИБП

**Отличительная особенность кластера**

Каждый узел кластера выполняет полезную работу в режиме нормального функционирования

В случае отказа одного из узлов кластера любой другой узел может взять на себя нагрузки отказавшего

Кластер включает в себя набор специального программного обеспечения

Программное обеспечение позволяет:

* обеспечивать непротиворечивость доступа приложений с разных машин к общим ресурсам
* мониторинг и конфигурирование
* утилиты гибкого конфигурирования
* программные модули управления томами

**Кластеризация**

**Кластер** – это несколько компьютеров, соединенных коммуникационным каналом и разделяющих общие ресурсы (например, дисковые накопители).

Кластер имеет общую файловую систему, а для пользователя он выглядит и управляется как единая система. Современные кластеры способны

обеспечить надежность до 99,9% (High availability). Все это достигается с помощью специального программного обеспечения, регулирующего

скоординированное использование общекластерных ресурсов, осуществляющего взаимный контроль работоспособности и обмен

специфической "кластерной" информацией между узлами кластера и т. д.

При этом различные простые аппаратные способы повышения надежности (RAID, ИБП) также используются. Отдельные компьютеры называются (RA узлами кластера. Отличительной особенностью кластера является то, что каждый его узел выполняет полезную работу в режиме нормального функционирования и может переключить на себя нагрузку отказавшего узла.

Кроме того, кластера включают в себя набор специального программного обеспечения для оптимального распределения ресурсов и удобного администрирования.

Некоторые решения имеют следующие составляющие:

* возможности выявления и корректировки всех видов серьезных системных сбоев;
* программу, обеспечивающую непротиворечивость доступа приложений с разных машин к общим ресурсам;
* утилиты конфигурирования и мониторинга состояния кластера;
* утилиты гибкого конфигурирования файловых систем;
* программные модули управления дисковыми томами и некоторые другие программы

Как правило, кластерные решения поставляются в комплекте с инструментарием разработчика, позволяющим конструировать необходимое ПО. Рассмотрим общие **свойства кластеров**:

**Открытость**

Как правило, создатели кластеров заявляют об их полной открытости. Однако ограничения все же есть. Прежде всего стоит отметить, что каждое кластерное решение создается под конкретную платформу и непереносимо на другие. Кроме того, при введении кластера в строй создаются специальные сценарии, которые приходится менять при расширении, изменении архитектуры кластера, а также смене системного или прикладного ПО.

**Масштабируемость**

Производительность кластеров действительно растет с добавлением новых ресурсов. Ее рост можно рассчитать с достаточной

точностью. Следует иметь в виду и то, что сбой в одном из узлов кластера повлечет за собой некоторое, также в простых решениях рассчитываемое, снижение производительности системы. Этот факт учитывается заранее при

планировании и выборе необходимой архитектуры кластера. Каждое из решений имеет свой предел расширяемости.

**Надежность**

В случае сбоя в системе типичный кластер действует приблизительно по **следующей схеме**:

* определение отказа;
* формирование нового кластера;
* запуск сценариев;
* тестирование файловой системы;
* запуск базы данных;
* восстановление базы данных;
* перезапуск приложений.

При использовании хорошо спланированных кластерных решений данные хорошо защищены в случае серьезного сбоя программного или аппаратного обеспечения. Однако при случайных ошибках в работе, например, процессора программный метод защиты данных не дает необходимых гарантий.

Что же касается **доступности системы**, то здесь ситуация следующая:

* В случае серьезного отказа какого-либо узла некоторые приложения могут быть недоступны в течение времени, необходимого для переключения на другой узел
* Для восстановления нормального режима работы при отказе центрального процессора, памяти, шины или контроллера требуется время. По некоторым оценкам это время составляет 70 секунд и более. Максимально необходимое время зависит от серьезности неполадки, а также от используемых приложений, СУБД и объема данных клиента. Не исключено, что пользователю при этом придется повторно войти в систему

**Управляемость**

В случае замены существенных элементов, операции, которые необходимо будет провести, могут создать определенную задержку в работе системы или даже потребовать прерывания работы пользователя на небольшой срок. Может понадобиться повторить процедуру для другого узла кластера, что опять отразится на пользователе. Подобный же эффект будет наблюдаться при изменении конфигурации кластера. Для улучшения работы с массивами данных многие создатели кластеров сотрудничают с фирмами, выпускающими СУБД, уже на этапе разработки своих идей. В результате создается программное обеспечение, оптимально работающее с данной СУБД. И хотя при разработке кластера учитываются все широко распространенные СУБД, наиболее эффективным получается взаимодействие с СУБД фирмы-партнера, непосредственно участвовавшей в создании конкретного кластера. Запись и архивация данных происходят в режиме on-line сразу в нескольких копиях. Обнаружение ошибок осуществляется программным способом.

**Стоимость**

Цены сильно различаются в зависимости от выбранной архитектуры, степени интеграции узлов в кластере, функциональной полноты и возможностей наращивания. Кроме того, покупателям кластеров следует учесть дополнительные затраты по статьям расходов:

* возможная закупка дополнительного оборудования;
* оплата консультантов;
* обучающие тренинги для персонала;
* тестирование системы;
* скрупулезное планирование всех возможных событий и изменений в работе кластера;
* подготовка необходимой документации.

**Отказоустойчивые компьютеры**

Основным соперником кластеров сегодня являются мощные одно- или многопроцессорные серверы с отказоустойчивой аппаратной "начинкой". В отличие от создателей кластеров разработчики Fault tolerant (FT) систем с самого начала ставили перед собой задачу предотвращения и по возможности исключения, а не сокращения времени простоев. В основу архитектуры FT-серверов положено использование задублированных элементов, число которых в разных серверах может быть разным.

Любая команда выполняется одновременно на всех элементах, и результаты операций логически сравниваются, что предотвращает распространение ошибки. Каждый из задублированных элементов способен продолжать

работу в случае отказа одного из элементов-партнеров таким образом, что

прикладная система не почувствует отказааа и ее функционирование не ухудшится. Например, на аппаратном уровне один процессор в таком сервере - это четыре ЦП. Каждая пара ЦП и основная память образуют один SPU (system processing unit), а пара SPU - один процессор. При нормальной работе на каждом SPU обрабатывается одна задача. Если результат работы на одном ЦП отличается от результата на втором, то система диагностирует

неисправность аппаратного модуля SPU и автоматически изолирует его. Другой модуль при этом продолжает работу до тех пор, пока его партнер не будет заменен в "горячем" режиме. Такая технология была названа **continuous availability** (постоянная доступность).

**Открытость**

Как правило, современные мощные FT-серверы оснащены специально модифицированной отказоустойчивой версией операционной системы UNIX.

**Масштабируемость**

Системы обладают хорошей расширяемостью, и вычислить прирост производительности возможно. Важно отметить, что бл благодаря аппаратурной избыточности таких серверов, как, например, Stratus Continuum, при отказе какого-либо ресурса снижения производительной мощности не происходит.

**Надежность** реализована преимущественно аппаратным способом. Как уже указывалось, максимальное время недоступности системы для пользователей составляет в среднем несколько минут в год. Целостность данных обеспечивается.

**Структура цен** на подобные системы следующая. Сам сервер может стоить довольно дорого. Но за в счет продуманной конструкции и несложности эксплуатации нет дополнительных затрат на консультантов и специализированный персонал. Отсутствуют и скрытые затраты на планирование возможных сбоев, написание и модификацию сценариев. Таким образом, при учете всех затрат FT-система обойдется примерно во столько же, что и кластерное решение, а часто и дешевле

**Причины программных и аппаратных сбоев и неполадок**

Основной причиной является поломка компонентов

Нарушение условий эксплуатации

Возникновение аппаратных и программных сбоев и конфликтов

Программные причины сбоев

* Вирусы
* Неправильная установка или удаление файлов
* Отсутствие важных файлов, в следствии предыдущей причины, а также сбой жёстких дисков
* Фатальные изменения в реестре
* Программная несовместимость
* Несовершенство ПО
* Несовершенство ОС
* Конфликты между устройствами
* Ограничения ОС
* Неверные настройки ОС
* Использование устаревшего ПО

При изменениях настроек **реестра** или удалении его частей необходимо сделать локальные копии

**Профилактика предотвращения сбоев**

* Обновление программного обеспечения и драйверов (не всегда)
* Своевременное обновление антивирусных баз
* Обновление системы безопасности ОС
* Периодическая проверка и чистка реестра
* Периодическая проверка на вредоносное ПО
* Проверка на совместимость аппаратных устройств с уже использующимися
* Производить правильное удаление и установку программ

Не занимать систему параллельными процессами, не вмешиваться в работу установщика, если того не требует процесс

* Производить защиту от сбоев в электропитании
* Производить правильное выключение компьютера (закрывать все приложения)
* Резервное копирование важных файлов и данных
* Осторожное обращение с диагностическими утилитами
* При явном снижении скорости работы жестких дисков, произвести дефрагментацию. Проверка на битые блоки

Основные пункты:

1. Диагностика и локализация
2. Создание нового кластера
3. Запуск
4. Тестирование файловой системы
5. Запуск БД
6. Восстановление из БД
7. Запуск программ

Параллельные вычислениях

Мажоритарная система (решение

Питание

ИБП (аккумуляторы)

Генератор дизельный

**Операционные системы серверного типа**

Microsoft Windows Server

**Различия между редакциями Microsoft Windows Server**

Рассматриваются 2012r2 и новее

* **Foundation** – серверная ОС общего названия

1 процессор, 32гб ОЗУ

15 пользователей (малый офис)

* **Essential** – SBS (Small Business Server)

2 процессора, 64гб ОЗУ

25 пользователей или 50 устройств (средние офисы)

Предварительно настроены роли сервера:

* Active Directory
* DNS
* Файловые службы
* IIS
* RDP

* **Standard**

Каждая лицензия распространяет на 2 физических процессора, 4тб ОЗУ

Неограниченное количество пользователей

Необходима клиентская лицензия (CAL)

Hyper-V – возможность запуска 2-х виртуальных экземпляров ОС

* **DataCenter**

Практически идентичен версии Standard

Возможность запуска неограниченных экземпляров Windows Server в качестве гостевой ОС на одном двухпроцессорном сервере

**Дополнительные различия для версий 2016 – 2019**

1. **Hyper-V**

Бесплатная редакция Windows Server, но предназначена **только для запуска гипервизора**. Является урезанной версией Server Core. Задачей данного гипервизора является основа для виртуальной среды. Отсутствует графический интерфейс

Управление конфигурацией – sconfig.cmd

Управление можно производить также через RSAT

**RSAT** (Remote Server Admitestranios Tools) – средство удаленного администрирования для пользовательской системы Windows 10. Набор инструментов:

* Диспетчер серверов
* Оснастки консоли управления MMC
* Всевозможные консольные программы управления
* Набор командлетов
* Средство управления ролями сервера

1. Небольшое отличие для **Essential**

Служит для базовых функций сервера.

Графический интерфейс как у версии Standard, за исключением мастера настройки.

Права на виртуализацию: можно запускать **только один экземпляр Hyper-V** в качестве хоста, при этом в экзамене будет запущен только один экземпляр Essential

1. **Standard**

**До 2-х виртуальных машин** или контенногов Hyper-V или один физический экземпляр. Лицензия предоставляется на основе ядра для каждого пользователя или устройства.

Доля каждой учётной записи потребуется лицензия CAL

1. **DataCenter**

Лицензия предоставляется на основе того, сколько ядер на хостах.

**Неограниченное количество** экземпляров и количества Hyper-V.

Клиентские лицензии необходимы для каждого устройства или пользователя, который косвенно или напрямую подключается к серверу

**Параметры установки**

**Desktop Experience**

Большинство функций и ролей при данной установке доступны “из коробки” + имеется графический интерфейс

Доступен диспетчер серверов

Недостатками данного варианта установки будет является наличие большего количества уязвимостей

Графический интерфейс потребляет больше виртуальных и аппаратных ресурсов

Большее количество обновлений

**Server Core**

Полное отсутствие графического интерфейса

Достоинства:

* Ядро занимающее гораздо меньше места
* Меньшее потребление ресурсов в процессе работы
* Меньше подвержена атакам
* Меньше получение обновлений

Специфика работы:

* Отсутствуют некоторые инструменты настройки сервера
* Отсутствуют некоторые специальные возможности
* Всё администрирование производится на командной строке

**Редакция Nano**

Является контеннерезированным образом операционной системы. Предназначается для запуска внутри хоста контейнеров, например, как Server Core.

Версию Nano можно запустить как при помощи DataCenter, так и в Standard

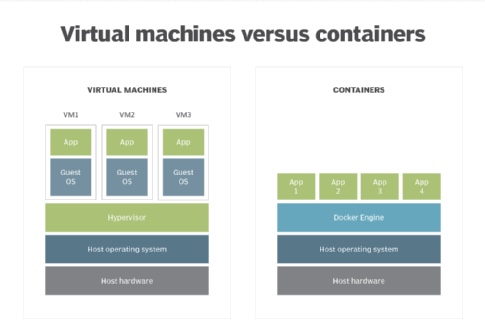
**Контейнеризация** – легковесная виртуализация и изоляция ресурсов на уровне операционной системы. Позволяет запускать приложения и необходимый им минимум (системные библиотеки) в полностью стандартизированном контейнере, который соединяется с хостом, либо другой внешней средой по отношению к нему (контейнеру)

**Контейнер** – файл или набор файлов на диске, которые запускается на контейнерном движке

**Контейнерный движок** – программное обеспечение, которое принимает запросы пользователя, после чего скачивает контейнерные образы и запускает их

Среда выполнения контейнера является низкоуровневым компонентом, который используется в составе контейнерного движка. После запуска контейнера, он становится системным процессом

**Отличия виртуальных машин от контейнеров**



**Виртуальная машина**:

* требует гипервизор, при этом для каждой машины требуется собственная операционная система, которая называется гостевая ОС
* позволяет создавать однородные вычислительные среды на одном компьютере
* занимает большее количество дискового пространства
* время запуска больше чем у контейнеров

**Контейнеры**:

* используется ядро хвостовой ОС
* позволяет создавать только однородные вычислительные среды
* намного легче в плане ресурсов, чем виртуальная машина
* высокая скорость запуска

**Установка ОС Windows Server**

На хостовой машине должна быть поддержка UEFI (>2.3.1), а также поддержка безопасной загрузки

Подключение по сети Ethernet обязательно

**Виды образов**

* Microsoft AZURE

облачная платформа, которая содержит различные вспомогательные службы, а также облачное хранение данных, работа в сети и работа в приложениях

* ISO-образ

* VHD

Файл виртуальной машины с уже установленной виртуальной машины

При выборе языка и методе ввода: рекомендуется метод ввода выбирать стандартную раскладку клавиатуры типа США

Пароли администратора вводить обязательно на английской раскладке клавиатуры, потом система может запросить пароль, который будет соответствовать требованиям по безопасности

Также нужно изменить имя хостовой машины

Самой главной учетной записью является Administrator/Администратор. Остальных пользователей желательно выключить

Необходимо зайти в Локальную политику безопасности:

Политики учетных записей – Политика паролей

Должен быть сброшен:

* Максимальный срок действия пароля (0 дн.)
* Минимальный срок действия пароля (0 дн.)
* Минимальная длина (0 зн.)
* Пароль должен отвечать требованиям сложности (Отключён)