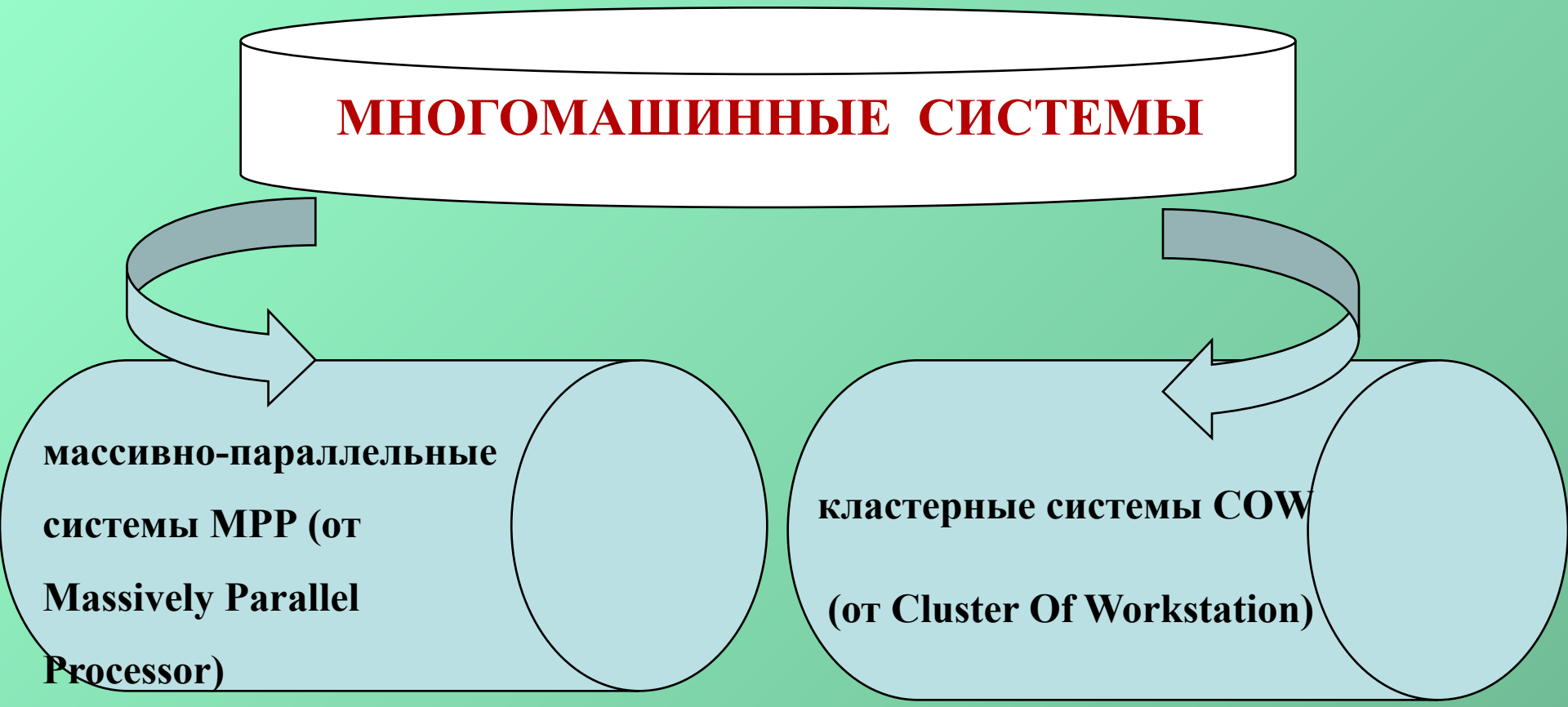
ТЕМА 1. АРХИТЕКТУРА И ФУНКЦИОНИРОВАНИЕ ПАРАЛЛЕЛЬНЫХ МНОГОПРОЦЕССОРНЫХ СИСТЕМ.

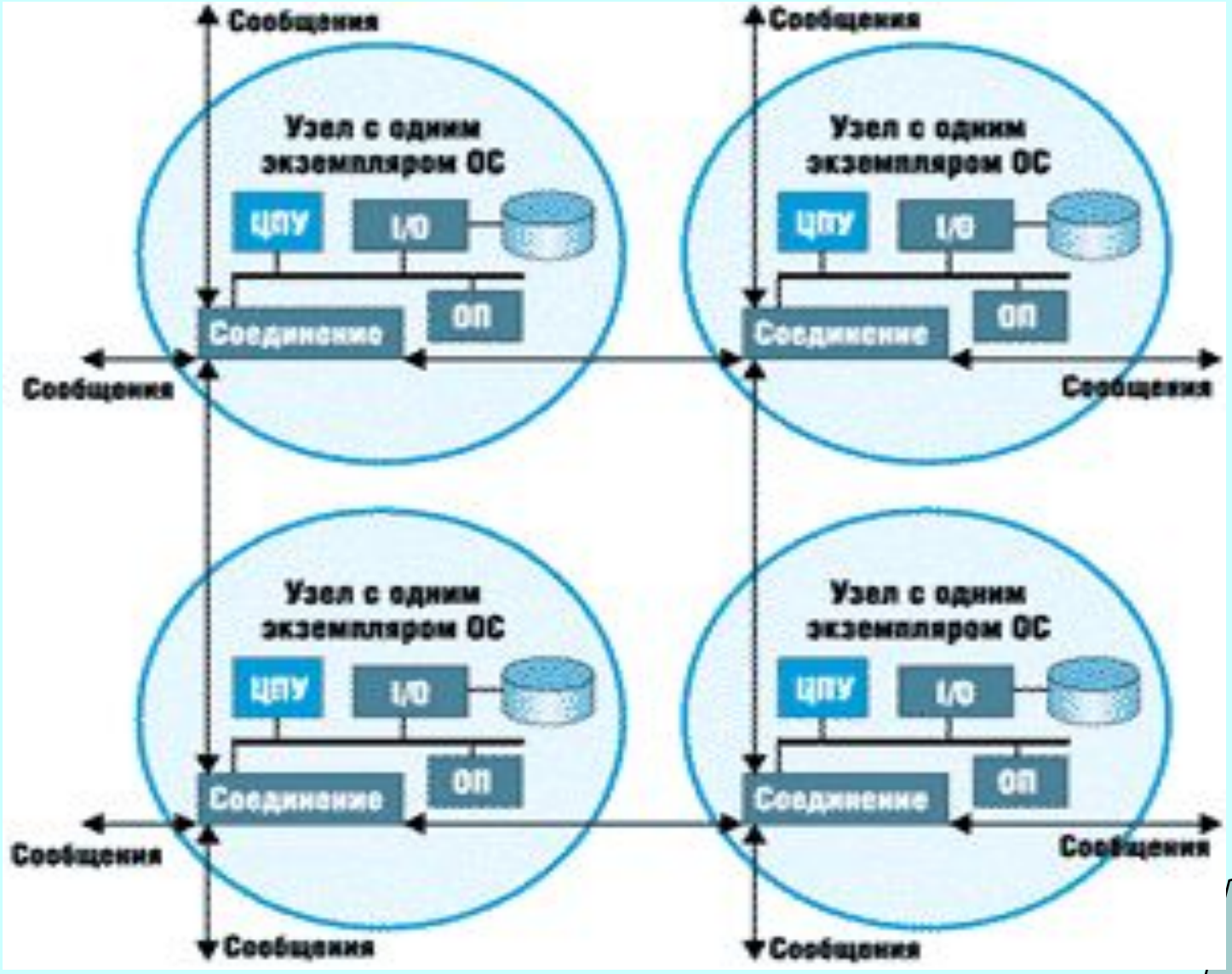
Лекция 4. Особенности архитектуры кластерных систем и процессоров с управлением потоком данных.

Первый вопрос.

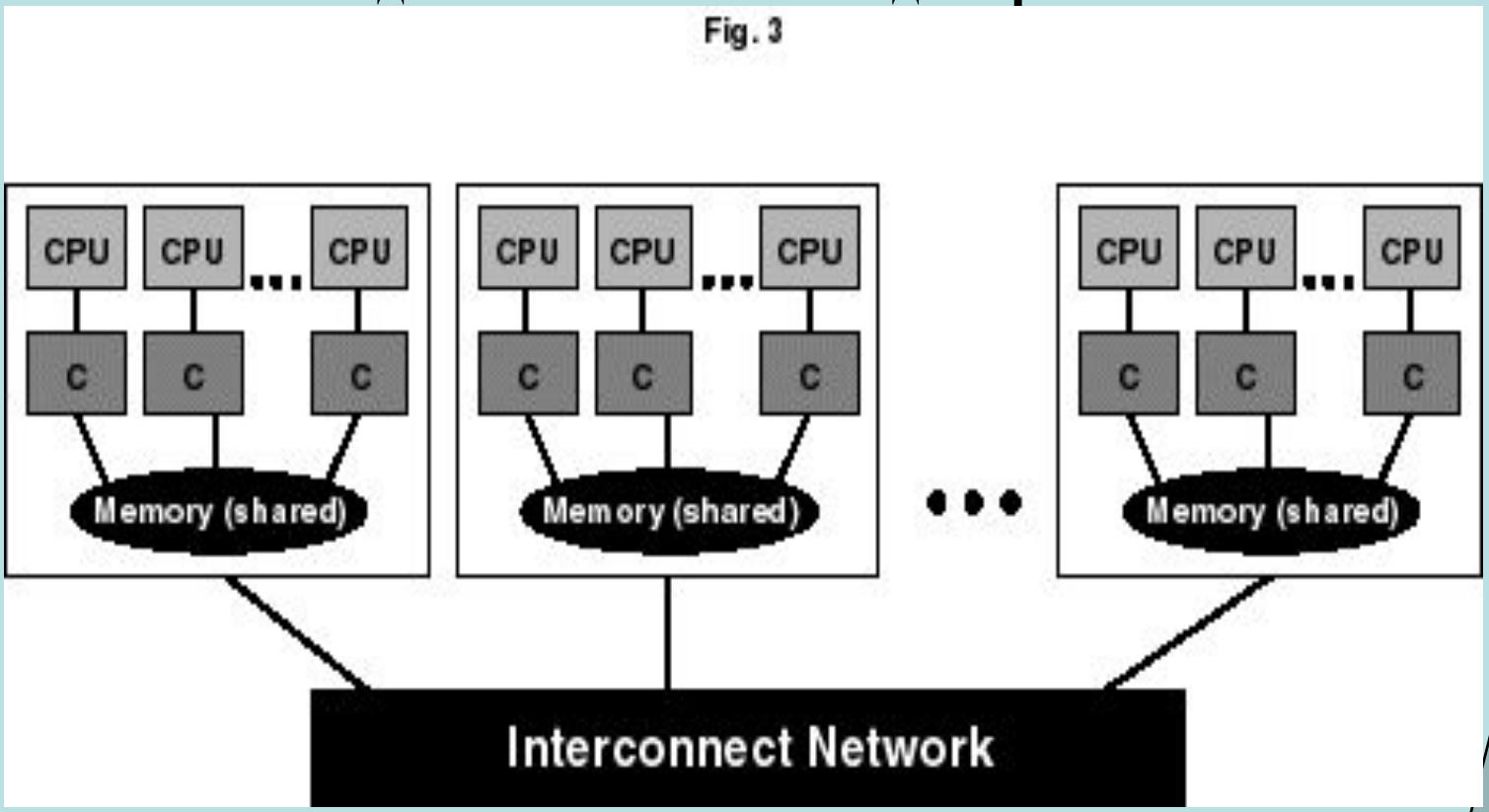
Кластерные системы COW (Cluster Of Workstation) .


МНОГОМАШИННЫЕ СИСТЕМЫ
массивно-параллельные системы MPP (от Massively Parallel Processor)
кластерные системы COW (от Cluster Of Workstation)

Массивно-параллельные системы MPP



Системы COW, или кластерные архитектуры, представляют собой объединение нескольких стандартных персональных компьютеров и/или серверов- посредством стандартных сетевых средств связи. Машины, входящие в кластерную систему, могут использоваться для совместного и одновременного выполнения



Кластер – это локальная (расположенная территориально в одном месте) вычислительная система, состоящая из множества независимых компьютеров и сети, связывающей их. Кроме того, кластер является локальной системой потому, что он управляется в рамках отдельного административного домена как единая компьютерная система.

Компьютерные узлы из которых он состоит, являются стандартными, универсальными (персональными) компьютерами, используемыми в различных областях и для разнообразных приложений.

Вычислительный узел может содержать либо один микропроцессор, либо несколько, образуя, в последнем случае, симметричную (SMP-) конфигурацию.

Сетевая компонента кластера может быть либо обычной локальной сетью, либо быть построена на основе специальных сетевых технологий, обеспечивающих сверхбыструю передачу данных между узлами кластера.

Сеть кластера предназначена для интеграции узлов кластера и, обычно, отделена от внешней сети, через которую осуществляется доступ пользователей к кластеру.

Программное обеспечение кластеров состоит из двух компонент:

- средств разработки/программирования и
- средств управления ресурсами.

К средствам разработки относятся компиляторы для языков, библиотеки различного назначения, средства измерения производительности, а также отладчики, что, всё вместе, позволяет строить параллельные приложения.

К программному обеспечению управления ресурсами относятся средства инсталляции, администрирования и планирования потоков работ.

Хотя для параллельной обработки существует очень много моделей программирования, но, на настоящий момент, доминирующим подходом является модель на основе “передачи сообщений” (message passing), реализованная в виде стандарта MPI (Message Passing Interface).

MPI – это библиотека функций, с помощью которых в программах на языках C или Фортран можно передавать сообщения между параллельными процессами, а также управлять этими процессами.
Альтернативами такому подходу являются языки на основе так называемого “глобального распределенного адресного пространства” (GPAS – global partitioned address space), типичными представителями которых являются языки HPF (High Performance Fortran) и UPC (Unified Parallel C).

Принципы построения быстрых сетей передачи данных

Выбор сетевой технологии зависит от ряда факторов, среди которых
- цена;
- скорость передачи данных;
- совместимость с другими аппаратными средствами и системным программным обеспечением;
- коммуникационные характеристики приложений, которые будут исполняться на кластере.

Технические характеристики сети, непосредственно связанные с передачей данных, выражаются в терминах задержки (latency) и широты полосы пропускания (bandwidth).

Задержка определяется как время, затрачиваемое на передачу данных от одного компьютера к другому, и включает в себя время, за которое программное обеспечение подготавливает сообщение, и непосредственно время передачи битов данных с компьютера на компьютер.

Широта полосы пропускания **есть** количество бит за секунду, которое может быть передано по участку сети.

Достижение низкой задержки и большой широты полосы пропускания требует применения эффективных коммуникационных протоколов, которые снижают издержки, вносимые программным обеспечением.

Коммуникационный протокол определяет правила и соглашения, которые используются двумя или более компьютерами для обмена данными по сети.

В кластерах используются как традиционные сетевые протоколы, предназначенные для Internet, так и протоколы, специально ориентированные на использование в кластерах.

Типичными IP (Internet Protocol)-протоколами являются TCP (Transmission Control Protocol) и UDP (User Datagram Protocol). Эти протоколы совместно с прикладным интерфейсом программиста (Application Programming Interface) на основе BSD-сокетов, были первой библиотекой для передачи сообщений для использования в кластерах.
Исследования по протоколам с малой задержкой привели к созданию стандарта VIA (Virtual Interface Architecture). В частности, существует реализация версии MPICH стандарта MPI, которая называется MVICH, работающая с использованием VIA.

Большой консорциум промышленных партнеров, включая Compaq, Dell, Hewlett-Packard, IBM и др., разработали и поддерживают стандарт Infiniband для передачи данных с малой задержкой. В Infiniband – архитектуре (см. Рис. 4) компьютеры связываются между собой на основе высокоскоростной, последовательной, расширяемой, переключаемой фабрики, работающей на основе передачи сообщений.

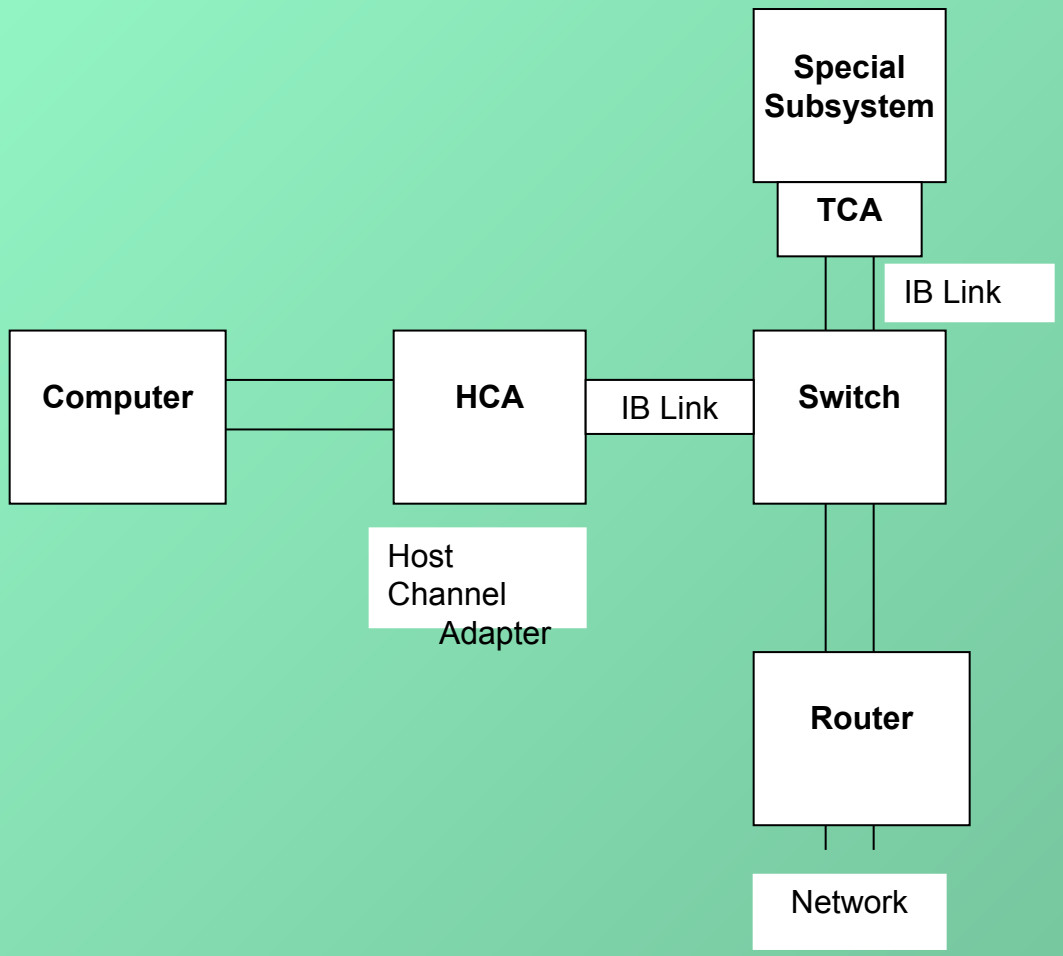


Рис. 4. Архитектура сети Infiniband.

Все системы и устройства подсоединяются к фабрике либо через HCA-адаптеры (host channel adapters), либо через TCA-адаптеры (target channel adapters). Скорость передачи данных по отдельной Infiniband-линии – свыше 2,5 Гб/сек.
Кроме того, в Infiniband поддерживается режим RDMA (Remote Direct Memory Access), который позволяет одному процессору обращаться к содержимому памяти другого процессора непосредственно.

Исчисление истории кластеров можно начать от первого проекта, в котором одной из основных целей являлось установление связи между компьютерами, – проекта ARPANET1). Именно тогда были заложены первые, оказавшиеся фундаментальными, принципы, приведшие впоследствии к созданию локальных и глобальных вычислительных сетей и, конечно же, всемирной глобальной компьютерной сети Интернет.

Первый в мире кластер - Beowulf, созданный под руководством Томаса Стерлинга и Дона Бекера в научно-космическом центре NASA – Goddard Space Flight Center – летом 1994 года.

Состав: 16 компьютеров на базе процессоров 486DX4 с тактовой частотой 100 MHz. Каждый узел имел 16 Mb оперативной памяти. Связь узлов обеспечивалась тремя параллельно работавшими 10 Mbit/s сетевыми адаптерами. Кластер функционировал под управлением операционной системы Linux, использовал GNU-компилятор и поддерживал параллельные программы на основе MPI.

В настоящее время под кластером типа Beowulf понимается система, которая состоит из одного серверного узла и одного или более клиентских узлов, соединенных при помощи Ethernet или некоторой другой сети.

Это система, построенная из готовых серийно выпускающихся промышленных компонентов, на которых может работать ОС Linux, стандартных адаптеров Ethernet и коммутаторов.

Она не содержит специфических аппаратных компонентов и легко воспроизводима. Серверный узел управляет всем кластером и является файл-сервером для клиентских узлов. Он также является консолью кластера и шлюзом во внешнюю сеть.

Большие системы Beowulf могут иметь более одного серверного узла, а также, возможно, специализированные узлы, например консоли или станции мониторинга.

В большинстве случаев клиентские узлы в Beowulf пассивны. Они конфигурируются и управляются серверными узлами и выполняют только то, что предписано серверным узлом.

Кластер Thunder

В настоящий момент число систем, собранных на основе процессоров корпорации Intel и представленных в списке Top 500, составляет **318**.

Самый мощный суперкомпьютер, представляющий собой кластер на основе Intel Itanium2, установлен в Ливерморской национальной лаборатории (США).

Аппаратная конфигурация кластера Thunder (http://www.llnl.gov/linux/thunder/):

- 1024 сервера, по 4 процессора Intel Itanium 1.4 GHz в каждом;
- 8 Gb оперативной памяти на узел;
- общая емкость дисковой системы 150 Tb.

Программное обеспечение:

- операционная система CHAOS 2.0;
- среда параллельного программирования MPICH2;
- отладчик параллельных программ TotalView;
- Intel и GNU Fortran, C/C++ компиляторы.

В данное время кластер Thunder с пиковой производительностью 22938 GFlops и максимально показанной на тесте LINPACK 19940 Gflops.

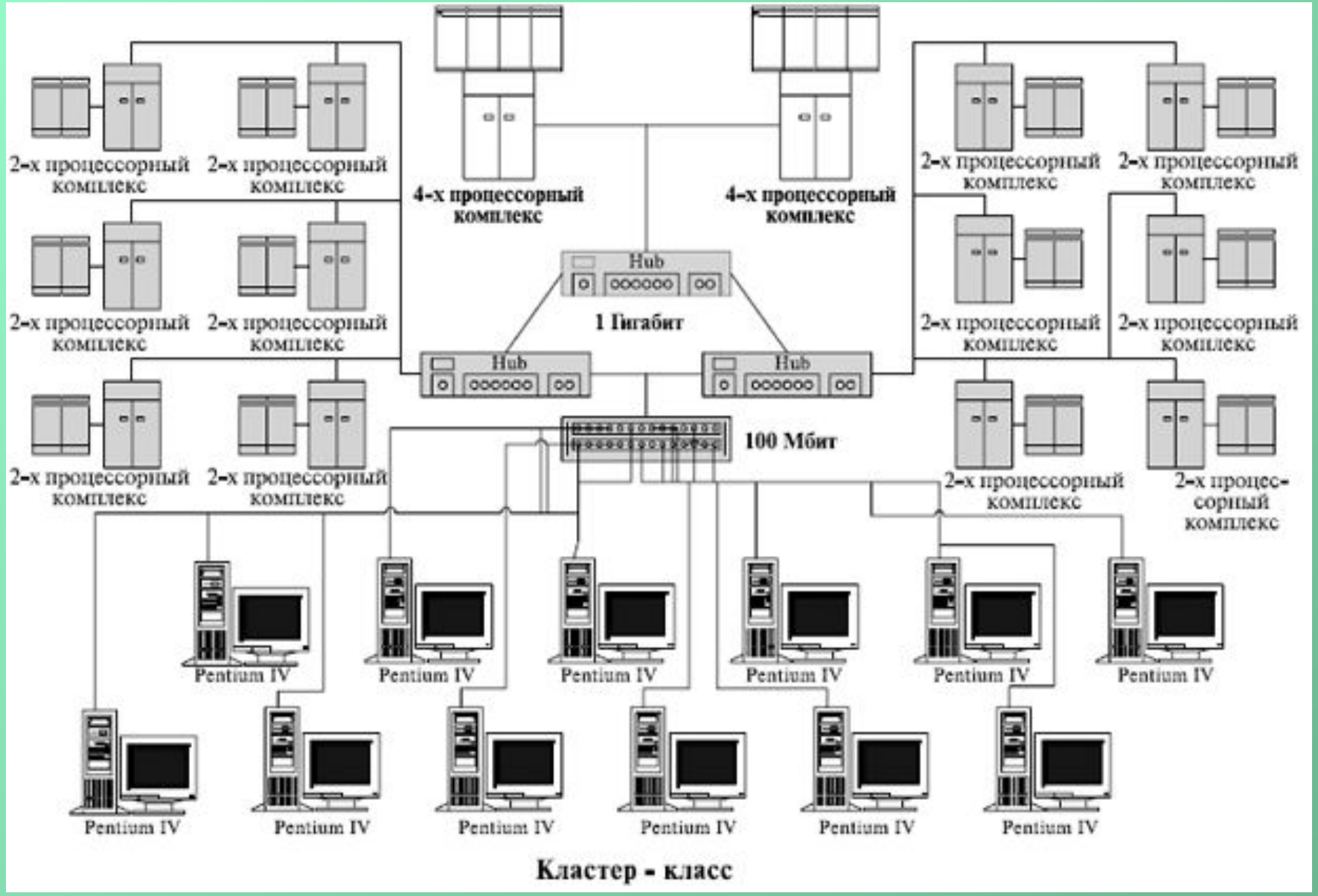
Высокопроизводительный вычислительный кластер ННГУ

В качестве следующего примера рассмотрим вычислительный кластер Нижегородского университета, оборудование для которого было передано в рамках Академической программы Интел в 2001 г.

В состав кластера входят (см. рис. 4):

- 2 вычислительных сервера, каждый из которых имеет 4 процессора Intel Pentium III 700 MHZ, 512 MB RAM, 10 GB HDD, 1 Gbit Ethernet card;
- 12 вычислительных серверов, каждый из которых имеет 2 процессора Intel Pentium III 1000 MHZ, 256 MB RAM, 10 GB HDD, 1 Gbit Ethernet card;
- 12 рабочих станций на базе процессора Intel Pentium 4 1300 MHZ, 256 MB RAM, 10 GB HDD, 10/100 Fast Ethernet card.
- **Важной отличительной особенностью кластера является его неоднородность (гетерогенность)**. В состав кластера входят рабочие места, оснащенные процессорами Intel Pentium 4 и соединенные относительно медленной сетью (100 Мбит), а также вычислительные 2- и 4-процессорные серверы, обмен данными между которыми выполняется при помощи **быстрых каналов передачи данных (1000 Мбит).** В результате кластер может использоваться не только для решения сложных вычислительно-трудоемких задач, но также и для проведения различных экспериментов по исследованию многопроцессорных кластерных систем и параллельных методов решения научно-технических задач.

Структура вычислительного кластера Нижегородского университета



Второй вопрос.
Особенности архитектуры и организации процесса в параллельных процессорах с управлением потоком данных.

"Активная" команда
"Пассивная" команда
Обобщенная структура параллельного процессора
Недостатки и достоинства процессоров с управлением потоком данных

Семан

Код операции (OP)	Адрес получателя (SA)
Тег 1	Данное 1
Тег 2	Данное 2

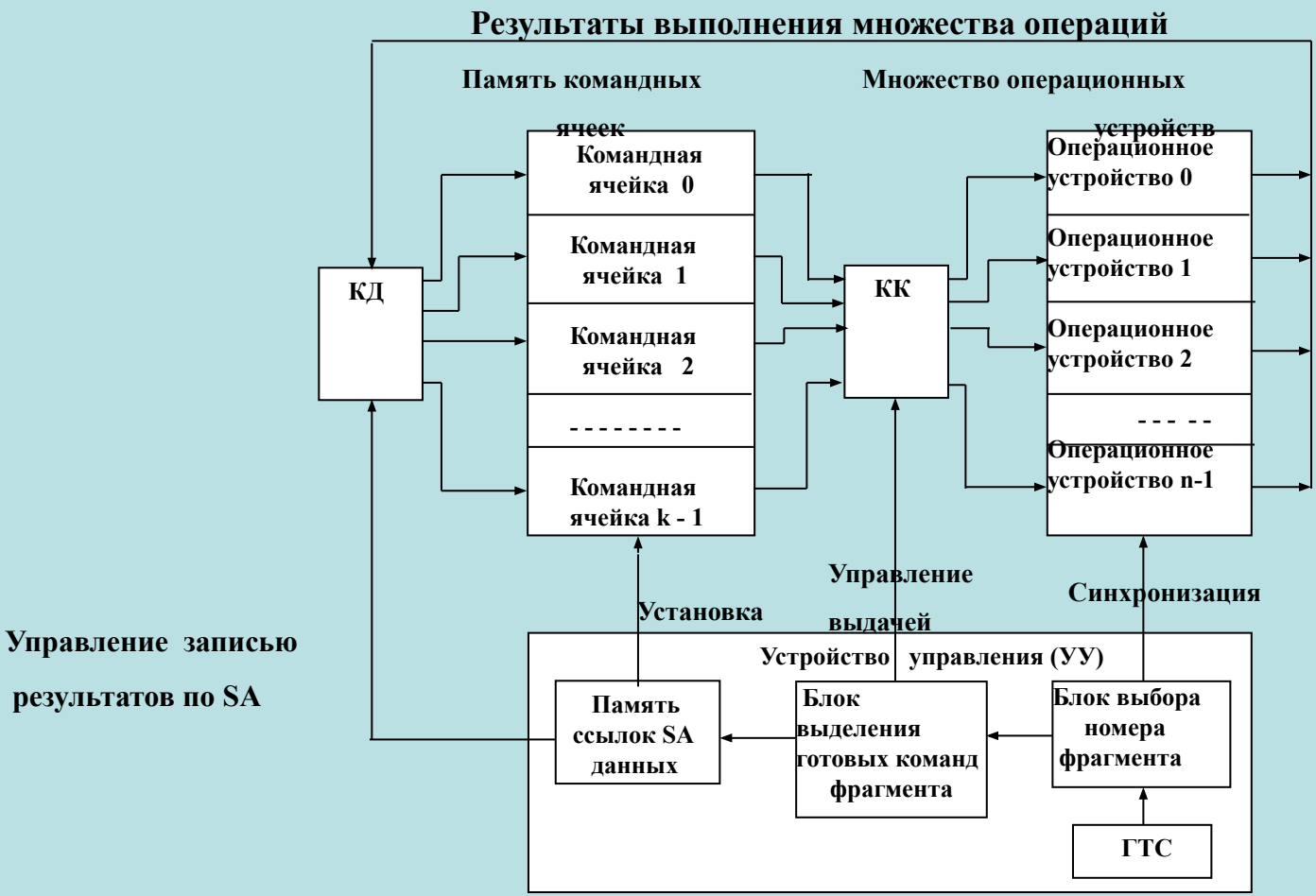
ой ячейки:

- **код операции** определяет тип выполняемой операции задачи;
- **адрес получателя** – номер командной ячейки, использующей результат выполнения текущей операции;
- **данное 1 и данное 2** – значения первого и второго операндов выполняемой (в данном случае двуместной) операции;
- **Тег 1 и Тег 2** – признаки наличия значений всех данных, необходимых для начала выполнения операции (до начала решения задачи устанавливаются значения тегов Тег 1 = Тег 2 = «готов» только для команд, зависящих от входных данных (то есть для команд «готовых» к выполнению), и «не готов» - для всех остальных

Суть принципа организации параллельного процесса на основе «управления потоком данных»:

1. Использование исходного текста задачи в виде традиционной последовательной программы, записанной на языке последовательного программирования.
2. Использование понятий «активная» команда (готовая к выполнению) при наличии значений всех ее операндов или «пассивная» команда (не готовая к выполнению) - при отсутствии значения хотя бы одного из операндов.
3. Присваивание всем командам, зависящим только от исходных данных, состояния активности, перевод остальных команд из исходного пассивного состояния в активное состояние (то есть формирование подмножеств активных команд) в динамике вычислительного процесса путем учета количества операндов каждой команды, для которых вычислены значения в процессе решения задачи.
4. Одновременное (параллельное) начало реализации всех активных команд

Обобщенная структура потокового процессора



Основные компоненты архитектуры потоковой ЭВМ :

- **память командных ячеек;**
- **множество операционных устройств (функциональных блоков или универсальных процессоров с номерами 0…n - 1), выполняющих операции над операндами командных ячеек;**
- **распределительное устройство данных (КД), обеспечивающее запись результатов выполнения множества операций в память командных ячеек в соответствии с множеством $SA = \{ SAj \}$ адресов SAj ;**
- **распределительное устройство команд (КК), реализующее функцию одновременного назначения готовых к выполнению команд на свободные операционные устройства;**

Параллельное выполнение программы сводится к итеративному процессу, при котором каждая итерация включает следующие шаги:

- **анализ значений тегов и выделение подмножества команд, готовых к выполнению в рассматриваемый момент времени;**
- **распределение подмножества готовых к реализации команд между свободными функциональными блоками/процессорами;**
- **одновременное (параллельное) выполнение подмножества готовых команд функциональными блоками соответствующих типов;**
- **занесение результата выполнения каждой из одновременно выполняемых команд в командную ячейку, с номером, определяемым значением поля «адрес получателя» в качестве**

Достоинствами ЭВМ с управлением потоком данных по сравнению с параллельными ЭВМ других классов являются [1,4,5]:

- **высокий параллелизм обработки данных, приближающийся к возможностям выбранного математического метода решения задачи;**
- **специализация отдельных функциональных блоков исполнительного устройства, что позволяет сделать их максимально быстродействующими.**

Недостатками ЭВМ с управлением потоком данных считаются [5]:

- **невозможность одновременного выполнения нескольких программ и снижение эффективности при наличии в задачах условных переходов;**
- **возможность практического применения управления потоком данных только для параллельной реализации скалярных вычислений и**